Econometrics (M.Sc.)

Prof. Dr. Dominik Liebl

2022-09-06

Contents

Preface

Organization of the Course

All lecture material (videos, eWhiteboard, and this script) can be found at eCampus under the following link:

- https://ecampus.uni-bonn.de/bl.php?id=197335

This lecture script is still under development, so please regularly check for updates.

Literature

It's not must, but you can read any of the usual econometrics textbooks additionally to this script.

- *A guide to modern econometrics*, by M. Verbeek
- *Introduction to econometrics*, by J. Stock and M.W. Watson
  - E-Book: https://bonnus.ulb.uni-bonn.de/SummonRecord/FETCH-bonn_catalog_45089983
- *Econometric theory and methods*, by R. Davidson and J.G. MacKinnon
- *A primer in econometric theory*, by J. Stachurski

- *Econometrics*, by F. Hayashi

Figure 0.1: Creative Commons License

Chapter 1

Introduction to R

This tutorial aims to serve as an introduction to the software package R. Other very good and much more exhaustive tutorials and useful reference-cards can be found at the following links:

- Reference card for R commands (always useful)
- Matlab/R reference card (for those who are more familiar with Matlab)
- The official Introduction to R (very detailed)
- And many more at www.r-project.org (see "Documents")
- An R-package for learning R: www.swirl.com
- An excellent book project which covers also advanced issues such as "writing performant code" and "package development": adv-r.had.co.nz

- Another excellent book: R for Data Science

Some other tutorials:

- Introduction to data science
- Scraping the web using R
- Creating dynamic graphics

Why R?

- R is **free** of charge from: www.r-project.org
- The celebrated IDE **RStudio** for R is also **free** of charge: www.rstudio.com
- R is equipped with one of the most flexible and powerful graphics routines available anywhere.
  For instance, check out one of the following repositories:
  - Clean Graphs
  - R graph catalog
  - Publication Ready Plots
- Today, R is the de-facto standard for statistical science.

1.1 Short Glossary

Lets start the tutorial with a (very) short glossary:

- **Console**: The thing with the "$>$" sign at the beginning.
- **Script file**: An ordinary text file with suffix "**.R**". For instance, **yourFavoritFileName.R**.
- **Working directory**: The file-directory you are working in. Useful commands: with `getwd()` you get the location of your current working directory and `setwd()` allows you to set a new location for it.
- **Workspace**: This is a hidden file (stored in the working directory), where all objects you use (e.g., data, matrices, vectors, variables, functions, etc.) are stored. Useful commands: `ls()` shows all elements in our current workspace and `rm(list=ls())` deletes all elements in our current workspace.

1.2 First Steps

A good idea is to use a script file such as **yourFavoritFileName.R** in order to store your R commands. You can send single lines or marked regions of your R-code to the console by pressing the keys **STRG+ENTER**.

To begin with baby steps, do some simple computations:

```
2+2 # and all the others: *,/,-,^2,^3,...
#> [1] 4
```

Note: Everything that is written after the #-sign is ignored by R, which is very useful to comment your code.

The **assignment operator** will be your most often used tool. Here an example to create a **scalar** variable:

```
x <- 4
x
#> [1] 4
4 -> x # possible but unusual
x
#> [1] 4
```

Note: The R community loves the <- assignment operator, which is a very unusual syntax. Alternatively, you can use the = operator.

And now a more interesting object - a **vector**:

```
y <- c(2,7,4,1)
y
#> [1] 2 7 4 1
```

The command `ls()` shows the total content of your current workspace, and the command `rm(list=ls())` deletes all elements of your current workspace:

```
ls()[1:5] # only the first 5 elements
#> [1] "a"          "A"          "alpha"      "auto.data" "b"
rm(list=ls())
ls()
#> character(0)
```

Note: RStudio's **Environment** pane also lists all the elements in your current workspace. That is, the command `ls()` becomes a bit obsolete when working with RStudio.

Let's try how we can compute with vectors and scalars in R.

```
x <- 4
y <- c(2,7,4,1)

x*y # each element in y (vector) is multiplied by x (scalar).
#> [1]  8 28 16  4
y*y # this is a term by term product of the elements in y
#> [1]  4 49 16  1
```

Performing vector multiplications as you might expect from your last math-course, e.g., an outer product: $y\,y^\top$:

```
y %*% t(y)
#>      [,1] [,2] [,3] [,4]
#> [1,]    4   14    8    2
#> [2,]   14   49   28    7
```

```
#> [3,]    8   28   16    4
#> [4,]    2    7    4    1
```

Or an inner product $y^\top y$:

```
t(y) %*% y
#>      [,1]
#> [1,]   70
```

Note: Sometimes, R's treatment of vectors can be annoying. The product `y %*% y` is treated as the product `t(y) %*% y`.

The term-by-term execution as in the above example, `y*y`, is actually a central strength of R. We can conduct many operations **vector-wisely**:

```
y^2
#> [1]  4 49 16  1
log(y)
#> [1] 0.6931472 1.9459101 1.3862944 0.0000000
exp(y)
#> [1]    7.389056 1096.633158   54.598150    2.718282
y-mean(y)
#> [1] -1.5  3.5  0.5 -2.5
(y-mean(y))/sd(y) # standardization
#> [1] -0.5669467  1.3228757  0.1889822 -0.9449112
```

This is a central characteristic of so called matrix based languages like R (or Matlab). Other programming languages often have to use **loops** instead:

```r
N <- length(y)
1:N

y.sq <- numeric(N)
y.sq

for(i in 1:N){
  y.sq[i] <- y[i]^2
  if(i == N){
    print(y.sq)
  }
}
```

The `for()`-loop is the most common loop. But there is also a `while()`-loop and a `repeat()`-loop. However, loops in R can be rather slow, therefore, try to avoid them!

Useful commands to produce **sequences** of numbers:

```r
1:10
-10:10
?seq # Help for the seq()-function
seq(from=1, to=100, by=7)
```

Using the sequence command `1:16`, we can go for our first **matrix**:

```r
?matrix
A <- matrix(data=1:16, nrow=4, ncol=4)
A
#>      [,1] [,2] [,3] [,4]
```

```
#> [1,]    1    5    9   13
#> [2,]    2    6   10   14
#> [3,]    3    7   11   15
#> [4,]    4    8   12   16
A <- matrix(1:16, 4, 4)
```

Note that a matrix has always two **dimensions**, but a vector has only one dimension:

```
dim(A)    # Dimension of matrix A?
#> [1] 4 4
dim(y)    # dim() does not operate on vectors.
#> NULL
length(y) # Length of vector y?
#> [1] 4
```

Lets play a bit with the matrix `A` and the vector `y`. As we have seen in the loop above, the `[]`-operator **selects elements** of vectors and matrices:

```
A[,1]
A[4,4]
y[c(1,4)]
```

This can be done on a more **logical** basis, too. For example, if you want to know which elements in the first column of matrix `A` are strictly greater than 2:

```
A[,1][A[,1]>2]
#> [1] 3 4
```

```
# Note that this give you a boolean vector:
A[,1]>2
#> [1] FALSE FALSE  TRUE  TRUE

# And you can use it in a non-sense relation, too:
y[A[,1]>2]
#> [1] 4 1
```

Note: Logical operations return so-called **boolean** objects, i.e., either a `TRUE` or a `FALSE`. For instance, if we ask R whether `1>2` we get the answer `FALSE`.

1.3 Further Data Objects

Besides classical data objects such as scalars, vectors, and matrices there are three further data objects in R:

1. The **array**: As a matrix but with more dimensions. Here is an example of a $2 \times 2 \times 2$-dimensional `array`:

```
myFirst.Array <- array(c(1:8), dim=c(2,2,2)) # Take a look at it!
```

2. The **list**: In `lists` you can organize different kinds of data. E.g., consider the following example:

```
myFirst.List <- list(
  "Some_Numbers" = c(66, 76, 55, 12, 4, 66, 8, 99),
```

```
  "Animals"      = c("Rabbit", "Cat", "Elefant"),
  "My_Series"    = c(30:1)
)
```

A very useful function to find specific values and entries within lists is the `str()`-function:

```
str(myFirst.List)
#> List of 3
#>  $ Some_Numbers: num [1:8] 66 76 55 12 4 66 8 99
#>  $ Animals     : chr [1:3] "Rabbit" "Cat" "Elefant"
#>  $ My_Series   : int [1:30] 30 29 28 27 26 25 24 23 22 21 ...
```

3. The **data frame**: A `data.frame` is a `list`-object but with some more formal restrictions (e.g., equal number of rows for all columns). As indicated by its name, a `data.frame`-object is designed to store data:

```
myFirst.Dataframe <- data.frame(
  "Credit_Default"   = c( 0, 0, 1, 0, 1, 1),
  "Age"              = c(35,41,55,36,44,26),
  "Loan_in_1000_EUR" = c(55,65,23,12,98,76)
)
# Take a look at it!
```

1.4 Simple Regression Analysis using R

Alright, let's do some statistics with real data. You can download the data HERE. Save it on your computer, at a place where you can find it, and give the path (e.g. `"C:\textbackslash path\textbackslash autodata.csv"`, which references to the data, to the *file*-argument of the function `read.csv()`:

```
# ATTENTION! YOU HAVE TO CHANGE "\" TO "/":
auto.data <- read.csv(file   = "C:/your_path/autodata.csv",
                      header = TRUE)
head(auto.data)
```

If you have problems to read the data into R, go on with these commands. (For this you need a working internet connection!):

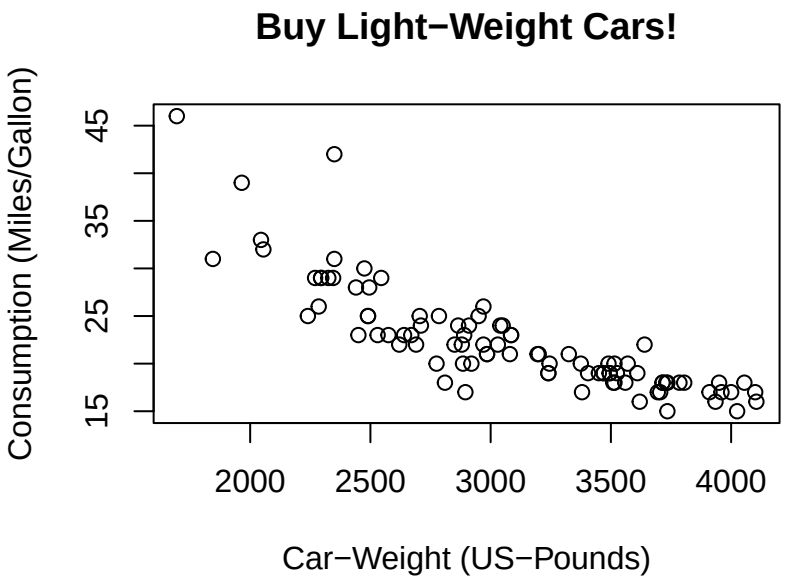
```
# install.packages("readr")
library("readr")
auto.data <- suppressMessages(
  read_csv(
  file = "https://cdn.rawgit.com/lidom/Teaching_Repo/bc692b56/autodata.
  col_names = TRUE)
)
# head(auto.data)
```

You can select specific variables of the `auto.data` using the `$`-operator:

```
gasolin.consumption      <- auto.data$MPG.city
car.weight               <- auto.data$Weight
## Take a look at the first elements of these vectors:
head(cbind(gasolin.consumption,car.weight))
#>      gasolin.consumption car.weight
#> [1,]                  25       2705
#> [2,]                  18       3560
#> [3,]                  20       3375
#> [4,]                  19       3405
#> [5,]                  22       3640
#> [6,]                  22       2880
```

This is how you can produce your first plot:

```
## Plot the data:
plot(y=gasolin.consumption, x=car.weight,
     xlab="Car-Weight (US-Pounds)",
     ylab="Consumption (Miles/Gallon)",
     main="Buy Light-Weight Cars!")
```



As a first step, we might assume a simple kind of linear relationship between the variables `gasolin.consumption` and `car.weight`. Let us assume that the data was generated by the following simple regression model:

$$y_i = \alpha + \beta_1 x_i + \varepsilon_i, \quad i = 1, \ldots, n$$

where y_i denotes the gasoline-consumption, x_i the weight of car i, and ε_i is a mean zero constant variance noise term. (This is clearly a non-sense model!)

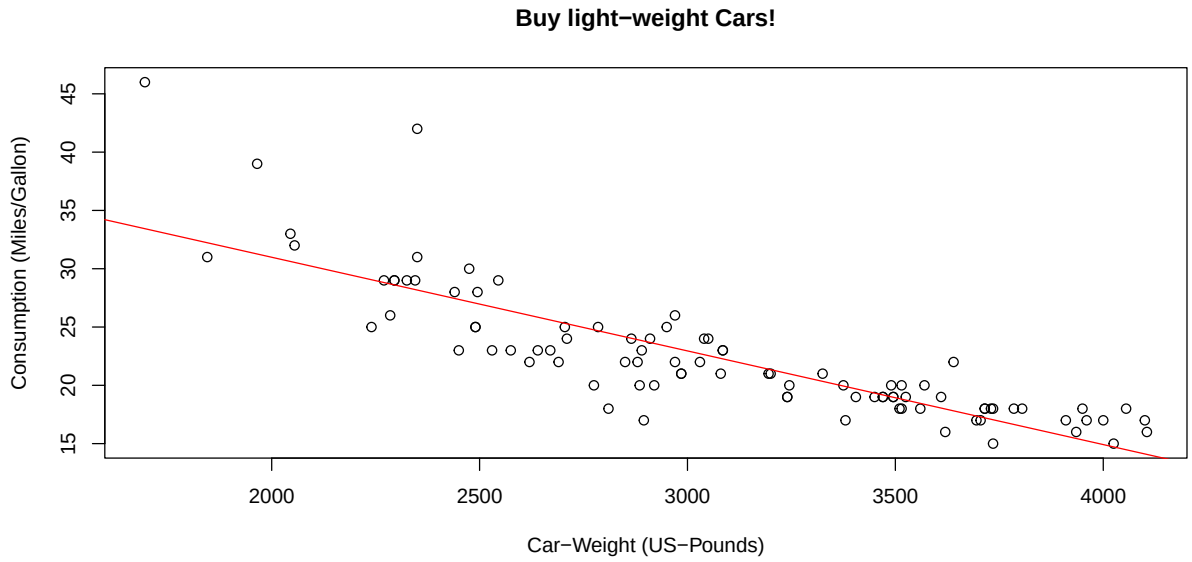
The command `lm()` computes the estimates of this linear regression model. The command (in fact it's a *method*) `summary()` computes further quantities of general interest from the *object* that was returned from the `lm()` function.

```
lm.result   <- lm(gasolin.consumption~car.weight)
lm.summary  <- summary(lm.result)
lm.summary
#>
#> Call:
#> lm(formula = gasolin.consumption ~ car.weight)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -6.7946 -1.9711  0.0249  1.1855 13.8278
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 47.048353   1.679912   28.01   <2e-16 ***
#> car.weight  -0.008032   0.000537  -14.96   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 3.038 on 91 degrees of freedom
#> Multiple R-squared:  0.7109, Adjusted R-squared:  0.7077
#> F-statistic: 223.8 on 1 and 91 DF,  p-value: < 2.2e-16
```

Of course, we want to have a possibility to access all the quantities computed so far, e.g., in order to plot the results. This can be done as following:

```
## Accessing the computed quantities
names(lm.summary) ## Alternatively: str(lm.summary)
#>  [1] "call"          "terms"         "residuals"     "coefficients"  "
```

```
#>  [6] "sigma"         "df"            "r.squared"     "adj.r.squared
#> [11] "cov.unscaled"

alpha <- lm.summary$coefficients[1]
beta  <- lm.summary$coefficients[2]

## Plot all:
plot(y=gasolin.consumption, x=car.weight,
     xlab="Car-Weight (US-Pounds)",
     ylab="Consumption (Miles/Gallon)",
     main="Buy light-weight Cars!")
abline(a=alpha,
       b=beta, col="red")
```



1.5 Programming in R

Let's write, i.e., program our own R-function for estimating linear regression models. In order to be able to validate our function, we start with **simulating data** for which we then *know* all true parameters. Simulating data is like being the "Data-God": For instance, we generate realizations of the error term ε_i, i.e., something which we *never* observe in real data.

Let us consider the following multiple regression model:

$$y_i = \beta_1 + \beta_2 x_{2i} + \beta_3 x_{3i} + \varepsilon_i, \quad i = 1, \ldots, n,$$

where ε_i is a heteroscedastic error term

$$\varepsilon_i \sim N(0, \sigma_i^2), \quad \sigma_i = |x_{3i}|,$$

and where for all $i = 1, \ldots, n = 50$:

- $x_{2i} \sim N(10, 1.5^2)$
- x_{3i} comes from a t-distribution with 5 degrees of freedom and non-centrality parameter 2

```
set.seed(109) # Sets the "seed" of the random number generators:
n   <- 50     # Number of observations

## Generate two explanatory variables plus an intercept-variable:
X.1 <- rep(1, n)                    # Intercept
X.2 <- rnorm(n, mean=10, sd=1.5) # Draw realizations form a normal distr.
X.3 <- rt(n, df=5, ncp=2)           # Draw realizations form a t-distr.
X   <- cbind(X.1, X.2, X.3)         # Save as a Nx3-dimensional data matrix.
```

OK, we have regressors, i.e., data that we also have in real data sets.

Now we define the elements of the β-vector. Be aware of the difference: In real data sets we do not know the true β-vector, but try to estimate it. However, when simulating data, we determine (as "Data-Gods") the true β-vector and can compare our estimate $\hat{\beta}$ with the true β:

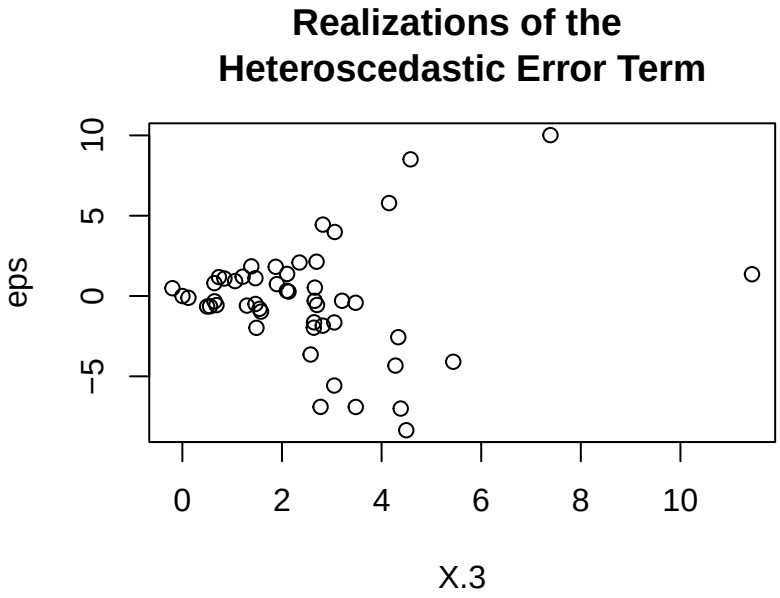
```
## Define the slope-coefficients
beta.vec  <- c(1,-5,5)
```

We still need to simulate realizations of the dependent variable y_i. Remember that $y_i = \beta_1 x_{1i} + \beta_1 x_{2i} + \beta_3 x_{3i} + \varepsilon_i$. That is, we only need realizations from the error terms ε_i in order to compute the realizations from y_i. This is how you can simulate realizations from the heteroscedastic error terms ε_i:

```
## Generate realizations from the heteroscadastic error term
eps       <- abs(X.3) * rnorm(n, mean=0, sd=1)
```

Take a look at the heteroscedasticity in the error term:

```
plot(y=eps, x=X.3,
     main="Realizations of the \nHeteroscedastic Error Term")
```

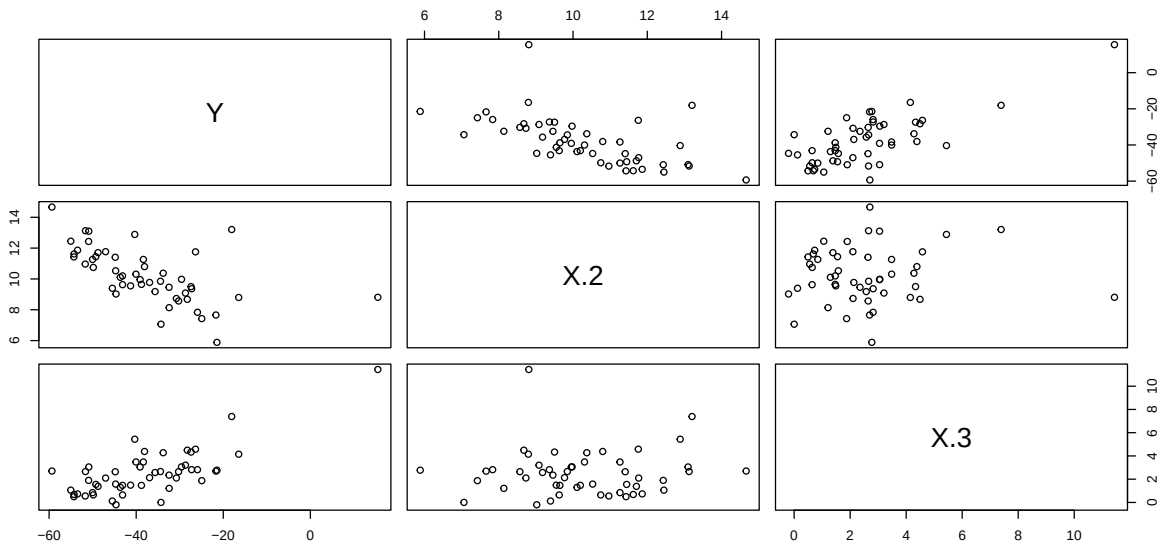


With the (pseudo-random) realizations from ε_i, we can finally generate realizations from the dependent variable y_i:

```
## Dependent variable:
y   <- X %*% beta.vec + eps
```

Let's take a look at the data:

```
mydata    <- data.frame("Y"=y, "X.1"=X.1, "X.2"=X.2, "X.3"=X.3)
pairs(mydata[,-2]) # The '-2' removes the intercept variable "X.1"
```



Once we have data, we can compute the OLS estimate of the true β vector. Remember the formula:

$$\hat{\beta} = (X^\top X)^{-1} X^\top y$$

In R-Code this is: $(X^\top X)^{-1}$ =`solve(t(X) %*% X)`, i.e.:

```
## Computation of the beta-Vector:
beta.hat <- solve(t(X) %*% X) %*% t(X) %*% y
beta.hat
#>          [,1]
#> X.1 -2.609634
#> X.2 -4.692735
#> X.3  5.078342
```

Well done. Using the above lines of code we can easily program our own `myOLSFun()` function!

```
myOLSFun <- function(y, x, add.intercept=FALSE){

  ## Number of Observations:
  n         <- length(y)

  ## Add an intercept to x:
  if(add.intercept){
    Intercept <- rep(1, n)
    x         <- cbind(Intercept, x)
  }

  ## Estimation of the slope-parameters:
  beta.hat.vec <- solve(t(x) %*% x) %*% t(x) %*% y

  ## Return the result:
  return(beta.hat.vec)
}

## Run the function:
myOLSFun(y=y, x=X)
#>           [,1]
#> X.1 -2.609634
#> X.2 -4.692735
#> X.3  5.078342
```

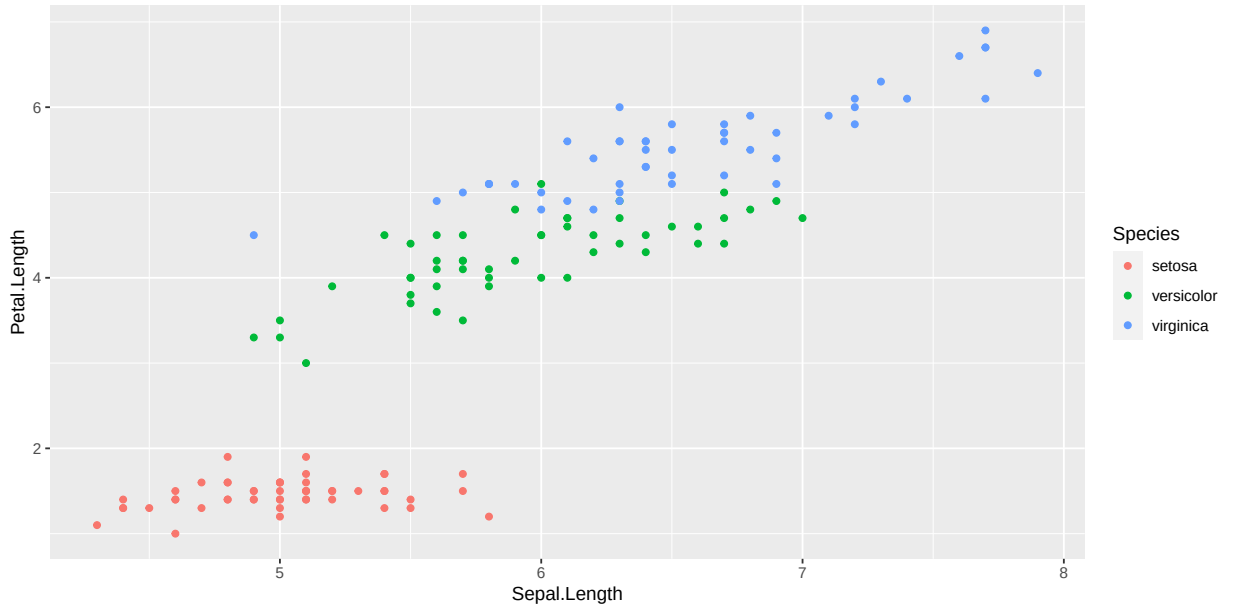
Can you extend the function for the computation of the covariance matrix of the slope-estimates, several measures of fits (R^2, adj.-R^2, etc.), t-tests, ...?

1.6 R-packages

One of the best features in R are its contributed packages. The list of all packages on CRAN is impressive! Take a look at it HERE

For instance, nice plots can be produced using the R-package is `ggplot2`. You can find an intro do this package HERE.

```
# install.packages("ggplot2")
library("ggplot2")

qplot(Sepal.Length, Petal.Length, data = iris, color = Species)
```



Of course, `ggplot2` concerns "only" plotting, but you'll find R-packages for almost any statistical method out there.

1.7 Tidyverse

The `tidyverse` package is a collection of packages that lets you import, manipulate, explore, visualize and model data in a harmonized and consistent way which helps you to be more productive.

Installing the `tidyverse` package:

```
install.packages("tidyverse")
```

To use the `tidyverse` package load it using the `library()` function:

```
library(tidyverse)
```

Chick Weight Data

R comes with many datasets installed. We will use the `ChickWeight` dataset to learn about the tidyverse. The help system gives a basic summary of the experiment from which the data was collect:

> *"The body weights of the chicks were measured at birth and every second day thereafter until day 20. They were also measured on day 21. There were four groups of chicks on different protein diets."*

You can get more information, including references by typing:

```
help("ChickWeight")
```

The Data: There are 578 observations (rows) and 4 variables:

- `Chick` – unique ID for each chick.
- `Diet` – one of four protein diets.
- `Time` – number of days since birth.
- `weight` – body weight of chick in grams.

Note: `weight` has a lower case `w` (recall R is case sensitive).

Store the data locally:

```
ChickWeight %>%
  select(Chick, Diet, Time, weight) %>%
  arrange(Chick, Diet, Time) %>%
  write_csv("ChickWeight.csv")
```

First we will import the data from a file called `ChickWeight.csv` using the `read_csv()` function from the `readr` package (part of the `tidyverse`). The first thing to do, outside of R, is to open the file `ChickWeight.csv` to check what it contains and that it makes sense. Now we can import the data as follows:

```
CW <- read_csv("ChickWeight.csv")
```

If all goes well then the data is now stored in an R object called `CW`. If you get the following error message then you need to change the working directory to where the data is stored.

```
Error: 'ChickWeight.csv' does not exist in current
working directory ...
```

Changing the working directory: In RStudio you can use the menu bar ("Session - Set Working Directory - Choose Directory..."). Alternatively, you can use the function `setwd()`.

Looking at the Dataset: To look at the data type just type the object (dataset) name:

```
CW
#> # A tibble: 578 x 4
#>    Chick  Diet  Time weight
#>    <dbl> <dbl> <dbl>  <dbl>
#>  1    18     1     0     39
#>  2    18     1     2     35
#>  3    16     1     0     41
#>  4    16     1     2     45
#>  5    16     1     4     49
#>  6    16     1     6     51
#>  7    16     1     8     57
#>  8    16     1    10     51
#>  9    16     1    12     54
#> 10    15     1     0     41
#> # ... with 568 more rows
```

If there are too many variables then not all them may be printed. To overcome this issue we can use the `glimpse()` function which makes it possible to see every column in your dataset (called a "data frame" in R speak).

```
glimpse(CW)
#> Rows: 578
#> Columns: 4
#> $ Chick  <dbl> 18, 18, 16, 16, 16, 16, 16, 16, 16, 15, 15, 15, 15,
#> $ Diet   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
#> $ Time   <dbl> 0, 2, 0, 2, 4, 6, 8, 10, 12, 0, 2, 4, 6, 8, 10, 12,
#> $ weight <dbl> 39, 35, 41, 45, 49, 51, 57, 51, 54, 41, 49, 56, 64,
```

The function `View()` allows for a spread-sheet type of view on the data:

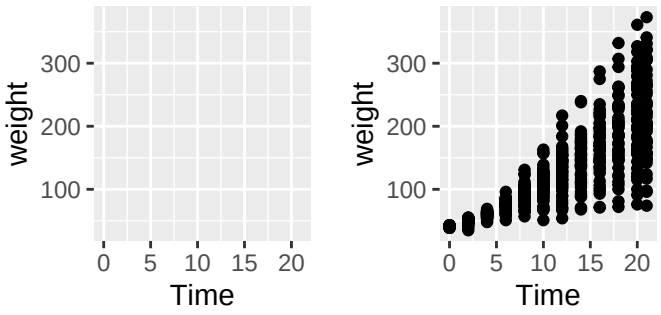
```
View(CW)
```

1.7.1 Tidyverse: Plotting Basics

To **visualise** the chick weight data, we will use the `ggplot2` package (part of the `tidyverse`). Our interest is in seeing how the *weight changes over time for the chicks by diet*. For the moment don't worry too much about the details just try to build your own understanding and logic. To learn more try different things even if you get an error messages.

