Econometrics Templates

Sven Otto

July 29, 2024

Table of contents

Welcome

This site contains different lecture note templates.

Part I

Stock and Watson Applications

1 Empirical Applications of Linear & Nonlinear Regressions

This chapter introduces the basics in linear and nonlinear regression models and shows how to perform regression analysis in R.

The following packages are needed for reproducing the code presented in this chapter:

- `AER` - accompanies the Book Applied Econometrics with R by C. Kleiber and Zeileis (2008) and provides useful functions and data sets.
- `MASS` - a collection of functions for applied statistics.
- `stargazer` - used for creating well-formatted regression and summary statistics tables (Hlavac 2022)

```
library(AER)
library(MASS)
library(stargazer)
```

1.1 Data Set Description

The California School data set (CASchools) is included in the R package "AER". This dataset contains information on various characteristics of schools in California, such as test scores, teacher salaries, and student demographics. It's commonly used in econometrics and statistical analysis to explore relationships between these variables and to illustrate various modeling techniques.

```
# load the the data set
data(CASchools)
# get an overview
summary(CASchools)
```

```
   district            school                   county      grades   
 Length:420         Length:420         Sonoma     : 29   KK-06: 61  
 Class :character   Class :character   Kern       : 27   KK-08:359  
 Mode  :character   Mode  :character   Los Angeles: 27              
                                       Tulare     : 24              
                                       San Diego  : 21              
                                       Santa Clara: 20              
                                       (Other)    :272              
    students          teachers          calworks          lunch       
 Min.   :   81.0   Min.   :   4.85   Min.   : 0.000   Min.   :  0.00  
 1st Qu.:  379.0   1st Qu.:  19.66   1st Qu.: 4.395   1st Qu.: 23.28  
 Median :  950.5   Median :  48.56   Median :10.520   Median : 41.75  
 Mean   : 2628.8   Mean   : 129.07   Mean   :13.246   Mean   : 44.71  
 3rd Qu.: 3008.0   3rd Qu.: 146.35   3rd Qu.:18.981   3rd Qu.: 66.86  
 Max.   :27176.0   Max.   :1429.00   Max.   :78.994   Max.   :100.00  

    computer       expenditure       income          english      
 Min.   :   0.0   Min.   :3926   Min.   : 5.335   Min.   : 0.000  
 1st Qu.:  46.0   1st Qu.:4906   1st Qu.:10.639   1st Qu.: 1.941  
 Median : 117.5   Median :5215   Median :13.728   Median : 8.778  
 Mean   : 303.4   Mean   :5312   Mean   :15.317   Mean   :15.768  
 3rd Qu.: 375.2   3rd Qu.:5601   3rd Qu.:17.629   3rd Qu.:22.970  
 Max.   :3324.0   Max.   :7712   Max.   :55.328   Max.   :85.540  

      read            math      
 Min.   :604.5   Min.   :605.4  
 1st Qu.:640.4   1st Qu.:639.4  
 Median :655.8   Median :652.5  
 Mean   :655.0   Mean   :653.3  
 3rd Qu.:668.7   3rd Qu.:665.9  
 Max.   :704.0   Max.   :709.5  
```

Upon examination we find that the dataset contains mostly numeric variables, but it lacks two important ones we're interested in: **average test scores** and **student-teacher ratios**. However, we can calculate them using the available data. To find the student-teacher ratio, we divide the total number of students by the number of teachers. For the average test score, we just need to average the math and reading scores. In the next code chunk, we'll demonstrate how to create these variables as vectors and add them to the `CASchools` dataset.

```
# compute student-teacher ratio and append it to CASchools
CASchools$STR <- CASchools$students/CASchools$teachers
```

```r
# compute test score and append it to CASchools
CASchools$score <- (CASchools$read + CASchools$math)/2
```

If we ran `summary(CASchools)` again we would find the two variables of interest as additional variables named `STR` and `score`.

2 Linear Regression

Let's suppose we were interested in the following regression model

$$TestScore = \beta_0 + \beta_1\, STR + \beta_2\, english + u$$

In this regression, we aim to explore how test scores (`TestScore`) are influenced by student-teacher ratio (`STR`) and the percentage of English learners (`english`). The variable `english` indicates the proportion of students who may require additional support or resources to improve their English language skills within each school.

We would run this model in R using the `lm()` function and explore the regression estimates with `coeftest()`.

```
# run the model
model <- lm(score ~ STR + english, data = CASchools)
# report estimates
coeftest(model, vcov. = vcovHC, type = "HC1")
```

```
t test of coefficients:

              Estimate Std. Error  t value Pr(>|t|)
(Intercept) 686.032245   8.728225  78.5993  < 2e-16 ***
STR          -1.101296   0.432847  -2.5443  0.01131 *
english      -0.649777   0.031032 -20.9391  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.0.1 Hypothesis Tests and Confidence Intervals for a Single Coefficient

The `coeftest()` function in R, along with suitable options such as `vcov. = vcovHC` for robust standard errors, automatically includes statistics such as standard errors, t-statistics, and p-values, which is exactly what we need to test hypotheses about single coefficients (β_j) in regression models.

We could also manually check these values calculating the t-statistics or p-values using the provided output above and using R as a calculator. For example, using the definition of the p-value for a two-sided test, we can confirm the p-value for a test of the hypothesis that the coefficient β_1, which represents the coefficient on`STR`, is approximately 0.01

```
# compute two-sided p-value
2 * (1 - pt(abs(coeftest(model, vcov. = vcovHC, type = "HC1")[2, 3]),
            df = model$df.residual))
```

```
[1] 0.01130921
```

We can also compute confidence intervals for individual coefficients in the multiple regression model by using the function `confint()`. This function computes confidence intervals at the 95% level by default.

```
# compute confidence intervals for all coefficients in the model
confint(model)
```

```
                  2.5 %      97.5 %
(Intercept) 671.4640580 700.6004311
STR          -1.8487969  -0.3537944
english      -0.7271113  -0.5724424
```

To obtain confidence intervals at a different level, say 90%, we set the argument `level` in our call of `confint()` accordingly.

```
confint(model, level = 0.9)
```

```
                    5 %        95 %
(Intercept) 673.8145793 698.2499098
STR          -1.7281904  -0.4744009
english      -0.7146336  -0.5849200
```

A limitation of using `confint()` is its failure to incorporate robust standard errors when computing confidence intervals. To address this, you can manually generate large-sample confidence intervals that consider robust standard errors with the following method.

```r
# compute robust standard errors
rob_se <- diag(vcovHC(model, type = "HC1"))^0.5

# compute robust 95% confidence intervals
rbind("lower" = coef(model) - qnorm(0.975) * rob_se,
      "upper" = coef(model) + qnorm(0.975) * rob_se)
```

```
      (Intercept)        STR    english
lower    668.9252 -1.9496606 -0.7105980
upper    703.1393 -0.2529307 -0.5889557
```

```r
# compute robust 90% confidence intervals
rbind("lower" = coef(model) - qnorm(0.95) * rob_se,
      "upper" = coef(model) + qnorm(0.95) * rob_se)
```

```
      (Intercept)        STR    english
lower    671.6756 -1.8132659 -0.7008195
upper    700.3889 -0.3893254 -0.5987341
```

The output above shows that zero is not an element of the confidence interval for the coefficient on `STR`, so we can reject the null hypothesis at significance levels of 5% and 10% (Note that rejection at the 5% level implies rejection at the 10% level anyway). We can bring this conclusion further via the p-value for `STR`: $0.00398 < 0.01$, which indicates that this coefficient estimate is significant at the 1% level.

2.0.2 Joint Hypothesis Testing

Let's suppose now that we are interested in investigating the average effect on test scores of reducing the student-teacher ratio when the expenditures per pupil and the percentage of english learning pupils are held constant. Let us augment our model by an additional regressor `expenditure`, that is a measure for the total expenditure per pupil in the district. For this model, we will include `expenditure` as measured in thousands of dollars. Our new model would be

$$TestScore = \beta_0 + \beta_1\, STR + \beta_2\, english + \beta_3\, expenditure + u$$

Let us now estimate the model:

```
# scale expenditure to thousands of dollars
CASchools$expenditure <- CASchools$expenditure/1000

# estimate the model
model <- lm(score ~ STR + english + expenditure, data = CASchools)
coeftest(model, vcov. = vcovHC, type = "HC1")
```

```
t test of coefficients:

              Estimate Std. Error  t value Pr(>|t|)
(Intercept) 649.577947  15.458344  42.0212  < 2e-16 ***
STR          -0.286399   0.482073  -0.5941  0.55277
english      -0.656023   0.031784 -20.6398  < 2e-16 ***
expenditure   3.867901   1.580722   2.4469  0.01482 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated impact of a one-unit change in the student-teacher ratio on test scores, while holding expenditure and the proportion of English learners constant, is -0.29. It is much smaller than the estimated coefficient in our initial model where we didn't include `expenditure`. Additionally, this coefficient of `STR` is no longer statistically significant, even at a 10% significance level, as indicated by a p-value of 0.55. This lack of significance for β_1 may stem from a larger standard error resulting from the inclusion of expenditure in the model, leading to less precise estimation of the coefficient on STR. This scenario highlights the challenge of dealing with strongly correlated predictors, known as imperfect multicollinearity. The correlation between STR and $expenditure$ can be determined using the `cor()` function.

```
# compute the sample correlation between 'STR' and 'expenditure'
cor(CASchools$STR, CASchools$expenditure)
```

```
[1] -0.6199822
```

This indicates a moderately strong negative correlation between the two variables.

The estimated model is

$$\widehat{TestScore} = \underset{(15.21)}{649.58} - \underset{(0.48)}{0.29}\, STR - \underset{(0.04)}{0.66}\, english + \underset{(1.41)}{3.87}\, expenditure$$

Could we reject the hypothesis that *both* the STR coefficient and the $expenditure$ coefficient are zero? To answer this, we need to conduct **joint hypothesis tests**, which involve placing restrictions on multiple regression coefficients. This differs from individual t-tests, where restrictions are applied to a single coefficient.

To test whether both coefficients are zero, we will conduct an F**-test**. To do this in R, we can use the function `linearHypothesis()` contained in the package `car`.

```
# execute the function on the model object and provide both linear restrictions
# to be tested as strings
linearHypothesis(model, c("STR=0", "expenditure=0"))
```

```
Linear hypothesis test

Hypothesis:
STR = 0
expenditure = 0

Model 1: restricted model
Model 2: score ~ STR + english + expenditure

  Res.Df   RSS Df Sum of Sq      F   Pr(>F)
1    418 89000
2    416 85700  2    3300.3 8.0101 0.000386 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The output reveals that the F-statistic for this joint hypothesis test is 8.01 and the corresponding p-value is about 0.0004. We can therefore reject the null hypothesis that both coefficients are zero at the 0.1% level of significance.

A **heteroskedasticity-robust version** of this F-test (which leads to the same conclusion) can be conducted as follows:

```
# heteroskedasticity-robust F-test
linearHypothesis(model, c("STR=0", "expenditure=0"), white.adjust = "hc1")
```

```
Linear hypothesis test

Hypothesis:
STR = 0
expenditure = 0

Model 1: restricted model
```

```
Model 2: score ~ STR + english + expenditure

Note: Coefficient covariance matrix supplied.

  Res.Df Df      F   Pr(>F)
1    418
2    416  2 5.4337 0.004682 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The standard output of a model summary in R also reports an F-statistic and the corresponding p-value. This F-test examines whether all of the population coefficients in the model except for the intercept are zero, so the hypotheses would be $H_0 : \beta_1 = 0, \beta_2 = 0, \beta_3 = 0$ vs. $H_1 : \beta_j \neq 0$ for at least one $j = 1, 2, 3$.

We now check whether the F-statistic belonging to the p-value listed in the model's summary matches with the result reported by `linearHypothesis()`

```r
# execute the function on the model object and provide the restrictions
# to be tested as a character vector
linearHypothesis(model, c("STR=0", "english=0", "expenditure=0"))
```

```
Linear hypothesis test

Hypothesis:
STR = 0
english = 0
expenditure = 0

Model 1: restricted model
Model 2: score ~ STR + english + expenditure

  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    419 152110
2    416  85700  3     66410 107.45 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Access the overall F-statistic from the model's summary
summary(model)$fstatistic
```

```
   value    numdf    dendf 
107.4547   3.0000 416.0000 
```

Both results match. The F-test rejects the null hypothesis that the model has no power in explaining test scores. It is nevertheless important to note that the F-statistic reported by summary is not robust to heteroskedasticity.

2.1 Multiple Regression

In order to reduce the risk of omitted variable bias, it is essential to include control variables in regression models. In our case, we are interested in estimating the causal effect of a change in the student-teacher ratio on test scores. We will now see an example of how to use multiple regression in order to alleviate omitted variable bias and how to report these results using R.

By including *english* as control variable, we aimed to control for unobservable student characteristics which correlate with the student-teacher ratio and are assumed to have an impact on test score. But there are other interesting variables to observe:

- `lunch`: the share of students that qualify for a subsidized or even a free lunch at school.
- `calworks`: the percentage of students that qualify for the *CalWorks* income assistance program.

Students eligible for *CalWorks* live in families with a total income below the threshold for the subsidized lunch program, so both variables are indicators for the share of economically disadvantaged children. We suspect both indicators are highly correlated.

```
# estimate the correlation between 'calworks' and 'lunch'
cor(CASchools$calworks, CASchools$lunch)
```

```
[1] 0.7394218
```

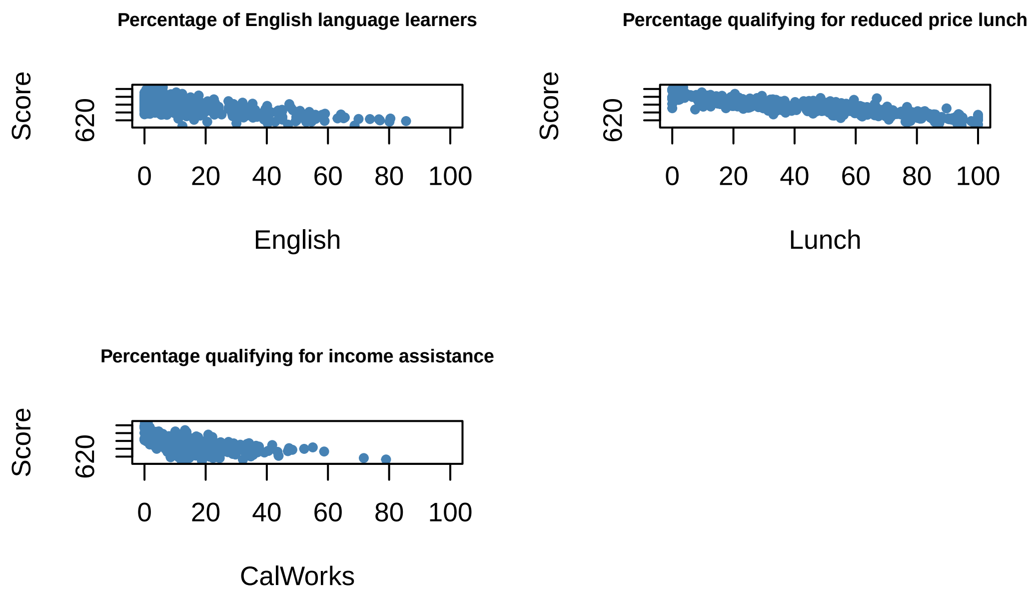
If they are highly correlated as we just confirmed, there is no standard way to proceed when deciding which variable to use. In any case it may not be a good idea to use both variables as regressors in view of collinearity. Let's first explore further these control variables and how they correlate with the dependent variable by plotting them against test scores. When computing simultaneously several plots, we may use `layout()` to divide the plotting area and the matrix `m` to specify the location of the plots (see `?layout`).

```
# set up arrangement of plots
m <- rbind(c(1, 2), c(3, 0))
graphics::layout(mat = m)

# scatterplots
plot(score ~ english,
     data = CASchools,
     col = "steelblue",
     pch = 20,
     xlim = c(0, 100),
     cex.main = 0.7,
     xlab="English",
     ylab="Score",
     main = "Percentage of English language learners")

plot(score ~ lunch,
     data = CASchools,
     col = "steelblue",
     pch = 20,
     cex.main = 0.7,
     xlab="Lunch",
     ylab="Score",
     main = "Percentage qualifying for reduced price lunch")

plot(score ~ calworks,
     data = CASchools,
     col = "steelblue",
     pch = 20,
     xlim = c(0, 100),
     cex.main = 0.7,
     xlab="CalWorks",
     ylab="Score",
     main = "Percentage qualifying for income assistance")
```

We observe negative relationships. Let's check the correlation coefficients.

```
# estimate correlation between student characteristics and test scores
cor(CASchools$score, CASchools$english)
```

```
[1] -0.6441238
```




```r
cor(CASchools$score, CASchools$lunch)
```

```
[1] -0.868772
```

```r
cor(CASchools$score, CASchools$calworks)
```

```
[1] -0.6268533
```

We shall consider five different model equations:

$$\text{TestScore} = \beta_0 + \beta_1 \, \text{STR} + u, \tag{2.1}$$

$$\text{TestScore} = \beta_0 + \beta_1 \, \text{STR} + \beta_2 \, \text{english} + u, \tag{2.2}$$

$$\text{TestScore} = \beta_0 + \beta_1 \, \text{STR} + \beta_2 \, \text{english} + \beta_3 \, \text{lunch} + u, \tag{2.3}$$

$$\text{TestScore} = \beta_0 + \beta_1 \, \text{STR} + \beta_2 \, \text{english} + \beta_4 \, \text{calworks} + u, \tag{2.4}$$

$$\text{TestScore} = \beta_0 + \beta_1 \, \text{STR} + \beta_2 \, \text{english} + \beta_3 \, \text{lunch} + \beta_4 \, \text{calworks} + u. \tag{2.5}$$

The best way to report regression results is in a table. The `stargazer` package is very convenient for this purpose. It provides a function that generates professionally looking HTML and LaTeX tables that satisfy scientific standards. One simply has to provide one or multiple object(s) of class `lm`. The rest is done by the function `stargazer()`.

```r
# load the stargazer library
library(stargazer)

# estimate different model specifications
spec1 <- lm(score ~ STR, data = CASchools)
spec2 <- lm(score ~ STR + english, data = CASchools)
spec3 <- lm(score ~ STR + english + lunch, data = CASchools)
spec4 <- lm(score ~ STR + english + calworks, data = CASchools)
spec5 <- lm(score ~ STR + english + lunch + calworks, data = CASchools)

# gather robust standard errors in a list
rob_se <- list(sqrt(diag(vcovHC(spec1, type = "HC1"))),
               sqrt(diag(vcovHC(spec2, type = "HC1"))),
               sqrt(diag(vcovHC(spec3, type = "HC1"))),
               sqrt(diag(vcovHC(spec4, type = "HC1"))),
               sqrt(diag(vcovHC(spec5, type = "HC1"))))

# generate a LaTeX table using stargazer
stargazer(spec1, spec2, spec3, spec4, spec5,
          se = rob_se,
          type = "text",
          digits = 3,
          header = F,
          column.labels = c("(I)", "(II)", "(III)", "(IV)", "(V)"))
```

```
=============================================================================================
                                                                        Dependent variable:
                    -------------------------------------------------------------------------
                                                                               score
                              (I)                     (II)                    (III)
                              (1)                     (2)                      (3)
---------------------------------------------------------------------------------------------
STR                        -2.280***                -1.101**                -0.998***
                            (0.519)                 (0.433)                  (0.270)

english                                            -0.650***                -0.122***
                                                    (0.031)                  (0.033)

lunch                                                                       -0.547***
                                                                             (0.024)
```

```
calworks


Constant                    698.933***              686.032***              700.150***
                             (10.364)                (8.728)                 (5.568)

-----------------------------------------------------------------------------------------------
Observations                   420                     420                     420
R2                            0.051                   0.426                   0.775
Adjusted R2                   0.049                   0.424                   0.773
Residual Std. Error    18.581 (df = 418)       14.464 (df = 417)       9.080 (df = 416)
F Statistic         22.575*** (df = 1; 418) 155.014*** (df = 2; 417) 476.306*** (df = 3; 416
===============================================================================================
Note:
```

Each column in this table contains most of the information provided also by `coeftest()` and `summary()` for each of the models under consideration. Each of the coefficient estimates includes its standard error in parenthesis and one, two or three asterisks representing their significance levels. Although t-statistics are not reported, one may compute them manually simply by dividing a coefficient estimate by the corresponding standard error. At the bottom of the table summary statistics for each model and a legend are reported.

From the model comparison we observe that including control variables approximately cuts the coefficient on STR in half. Additionally, the estimation seems to remain unaffected by the specific set of control variables employed. Thus, the inference drawn is that, under all other conditions held constant, reducing the student-teacher ratio by one unit is associated with an estimated average rise in test scores of roughly 1 point.

Incorporating student characteristics as controls increased both R^2 and $\bar{R}^2$ from about 0.05 (spec1) to about 0.77 (spec3 and spec5), indicating these variables' suitability as predictors for test scores.

We also observe that the coefficients for the control variables are not significant in all models. For example in spec5, the coefficient on *calworks* is not significantly different from zero at the 10% level.

Lastly, we see that the effect on the estimate (and its standard error) of the coefficient on STR when adding *calworks* to the base specification spec3 is minimal. Hence, we can identify calworks as an unnecessary control variable, especially considering the incorporation of *lunch* in this model.

3 Nonlinear Regression Functions

Sometimes a nonlinear regression function is better suited for estimating the population relationship between the regressor X and the regressand Y. Let's have a look at an example that explores the relationship between the income of schooling districts and their test scores.

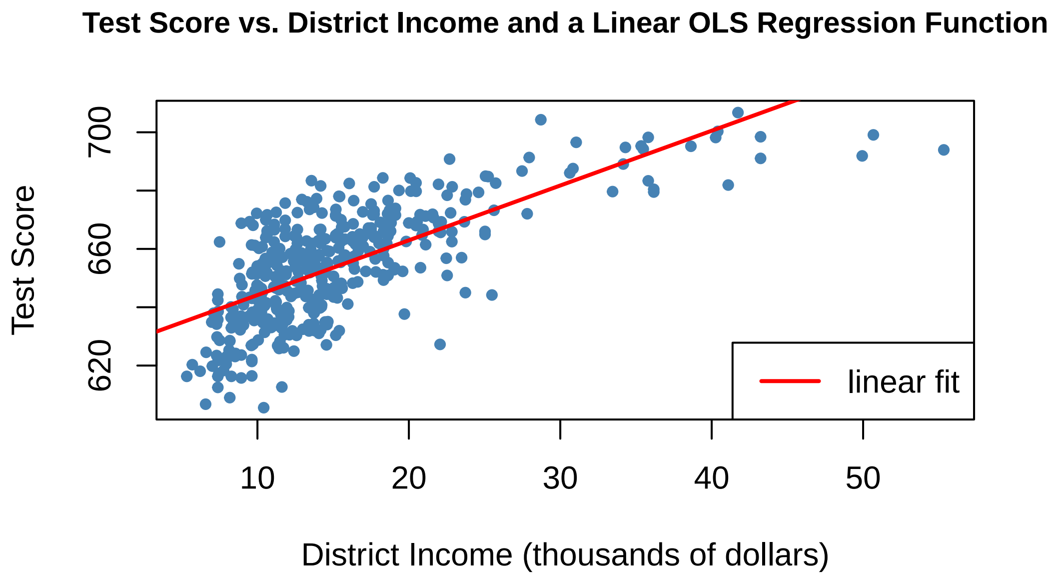
We start our analysis by computing the correlation between both variables.

```
cor(CASchools$income, CASchools$score)
```

```
[1] 0.7124308
```

Income and test score are positively correlated: school districts with above-average income tend to achieve above-average test scores. But does a linear regression adequately model the data? To investigate this further, let's visualize the data by plotting it and adding a linear regression line.

```
# fit a simple linear model
linear_model<- lm(score ~ income, data = CASchools)

# plot the observations
plot(CASchools$income, CASchools$score,
     col = "steelblue",
     pch = 20,
     xlab = "District Income (thousands of dollars)",
     ylab = "Test Score",
     cex.main = 0.9,
     main = "Test Score vs. District Income and a Linear OLS Regression Function")

# add the regression line to the plot
abline(linear_model,
       col = "red",
       lwd = 2)
legend("bottomright", legend="linear fit",lwd=2,col="red")
```



The plot shows that the linear regression line seems to overestimate the true relationship when income is either very high or very low and it tends to underestimates it for the middle income group. Luckily, Ordinary Least Squares (OLS) isn't limited to linear regressions of the predictors. We have the flexibility to model test scores as a function of income and the square of income. This leads us to the following regression model:

$$TestScore_i = \beta_0 + \beta_1 \, income_i + \beta_2 \, income_i^2 + u_i$$

which is a *quadratic regression model*. Here we treat $income^2$ as an additional explanatory variable.

In R, we can fit the model again with `lm()` but we have to use the ^ operator in conjunction with the function `I()` to add the quadratic term as an additional regressor to the argument `formula`. The reason is that the regression formula we pass to `formula` is converted to an object of the class formula, and for objects of this class, the operators +, -, * and ^ have a nonarithmetic interpretation. `I()` ensures that they are used as arithmetical operators (see `?I`)

```
# fit the quadratic Model
quadratic_model <- lm(score ~ income + I(income^2), data = CASchools)

# obtain the model summary
coeftest(quadratic_model, vcov. = vcovHC, type = "HC1")
```

```
t test of coefficients:
```

```
                 Estimate  Std. Error  t value  Pr(>|t|)
(Intercept) 607.3017435   2.9017544 209.2878 < 2.2e-16 ***
income        3.8509939   0.2680942  14.3643 < 2.2e-16 ***
I(income^2)  -0.0423084   0.0047803  -8.8505 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated function is

$$\widehat{TestScore} = \underset{(2.90)}{607.3} + \underset{(0.27)}{3.85}\, income_i - \underset{(0.0048)}{0.0423}\, income_i^2$$

We can test the hypothesis that the relationship between test scores and income is linear against the alternative that it is quadratic, by testing

$$H_0 : \beta_2 = 0 \;\; vs. \;\; H_1 : \beta_2 \neq 0$$

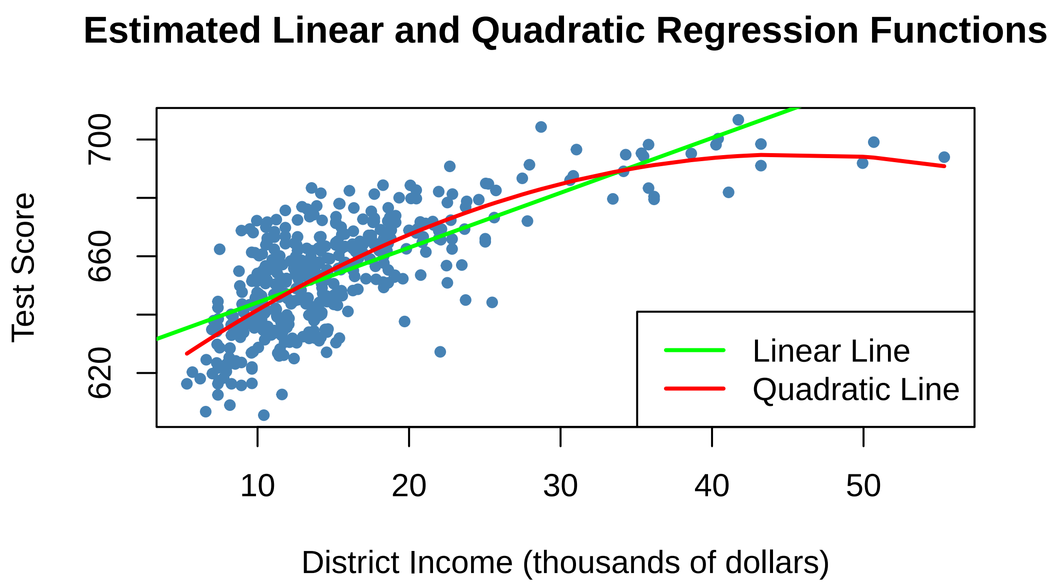
since $\beta_2 = 0$ would result in a simple linear equation and $\beta_2 \neq 0$ implies a quadratic relationship.

We can manually compute the t-value reported in the table as $t = (\hat{\beta}_2 - 0)/SE(\hat{\beta}_2) = -0.042308/0.00478 = -8.85$. With this t-value we can reject the null hypothesis at any common level of significance and we may conclude that the relationship is not linear. We could also have drawn the same conclusion by looking at the asterisks in the summary table, where we observe that the coefficient for the quadratic term is highly significant at the 0.1% level (***).

We will now draw the same scatter plot as for the linear model and add the regression line for the quadratic model. Since `abline()` only plots straight lines, it cannot be used here, but we can use `lines()` function instead, which is suitable for plotting nonstraight lines (see `?lines`). The most basic call of `lines()` is `lines(x_values, y_values)` where `x_values` and `y_values` are vectors of the same length that provide coordinates of the points to be sequentially connected by a line. This requires sorted coordinate pairs according to the X-values. We may use the function `order()` to sort the fitted values of score according to the observations of income, obtained from our quadratic model.

```r
# draw a scatterplot of the observations for income and test score
plot(CASchools$income, CASchools$score,
     col  = "steelblue",
     pch = 20,
     xlab = "District Income (thousands of dollars)",
     ylab = "Test Score",
```

```
    main = "Estimated Linear and Quadratic Regression Functions")

# add a linear function to the plot
abline(linear_model, col = "green", lwd = 2)

# add quatratic function to the plot
order_id <- order(CASchools$income)

lines(x = CASchools$income[order_id],
      y = fitted(quadratic_model)[order_id],
      col = "red",
      lwd = 2)
legend("bottomright",legend=c("Linear Line","Quadratic Line"),
       lwd=2,col=c("green","red"))
```



As the plot shows, the quadratic function appears to provide a better fit to the data compared to the linear function.

3.1 Polynomials

The method employed to derive a quadratic model can be extended to polynomial models of any degree r

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + ... + \beta_r X_i^r + u_i$$

We can estimate polynomial models in R using the function `poly()`. The polynomial degrees (r) must be indicated into the `degree` argument of the function. For a cubic model:

```
# estimate a cubic model
cubic_model <- lm(score ~ poly(income, degree = 3, raw = TRUE), data = CASchools)
```

The function `poly()` generates orthogonal polynomials that default to being orthogonal to the constant term. By setting `raw = TRUE`, we evaluate raw polynomials instead. For more information, refer to `?poly`.

3.1.1 Joint Hypothesis Testing

A common dilemma in practice is selecting the optimal polynomial order. Similar to the quadratic regression model, we can test the null hypothesis suggesting that the true relationship is linear, in contrast to the alternative hypothesis proposing a polynomial relationship.

$$H_0 : \beta_2 = 0, \beta_3 = 0, \dots, \beta_r = 0 \quad \text{vs.} \quad H_1 : \text{at least one } \beta_j \neq 0, \quad j = 2, \dots, r.$$

This represents a joint null hypothesis with $r-1$ restrictions, which can be tested using the F-test previously described. The function `linearHypothesis()` facilitates such testing. For instance, we can test the null of a linear model against the alternative of a polynomial with a maximum degree $r = 3$ as demonstrated below.

```
# test the hypothesis of a linear model against quadratic or cubic alternatives

# set up hypothesis matrix
R <- rbind(c(0, 0, 1, 0),
           c(0, 0, 0, 1))

# do the test
linearHypothesis(cubic_model,
                 hypothesis.matrix = R,
                 white.adj = "hc1")
```

```
Linear hypothesis test

Hypothesis:
poly(income, degree = 3, raw = TRUE)2 = 0
poly(income, degree = 3, raw = TRUE)3 = 0
```

```
Model 1: restricted model
Model 2: score ~ poly(income, degree = 3, raw = TRUE)

Note: Coefficient covariance matrix supplied.

  Res.Df Df      F    Pr(>F)
1    418
2    416  2 37.691 9.043e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We have created and supplied the hypothesis matrix R as the input argument `hypothesis.matrix`. This is convenient when the constraints involve several coefficients and when coefficients have long names, such as when using `poly()` (see `summary(cubic_model)`). The interpretation of the hypothesis matrix R by `linearHypothesis()` is best understood through matrix algebra. For our case with two linear constraints it would be as follows:

$$R\beta = s$$
$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Rightarrow \begin{pmatrix} \beta_2 \\ \beta_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

`linearHypothesis()` uses the zero vector for s by default, see `?linearHypothesis`.

From the results of the joint hypothesis test, with a very small p-value, we can reject the null hypothesis of a linear relationship. However, we still face the challenge of **choosing the right polynomial degree** r. In other words, how many powers of X should be included in a polynomial regression. Increasing the degree r introduces more flexibility into the regression function, but adding more regressors can reduce the precision of the estimated coefficients.

While there is no general rule to select r, this could be determined by **sequential testing**, where individual hypotheses are tested sequentially in the following steps:

1. Estimate the polynomial regression model for a maximum value of r.
2. Use a t-test to test $\beta_r = 0$. If the null hypothesis is rejected, then X^r belongs in the regression equation.
3. If the null is accepted, X^r can be excluded from the model. Then repeat step 1 with order $r-1$ and test whether $\beta_{r-1} = 0$. If the null is rejected, use a polynomial model of order $r-1$.

4. If the null is not rejected in step 3, continue this procedure until the coefficient on the highest power in your polynomial is statistically significant.

To choose the initial maximum value of r, Stock and Watson (2015) suggest to choose $2, 3$ or 4 for applications on economic data, due to its usual smoothness (absence of jumps or "spikes).

We will apply this sequential testing to our cubic model reporting robust standard errors:

```
# test the hypothesis using robust standard errors
coeftest(cubic_model, vcov. = vcovHC, type = "HC1")
```

```
t test of coefficients:

                                     Estimate  Std. Error  t value
(Intercept)                        6.0008e+02  5.1021e+00 117.6150
poly(income, degree = 3, raw = TRUE)1  5.0187e+00  7.0735e-01   7.0950
poly(income, degree = 3, raw = TRUE)2 -9.5805e-02  2.8954e-02  -3.3089
poly(income, degree = 3, raw = TRUE)3  6.8549e-04  3.4706e-04   1.9751
                                      Pr(>|t|)
(Intercept)                          < 2.2e-16 ***
poly(income, degree = 3, raw = TRUE)1 5.606e-12 ***
poly(income, degree = 3, raw = TRUE)2  0.001018 **
poly(income, degree = 3, raw = TRUE)3  0.048918 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated cubic regression function relating district income to test scores is

$$\widehat{TestScore} = \underset{(5.1)}{600.1} + \underset{(0.71)}{5.02}\, Income - \underset{(0.029)}{0.096}\, Income^2 + \underset{(0.00035)}{0.00069}\, Income^3$$

The t-statistic on $Income^3$ is 1.98, so the null hypothesis that the regression function is a quadratic is rejected against the alternative that it is a cubic at the 5% level.

We can additionally test if the coefficients for $Income^2$ and $Income^3$ are jointly significant using a robust version of the F-test:

```
# perform robust F-test
linearHypothesis(cubic_model,
                 hypothesis.matrix = R,
                 vcov. = vcovHC, type = "HC1")
```

```
Linear hypothesis test

Hypothesis:
poly(income, degree = 3, raw = TRUE)2 = 0
poly(income, degree = 3, raw = TRUE)3 = 0

Model 1: restricted model
Model 2: score ~ poly(income, degree = 3, raw = TRUE)

Note: Coefficient covariance matrix supplied.

  Res.Df Df      F    Pr(>F)
1    418
2    416  2 29.678 8.945e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

With a p-value below 0.001, we reject the null hypothesis that the regression function is linear against the alternative of a quadratic or cubic relationship.

3.1.2 Interpretation of coefficients

The coefficients in polynomial regressions do not have a simple interpretation. The best way to interpret them is to calculate the estimated effect on Y associated with a change in X for one or more values of X.

For example, if we would like to know the predicted change in test scores when income changes from 10 to 11 (thousand dollars) based on our estimated quadratic regression function

$$\widehat{TestScore} = 607.3 + 3.85\,\text{Income} - 0.0423\,\text{Income}^2$$

we would compute the $\Delta\hat{Y}$ associated with that specific unit change in income using the following formula:

$$\widehat{\Delta Y} = (\hat{\beta}_0 + \hat{\beta}_1 \times 11 + \hat{\beta}_2 \times 11^2) - (\hat{\beta}_0 + \hat{\beta}_1 \times 10 + \hat{\beta}_2 \times 10^2)$$

We can compute $\Delta\hat{Y}$ in R using `predict()`

```r
# compute and assign the quadratic model
quadratic_model <- lm(score ~ income + I(income^2), data = CASchools)

# set up data for prediction
new_data <- data.frame(income = c(10, 11))

# do the prediction
Y_hat <- predict(quadratic_model, newdata = new_data)

# compute the difference
diff(Y_hat)
```

```
       2
2.962517
```

The expected change in $TestScore$ when increasing $income$ from 10 to 11 (thousand dollars) is about 2.96 points. Note that, since the relationship is not linear, this unit change effect will vary depending on the pair of values of X selected. One way to notice this is by plotting the estimated quadratic regression function.

3.2 Logarithms

Another approach to express a nonlinear regression function involves using the natural logarithm of Y and/or X. Logarithms help convert variable changes into percentages, which is useful as many relationships are naturally described in terms of percentages. There are three different situations where logarithms are used: when X is transformed by taking its logarithm but Y is not; when Y is transformed to its logarithm but X is not; and when both Y and X are transformed to their logarithm.

3.2.1 Case I: X is in logarithms, Y is not.

In this case, sometimes referred to as a **linear-log model**, the regression model is

$$Y_i = \beta_0 + \beta_1 \ln(X_i) + u_i, \quad i = 1, \ldots, n.$$

As for polynomial regressions, there is no need to create the logged variable in advance, but we simply adjust the `formula` argument in `lm()` to log-transform the variable of interest.

```r
# estimate a level-log model
LinearLog_model <- lm(score ~ log(income), data = CASchools)

# compute robust summary
coeftest(LinearLog_model,
         vcov = vcovHC, type = "HC1")
```

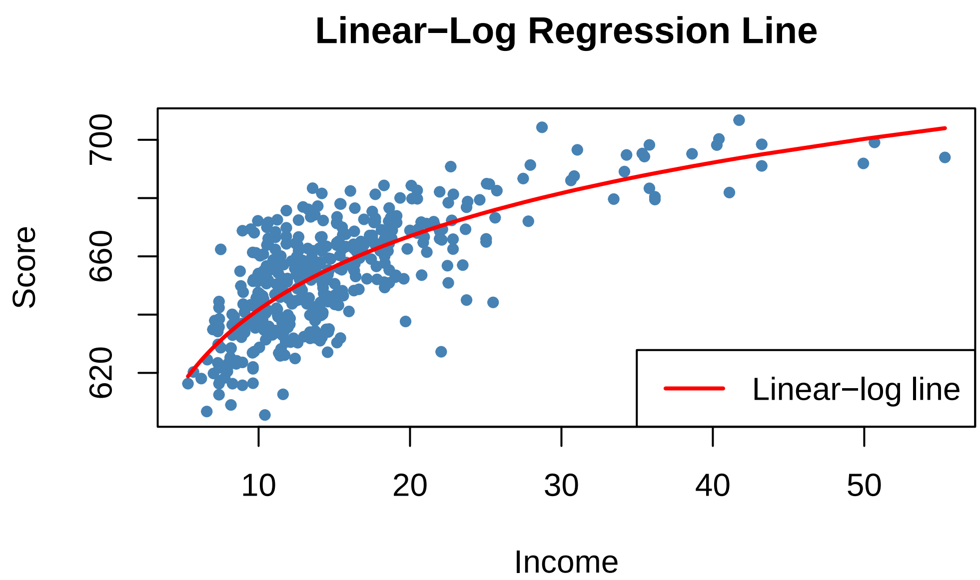
```
t test of coefficients:

            Estimate Std. Error t value  Pr(>|t|)
(Intercept) 557.8323     3.8399 145.271 < 2.2e-16 ***
log(income)  36.4197     1.3969  26.071 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated regression model is

$$\widehat{TestScore} = \underset{(3.84)}{557.8} + \underset{(1.40)}{36.42} \ln(Income)$$

We plot this function

```r
# draw a scatterplot
plot(score ~ income,
     col = "steelblue",
     pch = 20,
     data = CASchools,
     ylab="Score",
     xlab="Income",
     main = "Linear-Log Regression Line")

# add the linear-log regression line
order_id  <- order(CASchools$income)

lines(CASchools$income[order_id],
      fitted(LinearLog_model)[order_id],
      col = "red",
      lwd = 2)
legend("bottomright",legend = "Linear-log line",lwd = 2,col ="red")
```



We can interpret $\hat{\beta}_1$ as follows: a 1% increase in income is associated with an average increase in test scores of $0.01 \times 36.42 = 0.36$ points. If we wanted to compute the change in $TestScore$ of a one unit change in $income$, we would compute the $\Delta\hat{Y}$ just as we did with polynomials.

3.2.2 Case II: Y is in logarithms, X is not

In this second case, the **log-linear model**, the regression function is

$$\ln(Y_i) = \beta_0 + \beta_1 X_i + u_i, \quad i = 1, ..., n.$$

```
# estimate a log-linear model
LogLinear_model <- lm(log(score) ~ income, data = CASchools)

# obtain a robust coefficient summary
coeftest(LogLinear_model,
         vcov = vcovHC, type = "HC1")
```

```
t test of coefficients:

              Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 6.43936234 0.00289382 2225.210 < 2.2e-16 ***
income      0.00284407 0.00017509   16.244 < 2.2e-16 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated regression function is

$$\ln(\widehat{TestScore}) = \underset{(0.003)}{6.439} + \underset{(0.0002)}{0.00284}\, Income$$

An increase in *income* of one unit ($1000) is associated with an average increase in $TestScore$ of $100 \times 0.00284 = 0.284\%$

Note that when the dependent variable is in logarithms, one cannot use $e^{\log(\cdot)}$ to transform predictions back to the original scale, as pointed by Stock and Watson (2015).

3.2.3 Case III: X and Y are in logarithms

The **log-log regression model** is

$$\ln(Y_i) = \beta_0 + \beta_1 \ln(X_i) + u_i, \quad i = 1, \ldots, n.$$

```
# estimate the log-log model
LogLog_model <- lm(log(score) ~ log(income), data = CAShools)

# print robust coefficient summary
coeftest(LogLog_model,
         vcov = vcovHC, type = "HC1")
```

```
t test of coefficients:

             Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 6.3363494  0.0059246 1069.501 < 2.2e-16 ***
log(income) 0.0554190  0.0021446   25.841 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated log-log regression function is

$$\ln(\widehat{TestScore}) = \underset{(0.006)}{6.336} + \underset{(0.002)}{0.0554}\, \ln(Income)$$

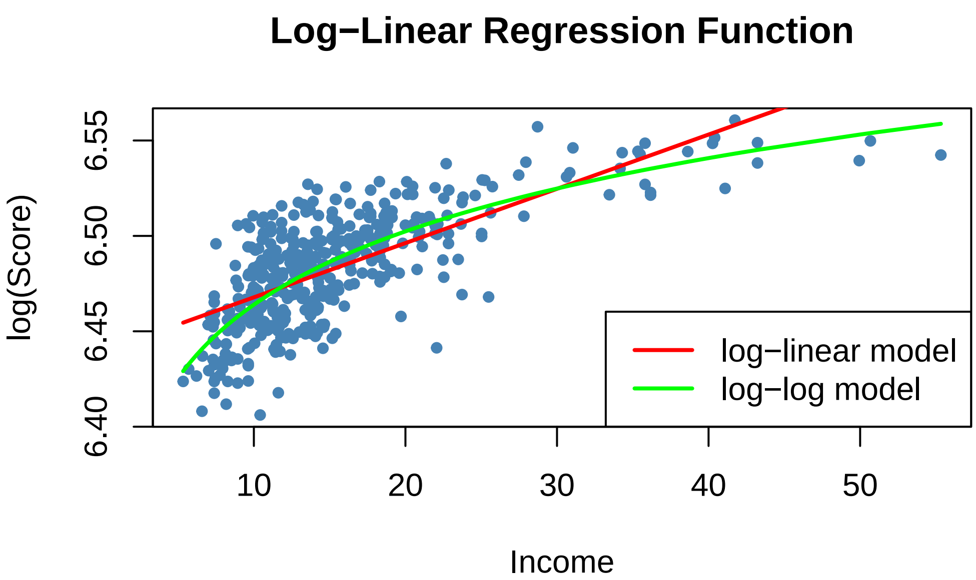
A 1% increase in $Income$ is associated with an average increase in $TestScore$ of 0.055%

We now plot the log-linear and the log-log regression models together

```
# generate a scatterplot
plot(log(score) ~ income,
     col = "steelblue",
     pch = 20,
     data = CASchools,
     ylab="log(Score)",
     xlab="Income",
     main = "Log-Linear Regression Function")

# add the log-linear regression line
order_id  <- order(CASchools$income)

lines(CASchools$income[order_id],
      fitted(LogLinear_model)[order_id],
      col = "red",
      lwd = 2)

# add the log-log regression line
lines(sort(CASchools$income),
      fitted(LogLog_model)[order(CASchools$income)],
      col = "green",
      lwd = 2)

# add a legend
legend("bottomright",
       legend = c("log-linear model", "log-log model"),
       lwd = 2,
       col = c("red", "green"))
```

3.2.4 Comparing logarithmic specifications

Which of the log regression models best fits the data? The $\bar{R}^2$ can be used to compare the log-linear and log-log models. Similarly, the $\bar{R}^2$ can be used to compare the linear-log regression and the linear regression of Y against X. But unfortunately, the $\bar{R}^2$ cannot be used to compare the linear-log and the log-log model because their dependent variables are different (one is Y, the other one is $\ln(Y)$). Because of this problem, the best thing to do in a particular application is to decide, using economic theory and experts' knowledge of the problem, whether it makes sense to specify Y in logarithms.




```
# compute the adj. R^2 for the nonlinear models
adj_R2 <-rbind("quadratic" = summary(quadratic_model)$adj.r.squared,
               "cubic" = summary(cubic_model)$adj.r.squared,
               "LinearLog" = summary(LinearLog_model)$adj.r.squared,
               "LogLinear" = summary(LogLinear_model)$adj.r.squared,
               "LogLog" = summary(LogLog_model)$adj.r.squared)

# assign column names
colnames(adj_R2) <- "adj_R2"

adj_R2
```

```
             adj_R2
quadratic 0.5540444
cubic     0.5552279
LinearLog 0.5614605
LogLinear 0.4970106
LogLog    0.5567251
```

From those models where the dependent variable is *TestScore*, we observe a very similar adjusted fit. We can compare the cubic and the linear-log model by plotting their estimated regression functions.

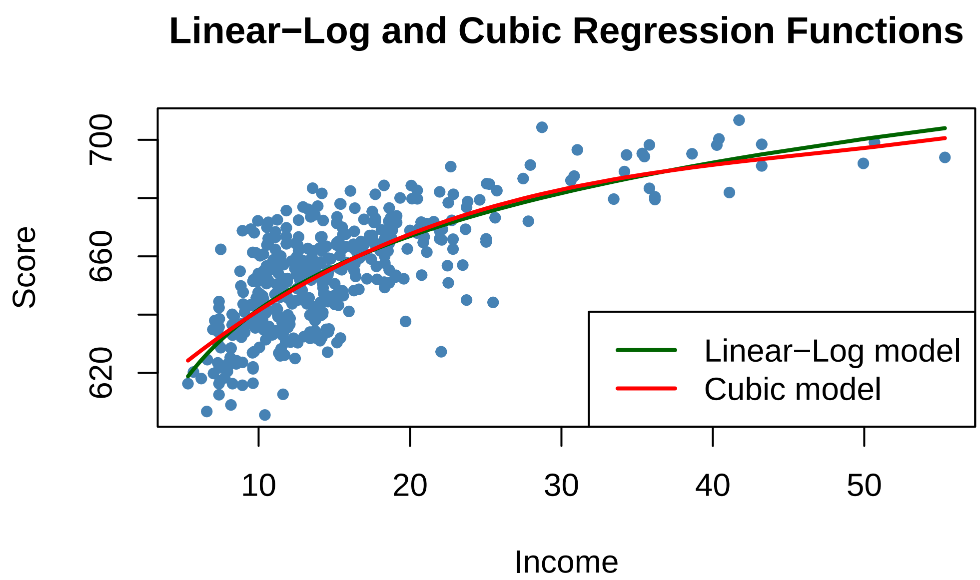
```r
# generate a scatterplot
plot(score ~ income,
     data = CASchools,
     col = "steelblue",
     pch = 20,
     ylab="Score",
     xlab="Income",
     main = "Linear-Log and Cubic Regression Functions")

# add the linear-log regression line
order_id  <- order(CASchools$income)

lines(CASchools$income[order_id],
      fitted(LinearLog_model)[order_id],
      col = "darkgreen",
      lwd = 2)

# add the cubic regression line
lines(x = CASchools$income[order_id],
      y = fitted(cubic_model)[order_id],
      col = "red",
      lwd = 2)
# add a legend
legend("bottomright",
       legend = c("Linear-Log model", "Cubic model"),
       lwd = 2,
       col = c("darkgreen", "red"))
```

We appreciate a nearly identical look for both models, although we may prefer the linear-log model for simplicity, since it does not include higher-degree polynomials.

3.3 Interactions Between Independent Variables

Sometimes it is interesting to learn how the effect on Y of a change in an independent variable depends on the value of another independent variable. For example, we may ask if districts with many English learners benefit differently from a decrease in the student-teacher ratio compared to those with fewer English learning students. We can assess this by using a multiple regression model and including an interaction term.

We consider three cases: when both independent variables are binary, when one is binary and the other is continuous, and when both are continuous.



3.3.1 Interactions Between Two Binary Variables

Let

$$\text{HiSTR} = \begin{cases} 1, & \text{if STR} \geq 20 \\ 0, & \text{else} \end{cases}$$

$$\text{HiEL} = \begin{cases} 1, & \text{if english} \geq 10 \\ 0, & \text{else} \end{cases}$$

In R, we construct this dummies as follows

```
# append HiSTR to CASchools
CASchools$HiSTR <- as.numeric(CASchools$STR >= 20)

# append HiEL to CASchools
CASchools$HiEL <- as.numeric(CASchools$english >= 10)
```

We now estimate the model

$$TestScore = \beta_0 + \beta_1\, HiSTR + \beta_2\, HiEL + \beta_3\, HiSTR \times HiEL + u_i.$$

We can simply indicate `HiEL * HiSTR` inside the `lm()` formula to add the interaction term to the model. Note that this adds $HiEL$, $HiSTR$ and their interaction as regressors, whereas indicating `HiEL:HiSTR` only adds the interaction term.