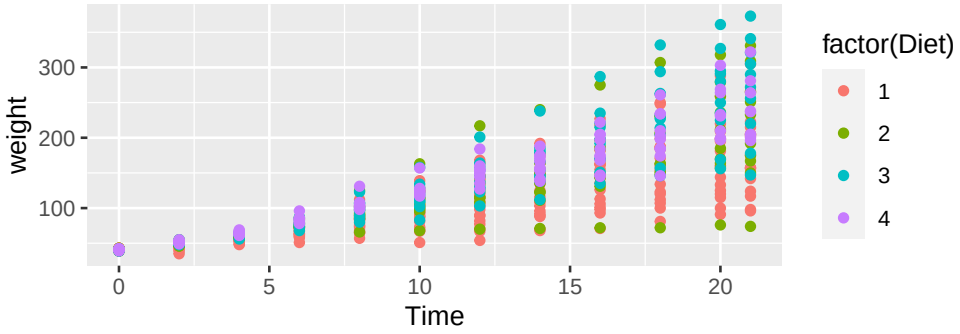
Let's plot the weight data (vertical axis) over time (horizontal axis).

```
# An empty plot (the plot on the left)
ggplot(CW, aes(Time, weight))
# With data (the plot on the right)
ggplot(CW, aes(Time, weight)) + geom_point()
```



Add color for `Diet`. The graph above does not differentiate between the diets. Let's use a different color for each diet.

```
# Adding colour for diet
ggplot(CW,aes(Time,weight,colour=factor(Diet))) +
  geom_point()
```



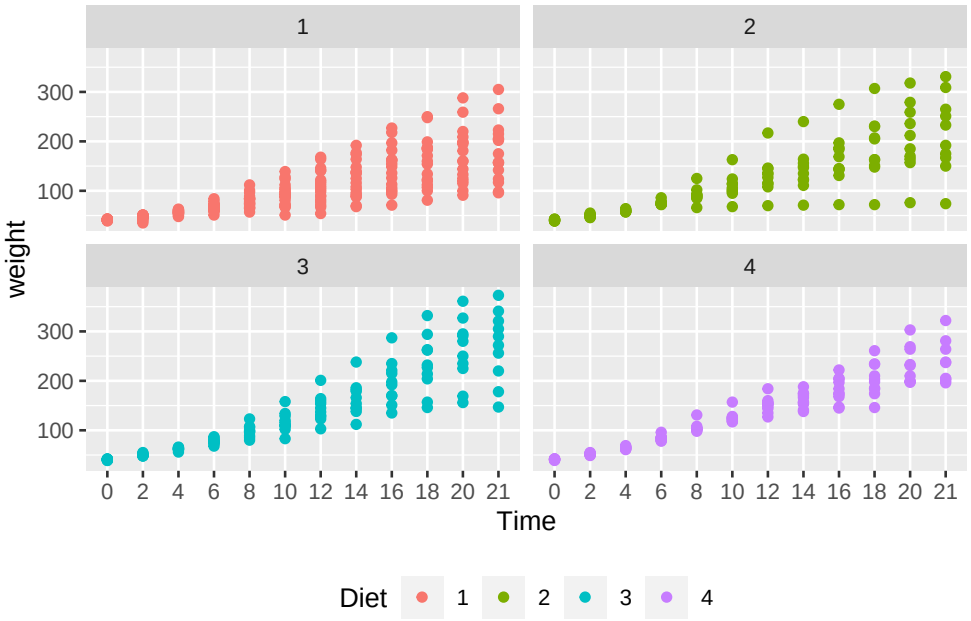
It is difficult to conclude anything from this graph as the points are printed on top of one another (with diet 1 underneath and diet 4 at the top).

Factor Variables: Before we continue, we have to make an important change to the `CW` dataset by making `Diet` and `Time` *factor variables*. This means that R will treat them as categorical variables (see the `<fct>` variables below) instead of continuous variables. It will simplify our coding. The next section will explain the `mutate()` function.

```
CW <- mutate(CW, Diet = factor(Diet))
CW <- mutate(CW, Time = factor(Time))
glimpse(CW)
#> Rows: 578
#> Columns: 4
#> $ Chick  <dbl> 18, 18, 16, 16, 16, 16, 16, 16, 16, 15, 15, 15, 15,
#> $ Diet   <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
#> $ Time   <fct> 0, 2, 0, 2, 4, 6, 8, 10, 12, 0, 2, 4, 6, 8, 10, 12,
#> $ weight <dbl> 39, 35, 41, 45, 49, 51, 57, 51, 54, 41, 49, 56, 64,
```

The `facet_wrap()` function: To plot each diet separately in a grid using `facet_wrap()`:

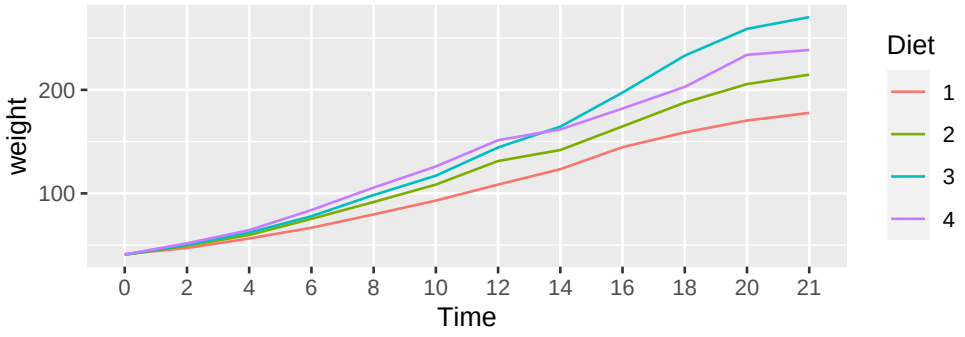
```
# Adding jitter to the points
ggplot(CW, aes(Time, weight, colour=Diet)) +
  geom_point() +
  facet_wrap(~Diet) +
  theme(legend.position = "bottom")
```



Interpretation: Diet 4 has the least variability but we can't really say anything about the mean effect of each diet although diet 3 seems to have the highest.

Next we will plot the **mean changes** over time for each diet using the `stat_summary()` function:

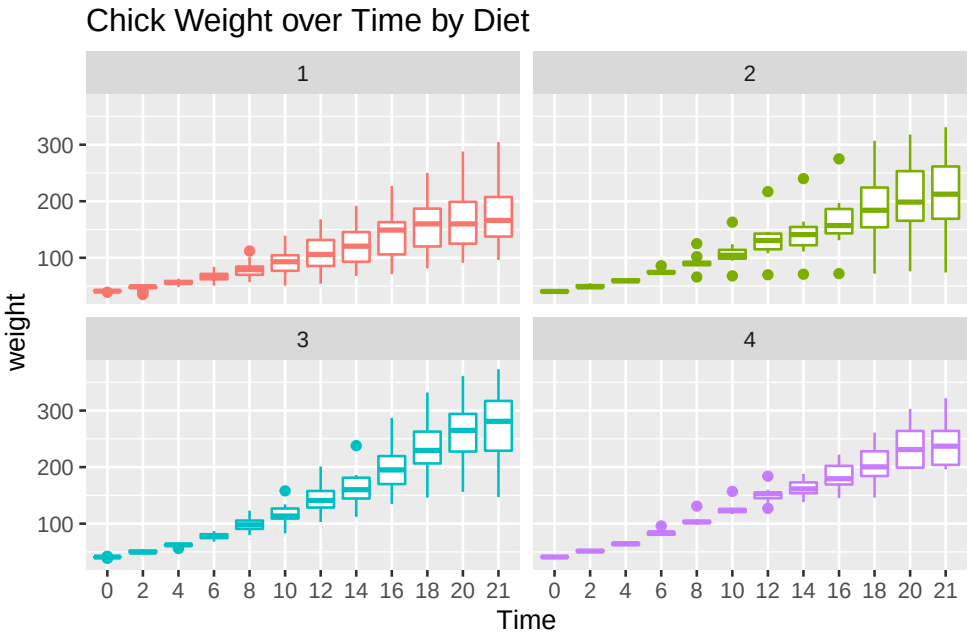
```
ggplot(CW, aes(Time, weight,
               group=Diet, colour=Diet)) +
  stat_summary(fun="mean", geom="line")
```



Interpretation: We can see that diet 3 has the highest mean weight gains by the end of the experiment. However, we don't have any information about the variation (uncertainty) in the data.

To see variation between the different diets we use `geom_boxplot` to plot a box-whisker plot. A note of caution is that the number of chicks per diet is relatively low to produce this plot.

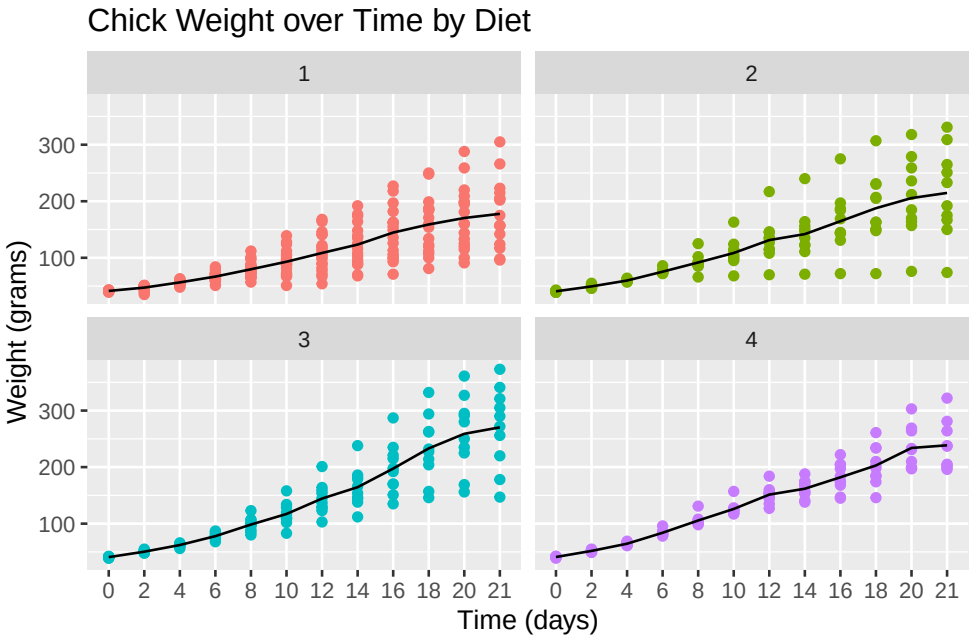
```
ggplot(CW, aes(Time, weight, colour=Diet)) +
  facet_wrap(~Diet) +
  geom_boxplot() +
  theme(legend.position = "none") +
  ggtitle("Chick Weight over Time by Diet")
```



Interpretation: Diet 3 seems to have the highest "average" weight gain but it has more variation than diet 4 which is consistent with our findings so far.

Let's finish with a plot that you might include in a publication.

```
ggplot(CW, aes(Time, weight, group=Diet,
                             colour=Diet)) +
  facet_wrap(~Diet) +
  geom_point() +
  # geom_jitter() +
  stat_summary(fun="mean", geom="line",
               colour="black") +
  theme(legend.position = "none") +
  ggtitle("Chick Weight over Time by Diet") +
  xlab("Time (days)") +
  ylab("Weight (grams)")
```



1.7.2 Tidyverse: Data Wrangling Basics

In this section we will learn how to wrangle (manipulate) datasets using the `tidyverse` package. Let's start with the `mutate()`, `select()`, `rename()`, `filter()` and `arrange()` functions.

`mutate()`: Adds a new variable (column) or modifies an existing one. We already used this above to create factor variables.

```
# Added a column
CWm1 <- mutate(CW, weightKg = weight/1000)
CWm1
#> # A tibble: 578 x 5
#>   Chick Diet  Time  weight weightKg
```

```
#>   <dbl> <fct> <fct>  <dbl>    <dbl>
#> 1    18 1     0         39    0.039
#> 2    18 1     2         35    0.035
#> 3    16 1     0         41    0.041
#> # ... with 575 more rows
# Modify an existing column
CWm2 <- mutate(CW, Diet = str_c("Diet ", Diet))
CWm2
#> # A tibble: 578 x 4
#>   Chick Diet   Time  weight
#>   <dbl> <chr>  <fct>  <dbl>
#> 1    18 Diet 1 0         39
#> 2    18 Diet 1 2         35
#> 3    16 Diet 1 0         41
#> # ... with 575 more rows
```

`select()`: Keeps, drops or reorders variables.

```
# Drop the weight variable from CWm1 using minus
select(CWm1, -weight)
#> # A tibble: 578 x 4
#>   Chick Diet  Time  weightKg
#>   <dbl> <fct> <fct>    <dbl>
#> 1    18 1     0        0.039
#> 2    18 1     2        0.035
#> 3    16 1     0        0.041
#> # ... with 575 more rows
# Keep variables Time, Diet and weightKg
```

```
select(CWm1, Chick, Time, Diet, weightKg)
#> # A tibble: 578 x 4
#>   Chick Time  Diet  weightKg
#>   <dbl> <fct> <fct>    <dbl>
#> 1    18 0     1        0.039
#> 2    18 2     1        0.035
#> 3    16 0     1        0.041
#> # ... with 575 more rows
```

`rename()`: Renames variables whilst keeping all variables.

```
rename(CW, Group = Diet, Weight = weight)
#> # A tibble: 578 x 4
#>   Chick Group Time  Weight
#>   <dbl> <fct> <fct>  <dbl>
#> 1    18 1     0         39
#> 2    18 1     2         35
#> 3    16 1     0         41
#> # ... with 575 more rows
```

`filter()`: Keeps or drops observations (rows).

```
filter(CW, Time==21 & weight>300)
#> # A tibble: 8 x 4
#>   Chick Diet  Time  weight
```

```
#>   <dbl> <fct> <fct>  <dbl>
#> 1     7 1     21       305
#> 2    29 2     21       309
#> 3    21 2     21       331
#> # ... with 5 more rows
```

For comparing values in vectors use: `<` (less than), `>` (greater than), `<=` (less than and equal to), `>=` (greater than and equal to), `==` (equal to) and `!=` (not equal to). These can be combined logically using `&` (and) and `|` (or).

`arrange()`: Changes the order of the observations.

```
arrange(CW, Chick, Time)
#> # A tibble: 578 x 4
#>   Chick Diet  Time  weight
#>   <dbl> <fct> <fct>  <dbl>
#> 1     1 1     0         42
#> 2     1 1     2         51
#> 3     1 1     4         59
#> # ... with 575 more rows
arrange(CW, desc(weight))
#> # A tibble: 578 x 4
#>   Chick Diet  Time  weight
#>   <dbl> <fct> <fct>  <dbl>
#> 1    35 3     21       373
#> 2    35 3     20       361
#> 3    34 3     21       341
#> # ... with 575 more rows
```

What does the `desc()` do? Try using `desc(Time)`.

1.7.3 The pipe operator %>%

In reality you will end up doing multiple data wrangling steps that you want to save. The pipe operator `%>%` makes your code nice and readable:

```
CW21 <- CW %>%
  filter(Time %in% c(0, 21)) %>%
  rename(Weight = weight) %>%
  mutate(Group = factor(str_c("Diet ", Diet))) %>%
  select(Chick, Group, Time, Weight) %>%
  arrange(Chick, Time)
CW21
#> # A tibble: 95 x 4
#>   Chick Group  Time  Weight
#>   <dbl> <fct>  <fct>  <dbl>
#> 1     1 Diet 1 0         42
#> 2     1 Diet 1 21       205
#> 3     2 Diet 1 0         40
#> # ... with 92 more rows
```

Hint: To understand the code above we should read the pipe operator `%>%` as "then".

> Create a new dataset (object) called `CW21` using dataset `CW` ***then*** keep the data for days 0 and 21 ***then*** rename variable `weight` to `Weight` ***then*** create a variable called `Group` ***then*** keep variables `Chick`, `Group`, `Time` and `Weight` and ***then*** finally arrange the data by variables `Chick` and `Time`.

This is the same code:

```
CW21 <- CW %>%
  filter(., Time %in% c(0, 21)) %>%
  rename(., Weight = weight) %>%
  mutate(., Group=factor(str_c("Diet ",Diet))) %>%
  select(., Chick, Group, Time, Weight) %>%
  arrange(., Chick, Time)
```

The pipe operator, `%>%`, replaces the dots (`.`) with whatever is returned from code preceding it. For example, the dot in `filter(., Time %in% c(0, 21))` is replaced by `CW`. The output of the `filter(...)` then replaces the dot in `rename(., Weight = weight)` and so on. Think of it as a data assembly line with each function doing its thing and passing it to the next.

1.7.4 The `group_by()` function

From the data visualizations above we concluded that the diet 3 has the highest mean and diet 4 the least variation. In this section, we will quantify the effects of the diets using **summmary statistics**. We start by looking at the number of observations and the mean by **diet** and **time**.

```
mnsdCW <- CW %>%
  group_by(Diet, Time) %>%
  summarise(N = n(), Mean = mean(weight)) %>%
  arrange(Diet, Time)
mnsdCW
#> # A tibble: 48 x 4
#> # Groups:   Diet [4]
#>   Diet  Time      N  Mean
```

```
#>   <fct> <fct> <int> <dbl>
#> 1 1     0        20  41.4
#> 2 1     2        20  47.2
#> 3 1     4        19  56.5
#> # ... with 45 more rows
```

For each distinct combination of `Diet` and `Time`, the chick weight data is summarized into the number of observations (`N`) and the mean (`Mean`) of `weight`.

Further summaries: Let's also calculate the standard deviation, median, minimum and maximum values but only at days 0 and 21.

```
sumCW <-  CW %>%
  filter(Time %in% c(0, 21)) %>%
  group_by(Diet, Time) %>%
  summarise(N = n(),
            Mean = mean(weight),
            SD = sd(weight),
            Median = median(weight),
            Min = min(weight),
            Max = max(weight)) %>%
  arrange(Diet, Time)
sumCW
#> # A tibble: 8 x 8
#> # Groups:   Diet [4]
#>   Diet  Time      N  Mean     SD Median   Min   Max
#>   <fct> <fct> <int> <dbl>  <dbl>  <dbl> <dbl> <dbl>
#> 1 1     0        20  41.4  0.995   41      39    43
#> 2 1     21       16 178.  58.7    166      96   305
```

```
#> 3 2     0        10  40.7  1.49    40.5    39    43
#> # ... with 5 more rows
```

Let's make the summaries "prettier", say, for a report or publication.

```
library("knitr") # to use the kable() function
prettySumCW <- sumCW %>%
 mutate(`Mean (SD)` = str_c(format(Mean, digits=1),
           " (", format(SD, digits=2), ")")) %>%
 mutate(Range = str_c(Min, " - ", Max)) %>%
 select(Diet, Time, N, `Mean (SD)`, Median, Range) %>%
 arrange(Diet, Time) %>%
 kable(format = "latex")
prettySumCW
```

Diet	Time	N	Mean (SD)	Median	Range
1	0	20	41 (0.99)	41.0	39 - 43
1	21	16	178 (58.70)	166.0	96 - 305
2	0	10	41 (1.5)	40.5	39 - 43
2	21	10	215 (78.1)	212.5	74 - 331
3	0	10	41 (1)	41.0	39 - 42
3	21	10	270 (72)	281.0	147 - 373
4	0	10	41 (1.1)	41.0	39 - 42
4	21	9	239 (43.3)	237.0	196 - 322

Interpretation: This summary table offers the same interpretation as before, namely that diet 3 has the highest mean and median weights at day 21 but a higher variation than group 4. However it should be noted that at day 21, diet 1 lost 4 chicks from 20 that started and diet 4 lost 1 from 10. This could be a sign of some health related issues.

1.8 Further Links

1.8.1 Further R-Intros

- https://eddelbuettel.github.io/gsir-te/Getting-Started-in-R.pdf
- https://www.datacamp.com/courses/free-introduction-to-r
- https://swcarpentry.github.io/r-novice-gapminder/
- https://support.rstudio.com/hc/en-us/articles/200526207-Using-Projects

1.8.2 Version Control (Git/GitHub)

- https://support.rstudio.com/hc/en-us/articles/200532077-Version-Control-with-Git-and-SVN
- http://happygitwithr.com/
- https://www.gitkraken.com/

1.8.3 R-Ladies

- https://rladies.org/

Chapter 2

Review: Probability and Statistics

2.1 Probability Theory

Probability is the mathematical language for quantifying uncertainty. We can apply probability theory to a diverse set of problems, from coin flipping to the analysis of econometric problems. The starting point is to specify the **sample space**, that is, the set of possible outcomes.

2.1.1 Sample Spaces and (Elementary) Events

The **sample space** Ω, is the set of possible outcomes of an experiment. Points ω in Ω are called **sample outcomes** or **realizations** or **elementary events**. **Events** are subsets of Ω.

Example: If we toss a coin twice then $\Omega = \{HH, HT, TH, TT\}$. The event that the first toss is heads is $A = \{HH, HT\}$.

Example: Let ω be the outcome of a measurement of some physical quantity, for example, temperature. Then $\Omega = \mathbb{R} = (-\infty, \infty)$. The event that the measurement is larger than 10 but less than or equal to 23 is $A = (10, 23]$.

Example: If we toss a coin forever then the sample space is the infinite set $\Omega = \{\omega = (\omega_1, \omega_2, \omega_3, \dots,) \,|\omega_i \in \{H, T\}\}$ Let A be the event that the first head appears on the third toss. Then $A = \{(\omega_1, \omega_2, \omega_3, \dots,) \,|\omega_1 = T, \omega_2 = T, \omega_3 = H, \omega_i \in \{H, T\} \text{ for } i > 3\}$.

Given an event A, let $A^c = \{\omega \in \Omega; \omega \notin A\}$ denote the **complement** of A. Informally, A^c can be read as "not A." The complement of Ω is the empty set $\emptyset$. The **union** of events A and B is defined as

$$A \cup B = \{\omega \in \Omega | \omega \in A \text{ or } \omega \in B \text{ or } \omega \in \text{ both}\}$$

which can be thought of as "A or B." If $A_1, A_2, \dots$ is a sequence of sets then

$$\bigcup_{i=1}^{\infty} A_i = \{\omega \in \Omega : \omega \in A_i \text{ for at least one i } \}.$$

The **intersection** of A and B is defined as

$$A \cap B = \{\omega \in \Omega; \omega \in A \text{ and } \omega \in B\}$$

which reads as "A and B." Sometimes $A \cap B$ is also written shortly as AB. If $A_1, A_2, \dots$ is a sequence of sets then

$$\bigcap_{i=1}^{\infty} A_i = \{\omega \in \Omega : \omega \in A_i \text{ for all i } \}.$$

If every element of A is also contained in B we write $A \subset B$ or, equivalently, $B \supset A$. If A is a finite set, let $|A|$ denote the number of elements in A. We say that $A_1, A_2, \dots$ are **disjoint** or **mutually exclusive** if $A_i \cap A_j = \emptyset$ whenever $i \neq j$. For example, $A_1 = [0, 1), A_2 = [1, 2), A_3 = [2, 3), \dots$ are disjoint. A **partition** of Ω is a sequence of disjoint sets $A_1, A_2, \dots$ such that $\cup_{i=1}^{\infty} A_i = \Omega$.

Summary: Sample space and events

Ω	sample space
ω	outcome
A	event (subset of Ω)
$\|A\|$	number of points in A (if A is finite)
A^c	complement of A(not A)
$A \cup B$	union (A or B)
$A \cap B$	intersection (A and B); short notation: AB
$A \subset B$	set inclusion (A is a subset of or equal to B)
$\emptyset$	null event (always false)
Ω	true event (always true)

2.1.2 Probability

We want to assign a real number $P(A)$ to every event A, called the **probability** of A. We also call P a **probability distribution** or a **probability measure**. To qualify as a probability, P has to satisfy three axioms. That is, a function P that assigns a real number $P(A) \in [0, 1]$ to each event A is a **probability distribution** or a **probability measure** if it satisfies the following three axioms:

Axiom 1: $P(A) \geq 0$ for every A

Axiom 2: $P(\Omega) = 1$

Axiom 3: If $A_1, A_2, \ldots$ are disjoint then

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P\left(A_i\right).$$

Note: It is not always possible to assign a probability to every event A if the sample space is large, such as, for instance, the whole real line, $\Omega = \mathbb{R}$. In case

of $\Omega = \mathbb{R}$ strange things can happen. There are pathological sets that simply break down the mathematics. An example of one of these pathological sets, also known as non-measurable sets because they literally can't be measured (i.e. we cannot assign probabilities to them), are the Vitali sets. Therefore, in such cases like $\Omega = \mathbb{R}$, we assign probabilities to a *limited* class of sets called a **σ-field** or **σ-algebra**. For $\Omega = \mathbb{R}$, the canonical **σ-algebra** is the **Borel σ-algebra**. The Borel σ-algebra on $\mathbb{R}$ is generated by the collection of all open subsets of R.

One can derive many properties of P from the axioms. Here are a few:

- $P(\emptyset) = 0$
- $A \subset B \Rightarrow P(A) \leq P(B)$
- $0 \leq P(A) \leq 1$
- $P\left(A^c\right) = 1 - P(A)$
- $A \cap B = \emptyset \Rightarrow P(A \cup B) = P(A) + P(B)$

A less obvious property is given in the following: For any events A and B we have that,

$$P(A \cup B) = P(A) + P(B) - P(AB).$$

Example. Two consecutive coin tosses. Let H_1 be the event that heads occurs on toss 1 and let H_2 be the event that heads occurs on toss 2. If all outcomes are equally likely, that is, $\mathrm{P}\left(\{H_1, H_2\}\right) = \mathrm{P}\left(\{H_1, T_2\}\right) = \mathrm{P}\left(\{T_1, H_2\}\right) = \mathrm{P}\left(\{T_1, T_2\}\right) = 1/4$, then

$$\mathrm{P}\left(H_1 \cup H_2\right) = \mathrm{P}\left(H_1\right) + \mathrm{P}\left(H_2\right) - \mathrm{P}\left(H_1 H_2\right) = \frac{1}{2} + \frac{1}{2} - \frac{1}{4} = \frac{3}{4}.$$

Probabilities as frequencies. One can interpret $P(A)$ in terms of **frequencies**. That is, $P(A)$ is the (infinitely) long run proportion of times that A is true in repetitions. For example, if we say that the probability of heads is 1/2, i.e $P(H) = 1/2$ we mean that if we flip the coin many times then the proportion of times we get heads tends to 1/2 as the number of tosses increases. An infinitely long, unpredictable sequence of tosses whose limiting proportion tends to a constant is an idealization, much like the idea of a straight line in geometry.
The following R codes approximates the probability $P(H) = 1/2$ using 5, 50 and 5,000 many (pseudo) random coin flips:

```
set.seed(869)
## 1 (fair) coin-flip:
results <- sample(x = c("H", "T"), size = 5, replace = TRUE)
## Relative frequency of "H" in 5 coin-flips
length(results[results=="H"])/5
#> [1] 0.2

## 10 (fair) coin-flips:
results <- sample(x = c("H", "T"), size = 50, replace = TRUE)
## Relative frequency of "H" in 50 coin-flips
length(results[results=="H"])/50
#> [1] 0.52

## 100000 (fair) coin-flips:
results <- sample(x = c("H", "T"), size = 5000, replace = TRUE)
## Relative frequency of "H" in 5000 coin-flips
length(results[results=="H"])/5000
#> [1] 0.5024
```

2.1.3 Independent Events

If we flip a fair coin twice, then the probability of two heads is $\frac{1}{2} \times \frac{1}{2}$. We multiply the probabilities because we regard the two tosses as independent. Two events A and B are called **independent** if

$$P(AB) = P(A)P(B).$$

Or more generally, a whole set of events $\{A_i | i \in I\}$ is independent if

$$P\left(\bigcap_{i \in J} A_i\right) = \prod_{i \in J} P\left(A_i\right)$$

for every finite subset J of I, where I denotes the not necessarily finite index set (e.g. $I = \{1, 2, \dots\}$).

Independence can arise in two distinct ways. Sometimes, we **explicitly assume** that two events are independent. For example, in tossing a coin twice, we usually assume the tosses are independent which reflects the fact that the coin has no memory of the first toss.
In other instances, we **derive** independence by verifying that the definition of independence $P(AB) = P(A)P(B)$ holds. For example, in tossing a fair die *once*, let $A = \{2, 4, 6\}$ be the event of observing an even number and let $B = \{1, 2, 3, 4\}$ be the event of observing no 5 and no 6. Then, $A \cap B = \{2, 4\}$ is the event of observing either a 2 or a 4. Are the events A and B independent?

$$P(AB) = \frac{2}{6} = P(A)P(B) = \frac{1}{2} \cdot \frac{2}{3}$$

and so A and B are independent. In this case, we didn't assume that A and B are independent it just turned out that they were.

Cautionary Notes. Suppose that A and B are **disjoint events** (i.e. $AB = \emptyset$), each with positive probability (i.e. $P(A) > 0$ and $P(B) > 0$). Can they be independent? No! This follows since

$$P(AB) = P(\emptyset) = 0 \neq P(A)P(B) > 0.$$

Except in this special case, there is no way to judge (in-)dependence by looking at the sets in a Venn diagram.

Summary: Independence

1. A and B are independent if $P(AB) = P(A)P(B)$.
2. Independence is sometimes assumed and sometimes derived.
3. Disjoint events with strictly positive probabilities are not independent.

2.1.4 Conditional Probability

If $P(B) > 0$ then the **conditional probability** of A given B is

$$P(A \mid B) = \frac{P(AB)}{P(B)}.$$

Think of $P(A \mid B)$ as the fraction of times A occurs among those in which B occurs. Here are some facts about conditional probabilities:

- The rules of probability apply to events on the left of the bar "|". That is, for any fixed B such that $P(B) > 0, P(\cdot \mid B)$ is a probability i.e. it satisfies the three axioms of probability: $P(A \mid B) \geq 0, P(\Omega \mid B) = 1$ and if $A_1, A_2, \ldots$ are disjoint then $P\left(\cup_{i=1}^{\infty} A_i \mid B\right) = \sum_{i=1}^{\infty} P\left(A_i \mid B\right)$.
- But it's generally not true that $P(A \mid B \cup C) = P(A \mid B) + P(A \mid C)$.