```r
# estimate the model with a binary interaction term
bi_model <- lm(score ~ HiSTR * HiEL, data = CASchools)

# print a robust summary of the coefficients
coeftest(bi_model, vcov. = vcovHC, type = "HC1")
```

```
t test of coefficients:

             Estimate Std. Error  t value  Pr(>|t|)    
(Intercept) 664.1433     1.3881 478.4589 < 2.2e-16 ***
HiSTR        -1.9078     1.9322  -0.9874    0.3240    
HiEL        -18.3155     2.3340  -7.8472 3.634e-14 ***
HiSTR:HiEL   -3.2601     3.1189  -1.0453    0.2965    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated regression model is

$$\widehat{TestScore} = \underset{(1.39)}{664.1} - \underset{(1.93)}{1.9} \text{ HiSTR} - \underset{(2.33)}{18.3} \text{ HiEL} - \underset{(3.12)}{3.3} \text{ (HiSTR} \times \text{HiEL)}$$

According to this model, when moving from a school district with a low student-teacher ratio to one with a high ratio, the average effect on test scores depends on the percentage of English learners (HiEL), and can be computed as $-1.9 - 3.3 \times HiEL$. This is, for districts with fewer English learners ($HiEL = 0$), the expected decrease in test scores is 1.9 points. However, for districts with a higher proportion of English learners ($HiEL = 1$), the predicted decrease in test scores is $1.9 + 3.3 = 5.2$ points.

We can estimate the mean test score for each possible combination of the included binary variables

```r
# estimate means for all combinations of HiSTR and HiEL

# 1.
predict(bi_model, newdata = data.frame("HiSTR" = 0, "HiEL" = 0))
```

```
       1 
664.1433 
```

```r
# 2.
predict(bi_model, newdata = data.frame("HiSTR" = 0, "HiEL" = 1))
```

```
       1
645.8278
```

```r
# 3.
predict(bi_model, newdata = data.frame("HiSTR" = 1, "HiEL" = 0))
```

```
       1
662.2354
```

```r
# 4.
predict(bi_model, newdata = data.frame("HiSTR" = 1, "HiEL" = 1))
```

```
       1
640.6598
```

We verify that these predictions are differences in the coefficient estimates presented in the regression equation

$$\begin{aligned}
\widehat{TestScore} = \hat{\beta}_0 = 664.1 &\Leftrightarrow HiSTR = 0, \quad HiEL = 0. \\
\widehat{TestScore} = \hat{\beta}_0 + \hat{\beta}_2 = 664.1 - 18.3 = 645.8 &\Leftrightarrow HiSTR = 0, \quad HiEL = 1. \\
\widehat{TestScore} = \hat{\beta}_0 + \hat{\beta}_1 = 664.1 - 1.9 = 662.2 &\Leftrightarrow HiSTR = 1, \quad HiEL = 0. \\
\widehat{TestScore} = \hat{\beta}_0 + \hat{\beta}_1 + \hat{\beta}_2 + \hat{\beta}_3 = 664.1 - 1.9 - 18.3 - 3.3 = 640.6 &\Leftrightarrow HiSTR = 1, \quad HiEL = 1.
\end{aligned}$$

3.3.2 Interactions Between a Continuous and a Binary Variable

This specification where the interaction term includes a continuous variable (X_i) and a binary variable (D_i) allows for the slope to depend on the binary variable. There are three different possibilities:

1. Different intercepts, same slope:

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 D_i + u_i$$

2. Different intercepts and slopes:

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 D_i + \beta_3 \times (X_i \times D_i) + u_i$$

3. Same intercept, different slopes:

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 (X_i \times D_i) + u_i.$$

Does the effect on test scores of cutting the student-teacher ratio depend on whether the percentage of students still learning English is high or low? One way to answer this question is to use a specification that allows for two different regression lines, depending on whether there is a high or a low percentage of English learners. This is achieved using the different intercept/different slope specification. We estimate the regression model

$$\widehat{TestScore}_i = \beta_0 + \beta_1\, STR_i + \beta_2\, HiEL_i + \beta_2\, (STR_i \times HiEL_i) + u_i$$

```
# estimate the model
bci_model <- lm(score ~ STR + HiEL + STR * HiEL, data = CASchools)

# print robust summary of coefficients
coeftest(bci_model, vcov. = vcovHC, type = "HC1")
```

```
t test of coefficients:

             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 682.24584   11.86781 57.4871   <2e-16 ***
STR          -0.96846    0.58910 -1.6440   0.1009    
HiEL          5.63914   19.51456  0.2890   0.7727    
STR:HiEL     -1.27661    0.96692 -1.3203   0.1875    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

$$\widehat{TestScore} = \underset{(11.87)}{682.2} - \underset{(0.59)}{0.97}\, STR + \underset{(19.51)}{5.6}\, HiEL - \underset{(0.97)}{1.28}\, (STR \times HiEL).$$

The estimated regression line for districts with a low fraction of English learners ($HiEL = 0$) is

$$\widehat{TestScore} = 682.2 - 0.97\, STR_i$$

while the one for districts with a high fraction of English learners ($HiEL = 1$) is

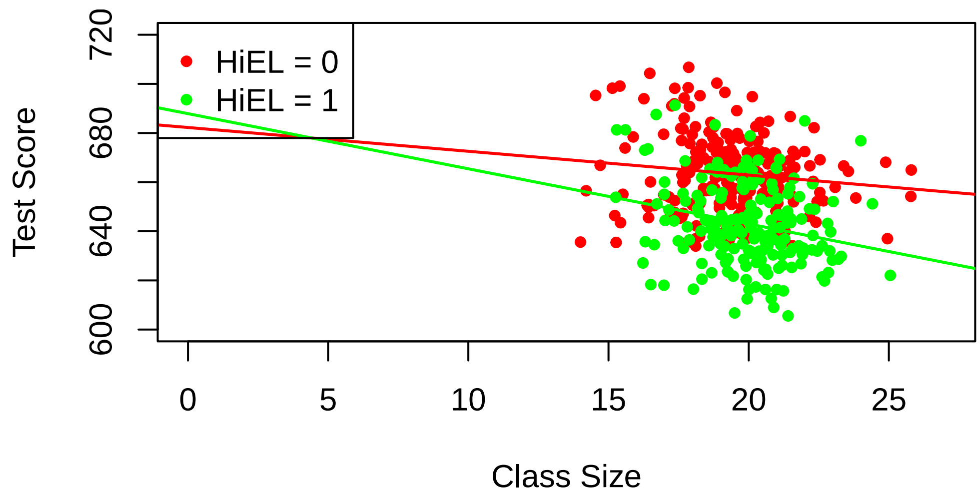
$$\begin{aligned}\widehat{TestScore} &= 682.2 + 5.6 - 0.97\,STR_i - 1.28\,STR_i \\ &= 687.8 - 2.25\,STR_i.\end{aligned}$$

The expected rise in test scores after decreasing the student-teacher ratio by one unit is roughly 0.97 points in districts with a low proportion of English learners, but 2.25 points in districts with a high concentration of English learners. The coefficient on the interaction term, "$STR \times HiEL$", indicates that the contrast between these effects amounts to 1.28 points.

We now plot both regression lines from the model by using different colors to differentiate each of the STR levels.

```
# identify observations with english >= 10
id <- CASchools$english >= 10

# plot observations with HiEL = 0 as red dots
plot(CASchools$STR[!id], CASchools$score[!id],
     xlim = c(0, 27),
     ylim = c(600, 720),
     pch = 20,
     col = "red",
     main = "",
     xlab = "Class Size",
     ylab = "Test Score")

# plot observations with HiEL = 1 as green dots
points(CASchools$STR[id], CASchools$score[id],
     pch = 20,
     col = "green")

# read out estimated coefficients of bci_model
coefs <- bci_model$coefficients

# draw the estimated regression line for HiEL = 0
abline(coef = c(coefs[1], coefs[2]),
       col = "red",
       lwd = 1.5)

# draw the estimated regression line for HiEL = 1
abline(coef = c(coefs[1] + coefs[3], coefs[2] + coefs[4]),
```

```
        col = "green",
        lwd = 1.5 )

# add a legend to the plot
legend("topleft",
        pch = c(20, 20),
        col = c("red", "green"),
        legend = c("HiEL = 0", "HiEL = 1"))
```



3.3.3 Interactions Between Two Continuous Variables

Let's now examine the interaction between the continuous variables student-teacher ratio (STR) and the percentage of English learners ($english$).

```
# estimate regression model including the interaction between 'english' and 'STR'
cci_model <- lm(score ~ STR + english + english * STR, data = CASchools)

# print summary
coeftest(cci_model, vcov. = vcovHC, type = "HC1")
```

```
t test of coefficients:

               Estimate  Std. Error t value Pr(>|t|)
(Intercept) 686.3385268 11.7593466 58.3654  < 2e-16 ***
STR          -1.1170184   0.5875136 -1.9013  0.05796 .
```

```
english      -0.6729119   0.3741231 -1.7986  0.07280 .
STR:english   0.0011618   0.0185357  0.0627  0.95005
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated regression function is

$$\widehat{TestScore} = \underset{(11.76)}{686.3} - \underset{(0.59)}{1.12}\, STR - \underset{(0.37)}{0.67}\, english + \underset{(0.02)}{0.0012}\, (STR \times english).$$

Before proceeding with the interpretations, let us explore the quartiles of *english*

```r
summary(CASchools$english)
```

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.000   1.941   8.778  15.768  22.970  85.540 
```

When the percentage of English learners is at the median ($english = 8.778$), the slope of the line is estimated to be $(-1.12 + 0.0012 * 8.778 = -1.12)$. When the percentage of English learners is at the 75th percentile ($english = 22.97$), this line is estimated to be slightly flatter, with a slope of $-1.12 + 0.0012 * 22.97 = -1.09$. In other words, for a district with 8.78% English learners, the estimated effect of a one-unit reduction in the student-teacher ratio is to increase on average test scores by 1.11 points, but for a district with 23% English learners, reducing the student-teacher ratio by one unit is predicted to increase test scores on average by 1.09 points. However, it is important to note from the output of `coeftest()` that the estimated coefficient on the interaction term (β_3) is not statistically significant at the 10% level, so we cannot reject the null hypothesis $H_0 : \beta_3 = 0$.

3.4 Nonlinear Effects on Test Scores of the Student-Teacher Ratio

This section examines three key questions about test scores and the student-teacher ratio. First, it explores if reducing the student-teacher ratio affects test scores differently based on the number of English learners, even when considering economic differences across districts. Second, it investigates if this effect varies depending on the student-teacher ratio. Lastly, it aims to determine the expected impact on test scores when the student-teacher ratio decreases by two students per teacher, considering both economic factors and potential nonlinear relationships.

We will answer these questions considering the previously explained nonlinear regression specifications, extended to include two measures of the economic background of the students: the percentage of students eligible for a subsidized lunch (*lunch*) and the logarithm of average district income ($ln(income)$). The logarithm of district income is used following our previous empirical analysis, which suggested that this specification captures the nonlinear relationship between scores and income. We leave out the expenditure per pupil (*expenditure*) from our analysis because including it would suggest that spending changes with the student-teacher ratio (in other words, we would not be holding expenditures per pupil constant).

We will consider 7 different model specifications:

$$TestScore_i = \beta_0 + \beta_1\, STR_i + \beta_2\, english_i + \beta_3\, lunch_i + u_i.$$

$$TestScore_i = \beta_0 + \beta_1\, STR_i + \beta_2\, english_i + \beta_3 \ln(income_i) + u_i.$$

$$TestScore_i = \beta_0 + \beta_1\, STR_i + \beta_2\, HiEL_i + \beta_3\,(HiEL_i \times STR_i) + u_i.$$

$$TestScore_i = \beta_0 + \beta_1\, STR_i + \beta_2\, HiEL_i + \beta_3\,(HiEL_i \times STR_i) + \beta_4\, lunch_i + \beta_5 \ln(income_i) + u_i.$$

$$TestScore_i = \beta_0 + \beta_1\, STR_i + \beta_2\, STR_i^2 + \beta_3\, HiEL_i + \beta_4\, lunch_i + \beta_5 \ln(income_i) + u_i.$$

$$\begin{aligned} TestScore_i = {} & \beta_0 + \beta_1\, STR_i + \beta_2\, STR_i^2 + \beta_3\, STR_i^3 + \beta_4\, HiEL_i + \beta_5\,(HiEL_i \times STR_i) \\ & + \beta_6\,(HiEL_i \times STR_i^2) + \beta_7\,(HiEL_i \times STR_i^3) + \beta_8\, lunch_i + \beta_9 \ln(income_i) + u_i. \end{aligned}$$

$$TestScore_i = \beta_0 + \beta_1\, STR_i + \beta_2\, STR_i^2 + \beta_3\, STR_i^3 + \beta_4\, english_i + \beta_5\, lunch_i + \beta_6 \ln(income_i) + u_i.$$

```r
# estimate all models
TS_mod1 <- lm(score ~ STR + english + lunch, data = CASchools)

TS_mod2 <- lm(score ~ STR + english + lunch + log(income), data = CASchools)

TS_mod3 <- lm(score ~ STR + HiEL + HiEL:STR, data = CASchools)

TS_mod4 <- lm(score ~ STR + HiEL + HiEL:STR + lunch + log(income), data = CASchools)

TS_mod5 <- lm(score ~ STR + I(STR^2) + I(STR^3) + HiEL + lunch + log(income),
    data = CASchools)

TS_mod6 <- lm(score ~ STR + I(STR^2) + I(STR^3) + HiEL + HiEL:STR + HiEL:I(STR^2) +
    HiEL:I(STR^3) + lunch + log(income), data = CASchools)

TS_mod7 <- lm(score ~ STR + I(STR^2) + I(STR^3) + english + lunch + log(income),
    data = CASchools)
```

We could use `summary()` to report the estimates of each model, but `stargazer()` conveniently reports the results of all models in a tabular form, which is more practical when comparing models.

```r
# gather robust standard errors in a list
rob_se <- list(sqrt(diag(vcovHC(TS_mod1, type = "HC1"))),
               sqrt(diag(vcovHC(TS_mod2, type = "HC1"))),
               sqrt(diag(vcovHC(TS_mod3, type = "HC1"))),
               sqrt(diag(vcovHC(TS_mod4, type = "HC1"))),
               sqrt(diag(vcovHC(TS_mod5, type = "HC1"))),
               sqrt(diag(vcovHC(TS_mod6, type = "HC1"))),
               sqrt(diag(vcovHC(TS_mod7, type = "HC1"))))

# generate a LaTeX table of regression outputs
stargazer(TS_mod1, TS_mod2, TS_mod3, TS_mod4,
          TS_mod5, TS_mod6, TS_mod7,
          digits = 3,
          type = "text",
          header = FALSE,
          dep.var.caption = "Dependent Variable: Test Score",
          se = rob_se,
          model.numbers = FALSE,
          column.labels = c("(1)", "(2)", "(3)", "(4)", "(5)", "(6)", "(7)"))
```

```
==============================================================================================================
                                                                                                          De
                    ------------------------------------------------------------------------------------------
                              (1)                     (2)                      (3)
--------------------------------------------------------------------------------------------------------------
STR                        -0.998***               -0.734***                 -0.968
                           (0.270)                 (0.257)                  (0.589)

english                    -0.122***               -0.176***
                           (0.033)                 (0.034)

I(STR2)


I(STR3)


lunch                      -0.547***               -0.398***
                           (0.024)                 (0.033)
```

log(income)		11.569***	
		(1.819)	
HiEL			5.639
			(19.515)
STR:HiEL			-1.277
			(0.967)
I(STR2):HiEL			
I(STR3):HiEL			
Constant	700.150***	658.552***	682.246***
	(5.568)	(8.642)	(11.868)
Observations	420	420	420
R2	0.775	0.796	0.310
Adjusted R2	0.773	0.794	0.305
Residual Std. Error	9.080 (df = 416)	8.643 (df = 415)	15.880 (df = 416)
F Statistic	476.306*** (df = 3; 416)	405.359*** (df = 4; 415)	62.399*** (df = 3; 416)
Note:			

What can be concluded from the results presented?

First, we we see the estimated coefficient on STR is highly significant in all models except from specifications (3) and (4). When we add $log(income)$ to model (1) in the second specification, all coefficients remain highly significant while the coefficient on the new regressor is also statistically significant at the 1% level. Additionally, the coefficient on STR is now 0.27 higher than in model (1), suggesting a possible mitigation of omitted variable bias when including $ln(income)$ as regressor. For these reasons, it makes sense to keep this variable in other models too.

Models (3) and (4) include the interaction term between STR and $HiEL$, first without control variables in the third specification and then controlling for economic factors in the fourth. The estimated coefficient for the interaction term is not significant at any common level in any of these models, nor is the coefficient on the dummy variable $HiEL$. Hence, despite accounting for economic factors, we cannot reject the null hypotheses that the impact of the student-teacher ratio on test scores remains consistent across districts with high and low proportions of English learning students.

In regression (5) we have included quadratic and cubic terms for STR, while omitting the interaction term between STR and $HiEL$, since it was not significant in specification (4). The results indicate high levels of significance for these estimated coefficients and we can therefore assume the presence of a nonlinear effect of the student-teacher ration on test scores. This could be also verified with an F-test of $H_0 : \beta_2 = \beta_3 = 0$.

Regression (6) delves deeper into examining whether the proportion of English learners influences the student-teacher ratio, incorporating the interaction terms $HiEL \times STR$, $HiEL \times STR^2$ and $HiEL \times STR^3$. Each individual t-test confirms significant effects. To validate this, we perform a robust F-test to assess $H_0 : \beta_5 = \beta_6 = \beta_7 = 0$.

```
# check joint significance of the interaction terms
linearHypothesis(TS_mod6,
                 c("STR:HiEL=0", "I(STR^2):HiEL=0", "I(STR^3):HiEL=0"),
                 vcov. = vcovHC, type = "HC1")
```

```
Linear hypothesis test

Hypothesis:
STR:HiEL = 0
I(STR^2):HiEL = 0
I(STR^3):HiEL = 0

Model 1: restricted model
Model 2: score ~ STR + I(STR^2) + I(STR^3) + HiEL + HiEL:STR + HiEL:I(STR^2) +
    HiEL:I(STR^3) + lunch + log(income)

Note: Coefficient covariance matrix supplied.

  Res.Df Df      F  Pr(>F)
1    413
2    410  3 2.1885 0.08882 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

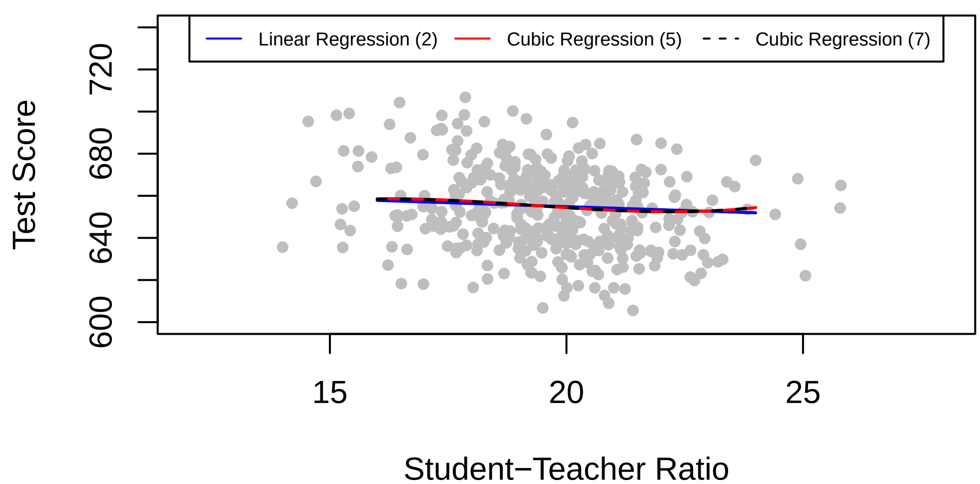
With a p-value of 0.08882 we can just reject the null hypothesis at the 10% level. This provides tentative evidence that the regression functions are different for districts with high and low percentages of English learners, but this will be further explored later.

In model (7), we employ a continuous measure for the proportion of English learners instead of a dummy variable (thus omitting interaction terms). We note minimal alterations in the coefficient estimates for the remaining regressors. Consequently, we infer that the findings observed in model (5) are robust and not influenced significantly by the method used to measure the percentage of English learners.

For better interpretation, we now plot the nonlinear specifications (2), (5) and (7) along with a scatterplot of the data, setting all regressors except STR to their sample averages. The plotted regression functions represent the predicted value of test scores as a function of the student-teacher ratio, holding fixed other values of the independent variables in the regression.

```
# scatterplot
plot(CASchools$STR,
     CASchools$score,
     xlim = c(12, 28),
     ylim = c(600, 740),
     pch = 20,
     col = "gray",
     xlab = "Student-Teacher Ratio",
     ylab = "Test Score")

# add a legend
legend("top",
       legend = c("Linear Regression (2)",
                  "Cubic Regression (5)",
                  "Cubic Regression (7)"),
       cex = 0.6,
       ncol = 3,
       lty = c(1, 1, 2),
       col = c("blue", "red", "black"))

# data for use with predict()
new_data <- data.frame("STR" = seq(16, 24, 0.05),
                       "english" = mean(CASchools$english),
                       "lunch" = mean(CASchools$lunch),
                       "income" = mean(CASchools$income),
                       "HiEL" = mean(CASchools$HiEL))

# add estimated regression function for model (2)
fitted <- predict(TS_mod2, newdata = new_data)

lines(new_data$STR,
      fitted,
      lwd = 1.5,
      col = "blue")
```

```r
# add estimated regression function for model (5)
fitted <- predict(TS_mod5, newdata = new_data)

lines(new_data$STR,
      fitted,
      lwd = 1.5,
      col = "red")

# add estimated regression function for model (7)
fitted <- predict(TS_mod7, newdata = new_data)

lines(new_data$STR,
      fitted,
      col = "black",
      lwd = 1.5,
      lty = 2)
```

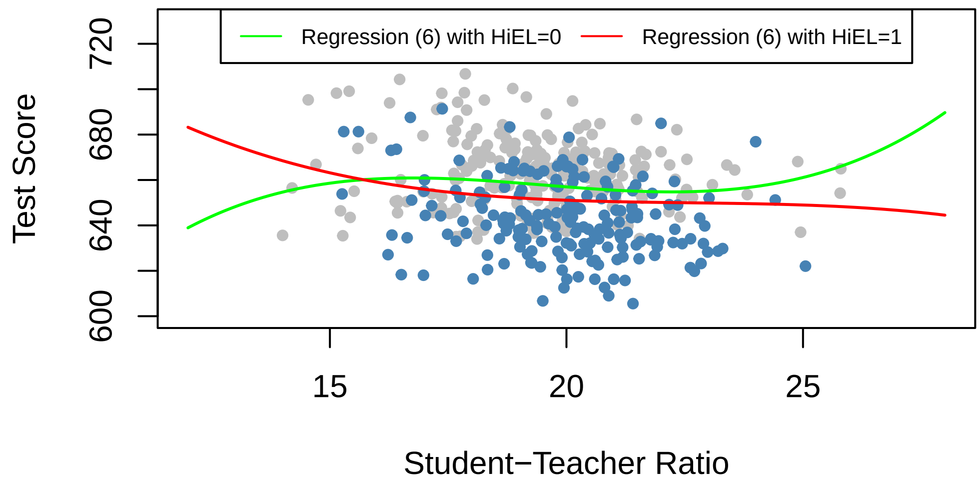


Cubic regressions (5) and (7) are represented by almost identical lines, and remarkably, all three estimated regression functions are close to one another. This may indicate that the relation between test scores and the student-teacher ratio has just a small amount of nonlinearity.

Regression (6) suggested that the cubic regression functions relating test scores and STR might depend on whether the percentage of English learners in the district is large or small, but the null was just rejected at the 10% level. We can further explore this by plotting the two estimated regression functions from this model and assessing the differences. Districts with low percentages of English learners ($HiEL = 0$) will be shown by gray dots, and districts with $HiEL = 1$ by colored dots. We use `plot()` and `points()` to color observations depending on $HiEL$, and we will make the predictions using the sample averages for all regressors except for STR, just as before.

```r
# draw scatterplot

# observations with HiEL = 0
plot(CASchools$STR[CASchools$HiEL == 0],
     CASchools$score[CASchools$HiEL == 0],
     xlim = c(12, 28),
     ylim = c(600, 730),
     pch = 20,
     col = "gray",
     xlab = "Student-Teacher Ratio",
     ylab = "Test Score")

# observations with HiEL = 1
points(CASchools$STR[CASchools$HiEL == 1],
       CASchools$score[CASchools$HiEL == 1],
       col = "steelblue",
       pch = 20)

# add a legend
legend("top",
       legend = c("Regression (6) with HiEL=0", "Regression (6) with HiEL=1"),
       cex = 0.7,
       ncol = 2,
       lty = c(1, 1),
       col = c("green", "red"))

# data for use with 'predict()'
new_data <- data.frame("STR" = seq(12, 28, 0.05),
                       "english" = mean(CASchools$english),
                       "lunch" = mean(CASchools$lunch),
                       "income" = mean(CASchools$income),
                       "HiEL" = 0)

# add estimated regression function for model (6) with HiEL=0
fitted <- predict(TS_mod6, newdata = new_data)

lines(new_data$STR,
      fitted,
      lwd = 1.5,
      col = "green")

# add estimated regression function for model (6) with HiEL=1
```

```r
new_data$HiEL <- 1

fitted <- predict(TS_mod6, newdata = new_data)

lines(new_data$STR,
      fitted,
      lwd = 1.5,
      col = "red")
```

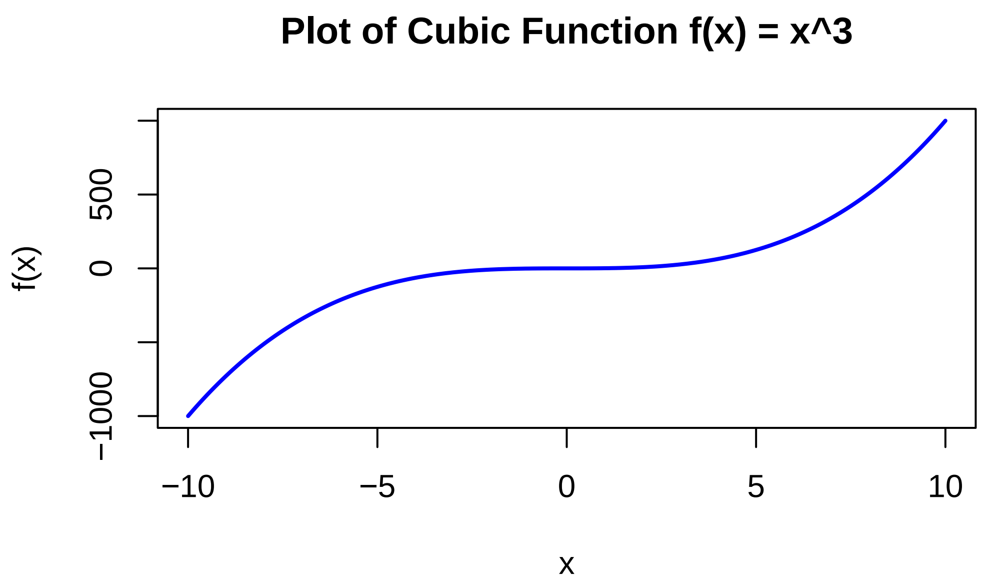


The plot shows that the difference between both isn't of practical importance in reality. It's a good example of why we need to be careful when understanding nonlinear models. Even though the two lines on the graph look different, they have almost the same slope between student-teacher ratios of 17 to 23. Since most of the data falls within this range, we can ignore any complicated relationships between the fraction of English learners and the student-teacher ratio. The two regression functions differ for student-teacher ratios below 17. However, we should be cautious not to draw conclusions beyond what is warranted. Districts with student-teacher ratios less than 16.5 make up only 6% of the total observations. Therefore, any discrepancies between the nonlinear regression functions primarily come from differences in these few districts with extremely low student-teacher ratios.

Additionally, the model is less accurate at the very low and very high ends of the data, since there aren't many observations there. This is a common problem with cubic functions - they can behave strangely at extreme values, as we can see in the graph of $f(x) = x^3$.

```r
# Define the range of x values
x <- seq(-10, 10, by = 0.1)

# Calculate the corresponding y values using the cubic function
```

```r
y <- x^3

# Plot the cubic function
plot(x, y, type = "l", col = "blue", lwd = 2,
     xlab = "x", ylab = "f(x)", main = "Plot of Cubic Function f(x) = x^3")
```



All in all, we conclude that the effect on test scores of a change in the student-teacher ratio does not depend on the percentage of English learners for the range of student-teacher ratios for which we have the most data.

3.4.1 Conclusions

We can now address the initial questions raised in this section:

First, in the linear models, the impact of the percentage of English learners on changes in test scores due to variations in the student-teacher ratio is minimal, a conclusion that holds true even after accounting for students' economic backgrounds. Although the cubic specification (6) suggests that the relationship between student-teacher ratio and test scores is influenced by the proportion of English learners, the magnitude of this influence is not significant.

Second, while controlling for students' economic backgrounds, we identify nonlinearities in the association between student-teacher ratio and test scores.

Lastly, under the **linear specification** (2), a reduction of two students per teacher in the student-teacher ratio is projected to increase test scores by approximately 1.46 points. As this model is linear, this effect remains consistent regardless of class size. For instance, assuming a student-teacher ratio of 20, the **nonlinear model** (5) indicates that the reduction in student-teacher ratio would lead to an increase in test scores by

$$64.33 \cdot 18 + 18^2 \cdot (-3.42) + 18^3 \cdot (0.059) - (64.33 \cdot 20 + 20^2 \cdot (-3.42) + 20^3 \cdot (0.059)) \approx 3.3$$

points. If the ratio was 22, a reduction to 20 leads to a predicted improvement in test scores of

$$64.33 \cdot 20 + 20^2 \cdot (-3.42) + 20^3 \cdot (0.059) - (64.33 \cdot 22 + 22^2 \cdot (-3.42) + 22^3 \cdot (0.059)) \approx 2.4$$

points. This suggests that the effect is more evident in smaller classes.

4 Empirical Applications in Panel Data Analysis

Welcome to the first empirical application in R! Here you will have the opportunity to bridge theory with practice by applying the concepts to real-world datasets available in R. This will help you better understanding the theory and hopefully motivate you to keep conducting your own applications in R.

Our journey begins always with a brief overview of each dataset, followed by simple analyses that progressively delve into more advanced applications. Along the way, you will find theory recaps to ensure you remember the essential concepts required for these applications.

Get ready to dive into the exciting world of empirical methods in R and enjoy the learning process.

Let's get started!

4.1 Dataset Description

The dataset `Fatalities`, contains panel data for traffic fatalities in the United States. Among others, it contains variables related to traffic fatalities and alcohol, including the number of traffic fatalities, the type of drunk driving laws and the tax on beer, reporting their values for each state and each year.

Here we will study how effective various government policies designed to discourage drunk driving actually are in reducing traffic deaths.

The measure of traffic deaths we use is the fatality rate, which is the annual number of traffic fatalities per 10,000 individuals within the state's population. The measure of alcohol taxes we use is the "real" tax on a case of beer, which is the beer tax, put into 1988 dollars by adjusting for inflation.

Let's start by loading the necessary packages and the dataset `fatalities`

```
# load the packages and the dataset
library(AER)
library(plm)
data(Fatalities)
```

First, we define the dataset as panel data, specifying the variables that should be used as index (in this case `state` and `year`). These will be used to organize the data frame, with each combination of state and year representing a unique observation in the panel.

```
# pdata.frame() declares the data as panel data.
Fatalities <- pdata.frame(Fatalities, index = c("state", "year"))
```

```
# inspect the structure and obtain the dimension
is.data.frame(Fatalities)
```

```
[1] TRUE
```

```
dim(Fatalities)
```

```
[1] 336  34
```

We can see the data has been effectively defined as a data frame, with 336 observations of 34 variables. For more detailed information on the variables inside the data frame, we could additionally call `str(Fatalities)`

It's always good to have a quick look at the first few observations. The `head()` function in R, by default, shows the first six observations (rows) of a data frame or data set. However, you can specify a different number of rows to display by providing the desired count as an argument to the function if needed, like `head(your_data_frame, n = 10)` to display the first 10 rows.

```
# list the first few observations
head(Fatalities)
```

```
        state year spirits unemp   income   emppop  beertax baptist  mormon
al-1982    al 1982    1.37  14.4 10544.15 50.69204 1.539379 30.3557 0.32829
al-1983    al 1983    1.36  13.7 10732.80 52.14703 1.788991 30.3336 0.34341
al-1984    al 1984    1.32  11.1 11108.79 54.16809 1.714286 30.3115 0.35924
al-1985    al 1985    1.28   8.9 11332.63 55.27114 1.652542 30.2895 0.37579
al-1986    al 1986    1.23   9.8 11661.51 56.51450 1.609907 30.2674 0.39311
al-1987    al 1987    1.18   7.8 11944.00 57.50988 1.560000 30.2453 0.41123
        drinkage     dry youngdrivers    miles breath jail service fatal nfatal
al-1982    19.00 25.0063     0.211572 7233.887     no   no      no   839    146
al-1983    19.00 22.9942     0.210768 7836.348     no   no      no   930    154
al-1984    19.00 24.0426     0.211484 8262.990     no   no      no   932    165
al-1985    19.67 23.6339     0.211140 8726.917     no   no      no   882    146
```

```
al-1986    21.00 23.4647     0.213400 8952.854     no    no       no  1081    172
al-1987    21.00 23.7924     0.215527 9166.302     no    no       no  1110    181
        sfatal fatal1517 nfatal1517 fatal1820 nfatal1820 fatal2124 nfatal2124
al-1982     99        53          9        99         34       120         32
al-1983     98        71          8       108         26       124         35
al-1984     94        49          7       103         25       118         34
al-1985     98        66          9       100         23       114         45
al-1986    119        82         10       120         23       119         29
al-1987    114        94         11       127         31       138         30
         afatal     pop  pop1517  pop1820  pop2124 milestot unempus emppopus
al-1982 309.438 3942002 208999.6 221553.4 290000.1    28516     9.7     57.8
al-1983 341.834 3960008 202000.1 219125.5 290000.2    31032     9.6     57.9
al-1984 304.872 3988992 197000.0 216724.1 288000.2    32961     7.5     59.5
al-1985 276.742 4021008 194999.7 214349.0 284000.3    35091     7.2     60.1
al-1986 360.716 4049994 203999.9 212000.0 263000.3    36259     7.0     60.7
al-1987 368.421 4082999 204999.8 208998.5 258999.8    37426     6.2     61.5
                gsp
al-1982 -0.02212476
al-1983  0.04655825
al-1984  0.06279784
al-1985  0.02748997
al-1986  0.03214295
al-1987  0.04897637
```

```r
# summarize the variables 'state' and 'year'
summary(Fatalities[, c(1, 2)])
```

```
     state        year   
 al     :  7   1982:48  
 az     :  7   1983:48  
 ar     :  7   1984:48  
 ca     :  7   1985:48  
 co     :  7   1986:48  
 ct     :  7   1987:48  
 (Other):294   1988:48  
```

Notice that the variable `state` is a factor variable with 48 levels (one for each of the 48 contiguous federal states of the U.S.). The variable `year` is also a factor variable that has 7 levels identifying the time period when the observation was made. This gives us $7 \times 48 = 336$ observations in total.

Since all variables are observed for all entities (states) and over all time periods, the panel is *balanced*. If there were missing data for at least one entity in at least one time period we would call the panel *unbalanced*.

Let's start by estimating simple regressions using data for years 1982 and 1988 that model the relationship between the beer tax (adjusted for 1988 dollars) and the traffic fatality rate, measured as the number of fatalities per 10000 inhabitants. Afterwards, we plot the data and add the corresponding estimated regression functions.

```
# define the fatality rate
Fatalities$fatal_rate <- Fatalities$fatal / Fatalities$pop * 10000

# subset the data
Fatalities1982 <- subset(Fatalities, year == "1982")
Fatalities1988 <- subset(Fatalities, year == "1988")

# estimate simple regression models using 1982 and 1988 data
fatal1982_mod <- lm(fatal_rate ~ beertax, data = Fatalities1982)
fatal1988_mod <- lm(fatal_rate ~ beertax, data = Fatalities1988)

coeftest(fatal1982_mod, vcov. = vcovHC, type = "HC1")
```

```
t test of coefficients:

            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.01038    0.14957 13.4408   <2e-16 ***
beertax      0.14846    0.13261  1.1196   0.2687
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(fatal1988_mod, vcov. = vcovHC, type = "HC1")
```

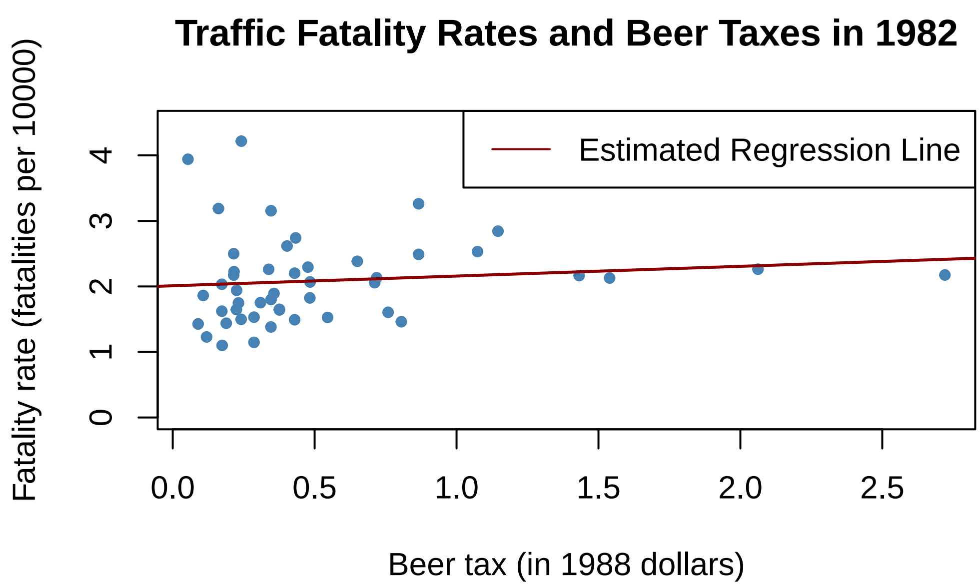
```
t test of coefficients:

            Estimate Std. Error t value  Pr(>|t|)
(Intercept)  1.85907    0.11461 16.2205 < 2.2e-16 ***
beertax      0.43875    0.12786  3.4314  0.001279 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated regression functions are

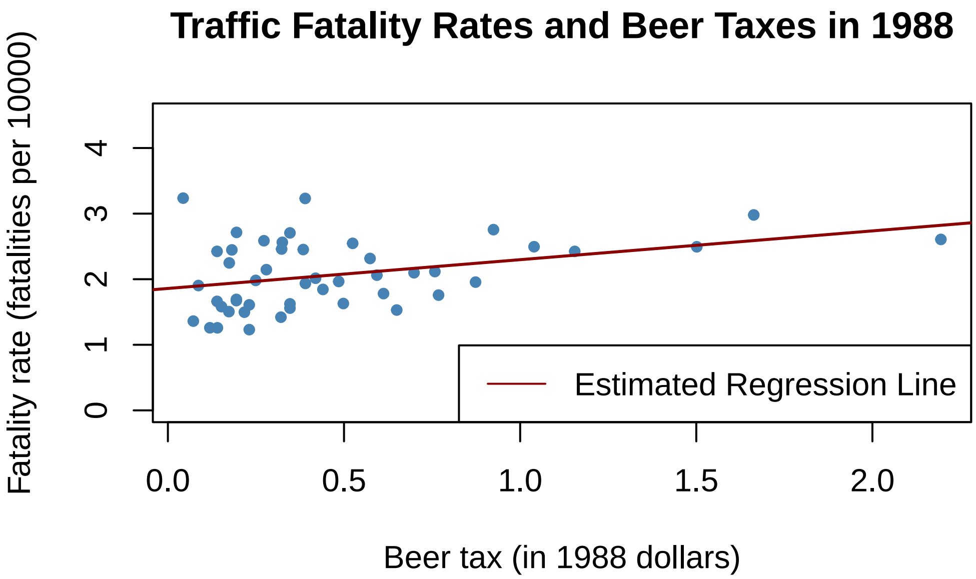
$$\widehat{FatalityRate} = \underset{(0.15)}{2.01} + \underset{(0.13)}{0.15}\, BeerTax \quad \text{(1982 data)}$$

$$\widehat{FatalityRate} = \underset{(0.11)}{1.86} + \underset{(0.13)}{0.44}\, BeerTax \quad \text{(1988 data)}$$

```r
# plot the observations and add the estimated regression line for 1982 data
plot(x = as.double(Fatalities1982$beertax), y = as.double(Fatalities1982$fatal_rate),
     xlab = "Beer tax (in 1988 dollars)", ylab = "Fatality rate (fatalities per 10000)",
     main = "Traffic Fatality Rates and Beer Taxes in 1982", ylim = c(0, 4.5),
     pch = 20, col = "steelblue")
abline(fatal1982_mod, lwd = 1.5, col="darkred")
legend("topright",lty=1,col="darkred","Estimated Regression Line")
```




```r
# plot observations and add estimated regression line for 1988 data
plot(x = as.double(Fatalities1988$beertax), y = as.double(Fatalities1988$fatal_rate),
     xlab = "Beer tax (in 1988 dollars)", ylab = "Fatality rate (fatalities per 10000)",
     main = "Traffic Fatality Rates and Beer Taxes in 1988", ylim = c(0, 4.5),
     pch = 20, col = "steelblue")

abline(fatal1988_mod, lwd = 1.5,col="darkred") # add the regression line to plot
legend("bottomright",lty=1,col="darkred","Estimated Regression Line") # add legend
```



In both plots, each point represents observations of beer tax and fatality rate for a given state in the respective year. The regression results indicate a positive relationship between the beer tax and the fatality rate for both years.

The estimated coefficient on beer tax for the 1988 data is almost three times as large as for the 1982 dataset. This is contrary to our expectations: alcohol taxes are supposed to lower the rate of traffic fatalities. This is possibly due to omitted variable bias, since none of the models include any covariates, e.g., economic conditions.

This could be corrected for using a multiple regression approach. However, this cannot account for omitted unobservable factors that differ from state to state but can be assumed to be constant over the observation span, e.g., the populations' attitude towards drunk driving. As shown in the next section, panel data allow us to hold such factors constant.

4.2 Two Time Periods: "Before and After" Comparisons

Let's suppose there are only $T = 2$ time periods $t = 1982, 1988$. This allows us to analyze differences in changes of the fatality rate from year 1982 to 1988. We start by considering the population regression model:

$$\text{FatalityRate}_{it} = \beta_0 + \beta_1 \text{BeerTax}_{it} + \beta_2 Z_i + u_{it}$$

where the Z_i are state specific characteristics that differ between states but are constant over time. For $t = 1982$ and $t = 1988$ we have

$$FatalityRate_{i,1982} = \beta_0 + \beta_1 \text{BeerTax}_{i,1982} + \beta_2 Z_i + u_{i,1982},$$
$$FatalityRate_{i,1988} = \beta_0 + \beta_1 \text{BeerTax}_{i,1988} + \beta_2 Z_i + u_{i,1988}.$$

We can eliminate the Z_i by regressing the difference in the fatality rate between 1988 and 1982 on the difference in beer tax between those years:

$$\text{FatalityRate}_{i,1988} - \text{FatalityRate}_{i,1982} = \beta_1(\text{BeerTax}_{i,1988} - \text{BeerTax}_{i,1982}) + u_{i,1988} - u_{i,1982}$$

This regression model, where the difference in fatality rate between 1988 and 1982 is regressed on the difference in beer tax between those years, yields an estimate for β_1 that is robust to a possible bias due to omission of Z_i, as these influences are eliminated from the model. Next we will estimate a regression based on the differenced data and plot the estimated regression function.

```r
# compute the differences
diff_fatal_rate <- Fatalities1988$fatal_rate - Fatalities1982$fatal_rate
diff_beertax <- Fatalities1988$beertax - Fatalities1982$beertax

# estimate a regression using differenced data
fatal_diff_mod <- lm(diff_fatal_rate ~ diff_beertax)
coeftest(fatal_diff_mod, vcov = vcovHC, type = "HC1")
```

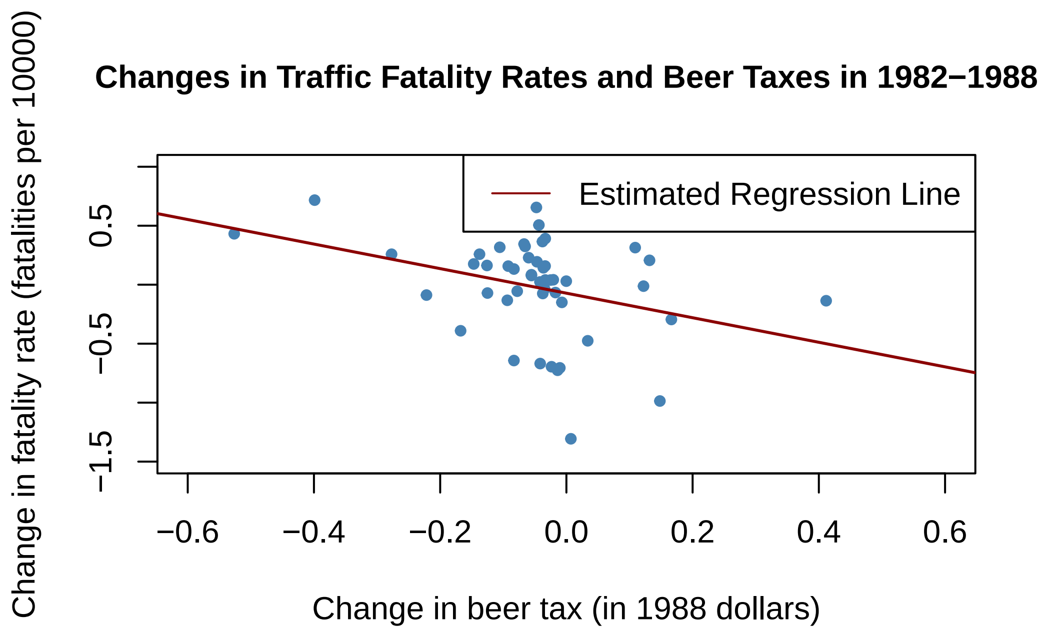
```
t test of coefficients:

              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.072037   0.065355 -1.1022 0.276091
diff_beertax -1.040973   0.355006 -2.9323 0.005229 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Including the intercept allows for a change in the mean fatality rate in the time between 1982 and 1988 in the absence of a change in the beer tax.

We obtain the OLS estimated regression function

$$\widehat{FatalityRate_{i1988} - FatalityRate_{i1982}} = \underset{(-0.065)}{0.072} - \underset{(0.36)}{1.04} \times (BeerTax_{i1988} - BeerTax_{i1982})$$

```r
# plot the differenced data
plot(x = as.double(diff_beertax), y = as.double(diff_fatal_rate),
     xlab = "Change in beer tax (in 1988 dollars)",ylab = "Change in fatality rate (fataliti
     main = "Changes in Traffic Fatality Rates and Beer Taxes in 1982-1988", cex.main=1,
     xlim = c(-0.6, 0.6), ylim = c(-1.5, 1), pch = 20, col = "steelblue")

abline(fatal_diff_mod, lwd = 1.5,col="darkred") # add the regression line to plot
legend("topright",lty=1,col="darkred","Estimated Regression Line") #add legend
```



The estimated coefficient on beer tax is now negative and significantly different from zero at the 1% significance level. Its interpretation is that raising the beer tax by $1 is associated with an average decrease of 1.04 fatalities per $10,000$ inhabitants. This is rather large as the average fatality rate is approximately 2 persons per $10,000$ inhabitants.

```r
# compute mean fatality rate over all states for all time periods
mean(Fatalities$fatal_rate)
```

```
[1] 2.040444
```

The outcome we obtained is likely to be a consequence of omitting factors in the single-year regression that influence the fatality rate and are correlated with the beer tax and change over time. The message is that we need to be more careful and control for such factors before drawing conclusions about the effect of a raise in beer taxes.