In general it is also **not** the case that $P(A \mid B) = P(B \mid A)$. People get this confused all the time. For example, the probability of spots given you have measles is 1 but the probability that you have measles given that you have spots is not 1. In this case, the difference between $P(A \mid B)$ and $P(B \mid A)$ is obvious but there are cases where it is less obvious. This mistake is made often enough in legal cases that it is sometimes called the "prosecutor's fallacy".

Example. A medical test for a disease D has outcomes $+$ and $-$. The probabilities are:

	D	D^c	
$+$	.0081	.0900	.0981
$-$	.0009	.9010	.9019
	.0090	.9910	1

From the definition of conditional probability, we have that:

- Sensitivity of the test:

$$P(+ \mid D) = P(+ \cap D)/P(D) = 0.0081/(0.0081 + 0.0009) = 0.9$$

- Specificity of the test:

$$P(- \mid D^c) = P(- \cap D^c)/P(D^c) = 0.9010/(0.9010 + 0.0900) \approx 0.9$$

Apparently, the test is fairly accurate. Sick people yield a positive test result 90 percent of the time and healthy people yield a negative test result about 90 percent of the time. Suppose you go for a test and get a positive result. What is the probability you have the disease? Most people answer $0.90 = 90\%$. The correct answer is $P(D \mid +) = P(+ \cap D)/P(+) = 0.0081/(0.0081 + 0.0900) =$

0.08. The lesson here is that you need to compute the answer numerically. Don't trust your intuition.

If A and B are **independent events** then

$$P(A \mid B) = \frac{P(AB)}{P(B)} = \frac{P(A)P(B)}{P(B)} = P(A)$$

So another **interpretation of independence** is that knowing B doesn't change the probability of A.

From the definition of conditional probability we can write

$$P(AB) = P(A \mid B)P(B) \quad \text{and also} \quad P(AB) = P(B \mid A)P(A).$$

Often, these formulas give us a convenient way to compute $P(AB)$ when A and B are not independent.

Note, sometimes $P(AB)$ is written as $P(A, B)$.

Example. Draw two cards from a deck, without replacement. Let A be the event that the first draw is Ace of Clubs and let B be the event that the second draw is Queen of Diamonds. Then $P(A, B) = P(A)P(B \mid A) = (1/52) \times (1/51)$

Summary: Conditional Probability

1. If $P(B) > 0$ then $P(A \mid B) = P(AB)/P(B)$
2. $P(\cdot \mid B)$ satisfies the axioms of probability, for fixed B. In general, $P(A \mid \cdot)$ does not satisfy the axioms of probability, for fixed A.
3. In general, $P(A \mid B) \neq P(B \mid A)$.
4. A and B are independent if and only if $P(A \mid B) = P(A)$.

2.2 Random Variables

Statistics and econometrics are concerned with data. How do we link sample spaces, events and probabilities to data? The link is provided by the concept of a **random variable**. A real-valued **random variable** is a mapping $X : \Omega \to \mathbb{R}$ that assigns a real number $X(\omega) \in \mathbb{R}$ to each outcome ω.

At a certain point in most statistics/econometrics courses, the sample space, Ω, is rarely mentioned and we work directly with random variables. But you should keep in mind that the sample space is really there, lurking in the background.

Example. Flip a coin ten times. Let $X(\omega)$ be the number of heads in the sequence ω. For example, if $\omega = \text{HHTHHTHHTT}$ then $X(\omega) = 6$.

Example. Let $\Omega = \{(x, y)|x^2 + y^2 \leq 1\}$ be the unit disc. Consider drawing a point "at random" from Ω. A typical outcome is then of the form $\omega = (x, y)$. Some examples of random variables are $X(\omega) = x, Y(\omega) = y, Z(\omega) = x + y, W(\omega) = \sqrt{x^2 + y^2}$.

Given a real-valued random variable $X \in \mathbb{R}$ and a subset A of the real line ($A \subset \mathbb{R}$), define $X^{-1}(A) = \{\omega \in \Omega | X(\omega) \in A\}$. This allows us to link the probabilities on the random variable X, i.e. the probabilities we are usually working with, to the underlying probabilities on the events, i.e. the probabilities lurking in the background.

Example. Flip a coin twice and let X be the number of heads. Then, $P_X(X = 0) = P(\{TT\}) = 1/4$, $P_X(X = 1) = P(\{HT, TH\}) = 1/2$ and

$P_X(X = 2) = P(\{HH\}) = 1/4$. Thus, the events and their associated probability distribution, P, and the random variable X and its distribution, P_X, can be summarized as follows:

ω	$P(\{\omega\})$	$X(\omega)$
TT	$1/4$	0
TH	$1/4$	1
HT	$1/4$	1
HH	$1/4$	2

x	$P_X(X = x)$
0	$1/4$
1	$1/2$
2	$1/4$

Here, P_X is not the same probability function as P because P maps from the sample space events, ω, to $[0, 1]$, while P_X maps from the random-variable events, $X(\omega)$, to $[0, 1]$. We will typically forget about the sample space Ω and just think of the random variable as an experiment with real-valued (possible multivariate) outcomes. We will therefore write $P(X = x_k)$ instead of $P_X(X = x_k)$ to simplify the notation.

2.2.1 Univariate Distribution and Probability Functions

2.2.1.1 Cumulative Distribution Function

The **cumulative distribution function (cdf)**

$$F_X : \mathbb{R} \to [0, 1]$$

of a real-valued random variable $X \in \mathbb{R}$ is defined by

$$F_X(x) = \mathbb{P}(X \leq x).$$

You might wonder why we bother to define the cdf. The reason is that it effectively contains all the information about the random variable. Indeed,

let $X \in \mathbb{R}$ have cdf F and let $Y \in \mathbb{R}$ have cdf G. If $F(x) = G(x)$ for all $x \in \mathbb{R}$ then $P(X \in A) = P(Y \in A)$ for all $A \subset \mathbb{R}$. In order to denote that two random variables, here X and Y, have the same distribution, one can write shortly $X \stackrel{d}{=} Y$.

Caution: Equality in distribution, $X \stackrel{d}{=} Y$, does generally **not** mean equality in realizations, that is $X \stackrel{d}{=} Y \not\Rightarrow X(\omega) = Y(\omega)$ for all $\omega \in \Omega$.

The defining properties of a cdf. A function F mapping the real line to $[0, 1]$, short $F : \mathbb{R} \to [0, 1]$, is called a cdf for some probability measure P if and only if it satisfies the following three properties:

1. F is non-decreasing i.e. $x_1 < x_2$ implies that $F(x_1) \leq F(x_2)$.

2. F is normalized: $\lim_{x \to -\infty} F(x) = 0$ and $\lim_{x \to \infty} F(x) = 1$

3. F is right-continuous, i. e. $F(x) = F(x^+)$ for all x, where

$$F\left(x^+\right) = \lim_{y \to x, y > x} F(y).$$

Alternatively to cumulative distribution functions one can use **probability (mass) functions** in order to describe the probability law of **discrete** random variables and **denstiy functions** in order to describe the probability law of **continuous** random variables.

2.2.1.2 Probability Functions for Discrete Random Variables.

A random variable X is *discrete* if it takes only countably many values

$$X \in \{x_1, x_2, \ldots\}.$$

For instance, $X \in \{1, 2, 3\}$ or $X \in \{2, 4, 6, \ldots\}$ or $X \in \mathbb{Z}$ or $X \in \mathbb{Q}$.

We define the **probability function** or **probability mass function (pmf)** for X by

$$f_X(x) = \mathbb{P}(X = x) \quad \text{for all} \quad x \in \{x_1, x_2, \ldots\}$$

2.2.1.3 Density Functions for Continuous Random Variables.

A random variable X is *continuous* if there exists a function f_X such that

1. $f_X(x) \geq 0$ for all x
2. $\int_{-\infty}^{\infty} f_X(x)dx = 1$ and
3. $\mathbb{P}(a < X < b) = \int_a^b f_X(x)dx$ for every $a \leq b$.

The function f_X is called the **probability density function (pdf)** or short **density function**. We have that

$$F_X(x) = \int_{-\infty}^{x} f_X(t)dt \quad \text{and} \quad f_X(x) = F'_X(x)$$

at all points x at which F_X is differentiable.

2.2.2 Multivariate Distribution and Probability Functions

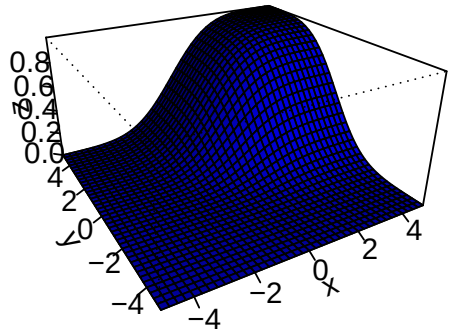
A d-dimensional random vector is a column-vector $X = (X_1, \ldots, X_d)'$, where each element is a univariate random variable.

2.2.2.1 Multidimensional Distribution Function

The **multivariate distribution function** F is given by

$$F(a_1, \ldots, a_d) = P(X_1 \leq a_1, \ldots, X_d \leq a_d).$$

```
## Install the package if not installed yet
# install.packages("mnormt")

library(mnormt)

x     <- seq(-5, 5, 0.25)
y     <- seq(-5, 5, 0.25)
mu    <- c(0, 0)
sigma <- matrix(c(2, -1, -1, 2), nrow = 2)
f     <- function(x, y) pmnorm(cbind(x, y), mu, sigma)
z     <- outer(x, y, f)

persp(x, y, z, theta = -30, phi = 25,
      shade = 0.75, col = "blue", expand = 0.5, r = 2,
      ltheta = 25, ticktype = "detailed")
```



2.2.2.2 Multidimensional Probability Function

Discrete random vectors. X takes only countably many (i.e. discrete) values $\mathbf{x}_1, \mathbf{x}_2, \cdots \in \mathbb{R}^d$ and has a **multidimensional probability function** $p(\mathbf{x}_i) = P(X = \mathbf{x}_i)$ for $i = 1, 2, \ldots$. That is,

$$P(X \in [a_1, b_1] \times \cdots \times [a_d, b_d]) = \sum_{\mathbf{x}_i \in [a_1, b_1] \times \cdots \times [a_d, b_d]} p(\mathbf{x}_i).$$

2.2.2.3 Multidimensional Density Function

Continouse random vectors. X takes values in $\mathbb{R}^d$ and has a **multidimensional density function** $f(x_1, \ldots, x_d)$. That is,

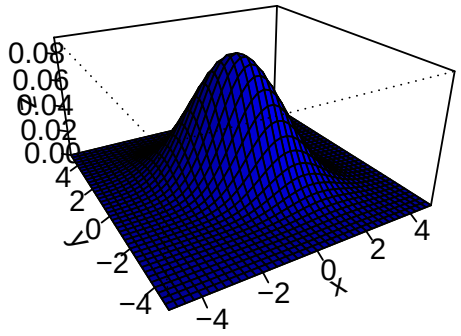
$$P(X \in [a_1, b_1] \times \cdots \times [a_d, b_d]) = \int_{a_d}^{b_d} \cdots \int_{a_1}^{b_1} f(x_1, \ldots, x_d) dx_1 \ldots dx_d.$$

In the following we focus only on continuous random vectors – the discrete cases are treated analogously. Properties of **multivariate density** functions:

- $f(x_1, \ldots, x_d) \geq 0$

- $\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(x_1, \ldots, x_d) dx_1 \ldots dx_d = 1$

```
## Load the package
library(mnormt)

x     <- seq(-5, 5, 0.25)
y     <- seq(-5, 5, 0.25)
mu    <- c(0, 0)
sigma <- matrix(c(2, -1, -1, 2), nrow = 2)
f     <- function(x, y) dmnorm(cbind(x, y), mu, sigma)
z     <- outer(x, y, f)

persp(x, y, z, theta = -30, phi = 25,
      shade = 0.75, col = "blue", expand = 0.5, r = 2,
      ltheta = 25, ticktype = "detailed")
```

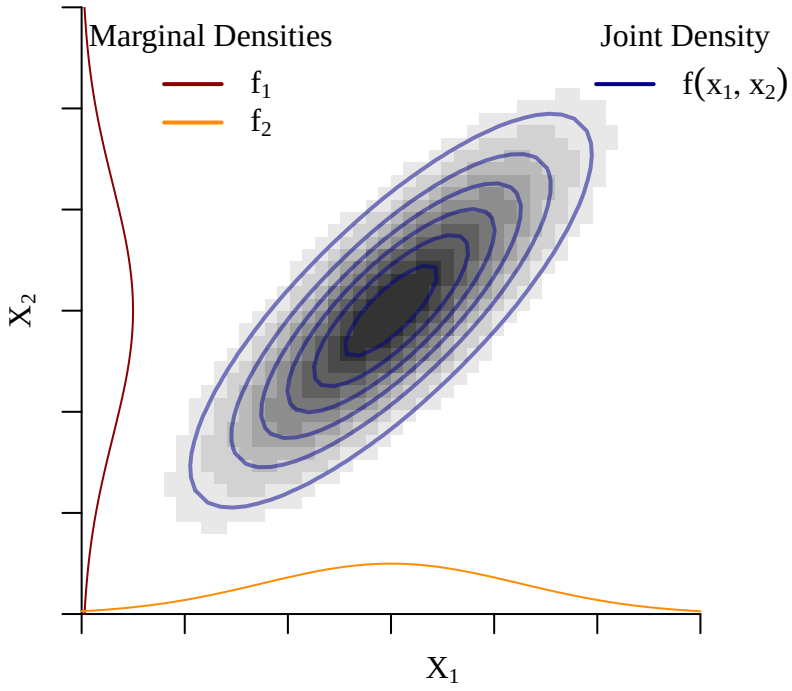


2.2.2.4 Marginal Distribution and Density Functions

Each random element, X_j, with $j = 1, \ldots, d$, of the random vector X has its own **marginal distribution** F_j. This is just the univariate distribution of X_j when ignoring all other random variables in X. Formally we have:

- **Marginal distribution function:** $F_j(x) = P(X_j \leq x)$
- **Marginal density function:** f_j, for instance, for $j = 1$:

$$f_1(x_1) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(x_1, x_2 \ldots, x_d) dx_2 \ldots dx_d$$



2.2.2.5 Conditional Distributions

Often, we are interested in the **conditional distribution** of X_j given certain values of all other random variables

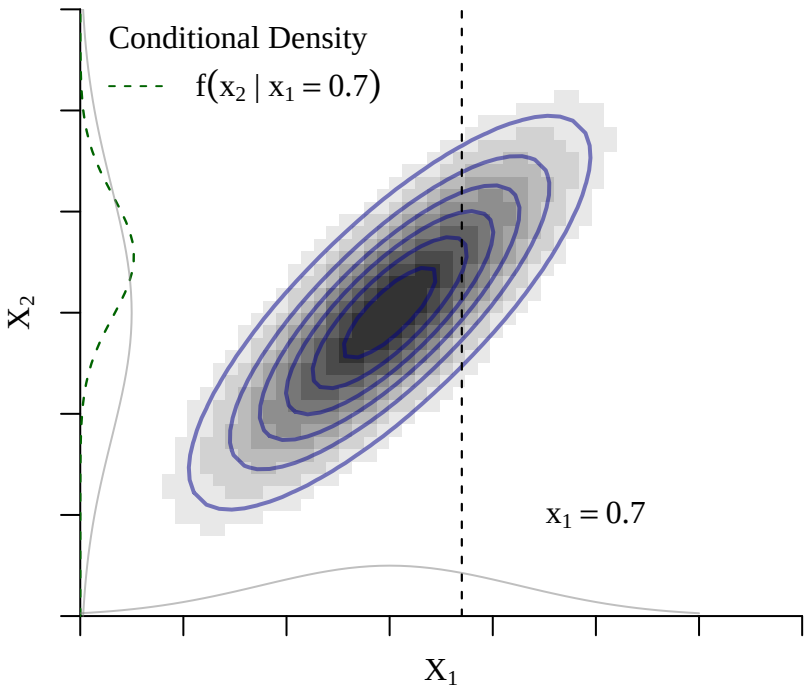
$$X_1 = x_1, \ldots, X_{j-1} = x_{j-1}, X_{j+1} = x_{j+1}, \ldots, X_d = x_d.$$

That is, the distribution of X_j when fixing the values of $X_1 = x_1, \ldots, X_{j-1} = x_{j-1}, X_{j+1} = x_{j+1}, \ldots, X_d = x_d$. An important tool is here the **conditional density** of, for instance, X_1 given $X_2 = x_2, \ldots, X_d =$

x_d:

$$f(x_1 \mid x_2, \ldots, x_d) = \frac{f(x_1, x_2, \ldots, x_d)}{f_{X_2,\ldots,X_d}(x_2, \ldots, x_d)},$$

where $f_{X_2,\ldots,X_d}$ denotes the joint density of $X_2, \ldots, X_d$.



2.2.3 Means and Moments

2.2.4 Unconditional Means

The **unconditional mean** of X_1 is given by

$$E(X_1) = \int x f_{X_1}(x) dx.$$

The unconditional mean of a random vector $X = (X_1, \ldots, X_d)'$ is given by the vector of element-wise means

$$E(X) = (E(X_1), \ldots, E(X_d))'.$$

2.2.5 Conditional Means

Of central importance in **regression analysis** is the **conditional mean**. The conditional mean of X_1 for given values $X_2 = x_2, \ldots, X_d = x_d$:

$$\begin{aligned} m(x_2, \ldots, x_d) :&= E(X_1 | X_2 = x_2, \ldots, X_d = x_d) \\ &= \int x_1 f(x_1 \mid x_2, \ldots, x_d) dx_1, \end{aligned}$$

where $m(x_2, \ldots, x_d)$ denotes the **regression function**.

2.2.6 Means of Transformed Random Variables and Moments

The **mean of a transformed random variable** $r(X)$ is given by

$$E(r(X)) = \int r(x) f_X(x) dx.$$

Typical transformations are, for instance

- centering $r(x) = x - E(X)$,
- centering and scaling $r(x) = (x - E(X))/\sqrt{Var(X)}$,
- or $r(x) = (x - E(X))^2$,

where the latter transformation leads to the **second central moment**, i.e. the variance of X, $Var(X) = \int (x - E(X))^2 f_X(x)dx$.

- The kth, $k > 0$, moment is given by

$$\mu_k = \mathrm{E}\left[X^k\right] = \int_{-\infty}^{+\infty} x^k f_X(x)dx.$$

- The kth, $k > 1$, central moment is given by

$$\mu_k^c = \mathrm{E}\left[(X - \mathrm{E}[X])^k\right] = \int_{-\infty}^{+\infty} (x - \mu)^k f_X(x)dx,$$

  where $\mu = E(X)$.

Note. Moments determine the tail of a distribution (but not much else); see Lindsay and Basak (2000). Roughly: The more moments a distribution has the faster converge its tails to zero. Distributions with compact supports (e.g. the uniform distribution $U[a, b]$) have infinitely many moments. The Normal distribution has also infinitely many moments – even though this distribution has not a compact support since $\phi(x) > 0$ for all $x \in \mathbb{R}$. .

2.2.6.1 Law of Total Expectation

As long as we do not fix the values of the conditioning variables, $X_2, \ldots, X_d$, they are random variables. Consequently, the conditional mean is generally itself a random variable

$$E(X_1|X_2, \ldots, X_d) = \int x_1 f(x_1 \mid X_2, \ldots, X_d)dx_1.$$

Note that $f(x_1 \mid X_2, \ldots, X_d)$ is just a transformation of the random variables $X_2, \ldots, X_d$. So we can easily compute the unconditional mean $E(X_1)$ by taking the mean of $E(X_1|X_2, \ldots, X_d)$ as following,

$$\begin{aligned}
&E\big(E(X_1|X_2, \ldots, X_d)\big) = \\
&= \int \cdots \int \int x_1 f(x_1 \mid x_2, \ldots, x_d) dx_1 \; f_{X_2,\ldots,X_d}(x_2, \ldots, x_d) dx_2 \ldots dx_d \\
&= \int x_1 \left(\int \cdots \int f(x_1, x_2, \ldots, x_d) dx_2 \ldots dx_d \right) dx_1 \\
&= \int x_1 f_{X_1}(x_1) dx_1 \\
&= E(X_1).
\end{aligned}$$

The result that $E\big(E(X_1|X_2, \ldots, X_d)\big) = E(X_1)$ is called **law of total expectation** or **law of iterated expectation**.

2.2.7 Independent Random Variables

Random variables $X_1, \ldots, X_d$ are mutually **independent** if for all $x = (x_1, \ldots, x_d)'$ it is true that

$$\begin{aligned}
F(x_1, \ldots, x_d) &= F_1(x_1) \cdot F_2(x_2) \cdot \ldots \cdot F_d(x_d) \\
f(x_1, \ldots, x_d) &= f_1(x_1) \cdot f_2(x_2) \cdot \ldots \cdot f_d(x_d)
\end{aligned}$$

The following holds true:

- Two real-valued random variables X and Y are independent from each other *if and only if* the marginal density of X equals the conditional density of X given $Y = y$ for all $y \in \mathbb{R}$,

  $$f_X(x) = f_{X|Y}(x \mid y) \quad \text{for all } y \in \mathbb{R}.$$

  Of course, the same statement applies to the marginal density of Y given $X = x$ for all $x \in \mathbb{R}$. That is, X and Y are two independent

real-valued random variables *if and only if* $f_Y(y) = f_{Y|X}(y \mid x)$ for all $x \in \mathbb{R}$.

- If a real-valued random variable X is independent from a real-valued random variable Y, then the conditional mean of X given $Y = y$ equals the unconditional mean of X for all $y \in \mathbb{R}$ (i.e. the regression function becomes a constant)

$$E(X \mid Y = y) = E(X) \quad \text{for all } y \in \mathbb{R}.$$

Of course, the same statement applies to the conditional mean of Y given $X = x$ for all $x \in \mathbb{R}$; i.e., if X and Y are two independent random variables, then $E(Y \mid X = x) = E(Y) \quad$ for all $x \in \mathbb{R}$.

Cautionary note. The properties that $E(X \mid Y = y) = E(X)$ for all $y \in \mathbb{R}$ or that $E(Y \mid X = x) = E(Y)$ for all $x \in \mathbb{R}$, do **not** imply that Y and X are independent.

2.2.8 I.I.D. Samples

Tradition dictates that the sample size is denoted by the natural number $n \in \{1, 2, \dots\}$. A random sample is a collection $X = (X_1, \dots, X_n)$ of random variables $X_1, \dots, X_n$. If $X_1, \dots, X_n$ are all **independent** from each other and if each random variable has the same marginal distribution, we say that the random sample

$X = (X_1, \dots, X_n)$ is **i.i.d. (independent and identically distributed)**.

2.2.9 Some Important Discrete Random Variables

2.2.9.1 The Discrete Uniform Distribution

Let $k > 1$ be a given integer. Suppose that X has probability mass function given by

$$f(x) = \begin{cases} 1/k & \text{for } x = 1, \ldots, k \\ 0 & \text{otherwise.} \end{cases}$$

We say that X has a uniform distribution on $\{1, \ldots, k\}$.

```
set.seed(51)
## Set the parameter k
k <- 10
## Draw one realization from the discrete uniform distribution
sample(x = 1:k, size = 1, replace = TRUE)
#> [1] 7
```

2.2.9.2 The Bernoulli Distribution

Let X represent a possibly unfair coin flip. Then $P(X = 1) = p$ and $P(X = 0) = 1 - p$ for some $p \in [0, 1]$. We say that X has a Bernoulli distribution written $X \sim \text{Bernoulli}(p)$. The probability function is $f(x) = p^x(1 - p)^{1-x}$ for $x \in \{0, 1\}$

```
set.seed(51)
## Set the parameter p
p <- 0.25
## Draw n realization from the discrete uniform distribution
n <- 5
sample(x = c(0,1), size = n, prob = c(1-p, p), replace=TRUE)
#> [1] 1 0 0 1 0
```

```
## Alternatively:
## (Bernoulli(p) equals Binomial(1,p))
rbinom(n = n, size = 1, prob = p)
#> [1] 1 1 0 1 0
```

2.2.9.3 The Binomial Distribution

Suppose we have a coin which falls heads with probability p for some $p \in [0, 1]$. Flip the coin n times and let X be the number of heads (or successes). Assume that the tosses are independent. Let $f(x) = P(X = x)$ be the mass function. It can be shown that

$$f(x) = \begin{cases} \binom{n}{x} p^x (1-p)^{n-x} & \text{for } x = 0, \ldots, n \\ 0 & \text{otherwise.} \end{cases}$$

A random variable with this mas function is called a **binomial random variable** and we write $X \sim \text{Binomial}(n, p)$. If $X_1 \sim \text{Binomial}\ (n_1, p1)$ and $X_2 \sim \text{Binomial}(n_2, p)$ and if X_1 and X_2 are independent, then $X_1 + X_2 \sim \text{Binomial}\,(n_1 + n_2, p)$

```
set.seed(51)
## Set the parameters n and p
size <-   10 # number of trials
p    <- 0.25 # prob of success

## Draw n realization from the binomial distribution:
n <- 5
rbinom(n = n, size = size, prob = p)
#> [1] 4 1 2 6 1
```

2.2.10 Some Important Continuous Random Variables

2.2.10.1 The Uniform Distribution

X has a Uniform(a, b) distribution, written $X \sim$ Uniform(a, b), if

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } x \in [a, b] \\ 0 & \text{otherwise} \end{cases}$$

where $a < b$. The distribution function is

$$F(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & x \in [a, b] \\ 1 & x > b \end{cases}$$

```
## Drawing from the uniform distribution:
n <- 10
a <- 0
b <- 1
runif(n = n, min = a, max = b)
#>  [1] 0.83442365 0.75138318 0.40601047 0.97101998 0.11233151 0.50750
#>  [8] 0.17104008 0.25448233 0.01813812
```

2.2.10.2 The Normal (or Gaussian) Distribution

X has a Normal (or Gaussian) distribution with parameters μ and σ, denoted by $X \sim N\left(\mu, \sigma^2\right)$, if

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}, \quad x \in \mathbb{R}$$

where $\mu \in \mathbb{R}$ and $\sigma > 0$. Later we shall see that μ is the "center" (or mean of the distribution and σ is the "spread" (or standard deviation) of the distribution. The Normal plays an important role in probability and statistics.

Many phenomena in nature have approximately Normal distributions. The **Central Limit Theorem** gives a special role to the Normal distribution by stating that the distribution of averages of random variables can be approximated by a Normal distribution.

We say that X has a standard Normal distribution if $\mu = 0$ and $\sigma = 1$. Tradition dictates that a standard Normal random variable is denoted by Z. The PDF and CDF of a standard Normal are denoted by $\phi(z)$ and $\Phi(z)$. There is no closed-form expression for Φ. Here are some useful facts:

(i) If $X \sim N(\mu, \sigma^2)$ then $Z = (X - \mu)/\sigma \sim N(0, 1)$

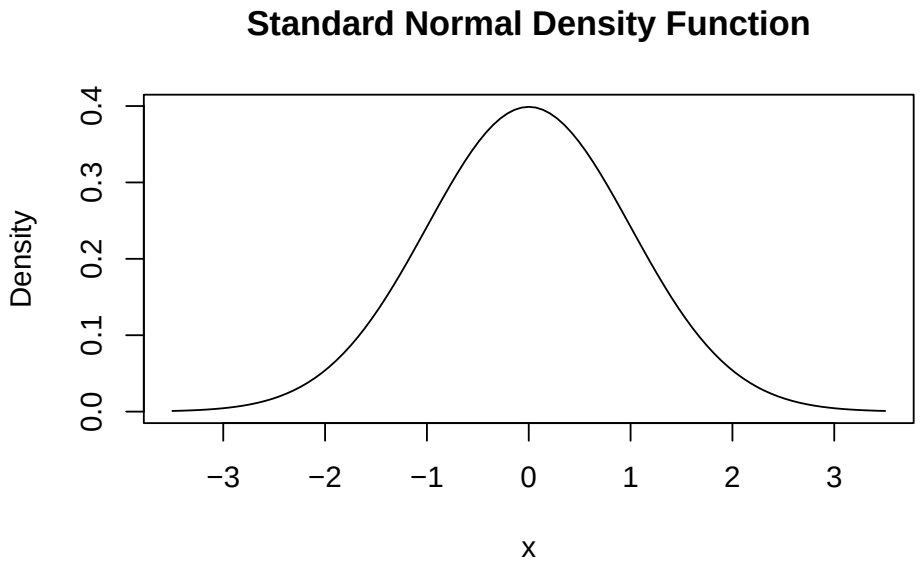
(ii) If $Z \sim N(0, 1)$ then $X = \mu + \sigma Z \sim N(\mu, \sigma^2)$

(iii) If $X_i \sim N(\mu_i, \sigma_i^2), i = 1, \ldots, n$ are independent then

$$\sum_{i=1}^{n} X_i \sim N\left(\sum_{i=1}^{n} \mu_i, \sum_{i=1}^{n} \sigma_i^2\right).$$

The following R-codes plots the standard Normal density function:

```
# draw a plot of the N(0,1) PDF
curve(dnorm(x),
      xlim = c(-3.5, 3.5),
      ylab = "Density",
      main = "Standard Normal Density Function")
```



This is how you can draw realizations from pseudo random Normal variables:

```
## Drawing from the uniform distribution:
n     <- 12
mu    <- 0
sigma <- 1
rnorm(n = n, mean = mu, sd = sigma)
#>  [1]  0.085602504 -0.695791615 -1.364410561 -0.183503290 -1.675347
#>  [7]  0.346965187  0.037914318  0.881345676 -0.882815597 -0.883560
```

An extension of the normal distribution in a univariate setting is the multivariate normal distribution. Let $X = (X_1, \ldots, X_k)'$ be a k-dimensional normal variable, short $X \sim N_k(\mu, \Sigma)$ with mean vector $E(X) = \mu \in \mathbb{R}^k$ and covariance matrix $\text{Cov}(X) = \Sigma$. The joint density function or **probability density function (pdf)** of the k-dimensional multivariate normal

distribution is

$$f_X(x_1, \ldots, x_k) = \frac{\exp\left(-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\right)}{\sqrt{(2\pi)^k|\Sigma|}},$$

where $|\Sigma|$ denotes the determinant of Σ. For $k = 2$ we have the bivariate pdf of two random normal variables, X and Y say

$$g_{X,Y}(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho_{XY}^2}} \cdot \exp\left\{\frac{1}{-2(1-\rho_{XY}^2)}\left[\left(\frac{x-\mu_x}{\sigma_X}\right)^2 - 2\rho_{XY}\left(\frac{x-\mu_X}{\sigma_X}\right)\left(\frac{y-\mu_Y}{\sigma_Y}\right) + \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]\right\}.$$

Lets consider the special case where X and Y are independent standard normal random variables with densities $f_X(x)$ and $f_Y(y)$. We then have the parameters $\sigma_X = \sigma_Y = 1$, $\mu_X = \mu_Y = 0$ (due to marginal standard normality) and correlation $\rho_{XY} = 0$ (due to independence). The joint density of X and Y then becomes

$$g_{X,Y}(x,y) = f_X(x)f_Y(y) = \frac{1}{2\pi} \cdot \exp\left\{-\frac{1}{2}\left[x^2 + y^2\right]\right\}.$$

2.2.10.3 The Chi-Squared Distribution

The chi-squared distribution is another distribution relevant in econometrics. It is often needed when testing special types of hypotheses frequently encountered when dealing with regression models.

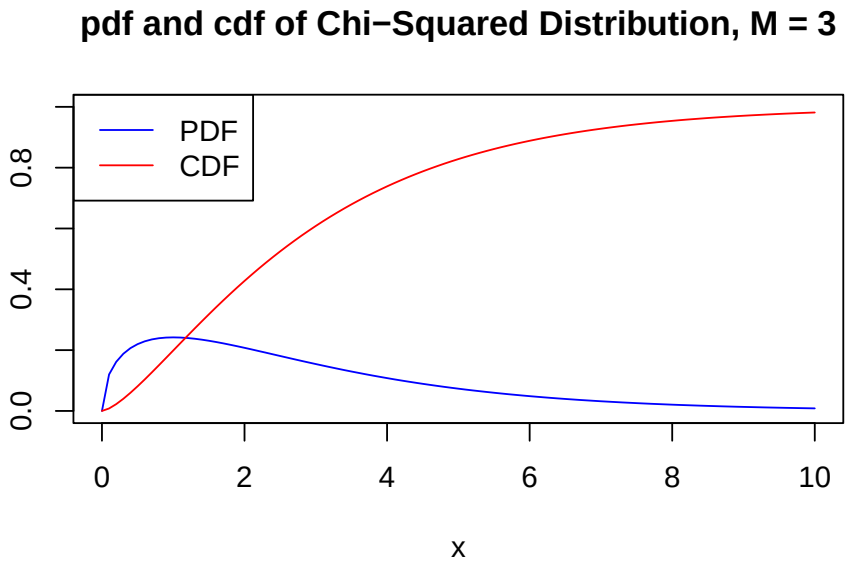
The sum of M squared *independent standard normal* distributed random variables, $Z_1, \ldots, Z_M$ follows a chi-squared distribution with M degrees of freedom:

$$Z_1^2 + \cdots + Z_M^2 = \sum_{m=1}^{M} Z_m^2 \sim \chi_M^2.$$

A χ^2 distributed random variable with M degrees of freedom has expectation M, mode at $M-2$ for $M \geq 2$ and variance $2 \cdot M$.

Using the code below, we can display the pdf and the distribution function or **cumulated density function (cdf)** of a χ^2_3 random variable in a single plot. This is achieved by setting the argument `add = TRUE"` in the second call of `"curve()"`. Further we adjust limits of both axes using `"xlim"` and `"ylim"` and choose different colors to make both functions better distinguishable. The plot is completed by adding a legend with help of `"legend()"`.

```
# plot the PDF
curve(dchisq(x, df = 3),
      xlim = c(0, 10),
      ylim = c(0, 1),
      col = "blue",
      ylab = "",
      main = "pdf and cdf of Chi-Squared Distribution, M = 3")

# add the CDF to the plot
curve(pchisq(x, df = 3),
      xlim = c(0, 10),
      add = TRUE,
      col = "red")

# add a legend to the plot
legend("topleft",
       c("PDF", "CDF"),
       col = c("blue", "red"),
       lty = c(1, 1))
```

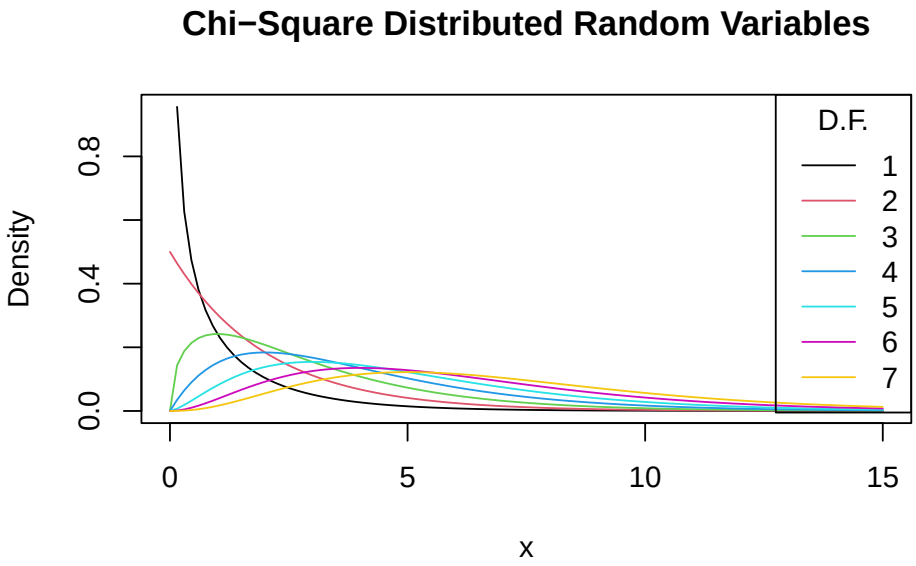


Since the outcomes of a χ^2_M distributed random variable are always positive, the support of the related PDF and CDF is $\mathbb{R}_{\geq 0}$.

As expectation and variance depend (solely!) on the degrees of freedom, the distribution's shape changes drastically if we vary the number of squared standard normals that are summed up. This relation is often depicted by overlaying densities for different M, see the Wikipedia Article.

We reproduce this here by plotting the density of the χ^2_1 distribution on the interval $[0, 15]$ with `"curve()"`. In the next step, we loop over degrees of freedom $M = 2, ..., 7$ and add a density curve for each M to the plot. We also adjust the line color for each iteration of the loop by setting `"col = M"`. At last, we add a legend that displays degrees of freedom and the associated colors.

```r
# plot the density for M=1
curve(dchisq(x, df = 1),
      xlim = c(0, 15),
      xlab = "x",
      ylab = "Density",
      main = "Chi-Square Distributed Random Variables")

# add densities for M=2,...,7 to the plot using a 'for()' loop
for (M in 2:7) {
  curve(dchisq(x, df = M),
        xlim = c(0, 15),
        add = T,
        col = M)
}

# add a legend
legend("topright",
       as.character(1:7),
       col = 1:7 ,
       lty = 1,
       title = "D.F.")
```



Increasing the degrees of freedom shifts the distribution to the right (the mode becomes larger) and increases the dispersion (the distribution's variance grows).

2.2.10.4 The Student t Distribution

Let Z be a standard normal random variable, W a χ^2_ν random variable and further assume that Z and W are independent. Then it holds that

$$\frac{Z}{\sqrt{W/\nu}} =: X \sim t_\nu$$

and X follows a *Student t distribution* (or simply t distribution) with ν degrees of freedom.

The shape of a t_ν distribution depends on ν. t distributions are symmetric, bell-shaped and look similar to a normal distribution, especially when ν

is large. This is not a coincidence: for a sufficiently large ν, the t_ν distribution can be approximated by the standard normal distribution. This approximation works reasonably well for $\nu \geq 30$.

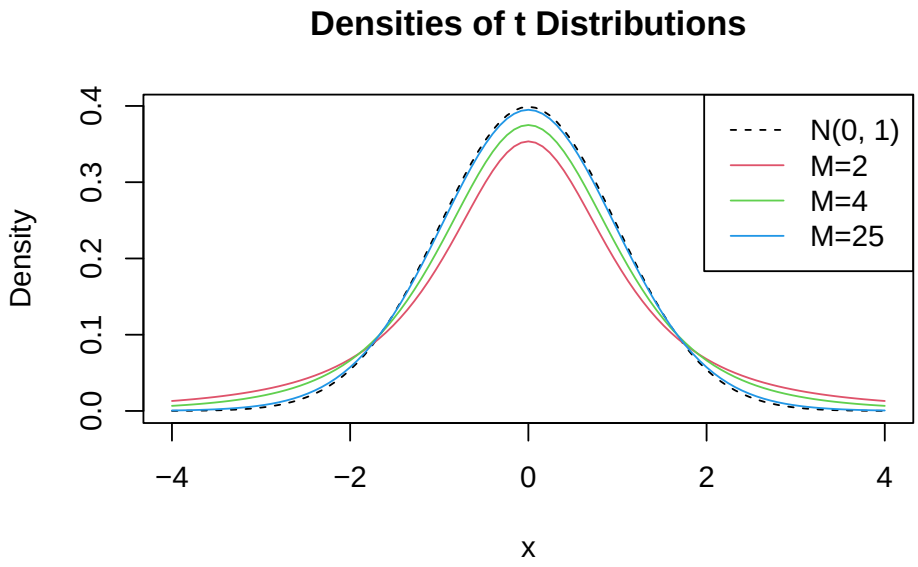
A t_ν distributed random variable X has an expectation if $\nu > 1$ and it has a variance if $\nu > 2$.

$$E(X) = 0, \ M > 1$$
$$\text{Var}(X) = \frac{M}{M-2}, \ M > 2$$

Let us plot some t distributions with different degrees of freedoms ν and compare them to the standard normal distribution.

```
# plot the standard normal density
curve(dnorm(x),
      xlim = c(-4, 4),
      xlab = "x",
      lty = 2,
      ylab = "Density",
      main = "Densities of t Distributions")

# plot the t density for M=2
curve(dt(x, df = 2),
      xlim = c(-4, 4),
      col = 2,
      add = T)

# plot the t density for M=4
curve(dt(x, df = 4),
      xlim = c(-4, 4),
      col = 3,
```

```
      add = T)

# plot the t density for M=25
curve(dt(x, df = 25),
      xlim = c(-4, 4),
      col = 4,
      add = T)

# add a legend
legend("topright",
       c("N(0, 1)", "M=2", "M=4", "M=25"),
       col = 1:4,
       lty = c(2, 1, 1, 1))
```



The plot illustrates that as the degrees of freedom increase, the shape of the

t distribution comes closer to that of a standard normal bell curve. Already for $\nu = 25$ we find little difference to the standard normal density. If ν is small, we find the distribution to have heavier tails than a standard normal.

2.2.10.5 Cauchy Distribution

The Cauchy distribution is a special case of the t distribution corresponding to $\nu = 1$. The density is

$$f(x) = \frac{1}{\pi\left(1 + x^2\right)}.$$

For the Cauchy distribution, the **expectation does not exist** – that is, it has no mean. Let's try to compute the mean of a Cauchy distribution and see what goes wrong. Its mean should be

$$\mu = E(X) = \int_{-\infty}^{\infty} \frac{x dx}{\pi\left(1 + x^2\right)}.$$

In order for this improper integral to exist, we need both integrals $\int_{-\infty}^{0}$ and $\int_{0}^{\infty}$ to be finite. Let's look at the second integral.

$$\int_{0}^{\infty} \frac{x dx}{\pi\left(1 + x^2\right)} = \frac{1}{2\pi} \log\left(1 + x^2\right)\Big|_{0}^{\infty} = \infty$$

Similarly, the other integral, $\int_{-\infty}^{0}$, is $-\infty$. Since they're not both finite, the integral $\int_{-\infty}^{\infty}$ doesn't exist. In other words $\infty - \infty$ is not a number. Thus, the Cauchy distribution has no mean.

What this means in practice is that if you take a sample $x_1, x_2, \ldots, x_n$ from the Cauchy distribution, then the average $\bar{x}$ does not tend to a particular number. Instead, every so often you will get such a huge number, either positive or negative, that the average is overwhelmed by it.

2.2.10.6 The F Distribution

Another ratio of random variables important to econometricians is the ratio of two independent χ^2 distributed random variables that are divided by their degrees of freedom M and n. The quantity

$$\frac{W/M}{V/n} \sim F_{M,n} \quad \text{with} \quad W \sim \chi^2_M \quad , \quad V \sim \chi^2_n$$

follows an F distribution with numerator degrees of freedom M and denominator degrees of freedom n, denoted $F_{M,n}$. The distribution was first derived by George Snedecor but was named in honor of Sir Ronald Fisher.

By definition, the support of both PDF and CDF of an $F_{M,n}$ distributed random variable is $\mathbb{R}_{\geq 0}$.

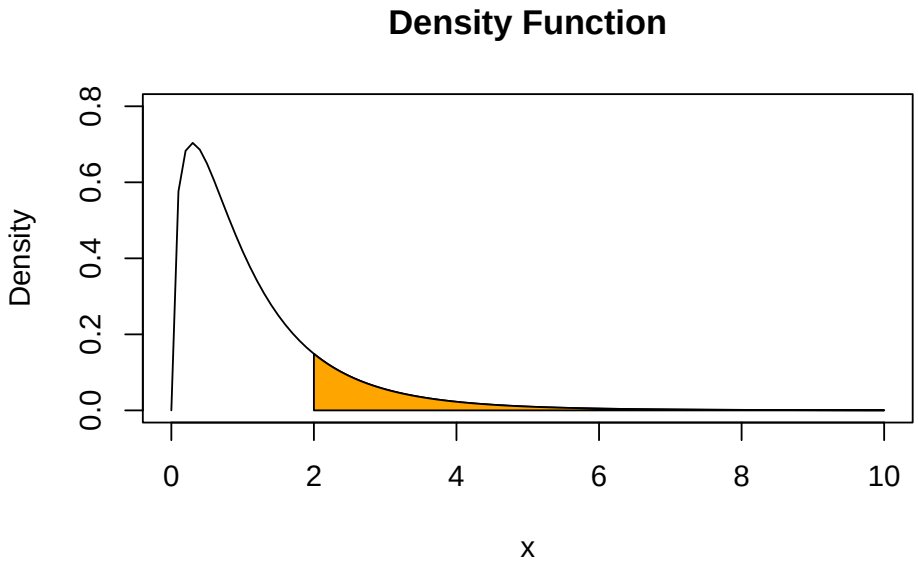
Say we have an F distributed random variable Y with numerator degrees of freedom 3 and denominator degrees of freedom 14 and are interested in $P(Y \geq 2)$. This can be computed with help of the function `"pf()"`. By setting the argument `"lower.tail"` to `"FALSE"` we ensure that R computes $1 - P(Y \leq 2)$, i.e,the probability mass in the tail right of 2.

```
pf(2, df1 = 3, df2 = 14, lower.tail = F)
#> [1] 0.1603538
```

We can visualize this probability by drawing a line plot of the related density and adding a color shading with `"polygon()"`.

```
# define coordinate vectors for vertices of the polygon
x <- c(2, seq(2, 10, 0.01), 10)
y <- c(0, df(seq(2, 10, 0.01), 3, 14), 0)

# draw density of F_{3, 14}
```

```r
curve(df(x ,3 ,14),
      ylim = c(0, 0.8),
      xlim = c(0, 10),
      ylab = "Density",
      main = "Density Function")

# draw the polygon
polygon(x, y, col = "orange")
```



The F distribution is related to many other distributions. An important special case encountered in econometrics arises if the denominator degrees of freedom are large such that the $F_{M,n}$ distribution can be approximated by the $F_{M,\infty}$ distribution which turns out to be simply the distribution of a χ^2_M random variable divided by its degrees of freedom M, i.e.

$$W/M \sim F_{M,\infty} \quad \text{with} \quad W \sim \chi^2_M.$$

Chapter 3

Multiple Linear Regression

Preamble. In the following we focus on case of random designs X (i.e. X being a random variable), since, first, this is the more relevant case in econometrics and, second, it includes the case of fixed designs (i.e. X being deterministic) as a special case ("degenerated random variable' '). Caution: A random X requires us to consider conditional means and variances "given X". That is, if we would be able to resample from the model, we do so by fixing (conditioning on) the in-principle random explanatory variable X.

3.1 Assumptions

The multiple linear regression model is defined by the following assumptions:

Assumption 1: The Linear Model Assumption (Data Generating Process)

$$Y_i = \sum_{k=1}^{K} \beta_k X_{ik} + \varepsilon_i, \quad i = 1, \ldots, n. \tag{3.1}$$

Usually, a constant (intercept) is included, in this case $X_{i1} = 1$ for all i. In the following we will always assume that $X_{i1} = 1$ for all i, unless otherwise

stated.
It is convenient to write equation (3.1) using matrix notation

$$\underset{(1\times 1)}{Y_i} = \underset{(1\times K)}{X_i'}\underset{(K\times 1)}{\beta} + \varepsilon_i, \quad i = 1, \ldots, n,$$

where $X_i = (X_{i1}, \ldots, X_{iK})'$ and $\beta = (\beta_1, \ldots, \beta_K)'$. Stacking all individual rows i leads to

$$\underset{(n\times 1)}{Y} = \underset{(n\times K)}{X}\underset{(K\times 1)}{\beta} + \underset{(n\times 1)}{\varepsilon}, \tag{3.2}$$

where

$$Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}, \quad X = \begin{pmatrix} X_{11} & \ldots & X_{1K} \\ \vdots & \ddots & \vdots \\ X_{n1} & \ldots & X_{nK} \end{pmatrix}, \quad \text{and} \quad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Moreover, we assume that the observed ("obs") data points

$$((Y_{1,obs}, X_{11,obs}, \ldots, X_{1K,obs}), (Y_{2,obs}, X_{21,obs}, \ldots, X_{2K,obs}), \ldots, (Y_{n,obs}, X_{n1,obs}, \ldots, X_{nK,obs})$$

are a realizations of an **independent and identically distributed (i.i.d.)** random sample

$$((Y_1, X_{11}, \ldots, X_{1K}), (Y_2, X_{21}, \ldots, X_{2K}), \ldots, (Y_n, X_{n1}, \ldots, X_{nK}))$$

That is, the ith observed $K+1$ dimensional data point $(Y_{i,obs}, X_{i1,obs}, \ldots, X_{iK,obs}) \in \mathbb{R}^{K+1}$ is a realization of a $K+1$ dimensional random variable $(Y_i, X_{i1}, \ldots, X_{iK}) \in \mathbb{R}^{K+1}$, where $(Y_i, X_{i1}, \ldots, X_{iK})$ has the identical joint distribution as $(Y_j, X_{j1}, \ldots, X_{jK})$ for all $i = 1, \ldots, n$ and all $j = 1, \ldots, n$, and where $(Y_i, X_{i1}, \ldots, X_{iK})$ is independent of $(Y_j, X_{j1}, \ldots, X_{jK})$ for all $i \neq j = 1, \ldots, n$.

Note: Due to (3.1), this i.i.d. assumption is equivalent to assuming that the multivariate random variables $(\varepsilon_i, X_{i1}, \ldots, X_{iK}) \in \mathbb{R}^{K+1}$ are i.i.d. across $i = 1, \ldots, n$.

Remark: Usually, we do not use a different notation for observed realizations $(Y_{i,obs}, X_{i1,obs}, \ldots, X_{iK,obs}) \in \mathbb{R}^{K+1}$ and for the corresponding random variable $(Y_i, X_{i1}, \ldots, X_{iK}) \in \mathbb{R}^{K+1}$ since often both interpretations (random variable and its realizations) can make sense in the same statement and then it depends on the considered context whether the random variables point if view or the realization point of view applies.

Assumption 2: Exogeneity

$$E(\varepsilon_i | X_i) = 0, \quad i = 1, \ldots, n$$

This assumption demands that the mean of the random error term ε_i is zero irrespective of the realizations of X_i. Note that together with the random sampling assumption (in Assumption 1) this assumption implies even **strict** exogeneity $E(\varepsilon_i \,|X) = 0$ since we have independence across $i = 1, \ldots, n$. Under strict exogeneity, the mean of ε_i is zero irrespective of the realizations of $X_1, \ldots, X_n$. The exogeneity assumption is also called "orthogonality assumption" or "mean independence assumption".

Assumption 3: Rank Condition (no perfect multicollinearity)

$$\text{rank}(X) = K \quad \text{a.s.}$$

This assumption demands that the event of one explanatory variable being linearly dependent on the others occurs with a probability equal to zero. (This is the literal translation of the "almost surely (a.s.)" concept.) The assumption implies that $n \geq K$.

This assumption is a bit dicey and its violation belongs to one of the classic problems in applied econometrics (keywords: multicollinearity, dummy variable trap, variance inflation). The violation of this assumption harms any economic interpretation as we cannot disentangle the explanatory variables' individual effects on Y. Therefore, this assumption is also often called an *identification* assumption.

Assumption 4: Error distribution. Depending on the context (i.e., parameter estimation vs. hypothesis testing and small n vs. large n) there are different more or less restrictive assumptions. Some of the most common ones are the following (see also Chapter **??** for a detailed discussion about homoscedastic and heteroscedastic error terms):

- **Conditional Distribution** $\varepsilon_i|X_i \sim f_{\varepsilon|X}$ for all $i = 1, \ldots, n$ and for any distribution $f_{\varepsilon|X}$ such that $\varepsilon_i|X_i$ has two (or more) finite moments.

- **Conditional Normal Distribution** $\varepsilon_i|X_i \sim \mathcal{N}(0, \sigma^2(X_i))$ for all $i = 1, \ldots, n$.

- **i.i.d.** $\varepsilon_i \overset{\text{i.i.d.}}{\sim} f_\varepsilon$ for all $i = 1, \ldots, n$ and for any distribution f_ε such that ε_i has two (or more) finite moments.
  Caution: Assuming that the error terms ε_i are themselves i.i.d. across $i = 1, \ldots, n$ implies that they do not depend on X_i.

- **i.i.d. Normal.** As above, but with $f = \mathcal{N}(0, 1)$, i.e., $\varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$ for all $i = 1, \ldots, n$.

- **Spherical errors ("Gauss-Markov assumptions").** The conditional distributions of $\varepsilon_i|X_i$ may generally depend on X_i, but without

affecting the second moments such that

$$E(\varepsilon_i^2|X_i) = \sigma^2 > 0 \quad \text{for all } i = 1, \ldots, n$$
$$E(\varepsilon_i \varepsilon_j|X) = 0 \quad \text{for all } i \neq j \quad \text{with} \quad i = 1, \ldots, n \quad \text{and} \quad j = 1, \ldots, n.$$

Thus, here one assumes that, for a given realization of X_i, the error process is uncorrelated (i.e. $\text{Cov}(\varepsilon_i, \varepsilon_j\,|X) = E(\varepsilon_i\,\varepsilon_j\,|X) = 0$ for all $i \neq j$) and homoscedastic (i.e. $\text{Var}(\varepsilon_i\,|X) = \sigma^2$ for all i).

Technical Note. When we write that $\text{Var}(\varepsilon_i|X) = \sigma^2$ or $\text{Var}(\varepsilon_i\,|X_i) = \sigma_i^2$, we implicitly assume that these second moments exists and that they are finite.

Homoscedastic versus Heteroscedastic Error Terms

The i.i.d. assumption is not as restrictive as it may seem on first sight. It allows for dependence *between* ε_i and $(X_{i1}, \ldots, X_{iK}) \in \mathbb{R}^K$. That is, the error term ε_i can have a conditional distribution which depends on $(X_{i1}, \ldots, X_{iK})$ (see Chapter 2.2.2.5).

The exogeneity assumption (Assumption 2: Exogeneity) requires that the conditional mean of ε_i is independent of X_i. Besides this, dependencies between ε_i and $X_{i1}, \ldots, X_{iK}$ are allowed. For instance, the variance of ε_i can be a function of $X_{i1}, \ldots, X_{iK}$. If this is the case, ε_i is said to be **heteroscedastic**.

Heteroscedastic error terms: The conditional variances $\text{Var}(\varepsilon_i|X_i = x_i) = \sigma^2(x_i)$ are equal to a non-constant variance-function $\sigma^2(x_i) > 0$ which is a function of the realization $x_i \in \mathbb{R}^K$ of $X_i \in \mathbb{R}^K$.

Example: $\varepsilon_i|X_i \sim U[-0.5|X_{i2}|, 0.5|X_{i2}|]$, with $X_{i2} \sim U[-4, 4]$. This error term is mean independent of X_i since $\mathbb{E}(\varepsilon_i|X_i) = 0$, but it has a heteroscedastic conditional variance since $\text{Var}(\varepsilon_i|X_i) = \frac{1}{12}X_i^2$ depends on X_i.

Sometimes, we need to be more restrictive by assuming that also the variances of the error terms ε_i are independent from X_i. (Higher moments may still depend on X_i.) This assumption leads to **homoscedastic** error terms.

Homoscedastic error terms: The conditional variances $\text{Var}(\varepsilon_i|X_i = x_i) = \sigma^2$ are equal to some constant $\sigma^2 > 0$ for every possible realization $x_i \in \mathbb{R}^K$ of $X_i \in \mathbb{R}^K$.

Sometimes, we need to be even more restrictive by assuming that the error terms ε_i are themselves **i.i.d.** across $i = 1, \ldots, n$. This is more restrictive than the assumption that the multivariate random variables $(\varepsilon_i, X_{i1}, \ldots, X_{iK})$ are i.i.d. across $i = 1, \ldots, n$ since it implies that the whole distribution (not only the first two moments) of ε_i does not depend on X_i. This assumption also implies **homoscedastic** error terms since when the whole distribution of ε_i does not depend on $X_i \in \mathbb{R}^K$ also its variance doesn't depend on X_i. **Example:** For doing small sample inference (see Chapter 4), we need to assume that the error terms ε_i are i.i.d. across $i = 1, \ldots, n$ plus the normality assumption, i.e., $\varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$ for all $i = 1, \ldots, n$ which leads to homoscedastic variances $\text{Var}(\varepsilon_i|X_i) = \sigma^2$ for every possible realization of X_i.

3.1.1 Some Implications of the Exogeneity Assumption

(a) If $E(\varepsilon_i\,|X_i) = 0$ for all $i = 1, \ldots, n$, then the unconditional mean of the error term is zero, i.e.

$$E(\varepsilon_i) \;=\; 0, \quad i = 1, \ldots, n$$

Proof: Using the *Law of Total Expectations* (i.e., $E[E(Z|X)] = E(Z)$) we can rewrite $E(\varepsilon_i)$ as

$$E(\varepsilon_i) = E[E(\varepsilon_i\,|X_i)], \text{ for all } i = 1, \ldots, n.$$

But the exogeneity assumption yields

$$E[E(\varepsilon_i\,|X_i)] = E[0] = 0 \text{ for all } i = 1, \ldots, n,$$

which completes the proof. □

(b) Exogeneity is sometimes also called orthogonality–the reason is the following. Generally, two random variables X and Y are said to be **orthogonal** if their cross moment is zero, i.e. if $E(XY) = 0$.