The approach presented in this section discards information for years 1983 to 1987. The fixed effects method allows us to use data for more than $T = 2$ time periods and enables us to add control variables to the analysis.

4.3 Fixed Effects Regression

Consider the panel regression model:

$$Y_{it} = \beta_0 + \beta_1 X_{it} + \beta_2 Z_i + u_{it} \tag{3.1}$$

where the Z_i are unobserved time-invariant heterogeneities across the entities $i = 1, \dots, n$. We aim to estimate β_1, the effect on Y_i of a change in X_i, holding constant Z_i. Letting $\alpha_i = \beta_0 + \beta_2 Z_i$, we obtain the model

$$Y_{it} = \alpha_i + \beta_1 X_{it} + u_{it} \tag{3.2}$$

Having individual specific intercepts $\alpha_i, i = 1, \dots, n$, where each of these can be understood as the fixed effect of entity i.

The Fixed Effects Regression Model is

$$Y_{it} = \beta_1 X_{1,it} + \cdots + \beta_k X_{k,it} + \alpha_i + u_{it} \tag{3.3}$$

with $i = 1, \dots, n$ and $t = 1, \dots, T$. The α_i are entity-specific intercepts that capture heterogeneities across entities. An equivalent representation of this model is given by

$$Y_{it} = \beta_0 + \beta_1 X_{1,it} + \cdots + \beta_k X_{k,it} + \gamma_2 D_{2i} + \gamma_3 D_{3i} + \cdots + \gamma_n D_{ni} + u_{it} \tag{3.4}$$

where the $D_{2i}, D_{3i}, \dots, D_{ni}$ are dummy variables.

To estimate the relation between traffic fatality rates and beer taxes, the simple fixed effects model is

$$FatalityRate_{it} = \beta_1 BeerTax_{it} + StateFixedEffects + u_{it} \tag{3.5}$$

a regression of the traffic fatality rate on beer tax and 48 binary regressors (one for each federal state). In this model, we are using a fixed effects approach to account for the effect of each federal state. Including a fixed effect for each state means that we're estimating separate intercepts (or constant terms) for each state.

In R, we can simply use the function `lm()` to obtain an estimate of β_1.

```r
fatal_fe_lm_mod <- lm(fatal_rate ~ beertax + state - 1, data = Fatalities)
```

The `-1` term tells R to exclude the intercept term that it would normally include by default. By doing this, we're essentially saying that we don't want to estimate an overall intercept for the model because we are already capturing the state-specific effects. This is a common practice in fixed effects models to avoid multicollinearity between the state-specific intercepts and the predictors.

```r
summary(fatal_fe_lm_mod)
```

Click here to view or hide summary output

```
::: {.cell}

```{.r .cell-code}
summary(fatal_fe_lm_mod)
```

::: {.cell-output .cell-output-stdout}

```

Call:
lm(formula = fatal_rate ~ beertax + state - 1, data = Fatalities)

Residuals:
     Min       1Q   Median       3Q      Max
-0.58696 -0.08284 -0.00127  0.07955  0.89780

Coefficients:
        Estimate Std. Error t value Pr(>|t|)
beertax -0.65587    0.18785  -3.491 0.000556 ***
stateal  3.47763    0.31336  11.098  < 2e-16 ***
stateaz  2.90990    0.09254  31.445  < 2e-16 ***
statear  2.82268    0.13213  21.364  < 2e-16 ***
stateca  1.96816    0.07401  26.594  < 2e-16 ***
stateco  1.99335    0.08037  24.802  < 2e-16 ***
statect  1.61537    0.08391  19.251  < 2e-16 ***
statede  2.17003    0.07746  28.016  < 2e-16 ***
statefl  3.20950    0.22151  14.489  < 2e-16 ***
```
```

```
stategа  4.00223    0.46403   8.625 4.43e-16 ***
stateid  2.80861    0.09877  28.437  < 2e-16 ***
stateil  1.51601    0.07848  19.318  < 2e-16 ***
statein  2.01609    0.08867  22.736  < 2e-16 ***
stateia  1.93370    0.10222  18.918  < 2e-16 ***
stateks  2.25441    0.10863  20.753  < 2e-16 ***
stateky  2.26011    0.08046  28.089  < 2e-16 ***
statela  2.63051    0.16266  16.171  < 2e-16 ***
stateme  2.36968    0.16006  14.805  < 2e-16 ***
statemd  1.77119    0.08246  21.480  < 2e-16 ***
statema  1.36788    0.08648  15.818  < 2e-16 ***
statemi  1.99310    0.11663  17.089  < 2e-16 ***
statemn  1.58042    0.09363  16.880  < 2e-16 ***
statems  3.44855    0.20936  16.472  < 2e-16 ***
statemo  2.18137    0.09252  23.576  < 2e-16 ***
statemt  3.11724    0.09441  33.017  < 2e-16 ***
statene  1.95545    0.10551  18.534  < 2e-16 ***
statenv  2.87686    0.08106  35.492  < 2e-16 ***
statenh  2.22318    0.14114  15.751  < 2e-16 ***
statenj  1.37188    0.07333  18.709  < 2e-16 ***
statenm  3.90401    0.10154  38.449  < 2e-16 ***
stateny  1.29096    0.07563  17.070  < 2e-16 ***
statenc  3.18717    0.25173  12.661  < 2e-16 ***
statend  1.85419    0.10193  18.191  < 2e-16 ***
stateoh  1.80321    0.10193  17.691  < 2e-16 ***
stateok  2.93257    0.18428  15.913  < 2e-16 ***
stateor  2.30963    0.08117  28.453  < 2e-16 ***
statepa  1.71016    0.08648  19.776  < 2e-16 ***
stateri  1.21258    0.07753  15.640  < 2e-16 ***
statesc  4.03480    0.35479  11.372  < 2e-16 ***
statesd  2.47391    0.14121  17.519  < 2e-16 ***
statetn  2.60197    0.09162  28.398  < 2e-16 ***
statetx  2.56016    0.10853  23.589  < 2e-16 ***
stateut  2.31368    0.15453  14.972  < 2e-16 ***
statevt  2.51159    0.13973  17.975  < 2e-16 ***
stateva  2.18745    0.14664  14.917  < 2e-16 ***
statewa  1.81811    0.08233  22.084  < 2e-16 ***
statewv  2.58088    0.10767  23.971  < 2e-16 ***
statewi  1.71836    0.07746  22.185  < 2e-16 ***
statewy  3.24913    0.07233  44.922  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.1899 on 287 degrees of freedom
Multiple R-squared:  0.9931,    Adjusted R-squared:  0.992
F-statistic: 847.8 on 49 and 287 DF,  p-value: < 2.2e-16
```

:::
:::

It is also possible to estimate β_1 by applying OLS to the demeaned data, that is, to run the regression

$$\widetilde{FatalityRate} = \beta_1 \widetilde{BeertTax}_{it} + u_{it}$$

```r
# obtain demeaned data
fatal_demeaned <- with(Fatalities,
            data.frame(fatal_rate = fatal_rate - ave(fatal_rate, state),
            beertax = beertax - ave(beertax, state)))

# estimate the regression
summary(lm(fatal_rate ~ beertax - 1, data = fatal_demeaned))
```

```

Call:
lm(formula = fatal_rate ~ beertax - 1, data = fatal_demeaned)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.58696 -0.08284 -0.00127  0.07955  0.89780 

Coefficients:
        Estimate Std. Error t value Pr(>|t|)    
beertax  -0.6559     0.1739  -3.772 0.000191 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1757 on 335 degrees of freedom
Multiple R-squared:  0.04074,   Adjusted R-squared:  0.03788 
F-statistic: 14.23 on 1 and 335 DF,  p-value: 0.0001913
```

The function `ave` is convenient for computing group averages. We use it to obtain state specific averages of the fatality rate and the beer tax.

The estimated coefficient is again -0.6559. The estimated regression function is

$$\widehat{FatalityRate} = -\underset{(0.17)}{0.66}\, BeerTax + StateFixedEffects \tag{3.6}$$

The coefficient on $BeerTax$ is negative and statistically significant at the $0,1\%$ level. Its interpretation is that raising the beer tax by \$1 is associated with an average decrease of 0.66 fatalities per $10,000$ people in traffic fatalities, which is still pretty high.

Although including state fixed effects eliminates the risk of a bias due to omitted factors that vary across states but not over time, we suspect that there are other omitted variables that vary over time and thus cause a bias.

4.4 Time Fixed Effects

Controlling for variables that are constant across entities but vary over time can be done by including time fixed effects. If there are *only* time fixed effects, the fixed effects regression model becomes

$$Y_{it} = \beta_0 + \beta_1 X_{it} + \delta_2 B_{2t} + \cdots + \delta_T BT_t + u_{it}$$

where only $T-1$ dummies are included ($B1$ is omitted) since the model includes an intercept. This model eliminates omitted variable bias caused by excluding unobserved variables that evolve over time but are constant across entities.

In some applications it is meaningful to include both entity (state) and time fixed effects. The **entity and time fixed effects model** is

$$Y_{it} = \beta_0 + \beta_1 X_{it} + \gamma_2 D_{2i} + \cdots + \gamma_n DT_i + \delta_2 B2_t + \cdots + \delta_T BT_t + u_{it}$$

The combined model allows to eliminate bias from unobservables that change over time but are constant over entities and it controls for factors that differ across entities but are constant over time. Such models can be estimated using the OLS algorithm that is implemented in R.

Let's estimate the combined entity and time fixed effects model of the relation between fatalities and beer tax,

$$FatalityRate_{it} = \beta_1 BeerTax_{it} + StateFixedEffects + TimeFixedEffects + u_{it}$$

It is straightforward to estimate this regression with `lm()`. We just have to adjust the `formula` argument by adding the additional regressor `year` for time fixed effects:

```r
# estimate a combined time and entity fixed effects regression model
fatal_tefe_lm_mod <- lm(fatal_rate ~ beertax + state + year - 1, data = Fatalities)
```

```r
summary(fatal_tefe_lm_mod)
```

Click here to view or hide summary output

```
::: {.cell}

```{.r .cell-code}
summary(fatal_tefe_lm_mod)
```

::: {.cell-output .cell-output-stdout}

```

Call:
lm(formula = fatal_rate ~ beertax + state + year - 1, data = Fatalities)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.59556 -0.08096  0.00143  0.08234  0.83883 

Coefficients:
         Estimate Std. Error t value Pr(>|t|)    
beertax  -0.63998    0.19738  -3.242  0.00133 ** 
stateal   3.51137    0.33250  10.560  < 2e-16 ***
stateaz   2.96451    0.09933  29.846  < 2e-16 ***
statear   2.87284    0.14162  20.286  < 2e-16 ***
stateca   2.02618    0.07857  25.787  < 2e-16 ***
stateco   2.04984    0.08594  23.851  < 2e-16 ***
statect   1.67125    0.08989  18.592  < 2e-16 ***
statede   2.22711    0.08264  26.951  < 2e-16 ***
statefl   3.25132    0.23590  13.782  < 2e-16 ***
statega   4.02300    0.49087   8.196 8.92e-15 ***
stateid   2.86242    0.10606  26.990  < 2e-16 ***
stateil   1.57287    0.08380  18.769  < 2e-16 ***
statein   2.07123    0.09512  21.775  < 2e-16 ***
```
```

```
stateia   1.98709    0.10976  18.103  < 2e-16 ***
stateks   2.30707    0.11663  19.781  < 2e-16 ***
stateky   2.31659    0.08604  26.923  < 2e-16 ***
statela   2.67772    0.17390  15.398  < 2e-16 ***
stateme   2.41713    0.17116  14.122  < 2e-16 ***
statemd   1.82731    0.08828  20.700  < 2e-16 ***
statema   1.42335    0.09272  15.352  < 2e-16 ***
statemi   2.04488    0.12516  16.338  < 2e-16 ***
statemn   1.63488    0.10051  16.266  < 2e-16 ***
statems   3.49146    0.22311  15.649  < 2e-16 ***
statemo   2.23598    0.09931  22.515  < 2e-16 ***
statemt   3.17160    0.10136  31.291  < 2e-16 ***
statene   2.00846    0.11329  17.729  < 2e-16 ***
statenv   2.93322    0.08671  33.827  < 2e-16 ***
statenh   2.27245    0.15116  15.033  < 2e-16 ***
statenj   1.43016    0.07773  18.399  < 2e-16 ***
statenm   3.95748    0.10903  36.296  < 2e-16 ***
stateny   1.34849    0.08051  16.748  < 2e-16 ***
statenc   3.22630    0.26770  12.052  < 2e-16 ***
statend   1.90762    0.10945  17.428  < 2e-16 ***
stateoh   1.85664    0.10945  16.963  < 2e-16 ***
stateok   2.97776    0.19670  15.139  < 2e-16 ***
stateor   2.36597    0.08684  27.244  < 2e-16 ***
statepa   1.76563    0.09272  19.044  < 2e-16 ***
stateri   1.26964    0.08272  15.348  < 2e-16 ***
statesc   4.06496    0.37606  10.809  < 2e-16 ***
statesd   2.52317    0.15123  16.684  < 2e-16 ***
statetn   2.65670    0.09833  27.017  < 2e-16 ***
statetx   2.61282    0.11653  22.423  < 2e-16 ***
stateut   2.36165    0.16532  14.286  < 2e-16 ***
statevt   2.56100    0.14966  17.112  < 2e-16 ***
stateva   2.23618    0.15698  14.245  < 2e-16 ***
statewa   1.87424    0.08813  21.266  < 2e-16 ***
statewv   2.63364    0.11560  22.782  < 2e-16 ***
statewi   1.77545    0.08264  21.485  < 2e-16 ***
statewy   3.30791    0.07641  43.291  < 2e-16 ***
year1983 -0.07990    0.03835  -2.083  0.03813 *
year1984 -0.07242    0.03835  -1.888  0.06001 .
year1985 -0.12398    0.03844  -3.225  0.00141 **
year1986 -0.03786    0.03859  -0.981  0.32731
year1987 -0.05090    0.03897  -1.306  0.19260
year1988 -0.05180    0.03962  -1.307  0.19215
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1879 on 281 degrees of freedom
Multiple R-squared:  0.9934,    Adjusted R-squared:  0.9921
F-statistic: 771.5 on 55 and 281 DF,  p-value: < 2.2e-16
```

:::
:::

Before discussing the outcomes we convince ourselves that state and year are of the class factor

```
# check the class of 'state' and 'year'
class(Fatalities$state)
```

```
[1] "pseries" "factor"
```

```
class(Fatalities$year)
```

```
[1] "pseries" "factor"
```

The `lm()` functions converts factors into dummies automatically. Since we exclude the intercept by adding -1 to the right-hand side of the regression formula, `lm()` estimates coefficients for $n + (T - 1) = 48 + 6 = 54$ binary variables (6 year dummies and 48 state dummies).

The estimated regression function is

$$\widehat{FatalityRate} = -\underset{(0.20)}{0.64} \, BeerTax + StateEffects + TimeFixedEffects \tag{3.7}$$

The result is close to the estimated coefficient for the regression model including only entity fixed effects, which was -0.66. Unsurprisingly, the coefficient is less precisely estimated, as we observe a slightly superior standard deviation for this new coefficient of -0.64. Nevertheless, it is still significantly different from zero at 1% level.

We conclude that the estimated relationship between traffic fatalities and the real beer tax is not affected by omitted variable bias due to factors that are constant either over time or across states.

4.5 Driving Laws and Economic Conditions

There are two major sources of omitted variable bias that are not accounted for by all of the models of the relation between traffic fatalities and beer taxes that we have considered so far: economic conditions and driving laws.

Fortunately, `Fatalities` has data on state-specific legal drinking age (`drinkage`), punishment (`jail`, `service`) and various economic indicators like unemployment rate (`unemp`) and per capita income (`income`). We may use these covariates to extend the preceding analysis.

These covariates are defined as follows:

- `unemp`: a numeric variable stating the state specific unemployment rate.
- `log(income)`: the logarithm of real per capita income (in 1988 dollars).
- `miles`: the state average miles per driver.
- `drinkage`: the state specific minimum legal drinking age.
- `drinkagec`: a discretized version of `drinkage` that classifies states into four categories of minimal drinking age; 18, 19, 20, 21 and older. R denotes this as `[18,19)`, `[19,20)`, `[20,21)` and `[21,22]`. These categories are included as dummy regressors where `[21,22]` is chosen as the reference category.
- `punish`: a dummy variable with levels yes and no that measures if drunk driving is severely punished by mandatory jail time or mandatory community service (first conviction).

First, we define some relevant variables to include in our following regression models:

```
# discretize the minimum legal drinking age
Fatalities$drinkagec <- cut(Fatalities$drinkage, breaks = 18:22, include.lowest = TRUE, righ

# set minimum drinking age [21, 22] to be the baseline level
Fatalities$drinkagec <- relevel(Fatalities$drinkagec, "[21,22]")

# mandatory jail or community service?
Fatalities$punish <- with(Fatalities, factor(jail == "yes" | service == "yes", labels = c("no

# the set of observations on all variables for 1982 and 1988
fatal_1982_1988 <- Fatalities[with(Fatalities, year == 1982 | year == 1988), ]
```

Next, we estimate six regression models using `plm()`.

```
# estimate all seven models
fat_mod1 <- lm(fatal_rate ~ beertax, data = Fatalities)
```

```r
fat_mod2 <- plm(fatal_rate ~ beertax + state, data = Fatalities)

fat_mod3 <- plm(fatal_rate ~ beertax + state + year,
                       index = c("state","year"), model = "within",
                       effect = "twoways", data = Fatalities)

fat_mod4 <- plm(fatal_rate ~ beertax + state + year + drinkagec
                       + punish + miles + unemp + log(income),
                       index = c("state", "year"), model = "within",
                       effect = "twoways", data = Fatalities)

fat_mod5 <- plm(fatal_rate ~ beertax + state + year + drinkagec
                       + punish + miles,
                       index = c("state", "year"), model = "within",
                       effect = "twoways", data = Fatalities)

fat_mod6 <- plm(fatal_rate ~ beertax + year + drinkage
                       + punish + miles + unemp + log(income),
                       index = c("state", "year"), model = "within",
                       effect = "twoways", data = Fatalities)
```

We use `stargazer()` to generate a comprehensive tabular presentation of the results.

```r
#load stargazer package
library(stargazer)

# gather clustered standard errors in a list
rob_se <- list(sqrt(diag(vcovHC(fat_mod1, type = "HC1"))),
               sqrt(diag(vcovHC(fat_mod2, type = "HC1"))),
               sqrt(diag(vcovHC(fat_mod3, type = "HC1"))),
               sqrt(diag(vcovHC(fat_mod4, type = "HC1"))),
               sqrt(diag(vcovHC(fat_mod5, type = "HC1"))),
               sqrt(diag(vcovHC(fat_mod6, type = "HC1"))))

# generate the table
stargazer(fat_mod1, fat_mod2, fat_mod3, fat_mod4, fat_mod5, fat_mod6,
          se = rob_se,
          type="html",
          omit.stat = "f", df=FALSE)
```

```
<table style="text-align:center"><tr><td colspan="7" style="border-bottom: 1px solid black">
<tr><td></td><td colspan="6" style="border-bottom: 1px solid black"></td></tr>
<tr><td style="text-align:left"></td><td colspan="6">fatal_rate</td></tr>
<tr><td style="text-align:left"></td><td><em>OLS</em></td><td colspan="5"><em>panel</em></td
<tr><td style="text-align:left"></td><td><em></em></td><td colspan="5"><em>linear</em></td><
<tr><td style="text-align:left"></td><td>(1)</td><td>(2)</td><td>(3)</td><td>(4)</td><td>(5)
<tr><td colspan="7" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-ali
<tr><td style="text-align:left"></td><td>(0.053)</td><td>(0.289)</td><td>(0.350)</td><td>(0.
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td style="text-align:left">drinkagec[18,19)</td><td></td><td></td><td></td><td>0.028</t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(0.068)</td><td>(0.081)<
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td style="text-align:left">drinkagec[19,20)</td><td></td><td></td><td></td><td>-0.018</
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(0.049)</td><td>(0.066)<
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td style="text-align:left">drinkagec[20,21)</td><td></td><td></td><td></td><td>0.032</t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(0.050)</td><td>(0.055)<
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td style="text-align:left">drinkage</td><td></td><td></td><td></td><td></td><td></td><t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td>(0.021
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td style="text-align:left">punishyes</td><td></td><td></td><td></td><td>0.038</td><td>0
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(0.101)</td><td>(0.109)<
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td style="text-align:left">miles</td><td></td><td></td><td></td><td>0.00001</td><td>0.0
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(0.00001)</td><td>(0.000
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td style="text-align:left">unemp</td><td></td><td></td><td></td><td>-0.063<sup>***</sup
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(0.013)</td><td></td><td
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td style="text-align:left">log(income)</td><td></td><td></td><td></td><td>1.816<sup>***
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(0.624)</td><td></td><td
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td style="text-align:left">Constant</td><td>1.853<sup>***</sup></td><td></td><td></td><
<tr><td style="text-align:left"></td><td>(0.047)</td><td></td><td></td><td></td><td></td><td
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td colspan="7" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-ali
<tr><td style="text-align:left">R<sup>2</sup></td><td>0.093</td><td>0.041</td><td>0.036</td>
<tr><td style="text-align:left">Adjusted R<sup>2</sup></td><td>0.091</td><td>-0.120</td><td>
<tr><td style="text-align:left">Residual Std. Error</td><td>0.544</td><td></td><td></td><td>
<tr><td colspan="7" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-ali
</table>
```

While columns 2 and 3 recap the results of regressions 3.6 and 3.7, column 1 presents an estimate of the coefficient of interest in the naive OLS regression of the fatality rate on beer tax without any fixed effects. There we obtain a positive estimate for the coefficient on beer tax that is likely to be upward biased.

The sign of the estimate changes as we extend the model by both entity and time fixed effects in models 2 and 3. Nonetheless, as discussed before, the magnitudes of both estimates may be too large.

The model specifications 4 to 6 include covariates that shall capture the effect of overall state economic conditions as well as the legal framework. Nevertheless, considering **model 4** as the baseline specification including covariates, we observe **four interesting results**:

1. Including these covariates is not leading to a major reduction of the estimated effect of the beer tax. The coefficient is not significantly different from zero at the 10% level, which means that it is considered imprecise.

2. According to this regression model, the minimum legal drinking age is not associated with an effect on traffic fatalities: none of the three dummy variables are significantly different from zero at any common level of significance. Moreover, an F-Test of the joint hypothesis that all three coefficients are zero does not reject the null hypothesis. The next code chunk shows how to test this hypothesis:

```
# test if legal drinking age has no explanatory power
linearHypothesis(fat_mod4,test = "F",
                 c("drinkagec[18,19)=0", "drinkagec[19,20)=0", "drinkagec[20,21)"),
                 vcov. = vcovHC, type = "HC1")
```

```
Linear hypothesis test

Hypothesis:
drinkagec[18,19) = 0
drinkagec[19,20) = 0
drinkagec[20,21) = 0

Model 1: restricted model
Model 2: fatal_rate ~ beertax + state + year + drinkagec + punish + miles +
    unemp + log(income)

Note: Coefficient covariance matrix supplied.

  Res.Df Df      F Pr(>F)
1    276
2    273  3 0.3782 0.7688
```

3. There is no statistical evidence indicating an association between punishment for first offenders and drunk driving: the corresponding coefficient is not significant at the 10% level.

4. The coefficients on the economic variables representing employment rate and income per capita indicate an statistically significant association between these and traffic fatalities. We can check that the employment rate and income per capita coefficients are jointly significant at the 0.1% level.

```r
# test if economic indicators have no explanatory power
linearHypothesis(fat_mod4, test = "F",
                 c("log(income)", "unemp"), vcov. = vcovHC, type = "HC1")
```

```
Linear hypothesis test

Hypothesis:
log(income) = 0
unemp = 0

Model 1: restricted model
Model 2: fatal_rate ~ beertax + state + year + drinkagec + punish + miles +
    unemp + log(income)

Note: Coefficient covariance matrix supplied.

  Res.Df Df      F    Pr(>F)
1    275
2    273  2 31.577 4.609e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 5 omits the economic factors. The result supports the notion that economic indicators should remain in the model as the coefficient on beer tax is sensitive to the inclusion of the latter.

Results for model 6 show that the legal drinking age has little explanatory power and that the coefficient of interest is not sensitive to changes in the functional form of the relation between drinking age and traffic fatalities.

4.6 Summary

We have not found statistical evidence to state that severe punishments and an increase the minimum drinking age could lead to a reduction of traffic fatalities due to drunk driving.

Nonetheless, there seems to be a negative effect of alcohol taxes on traffic fatalities according to our model estimate, However, this estimate is not precise and cannot be interpreted as the causal effect of interest, as there still may be a bias.

The issue is that there may be omitted variables that differ across states *and* change over time, and this bias remains even though we use a panel approach that controls for entity specific and time invariant unobservables.

A powerful method that can be used if common panel regression approaches fail is instrumental variables regression, which we will see in the next chapters.

5 Empirical Applications of Binary Regressions

In this chapter we will apply the concepts of binary regressions, those regression models that aim to explain a limited dependent variable. In particular, regression models where the dependent variable is binary. For this purpose, we will use a data set available in R called HDMA (Cross-section data on the Home Mortgage Disclosure Act).

5.1 Data Set Description

The data set HMDA provides data related to mortgage applications filed in Boston in 1990.

```
# load packages and attach the HMDA data
library(AER)
library(stargazer)
data(HMDA)
```

Let's start inspecting the first few observations and computing summary statistics.

```
#first observations
head(HMDA)
```

```
  deny pirat hirat     lvrat chist mhist phist unemp selfemp insurance condomin
1   no 0.221 0.221 0.8000000     5     2    no   3.9      no        no       no
2   no 0.265 0.265 0.9218750     2     2    no   3.2      no        no       no
3   no 0.372 0.248 0.9203980     1     2    no   3.2      no        no       no
4   no 0.320 0.250 0.8604651     1     2    no   4.3      no        no       no
5   no 0.360 0.350 0.6000000     1     1    no   3.2      no        no       no
6   no 0.240 0.170 0.5105263     1     1    no   3.9      no        no       no
  afam single hschool
1   no     no     yes
2   no    yes     yes
3   no     no     yes
4   no     no     yes
5   no     no     yes
6   no     no     yes
```

```r
#summary statistics
summary(HMDA)
```

```
  deny          pirat            hirat            lvrat         chist
 no :2095   Min.   :0.0000   Min.   :0.0000   Min.   :0.0200   1:1353
 yes: 285   1st Qu.:0.2800   1st Qu.:0.2140   1st Qu.:0.6527   2: 441
            Median :0.3300   Median :0.2600   Median :0.7795   3: 126
            Mean   :0.3308   Mean   :0.2553   Mean   :0.7378   4:  77
            3rd Qu.:0.3700   3rd Qu.:0.2988   3rd Qu.:0.8685   5: 182
            Max.   :3.0000   Max.   :3.0000   Max.   :1.9500   6: 201
 mhist    phist          unemp         selfemp    insurance  condomin
 1: 747   no :2205   Min.   : 1.800   no :2103   no :2332   no :1694
 2:1571   yes: 175   1st Qu.: 3.100   yes: 277   yes:  48   yes: 686
 3:  41              Median : 3.200
 4:  21              Mean   : 3.774
                     3rd Qu.: 3.900
                     Max.   :10.600
  afam       single     hschool
 no :2041   no :1444   no :  39
 yes: 339   yes: 936   yes:2341
```

5.2 Binary Dependent Variable and Linear Probability Model

The variable we are interested in modelling is `deny`, an indicator for whether an applicant's mortgage application has been accepted (`deny = no`) or denied (`deny = yes`).

A regressor that ought to have power in explaining whether a mortgage application has been denied is `pirat`, the size of the anticipated total monthly loan payments relative to the the applicant's income. It is straightforward to translate this into the simple regression model:

$$deny = \beta_0 + \beta_1 \, P/I \, ratio + u \tag{4.1}$$

We estimate this model just as any other linear regression model using `lm()`. Before we do so, the **variable `deny` must be converted to a numeric variable** using `as.numeric()`, as the function `lm()` does not accept the dependent variable to be of class `factor`.

Note that `as.numeric(HMDA$deny)` will turn `deny = no` into `deny = 1` and `deny = yes` into `deny = 2`. Instead of these, we would like to obtain the values 0 and 1, what we can achieve using `as.numeric(HMDA$deny)-1`.

```
# convert 'deny' to numeric
HMDA$deny <- as.numeric(HMDA$deny) - 1

# estimate a simple linear probabilty model
denymod1 <- lm(deny ~ pirat, data = HMDA)
denymod1
```

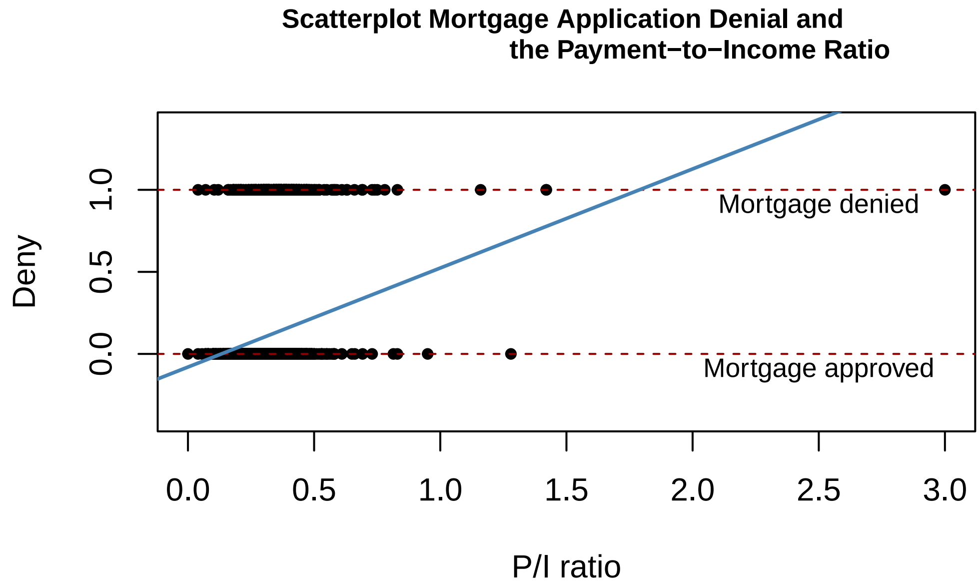
```
Call:
lm(formula = deny ~ pirat, data = HMDA)

Coefficients:
(Intercept)        pirat
   -0.07991      0.60353
```

Next, we plot the data and the regression line

```
# plot the data
plot(x = HMDA$pirat, y = HMDA$deny,
     main = "Scatterplot Mortgage Application Denial and
                               the Payment-to-Income Ratio",
     xlab = "P/I ratio", ylab = "Deny",
     pch = 20, ylim = c(-0.4, 1.4), cex.main = 0.8)

# add horizontal dashed lines and text
abline(h = 1, lty = 2, col = "darkred")
abline(h = 0, lty = 2, col = "darkred")
text(2.5, 0.9, cex = 0.8, "Mortgage denied")
text(2.5, -0.1, cex= 0.8, "Mortgage approved")

# add the estimated regression line
abline(denymod1, lwd = 1.8, col = "steelblue")
```

According to the estimated model, a payment-to-income ratio of 1 is associated with an expected probability of mortgage application denial of roughly 50%.

The model indicates that there is a positive relation between the payment-to-income ratio and the probability of a denied mortgage application. This suggests that individuals with a high ratio of loan payments to income are associated with a higher chance of being rejected.



We may use `coeftest()` to obtain robust standard errors for both coefficient estimates.

```
# print robust coefficient summary
coeftest(denymod1, vcov. = vcovHC, type = "HC1")
```

```
t test of coefficients:

             Estimate Std. Error t value  Pr(>|t|)
(Intercept) -0.079910   0.031967 -2.4998   0.01249 *
pirat        0.603535   0.098483  6.1283 1.036e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated regression line is

$$\widehat{deny} = \underset{(0.032)}{-0.080} + \underset{(0.098)}{0.604}\, P/I\, ratio \tag{4.2}$$

The coefficient on $P/I\, ratio$ is statistically different from 0 at the 0.1% level. Its estimate can be interpreted as follows: a 1 percentage point increase in $P/I\, ratio$ is associated with an average increase in the probability of a loan denial by $0.604 \cdot 0.01 = 0.00604 \approx 0.6$ percentage points.

5.3 Is there Racial Discrimination in the Mortgage Market?

We will now augment the simple model (4.2) by adding an additional regressor: `black`, which equals 1 if the applicant is African American and equals 0 otherwise.

Such a specification is the baseline for investigating if there is racial discrimination in the mortgage market: if being black has a significant (positive) influence on the probability of a loan denial when we control for factors that allow for an objective assessment of an applicant's creditworthiness, this could be an indicator for discrimination.

In this data set, the variable `afam` indicates whether the applicant is an African American or not. We will first rename this variable to `black` for consistency and then we will estimate the model including this new regressor.

```
# rename the variable 'afam'
colnames(HMDA)[colnames(HMDA) == "afam"] <- "black"

# estimate the model
denymod2 <- lm(deny ~ pirat + black, data = HMDA)
coeftest(denymod2, vcov. = vcovHC)
```

```
t test of coefficients:

             Estimate Std. Error t value  Pr(>|t|)
(Intercept) -0.090514   0.033430 -2.7076  0.006826 **
pirat        0.559195   0.103671  5.3939 7.575e-08 ***
blackyes     0.177428   0.025055  7.0815 1.871e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated regression function is

$$\widehat{deny} = \underset{(0.033)}{-0.091} + \underset{(0.104)}{0.559}\, P/I\ ratio + \underset{(0.025)}{0.177}\, black \tag{4.3}$$

The coefficient on `black` is positive and significantly different from zero at the 0.1% level. The interpretation is that, holding constant the $P/I\ ratio$, being black is associated with an average increase in the probability of a mortgage application denial by 17.7 percentage points.

This finding could be associated with racial discrimination. However, it might be distorted by omitted variable bias so discrimination could be a premature conclusion.

5.4 Probit and Logit Regression

The linear probability model has a major flaw: it assumes the conditional probability function to be linear. This does not restrict $P(Y = 1|X_1, \dots, X_k)$ to lie between 0 and 1.

We can easily observe this in our previous plot for model (4.2): for $P/I\ ratio = 1.75$, the model predicts the probability of a mortgage application denial to be bigger than 1. For applications with $P/I\ ratio$ close to 0, the predicted probability of denial is even negative, so that the model has no meaningful interpretation here.

From this we can infer the **need for a nonlinear function** to model the conditional probability function of a binary dependent variable. Commonly used methods are Probit and Logit regression.

5.4.1 Probit Regression

Assume that Y is a binary variable. The model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + u$$

with

$$P(Y = 1|X_1, X_2 \dots, X_k) = \Phi(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k)$$

is the population Probit model, with multiple regressors $X_1, X_2 \dots, X_k$ and $\Phi(\cdot)$ being the cumulative distribution function (CDF) of a **standard normal distribution**.

The **predicted probability** that $Y = 1$ given $X_1, X_2 \dots, X_k$ can be calculated in two steps:

1. Compute $z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k$

2. Look up $\Phi(z)$ by calling `pnorm()`

β_j is the effect on z of a one unit change in regressor X_j, holding constant all other $k-1$ regressors.

The effect on the predicted probability of a **change in a regressor** can be computed also in two steps:

1. Compute the predicted probability of $Y = 1$ for two cases:

- Case 1: Using the original values of the regressors $(X_1, X_2, \ldots, X_k)$.
- Case 2: Using the modified value of X_1 $(X_1 + \Delta X_1)$ while keeping other regressors constant.

2. The difference between the predicted probabilities in Case 1 and Case 2 gives the expected change in the predicted probability of $Y = 1$ associated with the change in X_1.

$$\Delta\hat{Y} = \hat{P}(Y = 1|X_1 + \Delta X_1, X_2, \ldots, X_k) - \hat{P}(Y = 1|X_1, X_2, \ldots, X_k)$$

Where $\hat{P}(Y = 1|X_1, X_2, \ldots, X_k)$ represents the predicted probability of $Y = 1$ based on the estimated probit model.

In R, Probit models can be estimated using the function `glm()` from the package `stats`. Using the argument `family` we specify that we want to use a Probit link function.

We can now estimate a simple Probit model of the probability of a mortgage denial. Since we have a binary dependent variable, we need to set `family = binomial` and for this case, we will set `link = "probit"`.

```
# estimate the simple probit model
denyprobit <- glm(deny ~ pirat, family = binomial(link = "probit"), data = HMDA)

coeftest(denyprobit, vcov. = vcovHC, type = "HC1")
```

```
z test of coefficients:

            Estimate Std. Error  z value  Pr(>|z|)    
(Intercept) -2.19415    0.18901 -11.6087 < 2.2e-16 ***
pirat        2.96787    0.53698   5.5269 3.259e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Just as in the linear probability model, we find that the relation between the probability of denial and the payments-to-income ratio is positive and that the corresponding coefficient is highly significant.

The estimated model is

$$P(deny|\widehat{P/I\, ratio}) = \Phi(\underset{(0.19)}{-2.19} + \underset{(0.54)}{2.97}\; P/I\, ratio) \tag{4.4}$$

We can plot this probit model with the following code chunk

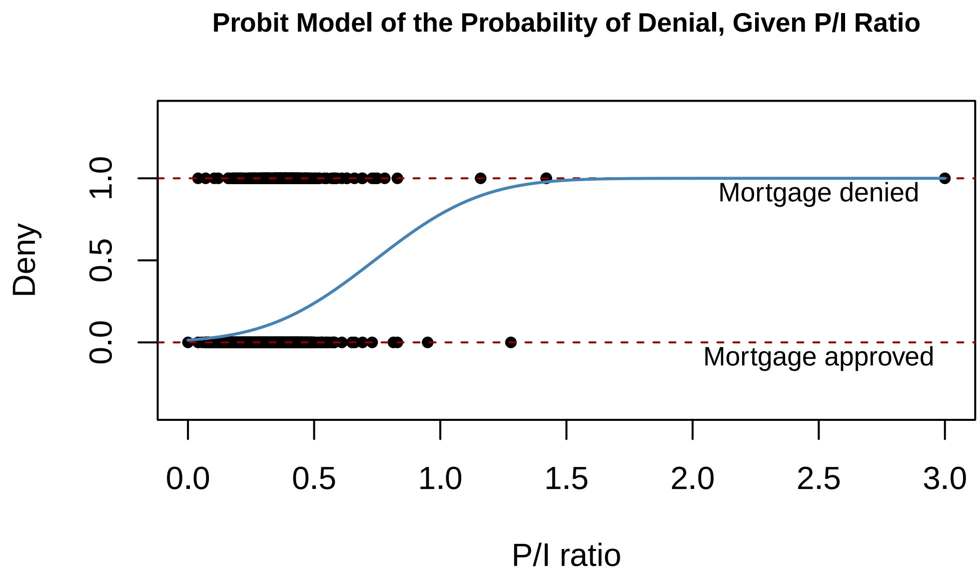
```
# plot data
plot(x = HMDA$pirat, y = HMDA$deny,
     main = "Probit Model of the Probability of Denial, Given P/I Ratio",
     xlab = "P/I ratio", ylab = "Deny",
     pch = 20, ylim = c(-0.4, 1.4), cex.main = 0.85)

# add horizontal dashed lines and text
abline(h = 1, lty = 2, col = "darkred")
abline(h = 0, lty = 2, col = "darkred")
text(2.5, 0.9, cex = 0.8, "Mortgage denied")
text(2.5, -0.1, cex= 0.8, "Mortgage approved")

# add estimated regression line
x <- seq(0, 3, 0.01)
y <- predict(denyprobit, list(pirat = x), type = "response")
lines(x, y, lwd = 1.5, col = "steelblue")
```

As observed here, the estimated regression function has a "stretched S-shape". This is typical for the cumulative distribution function of a continuous random variable with symmetric probability density function, like that of a normal random variable.

The function is clearly nonlinear and flattens out for large and small values of $P/I\, ratio$. The functional form thus ensures that the predicted conditional probabilities of a denial lie between 0 and 1.

How would the denial probability change if we increase the $P/I\, ratio$ from 0.3 to 0.4? We can use `predict()` and `diff()` functions to compute the predicted change:

```
# 1. compute predictions for P/I ratio = 0.3, 0.4
predictions <- predict(denyprobit, newdata = data.frame("pirat" = c(0.3, 0.4)),
                       type = "response")
predictions
```




```
         1          2 
0.09615344 0.15696777 
```

```
# 2. Compute difference in probabilities
diff(predictions)
```

```
         2 
0.06081433 
```

According to our model, an increase in the $P/I\ ratio$ from 0.3 to 0.4 leads to an average increase in the probability of denial of 6.1 percentage points.

Let's now include the variable `black` in our Probit model to further estimate the effect of race on the probability of a mortgage application denial.

```
# estimate the augmented probit model
denyprobit2 <- glm(deny ~ pirat + black, family = binomial(link = "probit"),
                   data = HMDA)

coeftest(denyprobit2, vcov. = vcovHC, type = "HC1")
```

```
z test of coefficients:
```

```
             Estimate Std. Error  z value  Pr(>|z|)
(Intercept) -2.258787   0.176608 -12.7898 < 2.2e-16 ***
pirat        2.741779   0.497673   5.5092 3.605e-08 ***
blackyes     0.708155   0.083091   8.5227 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The coefficients on $P/I, ratio$ and $black$ appear to be positive and highly significant.

While their interpretation can be sensitive and challenging, this probit model indicates two key findings: first, black applicants, on average, have a higher probability of denial than white applicants, holding the payments-to-income ratio constant; second, applicants with a higher payments-to-income ratio, regardless of their race, face on average a higher risk of rejection.

The estimated model equation is

$$P(deny|\widehat{P/I\,ratio}, black) = \Phi(\underset{(0.18)}{-2.26} + \underset{(0.50)}{2.74}\,P/I\,ratio + \underset{(0.08)}{0.71}\,black) \tag{4.5}$$

How big is the estimated difference in denial probabilities between two hypothetical applicants with the same payments-to-income ratio? Just like before, we can compute the difference in probabilities to answer this question according to our estimated model:

```r
# 1. compute predictions with a constant P/I ratio of 0.3
predictions <- predict(denyprobit2,
                       newdata = data.frame("black" = c("no", "yes"),
                                            "pirat" = c(0.3, 0.3)),
                       type = "response")
predictions
```

```
         1          2
0.07546516 0.23327685
```

```r
# 2. compute difference in probabilities
diff(predictions)
```

```
        2
0.1578117
```

The result indicates that the estimated difference in denial probabilities between a "black" and a "non-black" applicant, both with a payment-to-income ratio of 0.3, is on average 15.8 percentage points higher for the "black" applicant.

5.4.2 Logit regression

The population Logit regression function is

$$
\begin{aligned}
P(Y = 1|X_1, X_2, \dots, X_k) &= F(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k) \\
&= \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k)}}
\end{aligned}
$$

The idea is similar to the Probit regression except that here, the probability of the dependent variable Y being 1 given a set of independent variables $X_1, X_2, \dots, X_k$ is modeled using the cumulative distribution function (CDF) of a **standard logistically distributed** random variable:

$$F(x) = \frac{1}{1 + e^{-x}}$$

As for Probit regression, there is no simple interpretation of the model coefficients and it is best to consider predicted probabilities or differences in predicted probabilities.

The estimation of the Logit regression model in R is again a straightforward process. However, for this specific case, we should specify `link = "logit"`:

```r
denylogit <- glm(deny ~ pirat, family = binomial(link = "logit"), data = HMDA)

coeftest(denylogit, vcov. = vcovHC, type = "HC1")
```

```
z test of coefficients:

             Estimate Std. Error  z value  Pr(>|z|)
(Intercept) -4.02843    0.35898 -11.2218 < 2.2e-16 ***
pirat        5.88450    1.00015   5.8836 4.014e-09 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated model is

$$\widehat{P(deny|P/I\,ratio)} = F(\underset{(0.36)}{-4.03} + \underset{(1.00)}{5.88}\,P/I\,ratio) \tag{4.6}$$

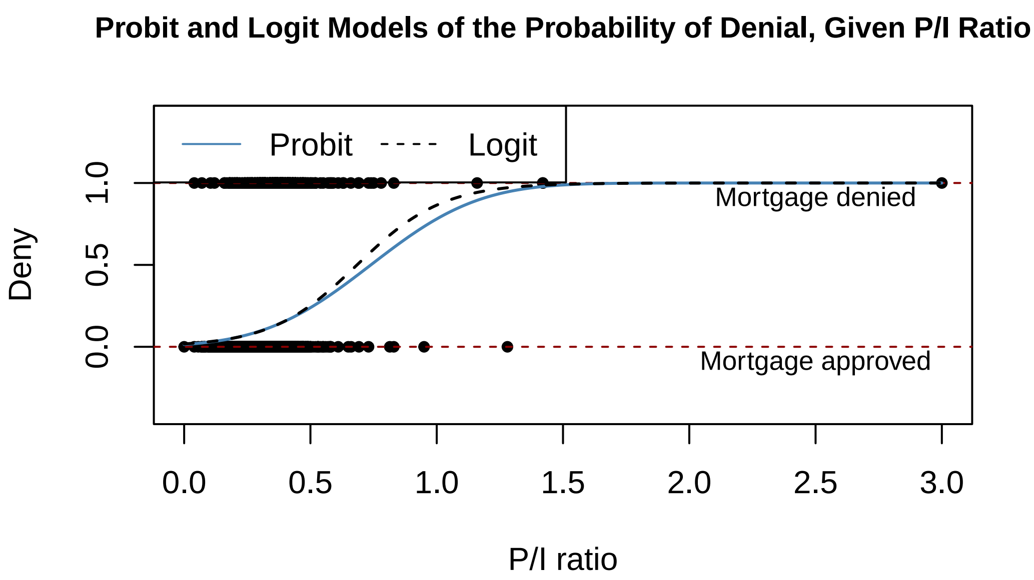
We can now plot both estimated models to visualize and compare results:

```r
# plot data
plot(x = HMDA$pirat, y = HMDA$deny,
     main = "Probit and Logit Models of the Probability of Denial, Given P/I Ratio",
     xlab = "P/I ratio", ylab = "Deny", pch = 20, ylim = c(-0.4, 1.4), cex.main = 0.9)

# add horizontal dashed lines and text
abline(h = 1, lty = 2, col = "darkred")
abline(h = 0, lty = 2, col = "darkred")
text(2.5, 0.9, cex = 0.8, "Mortgage denied")
text(2.5, -0.1, cex= 0.8, "Mortgage approved")

# add estimated regression line of Probit and Logit models
x <- seq(0, 3, 0.01)
y_probit <- predict(denyprobit, list(pirat = x), type = "response")
y_logit <- predict(denylogit, list(pirat = x), type = "response")
lines(x, y_probit, lwd = 1.5, col = "steelblue")
lines(x, y_logit, lwd = 1.5, col = "black", lty = 2)

# add a legend
legend("topleft",horiz = TRUE, legend = c("Probit", "Logit"),
       col = c("steelblue", "black"), lty = c(1, 2))
```

Both models produce very similar estimates of the probability of a mortgage application being denied based on the applicants' payment-to-income ratio.

Now we may also extend the Logit model by including the variable `black`

```r
# estimate a Logit regression with multiple regressors
denylogit2 <- glm(deny ~ pirat + black, family = binomial(link = "logit"),
                  data = HMDA)

coeftest(denylogit2, vcov. = vcovHC, type = "HC1")
```




```
z test of coefficients:

             Estimate Std. Error  z value  Pr(>|z|)    
(Intercept) -4.12556    0.34597 -11.9245 < 2.2e-16 ***
pirat        5.37036    0.96376   5.5723 2.514e-08 ***
blackyes     1.27278    0.14616   8.7081 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We obtain

$$P(deny|\widehat{P/I\, ratio}, black) = F(\underset{(0.35)}{-4.13} + \underset{(0.96)}{5.37}\, P/I\, ratio + \underset{(0.15)}{1.27}\, black) \tag{4.7}$$

As for the Probit model (4.6) all model coefficients are highly significant and we obtain positive estimates for the coefficients on `P/I ratio` and `black`.

For comparison we compute the predicted probability of denial for two hypothetical applicants that differ in race and have a `P/I ratio` of 0.3.

```
# 1. compute predictions for P/I ratio = 0.3
predictions <- predict(denylogit2,
                       newdata = data.frame("black" = c("no", "yes"),
                                            "pirat" = c(0.3, 0.3)),
                       type = "response")
```

```r
predictions
```

```
         1          2 
0.07485143 0.22414592 
```

```r
# 2. Compute difference in probabilities
diff(predictions)
```

```
        2 
0.1492945 
```

We find that, according to our model, white applicants with a payment-to-income of 0.3 face on average a denial probability of only 7.5%, while African Americans with the same payment-to-income are rejected on average with a probability of 22.4%, which is 14.9 percentage points higher.

5.5 Comparison of the models

The Probit and the Logit models deliver only approximations to the unknown population regression function $E(Y|X)$. It is not obvious how to decide which model to use in practice.

The **linear probability model** has the clear drawback of not being able to capture the nonlinear nature of the population regression function and it may predict probabilities to lie outside the interval $[0, 1]$.

Probit and **Logit** models are harder to interpret but they capture the nonlinearities better than the linear approach: both models produce predictions of probabilities that lie inside the interval $[0, 1]$. Predictions of all three models are often close to each other.

The best choice usually depends on the specific characteristics of the data, the theory behind the model relative to the case being studied, and practical considerations like interpretability and the preferences of the audience for the analysis.

It is often suggested to use the method that is easiest to use in the statistical software of choice. As we have seen, it is equally easy to estimate Probit and Logit model using R. The choice between them might come down to other considerations such as the specific distributional assumptions behind each model (Logit assumes a logistic distribution of the error terms, while

Probit assumes a normal distribution), the context of the analysis, or the preferences of the analyst. There is therefore no general recommendation for which method to use.

5.6 Controlling for applicant characteristics & financial variables

Models (11.6) and (11.7) indicate that denial rates are higher for African American applicants holding constant the payment-to-income ratio. Both results could be subject to omitted variable bias.

In order to obtain a more trustworthy estimate of the effect of being black on the probability of a mortgage application denial we estimate a linear probability model as well as several Logit and Probit models, but this time we control for financial variables and additional applicant characteristics which are likely to influence the probability of denial and differ between black and white applicants:

- `hirat`: inhouse expense-to-total-income ratio.
- `lvrat`: loan-to-value ratio
- `chist`: consumer credit score
- `mhist`: mortgage credit score
- `phist`: public bad credit record
- `insurance`: denied mortgage insurance (factor)
- `selfemp`: self-employed (factor)
- `single`: single (factor)
- `hschool`: high school diploma (factor)
- `unemp`: unemployment rate
- `condomin`: condominium (factor)

For more on variables contained in the HMDA data set use R's `help()` function.

Sample averages can be easily reproduced using the functions `mean()` (as usual for numeric variables) and `prop.table()` (for factor variables). For example:

```
# inhouse expense-to-total-income ratio
mean(HMDA$hirat)
```

```
[1] 0.2553461
```

```
# self-employed
prop.table(table(HMDA$selfemp))
```

```
       no       yes
0.8836134 0.1163866
```

Before estimating the models we transform the loan-to-value ratio (`lvrat`) into a factor variable, where

$$lvrat = \begin{cases} \text{low} & \text{if } lvrat < 0.8 \\ \text{medium} & \text{if } 0.8 \leq lvrat \leq 0.95 \\ \text{high} & \text{if } lvrat > 0.95 \end{cases}$$

and convert both credit scores to numeric variables.

```
# define low, medium and high loan-to-value ratio
HMDA$lvrat <- factor(
  ifelse(HMDA$lvrat < 0.8, "low",
  ifelse(HMDA$lvrat >= 0.8 & HMDA$lvrat <= 0.95, "medium", "high")),
  levels = c("low", "medium", "high"))

# convert credit scores to numeric
HMDA$mhist <- as.numeric(HMDA$mhist)
HMDA$chist <- as.numeric(HMDA$chist)
```

Next, we estimate different models for denial probability

```
# estimate 6 models for the denial probability
lpm <- lm(deny ~ black + pirat + hirat + lvrat + chist + mhist + phist
               + insurance + selfemp, data = HMDA)

logit <- glm(deny ~ black + pirat + hirat + lvrat + chist + mhist + phist
                  + insurance + selfemp,
                  family = binomial(link = "logit"),
                  data = HMDA)

probit1 <- glm(deny ~ black + pirat + hirat + lvrat + chist + mhist + phist
                    + insurance + selfemp,
                    family = binomial(link = "probit"),
                    data = HMDA)

probit2 <- glm(deny ~ black + pirat + hirat + lvrat + chist + mhist + phist
                    + insurance + selfemp + single + hschool + unemp,
```

```
                     family = binomial(link = "probit"),
                     data = HMDA)

probit3 <- glm(deny ~ black + pirat + hirat + lvrat + chist + mhist
                     + phist + insurance + selfemp + single + hschool + unemp
                     +condomin + I(mhist==3) + I(mhist==4) + I(chist==3)
                     + I(chist==4) + I(chist==5)+ I(chist==6),
                     family = binomial(link = "probit"), data = HMDA)

probit4 <- glm(deny ~ black * (pirat + hirat) + lvrat + chist + mhist + phist
                     + insurance + selfemp + single + hschool + unemp,
                     family = binomial(link = "probit"), data = HMDA)
```

Then we store heteroskedasticity-robust standard errors of the coefficient estimators in a list which is then used as the argument `se` in `stargazer()`

```
rob_se <- list(sqrt(diag(vcovHC(lpm, type = "HC1"))),
               sqrt(diag(vcovHC(logit, type = "HC1"))),
               sqrt(diag(vcovHC(probit1, type = "HC1"))),
               sqrt(diag(vcovHC(probit2, type = "HC1"))),
               sqrt(diag(vcovHC(probit3, type = "HC1"))),
               sqrt(diag(vcovHC(probit4, type = "HC1"))))

stargazer(lpm, logit, probit1, probit2, probit3, probit4,
          se = rob_se,
          type="html",
          omit.stat = "f", df=FALSE)
```

```
<table style="text-align:center"><tr><td colspan="7" style="border-bottom: 1px solid black"><
<tr><td></td><td colspan="6" style="border-bottom: 1px solid black"></td></tr>
<tr><td style="text-align:left"></td><td colspan="6">deny</td></tr>
<tr><td style="text-align:left"></td><td><em>OLS</em></td><td><em>logistic</em></td><td colsp
<tr><td style="text-align:left"></td><td>(1)</td><td>(2)</td><td>(3)</td><td>(4)</td><td>(5)<
<tr><td colspan="7" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-alig
<tr><td style="text-align:left"></td><td>(0.023)</td><td>(0.183)</td><td>(0.099)</td><td>(0.1
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td></
<tr><td style="text-align:left">pirat</td><td>0.449<sup>***</sup></td><td>4.764<sup>***</sup>
<tr><td style="text-align:left"></td><td>(0.114)</td><td>(1.332)</td><td>(0.673)</td><td>(0.6
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td></
<tr><td style="text-align:left">hirat</td><td>-0.048</td><td>-0.109</td><td>-0.185</td><td>-0
```

```
<tr><td style="text-align:left"></td><td>(0.110)</td><td>(1.298)</td><td>(0.689)</td><td>(0.6
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td></
<tr><td style="text-align:left">lvratmedium</td><td>0.031<sup>**</sup></td><td>0.464<sup>***
<tr><td style="text-align:left"></td><td>(0.013)</td><td>(0.160)</td><td>(0.082)</td><td>(0.0
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td></
<tr><td style="text-align:left">lvrathigh</td><td>0.189<sup>***</sup></td><td>1.495<sup>***</
<tr><td style="text-align:left"></td><td>(0.050)</td><td>(0.325)</td><td>(0.183)</td><td>(0.1
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td></
<tr><td style="text-align:left">chist</td><td>0.031<sup>***</sup></td><td>0.290<sup>***</sup>
<tr><td style="text-align:left"></td><td>(0.005)</td><td>(0.039)</td><td>(0.021)</td><td>(0.0
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td></
<tr><td style="text-align:left">mhist</td><td>0.021<sup>*</sup></td><td>0.279<sup>**</sup></t
<tr><td style="text-align:left"></td><td>(0.011)</td><td>(0.138)</td><td>(0.073)</td><td>(0.0
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td></
<tr><td style="text-align:left">phistyes</td><td>0.197<sup>***</sup></td><td>1.226<sup>***</s
<tr><td style="text-align:left"></td><td>(0.035)</td><td>(0.203)</td><td>(0.114)</td><td>(0.1
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td></
<tr><td style="text-align:left">insuranceyes</td><td>0.702<sup>***</sup></td><td>4.548<sup>**
<tr><td style="text-align:left"></td><td>(0.045)</td><td>(0.576)</td><td>(0.305)</td><td>(0.2
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td></
<tr><td style="text-align:left">selfempyes</td><td>0.060<sup>***</sup></td><td>0.666<sup>***
<tr><td style="text-align:left"></td><td>(0.021)</td><td>(0.214)</td><td>(0.113)</td><td>(0.1
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td></
<tr><td style="text-align:left">singleyes</td><td></td><td></td><td></td><td>0.229<sup>***</s
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(0.080)</td><td>(0.086)</
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td></
<tr><td style="text-align:left">hschoolyes</td><td></td><td></td><td></td><td>-0.613<sup>***
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(0.229)</td><td>(0.237)</
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td></
<tr><td style="text-align:left">unemp</td><td></td><td></td><td></td><td>0.030<sup>*</sup></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(0.018)</td><td>(0.018)</
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td></
<tr><td style="text-align:left">condominyes</td><td></td><td></td><td></td><td></td><td>-0.05
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td>(0.096)</td><td>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td></
<tr><td style="text-align:left">I(mhist == 3)</td><td></td><td></td><td></td><td></td><td>-0
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td>(0.301)</td><td>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td></
<tr><td style="text-align:left">I(mhist == 4)</td><td></td><td></td><td></td><td></td><td>-0
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td>(0.427)</td><td>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td></
<tr><td style="text-align:left">I(chist == 3)</td><td></td><td></td><td></td><td></td><td>-0
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td>(0.248)</td><td>
```

```
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td style="text-align:left">I(chist == 4)</td><td></td><td></td><td></td><td></td><td>-0
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td>(0.338)</td><td>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td style="text-align:left">I(chist == 5)</td><td></td><td></td><td></td><td></td><td>-0
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td>(0.412)</td><td>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td style="text-align:left">I(chist == 6)</td><td></td><td></td><td></td><td></td><td>-0
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td>(0.515)</td><td>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td style="text-align:left">blackyes:pirat</td><td></td><td></td><td></td><td></td><td><
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td>(1.550
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td style="text-align:left">blackyes:hirat</td><td></td><td></td><td></td><td></td><td><
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td>(1.709
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td style="text-align:left">Constant</td><td>-0.183<sup>***</sup></td><td>-5.707<sup>***
<tr><td style="text-align:left"></td><td>(0.028)</td><td>(0.484)</td><td>(0.250)</td><td>(0.3
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td><td></td><
<tr><td colspan="7" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-ali
<tr><td style="text-align:left">R<sup>2</sup></td><td>0.266</td><td></td><td></td><td></td><
<tr><td style="text-align:left">Adjusted R<sup>2</sup></td><td>0.263</td><td></td><td></td><
<tr><td style="text-align:left">Log Likelihood</td><td></td><td>-635.637</td><td>-636.847</t
<tr><td style="text-align:left">Akaike Inf. Crit.</td><td></td><td>1,293.273</td><td>1,295.6
<tr><td style="text-align:left">Residual Std. Error</td><td>0.279</td><td></td><td></td><td>
<tr><td colspan="7" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-ali
</table>
```

Models (1), (2) and (3) are baseline specifications that include several financial control variables. They differ only in the way they model the denial probability. Model (1) is a linear probability model, model (2) is a Logit regression and model (3) uses the Probit approach.

In the **linear model (1)**, the coefficients have **direct interpretation**. For example:

- An increase in the consumer credit score by 1 unit is estimated to increase the probability of a loan denial on average by 3.1 percentage points.
- Having a high loan-to-value ratio is detriment for credit approval: the coefficient for a loan-to-value ratio higher than 0.95 is 0.189 so clients with this property are estimated to face an almost 19 percentage points larger risk of denial on average than those with a low loan-to-value ratio, ceteris paribus.

- The estimated coefficient on the race dummy is 0.084, which indicates the denial probability for African Americans is estimated to be on average 8.4 percentage points larger than for white applicants with the same characteristics except for race.

Apart from the inhouse expense-to-total-income ratio, all coefficients are significant in the linear probability model.

Models (2) and (3) provide similar evidence of racial discrimination in the U.S. mortgage market. All coefficients except for the housing expense-to-income ratio (which is not significantly different from zero) and the mortgage credit score (which is statistically significant at the 5% level) are significant at the 1% level.

As discussed above, the **nonlinearity makes the interpretation** of the coefficient estimates **more difficult** than for model (1).

In order to make a statement about the effect of being black, we need to compute the estimated denial probability for two individuals that differ only in race. For the comparison we consider two individuals that share mean values for all numeric regressors.

For the qualitative variables we assign the property that is most representative for the data at hand. For example, consider self-employment: we have seen that about 88% of all individuals in the sample are not self-employed such that we set `selfemp = no`.

Using this approach, the estimate for the effect on the denial probability of being African American according to the Logit model (2) would be 4 percentage points. The next code chunk shows how to apply this approach for models (1) to (6) using R.

```
# compute regressor values for an average black person
new <- data.frame(
  "pirat" = mean(HMDA$pirat),
  "hirat" = mean(HMDA$hirat),
  "lvrat" = "low",
  "chist" = mean(HMDA$chist),
  "mhist" = mean(HMDA$mhist),
  "phist" = "no",
  "insurance" = "no",
  "selfemp" = "no",
  "black" = c("no", "yes"),
  "single" = "no",
  "hschool" = "yes",
  "unemp" = mean(HMDA$unemp),
  "condomin" = "no")

# difference predicted by the LPM (1)
```

```r
predictions <- predict(lpm, newdata = new)
diff(predictions)
```

```
         2
0.08369674
```

```r
# difference predicted by the logit model (2)
predictions <- predict(logit, newdata = new, type = "response")
diff(predictions)
```

```
         2
0.04042135
```

```r
# difference predicted by probit model (3)
predictions <- predict(probit1, newdata = new, type = "response")
diff(predictions)
```

```
         2
0.05049716
```

```r
# difference predicted by probit model (4)
predictions <- predict(probit2, newdata = new, type = "response")
diff(predictions)
```

```
         2
0.03978918
```

```r
# difference predicted by probit model (5)
predictions <- predict(probit3, newdata = new, type = "response")
diff(predictions)
```

```
         2
0.04972468
```

```r
# difference predicted by probit model (6)
predictions <- predict(probit4, newdata = new, type = "response")
diff(predictions)
```

```
         2
0.03955893
```

The estimates of the impact on the denial probability of being black are similar for models (2) and (3). It is interesting that the magnitude of the estimated effects is much smaller than for Probit and Logit models that do not control for financial characteristics (see models 4.5 and 4.7). This indicates that these simple models produced biased estimates due to omitted variables.

Regressions (4) to (6) include different applicant characteristics and credit rating indicator variables, as well as interactions. However, most of the corresponding coefficients are not significant and the estimates of the coefficient on black obtained for these models, as well as the estimated difference in denial probabilities, do not differ much from those obtained for models (2) and (3).

An interesting question related to racial discrimination can be investigated using the Probit model (6) where the interactions `blackyes:pirat` and `blackyes:hirat` are added to model (4).

If the coefficient on `blackyes:pirat` was significantly different from zero, the effect of the payment-to-income ratio on the denial probability would be different for black and white applicants.

Similarly, a non-zero coefficient on `blackyes:hirat` would indicate that loan officers weight the risk of bankruptcy associated with a high loan-to-value ratio differently for black and white mortgage applicants. We can **test** whether these coefficients are **jointly significant at the** 5% **level** using an F-Test.

```r
linearHypothesis(probit4,
                 test = "F",
                 c("blackyes:pirat=0", "blackyes:hirat=0"),
                 vcov = vcovHC, type = "HC1")
```

```
Linear hypothesis test

Hypothesis:
blackyes:pirat = 0
blackyes:hirat = 0

Model 1: restricted model
Model 2: deny ~ black * (pirat + hirat) + lvrat + chist + mhist + phist +
    insurance + selfemp + single + hschool + unemp

Note: Coefficient covariance matrix supplied.
```

```
  Res.Df Df      F Pr(>F)
1   2366
2   2364  2 0.2473 0.7809
```

Since p-value ≈ 0.78 for this test, the null cannot be rejected. There is not enough evidence to conclude that there is a significant interaction effect between being black and the variables `pirat` and `hirat` when considering the denial outcome.

Nonetheless, when we test whether the coefficients for the main effect of `blackyes` and the interaction terms `blackyes:pirat` and `blackyes:hirat` are jointly equal to zero at the 5% level, we obtain:

```
linearHypothesis(probit4, test = "F",
                 c("blackyes=0", "blackyes:pirat=0", "blackyes:hirat=0"),
                 vcov = vcovHC, type = "HC1")
```

```
Linear hypothesis test

Hypothesis:
blackyes = 0
blackyes:pirat = 0
blackyes:hirat = 0

Model 1: restricted model
Model 2: deny ~ black * (pirat + hirat) + lvrat + chist + mhist + phist +
    insurance + selfemp + single + hschool + unemp

Note: Coefficient covariance matrix supplied.

  Res.Df Df      F   Pr(>F)
1   2367
2   2364  3 4.7774 0.002534 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

With p-value ≈ 0.003 we can reject the hypothesis that there is no racial discrimination in the model. There is significant evidence to suggest that at least one of the coefficients for the main effect of `blackyes` or the interaction terms involving `blackyes` is not equal to zero.

This **suggests the presence of racial discrimination in the model**, as the inclusion of `blackyes` in addition to the interaction terms leads to a significant difference in the model fit.

5.7 Summary

Models (1) to (6) provide evidence that there is an effect of being African American on the probability of mortgage application denial.

In specifications (2) to (5), the effect is estimated to be positive (ranging from 4 to 5 percentage points) and statistically significant at the 1% level.

While the linear probability model (1) seems to slightly overestimate this positive effect at 8 percentage points, it still can be used as an approximation to an intrinsically nonlinear relationship.

Probit model (6) delved deeper, revealing the presence of racial discrimination through interaction effects between being African American and other variables.

6 Empirical Applications of Instrumental Variables Regression

In this chapter we will apply the concepts of Instrumental Variables Regression, which are those regression models that aim to solve the problem arising when the error term u is correlated with the regressor of interest, and so that the corresponding coefficient is estimated inconsistently.

We have previously addressed the issue of omitted variables bias by adding the omitted variables to the regression, trying to mitigate the risk of biased estimation of the causal effect of interest. However, if we don't have data on the omitted factors, multiple regression is not sufficient.

The same issue arises when causality runs both from X to Y and from Y to X, so that there is simultaneous causality bias. There will be again an estimation bias that cannot be corrected for by multiple regression.

Instrumental variables (IV) regression is a general solution to obtain a consistent estimator of the unknown causal coefficients when the regressor X is correlated with the error term u. In this chapter we focus on the IV regression tool called *two-stage least squares* (TSLS).

6.1 Data Set Description

We will use the data set `CigarettesSW` which comes with the package `AER` (Christian Kleiber and Zeileis 2008). It is a panel data set that contains observations on cigarette consumption and several economic indicators for all 48 continental federal states of the U.S. from 1985 to 1995.

```
# load the data set
library(AER)
data("CigarettesSW")
```

```
# get an overview
summary(CigarettesSW)
```

```
    state        year          cpi          population          packks
 AL     : 2   1985:48   Min.   :1.076   Min.   :  478447   Min.   : 49.27
 AR     : 2   1995:48   1st Qu.:1.076   1st Qu.: 1622606   1st Qu.: 92.45
 AZ     : 2             Median :1.300   Median : 3697472   Median :110.16
 CA     : 2             Mean   :1.300   Mean   : 5168866   Mean   :109.18
 CO     : 2             3rd Qu.:1.524   3rd Qu.: 5901500   3rd Qu.:123.52
 CT     : 2             Max.   :1.524   Max.   :31493524   Max.   :197.99
 (Other):84
     income               tax            price             taxs
 Min.   :  6887097   Min.   :18.00   Min.   : 84.97   Min.   : 21.27
 1st Qu.: 25520384   1st Qu.:31.00   1st Qu.:102.71   1st Qu.: 34.77
 Median : 61661644   Median :37.00   Median :137.72   Median : 41.05
 Mean   : 99878736   Mean   :42.68   Mean   :143.45   Mean   : 48.33
 3rd Qu.:127313964   3rd Qu.:50.88   3rd Qu.:176.15   3rd Qu.: 59.48
 Max.   :771470144   Max.   :99.00   Max.   :240.85   Max.   :112.63
```

Use `?CigarettesSW` for a detailed description of the variables.

6.2 Problem Description

The relation between commodity demand and prices is a fundamental and widely observed issue in economics. Health economics focuses on how individual health-related behaviors are influenced by healthcare systems and regulatory policies. Smoking serves as a prime example in public policy discussions due to its association with various illnesses and negative impacts on society.

Cigarette consumption could potentially be reduced by increasing taxes on cigarettes. The question is by *how much* taxes must be increased to reach a certain reduction in cigarette consumption.

Elasticity is commonly estimated and used by economists to answer this kind of questions. But an OLS regression of log quantity on log price cannot be used to estimate the price elasticity for the demand of cigarettes, since there is **simultaneous causality between demand and supply**.

In this case, the effect on demand quantity of a change in price can instead be estimated using **IV regression**.

6.3 The IV Estimator with a Single Regressor and a Single Instrument

Consider the simple regression model

$$Y_i = \beta_0 + \beta_1 X_i + u_i, \quad i = 1, \dots, n \tag{5.1}$$

where the error term u_i is correlated with the regressor X_i (X is endogenous) such that the OLS estimator is inconsistent for the true β_1 (the causal effect of X on Y). Instrumental variables estimation uses an additional, "instrumental" variable Z to isolate that part of X that is uncorrelated with u, to obtain a consistent estimator for β_1.

Z must satisfy two conditions to be a valid instrument:

1. Instrument relevance condition: X and its instrument Z *must* be correlated: $\rho_{Z_i,X_i} \neq 0$

2. Instrument exogeneity condition: The instrument Z *must not* be correlated with the error term u: $\rho_{Z_i,u_i} = 0$.

The Two-Stage Least Squares Estimator

As its name suggests, TSLS proceeds in two stages. In the first stage, the endogenous regressor X is decomposed into a problem-free component, uncorrelated with the error term, that is explained by the instrument Z, and a problematic component that may be correlated with the error u_i. The second stage uses the problem-free component to estimate β_1.

The **first stage** regression model is

$$X_i = \pi_0 + \pi_1 Z_i + \nu_i$$

where $\pi_0 + \pi_1 Z_i$ is the component of X_i explained by Z_i and ν_i is the problematic component that cannot be explained by Z_i and exhibits correlation with u_i.

With the OLS estimates $\hat{\pi_0}$ and $\hat{\pi_1}$ the predicted values $\widehat{X_i}, i = 1, \dots, n$ are obtained. If Z is a valid instrument, the predicted $\widehat{X}_i$ are problem-free so that in the second stage regression, the OLS regression of Y on $\widehat{X}$, $\widehat{X}$ is exogenous.

From the **second stage** regression we obtain the TSLS estimators $\hat{\beta}_0^{TSLS}$ and $\hat{\beta}_1^{TSLS}$. For a single instrument case the TSLS estimator of β_1 is:

$$\hat{\beta}_1^{TSLS} = \frac{s_{ZY}}{s_{ZX}} = \frac{\frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y})(Z_i - \bar{Z})}{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(Z_i - \bar{Z})}, \tag{5.2}$$

which is indeed the ratio of the sample covariance between Z and Y to the sample covariance between Z and X.

Assuming Z meets the requirements of a valid instrument, (5.2) is a **consistent estimator** for β_1 in (5.1). The Central Limit Theorem (CLT) suggests that as the sample size increases, the distribution of $\hat{\beta}_1^{TSLS}$ can be closely approximated by a normal distribution. Consequently, we can use t-statistics and confidence intervals, which can be calculated using certain functions in R.

For our problem, we are interested in estimating β_1 in

$$\log(Q_i^{\text{cigarettes}}) = \beta_0 + \beta_1 \log(P_i^{\text{cigarettes}}) + u_i \tag{5.3}$$

where $Q_i^{\text{cigarettes}}$ is the number of cigarette packs sold per capita (the demand), $P_i^{\text{cigarettes}}$ is the after-tax average real price per pack of cigarettes in state i and u_i represents other factors that affect the demand of cigarettes.

The instrumental variable we will use for instrumenting the endogenous regressor $\log(P_i^{\text{cigarettes}})$ is $SalesTax$, the portion of taxes on cigarettes arising from the general sales tax, measured in dollars per pack (in real dollars, deflated by the Consumer Price Index).

Before using TSLS, it is essential to ask whether the two conditions for instrument validity hold. First, the idea is that $SalesTax$ is a **relevant instrument**, considering a high sales tax increases the after-tax sales price.

Since the sales tax does not directly influence the sold quantity, but indirectly through the price, it is plausible that $SalesTax$ is **exogenous**. The credibility of this assumption will be further discussed later, but for now we keep it as a working hypothesis.

We first perform some transformations in order to obtain deflated cross section data for the year 1995, as we will consider data for the cross section of states in 1995 only. We also compute the sample correlation between the sales tax and price per pack.

```
# compute real per capita prices
CigarettesSW$rprice <- with(CigarettesSW, price / cpi)

#  compute the sales tax
CigarettesSW$salestax <- with(CigarettesSW, (taxs - tax) / cpi)

# check the correlation between sales tax and price
cor(CigarettesSW$salestax, CigarettesSW$price)
```

```
[1] 0.6141228
```

```
# generate a subset for the year 1995
c1995 <- subset(CigarettesSW, year == "1995")
```

The estimate of approximately 0.614 indicates that `salestax` and `price` exhibit positive correlation. However, a correlation analysis like this is not sufficient for checking whether the instrument is relevant. As mentioned, we will discuss later the issue of checking whether an instrument is relevant and exogenous.

The first stage regression is

$$\log(P_i^{\text{cigarettes}}) = \pi_0 + \pi_1 SalesTax_i + \nu_i$$

We can estimate this model in R using `lm()`.

```
# perform the first stage regression
cig_s1 <- lm(log(rprice) ~ salestax, data = c1995)
coeftest(cig_s1, vcov = vcovHC, type = "HC1")
```

```
t test of coefficients:

             Estimate Std. Error  t value  Pr(>|t|)    
(Intercept) 4.6165463  0.0289177 159.6444 < 2.2e-16 ***
salestax    0.0307289  0.0048354   6.3549 8.489e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The first stage regression yields:

$$\widehat{\log(P_i^{\text{cigarettes}})} = \underset{(0.029)}{4.617} + \underset{(0.005)}{0.031}\, SalesTax_i$$

indicating a positive relationship between the price of cigarettes and the sales tax.

How much of the observed variation in $\log(P_i^{\text{cigarettes}})$ is explained by the instrument $SalesTax$? This can be answered by looking at the regression's R^2

```
# inspect the R^2 of the first stage regression
summary(cig_s1)$r.squared
```

```
[1] 0.4709961
```

which states that about 47% of the variation in after tax prices is explained by the variation of the sales tax across states.

Next, we store $\widehat{\log(P_i^{\text{cigarettes}})}$, the fitted values obtained by the first stage regression `cig_s1`, in the variable `lcigp_pred`.

```
# store the predicted values
lcigp_pred <- cig_s1$fitted.values
```

Now in the second stage we run the regression of $\log(Q_i^{\text{cigarettes}})$ on $\widehat{\log(P_i^{\text{cigarettes}})}$ to obtain $\widehat{\beta}_0^{TSLS}$ and $\widehat{\beta}_1^{TSLS}$:

```
# perform the second stage regression
cig_s2 <- lm(log(c1995$packs) ~ lcigp_pred)
coeftest(cig_s2, vcov = vcovHC)
```

```
t test of coefficients:

            Estimate Std. Error t value  Pr(>|t|)
(Intercept)  9.71988    1.70304  5.7074 7.932e-07 ***
lcigp_pred  -1.08359    0.35563 -3.0469  0.003822 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Thus estimating the model (5.3) using TSLS yields

$$\widehat{\log(Q_i^{\text{cigarettes}})} = \underset{(1.70)}{9.72} - \underset{(0.36)}{1.08} \log(P_i^{\text{cigarettes}}) \tag{5.4}$$

This estimated regression function would be written using the regressor in the second stage, the predicted value $\widehat{\log(P_i^{\text{cigarettes}})}$. It is, however, conventional and more convenient simply to report the estimated regression function with $\log(P_i^{\text{cigarettes}})$ rather than $\widehat{\log(P_i^{\text{cigarettes}})}$.

Instead of manually performing TSLS in steps, we can use the function `ivreg()` from the `AER` package in R to compute the TSLS estimators in just one line of code. It is coded similarly as `lm()`. Instruments can be included in the standard regression formula by separating the model equation from the instruments using a vertical bar.

For our regression of interest the correct formula would be `log(packs) ~ log(rprice) | salestax`

```r
# perform TSLS using 'ivreg()'
cig_ivreg <- ivreg(log(packs) ~ log(rprice) | salestax, data = c1995)

coeftest(cig_ivreg, vcov = vcovHC, type = "HC1")
```

```
t test of coefficients:

            Estimate Std. Error t value  Pr(>|t|)
(Intercept)  9.71988    1.52832  6.3598 8.346e-08 ***
log(rprice) -1.08359    0.31892 -3.3977  0.001411 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We appreciate the same coefficient estimates for both approaches, although the latter standard errors differ from those previously computed with the manual approach in two steps. This is because the **standard errors** reported for the second-stage regression using `lm()` are **invalid**, as they do not account for the use of predictions from the first-stage regression as regressors in the second-stage regression.

Contrary to this, `ivreg()` performs the necessary adjustment automatically. Taking this into consideration together with the efficiency of the procedure, and although the step-by-step computation has been shown for demonstrating the mechanics of the procedure, it is **recommended to use `ivreg()` function when estimating TSLS**.

Additionally, it is important to compute heteroskedasticity-robust standard errors using `vcovHC()`, just like in multiple regression.

The TSLS estimate $\widehat{\beta}_1^{TSLS}$ of -1.08 suggests that the demand for cigarettes is actually elastic. Its interpretation is that an increase in the price of 1% is estimated to reduce consumption on average by approximately 1.08%.

Recalling the discussion of instrument exogeneity, perhaps this estimate should not yet be taken too seriously. Even though the elasticity was estimated using an instrumental variable, there might still be omitted variables that are correlated with the sales tax per pack. A multiple IV regression would be more appropriate to mitigate that risk.

6.4 Multiple IV Regression: The General IV Regression Model

The General Instrumental Variables Regression Model and Terminology

$$Y_i = \beta_0 + \beta_1 X_{1i} + ... + \beta_k X_{ki} + \beta_{k+1} W_{1i} + ... + \beta_{k+r} W_{ri} + u_i \tag{5.5}$$

with $i = 1, \dots, n$ is the general instrumental variables regression model where:

- Y_i is the dependent variable,
- $\beta_0, \dots, \beta_{k+1}$ are $1 + k + r$ unknown regression coefficients,
- $X_{1i}, \dots, X_{ki}$ are k endogenous regressors,
- $W_{1i}, \dots, W_{ri}$ are r exogenous regressors, which are uncorrelated with u_i,
- u_i is the error term,
- $Z_{1i}, \dots, Z_{mi}$ are m instrumental variables.

The coefficients are overidentified if $m > k$, they are underidentified if $m < k$, and they are exactly identified when $m = k$. Estimation of the IV regression model requires exact identification or overidentification.

TSLS in the General IV Model

First-stage regression(s): Regress each of the endogenous variables $(X_{1i}, \dots, X_{ki})$ on all instrumental variables $(Z_{1i}, \dots, Z_{mi})$, all exogenous variables $(W_{1i}, \dots, W_{ri})$ and an intercept. Compute the fitted values $(\hat{X}_{1i}, \dots, \hat{X}_{ki})$.

Second-stage regression: Regress the dependent variable on the predicted values of all endogenous regressors, all exogenous variables and an intercept using OLS. This gives $\hat{\beta}_0^{TSLS}, \dots, \hat{\beta}_{k+r}^{TSLS}$, the TSLS estimates of the model coefficients.

The IV Regression Assumptions

1. $E(u_i | W_{1i}, \dots, W_{ri}) = 0$
2. $(X_{1i}, \dots, X_{ki}, W_{1i}, \dots, W_{ri}, Z_{1i}, \dots, Z_{mi})$ are i.i.d. draws from their joint distribution.
3. All variables have nonzero finite fourth moments, i.e., outliers are unlikely.
4. The Zs are valid instruments

Two Conditions For Valid Instruments

For a set of m instruments $Z_{1i}, \dots, Z_{mi}$ to be valid, they must meet two conditions:

1. Instrument Relevance

If there are k endogenous variables, r exogenous variables and $m \geq k$ instruments Z, and $\widehat{X}^*_{1i}, \dots, \widehat{X}^*_{ki}$ are the predicted values from the k population first stage regressions, it must hold that $(\widehat{X}^*_{1i}, \dots, \widehat{X}^*_{ki}, W_{1i}, \dots, W_{ri}, 1)$ are not perfectly multicollinear. 1 denotes the constant regressor which equals 1 for all observations.

If there is only one endogenous regressor X_i, there must be at least one non-zero coefficient on the Z and the W in the population regression for this condition to be valid. If all of the coefficients are zero, all the $\widehat{X}^*_i$ are just the mean of X such that there is perfect multicollinearity.

2. Instrument Exogeneity

All m instruments must be uncorrelated with the error term: $\rho_{Z_{1i},u_i} = 0, \dots, \rho_{Z_{mi},u_i} = 0$

Employing TSLS functions in R such as `ivreg()` becomes more advantageous when dealing with a larger set of potentially endogenous regressors and instruments. It is straightforward, but there are, however, some specifications in correctly coding the regression formula.

Let's imagine we would like to estimate the model

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 W_{1i} + u_i$$

where X_{1i} and X_{2i} are endogenous regressors that shall be instrumented by Z_{1i}, Z_{2i} and Z_{3i}, and W_{1i} is an exogenous regressor.

The corresponding data is available in a `data.frame` with column names `y, x1, x2, w1, z1, z2` and `z3`.

While it might be tempting to specify the argument `formula` in the call of `ivreg()` as `y ~ x1 + x2 + w1 | z1 + z2 + z3` , this is wrong. It is necessary to list all exogenous variables as instruments too, that is joining them by +'s on the right of the vertical bar: `y ~ x1 + x2 + w1 | w1 + z1 + z2 + z3` where w1 is "instrumenting itself".

See `?ivreg` for the documentation of the function, where this is explained.

If we have a large number of exogenous variables, it might be convenient to provide an update formula with a `.` right after the `|` (this includes all variables except for the dependent variable) and to exclude all endogenous variables using a `-`.

For example, if there is one exogenous regressor `w1` and one endogenous regressor `x1` with instrument `z1`, the corresponding formula would be `y ~ w1 + x1 | w1 + z1`, which is equivalent to `y ~ w1 + x1 | . - x1 + z1`.

Application to the Demand for Cigarettes

As explained, although our previous regression function $\log(Q_i^{\text{cigarettes}}) = 9.72 - 1.08\log(P_i^{\text{cigarettes}})$ was estimated using IV regression, it is plausible that this estimate is biased, as the TSLS estimator is inconsistent for the true β_1 if the instrument (the real sales tax per pack) correlates with the error term.

There might still be omitted variables that are correlated with the sales tax per pack, such as income. States with higher incomes may rely less on sales tax and more on income tax to fund their state government. Additionally, the demand for cigarettes is likely influenced by income. Therefore, we aim to reevaluate our demand equation by incorporating income as a control variable:

$$\log(Q_i^{\text{cigarettes}}) = \beta_0 + \beta_1 \log(P_i^{\text{cigarettes}}) + \beta_2 \log(income_i) + u_i \tag{5.6}$$

Before estimating (5.6) using `ivreg()` we define *income* as real per capita income `rincome`, we append it to the data set `CigarettesSW` and we create a subset again for the year 1995. Then we estimate the model following the instructions previously explained.

```
# add rincome to the dataset and create subset for 1995
CigarettesSW$rincome <- with(CigarettesSW, income / population / cpi)
c1995 <- subset(CigarettesSW, year == "1995")

# estimate the model
cig_ivreg2 <- ivreg(log(packs) ~ log(rprice) + log(rincome) | log(rincome) +
                    salestax, data = c1995)
coeftest(cig_ivreg2, vcov = vcovHC, type = "HC1")
```

```
t test of coefficients:

             Estimate Std. Error t value  Pr(>|t|)
(Intercept)   9.43066    1.25939  7.4883 1.935e-09 ***
log(rprice)  -1.14338    0.37230 -3.0711  0.003611 **
log(rincome)  0.21452    0.31175  0.6881  0.494917
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We obtain

$$\log(Q_i^{\text{cigarettes}}) = \underset{(1.26)}{9.43} - \underset{(0.37)}{1.14} \log(P_i^{\text{cigarettes}}) + \underset{(0.31)}{0.21} \log(income_i) \tag{5.7}$$

We can now add the cigarette-specific taxes ($cigtax_i$) as a further instrumental variable and estimate again using TSLS.

```r
# add cigtax to the data set
CigarettesSW$cigtax <- with(CigarettesSW, tax/cpi)
c1995 <- subset(CigarettesSW, year == "1995")

# estimate the model
cig_ivreg3 <- ivreg(log(packs) ~ log(rprice) + log(rincome) |
                    log(rincome) + salestax + cigtax, data = c1995)
coeftest(cig_ivreg3, vcov = vcovHC, type = "HC1")
```

```
t test of coefficients:

             Estimate Std. Error t value  Pr(>|t|)    
(Intercept)   9.89496    0.95922 10.3157 1.947e-13 ***
log(rprice)  -1.27742    0.24961 -5.1177 6.211e-06 ***
log(rincome)  0.28040    0.25389  1.1044    0.2753    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If we use the instruments $salestax_i$ and $cigtax_i$ we would have 2 instruments ($m = 2$) and $k = 1$ so the coefficient on the endogenous regressor $\log(P_i^{\text{cigarettes}})$ is now *overidentified*.

The new TSLS estimate of (5.6) with two instruments is

$$\log(Q_i^{\text{cigarettes}}) = \underset{(0.96)}{9.89} - \underset{(0.25)}{1.28} \log(P_i^{\text{cigarettes}}) + \underset{(0.25)}{0.28} \log(income_i) \tag{5.8}$$

When we compare the estimates from models (5.7) and (5.8), we observe smaller standard errors in (5.8).

The standard error of the estimated price elasticity is smaller by one-third in this equation (0.25 versus 0.37). The reason is that more information is being used in this estimation: using two instruments explains more of the variation in cigarette prices than just one.

If the instruments are valid, which is something essential to be checked, (5.8) would be considered more reliable.

6.5 Instrument Validity

If the general sales tax and the cigarette-specific tax are not valid instruments, TSLS becomes inadequate for estimating the previously discussed demand elasticity for cigarettes. Although both variables are likely relevant, their exogeneity remains a separate issue.

Stock and Watson (2020) argue that cigarette-specific taxes could be endogenous due to state-specific historical factors, such as the economic significance of tobacco farming and cigarette production industries, which may advocate for lower cigarette-specific taxes.

Given the plausibility that states reliant on tobacco cultivation have higher smoking rates, this introduces endogeneity into cigarette-specific taxes. While incorporating data on the scale of the tobacco and cigarette industry into regression analysis could potentially address this concern, such data is unavailable.

Given that the role of the tobacco and cigarette industry varies across states but remains consistent over time, we will utilize the panel structure of `CigarettesSW`.

As outlined in the panel data chapter, conducting regressions based on data changes between two time periods eradicates state-specific and time-invariant effects. Our focus is on estimating the long-term elasticity of cigarette demand, thus we will examine changes in variables between 1985 and 1995.

Consequently, the model to be estimated via TSLS, employing the general sales tax and cigarette-specific sales tax as instruments, is as follows:

$$\begin{aligned}\log(Q_{i,1995}^{\text{cigarettes}}) - \log(Q_{i,1985}^{\text{cigarettes}}) = \beta_0 + \beta_1 \left[\log(P_{i,1995}^{\text{cigarettes}}) - \log(P_{i,1985}^{\text{cigarettes}})\right] \quad (6.1)\\ + \beta_2 \left[\log(\text{income}_{i,1995}) - \log(\text{income}_{i,1985})\right] + u_i \quad (5.9)\end{aligned}$$

We first create differences from 1985 to 1995 for the dependent variable, the regressors and both instruments:

```
# subset data for year 1985
c1985 <- subset(CigarettesSW, year == "1985")

# define differences in variables
packsdiff <- log(c1995$packs) - log(c1985$packs)

pricediff <- log(c1995$price/c1995$cpi) - log(c1985$price/c1985$cpi)

incomediff <- log(c1995$income/c1995$population/c1995$cpi) -
log(c1985$income/c1985$population/c1985$cpi)

salestaxdiff <- (c1995$taxs - c1995$tax)/c1995$cpi -
                                  (c1985$taxs - c1985$tax)/c1985$cpi

cigtaxdiff <- c1995$tax/c1995$cpi - c1985$tax/c1985$cpi
```

We now estimate three different IV regressions of (5.9) using `ivreg()`:

1. TSLS using just the difference in the sales taxes between 1985 and 1995 as instrument.
2. TSLS using just the difference in the cigarette-specific sales taxes 1985 and 1995 as instrument.
3. TSLS using both the difference in the sales taxes 1985 and 1995 and the difference in the cigarette-specific sales taxes 1985 and 1995 as instruments.

```
# estimate the three models
cig_ivreg_diff1 <- ivreg(packsdiff ~ pricediff + incomediff | incomediff +
                         salestaxdiff)

cig_ivreg_diff2 <- ivreg(packsdiff ~ pricediff + incomediff | incomediff +
                         cigtaxdiff)

cig_ivreg_diff3 <- ivreg(packsdiff ~ pricediff + incomediff | incomediff +
                         salestaxdiff + cigtaxdiff)
```

To obtain robust coefficient summaries for all models we use `coeftest()` together with `vcovHC()`

```
# robust coefficient summary for 1.
coeftest(cig_ivreg_diff1, vcov = vcovHC, type = "HC1")
```

```
t test of coefficients:

             Estimate Std. Error t value  Pr(>|t|)
(Intercept) -0.117962   0.068217 -1.7292   0.09062 .
pricediff   -0.938014   0.207502 -4.5205 4.454e-05 ***
incomediff   0.525970   0.339494  1.5493   0.12832
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# robust coefficient summary for 2.
coeftest(cig_ivreg_diff2, vcov = vcovHC, type = "HC1")
```

```
t test of coefficients:

             Estimate Std. Error t value  Pr(>|t|)
(Intercept) -0.017049   0.067217 -0.2536    0.8009
```

```
pricediff   -1.342515   0.228661 -5.8712 4.848e-07 ***
incomediff   0.428146   0.298718  1.4333    0.1587
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# robust coefficient summary for 3.
coeftest(cig_ivreg_diff3, vcov = vcovHC, type = "HC1")
```

```
t test of coefficients:

             Estimate Std. Error t value  Pr(>|t|)
(Intercept) -0.052003   0.062488 -0.8322    0.4097
pricediff   -1.202403   0.196943 -6.1053 2.178e-07 ***
incomediff   0.462030   0.309341  1.4936    0.1423
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can now present a tabulated summary of the estimation results with `stargazer()` (Hlavac 2022):

```r
# load stargazer
library(stargazer)

# gather robust standard errors in a list
rob_se <- list(sqrt(diag(vcovHC(cig_ivreg_diff1, type = "HC1"))),
               sqrt(diag(vcovHC(cig_ivreg_diff2, type = "HC1"))),
               sqrt(diag(vcovHC(cig_ivreg_diff3, type = "HC1"))))

# generate table
stargazer(cig_ivreg_diff1, cig_ivreg_diff2, cig_ivreg_diff3,
  se = rob_se,
  type="html",
  omit.stat = "f", df=FALSE)
```

```
<table style="text-align:center"><tr><td colspan="4" style="border-bottom: 1px solid black">
<tr><td></td><td colspan="3" style="border-bottom: 1px solid black"></td></tr>
<tr><td style="text-align:left"></td><td colspan="3">packsdiff</td></tr>
<tr><td style="text-align:left"></td><td>(1)</td><td>(2)</td><td>(3)</td></tr>
<tr><td colspan="4" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-ali
```

```
<tr><td style="text-align:left"></td><td>(0.208)</td><td>(0.229)</td><td>(0.197)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">incomediff</td><td>0.526</td><td>0.428</td><td>0.462</td></t
<tr><td style="text-align:left"></td><td>(0.339)</td><td>(0.299)</td><td>(0.309)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">Constant</td><td>-0.118<sup>*</sup></td><td>-0.017</td><td>-
<tr><td style="text-align:left"></td><td>(0.068)</td><td>(0.067)</td><td>(0.062)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td></tr>
<tr><td colspan="4" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-ali
<tr><td style="text-align:left">R<sup>2</sup></td><td>0.550</td><td>0.520</td><td>0.547</td>
<tr><td style="text-align:left">Adjusted R<sup>2</sup></td><td>0.530</td><td>0.498</td><td>0
<tr><td style="text-align:left">Residual Std. Error</td><td>0.091</td><td>0.094</td><td>0.09
<tr><td colspan="4" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-ali
</table>
```

In the table we observe different negative estimates for the coefficient on `pricediff`, all of them highly significant. How should we select the one to trust? This depends on the validity of the instruments employed. It would be useful to check for weak instruments.

6.5.1 Checking for Weak Instruments

Instruments that poorly explain changes in the endogenous regressor X are labeled as **weak instruments**. These weak instruments can lead to inaccurate estimates of the coefficient on the endogenous regressor.

Let's simplify this concept by considering a scenario with only one endogenous regressor, X, and m instruments denoted as $Z_1, \dots, Z_m$. If, in the population first-stage regression of a TSLS estimation, the coefficients for all instruments are zero, it implies that these instruments fail to explain any variation in X.

While encountering such a situation in practice is unlikely, there is a simple rule of thumb available for the most common situation in practice, the case of a single endogenous regressor.

Rule of Thumb for Checking for Weak Instruments

Compute the F-statistic which corresponds to the hypothesis that the coefficients on $Z_1, \dots, Z_m$ are all zero in the first-stage regression. If the F-statistic is less than 10, the instruments are weak, in which case the TSLS estimator is biased (also in large samples) and TSLS t-statistics and confidence intervals are unreliable.

In R this would be implemented by running the first-stage regression using `lm()` and computing the heteroskedasticity-robust F-statistic by means of `linearHypothesis()`. Let's compute this for all three models:

```
# first-stage regressions
mod_relevance1 <- lm(pricediff ~ salestaxdiff + incomediff)
mod_relevance2 <- lm(pricediff ~ cigtaxdiff + incomediff)
mod_relevance3 <- lm(pricediff ~ incomediff + salestaxdiff + cigtaxdiff)

# check instrument relevance for model (1)
linearHypothesis(mod_relevance1,
                 "salestaxdiff = 0",
                 vcov = vcovHC, type = "HC1")
```

```
Linear hypothesis test

Hypothesis:
salestaxdiff = 0

Model 1: restricted model
Model 2: pricediff ~ salestaxdiff + incomediff

Note: Coefficient covariance matrix supplied.

  Res.Df Df      F    Pr(>F)
1     46
2     45  1 28.445 3.009e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# check instrument relevance for model (2)
linearHypothesis(mod_relevance2,
                 "cigtaxdiff = 0",
                 vcov = vcovHC, type = "HC1")
```

```
Linear hypothesis test

Hypothesis:
cigtaxdiff = 0

Model 1: restricted model
```

```
Model 2: pricediff ~ cigtaxdiff + incomediff

Note: Coefficient covariance matrix supplied.

  Res.Df Df      F   Pr(>F)
1     46
2     45  1 98.034 7.09e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# check instrument relevance for model (3)
linearHypothesis(mod_relevance3,
                 c("salestaxdiff = 0", "cigtaxdiff = 0"),
                 vcov = vcovHC, type = "HC1")
```

```
Linear hypothesis test

Hypothesis:
salestaxdiff = 0
cigtaxdiff = 0

Model 1: restricted model
Model 2: pricediff ~ incomediff + salestaxdiff + cigtaxdiff

Note: Coefficient covariance matrix supplied.

  Res.Df Df      F    Pr(>F)
1     46
2     44  2 76.916 4.339e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

When coefficients are *overidentified* ($m > k$), like in our third model, we can apply the **overidentifying restrictions test** (also called the J-test), which is an approach to test the hypothesis that *additional* instruments are exogenous.

J-Statistic / Overidentifying Restrictions Test

Take $\hat{u}_i^{TSLS}$, $i = 1 \dots, n$, the residuals from TSLS estimation of the general IV regression model (5.5), and run the OLS regression to estimate the coefficients in

$$\hat{u}_{\mathrm{TSLS}_i} = \delta_0 + \delta_1 Z_{1i} + \dots + \delta_m Z_{mi} + \delta_{m+1} W_{1i} + \dots + \delta_{m+r} W_{ri} + e_i \tag{5.10}$$

where e_i is the regression error term. Now test the joint hypothesis

$$H_0 : \delta_1 = 0, \dots, \delta = 0$$

that states that all instruments are exogenous. Let F denote the homoskedasticity-only F-statistic testing the null hypothesis. The overidentifying restrictions test statistic is then

$$J = mF$$

also called the J-statistic. Under the null hypothesis that all the instruments are exogenous, if e_i is homoskedastic, in large samples

$$J \sim \chi^2_{m-k}$$

where $m - k$ is the degree of overidentification, or in other words, the number of instruments minus the number of endogenous regressors.

To conduct the overidentifying restrictions test for model three, which is the only model where the coefficient on the difference in log prices is overidentified ($m = 2, k = 1$), allowing computation of the J-statistic, we proceed as follows:

1. We use the residuals stored in `cig_ivreg_diff3` and regress them on both instruments and the presumably exogenous regressor `incomediff`.
2. Once more, we employ `linearHypothesis()` to examine whether the coefficients on both instruments are zero, a prerequisite for fulfilling the exogeneity assumption. It's important to note that we specify `test = "Chisq"` to obtain a chi-squared distributed test statistic instead of an F-statistic.