Under exogeneity, i.e. if $E(\varepsilon_i\,|X_i) = 0$, the regressors and the error term are orthogonal to each other, i.e,

$$E(X_{ik}\,\varepsilon_i) \;=\; 0 \quad \text{for all} \quad i = 1, \ldots, n \quad \text{and} \quad k = 1, \ldots, K.$$

Proof:

$$\begin{aligned} E(X_{ik}\,\varepsilon_i) &= E(E(X_{ik}\,\varepsilon_i\,|X_{ik})) && \text{(By the Law of Total Expectations)} \\ &= E(X_{ik}E(\varepsilon_i\,|X_{ik})) && \text{(By the linearity of cond. expectations)} \end{aligned}$$

Now, to show that $E(X_{ik}\,\varepsilon_i) = 0$, we need to show that $E(\varepsilon_i\,|X_{ik}) = 0$, which is done in the following:
Since X_{ik} is an element of X_i, a slightly more sophisticated use of the *Law of Total Expectations* (i.e., $E(Y|X) = E(E(Y|X, Z)|X)$)

implies that

$$E(\varepsilon_i \,|X_{ik}) = E(E(\varepsilon_i \,|X_i)|X_{ik}).$$

So, the exogeneity assumption, $E(\varepsilon_i \,|X_i) = 0$ yields

$$E(\varepsilon_i \,|X_{ik}) = E(\underbrace{E(\varepsilon_i \,|X_i)}_{=0} \,|X_{ik}) = E(0|X_{ik}) = 0.$$

I.e., we have that $E(\varepsilon_i \,|X_{ik}) = 0$ which allows us to conclude that

$$E(X_{ik}\,\varepsilon_i) = E(X_{ik}E(\varepsilon_i \,|X_{ik})) = E(X_{ik}0) = 0. \quad \square$$

(c) Because the mean of the error term is zero ($E(\varepsilon_i) = 0$ for all i; see point (a)), it follows that the orthogonality property ($E(X_{ik}\,\varepsilon_i) = 0$) is equivalent to a zero-correlation property. I.e., if $E(\varepsilon_i \,|X_i) = 0$, then

$$\text{Cov}(\varepsilon_i, X_{ik}) \;=\; 0 \quad \text{for all} \quad i = 1, \ldots, n \quad \text{and} \quad k = 1, \ldots, K.$$

Proof:

$$\begin{aligned} \text{Cov}(\varepsilon_i, X_{ik}) &= E(X_{ik}\,\varepsilon_i) - E(X_{ik})\,E(\varepsilon_i) \quad \text{(Def. of Cov)} \\ &= E(X_{ik}\,\varepsilon_i) \quad \text{(By point (a): } E(\varepsilon_i) = 0\text{)} \\ &= 0 \quad \text{(By orthogonality result in point (b))} \quad \square \end{aligned}$$

3.2 Deriving the Expression of the OLS Estimator

Similar to Section **??**, we can derive the expression for the OLS estimator $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_K)' \in \mathbb{R}^K$ as the vector-valued minimizing argument

of the sum of squared residuals, $S_n(b)$ with $b \in \mathbb{R}^K$, for a given sample $((Y_1, X_1), \ldots, (Y_n, X_n))$. In matrix terms this is

$$S_n(b) = (Y - Xb)'(Y - Xb) = Y'Y - 2Y'Xb + b'X'Xb.$$

To find the minimizing argument

$$\hat{\beta} = \arg\min_{b \in \mathbb{R}^K} S_n(b)$$

we compute all partial derivatives

$$\underset{(K\times 1)}{\frac{\partial S(b)}{\partial b}} = -2\left(X'Y - X'Xb\right).$$

and set them equal to zero which leads to K linear equations (the "normal equations") in K unknowns. This system of equations defines the OLS estimates, $\hat{\beta}$, for a given data-set:

$$-2\left(X'Y - X'X\hat{\beta}\right) = \underset{(K\times 1)}{0}.$$

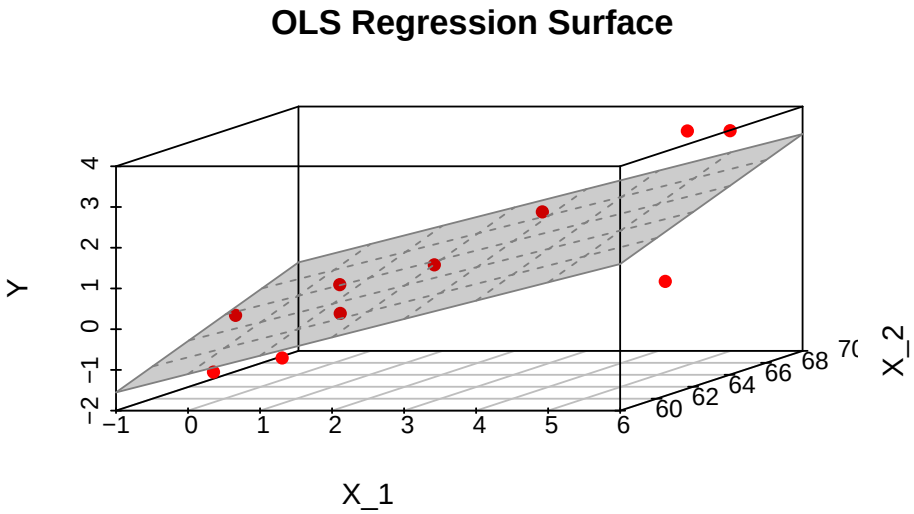
From our rank assumption (Assumption 3) it follows that $X'X$ is an invertible matrix which allows us to solve the equation system by

$$\underset{(K\times 1)}{\hat{\beta}} = \left(X'X\right)^{-1} X'Y$$

The following codes computes the estimate $\hat{\beta}$ for a given realization (Y, X) of the random sample (Y, X).

```
# Some given data
X_1 <- c(1.9,0.8,1.1,0.1,-0.1,4.4,4.6,1.6,5.5,3.4)
X_2 <- c(66, 62, 64, 61, 63, 70, 68, 62, 68, 66)
```

```r
Y   <- c(0.7,-1.0,-0.2,-1.2,-0.1,3.4,0.0,0.8,3.7,2.0)
dataset <-  cbind.data.frame(X_1,X_2,Y)
## Compute the OLS estimation
my.lm <- lm(Y ~ X_1 + X_2, data = dataset)
## Plot sample regression surface
library("scatterplot3d") # library for 3d plots
plot3d <- scatterplot3d(x = X_1, y = X_2, z = Y,
          angle=33, scale.y=0.8, pch=16,
          color ="red", main ="OLS Regression Surface")
plot3d$plane3d(my.lm, lty.box = "solid", col=gray(.5),
          draw_polygon=TRUE)
```



3.3 Some Quantities of Interest

Predicted values and residuals.

- The (OLS) **predicted values**: $\hat{Y}_i = X_i'\hat{\beta}$
  In matrix notation: $\hat{Y} = X \underbrace{(X'X)^{-1}X'Y}_{\hat{\beta}} = P_X Y$

- The (OLS) **residual**: $\hat{\varepsilon}_i = Y_i - \hat{Y}_i$
  In matrix notation: $\hat{\varepsilon} = Y - \hat{Y} = (I_n - X(X'X)^{-1}X')\,Y = M_X Y$

Projection matrices. The matrix $P_X = X(X'X)^{-1}X'$ is the $(n \times n)$ **projection matrix** that projects any vector (from $\mathbb{R}^n$) into the column space spanned by the column vectors of X and $M_X = I_n - X(X'X)^{-1}X' = I_n - P_X$ is the associated $(n \times n)$ **orthogonal projection matrix** that projects any vector (from $\mathbb{R}^n$) into the vector space that is orthogonal to that spanned by X.

The projection matrices P_X and M_X have some nice properties:

(a) P_X and M_X are symmetric, i.e. $P_X = P_X'$ and $M_X = M_X'$.

(b) P_X and M_X are idempotent, i.e. $P_X P_X = P_X$ and $M_X M_X = M_X$.

(c) Moreover $X'P_X = X'$, $P_X X = X$, $X'M_X = 0$, $M_X X = 0$, and $P_X M_X = 0$.

All of these properties follow directly from the definitions of P_X and M_X (check it out). Using these properties one can show that the residual vector

$\hat{\varepsilon} = (\hat{\varepsilon}_1, \ldots, \hat{\varepsilon}_n)'$ is orthogonal to each of the column vectors in X, i.e

$$\begin{aligned} X'\hat{\varepsilon} &= X'M_X Y \quad \text{(By Def. of } M_X\text{)} \\ \Leftrightarrow X'\hat{\varepsilon} &= \underset{(K\times n)}{0}\ \underset{(n\times 1)}{Y} \quad \text{(since } X'M_X = 0\text{)} \\ \Leftrightarrow X'\hat{\varepsilon} &= \underset{(K\times 1)}{0} \end{aligned} \tag{3.3}$$

Note that, in the case with intercept (3.3) implies that $\sum_{i=1}^n \hat{\varepsilon}_i = 0$. Moreover, equation (3.3) implies also that the residual vector $\hat{\varepsilon}$ is orthogonal to the predicted values vector, since

$$\begin{aligned} & X'\hat{\varepsilon} = 0 \\ \Rightarrow\ & \hat{\beta}'X'\hat{\varepsilon} = \hat{\beta}'0 \\ \Leftrightarrow\ & \hat{Y}'\hat{\varepsilon} = 0. \end{aligned}$$

Another insight from equation (3.3) is that the vector $\hat{\varepsilon}$ has to satisfy K linear restrictions ($X'\hat{\varepsilon} = 0$) which means it looses K degrees of freedom. Consequently, the vector of residuals $\hat{\varepsilon}$ has only $n - K$ so-called *degrees of freedom*. This loss of K degrees of freedom also appears in the definition of the *unbiased* variance estimator

$$s^2_{UB} = \frac{1}{n-K}\sum_{i=1}^n \hat{\varepsilon}_i^2. \tag{3.4}$$

Variance decomposition. A further useful result that can be shown using the properties of P_X and M_X is that $Y'Y = \hat{Y}'\hat{Y} + \hat{\varepsilon}'\hat{\varepsilon}$, i.e.

$$\begin{aligned} Y'Y &= (\hat{Y} + \hat{\varepsilon})'(\hat{Y} + \hat{\varepsilon}) \\ &= (P_X Y + M_X Y)'(P_X Y + M_X Y) \\ &= (Y'P_X' + Y'M_X')(P_X Y + M_X Y) \\ &= Y'P_X'P_X Y + Y'M_X'M_X Y + 0 \\ &= \hat{Y}'\hat{Y} + \hat{\varepsilon}'\hat{\varepsilon} \end{aligned} \tag{3.5}$$

Decomposition (3.5) is the basis for the well-known variance decomposition result for OLS regressions. For the OLS regression of the linear model (3.1) with intercept, the total sample variance of the dependent variable $Y_1, \ldots, Y_n$ can be decomposed as following:

$$\underbrace{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i-\bar{Y}\right)^2}_{\text{total sample variance}} = \underbrace{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{Y}_i-\bar{\hat{Y}}\right)^2}_{\text{explained sample variance}} + \underbrace{\frac{1}{n}\sum_{i=1}^{n}\hat{\varepsilon}_i^2}_{\text{unexplained sample variance}}, \tag{3.6}$$

where $\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$ and likewise for $\bar{\hat{Y}}$.

Proof of (3.6):

- As a consequence of (3.3) we have for regressions with intercept that $\sum_{i=1}^{n}\hat{\varepsilon}_i = 0$. Hence, from $Y_i = \hat{Y}_i + \hat{\varepsilon}_i$ it follows that

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^{n} Y_i &= \frac{1}{n}\sum_{i=1}^{n}\hat{Y}_i + \frac{1}{n}\sum_{i=1}^{n}\hat{\varepsilon}_i \\ \bar{Y} &= \bar{\hat{Y}} + 0 \end{aligned} \tag{3.7}$$

- From (3.5) we know that $Y'Y = \hat{Y}'\hat{Y} + \hat{\varepsilon}'\hat{\varepsilon}$, i.e.

$$\begin{aligned} Y'Y &= \hat{Y}'\hat{Y} + \hat{\varepsilon}'\hat{\varepsilon} \\ Y'Y - n\bar{Y}^2 &= \hat{Y}'\hat{Y} - n\bar{Y}^2 + \hat{\varepsilon}'\hat{\varepsilon} \\ Y'Y - n\bar{Y}^2 &= \hat{Y}'\hat{Y} - n\bar{\hat{Y}}^2 + \hat{\varepsilon}'\hat{\varepsilon} \quad \text{(by (3.7) } \bar{Y} = \bar{\hat{Y}}\text{)} \\ \sum_{i=1}^{n} Y_i^2 - n\bar{Y}^2 &= \sum_{i=1}^{n}\hat{Y}_i^2 - n\bar{\hat{Y}}^2 + \sum_{i=1}^{n}\hat{\varepsilon}_i^2 \\ \sum_{i=1}^{n}(Y_i-\bar{Y})^2 &= \sum_{i=1}^{n}(\hat{Y}_i-\bar{\hat{Y}})^2 + \sum_{i=1}^{n}\hat{\varepsilon}_i^2 \quad \square \end{aligned}$$

Coefficient of determination R^2. The larger the proportion of the explained variance, the better is the fit of the model. This motivates the definition of the so-called R^2 coefficient of determination:

$$R^2 = \frac{\sum_{i=1}^n \left(\hat{Y}_i - \bar{\hat{Y}}\right)^2}{\sum_{i=1}^n \left(Y_i - \bar{Y}\right)^2} = 1 - \frac{\sum_{i=1}^n \hat{\varepsilon}_i^2}{\sum_{i=1}^n \left(Y_i - \bar{Y}\right)^2}$$

Obviously, we have that $0 \leq R^2 \leq 1$. The closer R^2 lies to 1, the better is the fit of the model to the observed data. However, a high/low R^2 does not mean a validation/falsification of the estimated model. Any relation (i.e., model assumption) needs a plausible explanation from relevant economic theory. The most often criticized disadvantage of the R^2 is that additional regressors (relevant or not) will always increase the R^2. Here is an example of the problem.

```
set.seed(123)
n     <- 100                  # Sample size
X     <- runif(n, 0, 10)      # Relevant X variable
X_ir  <- runif(n, 5, 20)      # Irrelevant X variable
error <- rt(n, df = 10)*10    # True error
Y     <- 1 + 5 * X + error    # Y variable
lm1   <- summary(lm(Y~X))     # Correct OLS regression
lm2   <- summary(lm(Y~X+X_ir))# OLS regression with X_ir
lm1$r.squared < lm2$r.squared
#> [1] TRUE
```

So, R^2 increases here even though `X_ir` is a completely irrelevant explanatory variable. Because of this, the R^2 cannot be used as a criterion for model selection. Possible solutions are given by penalized criterions such as the

so-called *adjusted* R^2 defined as

$$\overline{R}^2 \;=\; 1 - \frac{\frac{1}{n-K}\sum_{i=1}^{n}\hat{\varepsilon}_i^2}{\frac{1}{n-1}\sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)^2} \leq R^2$$

The adjustment is in terms of the degrees of freedom $n - K$.

3.4 Method of Moments Estimator

The methods of moments estimator exploits the exogeneity assumption that $E(\varepsilon_i \,|X_i) = 0$ for all $i = 1, \ldots, n$ (Assumption 2). Remember that $E(\varepsilon_i \,|X_i) = 0$ implies that $E(X_{ik}\,\varepsilon_i) = 0$ for all $i = 1, \ldots, n$ and all $k = 1, \ldots, K$. The fundamental idea behind "method of moments estimation" is to use the sample analogues of these population moment restrictions $E(X_{ik}\,\varepsilon_i) = 0$, $k = 1, \ldots, K$, for deriving the estimator:

K population moment restrictions		K sample moment restrictions
$\left.\begin{array}{c} E(\varepsilon_i) = 0 \\ E(X_{i2}\,\varepsilon_i) = 0 \\ \vdots \\ E(X_{iK}\,\varepsilon_i) = 0 \end{array}\right\} \Leftrightarrow E(X_i\,\varepsilon_i) = \underset{(K\times 1)}{0}$	$\Bigg\Vert$	$\left.\begin{array}{c} \frac{1}{n}\sum_{i=1}^{n}\hat{\varepsilon}_i = 0 \\ \frac{1}{n}\sum_{i=1}^{n}X_{i2}\hat{\varepsilon}_i = 0 \\ \vdots \\ \frac{1}{n}\sum_{i=1}^{n}X_{iK}\hat{\varepsilon}_i = 0 \end{array}\right\} \Leftrightarrow \frac{1}{n}\sum_{i=1}^{n}X_i\hat{\varepsilon}_i = \underset{(K\times 1)}{0}$

Under our set of assumptions (Ass 1-4), the sample means $n^{-1}\sum_{i=1}^{n}X_i\hat{\varepsilon}_i$ are consistent estimators of the population means $E(X_i\,\varepsilon_i)$. The idea is now to find $\hat{\beta}_0, \ldots, \hat{\beta}_K$ values which lead to residuals $\hat{\varepsilon}_i = Y_i - \sum_{k=1}^{K}\hat{\beta}_k X_{ik}$ that fulfill the above sample moment restrictions. This should in principle be possible since we have a linear system of K equations $\frac{1}{n}\sum_{i=1}^{n}X_i\hat{\varepsilon} = 0$ and K unknowns $\hat{\beta} = (\hat{\beta}_0, \ldots, \hat{\beta}_K)'$. Solving the equation system yields,

$$
\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} X_i\hat{\varepsilon}_i &= \underset{(K\times 1)}{0} \\
\frac{1}{n}\sum_{i=1}^{n} X_i\left(Y_i - X_i'\hat{\beta}\right) &= \underset{(K\times 1)}{0} \\
\frac{1}{n}\sum_{i=1}^{n} X_iY_i - \frac{1}{n}\sum_{i=1}^{n} X_iX_i'\hat{\beta} &= \underset{(K\times 1)}{0} \\
\frac{1}{n}\sum_{i=1}^{n} X_iX_i'\hat{\beta} &= \frac{1}{n}\sum_{i=1}^{n} X_iY_i \\
\hat{\beta} &= \left(\frac{1}{n}\sum_{i=1}^{n} X_iX_i'\right)^{-1}\frac{1}{n}\sum_{i=1}^{n} X_iY_i \\
\hat{\beta} &= \left(X'X\right)^{-1}X'Y
\end{aligned}
$$

which equals the OLS estimator of β; although, we used now a different approach to derive the estimator.

Once again we see the importance of the exogeneity assumption $E(\varepsilon_i\,|X_i)$ which we used here as the starting point for the derivation of the methods of moments estimator. However, unlike with deriving the OLS estimator as the estimator that minimizes the sum of squared residuals, here we derived the estimator *from the exogeineity assumptions*. The method of moments is a very general method, which usually has good properties. We will return to the method of moments several times throughout the semester.

3.5 Unbiasedness of $\hat{\beta}|X$ and $\hat{\beta}$

Once again, but now using matrix algebra, we can show that the OLS (or likewise the Methods-of-Moments estimator) $\hat{\beta} = (X'X)^{-1} X'Y$ is unbiased:

$$\begin{aligned}
E[\hat{\beta}|X] &= E\left[(X'X)^{-1} X'Y|X\right] \\
&= E\left[(X'X)^{-1} X'(X\beta + \varepsilon)|X\right] \\
&= E\left[(X'X)^{-1} X'X\beta + (X'X)^{-1} X'\,\varepsilon\,|X\right] \\
&= \beta + E\left[(X'X)^{-1} X'\,\varepsilon\,|X\right] \\
&= \beta + (X'X)^{-1} X' \underbrace{E[\varepsilon\,|X]}_{=0} = \beta
\end{aligned}$$

$$\begin{aligned}
&\Leftrightarrow\ \text{Bias}[\hat{\beta}|X] = 0 \\
&\Leftrightarrow\ \underbrace{E\big(\text{Bias}[\hat{\beta}|X]\big)}_{=\text{Bias}[\hat{\beta}]} = E(0) \\
&\Leftrightarrow\ E\big(\text{Bias}[\hat{\beta}]\big) = 0
\end{aligned}$$

Note. This result only requires the strict exogeneity assumption $E(\varepsilon\,|X) = 0$ which follows from our Assumption 2 (i.e. $E(\varepsilon_i\,|X_i) = 0$ for all i) together with Assumption 1 (i.e. (Y_i, X_i) is i.i.d. across $i = 1, \ldots, n$). In particular, we did not need to assume homoscedasticity (and it also holds for auto-correlated error terms).

3.6 Variance of $\hat{\beta}|X$

The conditional variance of $\hat{\beta}$ given X is given by

$$\begin{aligned}
\mathrm{Var}(\hat{\beta}|X) =& E\left[(\hat{\beta}-\beta)(\hat{\beta}-\beta)'|X\right] \\
=& E\left[\left((X'X)^{-1}X'\varepsilon\right)\left((X'X)^{-1}X'\varepsilon\right)'|X\right] \\
=& E\left[(X'X)^{-1}X'\varepsilon\,\varepsilon'\,X\,(X'X)^{-1}|X\right] \\
=& (X'X)^{-1}X'E\left[\varepsilon\,\varepsilon'\,|X\right]X\,(X'X)^{-1} \\
=& \underbrace{(X'X)^{-1}X'\overbrace{\mathrm{Var}(\varepsilon\,|X)}^{(n\times n)}X\,(X'X)^{-1}}_{(K\times K)\text{-dimesnional}},
\end{aligned}$$

In the above derivations we used, first, that

$$\begin{aligned}
\hat{\beta}-\beta &= (X'X)^{-1}X'Y-\beta \\
&= (X'X)^{-1}X'(X\beta+\varepsilon)-\beta \\
&= \beta+(X'X)^{-1}X'\varepsilon)-\beta \\
&= (X'X)^{-1}X'\varepsilon
\end{aligned}$$

and, second, that $E\left[\varepsilon\,\varepsilon'\,|X\right] = \mathrm{Var}(\varepsilon\,|X)$ since $E(\varepsilon\,|X) = 0$ under our assumptions. The above expression is the general version of $\mathrm{Var}(\hat{\beta}|X)$ which can be further simplified using specific assumptions on the distribution of the error term ε:

- In case of spherical errors ("Gauss-Markov assumptions"), i.e. no heteroscedasticity and non auto-correlations, we have that $\mathrm{Var}(\varepsilon\,|X) = \sigma^2 I_n$ such that

$$\mathrm{Var}(\hat{\beta}|X) = \sigma^2\underset{(K\times K)}{(X'X)^{-1}}$$

- In case of conditional heteroscedasticitiy, we have that

$$\mathrm{Var}(\varepsilon\,|X) = \begin{pmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n^2 \end{pmatrix} = \mathrm{diag}(\sigma_1^2, \dots, \sigma_n^2),$$

  where the variances σ_i^2 may be functions of X_i, i.e. $\sigma_i^2 \equiv \sigma_i^2(X_i)$. Under conditional heteroscedasticitiy we have the following "sandwich" form

$$\mathrm{Var}(\hat{\beta}|X) = (X'X)^{-1} X' \,\mathrm{diag}(\sigma_1^2, \dots, \sigma_n^2) X \,(X'X)^{-1}.$$

Practical estimation of the standard errors. The diagonal elements in the $(K \times K)$ matrix $\mathrm{Var}(\hat{\beta}|X)$ are the variance expressions for the single estimators $\hat{\beta}_k$ with $k = 1, \dots, K$,

$$\begin{aligned} \mathrm{Var}(\hat{\beta}_k|X) =& \left[(X'X)^{-1} X' \,\mathrm{Var}(\varepsilon\,|X) X \,(X'X)^{-1}\right]_{kk} && \text{(generally)} \\ =& \sigma^2 \left[(X'X)^{-1}\right]_{kk}, && \text{(spherical errors)} \end{aligned}$$

where $[A]_{kl}$ denotes the klth element of A that is in the kth row and lth column of A. Taking square roots leads to the **standard errors** of the estimators $\hat{\beta}_k|X$

$$\begin{aligned} \mathrm{SE}(\hat{\beta}_k|X) =& \left(\left[(X'X)^{-1} X' \,\mathrm{Var}(\varepsilon\,|X) X \,(X'X)^{-1}\right]_{kk}\right)^{1/2} && \text{(generally)} \\ =& \left(\sigma^2 \left[(X'X)^{-1}\right]_{kk}\right)^{1/2}, && \text{(spherical errors)} \end{aligned}$$

Of course, the above expressions for $\mathrm{Var}(\hat{\beta}_k|X)$ and $\mathrm{SE}(\hat{\beta}_k|X)$ are generally useless in practice since we typically do not know the $(n \times n)$ variance matrix $\mathrm{Var}(\varepsilon\,|X)$ or σ^2, but need to estimate them from the data. So, we typically

need to work with

$$\begin{aligned}\widehat{\mathrm{Var}}(\hat{\beta}_k|X) =& \left[(X'X)^{-1} X'\widehat{\mathrm{Var}}(\varepsilon\,|X)X\,(X'X)^{-1}\right]_{kk} && \text{(generally)}\\ =& \hat{\sigma}^2\left[(X'X)^{-1}\right]_{kk}, && \text{(spherical errors)}\end{aligned}$$

and

$$\begin{aligned}\widehat{\mathrm{SE}}(\hat{\beta}_k|X) =& \left(\left[(X'X)^{-1} X'\widehat{\mathrm{Var}}(\varepsilon\,|X)X\,(X'X)^{-1}\right]_{kk}\right)^{1/2} && \text{(generally)}\\ =& \left(\hat{\sigma}^2\left[(X'X)^{-1}\right]_{kk}\right)^{1/2}, && \text{(spherical errors)}\end{aligned}$$

For the case of spherical errors, we already know a possible estimator, namely, $\hat{\sigma}^2 = s^2_{UB} = (n-K)^{-1}\sum_{i=1}^n \hat{\varepsilon}_i^2$. However, finding a reasonable estimator $\widehat{\mathrm{Var}}(\varepsilon\,|X) = \mathrm{diag}(\hat{v}_1,\ldots,\hat{v}_n)$ for the general heteroscedastic case is a little more tricky. The econometric literature knows the following heteroscedasticity-consistent (HC) robust approaches:

$$\begin{aligned}&\text{HC0:} && \hat{v}_i = \hat{\varepsilon}_i^2\\ &\text{HC1:} && \hat{v}_i = \frac{n}{n-K}\hat{\varepsilon}_i^2\\ &\text{HC2:} && \hat{v}_i = \frac{\hat{\varepsilon}_i^2}{1-h_i}\\ &\text{HC3:} && \hat{v}_i = \frac{\hat{\varepsilon}_i^2}{(1-h_i)^2} \quad (\leftarrow \text{Most often used})\\ &\text{HC4:} && \hat{v}_i = \frac{\hat{\varepsilon}_i^2}{(1-h_i)^{\delta_i}}\end{aligned}$$

where $h_i = X_i'(X'X)^{-1}X_i = [P_X]_{ii}$ is the ith diagonal element of the projection matrix P_X, $\bar{h} = n^{-1}\sum_{i=1}^n h_i$, and $\delta_i = \min\{4, h_i/\bar{h}\}$.

Side Note. The statistic $1/n \le h_i \le 1$ is called the "leverage" of X_i, where (by construction) average leverage is $n^{-1}\sum_{i=1}^n h_i = K/n$. Observations X_i

with leverage statistics h_i that greatly exceed the average leverage value K/n are referred to as "high-leverage" observations. High-leverage observations have potentially a large influence on the estimation result. Typically, high-leverage observations X_i distort the estimation results, $\hat{\beta}$, if the absolute value of the corresponding residual $|\hat{\varepsilon}_i|$ is unusually large ("outlier").

The estimator HC0 was suggested in the econometrics literature by White (1980) and is justified by asymptotic arguments. The estimators HC1, HC2 and HC3 were suggested by MacKinnon and White (1985) to improve the performance in small samples. A more extensive study of small sample behavior was carried out by Long and Ervin (2000) which arrive at the conclusion that HC3 provides the best performance in small samples as it inflates the $\hat{\varepsilon}_i$ values which is thought to adjust for the "over-influence" of observations with large leverage values h_i. Cribari-Neto (2004) suggested the estimator HC4 to further improve small sample performance, especially in the presence of influential observations (large h_i values).

Note. In small samples, inference is only possible in a rather restrictive framework requiring spherical and Gaussian errors; see Chapter 4. In large samples, inference requires less restrictive assumptions and one can work with the HC robust standard error estimators; see Chapter 5.

The following R code shows how to compute HC robust variance/standard error estimators:

```
set.seed(2)
n      <- 100
K      <- 3
X      <- matrix(runif(n*(K-1), 2, 10), n, K-1)
X      <- cbind(1,X)
beta   <- c(1,5,5)
# heteroscedastic errors:
```

```r
sigma  <- abs(X[,2] + X[,3])^1.5
error  <- rnorm(n, mean = 0, sd=sigma)
Y      <- beta[1]*X[,1] + beta[2]*X[,2] + beta[3]*X[,3] + error
##
lm_fit <- lm(Y~X -1 )
## Caution! By default R computes the standard errors
## assuming homoscedastic errors. This can lead to
## false inferences under heteroscedastic errors.
summary(lm_fit)$coefficients
#>     Estimate Std. Error    t value    Pr(>|t|)
#> X1 -3.818392  17.797787 -0.2145431 0.830573969
#> X2  5.474779   2.084915  2.6259006 0.010043024
#> X3  5.566453   2.011848  2.7668360 0.006778811

library("sandwich") # HC robust variance estimation
library("lmtest")
## Robust estimation of the variance of \hat{\beta}:
Var_beta_hat_robust <- sandwich::vcovHC(lm_fit, type="HC3")
Var_beta_hat_robust
#>           X1         X2         X3
#> X1 389.83293 -39.198061 -38.026861
#> X2 -39.19806   5.763624   2.260672
#> X3 -38.02686   2.260672   5.437287
## Corresponding regression-output:
lmtest::coeftest(lm_fit, vcov = Var_beta_hat_robust)
#>
#> t test of coefficients:
#>
#>    Estimate Std. Error t value Pr(>|t|)
#> X1  -3.8184    19.7442 -0.1934  0.84706
```

```
#> X2   5.4748     2.4008  2.2804  0.02477 *
#> X3   5.5665     2.3318  2.3872  0.01892 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Observe that the HC robust variance estimation leads to larger variances than the classic variance estimation for homoscedastic errors. This is typically, but not always, the case.

3.7 The Gauss-Markov Theorem

The Gauss-Markov Theorem. Let's assume Assumptions 1-4 hold with spherical errors, i.e., with $E(\varepsilon\,\varepsilon'\,|X) = \sigma^2 I_n$. Then the Gauss-Markov theorem states that of all linear and unbiased estimators, the OLS (or Methods of Moments) estimator $\hat{\beta} = (X'X)^{-1}X'Y$ will have the smallest variance, in a matrix sense. That is, for any alternative linear and unbiased estimator $\tilde{\beta}$ we have that

$$\begin{aligned} &\mathrm{Var}(\tilde{\beta}|X) \geq \mathrm{Var}(\hat{\beta}|X) \quad \text{(“in the matrix sense”)} \\ \Leftrightarrow &\mathrm{Var}(\tilde{\beta}|X) - \mathrm{Var}(\hat{\beta}|X) = \underset{(K\times K)}{D}, \end{aligned}$$

where D is a *positive semidefinite* $(K \times K)$ matrix, i.e., $a'Da \geq 0$ for any K-dimensional vector $a \in \mathbb{R}^K$. Observe that this implies that $\mathrm{Var}(\tilde{\beta}_k|X) \geq \mathrm{Var}(\hat{\beta}_k|X)$ for any $k = 1, \ldots, K$.

Proof of the Gauss-Markov Theorem. Since $\tilde{\beta}$ is assumed to be linear in Y, we can write

$$\tilde{\beta} = CY,$$

where C is some $(K \times n)$ matrix, which is a function of X and/or nonrandom components. Adding a $(K \times n)$ zero matrix 0 yields

$$\tilde{\beta} = \Big(C \overbrace{- (X'X)^{-1} X' + (X'X)^{-1} X'}^{=0}\Big) Y.$$

Let now $D = C - (X'X)^{-1} X'$, then

$$\begin{aligned}
\tilde{\beta} &= \left(D + (X'X)^{-1} X'\right) Y \\
\tilde{\beta} &= DY + (X'X)^{-1} X'Y \\
\tilde{\beta} &= D\left(X\beta + \varepsilon\right) + (X'X)^{-1} X'Y \\
\tilde{\beta} &= DX\beta + D\varepsilon + \hat{\beta} \qquad (3.8)
\end{aligned}$$

$$\begin{aligned}
\Rightarrow \quad E(\tilde{\beta}|X) &= E(DX\beta|X) + E(D\varepsilon|X) + E(\hat{\beta}|X) \\
&= DX\beta + 0 + \beta \qquad (3.9)
\end{aligned}$$

Since $\tilde{\beta}$ is (by assumption) unbiased, we have that $E(\tilde{\beta}|X) = \beta$. Therefore, (3.9) implies that $DX = 0_{(K \times K)}$ since we must have that $E(\tilde{\beta}|X) = DX\beta + 0 + \beta = \beta$. Plugging $DX = 0$ into (3.8) yields,

$$\begin{aligned}
\tilde{\beta} &= D\varepsilon + \hat{\beta} \\
\tilde{\beta} - \beta &= D\varepsilon + (\hat{\beta} - \beta) \\
\tilde{\beta} - \beta &= D\varepsilon + (X'X)^{-1} X'\varepsilon \\
\tilde{\beta} - \beta &= \left(D + (X'X)^{-1} X'\right) \varepsilon, \qquad (3.10)
\end{aligned}$$

where we used that

$$\begin{aligned}
\hat{\beta} - \beta &= (X'X)^{-1}X'Y - \beta \\
&= (X'X)^{-1}X'(X\beta + \varepsilon) - \beta \\
&= (X'X)^{-1}X'\varepsilon\,.
\end{aligned}$$

So,

$$\begin{aligned}
\operatorname{Var}(\tilde{\beta}|X) &= \operatorname{Var}(\tilde{\beta} - \beta|X) && \text{(since } \beta \text{ is not random)} \\
&= \operatorname{Var}((D + (X'X)^{-1}X')\varepsilon|X) && \text{(from equation (3.10))} \\
&= (D + (X'X)^{-1}X') \operatorname{Var}(\varepsilon|X)(D' + X(X'X)^{-1}) \\
&= \sigma^2(D + (X'X)^{-1}X')I_n(D' + X(X'X)^{-1}) \\
&= \sigma^2 \left(DD' + (X'X)^{-1}\right) && \text{(using that } DX = 0) \\
&\geq \sigma^2(X'X)^{-1} && \text{(Since } DD' \text{ is pos. semidef.)} \\
&= \operatorname{Var}(\hat{\beta}|X).
\end{aligned}$$

Showing that DD' is really positive semidefinite:

$$a'DD'a = (D'a)'(D'a) = \tilde{a}'\tilde{a} \geq 0,$$

where $\tilde{a}$ is a K dimensional column-vector.

Note. To reiterate: The unbiasedness of $\hat{\beta}$ did not depend on any assumptions about the distribution of ε, except that $E(\varepsilon\,|X) = 0$ which follows from our Assumption 2 together with the i.i.d. assumption in Assumption 1. Once we imposed additionally the assumption of spherical errors $E(\varepsilon\,\varepsilon\,|X) = \sigma^2 I_n$ we can show that $\hat{\beta}$ has the smallest variance of all linear unbiased estimators.

3.8 Practice: Real Data

The following practice part is taken from Kleiber and Zeileis (2008).

The R package `AER` contains many useful functions and data sets for applied regression analysis. In the following we consider `CPS1988` data frame collected in the March 1988. Current Population Survey (CPS) by the US Census Bureau. These are cross-section data on males aged between 18 and 70 with annual income greater than 50 US-Dollar in the year 1991.

```
## install.packages("AER")
library("AER") ## load the R package
data(CPS1988)  ## attach the data
```

The simplest option to get a summary of the data is `summary(CPS1988)`:

```
summary(CPS1988)
#>       wage             education       experience   ethnicity     sm
#>  Min.   :   50.05   Min.   : 0.00   Min.   :-4.0   cauc:25923   no
#>  1st Qu.:  308.64   1st Qu.:12.00   1st Qu.: 8.0   afam: 2232   yes
#>  Median :  522.32   Median :12.00   Median :16.0
#>  Mean   :  603.73   Mean   :13.07   Mean   :18.2
#>  3rd Qu.:  783.48   3rd Qu.:15.00   3rd Qu.:27.0
#>  Max.   :18777.20   Max.   :18.00   Max.   :63.0
#>        region     parttime
#>  northeast:6441   no :25631
#>  midwest  :6863   yes: 2524
#>  south    :8760
#>  west     :6091
#>
#>
```

Here, `wage` is the wage in dollars per week, `education` and `experience` are measured in years, and `ethnicity` is a factor with levels Caucasian (`"cauc"`) and African-American (`"afam"`). There are three further factors, `smsa`, `region`, and `parttime`, indicating residence in a standard metropolitan statistical area (SMSA), the region within the United States of America, and whether the individual works part-time. Experience is not actual experience but a proxi for potential experiance computed as `age - eduction - 6`; thus

this quantity may be negative which is actually the case for 438 observations in the `CPS1988` data frame.

Our model of interest is

$$\begin{aligned}\log(\texttt{wage}) =& \beta_1 + \beta_2 \texttt{experiences} + \beta_3 \texttt{experiences}^2 \\ &+ \beta_4 \texttt{education} + \beta_5 \texttt{ethnicity} + \varepsilon\end{aligned}$$

You can fit this model in R as following:

```
cps_lm <- lm(log(wage) ~ experience + I(experience^2) +
               education + ethnicity, data = CPS1988)
```

The formula in the `lm()` call takes into account the semilogarithmic form and also specifies the squared regressor $\texttt{experiences}^2$. It has to be insulated by `I()` so that the operator `^` has its original arithmetic meaning (and not its meaning as a formula operator for specifying interactions; see below).

```
## Regression output without robust standard errors (SEs):
## summary(cps_lm) ## Generally, do not do this.

## But do this:
## Heteroscedasticity robust variance estimation:
library("sandwich")
library("lmtest")
## Robust estimation of the variance of \hat{\beta}:
Var_beta_hat_robust <- sandwich::vcovHC(cps_lm, type="HC3")
## Regression output table with robust SEs:
lmtest::coeftest(cps_lm, vcov = Var_beta_hat_robust)
#>
#> t test of coefficients:
```

```
#>
#>                     Estimate  Std. Error t value  Pr(>|t|)
#> (Intercept)      4.3214e+00  2.0614e-02 209.630 < 2.2e-16 ***
#> experience       7.7473e-02  1.0188e-03  76.046 < 2.2e-16 ***
#> I(experience^2) -1.3161e-03  2.3486e-05 -56.035 < 2.2e-16 ***
#> education        8.5673e-02  1.3755e-03  62.283 < 2.2e-16 ***
#> ethnicityafam   -2.4336e-01  1.3119e-02 -18.550 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The summary reveals that all coefficients have the expected sign, and the corresponding variables are highly significant (not surprising in a sample, $n = 28155$, as large as the present one). Specifically, according to this specification, the return on education is 8.57% per year.

Interpretation of the Results. The linear model structure facilitates a very simple interpration. For the unknown parameters $\beta_1, \ldots, \beta_K$ we have that

$$\frac{\partial E[Y_i|X_i]}{\partial X_{ik}} = \beta_k \qquad \text{with} \qquad E[Y_i|X_i] = \beta_1 + \beta_2 X_{i2} + \cdots + \beta_2 X_{iK}.$$

That is, β_k is the true (unknown) **marginal effect** of a one unit change in X_{ik} on Y_i. Therefore, $\hat{\beta}_k$ is the **estimated marginal effect** of a one unit change in X_{ik} on Y_i:

$$\frac{\partial \widehat{E[Y_i|X_i]}}{\partial X_{ik}} = \hat{\beta}_k \qquad \text{with} \qquad \widehat{E[Y|X]} = \hat{\beta}_1 + \hat{\beta}_2 X_{i2} + \cdots + \hat{\beta}_2 X_{iK}.$$

Caution! Econometricians usually would argue that you cannot interpret the above empirical result *causally* since `wage` is also influenced by many

further factors that cannot be measured easily such as, for instance, `personal drive to get stuff done` and these missing factors potentially correlate with `education`.

Dummy Variables, Contrast Codings, and Interactions

Note that the level `"cauc"` of ethnicity does not occur in the output, as it is taken as the reference category. Hence, there is only one ethnicity effect, which gives the difference in intercepts between the `"afam"` and the `"cauc"` groups. In statistical terminology, this is called a "treatment contrast" (where the "treatment" `"afam"` is compared with the reference group `"cauc"`) and corresponds to what is called a "dummy variable" (or "indicator variable") for the level `"afam"` in econometric jargon.

In R, (unordered) factors are automatically handled like this when they are included in a regression model. Internally, R produces a dummy variable for each level of a factor and resolves the resulting overspecification of the model (if an intercept or another factor is included in the model) by applying "contrasts"; i.e., a constraint on the underlying parameter vector. Contrasts are attributed to each factor and can be queried and changed by `contrasts()`. The default for unordered factors is to use all dummy variables except the one for the reference category (`"cauc"` in the example above). This is typically what is required for fitting econometric regression models, and hence changing the contrasts is usually not necessary.

The above result shows that there is an associative effect of `ethnicity` on `wage`, since `ethnicityafam`, i.e., the `"afam"`-level of the `ethnicity`-factor variables has a significant mean-shift effect. In such cases, one has often also further heterogenouse effects that can be considered using **interactions**. The following code checks whether `education` has a different slope value for `cauc`-people and `afam`-people, by computing the **interaction-effect** between `education` and `ethnicity`. The formula-

notation in R for this is `education*ethnicity` which automatically adds the single regressors `education` and `ethnicity` and the interacted regressor `education`×`ethnicity`:

```
cps_lm_2 <- lm(log(wage) ~ experience + I(experience^2) +
                 education*ethnicity, data = CPS1988)
## Robust estimation of the variance of \hat{\beta}:
Var_beta_hat_robust <- sandwich::vcovHC(cps_lm_2, type="HC3")
## Regression output table with robust SEs:
lmtest::coeftest(cps_lm_2, vcov = Var_beta_hat_robust)
#>
#> t test of coefficients:
#>
#>                              Estimate  Std. Error  t value Pr(>|t|)
#> (Intercept)                4.3131e+00  2.1049e-02 204.9085  < 2e-16 *
#> experience                 7.7520e-02  1.0174e-03  76.1959  < 2e-16 *
#> I(experience^2)           -1.3179e-03  2.3449e-05 -56.2013  < 2e-16 *
#> education                  8.6312e-02  1.4152e-03  60.9905  < 2e-16 *
#> ethnicityafam             -1.2389e-01  6.7231e-02  -1.8427  0.06538 .
#> education:ethnicityafam -9.6481e-03  5.3174e-03  -1.8144  0.06962 .
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This result suggests that one year more `education` increases the average `wage` by 0.0863 for Caucasian-people (`cauc`), but only by $0.0863 - 0.0096 = 0.0767$ for African-American-people (`afam`). However, the direct effect of `ethnicity` and the interaction effect `education`×`ethnicity` are here not significant at the usual 0.05 level, but only at the 0.10 level.

The Function `I()`

Some further details on the specification of regression model formulas in R are in order. We have already seen that the arithmetic operator `+` has a different meaning in formulas: it is employed to add regressors (main effects). Additionally, the operators `:`, `*`, `/`, `^` have special meanings, all related to the specification of so-called interaction effects.

To be able to use the arithmetic operators in their original meaning in a formula, they can be protected from the formula interpretation by insulating them inside a function, as in `log(x1 * x2)`. If the problem at hand does not require a transformation, R's `I()` function can be used, which returns its argument "as is". This was used for computing experience squared in the regression above.

3.9 Practice: Simulation

3.9.1 Behavior of the OLS Estimates for Resampled Data (conditionally on X_i)

Usually, we only observe the **estimates** $\hat{\beta}_0$ and $\hat{\beta}_1$ computed for a given data set. However, in order to understand the statistical properties of the **estimators** $\hat{\beta}_0$ and $\hat{\beta}_1$ we need to view them as random variables which yield different realizations in repeated samples generated from (**??**) conditionally on $X_1, \ldots, X_n$. This allows us then to think about questions like:

- "Is the estimator able to estimate the unknown parameter-value correctly on average (conditionally on a given set of $X_1, \ldots, X_n$)?"
- "Are the estimation results more precise if we have more data?"

A first idea about the statistical properties of the estimators $\hat{\beta}_0$ and $\hat{\beta}_1$ can be gained using Monte Carlo simulations as following.

```
## Sample sizes
n_small      <-  10 # small sample size
n_large      <- 100 # large sample size

## True parameter values
beta0 <- 1
beta1 <- 1

## Generate explanatory variables (random design)
X_n_small  <- runif(n_small, min = 1, max = 10)
X_n_large  <- runif(n_large, min = 1, max = 10)

## Monte-Carlo (MC) Simulation
## 1. Generate data
## 2. Compute and store estimates
## Repeat steps 1. and 2. many times
set.seed(3)
## Number of Monte Carlo repetitions
## How many samples to draw from the models
rep          <- 1000

## Containers to store the lm-results
n_small_list <- vector(mode = "list", length = rep)
n_large_list <- vector(mode = "list", length = rep)

for(r in 1:rep){
## Sampling from the model conditionally on X_n_small
```

```
error_n_small     <- rnorm(n_small, mean = 0, sd = 5)
Y_n_small         <- beta0 + beta1 * X_n_small + error_n_small
n_small_list[[r]] <- lm(Y_n_small ~ X_n_small)
## Sampling from the model conditionally on X_n_large
error_n_large     <- rnorm(n_large, mean = 0, sd = 5)
Y_n_large         <- beta0 + beta1 * X_n_large + error_n_large
n_large_list[[r]] <- lm(Y_n_large ~ X_n_large)
}

## Reading out the parameter estimates
beta0_estimates_n_small <- rep(NA, rep)
beta1_estimates_n_small <- rep(NA, rep)
beta0_estimates_n_large <- rep(NA, rep)
beta1_estimates_n_large <- rep(NA, rep)
for(r in 1:rep){
beta0_estimates_n_small[r] <- n_small_list[[r]]$coefficients[1]
beta1_estimates_n_small[r] <- n_small_list[[r]]$coefficients[2]
beta0_estimates_n_large[r] <- n_large_list[[r]]$coefficients[1]
beta1_estimates_n_large[r] <- n_large_list[[r]]$coefficients[2]
}
```

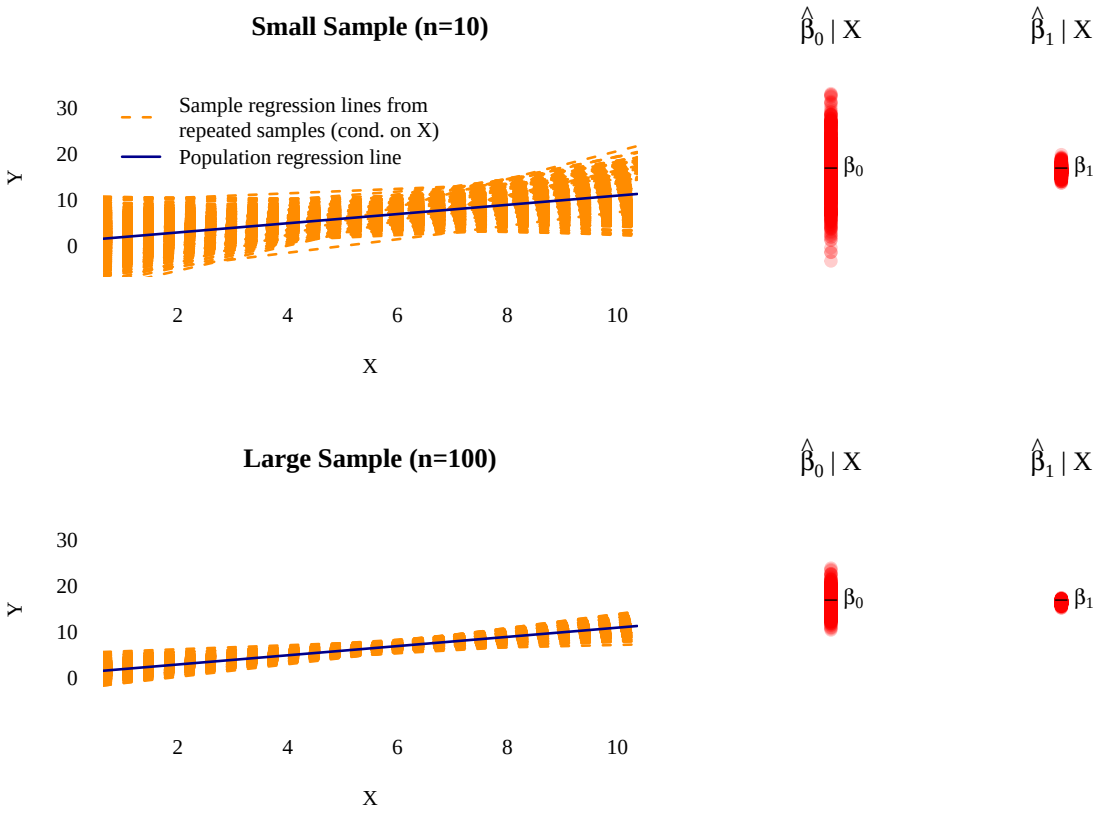
Now, we have produced realizations of the estimators $\hat{\beta}_0|X$ and $\hat{\beta}_1|X$ conditionally on

$$X = \begin{pmatrix} 1 & X_1 \\ \vdots & \vdots \\ 1 & X_n \end{pmatrix}$$

and we have saved these realizations in `beta0_estimates_n_small`, `beta1_estimates_n_small`, `beta0_estimates_n_large`, and `beta1_estimates_n_large`. This allows us to visualize the behavior of the OLS estimates for the repeatedly sampled data (conditionally on X_i).

```r
## Plotting the results
library("scales") # alpha() produces transparent colors

## Define a common y-axis range
y_range <- range(beta0_estimates_n_small,
                 beta1_estimates_n_small)*1.1

## Generate the plot
par(family = "serif") # Serif fonts
## Layout of plotting area
layout(matrix(c(1:6), 2, 3, byrow = TRUE), widths = c(3,1,1))
## Plot 1
plot(x=0, y=0, axes=FALSE, xlab="X", ylab="Y", type="n",
     xlim=c(1,10), ylim=c(-5,35), main="Small Sample (n=10)")
axis(1, tick = FALSE); axis(2, tick = FALSE, las = 2)
for(r in 1:rep){
abline(n_small_list[[r]], lty=2, lwd = 1.3, col="darkorange")
}
abline(a = beta0, b = beta1, lwd=1.3, col="darkblue")
legend("topleft", col=c("darkorange", "darkblue"), legend=c(
"Sample regression lines from\nrepeated samples (cond. on X)",
                  "Population regression line"),
       lwd=1.3, lty=c(2,1), bty="n")
## Plot 2
plot(x=rep(0,rep), y=beta0_estimates_n_small, axes=FALSE,
     xlab="", ylab="", pch=19, cex=1.2, ylim=y_range,
  main=expression(hat(beta)[0]~'|'~X), col=alpha("red",0.2))
points(x = 0, y=beta0, pch="-", cex = 1.2, col="black")
text(x=0, y=beta0, labels = expression(beta[0]), pos = 4)
## Plot 3
```

```
plot(x=rep(0,rep), y=beta1_estimates_n_small, axes=FALSE,
     xlab="", ylab="", pch=19, cex=1.2, ylim=y_range,
  main=expression(hat(beta)[1]~'|'~X), col=alpha("red",0.2))
points(x = 0, y=beta1, pch="-", cex = 1.2, col="black")
text(x=0, y=beta1, labels = expression(beta[1]), pos = 4)
## Plot 4
plot(x=0, y=0, axes=FALSE, xlab="X", ylab="Y", type="n",
     xlim=c(1,10), ylim=c(-5,35), main="Large Sample (n=100)")
axis(1, tick = FALSE); axis(2, tick = FALSE, las = 2)
for(r in 1:rep){
abline(n_large_list[[r]], lty=2, lwd = 1.3, col="darkorange")
}
abline(a = beta0, b = beta1, lwd=1.3, col="darkblue")
## Plot 5
plot(x=rep(0,rep), y=beta0_estimates_n_large, axes=FALSE,
     xlab="", ylab="", pch=19, cex=1.2, ylim=y_range,
  main=expression(hat(beta)[0]~'|'~X), col=alpha("red",0.2))
points(x = 0, y=beta0, pch="-", cex = 1.2, col="black")
text(x=0, y=beta0, labels = expression(beta[0]), pos = 4)
## Plot 6
plot(x=rep(0,rep), y=beta1_estimates_n_large, axes=FALSE,
     xlab="", ylab="", pch=19, cex=1.2, ylim=y_range,
  main=expression(hat(beta)[1]~'|'~X), col=alpha("red",0.2))
points(x=0, y=beta1, pch="-", cex = 1.2, col="black")
text(x=0, y=beta1, labels = expression(beta[1]), pos = 4)
```



This are promising plots:

- The realizations of $\hat{\beta}_0|X$ and $\hat{\beta}_1|X$ are scattered around the true (unknown) parameter values β_0 and β_1 for both small and large samples.
- The realizations of $\hat{\beta}_0|X$ and $\hat{\beta}_1|X$ concentrate more and more around the true (unknown) parameter values β_0 and β_1 as the sample size increases.

However, this was only a simulation for one specific data generating process. Such a Monte Carlo simulation does not allow us to generalize these properties.

Next we use theoretical arguments to show that these properties also hold in general.

Chapter 4

Small Sample Inference

Note on small sample sizes. Sometimes sample sizes of $n < 30$ are referred to as "small samples" since one often hopes that the central limit theorem is helping out for sample sizes $n \geq 30$. However, this is a very dangerous rule of thumb! A sufficiently large n that allows us to rely on the central limit theorem depends on many different distributional aspects. In this chapter, we consider the case where we cannot rely on the central limit theorem.

Exact inference. This chapter considers *exact* inference using the multiple linear regression model. By *exact* we mean correct distributions for each sample size n. That is, no asymptotic (large $n \to \infty$) arguments will be used.

Assumptions. Recall that we, in general, did not impose a complete distributional assumption on ε in Assumption 4 – the i.i.d. normal case in Assumption 4 was only one possible *option*. However, to do exact inference, the normality Assumption on the error terms is not a mere option, but a *necessity*. So for this chapter we assume that Assumptions 1-3 from Chapter 3 hold and that additionally the following assumption holds:

Assumption 4*: Error distribution. For small sample cases, we assume that the error terms are **i.i.d. normal**, i.e., $\varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$ for all $i = 1, \ldots, n$ which leads to spherical errors. That is,

$$\varepsilon \sim \mathcal{N}\left(0, \sigma^2 I_n\right),$$

where $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)'$.

Normality of $\hat{\beta}$. Under Assumptions 1-4* it can be shown that

$$\hat{\beta}_n | X \sim \mathcal{N}\left(\beta, \text{Var}(\hat{\beta}_n | X)\right), \tag{4.1}$$

where $\text{Var}(\hat{\beta}_n | X) = \sigma^2 (X'X)^{-1}$.

This result follows from noting that $\hat{\beta}_n = (X'X)^{-1}X'Y = \beta + (X'X)^{-1}X'\varepsilon$ and because $(X'X)^{-1}X'\varepsilon$ is just a linear combination of the normally distributed error terms ε which, therefore, is again normally distributed, conditionally on X. Note that (4.1) is the *exact* small sample distribution of $\hat{\beta}_n | X$ and fully relies on the conditional normality assumption Assumption 4*.

Note. The subscript n in $\hat{\beta}_n$ is here only to emphasize that the distribution of $\hat{\beta}_n$ depends on n; we will, however, often simply write $\hat{\beta}$.

4.1 Hypothesis Tests about Multiple Parameters

Let us consider the following system of q-many null hypotheses:

$$H_0 : \underset{(q \times K)}{R} \underset{(K \times 1)}{\beta} - \underset{(q \times 1)}{r} = \underset{(q \times 1)}{0},$$

where the $(q \times K)$ matrix R and the q-vector $r = (r_1, \ldots, r_q)'$ are chosen by the statistician to specify her/his null hypothesis about the unknown true parameter vector β. To make sure that there are no redundant equations, it is required that $\text{rank}(R) = q$.

We must also specify the alternative against which we are testing the null hypothesis, for instance

$$H_A : R\beta - r \neq 0$$

The above multiple parameter hypotheses cover also the special case of single parameter hypothesis; for instance, by setting $R = (0, 1, 0 \ldots, 0)$ and $r = 0$ one get's

$$\begin{aligned} H_0 : &\quad \beta_k = 0 \\ H_A : &\quad \beta_k \neq 0 \end{aligned}$$

Under our assumptions (Assumptions 1 to 4*), we have that

$$(R\hat{\beta}_n - r)|X \sim \mathcal{N}\left(R\beta - r, R\,\text{Var}(\hat{\beta}_n|X)R'\right).$$

So, the realizations of $(R\hat{\beta}_n - r)|X$ will scatter around the *unknown* $(R\beta - r)$ in a non-systematical, Gaussian fashion. Therefore, if the null hypothesis is correct (i.e., $(R\beta - r) = 0$), the realizations of $(R\hat{\beta}_n - r)|X$ scatter around the q-vector 0. If, however, the alternative hypothesis is correct (i.e., $(R\beta - r) = a \neq 0$), the realizations of $R\hat{\beta}_n - r|X$ scatter around the q-vector $a \neq 0$. So, under the alternative hypothesis, there will be a systematic location-shift of the q-dimensional random variable $R\hat{\beta}_n|X$ away from r which we try to detect using statistical hypothesis testing.

4.1.1 The Test Statistic and its Null Distribution

The fact that $(R\hat{\beta}_n - r) \in \mathbb{R}^q$ is a q-dimensional random variable makes it a little bothersome to use as a test-statistic. Fortunately, we can turn $(R\hat{\beta}_n - r)$

into a scalar-valued test statistic using the following quadratic form:

$$W = \underbrace{(R\hat{\beta}_n - r)'}_{(1\times q)} \underbrace{[R\,\mathrm{Var}(\hat{\beta}_n|X)R']^{-1}}_{(q\times q)} \underbrace{(R\hat{\beta}_n - r)}_{(q\times 1)}$$

Note that the test statistic W is simply measuring the distance (it's a weighted L2-distance) between the two q-vectors $R\hat{\beta}_n$ and r. Moreover, under the null hypothesis (i.e., if the null hypothesis is true), W is just a sum of q-many independent squared standard normal random variables. Therefore, under the null hypothesis, W is $\chi^2(q)$ distributed with q degrees of freedom (see Section 2.2.10.3),

$$W = (R\hat{\beta}_n - r)'[R\,\mathrm{Var}(\hat{\beta}_n|X)R']^{-1}(R\hat{\beta}_n - r) \overset{H_0}{\sim} \chi^2(q).$$

Usually, we do not know $\mathrm{Var}(\hat{\beta}_n|X)$ and, therefore, have to estimate this quantity. Unfortunately, the general heteroscedasticity consistent robust estimators $\widehat{\mathrm{Var}_{\mathrm{HC0}}}(\hat{\beta}_n|X)$, ..., $\widehat{\mathrm{Var}_{\mathrm{HC4}}}(\hat{\beta}_n|X)$ from Chapter 3 are only asymptotically justified and, therefore, inappropriate for *exact* small sample inference.

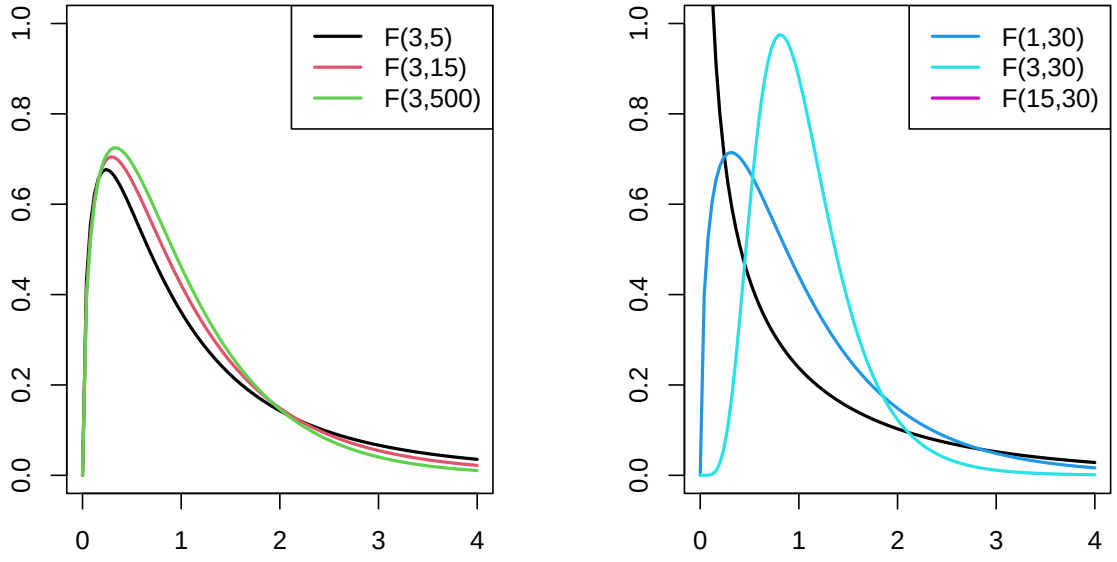
Thus, for truly *exact* finite sample inference, we need a variance estimator for which we can derive the *exact* small sample distribution. Therefore, we assume in Assumption 4* spherical errors (i.e., $\mathrm{Var}(\varepsilon\,|X) = I_n\sigma^2$) which yield that $\mathrm{Var}(\hat{\beta}_n|X) = \sigma^2(X'X)^{-1}$, and where σ^2 can be estimated by $s^2_{UB} = (n-K)^{-1}\sum_{i=1}^n \hat{\varepsilon}_i^2$. From the normality assumption in Assumption 4*, it follows then that $\sigma^{-2}(n-K)s^2_{UB} \sim \chi^2(n-K)$. This then leads to the following exact null distribution of

$$F = (R\hat{\beta}_n - r)'[R(s^2_{UB}(X'X)^{-1})R']^{-1}(R\hat{\beta}_n - r)/q \overset{H_0}{\sim} F(q, n-K), \quad (4.2)$$

where $F(q, n-K)$ denotes the F-distribution with q numerator and $n-K$ denominator degrees of freedom (see, for instance, Hayashi, 2000, Ch. 1). By contrast to W, F is a practically useful test-statistic, and we can use

observed values F_{obs} to measure the distance of our estimate $R\hat{\beta}_n$ from value r. Observed values, F_{obs}, that are unusually large under the null hypothesis, lead to a rejection of the null hypothesis. The null distribution $F(q, n-K)$ of F is used to judge what's unusually large under the null hypothesis.

The F distribution. The F distribution is a ratio of two χ^2 distributions. It has two parameters: the numerator degrees of freedom, and the denominator degrees of freedom. Each combination of the parameters yields a different F distribution. See Section 2.2.10.6 for more information on the F statistic.



4.2 Tests about One Parameter

For testing a hypothesis about only one parameter β_k, with $k = 1, \ldots, K$

$$\begin{aligned} H_0: &\quad \beta_k = r \\ H_A: &\quad \beta_k \neq r \end{aligned}$$

the $(q \times K{=}1 \times K)$-matrix R equals a row-vector of zeros but with a one as the kth element (e.g., for $k = 2$, $R = (0, 1, 0, \ldots, 0)$) such that F in (4.2) simplifies to

$$\frac{\left(\hat{\beta}_k - r\right)^2}{\widehat{\text{Var}}(\hat{\beta}_k|X)} \overset{H_0}{\sim} F(1, n-K),$$

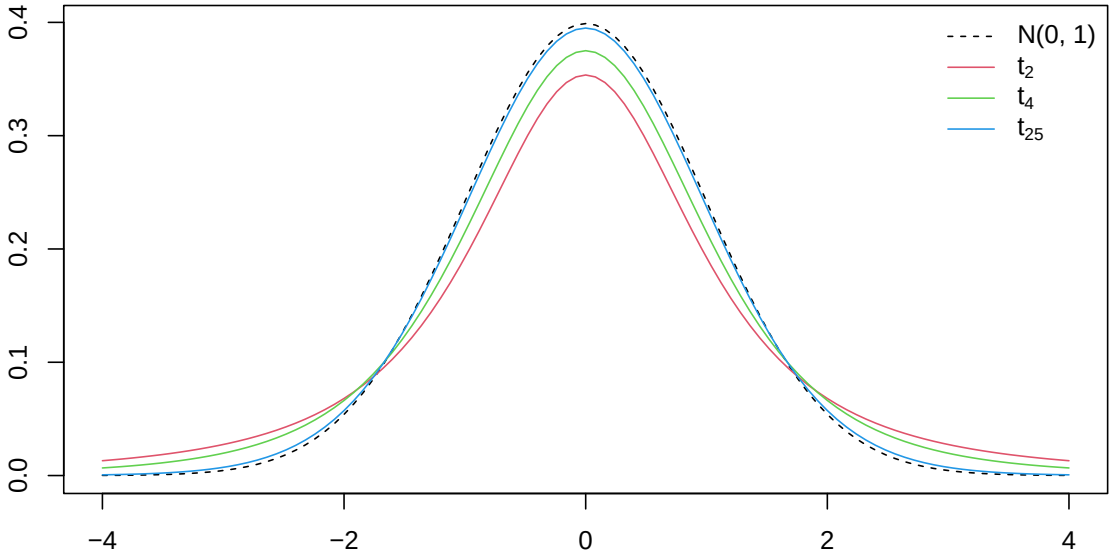
where $\widehat{\text{Var}}(\hat{\beta}_k|X) = R(s^2(X'X)^{-1})R'$. Taking square roots yields

$$t = \frac{\hat{\beta}_k - r}{\widehat{\text{SE}}(\hat{\beta}_k|X)} \overset{H_0}{\sim} t_{(n-K)}.$$

Thus the t-distribution with $n - K$ degrees of freedom is the appropriate reference metric for judging how "far away'' our estimates are from the hypothetical parameter value r under the null hypothesis.

Note. All commonly used statistical software packages report t-tests testing the null hypothesis $H_0 : \beta_k = 0$, i.e., with $r = 0$. This means to test the null hypothesis that X_k has "no (linear) effect" on Y.

The following plot illustrates that as the degrees of freedom increase, the shape of the t distribution comes closer to that of a standard normal bell curve. Already for 25 degrees of freedom we find little difference to the standard normal density. In case of small degrees of freedom values, we find the distribution to have heavier tails than a standard normal.



4.3 Testtheory

4.3.1 Significance Level

To actually test the null hypothesis (e.g., H_0: $R\beta - r = 0$ or $H_0 : \beta_k = 0$), we need to have a decision rule on when we will reject and not reject the null hypothesis. This amounts to deciding on a probability with which we are comfortable rejecting the null hypothesis when it is in fact true (Type I error or α error). The probability of such a Type I error shall be bounded from above by a (small) significance level α, that is

$$P(\text{reject } H_0 | H_0 \text{ is true}) = P(\text{Type I Error}) = \alpha$$

For a given significance level (e.g., $\alpha = 0.05$) and a given alternative hypothesis, we can divide the range of all possible values of the test statistic (i.e., $\mathbb{R}$ since both $t \in \mathbb{R}$ and $F \in \mathbb{R}$) into a **rejection region** and a **non-rejection region** by using certain quantiles called **critical values** of the test statistic distribution under the null. We can do this because the test statistics t and F have known distributions under the null hypothesis ($t \overset{H_0}{\sim} t_{n-K}$ and $F \overset{H_0}{\sim} F(q, n-K)$); indeed, under Assumption 4*, we know the *exact* null distributions for every sample size n. Having decided on the rejection and non-rejection regions, it is a simple matter of seeing where the observed (obs) sample values t_{obs} or F_{obs} of the statistics t or F are–either in the rejection or in the non-rejection region.

Non-conservative versus conservative tests. Since the test statistics F and t are continuous random variables of which we know the *exact* distributions (under Assumptions 1-4*), we can find critical values such that

$$P(\text{Type I Error}) = \alpha$$

We call such tests "non-conservative" since the probability of a type I error equals the significance level α. Test statistics with

$$P(\text{Type I Error}) < \alpha$$

are called *conservative* test statistics; they lead to valid inferences, but will detect a violation of the null hypothesis less often than a non-conservative test. A test statistic with $P(\text{Type I Error}) > \alpha$ leads to *invalid* inferences!

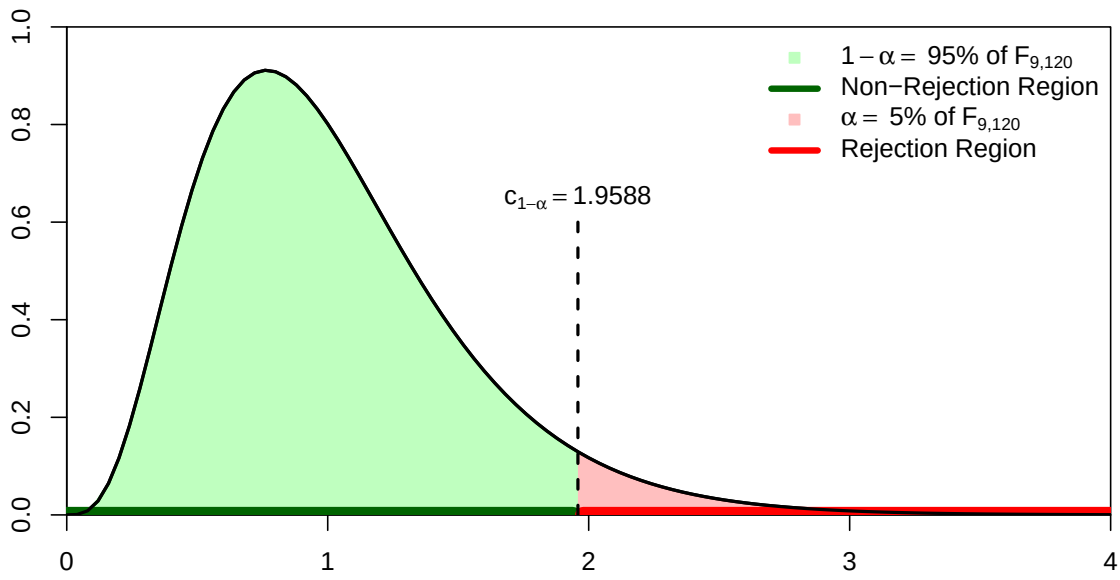
4.3.2 Critical Value for the F-Test

The critical value $c_{1-\alpha} > 0$ defines the rejection region, $]c_{1-\alpha}, \infty[$, and non-rejection region, $]0, c_{1-\alpha}]$ which divide the test-statistic space (here $\mathbb{R}^+$ since

$F \in \mathbb{R}^+$) for a given significance level $\alpha \in (0, 1)$, such that

$$P(\text{Type I Error}) = P_{H_0}\Big(F \in]c_{1-\alpha}, \infty[\Big) = \alpha,$$

where $c_{1-\alpha}$ is here the $(1-\alpha)$ quantile of the F-distribution with $(q, n-K)$ degrees of freedom, and where P_{H_0} means that we compute the probability under the assumption that H_0 is true.



The rejection region. The rejection region describes a range of values of the test statistic F which we rarely see if the null hypothesis is true (only in at most $\alpha \cdot 100\%$ cases). If the observed value of the test statistic, F_{obs}, falls in this region, we will reject the null hypothesis–and hereby, accept Type I errors in at most $\alpha \cdot 100\%$ of cases.

The non-rejection region. The non-rejection region describes a range of values of the test statistic F which we expect to see (in $(1-\alpha)\cdot 100\%$ cases) if the null hypothesis is true. If the observed value of the test statistic, F_{obs} falls in this region, we will not reject the null hypothesis.

Caution: Not rejecting the null hypothesis does not mean that we can conclude that the null hypothesis is true. We only had no sufficiently strong evidence against the null hypothesis. A violation of the null hypothesis, for instance $R\beta - r = a \neq 0$, may simply be too small (too small a value) to stand out from the estimation errors (measured by the standard error) in $\hat{\beta}_k$.

Reading the F-Table. Fortunately, you do not need to read old-school distribution tables to find the critical value $c_{1-\alpha}$, but can simply use R