```
# compute the J-statistic
cig_iv_OR <- lm(residuals(cig_ivreg_diff3) ~ incomediff + salestaxdiff + cigtaxdiff)

cig_OR_test <- linearHypothesis(cig_iv_OR,
                                c("salestaxdiff = 0", "cigtaxdiff = 0"),
                                test = "Chisq")
cig_OR_test
```

```
Linear hypothesis test

Hypothesis:
salestaxdiff = 0
cigtaxdiff = 0

Model 1: restricted model
Model 2: residuals(cig_ivreg_diff3) ~ incomediff + salestaxdiff + cigtaxdiff

  Res.Df     RSS Df Sum of Sq Chisq Pr(>Chisq)
1     46 0.37472
2     44 0.33695  2  0.037769 4.932    0.08492 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Caution! The p-Value provided by `linearHypothesis()` might be misleading, because the degrees of freedom are automatically set to 2. This differs from the degree of overidentification ($m - k = 2 - 1 = 1$), making the J-statistic follow a χ^2_1 distribution instead of the default assumption of χ^2_2 distribution in `linearHypothesis()`.

We can easily compute the correct p-Value using `pchisq()`.

```r
# compute correct p-value for J-statistic
pchisq(cig_OR_test[2, 5], df = 1, lower.tail = FALSE)
```

```
[1] 0.02636406
```

Since the reported value is smaller than 0.05, we reject the null hypothesis that both instruments are exogenous at the 5% level. From this we can deduce that one of the following statements is true:

1. The sales tax is an invalid instrument for the cigarettes package price.
2. The cigarettes-specific sales tax is an invalid instrument for the cigarettes package price.
3. Both instruments are invalid.

Stock and Watson (2020) argue that the case for the exogeneity of the general sales tax is stronger than that for the cigarette-specific tax, since the political process can link changes in the cigarette-specific tax to changes in the cigarette market and smoking policy.

Taking this into consideration, the IV estimate of the long-run elasticity of demand for cigarettes considered the most trustworthy would be -0.94, the TSLS estimate obtained using the general sales tax as the only instrument.

6.6 Summary

The instrument variable selected for our model is the general sales tax. The IV regression model making use of this instrument is

$$\log(Q_{i,1995}^{\text{cigarettes}}) - \log(Q_{i,1985}^{\text{cigarettes}}) = -0.118 - 0.938 \left[\log(P_{i,1995}^{\text{cigarettes}}) - \log(P_{i,1985}^{\text{cigarettes}})\right] \quad (6.2)$$
$$+ 0.526 \left[\log(\text{income}_{i,1995}) - \log(\text{income}_{i,1985})\right] + u_i \quad (5.9)$$

This estimate indicates that the cigarette consumption is elastic: over a 10-year period, an increase in the average price per package by 1% is expected to reduce consumption on average by 0.94 percentage points. This suggests that, over the long term, rises in the price per pack can significantly decrease cigarette consumption.

We have seen how easy and straightforward it is to estimate IV regression models in R with the `ivreg()` function from the package `AER`. This facilitates and simplifies the implementation of the TSLS estimation approach.

Besides treating IV estimation, we have also discussed how important it is to to test for weak instruments and how to conduct the corresponding tests, including the overidentifying restrictions test when there are more instruments than endogenous regressors.

Furthermore, we have implemented a long-run analysis of the demand for cigarettes and its elasticity, being able to make a conclusion after selecting the most trustworthy instrumental variable.

6.7 References

Hlavac, Marek. 2022. Stargazer: Well-Formatted Regression and Summary Statistics Tables. Bratislava, Slovakia: Social Policy Institute. https://CRAN.R-project.org/package=stargazer.

Kleiber, Christian, and Achim Zeileis. 2008. Applied Econometrics with R. New York: Springer-Verlag. https://CRAN.R-project.org/package=AER.

Stock, J. H., and M. W. Watson. 2020. Introduction to Econometrics, Fourth Update, Global Edition. Pearson Education Limited.

7 Empirical Applications of Experiments

In this chapter, we explore statistical techniques frequently used to quantify the causal impacts of programs, policies, or interventions. Statisticians advocate for an optimal research design known as an ideal randomized controlled experiment, which involves randomly allocating subjects into two distinct groups: a treatment group receiving the intervention and a control group not receiving it. By comparing outcomes between these groups, researchers can estimate the average treatment effect.

We will make use of the following packages in R:

- AER (Christian Kleiber and Zeileis 2008)
- dplyr (Wickham et al. 2023)
- MASS (Ripley 2023)
- mvtnorm (Genz et al. 2023)
- rddtools (Stigler and Quast 2022)
- scales (Wickham and Seidel 2022)
- stargazer(Hlavac 2022)
- tidyr (Wickham, Vaughan, and Girlich 2023)

```
library(AER)
library(dplyr)
library(MASS)
library(mvtnorm)
library(rddtools)
library(scales)
library(stargazer)
library(tidyr)
```

8 Experiments

8.1 Data Set Description & Experimental Design

The Project Student-Teacher Achievement Ratio (STAR) was a large-scale randomized controlled experiment aimed at determining the effectiveness of class size reduction in improving elementary education.

This 4-year experiment took place during the 1980s in 80 elementary schools across Tennessee by the State Department of Education.

During the initial year, approximately 6,400 students were randomly allocated to one of three interventions:

- **Treatment 1**: small class (13 to 17 students per teacher).
- **Treatment 2**: regular-with-aide class (22 to 25 students with a full-time teacher's aide).
- **Control group**: regular class (22 to 25 students per teacher).

Additionally, teachers were randomly assigned to the classes they taught. These interventions started as students entered kindergarten and continued until third grade.

The students' academic evolution was evaluated by aggregating the scores achieved on both the math and reading sections of the Stanford Achievement Test.

Let's start loading the `STAR` data set from the `AER` package and exploring it

```
# load STAR data set
data("STAR")

# get an overview
head(STAR, 2)
```

```
     gender ethnicity   birth stark star1 star2   star3 readk read1 read2 read3
1122 female      afam 1979 Q3  <NA>  <NA>  <NA> regular    NA    NA    NA   580
1137 female      cauc 1980 Q1 small small small   small   447   507   568   587
     mathk math1 math2 math3   lunchk lunch1   lunch2 lunch3 schoolk school1
1122    NA    NA    NA   564     <NA>   <NA>     <NA>   free    <NA>    <NA>
```

```
1137   473   538   579   593 non-free   free non-free   free   rural   rural
     school2  school3  degreek  degree1  degree2  degree3 ladderk ladder1
1122    <NA> suburban     <NA>     <NA>     <NA> bachelor    <NA>    <NA>
1137   rural    rural bachelor bachelor bachelor bachelor  level1  level1
        ladder2    ladder3 experiencek experience1 experience2 experience3
1122       <NA>     level1          NA          NA          NA          30
1137 apprentice apprentice           7           7           3           1
     tethnicityk tethnicity1 tethnicity2 tethnicity3 systemk system1 system2
1122        <NA>        <NA>        <NA>        cauc    <NA>    <NA>    <NA>
1137        cauc        cauc        cauc        cauc      30      30      30
     system3 schoolidk schoolid1 schoolid2 schoolid3
1122      22      <NA>      <NA>      <NA>        54
1137      30        63        63        63        63
```

```
dim(STAR)
```

```
[1] 11598    47
```

```
# get variable names
names(STAR)
```

```
 [1] "gender"      "ethnicity"   "birth"       "stark"       "star1"
 [6] "star2"       "star3"       "readk"       "read1"       "read2"
[11] "read3"       "mathk"       "math1"       "math2"       "math3"
[16] "lunchk"      "lunch1"      "lunch2"      "lunch3"      "schoolk"
[21] "school1"     "school2"     "school3"     "degreek"     "degree1"
[26] "degree2"     "degree3"     "ladderk"     "ladder1"     "ladder2"
[31] "ladder3"     "experiencek" "experience1" "experience2" "experience3"
[36] "tethnicityk" "tethnicity1" "tethnicity2" "tethnicity3" "systemk"
[41] "system1"     "system2"     "system3"     "schoolidk"   "schoolid1"
[46] "schoolid2"   "schoolid3"
```

We observe a variety of factor variables describing student and teacher characteristics, as well as several school indicators recorded for each of the four academic years.

The data set contains a total of 11598 observations on 47 variables and it is presented in what is called a *wide* format, that is, each column represents a variable and each student is represented by a row, where the values for each variable are recorded.

We see that most of the variable names end with a suffix (k, 1, 2, 3) which correspond to the grade for which the value of the variable was registered. This allows adjusting the `formula` argument in `lm()` for each grade by simply changing the variables' suffixes accordingly.

From the output of `head(STAR, 2)` we observe some missing values as `NA`. This is because the student entered the experiment in the third grade in a regular class.

Consequently, the class size is documented in `star3`, while the other class type indicator variables are marked as `NA`. The student's math and reading scores for the third grade are provided, while data for other grades are absent for the same reason.

To obtain only her non-missing recordings, we can easily remove the NAs using the `!is.na()` function.

```
# drop NA recordings for the first observation and print to the console
STAR[1, !is.na(STAR[1, ])]
```

```
     gender ethnicity   birth   star3 read3 math3 lunch3   school3  degree3
1122 female      afam 1979 Q3 regular   580   564   free suburban bachelor
     ladder3 experience3 tethnicity3 system3 schoolid3
1122  level1          30        cauc      22        54
```

`is.na(STAR[1, ])` returns a logical vector with `TRUE` at positions that correspond to missing entries for the first observation. By using the `!` operator, we invert the result to obtain only non-NA entries for the first student in the data set.

When using `lm()`, it is not necessary to remove rows with missing data, as it is done by default. Removing missing data might lead to a small number of observations, which can make our estimates less accurate and our conclusions unreliable.

However, this isn't a problem in our study because, as we'll see later, we have more than 5000 observations for each of the regressions we will conduct.

8.2 Analysis of the STAR data

Because there are two treatment groups (small class and regular-sized class with an aide), the regression version of the differences estimator requires adjustment to accommodate these groups along with the control group.

This adjustment involves introducing two binary variables: one indicating whether the student is in a small class and another indicating whether the student is in a regular-sized class with an aide. This leads to the population regression model

$$Y_i = \beta_0 + \beta_1 \, SmallClass_i + \beta_2 \, RegAide_i + u_i \tag{6.1}$$

where Y_i represents a test score, $SmallClass_i$ equals 1 if the i^{th} student is in a small class and 0 otherwise, and $RegAide_i$ equals 1 if the i^{th} student is in a regular class with an aide and 0 otherwise.

The effect on the test score of being in a small class relative to a regular class is β_1, and the effect of being in a regular class with an aide relative to a regular class is β_2.

The differences estimator for the experiment can then be calculated by estimating β_1 and β_2 in Equation (6.1) using ordinary least squares (OLS).

We will now perform regression (6.1) for each grade separately. The dependent variable will be the sum of the points scored in the math and reading parts, which can be constructed using `I()`.

```
# compute differences estimates for each grade
fmk <- lm(I(readk + mathk) ~ stark, data = STAR) # kindergarten
fm1 <- lm(I(read1 + math1) ~ star1, data = STAR) # first grade
fm2 <- lm(I(read2 + math2) ~ star2, data = STAR) # second grade
fm3 <- lm(I(read3 + math3) ~ star3, data = STAR) # third grade
```

```
# obtain coefficient matrix using robust standard errors
coeftest(fmk, vcov = vcovHC, type= "HC1")
```

```
t test of coefficients:

                   Estimate Std. Error  t value  Pr(>|t|)
(Intercept)       918.04289    1.63339 562.0473 < 2.2e-16 ***
starksmall         13.89899    2.45409   5.6636 1.554e-08 ***
starkregular+aide   0.31394    2.27098   0.1382    0.8901
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(fm1, vcov = vcovHC, type= "HC1")
```

```
t test of coefficients:

                   Estimate Std. Error  t value  Pr(>|t|)
(Intercept)       1039.3926     1.7846 582.4321 < 2.2e-16 ***
star1small          29.7808     2.8311  10.5190 < 2.2e-16 ***
star1regular+aide   11.9587     2.6520   4.5093  6.62e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coeftest(fm2, vcov = vcovHC, type= "HC1")
```

```
t test of coefficients:

                   Estimate Std. Error  t value  Pr(>|t|)    
(Intercept)       1157.8066     1.8151 637.8820 < 2.2e-16 ***
star2small          19.3944     2.7117   7.1522  9.55e-13 ***
star2regular+aide    3.4791     2.5447   1.3672    0.1716    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coeftest(fm3, vcov = vcovHC, type= "HC1")
```

```
t test of coefficients:

                    Estimate Std. Error  t value  Pr(>|t|)    
(Intercept)       1228.50636    1.68001 731.2483 < 2.2e-16 ***
star3small          15.58660    2.39604   6.5051 8.393e-11 ***
star3regular+aide   -0.29094    2.27271  -0.1280    0.8981    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can present as usual our results in a table using `stargazer()`

```r
# compute robust standard errors for each model and gather them in a list
rob_se_1 <- list(sqrt(diag(vcovHC(fmk, type = "HC1"))),
                 sqrt(diag(vcovHC(fm1, type = "HC1"))),
                 sqrt(diag(vcovHC(fm2, type = "HC1"))),
                 sqrt(diag(vcovHC(fm3, type = "HC1"))))

stargazer(fmk,fm1,fm2,fm3,
  se = rob_se_1,
  type="html",
  omit.stat = "f", df=FALSE)
```

```
<table style="text-align:center"><tr><td colspan="5" style="border-bottom: 1px solid black">
<tr><td></td><td colspan="4" style="border-bottom: 1px solid black"></td></tr>
```

```
<tr><td style="text-align:left"></td><td>I(readk + mathk)</td><td>I(read1 + math1)</td><td>I
<tr><td style="text-align:left"></td><td>(1)</td><td>(2)</td><td>(3)</td><td>(4)</td></tr>
<tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-alig
<tr><td style="text-align:left"></td><td>(2.454)</td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">starkregular+aide</td><td>0.314</td><td></td><td></td><td></
<tr><td style="text-align:left"></td><td>(2.271)</td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">star1small</td><td></td><td>29.781<sup>***</sup></td><td></t
<tr><td style="text-align:left"></td><td></td><td>(2.831)</td><td></td><td></td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">star1regular+aide</td><td></td><td>11.959<sup>***</sup></td>
<tr><td style="text-align:left"></td><td></td><td>(2.652)</td><td></td><td></td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">star2small</td><td></td><td></td><td>19.394<sup>***</sup></t
<tr><td style="text-align:left"></td><td></td><td></td><td>(2.712)</td><td></td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">star2regular+aide</td><td></td><td></td><td>3.479</td><td></
<tr><td style="text-align:left"></td><td></td><td></td><td>(2.545)</td><td></td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">star3small</td><td></td><td></td><td></td><td>15.587<sup>***
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(2.396)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">star3regular+aide</td><td></td><td></td><td></td><td>-0.291<
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(2.273)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">Constant</td><td>918.043<sup>***</sup></td><td>1,039.393<sup
<tr><td style="text-align:left"></td><td>(1.633)</td><td>(1.785)</td><td>(1.815)</td><td>(1.
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-alig
<tr><td style="text-align:left">R<sup>2</sup></td><td>0.007</td><td>0.017</td><td>0.009</td>
<tr><td style="text-align:left">Adjusted R<sup>2</sup></td><td>0.007</td><td>0.017</td><td>0
<tr><td style="text-align:left">Residual Std. Error</td><td>73.490</td><td>90.501</td><td>83
<tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-alig
</table>
```

Based on the estimates, students in kindergarten seem to benefit significantly from being in smaller classes, showing an average test score increase of 13.9 points compared to those in regular classes.

However, the effect of having an aide in a regular class is minimal, with an estimated increase of only 0.31 points on the test.

Across all grades, the data indicates that smaller classes lead to improved test scores, rejecting the idea that they provide no benefit at a 1% significance level.

Yet, the evidence for the effectiveness of having an aide in a regular class is less conclusive, except for first graders, even at a 10% significance level.

The estimated improvements in smaller classes are similar across kindergarten, 2nd, and 3rd grades, though the effect appears slightly stronger in first grade.

Overall, the results suggest that reducing class size has a noticeable impact on test performance, whereas adding an aide to a regular-sized class has only a minor effect, possibly close to zero.

8.3 Including Additional Regressors

In our study case, there may be other variables that explain the variation in the dependent variable. For this reason, by adding additional regressors to the model, we can enhance the precision of the estimated causal effects.

The differences estimator with additional regressors is more efficient than the differences estimator if the additional regressors explain some of the variation in the dependent variable.

Moreover, if the treatment allocation was not completely random due to protocol deviations, our previous estimates could be biased.

To address these concerns and provide more robust estimates, we will now include additional regressors measuring teacher, school, and student characteristics, particularly focusing on kindergarten. We consider the following variables:

- `experience` - Teacher's years of experience
- `boy` - Student is a boy (dummy)
- `lunch` - Free lunch eligibility (dummy)
- `black` - Student is African-American (dummy)
- `race` - Student's race is other than black or white (dummy)
- `schoolid` - School indicator variables

We will use these extra regressors to estimate the following models:

$$
\begin{aligned}
Y_i &= \beta_0 + \beta_1\, SmallClass_i + \beta_2\, RegAide_i + u_i, && (6.2)\\
Y_i &= \beta_0 + \beta_1\, SmallClass_i + \beta_2\, RegAide_i + \beta_3\, experience_i + u_i, && (6.3)\\
Y_i &= \beta_0 + \beta_1\, SmallClass_i + \beta_2\, RegAide_i + \beta_3\, experience_i + \ schoolid + u_i, && (6.4)\\
Y_i &= \beta_0 + \beta_1\, SmallClass_i + \beta_2\, RegAide_i + \beta_3\, experience_i + \beta_4\, boy + \beta_5\, lunch && (8.1)\\
&\quad + \ \beta_6\, black + \beta_7\, race + schoolid + u_i. && (6.5)
\end{aligned}
$$

With the help of functions from the `dplyr` and `tidyr` packages, we will create our custom subset of the data, including only kindergarten data.

First, we will use `transmute()` to keep only relevant variables (`gender`, `ethnicity`, `stark`, `readk`, `mathk`, `lunchk`, `experiencek` and `schoolidk`) and drop the rest.

Then, using `mutate()` and logical statements within the function `ifelse()`, we will add the additional binary variables `black`, `race` and `boy`.

```
# generate subset with kindergarten data
STARK <- STAR %>%
      transmute(gender,
                ethnicity,
                stark,
                readk,
                mathk,
                lunchk,
                experiencek,
                schoolidk) |>
      mutate(black = ifelse(ethnicity == "afam", 1, 0),
             race = ifelse(ethnicity == "afam" | ethnicity == "cauc", 1, 0),
             boy = ifelse(gender == "male", 1, 0))
```

```
# estimate the models
gradeK1 <- lm(I(mathk + readk) ~ stark + experiencek,
              data = STARK)

gradeK2 <- lm(I(mathk + readk) ~ stark + experiencek + schoolidk,
              data = STARK)

gradeK3 <- lm(I(mathk + readk) ~ stark + experiencek + boy + lunchk
              + black + race + schoolidk,
              data = STARK)
```

To keep it short, we skip displaying the coefficients for the indicator dummies in the `coeftest()` output by subsetting the matrices.

```
# obtain robust inference on the significance of coefficients
coeftest(gradeK1, vcov. = vcovHC, type = "HC1")
```

```
t test of coefficients:

                    Estimate Std. Error  t value  Pr(>|t|)    
(Intercept)        904.72124    2.22235 407.1020 < 2.2e-16 ***
starksmall          14.00613    2.44704   5.7237 1.095e-08 ***
starkregular+aide   -0.60058    2.25430  -0.2664    0.7899    
experiencek          1.46903    0.16929   8.6778 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(gradeK2, vcov. = vcovHC, type = "HC1")[1:4, ]
```

```
                     Estimate Std. Error     t value     Pr(>|t|)
(Intercept)       925.6748750  7.6527218 120.9602155 0.000000e+00
starksmall         15.9330822  2.2411750   7.1092540 1.310324e-12
starkregular+aide   1.2151960  2.0353415   0.5970477 5.504993e-01
experiencek         0.7431059  0.1697619   4.3773429 1.222880e-05
```

```
coeftest(gradeK3, vcov. = vcovHC, type = "HC1")[1:7, ]
```

```
                     Estimate Std. Error     t value     Pr(>|t|)
(Intercept)       937.6831330 14.3726687  65.2407117 0.000000e+00
starksmall         15.8900507  2.1551817   7.3729516 1.908960e-13
starkregular+aide   1.7869378  1.9614592   0.9110247 3.623211e-01
experiencek         0.6627251  0.1659298   3.9940097 6.578846e-05
boy               -12.0905123  1.6726331  -7.2284306 5.533119e-13
lunchkfree        -34.7033021  1.9870366 -17.4648529 1.437931e-66
black             -25.4305130  3.4986918  -7.2685776 4.125252e-13
```

And we display the results in a `stargazer()` table

```r
# compute robust standard errors for each model and gather them in a list
rob_se_2 <- list(sqrt(diag(vcovHC(fmk, type = "HC1"))),
                 sqrt(diag(vcovHC(gradeK1, type = "HC1"))),
                 sqrt(diag(vcovHC(gradeK2, type = "HC1"))),
                 sqrt(diag(vcovHC(gradeK3, type = "HC1"))))
stargazer(fmk, gradeK1, gradeK2, gradeK3,
          se = rob_se_2,
          type="html",
          omit.stat = "f", df=FALSE)
```

```
<table style="text-align:center"><tr><td colspan="5" style="border-bottom: 1px solid black">
<tr><td></td><td colspan="4" style="border-bottom: 1px solid black"></td></tr>
<tr><td style="text-align:left"></td><td>I(readk + mathk)</td><td colspan="3">I(mathk + readk
<tr><td style="text-align:left"></td><td>(1)</td><td>(2)</td><td>(3)</td><td>(4)</td></tr>
<tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-alig
<tr><td style="text-align:left"></td><td>(2.454)</td><td>(2.447)</td><td>(2.241)</td><td>(2.
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">starkregular+aide</td><td>0.314</td><td>-0.601</td><td>1.215
<tr><td style="text-align:left"></td><td>(2.271)</td><td>(2.254)</td><td>(2.035)</td><td>(1.
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">experiencek</td><td></td><td>1.469<sup>***</sup></td><td>0.7
<tr><td style="text-align:left"></td><td></td><td>(0.169)</td><td>(0.170)</td><td>(0.166)</t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">boy</td><td></td><td></td><td></td><td>-12.091<sup>***</sup>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(1.673)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">lunchkfree</td><td></td><td></td><td></td><td>-34.703<sup>**
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(1.987)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">black</td><td></td><td></td><td></td><td>-25.431<sup>***</su
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(3.499)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">race</td><td></td><td></td><td></td><td>8.501</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(12.520)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk2</td><td></td><td></td><td>-81.716<sup>***</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.134)</td><td>(9.096)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk3</td><td></td><td></td><td>-7.175</td><td>-7.833</
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.474)</td><td>(8.809)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
```

```
<tr><td style="text-align:left">schoolidk4</td><td></td><td></td><td>-44.735<sup>***</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.093)</td><td>(8.739)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk5</td><td></td><td></td><td>-48.425<sup>***</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(12.594)</td><td>(11.710)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk6</td><td></td><td></td><td>-38.441<sup>***</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.456)</td><td>(10.042)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk7</td><td></td><td></td><td>22.672<sup>**</sup></td>
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.891)</td><td>(9.916)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk8</td><td></td><td></td><td>-34.505<sup>***</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.000)</td><td>(8.451)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk9</td><td></td><td></td><td>4.032</td><td>2.880</td>
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.875)</td><td>(9.174)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk10</td><td></td><td></td><td>38.558<sup>***</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(13.524)</td><td>(12.536)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk11</td><td></td><td></td><td>57.981<sup>***</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(13.795)</td><td>(12.332)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk12</td><td></td><td></td><td>1.828</td><td>11.213</
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.199)</td><td>(10.627)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk13</td><td></td><td></td><td>36.435<sup>***</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.658)</td><td>(11.391)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk14</td><td></td><td></td><td>-32.964<sup>**</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(13.367)</td><td>(13.104)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk15</td><td></td><td></td><td>-51.949<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.803)</td><td>(9.641)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk16</td><td></td><td></td><td>-76.829<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(8.916)</td><td>(8.933)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk17</td><td></td><td></td><td>-18.900<sup>*</sup></t
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.388)</td><td>(10.358)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk18</td><td></td><td></td><td>-51.424<sup>***</sup><
```

```
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.513)</td><td>(8.909)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk19</td><td></td><td></td><td>-29.031<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.806)</td><td>(9.615)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk20</td><td></td><td></td><td>-24.413<sup>**</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.345)</td><td>(9.960)</td></tr
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk21</td><td></td><td></td><td>16.309</td><td>32.496<
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.281)</td><td>(10.092)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk22</td><td></td><td></td><td>-16.033</td><td>27.407
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.091)</td><td>(9.984)</td></tr
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk23</td><td></td><td></td><td>31.210<sup>***</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.868)</td><td>(11.327)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk24</td><td></td><td></td><td>-39.174<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.481)</td><td>(9.151)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk25</td><td></td><td></td><td>-33.916<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.540)</td><td>(10.580)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk26</td><td></td><td></td><td>-65.901<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.595)</td><td>(9.825)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk27</td><td></td><td></td><td>20.344<sup>**</sup></t
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.242)</td><td>(9.206)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk28</td><td></td><td></td><td>-43.699<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.613)</td><td>(9.531)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk29</td><td></td><td></td><td>-19.793<sup>*</sup></t
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.724)</td><td>(10.518)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk30</td><td></td><td></td><td>73.697<sup>***</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.865)</td><td>(12.013)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk31</td><td></td><td></td><td>5.703</td><td>51.622<s
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.708)</td><td>(11.818)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk32</td><td></td><td></td><td>-76.597<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(8.915)</td><td>(8.876)</td></tr>
```

```
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk33</td><td></td><td></td><td>-74.895<sup>***</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.119)</td><td>(9.079)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk34</td><td></td><td></td><td>-43.062<sup>***</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.930)</td><td>(9.615)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk35</td><td></td><td></td><td>-46.238<sup>***</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.735)</td><td>(9.907)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk36</td><td></td><td></td><td>-13.686</td><td>-21.77
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.091)</td><td>(10.422)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk37</td><td></td><td></td><td>-12.351</td><td>-7.768
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.731)</td><td>(9.043)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk38</td><td></td><td></td><td>-10.610</td><td>-2.694
<tr><td style="text-align:left"></td><td></td><td></td><td>(13.425)</td><td>(12.969)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk39</td><td></td><td></td><td>-23.309<sup>**</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.394)</td><td>(11.003)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk40</td><td></td><td></td><td>17.470</td><td>30.083<
<tr><td style="text-align:left"></td><td></td><td></td><td>(12.397)</td><td>(11.823)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk41</td><td></td><td></td><td>51.231<sup>***</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(12.830)</td><td>(12.484)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk42</td><td></td><td></td><td>-2.847</td><td>-10.336
<tr><td style="text-align:left"></td><td></td><td></td><td>(15.449)</td><td>(14.884)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk43</td><td></td><td></td><td>-16.442</td><td>-25.76
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.389)</td><td>(10.963)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk44</td><td></td><td></td><td>6.826</td><td>43.208<s
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.638)</td><td>(10.780)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk45</td><td></td><td></td><td>-105.064<sup>***</sup>
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.504)</td><td>(9.470)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk46</td><td></td><td></td><td>-17.011</td><td>-21.00
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.569)</td><td>(9.861)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
```

```
<tr><td style="text-align:left">schoolidk47</td><td></td><td></td><td>-18.214<sup>*</sup></t
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.842)</td><td>(10.458)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk48</td><td></td><td></td><td>-4.983</td><td>0.849</
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.414)</td><td>(9.766)</td></tr
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk49</td><td></td><td></td><td>9.272</td><td>12.206</
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.010)</td><td>(10.467)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk50</td><td></td><td></td><td>-3.353</td><td>1.796</
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.661)</td><td>(9.751)</td></tr
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk51</td><td></td><td></td><td>11.627</td><td>7.187</
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.856)</td><td>(9.426)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk52</td><td></td><td></td><td>30.792<sup>**</sup></t
<tr><td style="text-align:left"></td><td></td><td></td><td>(13.018)</td><td>(12.542)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk53</td><td></td><td></td><td>-66.974<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.075)</td><td>(9.655)</td></tr
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk54</td><td></td><td></td><td>-0.463</td><td>-6.485<
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.873)</td><td>(11.069)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk55</td><td></td><td></td><td>-35.635<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.893)</td><td>(9.439)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk56</td><td></td><td></td><td>-90.780<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.488)</td><td>(10.056)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk57</td><td></td><td></td><td>-43.141<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.092)</td><td>(10.424)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk58</td><td></td><td></td><td>19.694<sup>**</sup></t
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.284)</td><td>(8.753)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk59</td><td></td><td></td><td>10.294</td><td>8.543</
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.952)</td><td>(11.501)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk60</td><td></td><td></td><td>-40.156<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.474)</td><td>(9.726)</td></tr
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk61</td><td></td><td></td><td>-28.758<sup>***</sup><
```

```
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.152)</td><td>(9.478)</td></tr
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk62</td><td></td><td></td><td>-36.164<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.480)</td><td>(8.703)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk63</td><td></td><td></td><td>18.475<sup>*</sup></td
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.873)</td><td>(9.166)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk64</td><td></td><td></td><td>-24.511<sup>**</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.897)</td><td>(10.126)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk65</td><td></td><td></td><td>17.476</td><td>16.055<
<tr><td style="text-align:left"></td><td></td><td></td><td>(13.031)</td><td>(12.792)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk66</td><td></td><td></td><td>-36.682<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.523)</td><td>(9.020)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk67</td><td></td><td></td><td>-13.550</td><td>-10.02
<tr><td style="text-align:left"></td><td></td><td></td><td>(13.024)</td><td>(12.418)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk68</td><td></td><td></td><td>26.870<sup>***</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.380)</td><td>(9.764)</td></tr
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk69</td><td></td><td></td><td>24.625<sup>**</sup></t
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.495)</td><td>(10.898)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk70</td><td></td><td></td><td>-20.916<sup>**</sup></
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.251)</td><td>(9.912)</td></tr
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk71</td><td></td><td></td><td>-39.109<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(9.869)</td><td>(9.352)</td></tr>
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk72</td><td></td><td></td><td>11.188</td><td>19.390<
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.451)</td><td>(9.696)</td></tr
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk73</td><td></td><td></td><td>-0.435</td><td>0.456</
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.868)</td><td>(11.019)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk74</td><td></td><td></td><td>4.574</td><td>6.229</t
<tr><td style="text-align:left"></td><td></td><td></td><td>(12.049)</td><td>(11.388)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk75</td><td></td><td></td><td>7.822</td><td>-1.319</
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.975)</td><td>(11.415)</td></t
```

```
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk76</td><td></td><td></td><td>-15.980</td><td>-19.275
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.087)</td><td>(9.681)</td></tr
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk78</td><td></td><td></td><td>-48.027<sup>***</sup><
<tr><td style="text-align:left"></td><td></td><td></td><td>(10.058)</td><td>(9.656)</td></tr
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk79</td><td></td><td></td><td>-15.241</td><td>-15.33
<tr><td style="text-align:left"></td><td></td><td></td><td>(11.521)</td><td>(10.519)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">schoolidk80</td><td></td><td></td><td>-4.414</td><td>-6.436<
<tr><td style="text-align:left"></td><td></td><td></td><td>(12.811)</td><td>(12.002)</td></t
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td style="text-align:left">Constant</td><td>918.043<sup>***</sup></td><td>904.721<sup>*
<tr><td style="text-align:left"></td><td>(1.633)</td><td>(2.222)</td><td>(7.653)</td><td>(14
<tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-ali
<tr><td style="text-align:left">R<sup>2</sup></td><td>0.007</td><td>0.020</td><td>0.234</td>
<tr><td style="text-align:left">Adjusted R<sup>2</sup></td><td>0.007</td><td>0.020</td><td>0
<tr><td style="text-align:left">Residual Std. Error</td><td>73.490</td><td>73.085</td><td>65
<tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-ali
</table>
```

We observe that the multiple regression estimates of the effects of both treatments (small class and regular-sized class with an aide) are similar across different models.

This suggests that adding more regressors to the analysis (student characteristics and school fixed effects) doesn't change how these treatments affect the outcome. It makes it more plausible that assigning students to smaller classes is random and not influenced by hidden factors.

As anticipated, including more factors improves the accuracy of the regression model (measured by R^2), and the margin of error for the class size effect decreases from 4.23 in column (1) to 3.95 in column (4).

Since teachers were randomly assigned to different classes within a school, the experiment also allows us to measure how teacher experience impacts test scores in kindergarten, by controlling for school fixed effects as in column (3)

Regression (3) estimates the average effect of 10 years teaching experience to be $10 \cdot 0.74 = 7.4$ points on test scores. Note that the additional estimates regarding student characteristics in regression (4) lack a causal interpretation due to their non-random assignment.

To assess and compare the predicted effects of class size, we must first translate the estimated changes in raw test scores into units of standard deviations of test scores, so that the estimates are comparable across grades.

```
# compute the sample standard deviations of test scores
SSD <- c("K" = sd(na.omit(STAR$readk + STAR$mathk)),
         "1" = sd(na.omit(STAR$read1 + STAR$math1)),
         "2" = sd(na.omit(STAR$read2 + STAR$math2)),
         "3" = sd(na.omit(STAR$read3 + STAR$math3)))

# translate the effects of small classes to standard deviations
Small <- c("K" = as.numeric(coef(fmk)[2]/SSD[1]),
           "1" = as.numeric(coef(fm1)[2]/SSD[2]),
           "2" = as.numeric(coef(fm2)[2]/SSD[3]),
           "3" = as.numeric(coef(fm3)[2]/SSD[4]))

# adjust the standard errors
SmallSE <- c("K" = as.numeric(rob_se_1[[1]][2]/SSD[1]),
             "1" = as.numeric(rob_se_1[[2]][2]/SSD[2]),
             "2" = as.numeric(rob_se_1[[3]][2]/SSD[3]),
             "3" = as.numeric(rob_se_1[[4]][2]/SSD[4]))

# translate the effects of regular classes with aide to standard deviations
RegAide<- c("K" = as.numeric(coef(fmk)[3]/SSD[1]),
            "1" = as.numeric(coef(fm1)[3]/SSD[2]),
            "2" = as.numeric(coef(fm2)[3]/SSD[3]),
            "3" = as.numeric(coef(fm3)[3]/SSD[4]))

# adjust the standard errors
RegAideSE <- c("K" = as.numeric(rob_se_1[[1]][3]/SSD[1]),
               "1" = as.numeric(rob_se_1[[2]][3]/SSD[2]),
               "2" = as.numeric(rob_se_1[[3]][3]/SSD[3]),
               "3" = as.numeric(rob_se_1[[4]][3]/SSD[4]))

# gather the results in a data.frame and round
df <- t(round(data.frame(
                        Small, SmallSE, RegAide, RegAideSE, SSD),
                        digits =  2))

# generate a simple table using stargazer
stargazer(df,
          title = "Estimated Class Size Effects
          (in Units of Standard Deviations)",
```

```
          type = "html",
          summary = FALSE,
          header = FALSE
          )
```

```
<table style="text-align:center"><caption><strong>Estimated Class Size Effect</strong></capti
<tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-alig
<tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-alig
<tr><td style="text-align:left">SmallSE</td><td>0.030</td><td>0.030</td><td>0.030</td><td>0.0
<tr><td style="text-align:left">RegAide</td><td>0</td><td>0.130</td><td>0.040</td><td>0</td>
<tr><td style="text-align:left">RegAideSE</td><td>0.030</td><td>0.030</td><td>0.030</td><td>0
<tr><td style="text-align:left">SSD</td><td>73.750</td><td>91.280</td><td>84.080</td><td>73.2
<tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr></table>
```

In terms of standard deviation units, the estimated impact of being in a small class remains consistent across grades K, 2, and 3, at approximately one-fifth of a standard deviation in test scores.

Similarly, for grades K, 2, and 3, the effect of being in a regular-sized class with an aide is negligible, approximately 0.

Although the treatment effects appear larger for first grade, the contrast between the small class and the regular-sized class with an aide remains consistent at 0.20 for first grade, mirroring the other grades.

One possible explanation for the first-grade results is that students in the control group-those in regular-sized classes without an aide-may have performed poorly on the test that year due to some unusual circumstance, perhaps random sampling variation.

9 Quasi-Experiments

In quasi-experiments, we use "as if" randomness to mimic random assignment. There are two main types:

- When random variations make it seem like the treatment is randomly assigned.
- When the treatment assignment is only partially random.

The first type lets us estimate effects using methods like the difference estimator or differences-in-differences (DID). If there's doubt about systematic differences, we might use an instrumental variable (IV) approach.

For more complex situations, like when treatment depends on a threshold in a continuous variable, we use techniques like sharp regression discontinuity design (RDD) and fuzzy regression discontinuity design (FRDD).

Since there are no empirical examples in this section of the book, we'll explore this section using simulated data in R to explain how DID, RDD, and FRDD work.

9.1 Differences-in-Differences Estimator

The Differences-in-Differences (DID) estimator compares changes in outcomes over time between treated and control groups to estimate the causal effect of an intervention. The DID estimator is

$$\begin{aligned}
\hat{\beta}_1^{\text{diffs-in-diffs}} &= (\overline{Y}^{\text{treatment,after}} - \overline{Y}^{\text{treatment,before}}) - (\overline{Y}^{\text{control,after}} - \overline{Y}^{\text{control,before}}) && (9.1) \\
&= \Delta\overline{Y}^{\text{treatment}} - \Delta\overline{Y}^{\text{control}} && (6.6)
\end{aligned}$$

with

- $\overline{Y}^{\text{treatment,before}}$ - the sample average in the treatment group before the treatment
- $\overline{Y}^{\text{treatment,after}}$ - the sample average in the treatment group after the treatment
- $\overline{Y}^{\text{control,before}}$ - the sample average in the control group before the treatment

- $\overline{Y}^{\text{control,after}}$ - the sample average in the control group after the treatment.

This is always much easier to understand with a graphical representation, so we will reproduce Figure 13.1 of the book by Stock and Watson in R:

```
# initialize plot and add control group
plot(c(0, 1), c(6, 8), type = "p",
     ylim = c(5, 12), xlim = c(-0.3, 1.3),
     main = "The Differences-in-Differences Estimator",
     xlab = "Period", ylab = "Y",
     col = "steelblue", pch = 20, xaxt = "n", yaxt = "n")

axis(1, at = c(0, 1), labels = c("before", "after"))
axis(2, at = c(0, 13))

# add treatment group
points(c(0, 1, 1), c(7, 9, 11), col = "darkred", pch = 20)

# add line segments
lines(c(0, 1), c(7, 11), col = "darkred")
lines(c(0, 1), c(6, 8), col = "steelblue")
lines(c(0, 1), c(7, 9), col = "darkred", lty = 2)
lines(c(1, 1), c(9, 11), col = "black", lty = 2, lwd = 2)

# add annotations
text(1, 10, expression(hat(beta)[1]^{DID}), cex = 0.8, pos = 4)
text(0, 5.5, "s. mean control", cex = 0.8 , pos = 4)
text(0, 6.8, "s. mean treatment", cex = 0.8 , pos = 4)
text(1, 7.9, "s. mean control", cex = 0.8 , pos = 4)
text(1, 11.1, "s. mean treatment", cex = 0.8 , pos = 4)
```

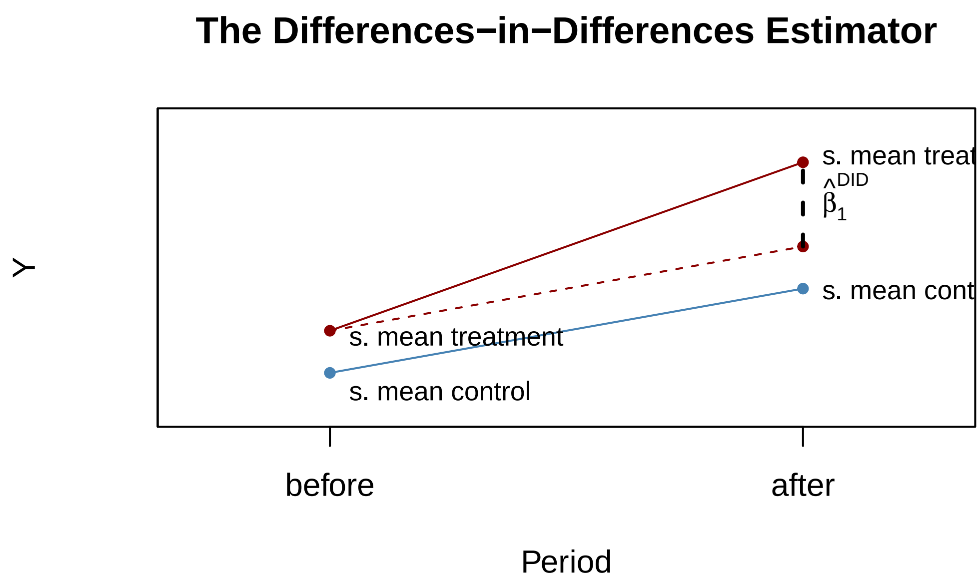
$\widehat{\beta}_1^{\text{DID}}$ is the OLS estimator of β_1 in

$$\Delta Y_i = \beta_0 + \beta_1 X_i + u_i \tag{6.7}$$

where ΔY_i is the difference in pre- and post-treatment outcomes of individual i and X_i is the treatment indicator of interest.

If we add regressors measuring pre-treatment characteristics we have

$$\Delta Y_i = \beta_0 + \beta_1 X_i + \beta_2 W_{1i} + \cdots + \beta_{1+r} W_{ri} + u_i, \tag{6.8}$$



which is the *difference-in-differences estimator* with additional regressors.

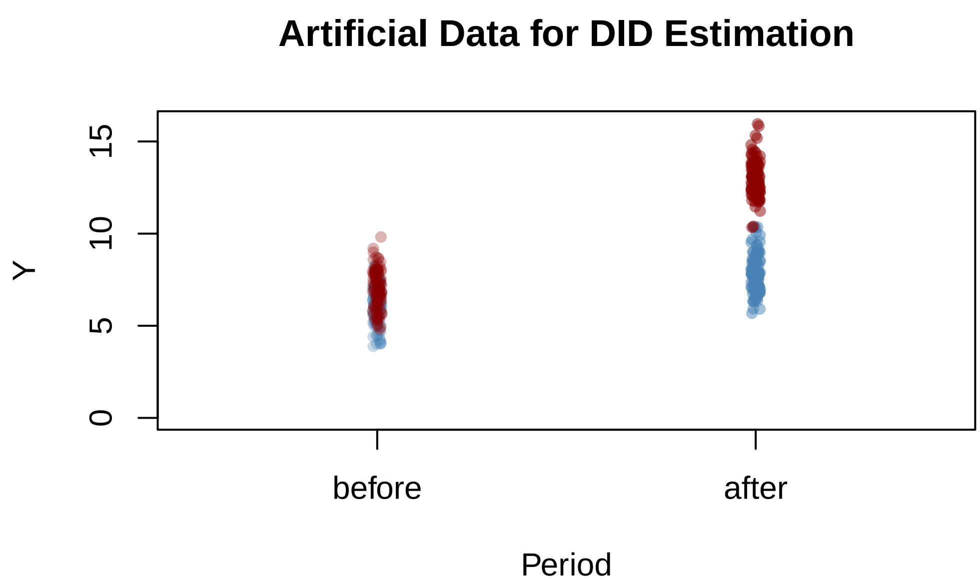
Let's simulate pre- and post-treatment data in R

```
# set sample size
n <- 200

# define treatment effect
TEffect <- 4

# generate treatment dummy
TDummy <- c(rep(0, n/2), rep(1, n/2))

# simulate pre- and post-treatment values of the dependent variable
y_pre <- 7 + rnorm(n)
y_pre[1:n/2] <- y_pre[1:n/2] - 1
y_post <- 7 + 2 + TEffect * TDummy + rnorm(n)
y_post[1:n/2] <- y_post[1:n/2] - 1
```

Now we plot the data. The `jitter()` function adds a bit of randomness to the horizontal positions of points, reducing overlap. Additionally, the `alpha()` function from the package `scales` lets you control how transparent the colors are in your plots.

```
library(scales)

pre <- rep(0, length(y_pre[TDummy==0]))
```

```r
post <- rep(1, length(y_pre[TDummy==0]))

# plot control group in t=1
plot(jitter(pre, 0.6), y_pre[TDummy == 0],
     ylim = c(0, 16), col = alpha("steelblue", 0.3),
     pch = 20, xlim = c(-0.5, 1.5),
     ylab = "Y", xlab = "Period",
     xaxt = "n", main = "Artificial Data for DID Estimation")

axis(1, at = c(0, 1), labels = c("before", "after"))

# add treatment group in t=1
points(jitter(pre, 0.6), y_pre[TDummy == 1],
       col = alpha("darkred", 0.3), pch = 20)

# add control group in t=2
points(jitter(post, 0.6), y_post[TDummy == 0],
       col = alpha("steelblue", 0.5), pch = 20)

# add treatment group in t=2
points(jitter(post, 0.6), y_post[TDummy == 1],
       col = alpha("darkred", 0.5), pch = 20)
```



We observe higher average values for both groups after treatment, with a more pronounced increase observed in the treatment group. By employing the Differences-in-Differences (DID) method, we can assess the extent to which this disparity can be attributed to the treatment itself.

```
# compute the DID estimator for the treatment effect 'by hand'
mean(y_post[TDummy == 1]) - mean(y_pre[TDummy == 1]) -
(mean(y_post[TDummy == 0]) - mean(y_pre[TDummy == 0]))
```

```
[1] 4.250925
```

The reported estimate is close to 4, the treatment effect value we previously selected for `TEffect`.

We can also obtain the DID estimator by performing OLS estimation of the simple linar model (6.7).

```
# compute the DID estimator using a linear model
lm(I(y_post - y_pre) ~ TDummy)
```

```

Call:
lm(formula = I(y_post - y_pre) ~ TDummy)

Coefficients:
(Intercept)       TDummy
      1.753        4.251
```

Lastly, we could alternatively compute the treatment effect by estimating β_{TE} in

$$Y_i = \beta_0 + \beta_1 D_i + \beta_2 Period_i + \beta_{TE}(Period_i \times D_i) + \epsilon_i, \tag{6.9}$$

where D_i is the binary treatment indicator, $Period$ a binary indicator for the after-treatment period and $Period_i \times D_i$ is the interaction term of both.