```
df1   <- 9    # numerator df
df2   <- 120  # denominator df
alpha <- 0.05 # significance level
## Critical value:
crit_value <- qf(p = 1-alpha, df1 = df1, df2 = df2)
crit_value
#> [1] 1.958763
```

Changing the significance level from $\alpha = 0.05$ to $\alpha = 0.01$ makes the critical value $c_{1-\alpha}$ larger and, therefore, the rejection region smaller (fewer Type I errors)

```
alpha <- 0.01
## Critical value:
crit_value <- qf(p = 1-alpha, df1 = df1, df2 = df2)
```

```
crit_value
#> [1] 2.558574
```

4.3.3 Critical Value(s) for the t-Test

In case of the t-test, we need to differentiate between two-sided and one-sided testing.

Two-Sided t-Test

Two-sided hypothesis:

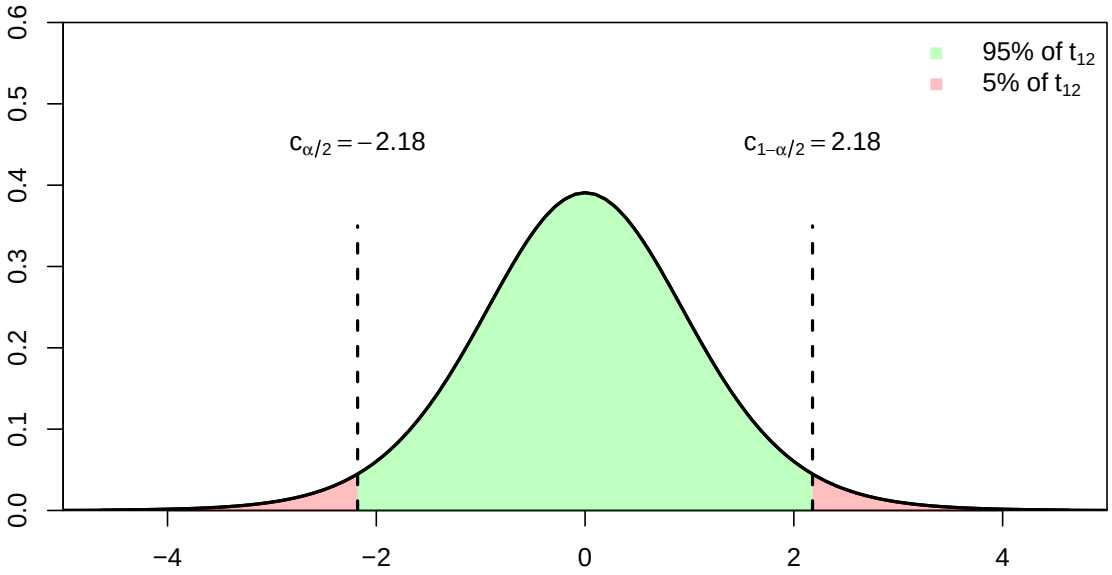
$$\begin{aligned} H_0: &\quad \beta_k = r \\ H_A: &\quad \beta_k \neq r \end{aligned}$$

In case of a two-sided t-test, we reject the null hypothesis if the observed realization of the t-test, t_{obs}, is "far away" from zero either by being sufficiently smaller or greater than r. The corresponding two-sided critical values are denoted by $-c_{1-\alpha/2} = c_{\alpha/2} < 0$ and $c_{1-\alpha/2} > 0$, where $c_{1-\alpha/2} > 0$ is the $(1-\alpha/2)$ quantile of the t-distribution with $(n-K)$ degrees of freedom, and where $-c_{1-\alpha/2} = c_{\alpha/2}$ due to the symmetry of the t-distribution. These critical values defines the following rejection and the non-rejection regions

$$\begin{aligned} \text{rejection region:} &\quad]-\infty, c_{\alpha/2}[\ \cup \ ]c_{1-\alpha/2}, \infty[\\ \text{non-rejection region:} &\quad [c_{\alpha/2}, c_{1-\alpha/2}]. \end{aligned}$$

For this rejection region it holds true that

$$P(\text{Type I Error}) = P_{H_0}\Big(t \in]-\infty, c_{\alpha/2}[\ \cup \ ]c_{1-\alpha/2}, \infty[\Big) = \alpha.$$



One-Sided t-Test

One-sided hypothesis:

$$\begin{aligned} H_0: &\quad \beta_k = r \\ H_A: &\quad \beta_k > r \\ (\text{or} \quad H_A: &\quad \beta_k < r) \end{aligned}$$

In case of a one-sided t-test, we will reject the null if t_{obs} is sufficiently "far away'' from zero in the relevant direction of H_A. The corresponding critical value is either $-c_{1-\alpha}$ $(H_A : \beta_k < r)$ or $c_{1-\alpha}$ $(H_A : \beta_k > r)$, where $c_{1-\alpha}$ is the $(1-\alpha)$ quantile of the t-distribution with $(n-K)$ degrees of freedom, and where $-c_{1-\alpha} = c_\alpha$ due to the symmetry of the t-distribution. The critical value $c_{1-\alpha}$ defines the following rejection and the non-rejection regions:

For $H_0 : \beta_k = 0 \quad$ versus $\quad H_A : \beta_k < 0$:

$$\begin{aligned} \text{rejection region:} \quad &]-\infty, c_\alpha[\\ \text{non-rejection region:} \quad & [c_\alpha, \infty[\end{aligned}$$

such that

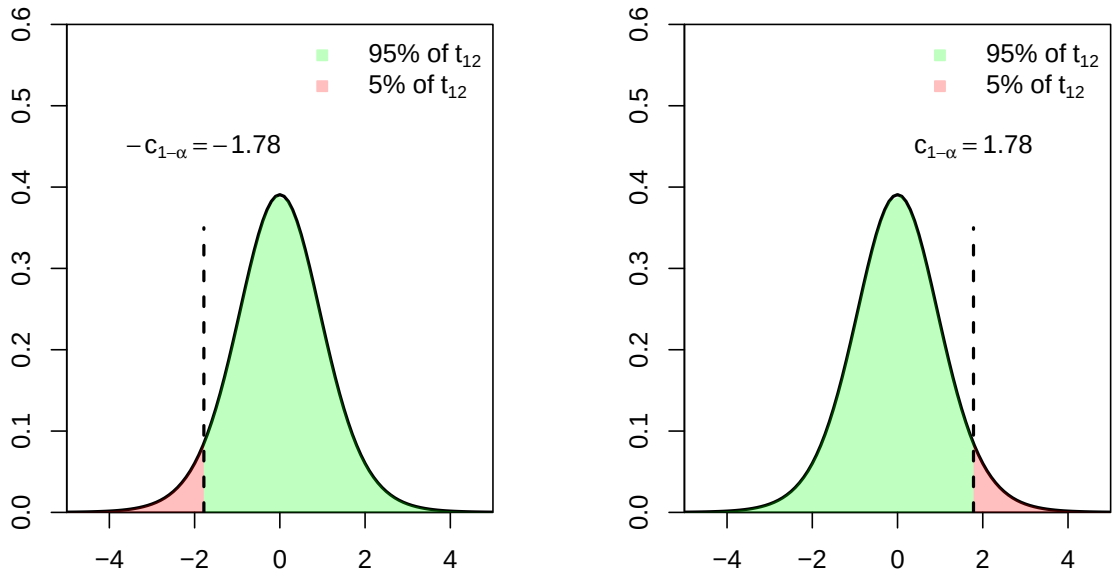
$$P(\text{Type I Error}) = P_{H_0}\Big(t \in]-\infty, c_\alpha[\Big) = \alpha.$$

For $H_0 : \beta_k = 0 \quad$ versus $\quad H_A : \beta_k > 0$:

$$\begin{aligned} \text{rejection region:} \quad &]c_{1-\alpha}, \infty[\\ \text{non-rejection region:} \quad &]-\infty, c_{1-\alpha}] \end{aligned}$$

such that

$$P(\text{Type I Error}) = P_{H_0}\Big(t \in]c_{1-\alpha}, \infty[\Big) = \alpha.$$



Reading the t-Table. Fortunately, you do not need to read old-school distribution tables to find the critical values, but you can simply use R

```
df    <- 16   # degrees of freedom
alpha <- 0.05 # significance level
## One-sided critical value (= (1-alpha) quantile):
c_oneSided <- qt(p = 1-alpha, df = df)
c_oneSided
#> [1] 1.745884
## Two-sided critical value (= (1-alpha/2) quantile):
c_twoSided <- qt(p = 1-alpha/2, df = df)
## lower critical value
-c_twoSided
#> [1] -2.119905
## upper critical value
c_twoSided
#> [1] 2.119905
```

4.4 Type II Error and Power

A Type II error is the mistake of not rejecting the null hypothesis when in fact it should have been rejected. The probability of making a Type II error equals one minus the probability of correctly rejecting the null hypothesis ("Power"). For instance, in the case of using the t-test to test the null hypothesis $H_0 : \beta_k = 0$ versus the one-sided alternative hypothesis

$H_A : \beta_k > 0$) we have that

$$\begin{aligned} P(\text{Type II Error}) &= P_{H_A}\Big(t \in \overbrace{\big]-\infty, c_{1-\alpha}\big]}^{\text{non-rejection region}}\Big) \\ &= 1 - \underbrace{P_{H_A}\Big(t \in \overbrace{\big]c_{1-\alpha}, \infty\big[}^{\text{rejection region}}\Big)}_{\text{"Power"}}, \end{aligned}$$

where P_{H_A} means that we compute the probability under the assumption that H_A is true.

There is a trade off between the probability of making a Type I error and the probability of making a Type II error: a lower significance level α, decreases $P(\text{Type I Error})$, but necessarily increases $P(\text{Type II Error})$ and vice versa. Ideally, we would have some sense of the costs of making each of these errors, and would choose our significance level to minimize these total costs. However, the costs are often difficult to know. Moreover, the probability of making a Type II error is usually impossible to compute, since we usually do not know the true distribution of $\hat{\beta}_k$ under the alternative.

For illustration purposes, however, consider the case of a t test for a one-sided hypothesis

$$\begin{aligned} H_0 : &\quad \beta_k = 0 \\ H_A : &\quad \beta_k > 0, \end{aligned}$$

where the true (usually unknown) parameter value is $\beta_k = 3$ and where the true (usually also unknown) standard error is $\text{SE}(\hat{\beta}_k|X) = \sqrt{\sigma^2[(X'X)^{-1}]_{kk}} = 1.5$. The advantage here is that we can derive the distribution of the t-test statistic even under the alternative hypothesis. Note that the distribution of the t-test statistic becomes here a standard normal distribution, since we assume $\text{SE}(\hat{\beta}_k|X) = \sqrt{\sigma^2[(X'X)^{-1}]_{kk}} = 1.5$ to be a **known** (deterministic) quantity. (This completely unrealistic assumption is only used for illustrative purposes!)

Under this setup, the distribution "under the null hypothesis" (i.e., if $\beta_k = 0$ were true) is:

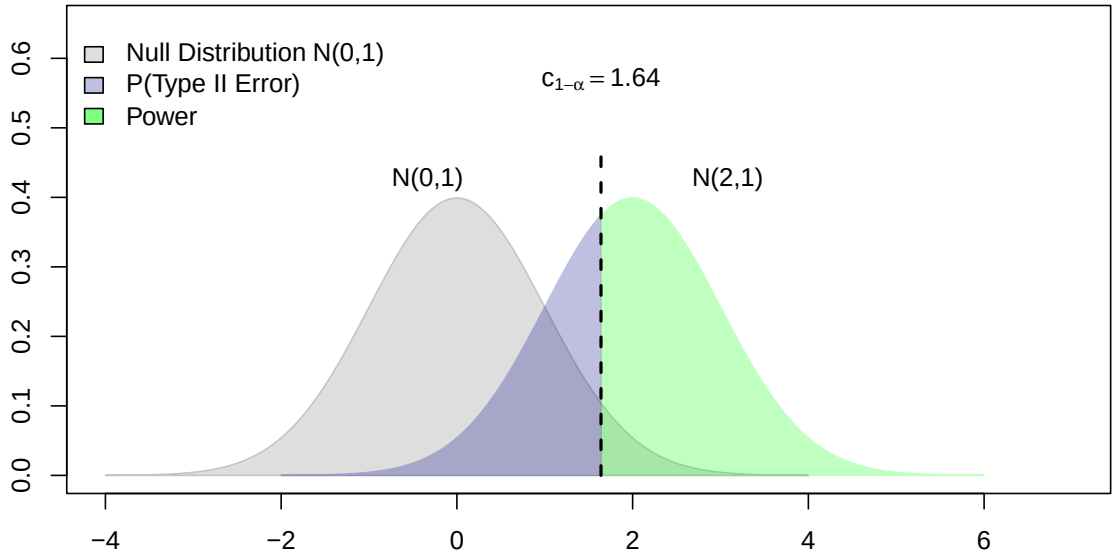
$$t \big| X = \frac{\hat{\beta}_k}{\sqrt{\sigma^2 [(X'X)^{-1}]_{kk}}} \Big| X \overset{H_0}{\sim} \mathcal{N}(0,1)$$

Likewise, the distribution "under the alternative hypothesis" for the true parameter value (i.e., for the actual $\beta_k = 3$) is:

$$t = \frac{\hat{\beta}_k}{\sqrt{\sigma^2 [(X'X)^{-1}]_{kk}}} = \frac{\hat{\beta}_k + 3 - 3}{\sqrt{\sigma^2 [(X'X)^{-1}]_{kk}}}$$

$$\Rightarrow \quad \underbrace{\frac{\hat{\beta}_k - 3}{\sqrt{\sigma^2 [(X'X)^{-1}]_{kk}}}}_{\substack{\sim \mathcal{N}(0,1) \\ \text{(conditionally on } X\text{)}}} + \underbrace{\frac{3}{\sqrt{\sigma^2 [(X'X)^{-1}]_{kk}}}}_{=\Delta \text{ (mean-shift)}} \Big| X \overset{H_A}{\sim} \mathcal{N}(\Delta, 1)$$

So, the mean-shift Δ depends on the value of $\sqrt{\sigma^2 [(X'X)^{-1}]_{kk}}$ and the difference between the actual parameter value ($\beta_k = 3$) and the hypothetical parameter value under the null-hypothesis (here 0). The following Graphic illustrates the probability of a type II error and the power for the case where $\sqrt{\sigma^2 [(X'X)^{-1}]_{kk}} = 1.5$ such that $\Delta = 3/1.5 = 2$.



4.5 p-Value

The p-value of a test statistic is the significance level we would obtain if we took the sample value of the observed test statistic, F_{obs} or t_{obs}, as the border between the rejection and non-rejection regions.

F-test $p = P_{H_0}(F \geq F_{\text{obs}})$

t-test $(H_A : \beta_k \neq r)$ $p = 2 \cdot \min\{P_{H_0}(t \leq t_{\text{obs}}), P_{H_0}(t \geq t_{\text{obs}})\}$

t-test $(H_A : \beta_k > r)$ $p = P_{H_0}(t \geq t_{\text{obs}})$

t-test $(H_A : \beta_k < r)$ $p = P_{H_0}(t \leq t_{\text{obs}})$

Put another way, the p-value is the greatest significance level for which we just fail to reject the null. Therefore, the p-value is sometimes also called the marginal significance value.

If the p-value is strictly smaller than the chosen significance level α, we reject the null hypothesis.

4.6 Confidence Intervals

We define a two-sided $(1-\alpha)\cdot 100\%$ percent confidence interval for the *deterministic* (unknown) true β_k as the **random interval** $\mathrm{CI}_{1-\alpha}$ for which

$$P\Big(\beta_k \in \mathrm{CI}_{1-\alpha}\Big) \geq 1-\alpha.$$

Derivation of the **random interval** $\mathrm{CI}_{1-\alpha}$: Observe that

$$\frac{\hat{\beta}_k - \beta_k}{\widehat{\mathrm{SE}}(\hat{\beta}_k|X)} \sim t_{(n-K)}$$

Therefore,

$$P\left(-t_{1-\alpha/2,n-K} \leq \frac{\hat{\beta}_k - \beta_k}{\widehat{\mathrm{SE}}(\hat{\beta}_k|X)} \leq t_{1-\alpha/2,n-K}\right) = 1-\alpha,$$

where $t_{1-\alpha/2,n-K}$ denotes the $(1-\alpha)$ quantile of the t-distribution with $n-K$ degrees of freedom. Next, we can do the following equivalent transformations

$$P\left(-t_{1-\alpha/2,n-K} \leq \frac{\hat{\beta}_k - \beta_k}{\widehat{\mathrm{SE}}(\hat{\beta}_k|X)} \leq t_{1-\alpha/2,n-K}\right) = 1-\alpha$$

$$\Leftrightarrow P\left(\hat{\beta}_k - t_{1-\alpha/2,n-K}\widehat{\mathrm{SE}}(\hat{\beta}_k|X) \leq \beta_k \leq \hat{\beta}_k + t_{1-\alpha/2,n-K}\widehat{\mathrm{SE}}(\hat{\beta}_k|X)\right) = 1-\alpha$$

$$\Leftrightarrow P\left(\beta_k \in \underbrace{\left[\hat{\beta}_k - t_{1-\alpha/2,n-K}\widehat{\mathrm{SE}}(\hat{\beta}_k|X),\ \hat{\beta}_k + t_{1-\alpha/2,n-K}\widehat{\mathrm{SE}}(\hat{\beta}_k|X)\right]}_{\mathrm{CI}_{1-\alpha}}\right) = 1-\alpha$$

That is, the random interval

$$\text{CI}_{1-\alpha} = \left[\hat{\beta}_k - t_{1-\alpha/2,n-K}\widehat{\text{SE}}(\hat{\beta}_k|X),\ \hat{\beta}_k + t_{1-\alpha/2,n-K}\widehat{\text{SE}}(\hat{\beta}_k|X)\right]$$

is our $(1-\alpha)\cdot 100\%$ percent confidence interval for β_k.

Interpretation. The random interval $\text{CI}_{1-\alpha}$ for β_k contains the true parameter value β_k in $(1-\alpha)\cdot 100\%$ cases of its realizations - when considering infinitely many realizations.[1] Unfortunately, this "interpretation" is not a statement about a single $\text{CI}_{1-\alpha}$ computed for given data. So, the problem with CIs is that they are quite unintuitive. However, they are very well suited for a visual comparison of the estimation uncertainties in different parameter estimators, for instance, across $\hat{\beta}_k$, $k = 1, \ldots, K$.

4.7 Practice: Small Sample Inference

Let's apply the above exact inference methods using R. First, we program a function `myDataGenerator()` which allows us to generate data from the following model, i.e., from the following fully specified data generating process:

$$\begin{aligned}
Y_i &= \beta_1 + \beta_2 X_{i2} + \beta_3 X_{i3} + \varepsilon_i, \qquad i = 1, \ldots, n \\
\beta &= (\beta_1, \beta_2, \beta_3)' = (2, 3, 4)' \\
X_{i2} &\sim U[2, 10] \\
X_{i3} &\sim U[12, 22] \\
\varepsilon_i &\sim \mathcal{N}(0, 3^2),
\end{aligned}$$

where (Y_i, X_i) is assumed i.i.d. across $i = 1, \ldots, n$. Below, in the codes, I use $n = 10$, but, of course, other sample sizes can be considered too. Under

[1] Take a look at this visualization: https://rpsychologist.com/d3/ci/

the assumptions of this chapter, we do exact inference that is specific to any given sample size n.

The below function `myDataGenerator()` allows to sample new realizations of $Y_1, \ldots, Y_n$ conditionally on a given data matrix X. Moreover, you can provide your own values for the sample size n and for the parameter vector $\beta = (\beta_1, \beta_2, \beta_3)'$.