```
# prepare data for DID regression using the interaction term
d <- data.frame("Y" = c(y_pre,y_post),
                "Treatment" = TDummy,
                "Period" = c(rep("1", n), rep("2", n)))

# estimate the model
lm(Y ~ Treatment * Period, data = d)
```

```
Call:
lm(formula = Y ~ Treatment * Period, data = d)

Coefficients:
      (Intercept)          Treatment            Period2  Treatment:Period2
           6.1330             0.8881             1.7533             4.2509
```

As we can see, the estimated coefficient on the interaction term is again the same DID estimate we computed before.

9.2 Regression Discontinuity Estimators

9.2.1 Sharp Regression Discontinuity

Let's consider the model

$$Y_i = \beta_0 + \beta_1 X_1 + \beta_2 W_i + u_i \tag{6.10}$$

and let

$$X_i = \begin{cases} 1, & \text{if } W_i \geq c \\ 0, & \text{if } W_i < c, \end{cases}$$

so that the treatment receipt represented by X_i depends on a certain threshold c of a continuous variable W_i, known as the running variable.

We call (6.10)) a *sharp regression discontinuity design* because the treatment assignment is deterministic and continuous at the threshold: all observations with $W_i \geq c$ are treated and those with $W_i < c$ do not receive treatment.

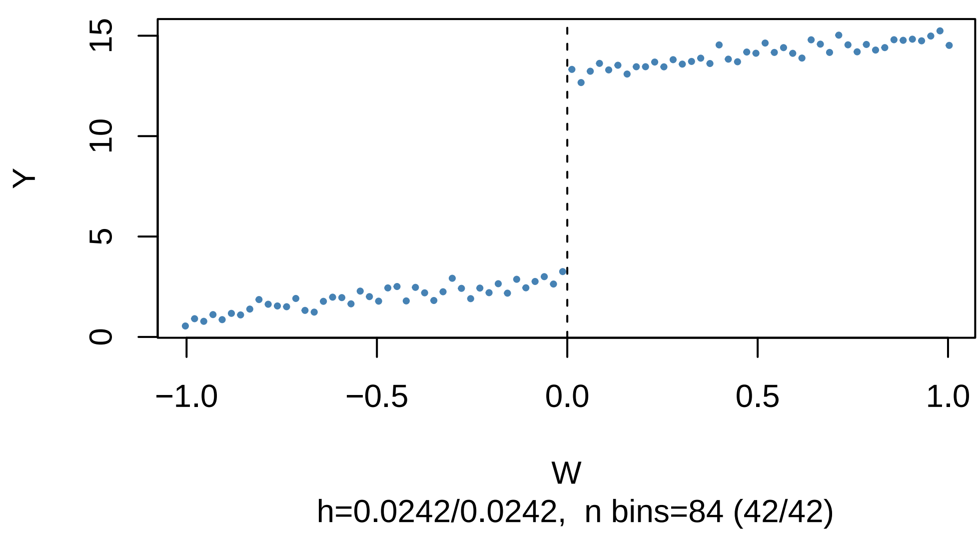
The idea of regression discontinuity design is to use observations with a W_i close to c for the estimation of β_1, which is the average treatment effect for individuals with $W_i = c$, and is assumed to be a good approximation to the treatment effect in the population.

We will now estimate a linear SRDD, but first, we generate and plot some sample data

```r
# generate some sample data
W <- runif(1000, -1, 1)
y <- 3 + 2 * W + 10 * (W>=0) + rnorm(1000)
```

```r
# load the package 'rddtools'
library(rddtools)

# construct rdd_data
data <- rdd_data(y, W, cutpoint = 0)

# plot the sample data
plot(data,
     col = "steelblue",
     cex = 0.35,
     xlab = "W",
     ylab = "Y")
```

```
Warning in title(main = main, sub = sub): Zeichenbreite unbekannt für das
Zeichen 0x9

Warning in title(main = main, sub = sub): Zeichenbreite unbekannt für das
Zeichen 0x9
```



The dots in the plot represent bin averages of the outcome variable.

To estimate the treatment effect using model (6.10) on our generated data we can use `rdd_reg_lm()` from the `rddtools` package. By setting `slope ="same"` we ensure that the slopes of the regression function stay consistent on both sides of the threshold $W = 0$.

```
# estimate the sharp RDD model
rdd_mod <- rdd_reg_lm(rdd_object = data,
                      slope = "same")
summary(rdd_mod)
```

```
Call:
lm(formula = y ~ ., data = dat_step1, weights = weights)

Residuals:
     Min       1Q   Median       3Q      Max
-2.78536 -0.68390 -0.02412  0.66245  2.55417

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.9237     0.0683   42.81   <2e-16 ***
D            10.1821     0.1220   83.43   <2e-16 ***
x             1.9121     0.1042   18.36   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9645 on 997 degrees of freedom
Multiple R-squared:  0.9756,    Adjusted R-squared:  0.9755
F-statistic: 1.991e+04 on 2 and 997 DF,  p-value: < 2.2e-16
```

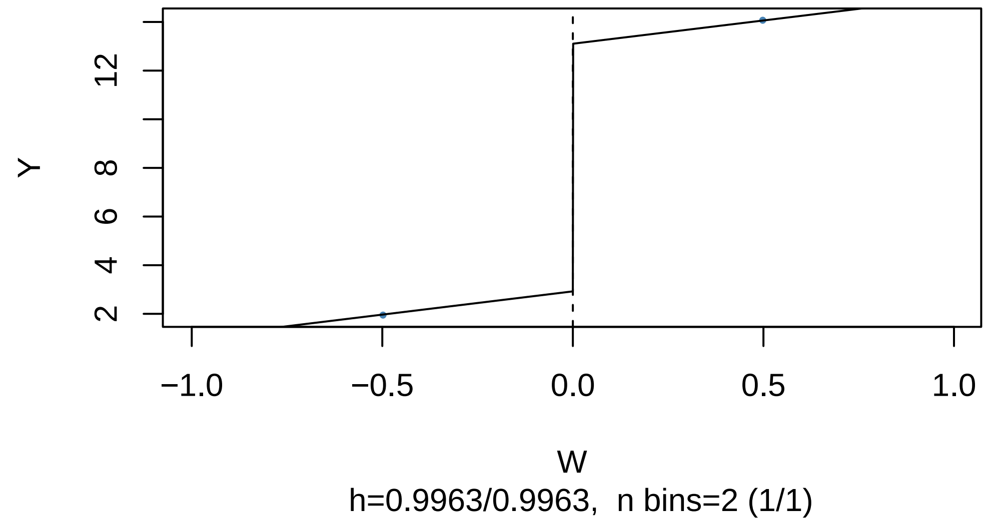
The estimated coefficient on D represents the estimated treatment effect, which is very close to 10, the treatment effect we chose when generating the simulated data.

Let's now visualize the result by plotting the estimated sharp RDD model

```
# plot the RDD model along with binned observations
plot(rdd_mod,
     cex = 0.35,
     col = "steelblue",
     xlab = "W",
     ylab = "Y")
```

```
Warning in title(main = main, sub = sub): Zeichenbreite unbekannt für das
```

```
Zeichen 0x9

Warning in title(main = main, sub = sub): Zeichenbreite unbekannt für das
Zeichen 0x9
```



9.2.2 Fuzzy Regression Discontinuity

In the traditional setup, we assumed that crossing a threshold automatically leads to treatment, allowing us to see the jump in population regression functions at that point as the treatment's effect.

However, when crossing the threshold doesn't guarantee treatment (e.g. when other factors also influence who gets treated) we can't rely on this assumption. Instead, we can view the threshold as a point where the likelihood of getting treated suddenly increases.

This increase might happen because of hidden factors affecting the chance of getting treated. So, the treatment variable X_i in the equation becomes correlated to the error term u_i, making it harder to accurately estimate the treatment's effect.

In such cases, a fuzzy regression discontinuity design, which uses an instrumental variable (IV) approach, might help. We can use a binary variable Z_i to indicate whether the threshold is crossed or not.

$$Z_i = \begin{cases} 1, & \text{if } W_i \geq c \\ 0, & \text{if } W_i < c, \end{cases}$$

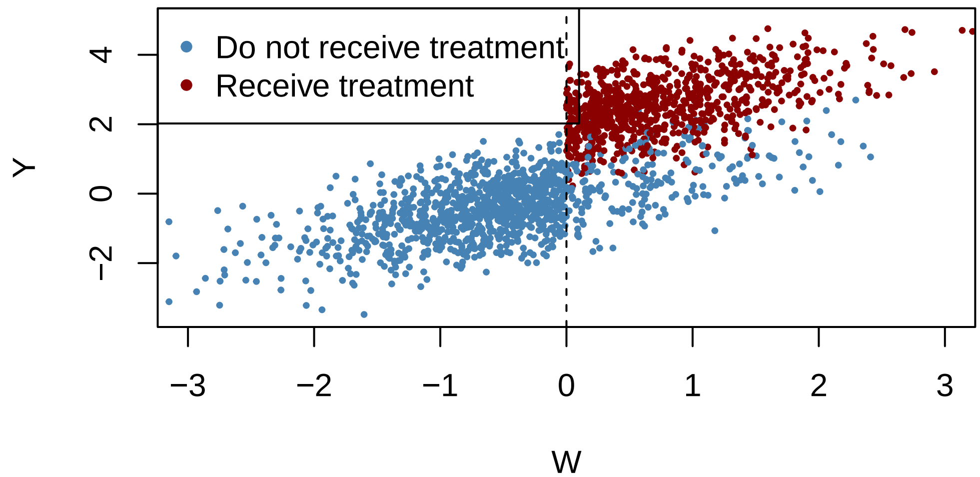
We assume that Z_i relates to Y_i only through the treatment indicator X_i, so Z_i and u_i are uncorrelated but Z_i influences the receipt of the treatment, so it is correlated with X_i. Therefore, Z_i is a valid instrument for X_i and we can estimate (6.10) via TSLS.

Let's now assume that observations with a value of W_i below 0 do not receive the treatment and those with $W_i \geq 0$ have a 80% probability of being treated. The treatment effect leads to an increase in the dependent variable of 2 points.

```
library(MASS)

# generate sample data
mu <- c(0, 0)
sigma <- matrix(c(1, 0.7, 0.7, 1), ncol = 2)

set.seed(1234)
d <- as.data.frame(mvrnorm(2000, mu, sigma))
colnames(d) <- c("W", "Y")

# introduce fuzziness
d$treatProb <- ifelse(d$W < 0, 0, 0.8)

fuzz <- sapply(X = d$treatProb, FUN = function(x) rbinom(1, 1, prob = x))

# treatment effect
d$Y <- d$Y + fuzz * 2
```

We now plot the observations using blue for non-treated and red for treated units.

```
# generate a colored plot of treatment and control group
plot(d$W, d$Y,
     col = c("steelblue", "darkred")[factor(fuzz)],
     pch= 20,
     cex = 0.5,
     xlim = c(-3, 3),
     ylim = c(-3.5, 5),
     xlab = "W",
     ylab = "Y")

# add a dashed vertical line at cutoff
abline(v = 0, lty = 2)
#add legend
legend("topleft",pch=20,col=c("steelblue","darkred"),
       legend=c("Do not receive treatment","Receive treatment"))
```



As we can observe, the receipt of treatment is no longer a deterministic function of the running variable W, since some observations with $W \geq 0$ did not receive the treatment.

We can estimate a FRDD by setting `treatProb` as the assignment variable z in `rdd_data()`. The function `rdd_reg_lm()` applies a TSLS procedure:

1. In the first stage regression, treatment is predicted using W_i and the cutoff dummy Z_i, the instrumental variable.
2. Using the second stage, where the outcome Y is regressed on the fitted values and the running variable W, we obtain a consistent estimate of the treatment effect.

```
# estimate the Fuzzy RDD
data <- rdd_data(d$Y, d$W,
                 cutpoint = 0,
                 z = d$treatProb)

frdd_mod <- rdd_reg_lm(rdd_object = data,
                       slope = "same")
frdd_mod
```

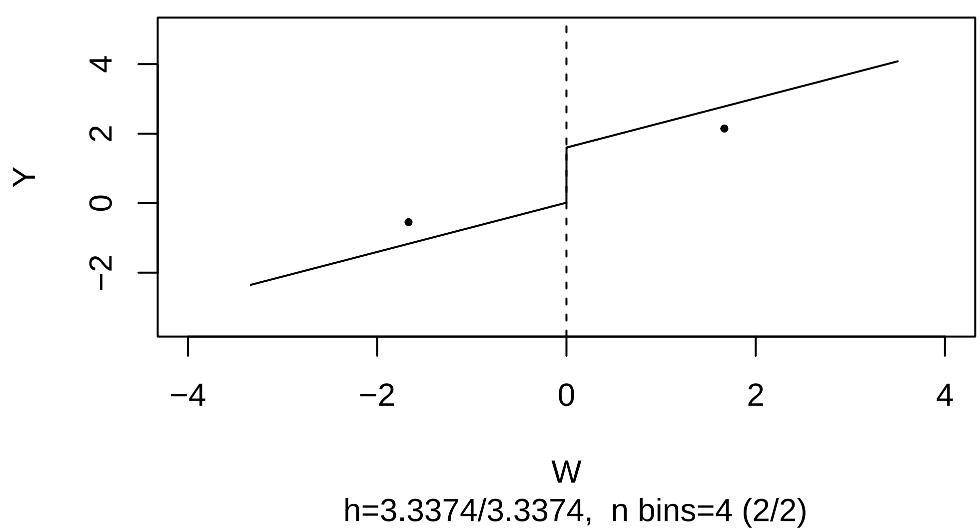
```
### RDD regression: parametric ###
    Polynomial order:  1
    Slopes:  same
    Number of obs: 2000 (left: 999, right: 1001)

    Coefficient:
  Estimate Std. Error t value  Pr(>|t|)
D 1.981297   0.084696  23.393 < 2.2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated treatment effect is very close to 2, which is the real treatment effect. We can now plot the estimated regression function and the binned data.

```r
# plot estimated FRDD function
plot(frdd_mod,
     cex = 0.5,
     lwd = 0.4,
     xlim = c(-4, 4),
     ylim = c(-3.5, 5),
     xlab = "W",
     ylab = "Y")
```

```
Warning in title(main = main, sub = sub): Zeichenbreite unbekannt für das
Zeichen 0x9

Warning in title(main = main, sub = sub): Zeichenbreite unbekannt für das
Zeichen 0x9
```



What if we opted for a Sharp Regression Discontinuity Design (SRDD), disregarding the fact that treatment isn't solely determined by the cutoff in W? We can explore the potential outcomes by estimating an SRDD using the data we simulated earlier.

```r
# estimate SRDD
data <- rdd_data(d$Y,
                 d$W,
                 cutpoint = 0)

srdd_mod <- rdd_reg_lm(rdd_object = data,
                       slope = "same")
srdd_mod
```

```
### RDD regression: parametric ###
    Polynomial order:  1
    Slopes:  same
    Number of obs: 2000 (left: 999, right: 1001)

    Coefficient:
  Estimate Std. Error t value  Pr(>|t|)
D 1.585038   0.067756  23.393 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimate using SRDD indicates a significant downward bias. This method is not reliable for determining the true causal effect, meaning that increasing the sample size wouldn't fix the bias issue.

9.3 Discussion

In the book by Stock and Watson the potential **problems with quasi-experiments** are discussed, focusing on threats to internal and external validity.

Internal validity threats include failure of randomization and failure to follow the treatment protocol, which can lead to biased estimators. They highlight the importance of testing for systematic differences between treatment and control groups to assess the reliability of quasi-experiments.

Additionally, they address attrition and instrument validity, emphasizing the need for careful consideration of instrument relevance and exogeneity.

External validity threats in quasi-experiments are similar to those in conventional regression studies, with special events creating challenges for generalizability.

Lastly, Stock and Watson discuss estimating **causal effects in heterogeneous populations**, where individuals may have different treatment effects; OLS estimators are consistent for the average causal effect, but instrumental variables (IV) estimators may estimate a weighted average of individual effects, known as the local average treatment effect (LATE), highlighting the importance of understanding how individuals' treatment decisions affect estimation.

10 Empirical Applications of Time Series Regression and Forecasting

In this chapter, we will explore the concepts of Time Series Regression and Forecasting. It will introduce you to the basic techniques for analyzing time series data, focusing on visualizing data, estimating autoregressive models, and understanding the concept of stationarity.

We will use empirical examples, primarily involving U.S. macroeconomic indicators and financial time series such as GDP, unemployment rates, and stock returns, to illustrate these concepts.

```
library(AER)
library(dynlm)
library(forecast)
library(readxl)
library(stargazer)
library(scales)
library(quantmod)
library(urca)
```

10.1 Data Set Description

The dataset us_macro_quarterly.xlsx contains quarterly data on U.S. real GDP (inflation-adjusted) from 1947 to 2004.

The first column contains text, while the remaining columns are numeric. We can specify the column types by using `col_types = c("text", rep("numeric", 9))` when reading the data.

```
# load US macroeconomic data
USMacroSWQ <- read_xlsx("us_macro_quarterly.xlsx",
                         sheet = 1,
                         col_types = c("text", rep("numeric", 9)))

# format date column
```

```
USMacroSWQ$...1 <- as.yearqtr(USMacroSWQ$...1, format = "%Y:0%q")

# adjust column names
colnames(USMacroSWQ) <- c("Date", "GDPC96", "JAPAN_IP", "PCECTPI",
                          "GS10", "GS1", "TB3MS", "UNRATE", "EXUSUK", "CPIAUCSL")
```

10.2 Time Series Data and Serial Correlation

Working with time-series objects that track the frequency of the data and are extensible is useful for an effective time series analysis. We will use objects of the class `xts` for this purpose, which have a time-based ordered index. See `?xts`.

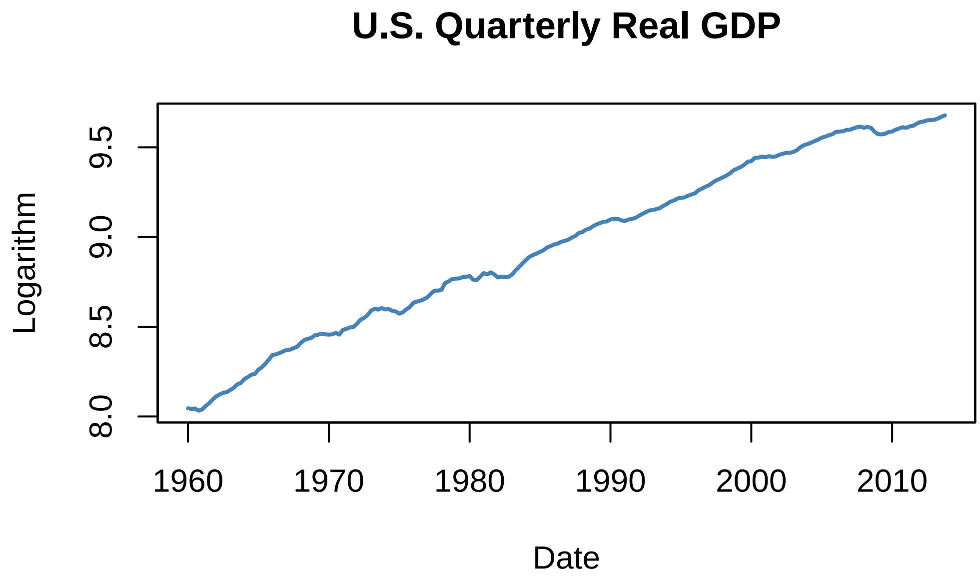
The data in `USMacroSWQ` are in quarterly frequency, so we convert the first column to `yearqtr` format before generating the `xts` object GDP.

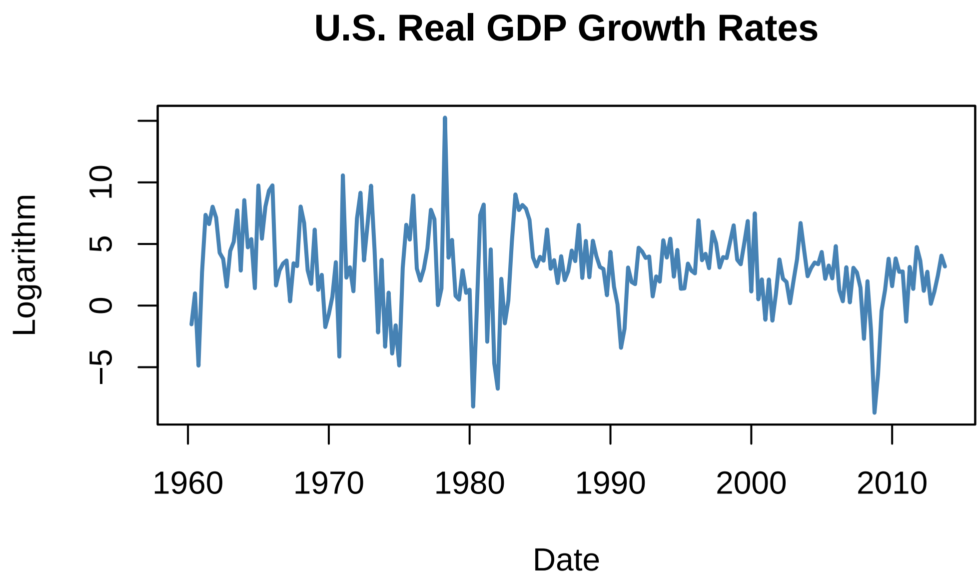
```
# GDP series as xts object
GDP <- xts(USMacroSWQ$GDPC96, USMacroSWQ$Date)["1960::2013"]

# GDP growth series as xts object
GDPGrowth <- xts(400 * log(GDP/lag(GDP)))
```

As with any data analysis, a good starting point is to visualize the data. For this purpose, we will use the `quantmod` package, which offers convenient functions for plotting and computing with time series data.

```
# GDP in log scale
plot(log(as.zoo(GDP)),
     col = "steelblue",
     lwd = 2,
     ylab = "Logarithm",
     xlab = "Date",
     main = "U.S. Quarterly Real GDP")
```

```
# Growth rate of GDP
plot(as.zoo(GDPGrowth),
     col = "steelblue",
     lwd = 2,
     ylab = "Logarithm",
     xlab = "Date",
     main = "U.S. Real GDP Growth Rates")
```





10.3 Lags, Differences, Logarithms and Growth Rates

For observations of a variable Y recorded over time, Y_t represents the value observed at time t. The interval between two consecutive observations, Y_{t-1} and Y_t, defines a unit of time (e.g. hours, days, weeks, months, quarters or years).

Previous values of a time series are called lags. The first lag of Y_t is Y_{t-1}. The j^{th} lag of Y_t is Y_{t-j}. In R, lags of univariate or multivariate time series objects are computed by `lag()` (see `?lag`).

It is sometimes convenient to work with a differenced series. The first difference of a series is $\Delta Y_t = Y_t - Y_{t-1}$. For a time series Y, we compute the series of first differences as `diff(Y)` in R.

Since it is common to report growth rates in macroeconomic series, it is convenient to work with the first difference in logarithms of a series, denoted by $\Delta \log(Y_t) = \log(Y_t) - \log(Y_{t-1})$. We can obtain this in R by using `log(Y/lag(Y))`.

We may additionally approximate the percentage change between Y_t and Y_{t-1} as $100\Delta \log(Y_t)$.

We can now present the quarterly U.S. GDP time series, its logarithm, the annualized growth rate and the first lag of the annualized growth rate series for the period 2012:Q1 - 2013:Q1. The following function `quants` can be used to compute these quantities for a quarterly time series.

```
# compute logarithms, annual growth rates and 1st lag of growth rates
quants <- function(series) {
  s <- series
  return(
    data.frame("Level" = s,
               "Logarithm" = log(s),
               "AnnualGrowthRate" = 400 * log(s / lag(s)),
               "1stLagAnnualGrowthRate" = lag(400 * log(s / lag(s))))
    )
}
```

Since $100\Delta \log(Y_t)$ is an approximation of the quarterly percentage changes, we compute the annual growth rate using the approximation

$$\text{AnnualGrowth}Y_t = 400 \cdot \Delta \log(Y_t)$$

We may now call `quants()` on observations for the period 2011:Q3 - 2013:Q1.

```
# obtain a data.frame with level, logarithm, annual growth rate and its 1st lag of GDP
quants(GDP["2011-07::2013-01"])
```

```
           Level Logarithm AnnualGrowthRate X1stLagAnnualGrowthRate
2011 Q3 15062.14  9.619940               NA                      NA
2011 Q4 15242.14  9.631819        4.7518062                      NA
2012 Q1 15381.56  9.640925        3.6422231               4.7518062
2012 Q2 15427.67  9.643918        1.1972004               3.6422231
2012 Q3 15533.99  9.650785        2.7470216               1.1972004
```

```
2012 Q4 15539.63  9.651149        0.1452808                2.7470216
2013 Q1 15583.95  9.653997        1.1392015                0.1452808
```

10.4 Autocorrelation

Observations of a time series are typically correlated. This is called *autocorrelation* or *serial correlation*.

We can compute the first four sample autocorrelations of the series `GDPGrowth` using `acf()`.

```
acf(na.omit(GDPGrowth), lag.max = 4, plot = F)
```

```
Autocorrelations of series 'na.omit(GDPGrowth)', by lag

 0.00  0.25  0.50  0.75  1.00
1.000 0.352 0.273 0.114 0.106
```

These values suggest a mild positive autocorrelation in GDP growth: when GDP grows faster than average in one period, it tends to continue growing faster than average in subsequent periods.

10.5 Additional Examples of Economic Time Series

The book by Stock and Watson (2020, Global Edition) presents four plots in figure 15.2: the U.S. unemployment rate, the U.S. Dollar / British Pound exchange rate, the logarithm of the Japanese industrial production index and daily changes in the Wilshire 5000 stock price index, a financial time series.

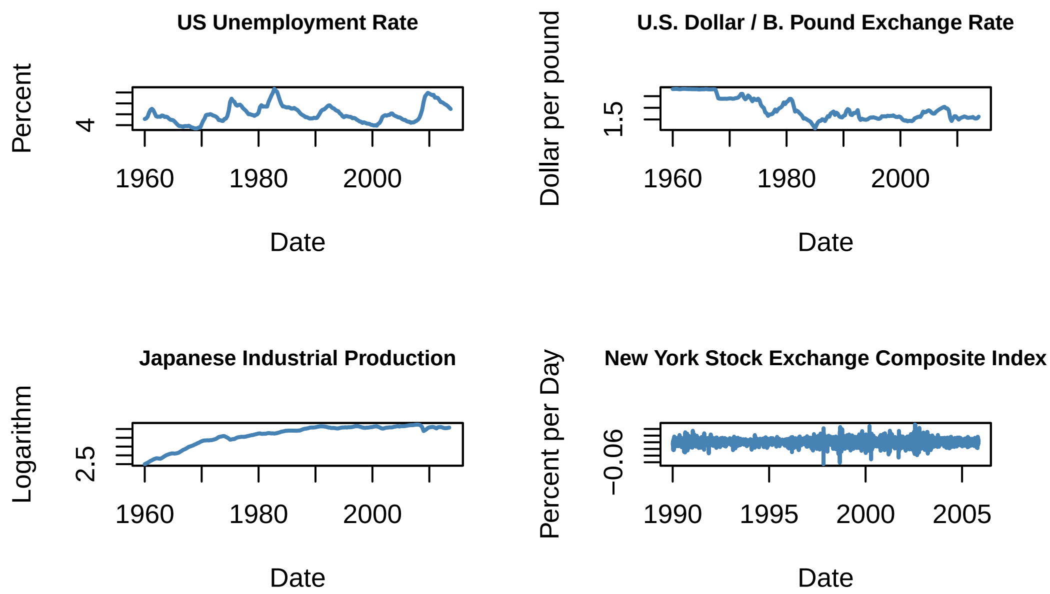
To reproduce these plots, we additionally use the data set `NYSESW` included in the `AER` package. We now plot the three macroeconomic series and add percentage changes in the daily values of the New York Stock Exchange Composite index as a fourth plot.

```
# define series as xts objects
USUnemp <- xts(USMacroSWQ$UNRATE, USMacroSWQ$Date)["1960::2013"]

DollarPoundFX <- xts(USMacroSWQ$EXUSUK, USMacroSWQ$Date)["1960::2013"]

JPIndProd <- xts(log(USMacroSWQ$JAPAN_IP), USMacroSWQ$Date)["1960::2013"]
```

```r
# attach NYSESW data
data("NYSESW")
NYSESW <- xts(Delt(NYSESW))
```

```r
# divide plotting area into 2x2 matrix
par(mfrow = c(2, 2))

# plot the series
plot(as.zoo(USUnemp),
     col = "steelblue",
     lwd = 2,
     ylab = "Percent",
     xlab = "Date",
     main = "US Unemployment Rate",
     cex.main = 0.8)

plot(as.zoo(DollarPoundFX),
     col = "steelblue",
     lwd = 2,
     ylab = "Dollar per pound",
     xlab = "Date",
     main = "U.S. Dollar / B. Pound Exchange Rate",
     cex.main = 0.8)

plot(as.zoo(JPIndProd),
     col = "steelblue",
     lwd = 2,
     ylab = "Logarithm",
     xlab = "Date",
     main = "Japanese Industrial Production",
     cex.main = 0.8)

plot(as.zoo(NYSESW),
     col = "steelblue",
     lwd = 2,
     ylab = "Percent per Day",
     xlab = "Date",
     main = "New York Stock Exchange Composite Index",
     cex.main = 0.8)
```

We observe different characteristics in the series:



- The unemployment rate rises during recessions and falls during periods of economic recovery and growth.
- The Dollar/Pound exchange rate followed a deterministic pattern until the Bretton Woods system ended.
- Japan's industrial production shows an upward trend with diminishing growth.
- Daily changes in the New York Stock Exchange composite index appear to fluctuate randomly around zero. The sample autocorrelations support this observation.

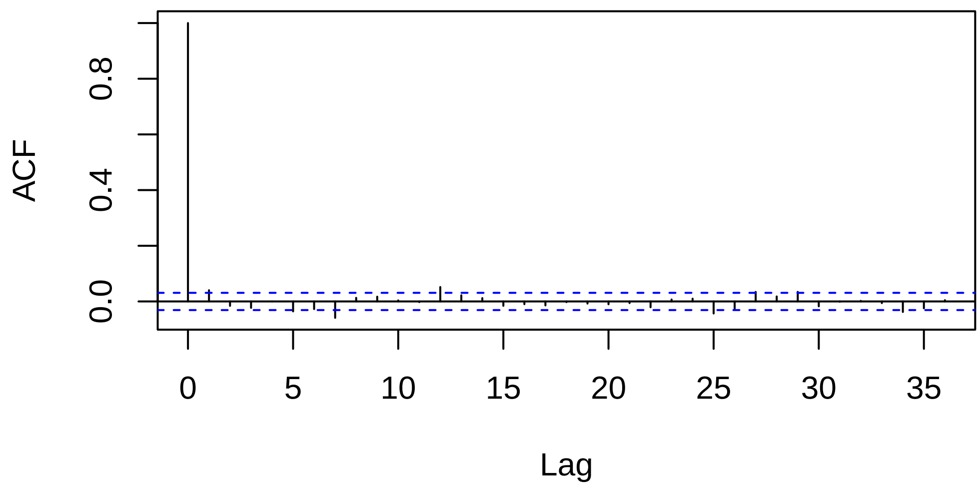
```
# compute sample autocorrelation for the NYSESW series
acf(na.omit(NYSESW), plot = F, lag.max = 10)
```

```
Autocorrelations of series 'na.omit(NYSESW)', by lag

     0      1      2      3      4      5      6      7      8      9     10
 1.000  0.040 -0.016 -0.023  0.000 -0.036 -0.027 -0.059  0.013  0.017  0.004
```

The first 10 sample autocorrelation coefficients are nearly zero. The default plot produced by `acf()` offers additional confirmation.

```
# plot sample autocorrelation for the NYSESW series
acf(na.omit(NYSESW), main = "Sample Autocorrelation for NYSESW Data")
```

Sample Autocorrelation for NYSESW Data



The blue dashed bands represent values beyond which the autocorrelations are significantly different from zero at 5% level. For most lags, the sample autocorrelation remains within the bands, with only a few instances slightly exceeding the limits.

Additionally, the NYSESW series show *volatility clustering*, characterized by periods of high and low variance. This pattern is typically observed in many financial time series.

10.6 Autoregressions

10.6.1 The First-Order Autoregressive Model

The simplest autoregressive model uses only the most recent outcome of the time series observed to predict future values. For a time series Y_t, this model is known as a first-order autoregressive model, commonly abbreviated as AR(1).

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + u_t$$

is the AR(1) population model of a time series Y_t.

The first-order autoregression model of GDP growth can be estimated by computing OLS estimates in the regression of $GDPGR_t$ on $GDPGR_{t-1}$

$$\widehat{GDPGR}_t = \hat{\beta}_0 + \hat{\beta}_1 GDPGR_{t-1}$$

To estimate this regression model, we use data from 1962 to 2012 and we use `ar.ols()` from the package `stats`.

```
# subset data
GDPGRSub <- GDPGrowth["1962::2012"]

# estimate the model
ar.ols(GDPGRSub,
       order.max = 1,
       demean = F,
       intercept = T)
```

```
Call:
ar.ols(x = GDPGRSub, order.max = 1, demean = F, intercept = T)

Coefficients:
     1
0.3384

Intercept: 1.995 (0.2993)

Order selected 1  sigma^2 estimated as  9.886
```

We see that the computations done by `ar.ols()` are the same as done by `lm()`.

```
# length of data set
N <-length(GDPGRSub)

GDPGR_level <- as.numeric(GDPGRSub[-1])
GDPGR_lags <- as.numeric(GDPGRSub[-N])

# estimate the model
armod <- lm(GDPGR_level ~ GDPGR_lags)
armod
```

```
Call:
lm(formula = GDPGR_level ~ GDPGR_lags)

Coefficients:
(Intercept)   GDPGR_lags
     1.9950       0.3384
```

We obtain a robust summary on the estimated regression coefficients as usual with `coeftest()`.

```
# robust summary
coeftest(armod, vcov. = vcovHC, type = "HC1")
```

```
t test of coefficients:

            Estimate Std. Error t value  Pr(>|t|)
(Intercept) 1.994986   0.351274  5.6793 4.691e-08 ***
GDPGR_lags  0.338436   0.076188  4.4421 1.470e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated regression model is

$$\widehat{GDPGR}_t = \underset{(0.351)}{1.995} + \underset{(0.076)}{0.338}\, GDPGR_{t-1}.$$

We omit the first observation for $GDPGR_{1962\,Q1}$ from the vector of the dependent variable since $GDPGR_{1962\,Q1-1} = GDPGR_{1961\,Q4}$ is not included in the sample. Similarly, the last observation, $GDPGR_{2012\,Q4}$ is excluded from the predictor vector since the data does not include $GDPGR_{2012\,Q4+1} = GDPGR_{2013\,Q1}$.

Put differently, when estimating the model, one observation is lost because of the time series structure of the data.

10.6.2 Forecasts and Forecast Errors

When Y_t follows an AR(1) model with an intercept and we have an OLS estimate of the model on the basis of observations for T periods, then we may use the AR(1) model to obtain $\widehat{Y}_{T+1|T}$, a forecast for Y_{T+1} using data up to period T, where

$$\widehat{Y}_{T+1|T} = \hat{\beta}_0 + \hat{\beta}_1 Y_T.$$

The forecast error is then

$$\text{Forecast error} = Y_{T+1} - \widehat{Y}_{T+1|T}$$

10.6.3 Forecasts and Forecasted Values

Forecasted values of Y_t are different from what we call OLS predicted values of Y_t. Additionally, the forecast error differs from an OLS residual. Forecasts and forecast errors are derived using out-of-sample data, whereas predicted values and residuals are calculated using in-sample data that has been observed.

The root mean squared forecast error (RMSFE) quantifies the typical magnitude of the forecast error and is defined as

$$\text{RMSFE} = \sqrt{E\left[\left(Y_{T+1} - \widehat{Y}_{T+1|T}\right)^2\right]}$$

The RMSFE consists of future errors u_t and the error derived from estimating the coefficients. When the sample size is large, the future errors often dominate, making RMSFE approximately equal to $\sqrt{\text{Var}(u_t)}$, which can be estimated by the standard error of the regression.

10.6.4 Application to GDP Growth

Using the estimated AR(1) model of GDP growth, we can perform the forecast for the GDP growth for the first quarter of 2013. Since we estimated the model using data from 1962:Q1 to 2012:Q4, 2013:Q1 is an out-of-sample period.

Substituting $GDPGR_{2012\,Q4} \approx 0.15$ into the equation, we obtain:

$$\widehat{GDPGR}_{2013:Q1} = 1.995 + 0.348 \cdot 0.15 = 2.047$$

The `forecast()` function from the forecast package provides useful features for making time series predictions.

```r
# assign GDP growth rate in 2012:Q4
new <- data.frame("GDPGR_lags" = GDPGR_level[N-1])

# forecast GDP growth rate in 2013:Q1
forecast(armod, newdata = new)
```

```
  Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
1       2.044155 -2.036225 6.124534 -4.213414 8.301723
```

We observe the same point forecast of approximately 2.0, together with the 80% and 95% forecast intervals.

We conclude that the AR(1) model predicts the GDP growth to be 2% in 2013:Q1.

But how reliable is this forecast? The forecast error is substantial: $GDPGR_{2013\,Q1} \approx 1.1\%$, while our prediction is 2%. Additionally, using `summary(armod)` reveals that the model accounts for only a small portion of the GDP growth rate variation, with the SER around 3.16. Ignoring the forecast uncertainty from estimating coefficients β_0 and β_1, the RMSFE should be at least 3.16%, which is the estimated standard deviation of the errors. Thus, we conclude that this forecast is quite inaccurate.

```r
# compute the forecast error
forecast(armod, newdata = new)$mean - GDPGrowth["2013"][1]
```

```
                x
2013 Q1 0.9049532
```

```r
# R^2
summary(armod)$r.squared
```

```
[1] 0.1149576
```

```r
# SER
summary(armod)$sigma
```

```
[1] 3.15979
```

10.6.5 Autoregressive Models of Order p

The AR(1) model only considers information from the most recent period to forecast GDP growth. In contrast, an AR(p) model includes information from the past p lags of the series.

An AR(p) model assumes that a time series Y_t can be modeled by a linear function of the first p of its lagged values.

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \dots + \beta_p Y_{t-p} + u_t$$

is an autoregressive model of order p where $E(u_t \mid Y_{t-1}, Y_{t-2}, \dots, Y_{t-p}) = 0$.

Let's estimate an AR(2) model of the GDP growth series from 1962:Q1 to 2012:Q4.

```r
# estimate the AR(2) model
GDPGR_AR2 <- dynlm(ts(GDPGR_level) ~ L(ts(GDPGR_level)) + L(ts(GDPGR_level), 2))

coeftest(GDPGR_AR2, vcov. = sandwich)
```

```

t test of coefficients:

                      Estimate Std. Error t value  Pr(>|t|)
(Intercept)           1.631747   0.402023  4.0588 7.096e-05 ***
L(ts(GDPGR_level))    0.277787   0.079250  3.5052 0.0005643 ***
L(ts(GDPGR_level), 2) 0.179269   0.079951  2.2422 0.0260560 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We obtain

$$\widehat{GDPGR}_t = \underset{(0.40)}{1.63} + \underset{(0.08)}{0.28}\, GDPGR_{t-1} + \underset{(0.08)}{0.18}\, GDPGR_{t-2}$$

We see that the coefficient on the second lag is significantly different from zero at the 5% level. Compared to the AR(1) model, the fit shows a slight improvement: $\bar{R}^2$ increases from 0.11 in the AR(1) model to about 0.14 in the AR(2), and the SER decreases to 3.13.

```r
# R^2
summary(GDPGR_AR2)$r.squared
```

```
[1] 0.1425484
```

```r
# SER
summary(GDPGR_AR2)$sigma
```

```
[1] 3.132122
```

We can use the AR(2) model to forecast GDP growth for 2013 in the same way as we did with the AR(1) model.

```r
# AR(2) forecast of GDP growth in 2013:Q1
forecast <- c("2013:Q1" = coef(GDPGR_AR2) %*% c(1, GDPGR_level[N-1],
                                                GDPGR_level[N-2]))
forecast
```

```
2013:Q1
2.16456
```

The forecast error is approximately -1%.

```r
# compute AR(2) forecast error
GDPGrowth["2013"][1] - forecast
```

```
                x
2013 Q1 -1.025358
```

10.7 Additional Predictors and The ADL Model

An **autoregressive distributed lag (ADL)** model is called *autoregressive* because it includes lagged values of the dependent variable as regressors (similar to an autoregression), but it's also termed a *distributed lag model* because the regression incorporates multiple lags (a "distributed lag") of an additional predictor.

The autoregressive distributed lag model with p lags of Y_t and q lags of X_t, denoted ADL(p,q), is

$$\begin{aligned} Y_t = &\beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_p Y_{t-p} \\ &+ \delta_1 X_{t-1} + \delta_2 X_{t-2} + \cdots + \delta_q X_{t-q} + u_t \end{aligned}$$

where $\beta_0, \beta_1, \dots, \beta_p, \delta_1, \dots, \delta_q$, are unknown coefficients and u_t is the error term with $E(u_t \mid Y_{t-1}, Y_{t-2}, \dots, X_{t-1}, X_{t-2}, \dots) = 0$.

10.7.1 Forecasting GDP Growth Using the Term Spread

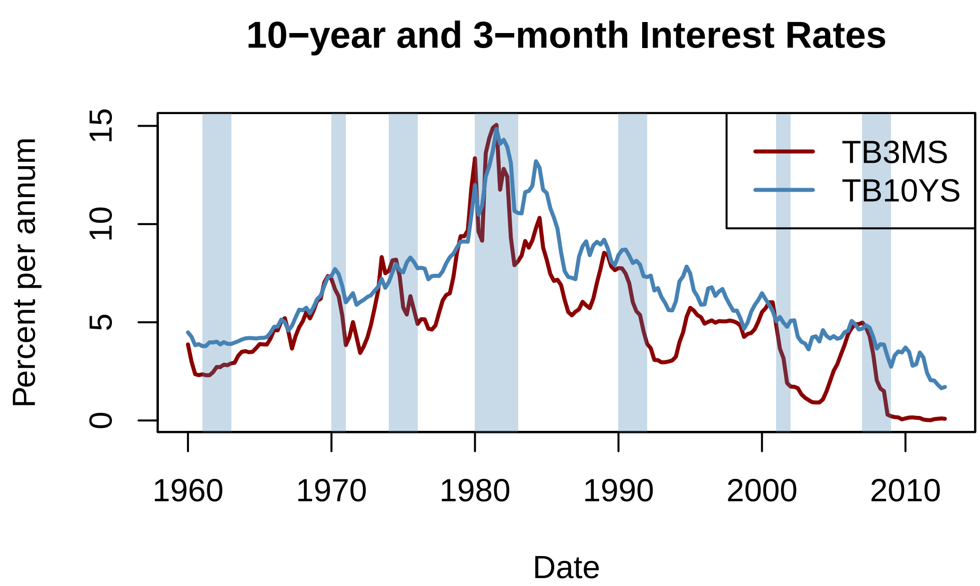
Interest rates on long-term and short-term treasury bonds are closely tied to macroeconomic conditions. Although both types of bonds share similar long-term trends, their short-term behaviors differ significantly. The disparity in interest rates between two bonds with different maturities is known as the term spread.

Figure 15.3 of the book by Stock and Watson (2020, Global Edition) displays interest rates of 10-year U.S. Treasury bonds and 3-month U.S. Treasury bills from 1960 to 2012. The following code chunks reproduce this figure.

```
# 3-month Treasury bills interest rate
TB3MS <- xts(USMacroSWQ$TB3MS, USMacroSWQ$Date)["1960::2012"]

# 10-year Treasury bonds interest rate
TB10YS <- xts(USMacroSWQ$GS10, USMacroSWQ$Date)["1960::2012"]

# term spread
TSpread <- TB10YS - TB3MS
```

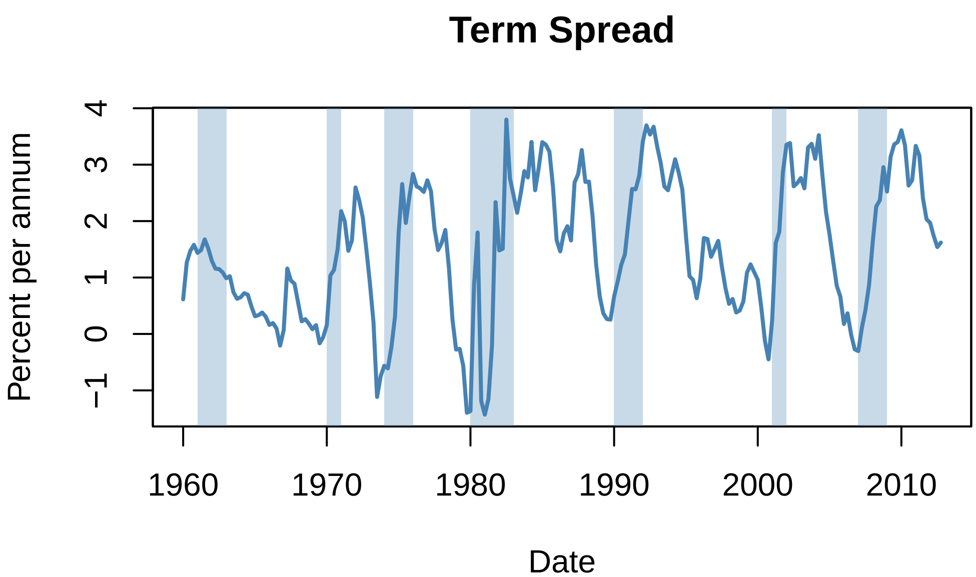
```
# reproduce Figure 15.3 (a) of the book
plot(merge(as.zoo(TB3MS), as.zoo(TB10YS)),
     plot.type = "single",
     col = c("darkred", "steelblue"),
     lwd = 2,
     xlab = "Date",
     ylab = "Percent per annum",
     main = "10-year and 3-month Interest Rates")

# define function that transform years to class 'yearqtr'
YToYQTR <- function(years) {
  return(
      sort(as.yearqtr(sapply(years, paste, c("Q1", "Q2", "Q3", "Q4"))))
  )
}
```

```r
# recessions
recessions <- YToYQTR(c(1961:1962, 1970, 1974:1975, 1980:1982, 1990:1991, 2001,
                        2007:2008))

# add color shading for recessions
xblocks(time(as.zoo(TB3MS)),
        c(time(TB3MS) %in% recessions),
        col = alpha("steelblue", alpha = 0.3))

# add a legend
legend("topright",
       legend = c("TB3MS", "TB10YS"),
       col = c("darkred", "steelblue"),
       lwd = c(2, 2))
```




```r
# reproduce Figure 15.3 (b) of the book
plot(as.zoo(TSpread),
     col = "steelblue",
     lwd = 2,
     xlab = "Date",
     ylab = "Percent per annum",
     main = "Term Spread")

# add color shading for recessions
xblocks(time(as.zoo(TB3MS)),
```

```
        c(time(TB3MS) %in% recessions),
        col = alpha("steelblue", alpha = 0.3))
```



Before recessions, the gap between interest rates on long-term bonds and short-term bills narrows, causing the term spread to decline significantly, sometimes even turning negative during economic stress. This information can be used to improve future GDP growth forecasts.

We can verify this by estimating an ADL$(2, 1)$ model and an ADL$(2, 2)$ model of the GDP growth rate, using lags of GDP growth and lags of the term spread as regressors. Then we use both models to forecast GDP growth for 2013.

```
# convert growth and spread series to ts objects
GDPGrowth_ts <- ts(GDPGrowth,
                  start = c(1960, 1),
                  end = c(2013, 4),
                  frequency = 4)

TSpread_ts <- ts(TSpread,
                start = c(1960, 1),
                end = c(2012, 4),
                frequency = 4)

# join both ts objects
ADLdata <- ts.union(GDPGrowth_ts, TSpread_ts)
```

```r
# estimate the ADL(2,1) model of GDP growth
GDPGR_ADL21 <- dynlm(GDPGrowth_ts ~ L(GDPGrowth_ts) + L(GDPGrowth_ts, 2) +
                       L(TSpread_ts), start = c(1962, 1), end = c(2012, 4))

coeftest(GDPGR_ADL21, vcov. = sandwich)
```

```
t test of coefficients:

                   Estimate Std. Error t value Pr(>|t|)   
(Intercept)        0.954990   0.486976  1.9611 0.051260 . 
L(GDPGrowth_ts)    0.267729   0.082562  3.2428 0.001387 **
L(GDPGrowth_ts, 2) 0.192370   0.077683  2.4763 0.014104 * 
L(TSpread_ts)      0.444047   0.182637  2.4313 0.015925 * 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We obtain for ADL $(2,1)$ the following equation

$$\widehat{GDPGR}_t = \underset{(0.49)}{0.96} + \underset{(0.08)}{0.26}\, GDPGR_{t-1} + \underset{(0.08)}{0.19}\, GDPGR_{t-2} + \underset{(0.18)}{0.44}\, TSpread_{t-1}$$

with all coefficients significant at the 5% level.

Let's now predict the GDP growth for 2013:Q1 and compute the forecast error.

```r
# 2012:Q3 / 2012:Q4 data on GDP growth and term spread
subset <- window(ADLdata, c(2012, 3), c(2012, 4))

# ADL(2,1) GDP growth forecast for 2013:Q1
ADL21_forecast <- coef(GDPGR_ADL21) %*% c(1, subset[2, 1], subset[1, 1],
                                          subset[2, 2])
ADL21_forecast
```

```
         [,1]
[1,] 2.241689
```

```r
# compute the forecast error
window(GDPGrowth_ts, c(2013, 1), c(2013, 1)) - ADL21_forecast
```

```
          Qtr1
2013 -1.102487
```

The ADL$(2, 1)$ model predicts the GDP growth in 2013:Q1 to be 2.24%, which leads to a forecast error of -1.10%.

We now estimate the ADL$(2, 2)$ model to determine if incorporating additional information from past term spreads enhances the forecast.

```
# estimate the ADL(2,2) model of GDP growth
GDPGR_ADL22 <- dynlm(GDPGrowth_ts ~ L(GDPGrowth_ts) + L(GDPGrowth_ts, 2)
                     + L(TSpread_ts) + L(TSpread_ts, 2),
                     start = c(1962, 1), end = c(2012, 4))

coeftest(GDPGR_ADL22, vcov. = sandwich)
```

```
t test of coefficients:

                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         0.967967   0.472470  2.0487 0.041800 *
L(GDPGrowth_ts)     0.243175   0.077836  3.1242 0.002049 **
L(GDPGrowth_ts, 2)  0.177070   0.077027  2.2988 0.022555 *
L(TSpread_ts)      -0.139554   0.422162 -0.3306 0.741317
L(TSpread_ts, 2)    0.656347   0.429802  1.5271 0.128326
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated AR$(2, 2)$ model equation is

$$
\begin{aligned}
\widehat{GDPGR}_t = \underset{(0.47)}{0.98} + \underset{(0.08)}{0.24}\, GDPGR_{t-1} + \underset{(0.08)}{0.18}\, GDPGR_{t-2} \\
- \underset{(0.42)}{0.14}\, TSpread_{t-1} + \underset{(0.43)}{0.66}\, TSpread_{t-2}
\end{aligned}
$$

While the coefficients on the lagged growth rates are still significant, the coefficients on both lags of the term spread are not significant at the 10% level.

```
# ADL(2,2) GDP growth forecast for 2013:Q1
ADL22_forecast <- coef(GDPGR_ADL22) %*% c(1, subset[2, 1], subset[1, 1],
                                          subset[2, 2], subset[1, 2])
ADL22_forecast
```

```
         [,1]
[1,] 2.274407
```

```
# compute the forecast error
window(GDPGrowth_ts, c(2013, 1), c(2013, 1)) - ADL22_forecast
```

```
          Qtr1
2013 -1.135206
```

The ADL$(2,2)$ model forecasts a GDP growth of 2.27% for 2013:Q1, which implies a forecast error of -1.14%.

Are ADL$(2,1)$ and ADL$(2,2)$ models better than the simple AR(2) model? Yes, while SER and $\bar{R}^2$ improve only slightly, an F-test on the term spread coefficients in the ADL$(2,2)$ model provides evidence that the model does better in explaining GDP growth than the AR(2) model, as the hypothesis that both coefficients are zero can be rejected at the 5% level.

```
# compare adj. R2
c("Adj.R2 AR(2)" = summary(GDPGR_AR2)$adj.r.squared,
  "Adj.R2 ADL(2,1)" = summary(GDPGR_ADL21)$adj.r.squared,
  "Adj.R2 ADL(2,2)" = summary(GDPGR_ADL22)$adj.r.squared)
```

```
   Adj.R2 AR(2) Adj.R2 ADL(2,1) Adj.R2 ADL(2,2)
      0.1338873       0.1620156       0.1691531
```

```
# compare SER
c("SER AR(2)" = summary(GDPGR_AR2)$sigma,
  "SER ADL(2,1)" = summary(GDPGR_ADL21)$sigma,
  "SER ADL(2,2)" = summary(GDPGR_ADL22)$sigma)
```

```
   SER AR(2) SER ADL(2,1) SER ADL(2,2)
    3.132122     3.070760     3.057655
```

```
# F-test on coefficients of term spread
linearHypothesis(GDPGR_ADL22,
                 c("L(TSpread_ts)=0", "L(TSpread_ts, 2)=0"),
                 vcov. = sandwich)
```

```
Linear hypothesis test

Hypothesis:
L(TSpread_ts) = 0
L(TSpread_ts, 2) = 0

Model 1: restricted model
Model 2: GDPGrowth_ts ~ L(GDPGrowth_ts) + L(GDPGrowth_ts, 2) + L(TSpread_ts) +
    L(TSpread_ts, 2)

Note: Coefficient covariance matrix supplied.

  Res.Df Df      F  Pr(>F)
1    201
2    199  2 4.4344 0.01306 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

10.7.2 Stationarity

Time series forecasts rely on past data to predict future values, assuming that the correlations and distributions of the data will remain consistent over time. This assumption is formalized by the concept of stationarity.

A time series Y_t is *stationary* if its probability distribution does not change over time - that is, if the joint distribution of $(Y_{s+1}, Y_{s+2}, \dots, Y_{s+T})$ does not depend on s, regardless of the value of T; otherwise, Y_t is said to be *nonstationary*.

Similarly, a pair of time series, X_t and Y_t, are said to be jointly stationary if the joint distribution of $(X_{s+1}, Y_{s+1}, X_{s+2}, Y_{s+2}, \dots, X_{s+T}, Y_{s+T})$ does not depend on s, regardless of the value of T.

10.7.3 Time Series Regression with Multiple Predictors

The general time series regression model extends the ADL model to include multiple regressors and their lags. It incorporates p lags of the dependent variable and q_l lags of l additional predictors where $l = 1, \dots, k$:

$$
\begin{aligned}
Y_t = &\beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_p Y_{t-p} \\
&+ \delta_{11} X_{1,t-1} + \delta_{12} X_{1,t-2} + \cdots + \delta_{1q} X_{1,t-q} \\
&+ \cdots \\
&+ \delta_{k1} X_{k,t-1} + \delta_{k2} X_{k,t-2} + \cdots + \delta_{kq} X_{k,t-q} \\
&+ u_t.
\end{aligned}
$$

The following **assumptions** are made for estimation:

1. The error term u_t has conditional mean zero given all regressors and their lags:

$$E(u_t \mid Y_{t-1}, Y_{t-2}, \dots, X_{1,t-1}, X_{1,t-2}, \dots, X_{k,t-1}, X_{k,t-2}, \dots)$$

This assumption extends the conditional mean zero assumption used for AR and ADL models and ensures that the general time series regression model described above provides the optimal forecast of Y_t given its lags, the additional regressors $X_{1,t}, \dots, X_{k,t}$, and their lags.

2. The i.i.d. assumption for cross-sectional data is not entirely applicable to time series data. Instead, we replace it with the following assumption, which has two components:

   a. The $(Y_t, X_{1,t}, \dots, X_{k,t})$ have a stationary distribution (the "identically distributed" part of the i.i.d. assumption for cross-sectional data). If this condition does not hold, forecasts *may* be biased and inference *can* be significantly misleading.

   b. The $(Y_t, X_{1,t}, \dots, X_{k,t})$ and $(Y_{t-j}, X_{1,t-j}, \dots, X_{k,t-j})$ become independent as j becomes large (the "independently" distributed part of the i.i.d. assumption for cross-sectional data). This assumption is also referred to as *weak dependence* and ensures that the WLLN and the CLT hold in large samples.

3. Large outliers are unlikely: $E(X_{1,t}^4), E(X_{2,t}^4), \dots, E(X_{k,t}^4)$ and $E(Y_t^4)$ have nonzero, finite fourth moments.

4. No perfect multicollinearity.

Given the nonstationary nature observed in many economic time series, assumption two plays a crucial role in applied macroeconomics and finance, leading to the development of statistical tests designed to determine stationarity or nonstationarity that will be explained later.

10.7.4 Statistical Inference and the Granger Causality Test

If X serves as a valuable predictor for Y, then in a regression where Y_t is regressed on its own lags and lags of X_t, some coefficients on the lags of X_t are expected to be non-zero. This concept is known as *Granger causality* and presents an interesting hypothesis for testing.

The Granger causality test is an F-test of the null hypothesis that all lags of a variable X included in a time series regression model do not have predictive power for Y_t. It does not test whether X actually causes Y, but whether the included lags are informative in terms of predicting Y.

This is the test we have previously performed on the ADL$(2,2)$ model of GDP growth and we concluded that at least one of the first two lags of term spread has predictive power for GDP growth.

10.8 Forecast Uncertainty and Forecast Intervals

It is typically good practice to include a measure of uncertainty when presenting results affected by it. Uncertainty becomes especially important in the context of time series forecasting.

For instance, consider a basic ADL$(1,1)$ model

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \delta_1 X_{t-1} + u_t,$$

where u_t is a homoskedastic error term. The forecast error is then

$$Y_{T+1} - \widehat{Y}_{T+1|T} = u_{T+1} - [(\hat{\beta}_0 - \beta_0) + (\hat{\beta}_1 - \beta_1)Y_T + (\widehat{\delta_1} - \delta_1)X_T]$$

The mean squared forecast error (MSFE) and the RMSFE are

$$\begin{aligned} MSFE &= E\left[(Y_{T+1} - \widehat{Y}_{T+1|T})^2\right] \\ &= \sigma_u^2 + Var\left[(\hat{\beta}_0 - \beta_0) + (\hat{\beta}_1 - \beta_1)Y_T + (\hat{\delta}_1 - \delta_1)X_T\right], \\ RMSE &= \sqrt{\sigma_u^2 + \text{Var}\left[(\hat{\beta}_0 - \beta_0) + (\hat{\beta}_1 - \beta_1)Y_T + (\hat{\delta}_1 - \delta_1)X_T\right]}. \end{aligned}$$

A 95% forecast interval is an interval that, in 95% of repeated applications, includes the true value of Y_{T+1}.

There is a fundamental distinction between computing a confidence interval and a forecast interval. When deriving a confidence interval for a point estimate, we use large sample approximations justified by the Central Limit Theorem (CLT), and these are valid across a broad range of error term distributions.

On the other hand, to compute a forecast interval for Y_{T+1}, an additional assumption about the distribution of u_{T+1}, the error term in period $T+1$, is necessary.

Assuming that u_{T+1} follows a normal distribution, it is possible to create a 95% forecast interval for Y_{T+1} using $SE(Y_{T+1} - \widehat{Y}_{T+1|T})$, which represents an estimate of the Root Mean Squared Forecast Error (RMSFE).

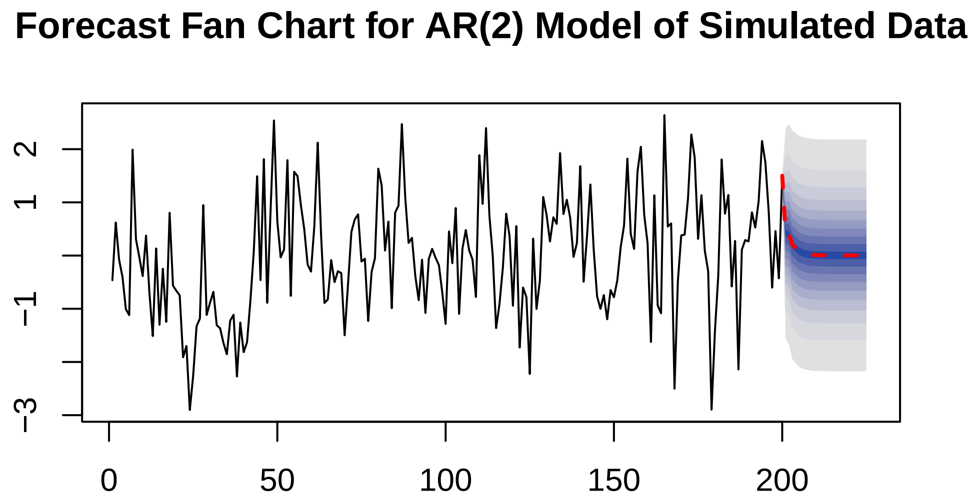
$$\widehat{Y}_{T+1|T} \pm 1.96 \cdot SE(Y_{T+1} - \widehat{Y}_{T+1|T})$$

Nevertheless, the computation gets more complicated when the error term is heteroskedastic or if we are interested in computing a forecast interval for $T+s$ when $s>1$.

In some cases it is useful to report multiple forecast intervals for subsequent periods. To ilustrate an example, we will use simulated time series data and estimate an AR(2) model which is then used for forecasting the subsequent 25 future outcomes of the series.

```
# set seed
set.seed(1234)

# simulate the time series
Y <- arima.sim(list(order = c(2, 0, 0), ar = c(0.2, 0.2)),  n = 200)

# estimate an AR(2) model using 'arima()', see ?arima
model <- arima(Y, order = c(2, 0, 0))

# compute points forecasts and prediction intervals for the next 25 periods
fc <- forecast(model, h = 25, level = seq(5, 99, 10))

# plot a fan chart
plot(fc,
     main = "Forecast Fan Chart for AR(2) Model of Simulated Data",
     showgap = F,
     fcol = "red",
     flty = 2)
```

`arima.sim()` simulates autoregressive integrated moving average (ARIMA) models. These are the class of models AR models belong to.



We use `list(order = c(2, 0, 0), ar = c(0.2, 0.2))` so the data generating process (DGP) is

$$Y_t = 0.2Y_{t-1} + 0.2Y_{t-2} + u_t.$$

We choose `level = seq(5, 99, 10)` in the call of `forecast()` so that forecast intervals with levels $5\%, 15\%, \ldots, 95\%$ are computed for each point forecast of the series.

The dashed red line displays the series' point forecasts for the next 25 periods using an AR(2) model, while the shaded areas represent prediction intervals.

The shading intensity corresponds to the interval's level, with the darkest blue band representing the 5% forecast intervals, gradually fading to grey with higher interval levels.

10.9 Lag Length Selection using Information Criteria

The determination of lag lengths in AR and ADL models may be influenced by economic theory, yet statistical techniques are useful in selecting the appropriate number of lags as regressors.

Including too many lags typically inflates the standard errors of coefficient estimates, leading to increased forecast errors, whereas omitting essential lags can introduce estimation biases into the model.

The order of an AR model can be determined using two approaches:

1. The F-test approach

Estimate an AR(p) model and test the significance of its largest lag(s). If statistical tests suggest that certain lag(s) are not significant, we may consider removing them from the model. However, this method often leads to overfitting, as significance tests can sometimes incorrectly reject a true null hypothesis.

2. Relying on an information criterion

To avoid the problem of overly complex models, one can select the lag order that minimizes one of the following two information criteria:

- The *Bayes Information Criterion* (BIC):

$$BIC(p) = \log\left(\frac{SSR(p)}{T}\right) + (p+1)\frac{\log(T)}{T}$$

- The *Akaike Information Criterion* (AIC):

$$AIC(p) = \log\left(\frac{SSR(p)}{T}\right) + (p+1)\frac{2}{T}$$

Both criteria are estimators of the optimal lag length p. The lag order $\widehat{p}$ that minimizes the respective criterion is called the BIC estimate or the AIC estimate of the optimal model order.

The basic idea of both criteria is that the SSR decreases as additional lags are added to the model, such that the first term decreases whereas the second increases as the lag order grows.

BIC decreases because of its logarithmic penalty term, while AIC's penalty term is less severe. BIC is consistent in estimating the true lag order, whereas AIC's consistency is less assured due to its different penalty factor.

Despite this, both criteria are commonly employed, with AIC sometimes favored when BIC suggests a model with too few lags.

The `dynlm()` function does not compute information criteria by default. Hence, we will create a custom function to calculate and display the Bayesian Information Criterion (BIC), alongside the selected lag order p and the adjusted $\bar{R}^2$, for objects of class `dynlm`.

```
# compute BIC for AR model objects of class 'dynlm'
BIC <- function(model) {

  ssr <- sum(model$residuals^2)
  t <- length(model$residuals)
  npar <- length(model$coef)
```

```r
  return(
    round(c("p" = npar - 1,
          "BIC" = log(ssr/t) + npar * log(t)/t,
          "Adj.R2" = summary(model)$adj.r.squared), 4)
  )
}
```

The following code computes the Bayesian Information Criterion (BIC) for autoregressive (AR) models of GDP growth with orders $p = 1, \ldots, 6$.

`sapply()` function is used to apply the BIC calculation to each model and display the results, including the BIC values and the adjusted $\bar{R}^2$ for each order. This allows for a comparison of model fit across different lag lengths.

```r
# apply the BIC() to an intercept-only model of GDP growth
BIC(dynlm(ts(GDPGR_level) ~ 1))
```

```
     p    BIC Adj.R2
0.0000 2.4394 0.0000
```

```r
# loop BIC over models of different orders
order <- 1:6

BICs <- sapply(order, function(x)
        "AR" = BIC(dynlm(ts(GDPGR_level) ~ L(ts(GDPGR_level), 1:x))))

BICs
```

```
         [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
p      1.0000 2.0000 3.0000 4.0000 5.0000 6.0000
BIC    2.3486 2.3475 2.3774 2.4034 2.4188 2.4429
Adj.R2 0.1099 0.1339 0.1303 0.1303 0.1385 0.1325
```

Increasing the lag order tends to increase R^2 because adding more lags generally reduces the sum of squared residuals SSR. However, $\bar{R}^2$ adjusts for the number of parameters in the model, mitigating the inflation of R^2 due to additional variables.

Despite $\bar{R}^2$ considerations, according to the BIC criterion, opting for the AR(2) model over the AR(5) model is recommended. The BIC helps in assessing whether the reduction in SSR justifies the inclusion of an additional regressor.

If we had to compare a bigger set of models, we may use the function `which.min()` to select the model with the lowest BIC.

```r
# select the AR model with the smallest BIC
BICs[, which.min(BICs[2, ])]
```

```
     p    BIC Adj.R2
2.0000 2.3475 0.1339
```

The *BIC* may also be used to select lag lengths in time series regression models with multiple predictors. In a model with K coefficients, including the intercept, we have

$$\text{BIC}(K) = \log\left(\frac{SSR(K)}{T}\right) + K\frac{\log(T)}{T}.$$

Choosing the optimal model according to the *BIC* can be computationally demanding, since there may be many different combinations of lag lengths when there are multiple predictors.

As an example, we estimate ADL(p, q) models of GDP growth, incorporating the term spread between short-term and long-term bonds as an additional variable.

We impose the constraint $p = q_1 = \ldots = q_k$ so that only a maximum of $p_{\max}$ models ($p = 1, \ldots, p_{\max}$) need to be estimated. In the example below, we set $p_{\max} = 12$.

```r
# loop 'BIC()' over multiple ADL models
order <- 1:12

BICs <- sapply(order, function(x)
         BIC(dynlm(GDPGrowth_ts ~ L(GDPGrowth_ts, 1:x) + L(TSpread_ts, 1:x),
                   start = c(1962, 1), end = c(2012, 4))))

BICs
```

```
         [,1]   [,2]   [,3]   [,4]    [,5]    [,6]    [,7]    [,8]    [,9]
p      2.0000 4.0000 6.0000 8.0000 10.0000 12.0000 14.0000 16.0000 18.0000
BIC    2.3411 2.3408 2.3813 2.4181  2.4568  2.5048  2.5539  2.6029  2.6182
Adj.R2 0.1332 0.1692 0.1704 0.1747  0.1773  0.1721  0.1659  0.1586  0.1852
         [,10]   [,11]   [,12]
p      20.0000 22.0000 24.0000
BIC     2.6646  2.7205  2.7664
Adj.R2  0.1864  0.1795  0.1810
```

According to the definition of `BIC()`, for ADL models where $p = q$, p represents the count of estimated coefficients excluding the intercept. Consequently, the lag order is derived by dividing p by 2.

```r
# select the ADL model with the smallest BIC
BICs[, which.min(BICs[2, ])]
```

```
     p    BIC Adj.R2
4.0000 2.3408 0.1692
```

The *BIC* favors the previously estimated ADL$(2, 2)$ model.

Part II

Empirical Methods 2023

11 Basic Principles

11.1 The frequentist approach

Observations are generated by a data generating process

Probabilistic model:

$$\binom{X_i}{Y_i} \sim F_{xy}(\theta)$$

For example $F_{xy}(\theta)$ represents $N(, \Sigma)$ (joint normal)

If the conditional distribution is linear in X, we have

$$\mathbb{E}(Y_i|X_i) = \alpha + \beta X_i$$

where Proof

$$\begin{aligned}
\alpha &= \mathbb{E}(Y_i) - \beta\,\mathbb{E}(X_i) \\
\beta &= \frac{\mathbb{E}(X_i Y_i) - \mathbb{E}(X_i)\mathbb{E}(Y_i)}{\mathbb{E}(X_i^2) - [\mathbb{E}(X_i)]^2} = \frac{cov(Y_i, X_i)}{var(X_i)}
\end{aligned}$$

The OLS estimator can be seen as replacing the population moments by the sample moments

The conditional expectation answers the question: What is the expected value of Y_i if we were able to fix X_i at some prespecified value $X_i = x$?

The parameters of interest $\theta = (\alpha, \beta)'$ result as a function of the joint distribution of X_i and Y_i, that is,

$$\theta = t(X_i, Y_i)$$

where $\hat{\theta}$ denotes the estimated analog based on the available sample

The accuracy of the estimate is measured by

$$\text{bias} = \mathbb{E}(\hat{\theta}) - \theta \qquad \text{(systematic deviation)}$$

$$\text{var} = \mathbb{E}\left\{\left[\hat{\theta} - \mathbb{E}(\theta)\right]^2\right\} \qquad \text{(unsystematic deviation)}$$

$$\text{MSE} = \mathbb{E}\left[(\hat{\theta} - \theta)^2\right] = \text{bias}^2 + \text{var} \qquad \text{(total deviation)}$$

The frequentist notion refers to “an infinite sequence of future trials”.

11.1.1 Estimation principles

a) Plug-in principle Replace $t(X_i, Y_i)$ by its sample analogs:

$$s_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

$$s_x^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

and compute the estimator as $b = s_{xy}/s_x^2$

This often yields “optimal” estimators, but not always Accurrency measures and hypothesis tests can be obtained from the bootstrap principle

b) Maximum Likelihood

Joint density:

$$f_\theta(X_i, Y_i) = f_{\theta_1}(Y_i|X_i)f_{\theta_2}(X_i)$$

Maximizing the log-likelihood function:

$$\ell(\theta|X_i, Y_i) = \log f(X_i, Y_i) = \log f_{\theta_1}(Y_i|X_i) + \log f_{\theta_2}(X_i)$$

if θ_1 is independent of θ_2 we may maximize the conditional log-likelihood function:

$$\ell_c(\theta_1|X_i, Y_i) = \log f_{\theta_1}(Y_i|X_i)$$

where in a simple regression: $\theta_1 = (\alpha, \beta, \sigma^2)$

Problem: the (family of) distribution needs to be known

Often some "natural" distribution is supposed, e.g. normal (Gaussian) distribution

ML estimators have optimal properties: - ML estimators are (asymptotically) unbiased - ML estimators are (asymptotically) efficient - ML estimators are (asymptotically) normally distributed

11.1.2 Further properties of ML estimators

- Consistency of the ML estimator just requires: $E[D(\theta)] = 0$ where

$$D(\theta) = \frac{\partial \ell(\theta)}{\partial \theta}$$

If the likelihood is misspecified but this condition is nevertheless fulfilled, then the estimator is called "pseudo ML"

- For large N and correctly specified likelihood the covariance matrix can be estimated by the information matrix

$$\text{var}(\hat{\theta}) = I(\theta)^{-1} \quad \text{where} \quad I(\theta) = -E\left[\frac{\partial^2 \ell(\theta)}{\partial \theta \partial \theta'}\right] = E[D(\theta)D(\theta)']$$

where θ may be replaced by a consistent estimator. Example

- For pseudo ML estimators the covariance matrix needs to be adjusted ("sandwich estimator")

11.2 The Bayesian approach

Bayes' theorem: Reorganizing $p(y, \theta) = p(\theta)p(y|\theta) = p(y)p(\theta|y)$ we obtain:

$$\underbrace{p(\theta|y)}_{\text{posterior dis.}} = \underbrace{p(\theta)}_{\text{prior dist.}} \cdot \underbrace{\frac{L(\theta)}{p(y)}}_{\text{updating factor}} \propto p(\theta)L(\theta) \quad (\propto \ : \text{proportional to})$$

where $L(\theta) = p(y|\theta)$ denotes the likelihood function

Bayesians prefer employing a conjugate family of distribution where the prior and posterior distribution are special cases of the same family of distributions

Example: $y_i \sim \mathcal{N}(\mu, \sigma^2)$, where σ^2 is treated as known

$$\text{prior distribution} \quad \mu \sim \mathcal{N}(\mu_0, \sigma_0^2)$$

This results in the posterior distribution:

$$\mu|y_1, \dots, y_n \sim \mathcal{N}(\bar{\mu}, \bar{\sigma}^2)$$

with

$$\bar{\mu} = \frac{1}{\psi_0 + \psi_1}(\psi_0\mu_0 + \psi_1 Y) \qquad \bar{\sigma}^2 = \frac{1}{\psi_0 + \psi_1}$$

$\psi_0 = \frac{1}{\sigma_0^2}$ and $\psi_1 = \frac{n}{\sigma^2}$ (precision)

11.2.1 Parameter Estimation

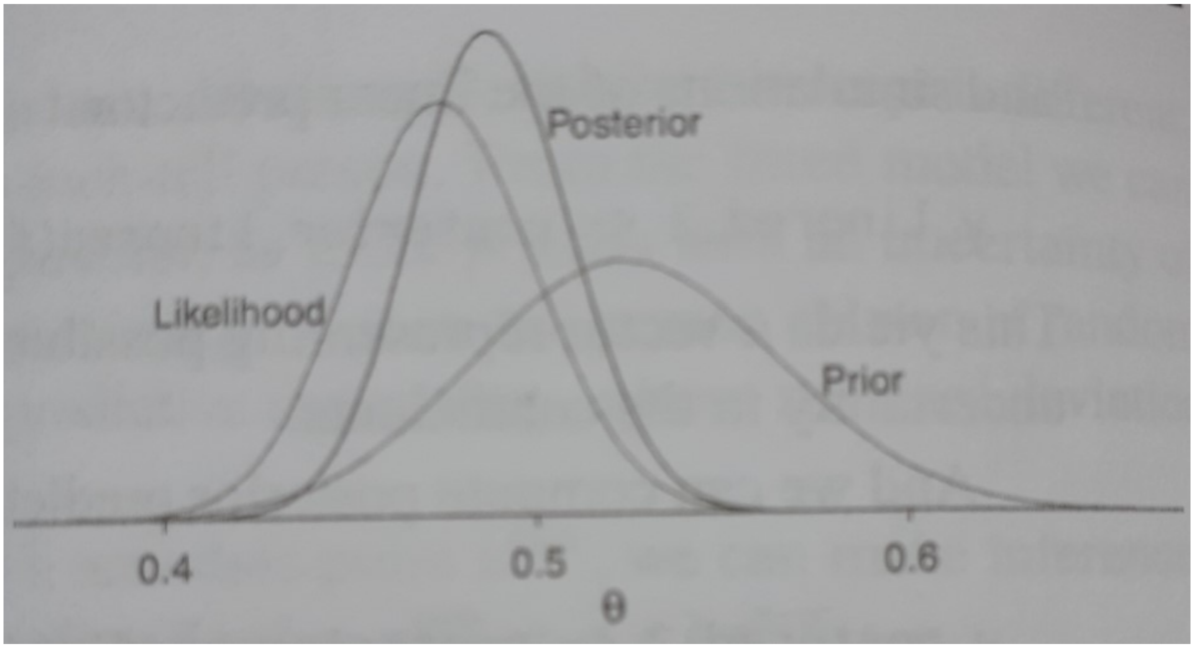
The MSE optimal estimate is obtained as

$$\hat{\theta} = E(\theta|y)$$

relationship to maximum likelihood:

$$\log p(\theta|y) = \text{const} + \underbrace{\log p(\theta)}_{O(1)} + \underbrace{\log L(\theta)}_{O(N)}$$

$\Rightarrow$ as $N \to \infty$ the mode of the posterior converge to ML



in most cases the posterior distribution is too difficult to be obtained analytically ⇒ Monte Carlo methods (Gibb sampler, MCMC simulator etc.)

Uniformative priors: Laplace's principle of insufficient reason ⇒ uniform distribution (flat prior)

Uniform distribution does not need to be uninformative (parameter transformation, e.g. $\psi = e^{\theta}$)

Jeffreys' prior is proportional to $1/\sigma_\theta$ (or square root of the Fisher information). Uniform prior is uninformative whenever σ_θ does not depend on unknown parameters.

11.2.2 Comparison with the frequentist approach

Bayesian approach takes care of knowledge accumulation

Bayesian machinery (MCMC) used to estimate extremely complicated models (frequentist approach fails)

Frequentist methods provide criteria for accessing the validity of the model. No such criteria for a Bayesian framework

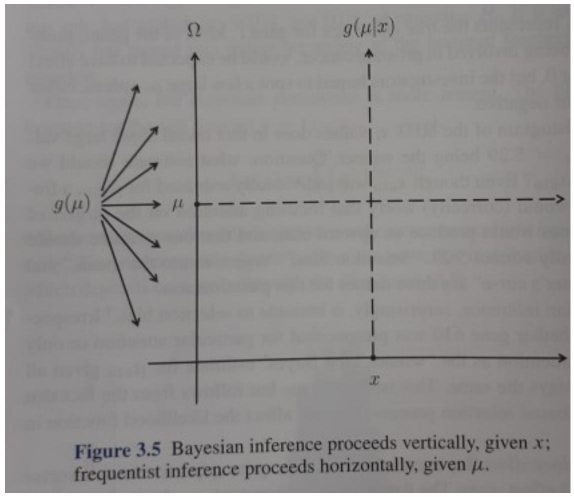


Figure 3.5 Bayesian inference proceeds vertically, given x; frequentist inference proceeds horizontally, given μ.

11.3 Machine Learning Approach

Characteristics of the MLearn approach:

- Big data. The data sets typically cover a large number of observations (nominal/qualitative, ordinal, metric). Large dimensional: many variables potentially useful for prediction
- Algorithmic approach: The data is typically unstructured with no specific "data generating model". Algorithms are constructed to learn the structure from the data
- Limited theory. The algorithms are flexible and "trained" (instead of estimated) by the data. Avoiding overfitting by splitting data into training and test sets
- MLern approaches are designed to cope with nonlinear data features
- Consider the conditional mean function:

$$y_i = m(x_i) + u_i$$

where x_i is high dimensional (K may be even larger than N) and the functional form of $m(\cdot)$ is unknown. The goal is to minimize

$$MSE = E\left[y_i - m(x_i)\right]^2$$

- Supervised learning. Develop prediction rules for y_i given the vector x_i
- Unsupervised learning. Uncovering structure amongst high-dimensional x_i
- Classification. Assigning observations to groups (classes).

- Sparsity. Finding out which variables can be ignored.

Computational Feasibility

Algorithmic learning requires powerful computational tools

Extensive packages in R and Python

An input produces some output. In between a black box. The (relative) performance is often not clear

"causal machine learning" tries to circumvent the correlation-is-not-causality critique

11.3.1 Regression as conditional expectation

Assume that the conditional expectation is a linear function such that

$$E(Y_i|X_i) = \alpha + \beta X_i$$

Taking expectations with respect to X_i yields

$$\alpha = E(Y_i) - \beta E(X_i)$$

Furthermore we have

$$\begin{aligned} E(X_i Y_i) &= \underset{x}{E}[X_i E(Y_i|X_i)] = \alpha E(X_i) + \beta E(X_i^2) \\ E(X_i)E(Y_i) &= E(X_i)\underset{x}{E}[E(Y_i|X_i)] = \alpha E(X_i) + \beta [E(X_i)]^2 \end{aligned}$$

Inserting the expression for α yields

$$\beta = \frac{E(X_i Y_i) - E(X_i)E(Y_i)}{E(X_i^2) - [E(X_i)]^2} = \frac{\text{cov}(Y_i, X_i)}{\text{var}(X_i)}$$

Back

11.3.2 ML estimation for the waiting time

Assume that the waiting time τ_i is exponentially distributed with

$$\tau_i \sim \lambda e^{-\lambda\tau} \quad E(\tau_i) = \frac{1}{\lambda} \quad \text{var}(\tau_i) = \frac{1}{\lambda^2}$$

The log-likelihood function results as

$$\ell(\lambda) = N\log(\lambda) - \lambda\sum_{i=1}^{N}\tau_i$$

with derivative

$$\frac{\partial\ell(\lambda)}{\partial\lambda} = \frac{N}{\lambda} - \sum_{i=1}^{N}\tau_i$$

The ML estimator results as $\hat{\lambda} = 1/\hat{\tau}$. The information (matrix) results as

$$I(\lambda) = -E\left(-\frac{N}{\lambda^2}\right) = \frac{N}{\lambda^2}$$

yielding $var(\hat{\lambda}) = \lambda^2/N$

The joint density results as

$$\begin{aligned}\log L(X,\mu) + \log p(\mu) &= \text{const} - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(Y_i-\mu)^2 - \underbrace{\frac{1}{2\sigma_0^2}(\mu-\mu_0)^2}_{\text{prior distribution}} \\ &= \text{const} - \underbrace{\frac{\psi_1+\psi_0}{2}}_{1/(2\bar{\sigma}^2)}\mu^2 + 2\mu\underbrace{\left(\frac{\psi_1\bar{Y}+\psi_0\mu_0}{2}\right)}_{\bar{\mu}/(2\bar{\sigma}^2)} + \ldots\end{aligned}$$

such that

$$\begin{aligned}\bar{\sigma}^2 &= \frac{1}{\psi_1+\psi_0} = \frac{1}{(n/\sigma^2)+(1/\sigma_0^2)} \\ \bar{\mu} &= \frac{1}{\psi_1+\psi_0}(\psi_1\bar{Y}+\psi_0\mu_0)\end{aligned}$$

Back

12 Regression Analysis

12.1 Data Collection

Many datasets provided via the WWW:

- Excel/CSV files provided by some organisation (Bundesbank, EZB, Statistisches Bundesamt, Eurostat …)
- Application programming interface (API): Fred Database
- Data scraping (extract data from a HTML code using R or Python)

CSV (Comma-separated values) is the most common format

Checking data for missing values and errors

Tidy data format (variables in columns, obs. in rows)

Compute descriptive statistics (mean, std.dev, min/max, distribution)

Report sufficient info on the data source (for replication)

12.2 Data Preparation

Assess the quality of the data source

Transform text into numerical values (dummy variables)

Plausibility checks / descriptive statistics

data set may contain missing values ('NA', dots, blank)

few NA: just ignore them (the row will be dropped)

when many observations lost: imputation (replace NA by estimated values)

a) Multiple Imputation: Assume that $x_{k,t}$ is missing. For available observations run the regression

$$x_{k,t} = \gamma_0 + \sum_{j=1}^{k-1} \gamma_j x_{j,t} + \epsilon_i$$

$\Rightarrow$ replace the missing values by $\hat{x}_{k,t}$.

For missing values in more regressors: iterative approach

MaxLike approach available for efficient imputation

12.3 OLS estimator

OLS: Ordinary least-square estimator

$$b = \underset{\beta}{\operatorname{argmin}} \{(y - X\beta)'(y - X\beta)\}$$

yields the least-squares estimator:

$$b = (X'X)^{-1}X'y$$

Unbiased estimator for σ^2: (note that $X'e = 0$)

$$s^2 = \frac{1}{N-K}(y - Xb)'(y - Xb)$$

Maximum-Likelihood (ML) estimator

Log-likelihood function assuming normal distribution:

$$\begin{aligned} \ell(\beta, \sigma^2) = \ln L(\beta, \sigma^2) &= \ln \left[\prod_{i=1}^{N} f(u_i) \right] \\ &= -\frac{N}{2} \ln 2\pi - \frac{N}{2} \ln \sigma^2 - \frac{1}{2\sigma^2}(y - X\beta)'(y - X\beta) \end{aligned}$$

ML and OLS of β are identical under normality

ML estimator for σ^2:

$$\tilde{\sigma}^2 = \frac{1}{N}(y - Xb)'(y - Xb)$$

Goodness of fit:

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{SSR}{TSS} = 1 - \frac{e'e}{y'y - N\bar{y}^2} = r_{xy}^2$$

adjusted R^2:

$$\bar{R}^2 = 1 - \frac{e'e/(N-K)}{(y'y - N\bar{y}^2)/(N-1)}$$

12.4 Properties of the OLS estimator

a) Expectation [note that $b = \beta + \underbrace{(X'X)^{-1}X'u}_{\text{estimation error}}$]

$$E(b) = \beta$$
$$E(s^2) = \sigma^2$$
$$E(\tilde{\sigma}^2) = \sigma^2(N-K)/N$$

b) Distribution assuming $u \sim \mathcal{N}(0, \sigma^2 I_N)$

$$b \sim \mathcal{N}(\beta, \Sigma_b), \quad \Sigma_b = \sigma^2(X'X)^{-1}$$

$$\frac{(N-K)}{\sigma^2}s^2 \sim \chi^2_{N-K}$$

c) Efficiency

b is BLUE

under normality: b and s^2 are MVUE

12.5 Testing Hypotheses

Significance level or **size** of a test (Type I error)

$$P(|t_k| \geq c_{\alpha/2} | \beta = \beta_0) = \alpha^*$$

where α is the nominal: size and α^* is the actual size

a test is unbiased (controls the size) if $\alpha^* = \alpha$

a test is asymptotically valid if $\alpha^* \to \alpha$ for $N \to \infty$

1 - type II error or power of the test:

$$P(|t_k| \geq c_{\alpha/2} | \beta = \beta^1) = \pi(\beta^1)$$

a test is consistent if

$$\pi(\beta^1) \to 1 \quad \text{for all} \quad \beta^1 \neq \beta_0$$

The conventional significance level is $\alpha = 0.05$ for a moderate sample size ($N \in [50, 500]$, say)

a test is uniform most powerful (UMP) if

$$\pi(\beta) \geq \pi^*(\beta) \quad \text{for all} \quad \beta \neq \beta^0$$

where $\pi^*(\beta)$ denotes the power function of any other unbiased test statistic.

⇒ The one-sided t-test is UMP but in many cases there does not exist a UMP test.

The p-value (or marginal significance level) is defined as

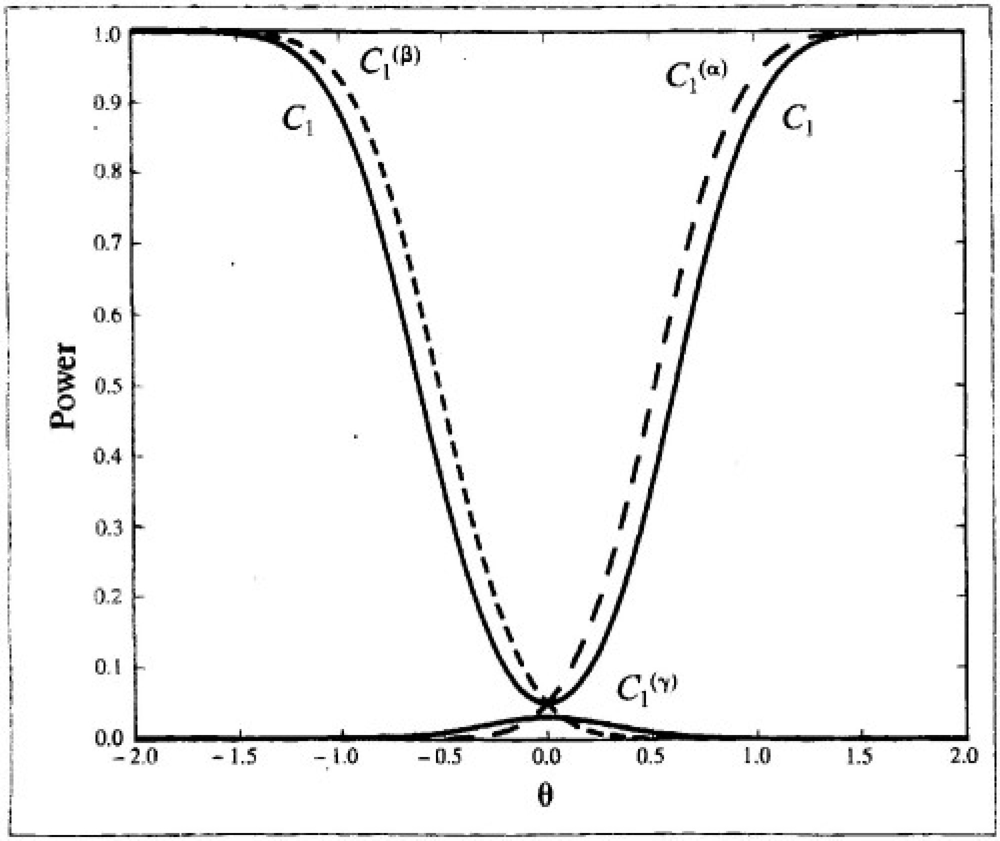


Figure 14.6 Comparison of power function

$$\text{p-value} = P(t_k \geq \bar{t}_k|\beta = \beta^0) = 1 - F_0(t_k)$$

that is, the probability to observe a larger value of the observed statistic $\bar{t}_k$.

Under the null hypothesis the p-value is uniformly distributed on [0, 1]. Since it is a random variable, it is NOT a probability (that the null hypothesis is correct).

Testing general linear hypotheses on β

J linear hypotheses on β represented by

$$H_0: \quad R\beta = q, \quad J \times 1$$

Wald statistic

$$Rb - q \sim \mathcal{N}\left(0, \sigma^2 R(X'X)^{-1}R'\right)$$

if σ^2 is known:

$$\frac{1}{\sigma^2}(Rb-q)'[R(X'X)^{-1}R']^{-1}(Rb-q) \sim \chi^2_J$$

if σ^2 is replaces by s^2:

$$F = \frac{1}{Js^2}(Rb-q)'[R(X'X)^{-1}R']^{-1}(Rb-q) = \frac{N-K}{J}\,\frac{(e_r'e_r - e'e)}{e'e}$$
$$\sim \frac{\chi^2_J/J}{\chi^2_{N-K}/(N-K)} \equiv F^J_{N-K}$$

Alternatives to the F statistic

Generalized LR test: $GLR = 2\left(\ell(\hat{\theta}) - \ell(\hat{\theta}_r)\right) = N(\log e_r'e_r - \log e'e) \sim \chi^2_J$

$\Rightarrow$ first order Taylor expansion yields the Wald/F statistic

LM (score) test: Define the "'score vector" as:

$$s(\hat{\theta}_r) = \left.\frac{\partial \log L(\theta)}{\partial \theta}\right|_{\theta=\hat{\theta}_r} = \frac{1}{\hat{\sigma}_r^2}X'e_r$$

The LM test statistic is given by

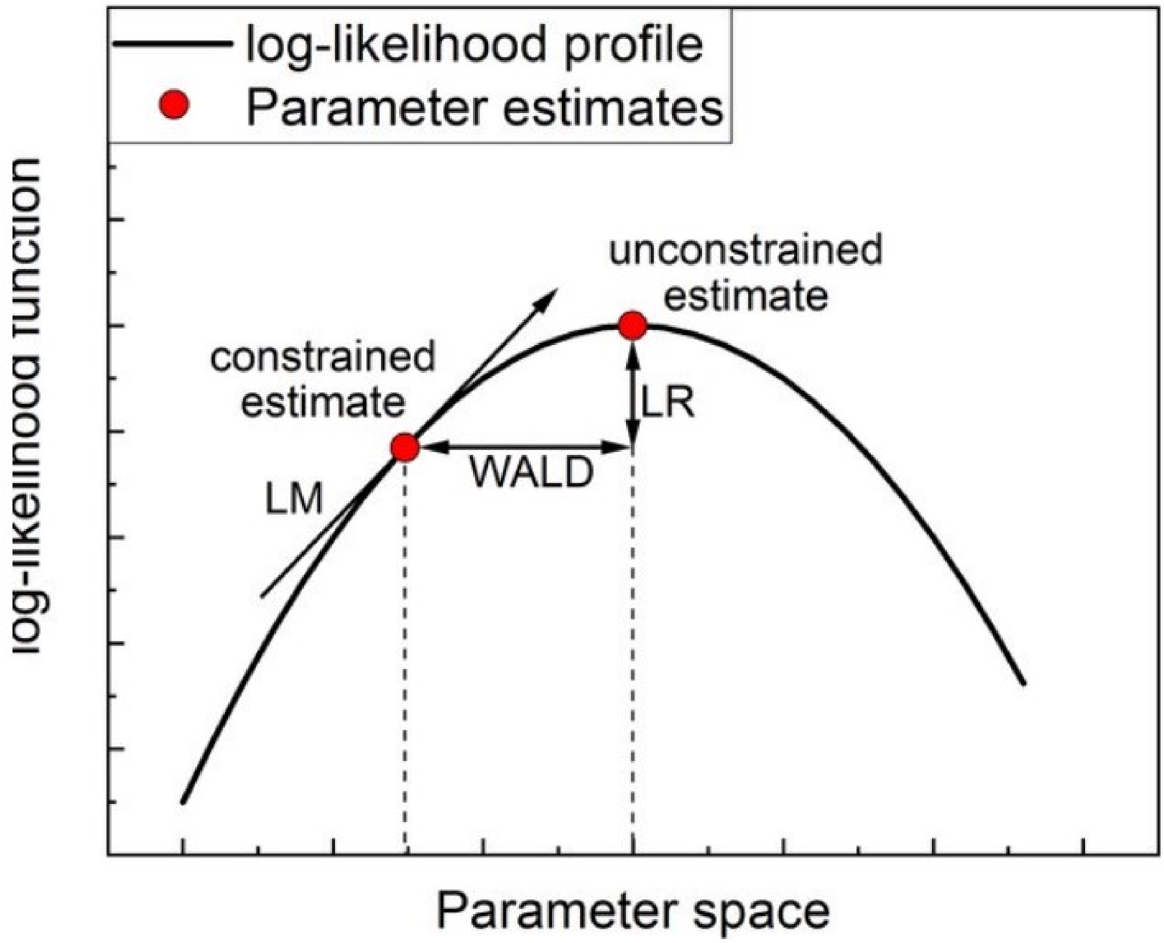
$$\text{LM} = s(\hat{\theta}_r)'I(\hat{\theta}_r)^{-1}s(\hat{\theta}_r) \sim \chi^2_J$$

where $I(\hat{\theta}_r)$ is some estimate of the information matrix

In the regression the LM statistic can be obtained from testing $\gamma = 0$ the auxiliary regression

$$1 = \gamma' s_i(\hat{\theta}_r) + \nu_i$$

$\Rightarrow$ uncentered R^2: $R_u^2 = \bar{s}'(\sum s_i s_i')^{-1}\bar{s}$. $N \cdot R_u^2 \sim \chi_J^2$



12.5.1 Specification tests

a) Test for Heteroskedasticity (Breusch-Pagan / Koenker)

variance function: $\sigma_i^2 = \alpha_0 + z_i'\alpha$

since $E(\hat{u}_i^2) \approx \sigma^2$ estimate the regression

$$\hat{u}_i^2 = \alpha_0 + z_i'\alpha + \nu_i$$

$\Rightarrow F$ or LM test statistic for H_0: $\alpha = 0$

in practice $z_i = x_i$ but also cross-products and squares of the regressors (White test)

robust (White) standard errors: replace invalid formula $Var(b) = \sigma^2(X'X)^{-1}$ by the estimator:

$$\widehat{Var}(b) = (X'X)^{-1}\left(\sum_{i=1}^{n} \hat{u}_i^2 x_i x_i'\right)(X'X)^{-1}$$

b) Tests for Autocorrelation

(i) Durbin-Watson-Test:

$$dw = \frac{\sum_{t=2}^{N}(\hat{u}_t - \hat{u}_{t-1})^2}{\sum_{t=1}^{N}\hat{u}_t^2} \approx 2(1-\hat{\rho})$$

Problem: Distribution depends on $X \Rightarrow$ uncertainty range

(ii) Breusch-Godfrey Test: $u_t = \rho_1 u_{t-1} + \cdots + \rho_m u_{t-m} + v_t$

replace u_t by $\hat{u}_t$ and include x_t to control for the estimation error in u_t and testing H_0: $\rho_1 = \cdots = \rho_m = 0$

(iii) Box-Pierce Test:

$$Q_m = T\sum_{j=1}^{m} \hat{\rho}_j^{\,2} \overset{a}{\sim} \chi_m^2$$

test of autocorrelation up to lag order m

HAC standard errors:

Heteroskedasticity and Autocorrelation Consistent standard errors (Newey/West 1987)

standard errors that account for autocorrelation up to lag h (truncation lag)

"Rule of thumb" for choosing H (e.g. Eviews/Gretl)

$$h = int[4(T/100)^{2/9}]$$

Relationship between autocorrelation and dynamic models:

Inserting $u_t = \rho u_{t-1} + v_t$ yields

$$y_i = \rho y_{t-1} + \beta' x_i - \underbrace{\rho\beta'}_{\gamma} x_{t-1} + v_i$$

$\Rightarrow$ Common factor restriction: $\gamma = -\beta\rho$

Test for normality

The asymptotic properties of the OLS estimator do not depend on the validity of the normality assumption

Deviations from the normal distribution only relevant in very small samples

Outliers may be modeled by mixing distributions

Tests for normality are very sensitive against outliers

Under the null hypothesis $E(u_i^3) = 0$ and $E(u_i^4) = 3\sigma^4$

Jarque-Bera test statistic:

$$JB = n\left[\frac{1}{6}\hat{m}_3^2 + \frac{1}{24}(\hat{m}_4 - 3)^2\right] \stackrel{d}{\to} \chi_2^2$$

where

$$\hat{m}_3 = \frac{1}{T\hat{\sigma}^3}\sum_{t=1}^{T}\hat{u}_i^3 \qquad \hat{m}_4 = \frac{1}{T\hat{\sigma}^4}\sum_{t=1}^{T}\hat{u}_i^4$$

Other tests: χ^2 and Kolmogorov-Smirnov Test

12.6 Nonlinear regression models

a) Polynomial regression

including squares, cubic etc. transformations of the regressors:

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \cdots + \beta_p X_i^p + u_i$$

where p is the degree of the polynomial (typically $p = 2$)

Interpretation (for $p = 2$)

$$\begin{aligned}
\frac{\partial Y}{\partial X} &= \beta_1 + 2\beta_2 X \\
\Rightarrow \Delta Y &\approx (\beta_1 + 2\beta_2 X)\Delta X \\
\text{exact: } \Delta Y &= \beta_1 \Delta X + \beta_2 (X + \Delta X)^2 - \beta_2 X^2 \\
&= (\beta_1 + 2\beta_2 X)\Delta X + \beta_2 (\Delta X)^2
\end{aligned}$$

$\Rightarrow$ the effect on Y depends on the level of X

for small changes in X the derivative provides a good approximation

Computing standard errors for the nonlinear effect:

Method 1:

$$\begin{aligned}
\text{s.e.}\left(\Delta\hat{Y}\right) &= \sqrt{\text{var}(b_1) + 4X^2\text{var}(b_2) + 8X\text{cov}(b_1, b_2)} \\
&= |\Delta\hat{Y}|/\sqrt{F}
\end{aligned}$$

where F is the F statistic for the test $E(\Delta\hat{Y}_i) = \beta_1 + 2X\beta_2 = 0$

Method 2:

$$Y_i = \beta_0 + \underbrace{(\beta_1 + 2X\beta_2)}_{\beta_1^*} X_i + \beta_2 \underbrace{\left(1 - 2\frac{X}{X_i}\right) X_i^2}_{X_i^*} + u_i$$

Regression $Y_i = \beta_0 + \beta_1^* X_i + \beta_2^* X_i^* + u_i$ and t-test of $\beta_1^* = 0$

Confidence interval for the effect are obtained as $\Delta\hat{Y} \pm z_{\alpha/2} \cdot s.e.(\Delta\hat{Y})$ or $b_1^* \pm \text{s.e.}(b_1^*)$

Logarithmic transformation

Three possible specifications:

$$\begin{aligned} \text{log-linear:} \qquad & \log(Y_i) = \beta_0 + \beta_1 X_i + u_i \\ \text{linear-log:} \qquad & Y_i = \beta_0 + \beta_1 \log(X_i) + u_i \\ \text{log-log:} \qquad & \log(Y_i) = \beta_0 + \beta_1 \log(X_i) + u_i \end{aligned}$$

Note that in the log-linear model

$$\beta_1 = \frac{d\log(Y)}{dX} = \underbrace{\frac{1}{Y}}_{outer} \cdot \underbrace{\frac{dY}{dX}}_{inner} = \frac{dY/Y}{dX}$$

where dY/Y indicates the relative change

In a similar manner it can be shown that for the log-log model $\beta_1 = (dY/Y)/(dX/X)$ is the elasticity

Note that the derivative refers to a small change. Exact:

$$\frac{Y_1 - Y_0}{Y_0} = e^{\beta_1 \Delta X} - 1$$

where $log(Y_0) = \beta_0 + \beta_1 X$ and $log(Y_1) = \beta_0 + \beta_1 (X + \Delta X)$.

For small ΔX we have $(Y_1 - Y_0)/Y_0 \approx \beta_1 \Delta X$

Interaction effects

Interaction terms are products of regressors:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 (X_{1i} \times X_{2i}) + u_i$$

where X_{1i}, X_{2i} may be discrete or continuous

Note that we can also write the model with interaction term as

$$Y_i = \beta_0 + \beta_1 X_{1i} + \underbrace{(\beta_2 + \beta_3 X_{1i})}_{\text{effect depends on } X_{1i}} X_{2i} + u_i$$

If X_{2i} is discrete (dummy), then the coefficient is different for $X_{2i} = 1$ and $X_{2i} = 0$

Standard errors also depend on X_{2i}:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2^* X_{2i} + \beta_3 (X_{1i} - \overline{X}_{1i}) X_{2i} + u_i$$

where $\beta_2^* = \beta_2 + \beta_3 \overline{X}_{1i}$ and $\overline{X}_{1i}$ is a fixed value of X_{1i}.

Nonlinear least-squares (NLS)

Assume a nonlinear relationship between Y_i and X_i where the parameters enter nonlinearly

$$Y_i = f(X_i, \beta) + u_i$$

Example:

$$f(X_i, \beta) = \beta_1 + \beta_2 X^{\beta_3}{}_i + u_i$$

Assuming i.i.d. normally distributed errors, the maximum likelihood principle results in minimizing the sum of squared residuals:

$$SSR(\beta) = \sum_{i=1}^{n} (y_i - f(X_i, \beta))^2$$

The SSR can be minimized by using iterative algorithms (Gauss-Newton method)

The Gauss-Newton method requires the first derivative of the function $f(X_i, \beta)$ with respect to β.

13 Machine Learning Methods

OLS regression requires sufficient degrees of freedom (N-K)

Asymptotic theory assumes $N - K \to \infty$

Standard asymptotic results are invalid if $K/N \to \kappa > 0$

OLS estimation typically no/small bias but large variance

Performance is measured by the “risk”, typically mean-squared error:

$$E(b-\beta)^2 = \underbrace{E\left\{[b-E(b)]^2\right\}}_{\text{variance}} + \underbrace{[E(b)-\beta]^2}_{\text{bias}^2}$$

The variance represents the unsystematic error and the bias the systematic error. If the parameters are estimated only oncy, the distinction becomes irrelevant.

Ridge estimation

Introducing an L_2 penalty:

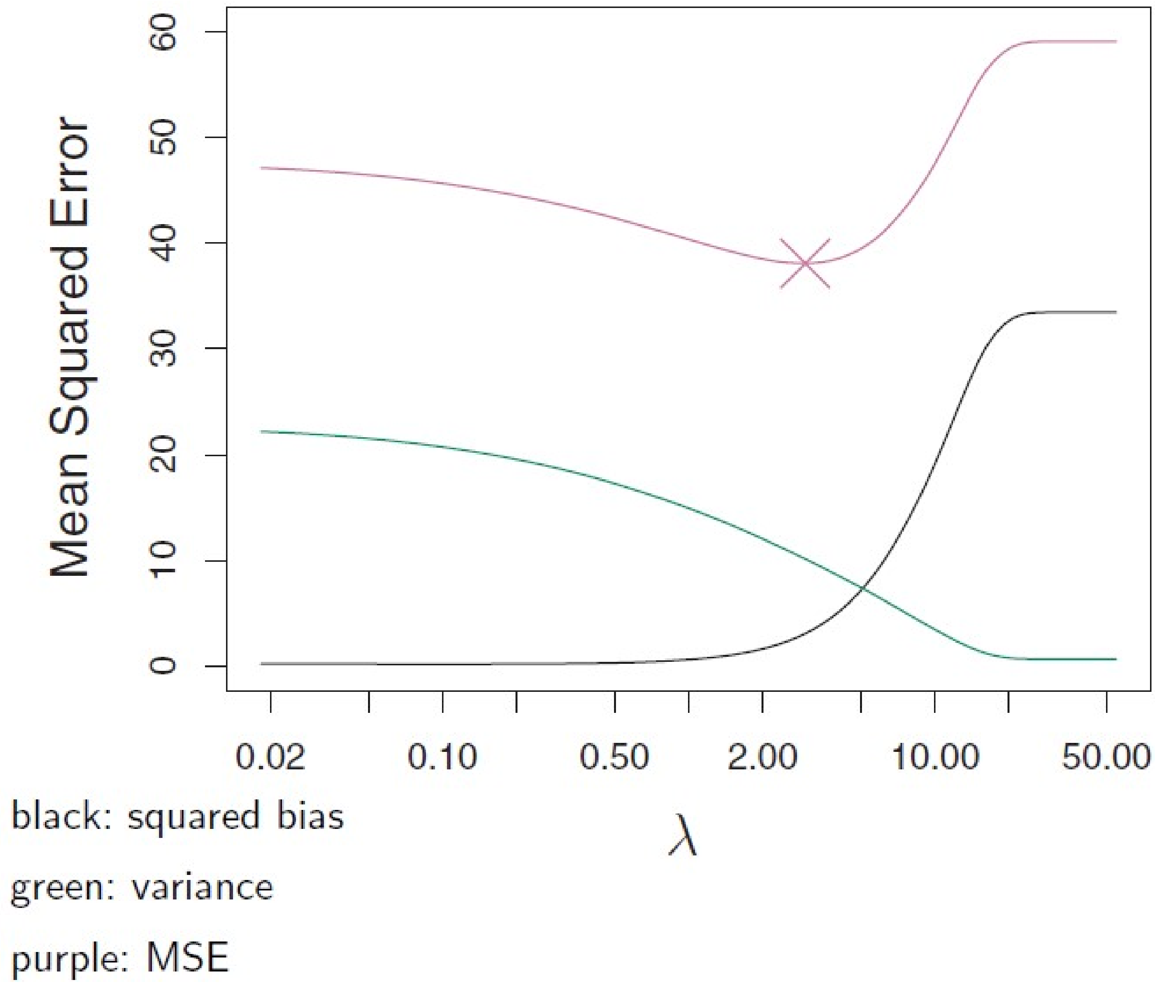
$$S_\lambda^R = (y - X\beta)'(y - X\beta) + \lambda\|\beta\|^2$$

where $\|\beta\| = \sqrt{\sum_{j=1}^K \beta_j^2}$ denotes the Gaussian norm

minimizing S_λ^R yields the Ridge estimator

$$\hat{\beta}_R = \left(X'X + \lambda I_K\right)^{-1} X'y$$

If $K > N$ then $X'X$ is singular but $X'X + \lambda I_K$ is not!



Since $X'X + \lambda I_K$ is "larger" than $X'X$ (in a matrix sense) the coefficients in $\hat{\beta}_R$ "shrink towards zero" as λ gets large

Choice of λ is typically data driven (see below)

13.1 Lasso Regression

"Sparse regression": Many of the coefficients are actually zero

Define the L_r norm as

$$\|\beta\|_r = \left(\sum_{j=1}^{K} |\beta_j|^r\right)^{1/r}$$

such that

$$S_\lambda^L = (y - X\beta)'(y - X\beta) + \lambda\|\beta\|_1$$

L_1 penalty corresponds to the constraint:

$$\sum_{j=1}^{K} |\beta_j| \leq \tau$$

Solution of the minimization problem by means of quadratic programming

The solution typically involves zero coefficients

Some more details

The regressors and dependent variables are typically standardized:

$$\tilde{x}_{j,i} = (x_{j,i} - \bar{x}_j)/s_j$$

where $\bar{x}_j$ and s_j are the mean and standard deviation of x_j

Relationship to pre-test estimator: For a simple regression The first order condition is given by:

$$\frac{2}{N}\left(\sum_{i=1}^{N} \tilde{x}_i \tilde{y}_i - \frac{\lambda}{N}\sum_{i=1}^{N} \tilde{x}_i^2 \beta\right) \pm \lambda \stackrel{!}{=} 0$$

$$\beta - \hat{\beta}_\lambda^L \pm \frac{N}{2}\lambda \stackrel{!}{=} 0$$

and therefore:

$$\hat{\beta}_\lambda^L = \begin{cases} b + \lambda^* & \text{if } b < -\lambda^* \\ 0 & \text{if } -\lambda^* \leq b \leq \lambda^* \\ b - \lambda^* & \text{if } b > \lambda^* \end{cases}$$

where $\lambda^* = \lambda \cdot N/2$

Selecting the shrinkage parameter

Trade-off between bias (large λ) and variance (small λ).

Choose λ that minimizes the $MSE = Bias^2 + Var$

leave-one-out **cross validation:**

Drop one observation and forecast it based on the remaining N-1 observations

k-fold **cross validation:**

divide randomly the set of observations into k groups (folds) of approximately equal size. The first fold is treated as a validation set and the remaining k-1 folds are employed for parameter estimation

evaluate the loss (MSE) for each observation conditional on λ and compute the average loss as a function of λ

Minimize the loss (MSE) with respect to λ

k is typically between 5 - 10

Refinements

post-Lasso estimation: Re-estimate the parameters by OLS leaving out the coefficients that were set to zero by LASSO

oracle property: the asymptotic distribution of the estimator is the same as if we knew which coefficient is equal to zero

Original LASSO does not exhibit the oracle property

Adaptive LASSO: with weighted penalty $\sum_{j=1}^{K} \widehat{w}_j |\beta_j|$ and

$$\widehat{w}_j = 1/|b_j|^\nu \qquad \text{with some } v > 0$$

where b_j denotes the OLS estimate

if K > N replace b_j by the simple regression coefficient

Adaptive LASSO possesses the oracle property

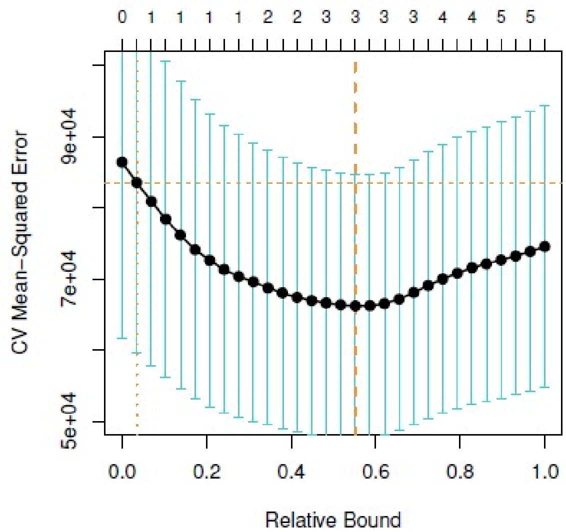


Figure 2.3 *Cross-validated estimate of mean-squared prediction error, as a function of the relative ℓ_1 bound $\tilde{t} = \|\widehat{\beta}(t)\|_1/\|\tilde{\beta}\|_1$. Here $\widehat{\beta}(t)$ is the lasso estimate corresponding to the ℓ_1 bound t and $\tilde{\beta}$ is the ordinary least-squares solution. Included are the location of the minimum, pointwise standard-error bands, and the "one-standard-error" location. The standard errors are large since the sample size N is only 50.*

Elastic net: hybrid method LASSO/Ridge

$$S_\lambda^L = (y - X\beta)'\,(y - X\beta) + \lambda_1 \,\|\beta\|_1 \, + \lambda_2 \,\|\beta\|_2^2$$

13.2 Dimension reduction techniques

if k is large, it is desirable to reduce the dimensionality by using linear combinations

$$Z_m = \sum_{j=1}^{k} \phi_{jm} X_j \qquad m = 1, \dots, k$$

where $\phi_{1m}, \dots, \phi_{km}$ as unknown constants such that M « k

Using these linear combinations ("common factors", principal components) the regression becomes

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i$$

Inserting shows that the elements of β fulfills the restriction

$$\beta_j = \sum_{m=1}^{M} \theta_m \phi_{jm}$$

choose $\phi_{1m}, \dots, \phi_{km}$ such that the "loss of information" is minimal

Principal component regression

choose the linear combinations Z_m such that they explain most of X_j:

$$X_j = \sum_{m=1}^{M} \alpha_{jm} Z_m + v_j$$

such that the variance of v_j is minimized:

$$S^2(\alpha, \phi) = \sum_{j=1}^{k} \sum_{i=1}^{n} v_{ji}^2$$

The linear combination is obtained from the eigenvectors of the sample covariance matrix of X

The number of linear combinations, M, can be determined by considering the ordered eigenvalues

Another approach is the method of Partial Least-Squares, where the linear combinations are found sequentially by considering the covariance with the dependent variable

Computational Details

let X denote an $N \times K$ matrix of regressors, then

$$X = ZA' + V$$

where Z is $N \times M$ where $M < K$. Inserting in the model for y yields

$$
\begin{aligned}
y &= Z \underbrace{A'\beta} + \underbrace{u + V\beta} \\
&= \quad Z'\theta \;+ \quad \epsilon
\end{aligned}
$$

the estimates can be obtained from the “singular value decomposition”:

$$
\begin{gathered}
X = UDV' \\
\text{where} \quad U'U = I_N \quad \text{and} \quad V'V = I_k
\end{gathered}
$$

and D an $N \times K$ diagonal matrix of the K singular values

the dimension reduction is obtained by dropping the last $N - M$ and $K - M$ columns of U and V respectively.

the matrix Z (loading matrix) is obtained as U_M and $F = D_M V_M'$ is the matrix of factors

13.3 Regression trees / Random forest

piecewise constant function:

$$
f(x) = \sum_{j=1}^{J} \mathbf{1}(x \in R_j) c_j
$$

where $R_j = [s_{j-1}, s_j)$ is some region in the real space

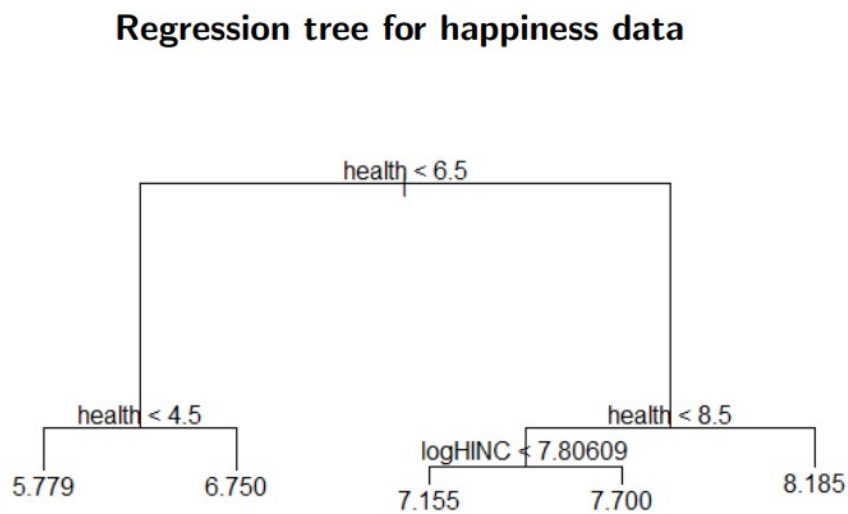
finding the best split point s for some splitting variable $x_{j,i}$ is simple if there are only two regions ($J = 2$) such that

$$
\min_s \left[\sum_{i \in R_1} (y_i - \hat{c}_1)^2 + \sum_{i \in R_2} (y_i - \hat{c}_2)^2 \right]
$$

we can proceed by searching for the next optimal split in the regions R_1 and R_2 and so on

how far do we need to extend this tree? Typically we stop if the region becomes dense (node size < 5, say)

Pruning the tree: Keep only the branches that result in a sufficient (involving some parameter α) reduction of the objective function. α is chosen by cross validation.



Bagging: Full sample $Z = \{(x_1', y_1), (x_2', y_2), \ldots, (x_N', y_N)\}$

The bootstrap sample: is obtained by drawing randomly from the set $\{1, 2, \cdots, N\}$ such that

$$Z_b^* = \{(x_1^{*\prime}, y_1^*), (x_2^{*\prime}, y_2^*), \ldots, (x_N^{*\prime}, y_N^*)\}$$

Let $\hat{f}_b^*(x)$ denote the prediction based on the regression fit based on Z_b^*. The aggregated prediction is obtained as

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b^*(x)$$

Bootstrap aggregation stabilizes the unstable outcome of a regression tree

Specifically, when growing a tree on a bootstrapped dataset: *Before each split, select* $m < p$, *e.g.* $m = int(\sqrt{p})$ *of the input variables at random as candidates for splitting.*

This reduces the correlation among the bootstrap draws

14 Limited Dependent Variables

Binary choice models

Choice based on Utility

$$U_{i0} = x_i' \gamma_0 + \epsilon_{i0}$$
$$U_{i1} = x_i' \gamma_1 + \epsilon_{i1}$$

where

$$U_{ij} : \text{utility due to the choice of j}$$
$$x_i : \text{variables characterizing the individual i}$$

Decision rule:

$$y_i^* = U_{i1} - U_{i0} = \begin{cases} > 0 & \Rightarrow \text{ choose 1} \\ \leq 0 & \Rightarrow \text{ choose 0} \end{cases}$$
$$y_i^* = x_i'(\gamma_1 - \gamma_0) + \epsilon_{i1} - \epsilon_{i0}$$
$$= x_i' \beta + \epsilon_i$$

where $\varepsilon_i = \epsilon_{i1} - \epsilon_{i0}$

14.1 Linear probability model

y_i^* in the binary choice model is typically not observed. What we observe is:

$$y_i = \begin{cases} 1 & \text{for } y_i^* > 0, \\ 0 & \text{for } y_i^* \leq 0. \end{cases}$$

Assuming that the probability function is linear we have

$$\begin{aligned} E(y_i|x_i) &= P(y_i = 1|x_i) \cdot 1 + P(y_i = 0|x_i) \cdot 0 \\ &= x_i'\beta \end{aligned}$$

In this case we can estimate the linear regression:

$$y_i = x_i'\beta + u_i$$

A linear probability function is pretty unrealistic and implies that ε_i is uniformly distributed (see below)

The errors u_i are heteroskedastic (variance depends on x_i). Robust standard errors are required.

14.2 Probit/Logit models

Consider the binary choice model with

$$\begin{aligned} P(y_i = 1) &= P(\varepsilon_i > -x_i'\beta) \\ &= 1 - F(-x_i'\beta) \end{aligned}$$

where $F(\cdot)$ denotes the distribution function of ε_i

It follows that

$$\begin{aligned} E(y_i|x_i) &= 1 - F(-x_i'\beta) \\ &= F(x_i'\beta) \text{ if the distribution is symmetric} \end{aligned}$$

Nonlinear regression model:

$$\begin{aligned} y_i &= E(y_i|x_i) + u_i \\ &= F(x_i'\beta) + u_i \quad \text{for symmetric distributions} \end{aligned}$$

error is (centered) binomially distributed with $p_i = F(x_i'\beta)$

estimation with Maximum Likelihood (similar to nonlinear regression)

Popular distributions:

$$\begin{aligned} F &\sim \text{normal distribution} \\ &\sim \text{logistic distribution} \end{aligned}$$

Choice of the Distribution:

- Usually no information about the distribution
- Referring to the central limit theorem
- Practical reasons
- Specification tests
- Nonparametric estimation

Normal distribution ("Probit")

$$F \equiv \Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-u^2/2}\, du$$

Logistic distribution ("Logit")

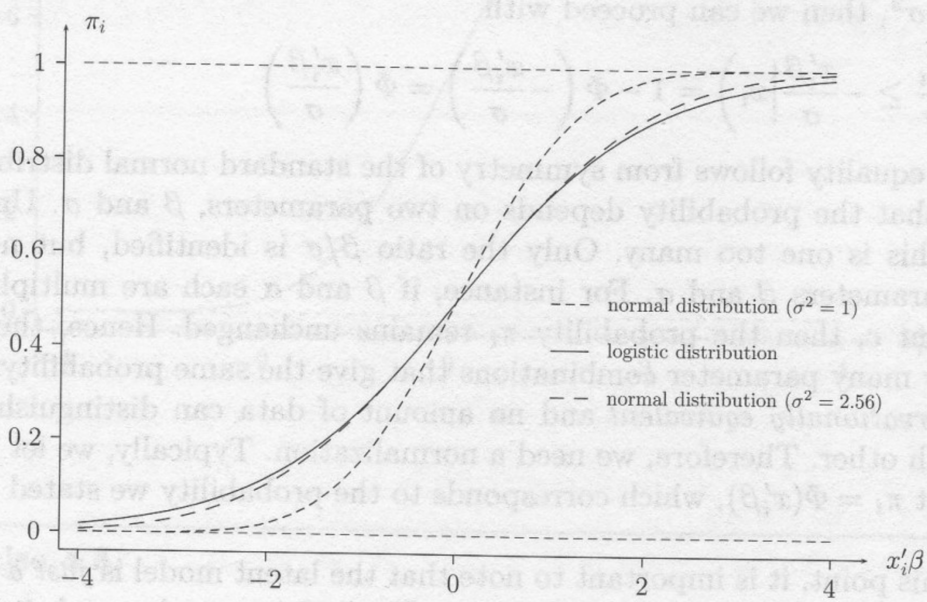
$$F \equiv L(z) = \frac{1}{1 + e^{-z}}$$

Both distributions are symmetric:

$$1 - F(-z) = F(z)$$

and therefore: $y_i = F(x_i'\beta) + v_i$

Fig. 4.2. *Comparison of Probit and Logit Model*



Both distributions are very similar

$$\Phi(z) \approx L\left(\frac{\pi}{\sqrt{3}}z\right)$$

Marginal probability effect

partial effect of x_i on y_i

$$MPE_i = \frac{\partial F(x_i'\beta)}{\partial x_i} = \phi(x_i'\beta)\beta$$

$\Rightarrow$ effect depends on the level of x_i

Maximum likelihood (ML) estimator

log-likelihood function for a symmetric distribution:

$$\log L(\beta) = \sum_{i=1}^{N} y_i \log F(x_i'\beta) + (1 - y_i) \log[1 - F(x_i'\beta)]$$

Differentiation with respect to β yields the first order condition:

$$s(\hat{\beta}) = \sum_{i=1}^{N} \frac{e_i f(x_i'\hat{\beta})}{F(x_i'\hat{\beta})(1 - F(x_i'\hat{\beta}))} x_i = 0$$

where $e_i = y_i - F(x_i'\hat{\beta})$

Nonlinear system of K equations: Iterative algorithm

Estimator is equivalent to nonlinear LS with heteroskedasticn errors

Goodness of fit

(i) McFadden R^2:

$$\text{MF-}R^2 = 1 - \frac{\log L(\hat{\beta})}{\log L(\beta = 0)}$$

(ii) forecasting y_i: Let

$$\hat{y}_i = \begin{cases} 1 & \text{if } F(x_i'\hat{\beta}) > 0.5 \text{ or } x_i'\hat{\beta} > 0, \\ 0 & \text{otherwise} \end{cases}$$

frequency of wrong forecasts:

$$\frac{n_{01} + n_{10}}{n} = \frac{\sum_{i=1}^{n} (y_i - \widehat{y_i})^2}{n}$$

$\Rightarrow R^2$ based on the number of wrong forecasts

14.3 Classification

Let F_i denote the estimated probability for $y_i = 1$. The optimal assignment to the unknown alternatives $\{0, 1\}$ is $\widehat{y_i} = 1$ if $F_i > 0.5$.

This classification rule works poorly if F_i is small. Assume that $x_i \sim \mathrm{U}[0,1]$ and

$$y_i^* = -2 + 2x + u_i$$

then the probability for $y_i = 1$ is 0.2, but in the sample, no unit value is predicted!

One may calibrate the threshold to reduce the classification error such that

$$\sum_{i=1}^{n} 1(\widehat{F_i} > \tau) = \sum_{i=1}^{n} y_i$$

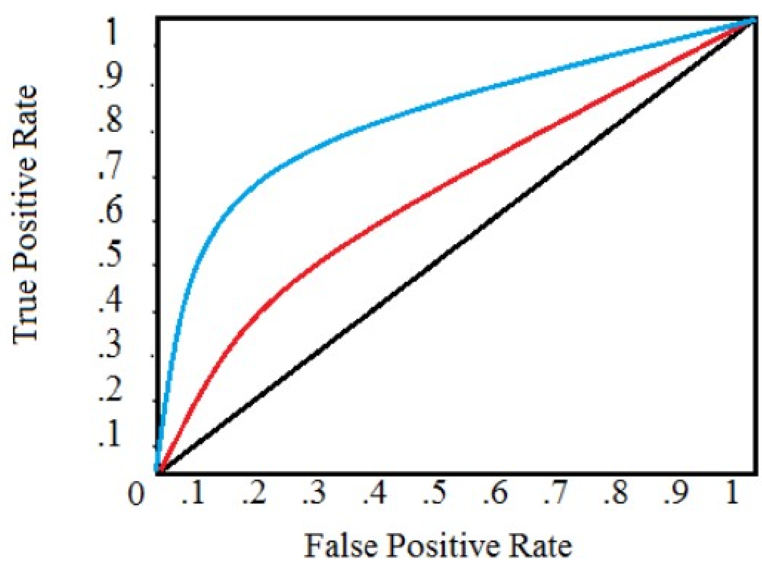
⇒ match the unconditional probabilities.

Trade-off between the two types of misclassification

Useful tool: ROC curve (true positive vs. false positive) If τ is decreased → more ONEs. These can be correct and false detections.

A classification blue is uniformly better than red if ROC is always above ROC

⇒ maximize the area below the ROC curve



The target of the Probit/Logit estimator is $P(y_i = 1) = F(x_i'\beta)$. The optimal estimator of the probability coincides with the efficient estimator of β.

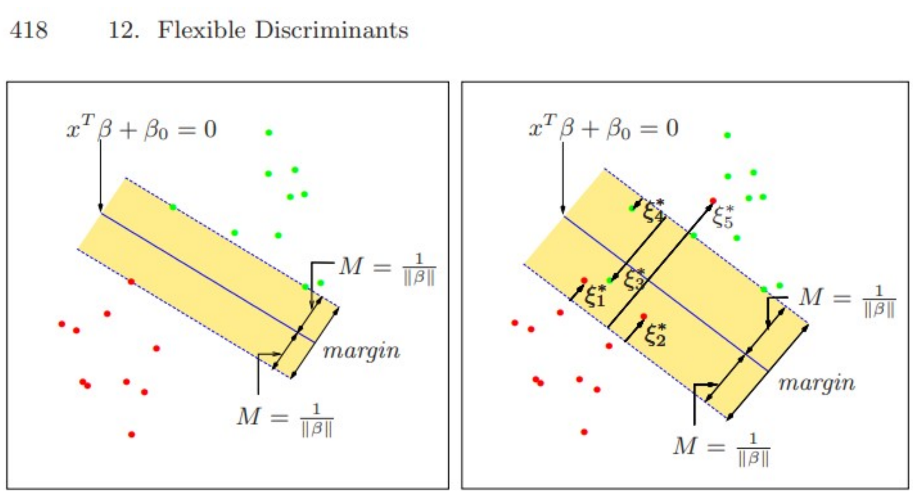
The classification problem seeks an "optimal" estimator for y_i based on the indicator function $\widehat{y_i}$ by minimizing some combination of the (error rates):

$$\text{False Positive} = \sum_i y_i(1-\widehat{y_i})/\sum_i y_i \quad \text{and}$$
$$\text{False Negative} = \sum_i (1-y_i)\widehat{y_i}/\sum_i (1-y_i)$$

Note that $F(x_i'\beta) > \tau$ is equivalent to $x_i'\beta > \tau^*$ with $\tau^* = F^{-1}(\tau)$. $\Rightarrow$ distribution not relevant for classification

Support vector classifier: Maximize M subject to:

$$(2y_i - 1)(x_i'\beta) \geq M(1-\xi_i)$$
$$\xi_i > 0, \quad \sum \xi_i \leq C$$
$$\beta'\beta = 1$$



14.4 Sample selection model

$$\text{Regression model:} \quad y_i = x_{1i}'\beta_1 + u_{1i} \quad \text{if } h_i = 1$$
$$\text{Selection rule:} \quad h_i^* = x_{2i}'\beta_2 + u_{2i} \quad \text{with } E(u_{2i}^2) = 1$$

$$h_i = \begin{cases} 1 & \text{if } h_i^* > 0 \quad \text{observed} \\ 0 & \text{otherwise} \quad \text{not observed} \end{cases}$$

Equivalent to the Tobit model if:

$$x_{1i} = x_{2i}, \quad \beta_1/\sigma = \beta_2, \quad u_{1i}/\sigma = u_{2i}$$

truncated joint density

$$E(y_i | y_i \text{ observed}) = x_{1i}'\beta + \varrho\sigma\lambda_i$$

where $\varrho = E(u_{1i}u_{2i})/\sigma$ and

$$\lambda_i = \frac{\phi(x_{2i}'\beta_2)}{\Phi(x_{2i}'\beta_2)}$$

Heckman estimator

First step: Probit estimator

$$\tilde{y}_i^* = x_i'\tilde{\beta} + u_i$$

where $\tilde{\beta} = \beta/\sigma$ and

$$y_i = \begin{cases} 1 & \text{if } \tilde{y}_i^* > 0 \\ 0 & \text{otherwise} \end{cases}$$

Second step: augmented regression:

$$\lambda_i = \frac{\phi(x_i'\tilde{\beta})}{\Phi(x_i'\tilde{\beta})}$$

and

$$y_i|y_i^* > 0 = x_i'\beta + \sigma\hat{\lambda}_i + \nu_i$$

Standard errors are biased

ML estimator is available

15 Causal Inference

15.1 Experiments and Treatment Effects

Causal effects as measured in (double blind) clinical trials

Separation into two groups a) with treatment b) without treatment (placebo)

Quasi-experiment: because of external events the treatment of some individual occurs as if it is random

Let $Y(X)$ denote the (potential) outcome variable, depending on the binary treatment indicator X_i:

$$\begin{aligned} Y_i|X_i = 1: &\ \text{outcome with treatment} \\ Y_i|X_i = 0: &\ \text{outcome without treatment} \end{aligned}$$

Average causal effect: $E(Y_i|X_i = 1) - E(Y_i|X_i = 0)$

Problem: only one of the two possible outcomes is observed the other is counterfactual

Regression based analysis of treatment effects

a) **Difference estimator**

$$Y_i = \beta_0 + \beta_1 X_i + u_i$$

The OLS estimator is equivalent to

$$\widehat{\beta_1} = \frac{1}{n_1} \sum_{i:X_i=1} Y_i - \frac{1}{n_0} \sum_{i:X_i=0} Y_i$$

with $n_1 = \sum X_i$ (number of treated units) and $n_0 = n???n_1$

The estimator is unbiased for random assignment: $E(u_i|X_i = 1) = E(u_i) = 0$

Regression with pre-treatment characteristics W_i

$$\begin{aligned} y_i &= \beta_0 + \beta_1 X_i + \beta_2 W_{1i} + \cdots + \beta_{r+1} W_{ri} + u_i \\ &= \beta_0 + \beta_1 X_i + \beta_2' \mathbf{w}_i + u_i \quad \text{where } \mathbf{w}_i = (W_{1i}, \dots, W_{ri})' \end{aligned}$$

$E(u_i|X_i = 1, w_i) = E(u_i|w_i) = 0$

15.2 Difference-in-Difference (DiD) estimation

"Before and After" comparisons

Example: happiness before and after marriage

estimation by entity-demeaning is equivalent to:

$$Y_{it} = \beta_0 + \underbrace{\beta_1 (t \cdot X_i)}_{\text{treatment effect}} + \beta_2 X_i + \beta_3 t + u_{it}$$

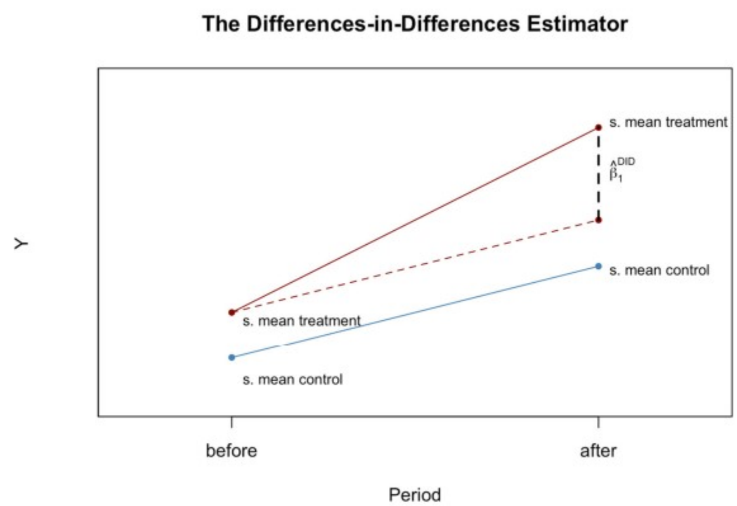
where X_i is the treatment dummy and $t \in \{0, 1\}$ is the period dummy

How does the Fatality Rate (FR) change after a change in the beer tax?

$$FR_{1988} - FR_{1982} = -0.072 - 1.04(tax_{1988}???tax_{1982})$$

where the relevant t-statistic is $-1.04/0.36 = 2.888$ (significant)

The parallel trend assumption



16 Panel Data Models

Two data dimensions:

$$i = 1, 2, \dots, N \quad \text{(cross-section units)}$$
$$t = 1, 2, \dots, T \quad \text{(time periods)}$$

Observations from the same units

Usually: $N >> T$

Observed (controlled) heterogeneity:

$$y_{it} = x'_{it}\beta + \underbrace{z'_i\gamma}_{\alpha_i} + u_{it}$$

⇒ individual characteristics are assumed to be constant in time

dealing with α_i by

- dummy variables
- substracting the means

16.1 Fixed effect model

α_i is "deterministic": Dummy variable model

$$
\begin{aligned}
y_{it} &= x_{it}'\beta + \alpha_i + u_{it} && (16.1) \\
&= x_{it}'\beta + \gamma_2 D_{2i} + \gamma_3 D_{3i} + \cdots + \gamma_n D_{ni} + u_{it} && (16.2)
\end{aligned}
$$

$u_{it} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$

subtracting individual specific means ("entity-demeaned") yields:

$$y_{it} - \bar{y}_i = (x_{it} - \bar{x}_i)'\beta + u_{it} - \bar{u}_i$$

with $\bar{y}_i = T^{-1} \sum_{t=1}^{T} y_{it}$

$\Rightarrow$ individual effects cancel out

both approaches yield the same results

individual and time effects (two-way effects):

$$y_{it} = x_{it}'\beta + \alpha_i + \lambda_t + u_{it}$$

$\Rightarrow$ including also time dummies

16.2 Random effects model

If $\alpha_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\alpha^2)$ then the GLS estimator is obtained from

$$y_{it} - \theta\bar{y}_i = (x_{it} - \theta\bar{x}_i)'\beta + u_{it} - \theta\bar{u}_i$$

$$\text{where} \quad \theta = 1 - \sqrt{\frac{\sigma_u^2}{T\sigma_\alpha^2 + \sigma_u^2}}$$

Estimation of σ_α^2 is based on the fact that

$$\text{var}(\bar{u}_i) = \text{var}\left(\frac{1}{T}\sum_{t=1}^{T} u_{it}\right) = \sigma_\alpha^2 + \frac{1}{T}\sigma_u^2$$

such that

$$\hat{\sigma}_\alpha^2 = \frac{1}{N}\sum_{i=1}^{N} \overbrace{(\bar{y}_i - \bar{x}_i'\hat{\beta})^2}^{\bar{u}_i} - \frac{1}{T}\hat{\sigma}_u^2 \tag{16.3}$$

$$\hat{\sigma}_u^2 = \frac{1}{N(T-1)-k}\sum_{i=1}^{N}\sum_{t=1}^{T}\hat{u}_{it}^2 \tag{16.4}$$

$$\hat{u}_{it} = y_{it} - \bar{y}_i - (x_{it} - \bar{x}_i)'\hat{\beta} \tag{16.5}$$

Goodness of fit

Some software packages compute the dummy-variable R^2, i.e., the regression R^2 that includes the dummies as 'explanatory' variables

The dummy variables do not 'explain' anything but just represent heterogeneity $\Rightarrow$ R^2 is too large

Good practice to present the "within-R^2", that is, the R^2 of the demeaned (within) regression

Interpretation of the panel data model. Assume that α_i is correlated with $\bar{x}_i$ such that $\alpha_i = \lambda\bar{x}_i + \mu_i$ yielding

$$y_i = \underbrace{(x_{it} - \bar{x}_i)'}_{\text{"short-run"}}\beta \;+\; \underbrace{\bar{x}_i'}_{\text{"long-run"}}\gamma + \mu_i + u_{it}$$

where $\gamma = \beta + \lambda$

Estimating this model yields $\hat{\beta}_{FE}$ as an estimator for the "short-run" coefficients. The random effects model implies $\lambda = 0$ and therefore $\beta = \gamma$.

16.3 Model specification

a) Tests for individual specific effects: Null hypothesis:

$$H_0 : \alpha_1 = \alpha_2 = \cdots = \alpha_N = \mu$$

F-statistic:

$$F = \frac{(S_0 - S_1)/(N-1)}{S_1/(NT - N - K)} \sim F(N-1, NT - N - K)$$

where: S_0 and S_1 are RSS of the pooled OLS and FE estimation

b) Hausman test: Deciding between random and fixed effects:

H_0: random effects or $E(x_{it}\alpha_i) = 0$

Under the null hypothesis $\hat{\beta}_{FE}$ and $\tilde{\beta}_{RE}$ are "similar" or $E(\hat{\beta}_{\text{FE}} - \tilde{\beta}_{\text{RE}}) = 0$

Hausman-Wu Test: test of $\delta = 0$ in

$$\tilde{y}_{it} = \tilde{x}'_{it}\beta + (x_{it} - \bar{x}_i)'\delta + \epsilon_{it}$$

with $\tilde{y}_{it}$ and $\tilde{x}_{it}$ as GLS-transformed variables.

17 Econometric Analysis of Time Series

17.1 ARIMA models

Let $y_t = Y_t - \mu$ with $\mu = E(Y_t)$ a demeaned time series for $t = 1, \dots, T$

Autoregressive model of order p:

$$\begin{aligned} \text{AR}(p) \quad & y_t = \theta_1 y_{t-1} + \dots + \theta_p y_{t-p} + \varepsilon_t \\ & \theta(L) y_t = \varepsilon_t \end{aligned}$$

where $\theta(L) = 1 - \theta_1 L - \dots - \theta_p L^p$

Moving-Average model of order q:

$$\begin{aligned} \text{MA}(q) \quad & y_t = \varepsilon_t + \alpha_1 \varepsilon_{t-1} + \dots + \alpha_q \varepsilon_{t-q} \\ & y_t = \alpha(L) \varepsilon_t \end{aligned}$$

where $\alpha(L) = 1 + \alpha_1 L + \dots + \alpha_q L^q$

ARMA (p, q) model:

$$\theta(L) y_t = \alpha(L) \varepsilon_t$$

Autoregressive representation of a ARMA(p, q):

$$\frac{\theta(L)}{\alpha(L)} y_t = \tilde{\theta}(L) y_t = \varepsilon_t$$

$\tilde{\theta}(L)$ can be determined by comparing coefficients from

$$\alpha(L)\tilde{\theta}(L) = \theta(L)$$

Any ARMA(p, q) model can be approximated by a AR(p) model choosing $\tilde{p}$ large enough

A time series is stationary if $\theta(L)$ is invertible, i.e., if it can be factorized as

$$\theta(L) = (1 - \phi_1 L)(1 - \phi_2 L) \cdots (1 - \phi_p L)$$

such that it holds that $|\phi_i| < 1$ for all $1 = i, \dots, p$.

Alternatively, $\theta(L)$ is invertible if the p roots $z_1, \dots, z_p$ of the characteristic equation

$$\theta(z) = 0$$

are all outside the unit circle of the complex plane. For real root we have $z_i = 1/\phi_i$.

17.2 Unit roots

An important special case results if $\phi_1 = 1$, that is,

$$\theta(L)y_t = (1 - L)(1 - \phi_2 L) \cdot (1 - \phi_p L) = \theta^*(L)\Delta y_t = \varepsilon_t$$

where all other roots are outside the unit circle, i.e., Δy_t is stationary.

if $p = 1$, then y_t is white noise (serially uncorrelated) and y_t is a random walk with

$$y_t = y_{t-1} + \varepsilon_t = \varepsilon_t + \varepsilon_{t-1} + \cdots + \varepsilon_1 + y_0$$

such that $\text{var}(y_t) = \text{var}(y_0) + t\sigma^2$

a time series is (weakly) stationary if

$$E(y_t) = 0 \quad \text{and} \quad var(y_t) = \sigma_y^2 \quad \text{for all } t$$

$\Rightarrow$ a random walk with $\theta(L) = 1 - L$ is nonstationary

Unit root test

$\phi_1 = 1$ implies $\theta(1) = 0$ (one root is on the unit circle)

$$\begin{aligned} y_t &= \theta y_{t-1} + \varepsilon_t \\ \Leftrightarrow \Delta y_t &= \underbrace{(\theta - 1)}_{\pi} y_{t-1} + \varepsilon_t \end{aligned}$$

can be tested by using the t-statistic for $\pi = 0$ (Dickey-Fuller statistic):

$$\text{DF-t} = \frac{\hat{\theta} - 1}{\text{se}(\hat{\theta})} = \frac{\hat{\pi}}{\text{se}(\hat{\pi})}$$

Problem: t-statistic is NOT t-distributed

Extension to unknown mean and trend:

$$\begin{aligned} \Delta Y_t &= \delta + \pi y_{t-1} + \varepsilon_t \\ \text{or} \quad \Delta Y_t &= \delta + \gamma t + \pi y_{t-1} + \varepsilon_t \end{aligned}$$

Different critical values for models (i) no constant (ii) with a constant and (iii) with a time trend.

Include a trend if the series seem to evolve around a (linear) time trend

Extension to AR(p) models:

$$\begin{aligned} y_t &= \delta\,[+\gamma t] + \theta_1 y_{t-1} + \theta_2 y_{t-2} + \cdots + \theta_p y_{t-p} + \varepsilon_t \\ \Leftrightarrow \Delta y_t &= \delta\,[+\gamma t] + \pi y_{t-1} + c_1 \Delta y_{t-1} + \cdots + c_{p-1} \Delta y_{t-p} + \varepsilon_t \end{aligned}$$

critical values do NOT depend on the lag-order p

A series is called "integrated of order d" or $y_t \sim I(d)$ if $\Delta^d y_t$ is stationary but Δ^{d-1} is nonstationary

$\Rightarrow$ DF tests are used to determine d empirically

17.3 Cointegration

Assume:

$$Y_t \sim I(1) \quad \text{and} \quad X_t \sim I(1)$$

$\Rightarrow$ In general $Y_t - \beta X_t$ is also $I(1)$

Spurious regression: If y_t and x_t are independent random walks:

- t-values are often significant
- large R^2
- Low Durbin-Watson statistic

Common trend model ("cointegration")

$$\begin{aligned} X_t &= r_t + u_{1t} \sim I(1) \\ Y_t &= \beta r_t + u_{2t} \sim I(1) \\ Y_t - \beta X_t &= u_{2t} - \beta u_{1t} = u_t \sim I(0) \end{aligned}$$

where $r_t \sim I(1)$ (stochastic trend) and u_t is stationary

Estimation and testing

Properties of OLS in cointegrating regressions:

- $\hat{\beta} - \beta$ is $O_p(T^{-1})$ ("super-consistent")
- robust against endogenous X_t
- Efficient only if (i) X_i is exogenous (ii) u_t is serially uncorrelated
- t statistics are generally invalid

Test for cointegration:

1. Step: ADF test of Y_t and X_t
2. Step: ADF test of the residuals $e_t = Y_t - X_t\hat{\beta}$

Critical values depend also on K

Critical Values for the Phillips Z_t Statistic or the Dickey-Fuller t Statistic When Applied to Residuals from Spurious Cointegrating Regression

Number of right-hand variables in regression, excluding trend or constant $(n-1)$	*Sample size* (T)	*Probability that* $(\hat{\rho}-1)/\hat{\sigma}_{\hat{\rho}}$ *is less than entry*						
		0.010	*0.025*	*0.050*	*0.075*	*0.100*	*0.125*	*0.150*
				Case 1				
1	500	−3.39	−3.05	−2.76	−2.58	−2.45	−2.35	−2.26
2	500	−3.84	−3.55	−3.27	−3.11	−2.99	−2.88	−2.79
3	500	−4.30	−3.99	−3.74	−3.57	−3.44	−3.35	−3.26
4	500	−4.67	−4.38	−4.13	−3.95	−3.81	−3.71	−3.61
5	500	−4.99	−4.67	−4.40	−4.25	−4.14	−4.04	−3.94
				Case 2				
1	500	−3.96	−3.64	−3.37	−3.20	−3.07	−2.96	−2.86
2	500	−4.31	−4.02	−3.77	−3.58	−3.45	−3.35	−3.26
3	500	−4.73	−4.37	−4.11	−3.96	−3.83	−3.73	−3.65
4	500	−5.07	−4.71	−4.45	−4.29	−4.16	−4.05	−3.96
5	500	−5.28	−4.98	−4.71	−4.56	−4.43	−4.33	−4.24
				Case 3				
1	500	−3.98	−3.68	−3.42	—	−3.13	—	—
2	500	−4.36	−4.07	−3.80	−3.65	−3.52	−3.42	−3.33
3	500	−4.65	−4.39	−4.16	−3.98	−3.84	−3.74	−3.66
4	500	−5.04	−4.77	−4.49	−4.32	−4.20	−4.08	−4.00
5	500	−5.36	−5.02	−4.74	−4.58	−4.46	−4.36	−4.28

Engle-Granger two-step approach

Error correction representation:

$$Y_t = \delta + \alpha Y_{t-1} + \beta_1 X_{t-1} + \beta_2 X_{t-2} + u_t$$

can be rewritten as

$$\Delta Y_t = \delta + \phi_1 \Delta X_{t-1} + \gamma (Y_{t-1} - \beta X_{t-1}) + u_t$$

$$\text{where} \quad \phi_1 = -\beta_2, \quad \gamma = \alpha - 1 < 0, \quad \text{and} \quad \beta = (\beta_1 + \beta_2)/(1 - \alpha)$$

$(Y_{t-1} - \beta X_{t-1}) \sim I(0)$ is called the error correction term

replace β by $\hat{\beta}$ (Engle/Granger 2-step estimator)

Coefficients attached to stationary variables have the usual asymptotic distributions (t-statistics yield valid tests)