```
## Function to generate artificial data
## If X=NULL: new X variables are generated
## If the user gives X variables,
## the sampling of new Y variables is conditionally on
## the given X variables.
myDataGenerator <- function(n, beta, X=NULL, sigma=3){
  if(is.null(X)){
    X   <- cbind(rep(1, n),
                 runif(n, 2, 10),
                 runif(n,12, 22))
  }
  eps  <- rnorm(n, sd=sigma)
  Y    <- X %*% beta + eps
  data <- data.frame("Y"=Y,
                     "X_1"=X[,1], "X_2"=X[,2], "X_3"=X[,3])
  ##
  return(data)
}

## Define a true beta vector
beta_true <- c(2,3,4)

## Check:
```

```r
## Generate Y and X data
test_data     <- myDataGenerator(n = 10, beta=beta_true)
## Generate new Y data conditionally on X
X_cond <- cbind(test_data$X_1,
                test_data$X_2,
                test_data$X_3)
test_data_new <- myDataGenerator(n    = 10,
                                 beta = beta_true,
                                 X    = X_cond)
## compare
round(head(test_data,     3), 2) # New Y, new X
#>       Y X_1  X_2   X_3
#> 1 89.36   1 4.04 19.62
#> 2 96.87   1 8.19 17.55
#> 3 88.71   1 2.81 20.34
round(head(test_data_new, 3), 2) # New Y, conditionally on X
#>       Y X_1  X_2   X_3
#> 1 96.06   1 4.04 19.62
#> 2 95.72   1 8.19 17.55
#> 3 94.02   1 2.81 20.34
```

4.7.1 Normally Distributed $\hat{\beta}|X$

The above data generating process fulfills our regulatory assumptions Assumption 1-4*. So, by theory, the estimators $\hat{\beta}_k|X$ should be normal distributed conditionally on X

$$\hat{\beta}_k|X \sim \mathcal{N}(\beta_k, \sigma^2[(X'X)^{-1}]_{kk})$$

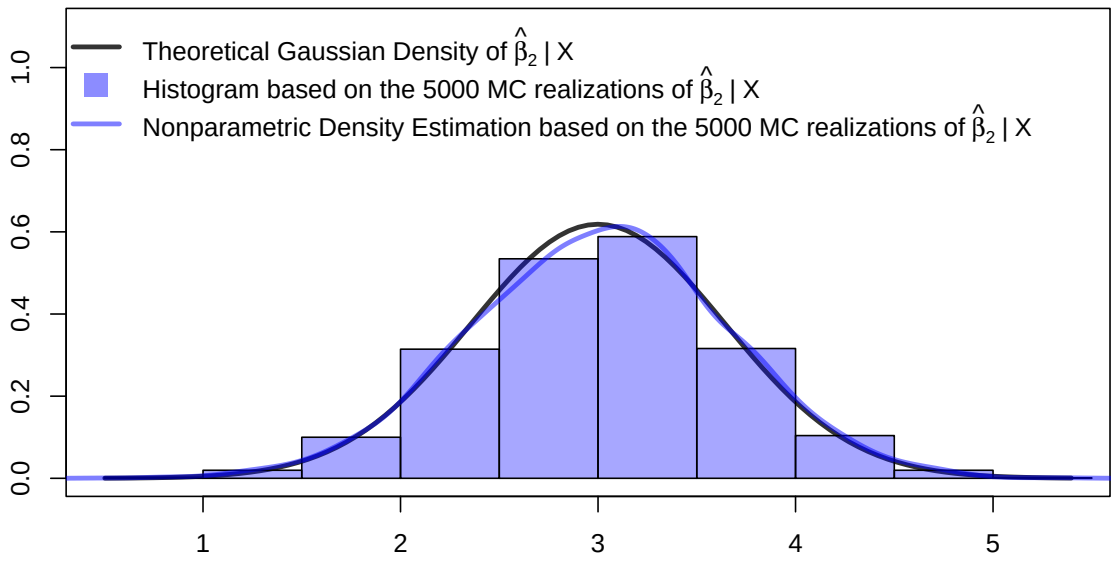
where $[(X'X)^{-1}]_{kk}$ denotes the element in the kth row and kth column of the matrix $(X'X)^{-1}$. Let's check the distribution by means of a Monte Carlo

simulation for the case of $\hat{\beta}_2|X$ with a small sample size of $n = 10$.

```
set.seed(123)
n         <- 10       # a small sample size
beta_true <- c(2,3,4) # true data vector
sigma     <- 3        # true standard deviation of the error term (var

## Let's generate a data set from our data generating process
mydata  <- myDataGenerator(n = n, beta=beta_true)
X_cond  <- cbind(mydata$X_1, mydata$X_2, mydata$X_3)

## True mean and variance of the true normal distribution
## of beta_hat_2|X=X_cond:
# true mean
beta_true_2     <- beta_true[2]
# true variance
var_true_beta_2 <- sigma^2 * diag(solve(t(X_cond) %*% X_cond))[2]

## Let's generate 5000 realizations from beta_hat_2
## conditionally on X=X_cond and check whether the empirical
## distribution of these 5000 realizations is close
## to the true normal distribution of beta_hat_2:
rep        <- 5000 # MC replications
beta_hat_2 <- rep(NA, times=rep)
##
for(r in 1:rep){
    MC_data <- myDataGenerator(n    = n,
                               beta = beta_true,
                               X    = X_cond)
    lm_obj        <- lm(Y ~ X_2 + X_3, data = MC_data)
```

```
    beta_hat_2[r] <- coef(lm_obj)[2]
}

## Compare
## True beta_2 versus average of beta_hat_2 estimates
beta_true_2
#> [1] 3
round(mean(beta_hat_2), 4)
#> [1] 3.0091
## True variance of beta_hat_2 versus
## empirical variance of beta_hat_2 estimates
round(var_true_beta_2, 4)
#> [1] 0.416
round(var(beta_hat_2), 4)
#> [1] 0.4235

## True normal distribution of beta_hat_2 versus
## empirical density of beta_hat_2 estimates
library("scales")
curve(expr = dnorm(x, mean = beta_true_2,
                   sd=sqrt(var_true_beta_2)),
      xlab="",ylab="", col=gray(.2), lwd=3, lty=1,
      xlim=range(beta_hat_2), ylim=c(0,1.1))
hist(beta_hat_2, freq=FALSE, col=alpha("blue",.35), add=TRUE)
lines(density(beta_hat_2, bw = bw.SJ(beta_hat_2)),
      col=alpha("blue",.5), lwd=3)
legend("topleft", lty=c(1,NA,1), lwd=c(3,NA,3), pch=c(NA,15,NA), pt.cex=c(N
     col=c(gray(.2), alpha("blue",.45), alpha("blue",.5)), bty="n", legend=
c(expression(
  "Theoretical Gaussian Density of"~hat(beta)[2]~'|'~X),
```

```
  expression(
  "Histogram based on the 5000 MC realizations of"~
  hat(beta)[2]~'|'~X),
  expression(
  "Nonparametric Density Estimation based on the 5000 MC realizations o
  hat(beta)[2]~'|'~X)))
```

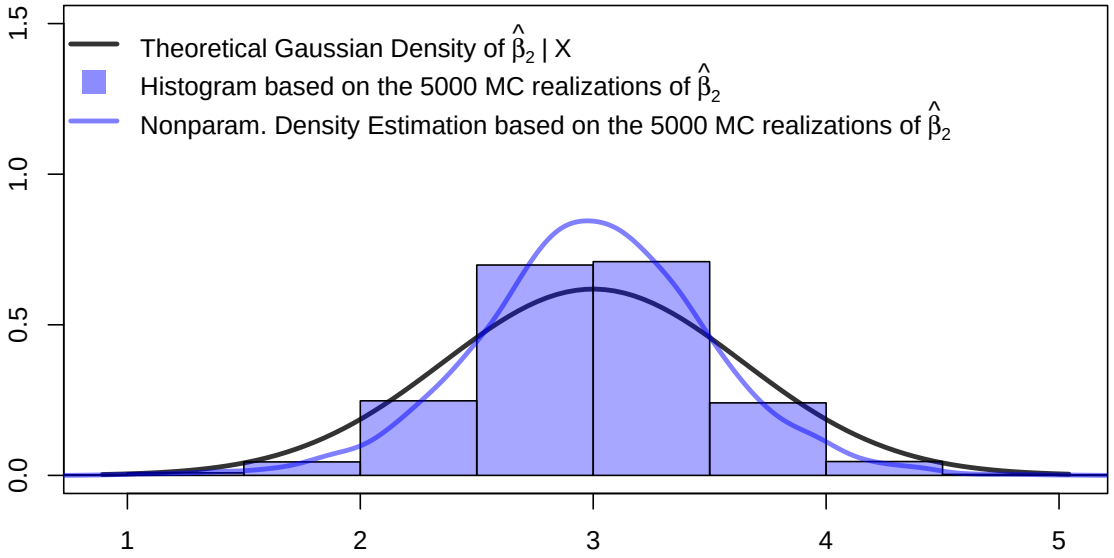


Great! The nonparametric density estimation (estimated via `density()`) and the histogram computed based on the 5000 simulated realizations of $\hat{\beta}_2|X$ are indicating that $\hat{\beta}_2|X$ is really normally distributed as described by our theoretical result in Equation (4.1).

However, what would happen if we would sample *unconditionally* on X? How does the distribution of $\hat{\beta}_2$ would then look like?

```
set.seed(123)
## Let's generate 5000 realizations from beta_hat_2
## WITHOUT conditioning on X
beta_hat_2_uncond <- rep(NA, times=rep)
##
for(r in 1:rep){
    MC_data <- myDataGenerator(n    = n,
                               beta = beta_true)
    lm_obj                <- lm(Y ~ X_2 + X_3, data = MC_data)
    beta_hat_2_uncond[r] <- coef(lm_obj)[2]
}

## Compare
## True beta_2 versus average of beta_hat_2 estimates
beta_true_2
#> [1] 3
round(mean(beta_hat_2_uncond), 4)
#> [1] 2.9973
## True variance of beta_hat_2 versus
## empirical variance of beta_hat_2 estimates
round(var_true_beta_2, 4)
#> [1] 0.416
round(var(beta_hat_2_uncond), 4)
#> [1] 0.2521

## Plot
curve(expr = dnorm(x, mean = beta_true_2,
```

```
                  sd=sqrt(var_true_beta_2)),
      xlab="",ylab="", col=gray(.2), lwd=3, lty=1,
      xlim=range(beta_hat_2_uncond), ylim=c(0,1.5))
hist(beta_hat_2_uncond, freq=FALSE, col=alpha("blue",.35), add=TRUE)
lines(density(beta_hat_2_uncond, bw=bw.SJ(beta_hat_2_uncond)),
      col=alpha("blue",.5), lwd=3)
legend("topleft", lty=c(1,NA,1), lwd=c(3,NA,3), pch=c(NA,15,NA), pt.cex
     col=c(gray(.2), alpha("blue",.45), alpha("blue",.5)), bty="n", leg
c(expression(
  "Theoretical Gaussian Density of"~hat(beta)[2]~'|'~X),
expression(
  "Histogram based on the 5000 MC realizations of"~
  hat(beta)[2]),
expression("Nonparam. Density Estimation based on the 5000 MC realizati
  hat(beta)[2])))
```



Not so good. Since we do not condition on X, the realizations of X affect the distribution of $\hat{\beta}$ and our theoretical Gaussian distribution result in Equation (4.1) does not apply anymore.

4.7.2 Testing Multiple Parameters

In the following, we do inference about multiple parameters. We test

$$\begin{aligned} & H_0 : \beta_2 = 3 \quad \text{and} \quad \beta_3 = 4 \\ \text{versus} \quad & H_A : \beta_2 \neq 3 \quad \text{and/or} \quad \beta_3 \neq 4. \end{aligned}$$

Or equivalently

$$H_0 : R\beta - r = 0$$
$$H_A : R\beta - r \neq 0,$$

where

$$R = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad r = \begin{pmatrix} 3 \\ 4 \end{pmatrix}.$$

The following R code can be used to test this hypothesis:

```
suppressMessages(library("car")) # for linearHyothesis()
# ?linearHypothesis

## Estimate the linear regression model parameters
lm_obj <- lm(Y ~ X_2 + X_3, data = mydata)

## Option 1:
car::linearHypothesis(model = lm_obj,
                      hypothesis.matrix = c("X_2=3", "X_3=4"))
#> Linear hypothesis test
#>
#> Hypothesis:
#> X_2 = 3
#> X_3 = 4
#>
#> Model 1: restricted model
#> Model 2: Y ~ X_2 + X_3
#>
#>   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
#> 1      9 87.285
#> 2      7 37.599  2    49.686 4.6252 0.05246 .
```

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

## Option 2:
R <- rbind(c(0,1,0),
           c(0,0,1))
car::linearHypothesis(model = lm_obj,
                      hypothesis.matrix = R,
                      rhs = c(3,4))
#> Linear hypothesis test
#>
#> Hypothesis:
#> X_2 = 3
#> X_3 = 4
#>
#> Model 1: restricted model
#> Model 2: Y ~ X_2 + X_3
#>
#>   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
#> 1      9 87.285
#> 2      7 37.599  2    49.686 4.6252 0.05246 .
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Not surprisingly, we cannot reject the null hypothesis at a significance level of, for instance, $\alpha = 0.05$ since we actually test the true null hypothesis. However, in repeated samples we should nevertheless observe $\alpha \cdot 100\%$ type I errors (false rejections of H_0). Let's check this using the following Monte Carlo simulation:

```
## Let's generate 5000 F-test decisions and check
## whether the empirical rate of type I errors is
## close to the theoretical significance level.
rep             <- 5000 # MC replications
F_test_pvalues  <- rep(NA, times=rep)
##
for(r in 1:rep){
  ## generate new MC_data conditionally on X_cond
    MC_data <- myDataGenerator(n    = n,
                               beta = beta_true,
                               X    = X_cond)
    lm_obj            <- lm(Y ~ X_2 + X_3, data = MC_data)
    ## save the p-value
    p <- linearHypothesis(lm_obj,
                          c("X_2=3", "X_3=4"))$`Pr(>F)`[2]
    F_test_pvalues[r] <- p
}
##
signif_level <-  0.05
rejections   <- F_test_pvalues[F_test_pvalues < signif_level]
round(length(rejections)/rep, 3)
#> [1] 0.05
##
signif_level <-  0.01
rejections   <- F_test_pvalues[F_test_pvalues < signif_level]
round(length(rejections)/rep, 3)
#> [1] 0.009
```

Note that this is actually a very strong result. First, it means that we correctly control for the type I error rate since the type I error rate is not

larger than the chosen significance level α. Second, it means that the test is not conservative (i.e. very efficient) since the empirical type I error rate is really close to the chosen significance level α. (In fact, if we would increase the number of Monte Carlo repetitions, the empirical type I error rate would converge to the selected significance level α due to the law of large numbers.)

Next, we check how well the F test detects certain violations of the null hypothesis. We do this by using the same data generating process, but by testing the following incorrect null hypothesis:

$$\begin{aligned} H_0 &: \beta_2 = 4 \quad \text{and} \quad \beta_3 = 4 \\ H_A &: \beta_2 \neq 4 \quad \text{and/or} \quad \beta_3 \neq 4 \end{aligned}$$

```
set.seed(321)
rep             <- 5000 # MC replications
F_test_pvalues  <- rep(NA, times=rep)
##
for(r in 1:rep){
  ## generate new MC_data conditionally on X_cond
    MC_data <- myDataGenerator(n    = n,
                               beta = beta_true,
                               X    = X_cond)
    lm_obj            <- lm(Y ~ X_2 + X_3, data = MC_data)
    ## save p-values of all rep-many tests
    F_test_pvalues[r] <- linearHypothesis(lm_obj,
                           c("X_2=4","X_3=4"))$`Pr(>F)`[2]
}
##
signif_level <-  0.05
rejections   <- F_test_pvalues[F_test_pvalues < signif_level]
length(rejections)/rep
#> [1] 0.3924
```

Indeed, we can now reject the (false) null hypothesis in approximately 39% of all resamplings from the true data generating process. **Caution:** This also means that we are not able to "see" the violation of the null hypothesis in $100\% - 39\% = 62\%$ of cases. Therefore, we can never use an insignificant test result (p-value$\geq \alpha$) as a justification to accept the null hypothesis.

Moreover, note that the F test is not informative about which part of the null hypothesis ($\beta_2 = 4$ and/or $\beta_3 = 4$) is violated – we only get the information that at least one of the multiple parameter hypotheses is violated:

```
car::linearHypothesis(lm_obj, c("X_2=4", "X_3=4"))
#> Linear hypothesis test
#>
#> Hypothesis:
#> X_2 = 4
#> X_3 = 4
#>
#> Model 1: restricted model
#> Model 2: Y ~ X_2 + X_3
#>
#>   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
#> 1      9 141.245
#> 2      7  32.572  2    108.67 11.677 0.005889 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4.7.3 Dualty of Confidence Intervals and Hypothesis Tests

Confidence intervals can be computed using R as following:

```
signif_level <- 0.05
## 95% CI for beta_2
confint(lm_obj, parm = "X_2", level = 1 - signif_level)
#>        2.5 %   97.5 %
#> X_2 1.370315 3.563536
## 95% CI for beta_3
confint(lm_obj, parm = "X_3", level = 1 - signif_level)
#>        2.5 %   97.5 %
#> X_3 3.195389 4.695134
```

We can use these two-sided confidence intervals to do hypothesis tests. For instance, when testing the null hypothesis

$$\begin{aligned} & H_0 : \beta_3 = 4 \\ \text{versus} \quad & H_A : \beta_3 \neq 4 \end{aligned}$$

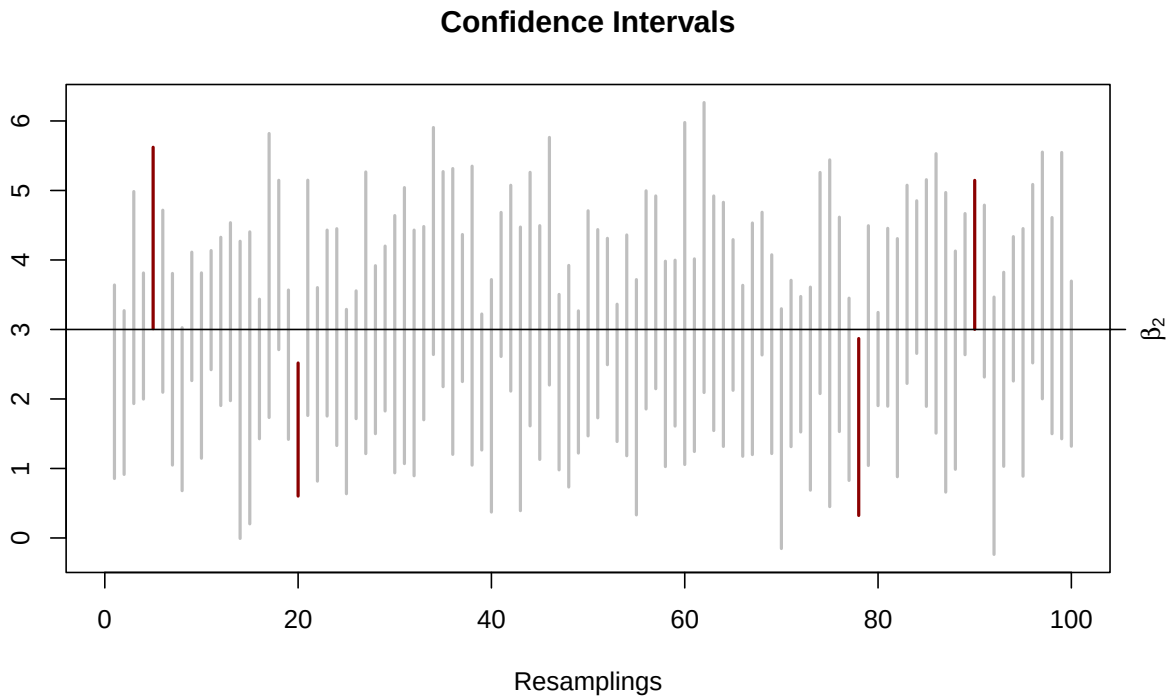
we can check whether the confidence interval $\text{CI}_{1-\alpha}$ contains the hypothetical value 4 or not. In case of $4 \in \text{CI}_{1-\alpha}$, we cannot reject the null hypothesis. In case of $4 \notin \text{CI}_{1-\alpha}$, we reject the null hypothesis.

If the Assumption 1-4* hold true, then $\text{CI}_{1-\alpha}$ is an exact confidence interval. That is, under the null hypothesis, it falsely rejects the null hypothesis in only $\alpha \cdot 100\%$ of resamplings. Let's check this in the following R code:

```
## Let's generate 1000 CIs
set.seed(123)
signif_level <-  0.05
```

```
rep          <- 5000 # MC replications
confint_m    <- matrix(NA, nrow=2, ncol=rep)
##
for(r in 1:rep){
  ## generate new MC_data conditionally on X_cond
    MC_data <- myDataGenerator(n    = n,
                               beta = beta_true,
                               X    = X_cond)
    lm_obj             <- lm(Y ~ X_2 + X_3, data = MC_data)
    ## save the p-value
    CI <- confint(lm_obj, parm="X_2", level=1-signif_level)
    confint_m[,r] <- CI
}
##
inside_CI  <- confint_m[1,] <= beta_true_2 &
                beta_true_2 <= confint_m[2,]

## CI-lower, CI-upper, beta_true_2 inside?
head(cbind(t(confint_m), inside_CI))
#>                           inside_CI
#> [1,] 0.8555396 3.639738         1
#> [2,] 0.9143542 3.270731         1
#> [3,] 1.9336526 4.984167         1
#> [4,] 1.9985874 3.812695         1
#> [5,] 3.0108642 5.621791         0
#> [6,] 2.0967675 4.716398         1

round(length(inside_CI[inside_CI == FALSE])/rep, 2)
#> [1] 0.05
```

```
nCIs <- 100
plot(x=0,y=0,type="n",xlim=c(0,nCIs),ylim=range(confint_m[,1:nCIs]),
     ylab="", xlab="Resamplings", main="Confidence Intervals")
for(r in 1:nCIs){
  if(inside_CI[r]==TRUE){
      lines(x=c(r,r), y=c(confint_m[1,r], confint_m[2,r]),
            lwd=2, col=gray(.5,.5))
  }else{
      lines(x=c(r,r), y=c(confint_m[1,r], confint_m[2,r]),
            lwd=2, col="darkred")
    }
}
axis(4, at=beta_true_2, labels = expression(beta[2]))
abline(h=beta_true_2)
```


Confidence Intervals
0
1
2
3
4
5
6
0
20
40
60
80
100
Resamplings
β2

Chapter 5

Large Sample Inference

5.1 Tools for Asymptotic Statistics

5.1.1 Modes of Convergence

In the following we will discuss the four most important convergence concepts for sequences of random variables $(z_1, z_2, \ldots, z_n)$ shortly denoted by $\{z_n\}$. Non-random scalars (or vectors or matrices) will be denoted by Greek letters such as α.

Four Important Modes of Convergence

1. Convergence in Probability: A sequence of random scalars $\{z_n\}$ **converges in probability** to a constant (non-random) α if, for any (arbitrarily small) $\varepsilon > 0$,

$$\lim_{n\to\infty} P\left(|z_n - \alpha| > \varepsilon\right) = 0.$$

We write: $\text{plim}_{n\to\infty} z_n = \alpha$, or $z_n \xrightarrow{\text{p}} \alpha$. Convergence in probability of a sequence of random vectors (or matrices) $\{z_n\}$ to a constant vector (or matrix) α requires *element-wise* convergence in probability.

2. Almost Sure Convergence: A sequence of random scalars $\{z_n\}$ **converges almost surely** to a constant (non-random) α if

$$P\left(\lim_{n\to\infty} z_n = \alpha\right) = 1.$$

We write: $z_n \xrightarrow{\text{a.s.}} \alpha$. Almost sure convergence of a sequence of random vectors (or matrices) $\{z_n\}$ to a constant vector (or matrix) α requires *element-wise* almost sure convergence.

Note. Almost sure convergence is (usually) rather hard to derive, since the probability is about an event concerning an infinite sequence. Fortunately, however, there are established strong laws of large numbers that we can use for showing almost sure convergence.

3. Convergence in Mean Square: A sequence of random scalars $\{z_n\}$ **converges in mean square** (or **in quadratic mean**) to a constant (non-random) α if

$$\lim_{n\to\infty} E\left((z_n - \alpha)^2\right) = 0$$

We write: $z_n \xrightarrow{\text{m.s.}} \alpha$. If z_n is an estimator (e.g., $z_n = \hat{\beta}_{k,n}$) the expression $E\left((z_n - \alpha)^2\right)$ is termed the **mean squared error**: $\text{MSE}(z_n) = E\left((z_n - \alpha)^2\right)$. Mean square convergence of a sequence of random vectors (or matrices) $\{z_n\}$ to a deterministic vector (or matrix) α requires *element-wise* mean square convergence.

4. Convergence in Distribution: Let F_n be the cumulative distribution function (cdf) of z_n and F the cdf of z. A sequence of random scalars $\{z_n\}$ **converges in distribution** to a random scalar z if for all t such that $F(t)$ is continuous at t,

$$\lim_{n\to\infty} F_n(t) = F(t).$$

We write: $z_n \xrightarrow{\text{d}} z$ and call F the **asymptotic** (or **limit**) **distribution** of z_n. Sometimes you will see statements like $z_n \xrightarrow{\text{d}} N(0,1)$ or $z_n \overset{a}{\sim} N(0,1)$, which should be read as $z_n \xrightarrow{\text{d}} z$, where $z \sim N(0,1)$.

Note. A stochastic sequence $\{z_n\}$ can also convergence in distribution to a **deterministic scalar** α. In this case α is treated as a degenerated random variable with cdf

$$F_\alpha(t) = \begin{cases} 0 & \text{if } t < \alpha \\ 1 & \text{if } t \geq \alpha \end{cases}$$

4′. Multivariate Convergence in Distribution: Let $z_n, z \in \mathbb{R}^K$ be K-dimensional random variables, then

$$z_n \xrightarrow{\text{d}} z \quad \text{if and only if} \quad \lambda' z_n \xrightarrow{\text{d}} \lambda' z$$

for any $\lambda \in \mathbb{R}^K$. This statement is known as the **Cramér Wold Device**. It is needed since element-wise convergence in distribution does generally not imply convergence of the *joint* distribution of z_n to the *joint* distribution of z; except, if all elements are independent from each other.

Relations among Modes of Convergence

Lemma 5.1.1. ***Relationship among the four modes of convergence:***

(i) $z_n \xrightarrow{\text{m.s.}} \alpha \Rightarrow z_n \xrightarrow{\text{p}} \alpha$.

(ii) $z_n \xrightarrow{\text{a.s.}} \alpha \Rightarrow z_n \xrightarrow{\text{p}} \alpha$.

(iii) $z_n \xrightarrow{\text{d}} \alpha \Leftrightarrow z_n \xrightarrow{\text{p}} \alpha$. *I.e., if the limiting random variable is a constant (i.e., a degenerated random variable), then convergence in distribution is equivalent to convergence in probability.*

Proofs can be found, e.g., here: https://www.statlect.com/asymptotic-theory/relations-among-modes-of-convergence

5.1.2 Continuous Mapping Theorem (CMT)

Lemma 5.1.2. ***Preservation of convergence for continuous transformations (or "continuous mapping theorem (CMT)"):*** *Suppose* $\{z_n\}$ *is a stochastic sequence of random scalars, vectors, or matrices and that* $a(\cdot)$ *is a continuous function that does not depended on* n*. Then*

(i) $z_n \xrightarrow{\text{p}} \alpha \Rightarrow a(z_n) \xrightarrow{\text{p}} a(\alpha)$

(ii) $z_n \xrightarrow{\text{a.s.}} \alpha \Rightarrow a(z_n) \xrightarrow{\text{a.s.}} a(\alpha)$

(iii) $z_n \xrightarrow{\text{d}} \alpha \Rightarrow a(z_n) \xrightarrow{\text{d}} a(\alpha)$

Proof can be found, e.g., in *Asymptotic Statistics*, van der Vaart (1998), Theorem 2.3. Or here: https://www.statlect.com/asymptotic-theory/continuous-mapping-theorem

Note. The CMT does *not* hold for m.s.-convergence except for the case where $a(.)$ is linear.

Examples. As a consequence of the CMT (Lemma 5.1.2) we have that the usual arithmetic operations preserve convergence in probability (equivalently for almost sure convergence and convergence in distribution):

$$\text{If } x_n \xrightarrow{\text{p}} \beta \quad \text{and} \quad y_n \xrightarrow{\text{p}} \gamma \quad \text{then} \quad x_n + y_n \xrightarrow{\text{p}} \beta + \gamma$$

$$\text{If } x_n \xrightarrow{\text{p}} \beta \quad \text{and} \quad y_n \xrightarrow{\text{p}} \gamma \quad \text{then} \quad x_n \cdot y_n \xrightarrow{\text{p}} \beta \cdot \gamma$$

If $x_n \xrightarrow{\text{p}} \beta$ and $y_n \xrightarrow{\text{p}} \gamma$ then $x_n/y_n \xrightarrow{\text{p}} \beta/\gamma$, provided that $\gamma \neq 0$

If $X_n' X_n \xrightarrow{\text{p}} \Sigma_{X'X}$ then $(X_n' X_n)^{-1} \xrightarrow{\text{p}} \Sigma_{X'X}^{-1}$, provided $\Sigma_{X'X}$ is a non-singular matrix.

5.1.3 Slutsky Theorem

The following results are concerned with combinations of convergence in probability and convergence in distribution. These are particularly important for the derivation of the asymptotic distribution of estimators.

Lemma 5.1.3 (Slutsky Theorem)**.** *Let x_n and y_n denote sequences of random scalars or vectors and let A_n denote a sequences of random matrices. Moreover, α and A are deterministic limits of appropriate dimensions and x is a random limit of appropriate dimension.*

(i) If $x_n \xrightarrow{\text{d}} x$ and $y_n \xrightarrow{\text{p}} \alpha$ then $x_n + y_n \xrightarrow{\text{d}} x + \alpha$

(ii) If $x_n \xrightarrow{\text{d}} x$ and $y_n \xrightarrow{\text{p}} 0$ then $x_n' y_n \xrightarrow{\text{p}} 0$

(iii) If $x_n \xrightarrow{\text{d}} x$ and $A_n \xrightarrow{\text{p}} A$ then $A_n x_n \xrightarrow{\text{d}} Ax$, where it is assumed that A_n and x_n are "conformable" (i.e., the matrix- and vector-dimensions fit to each other).

Important special case:
If $x_n \xrightarrow{\text{d}} N(0, \Sigma)$ and $A_n \xrightarrow{\text{p}} A$ then $A_n x_n \xrightarrow{\text{d}} N(0, A\Sigma A')$

Proofs can be found, e.g., in *Asymptotic Statistics*, van der Vaart, Theorem 2.8. Or here: https://www.statlect.com/asymptotic-theory/Slutsky-theorem

Note. Sometimes, only parts *(i)* and *(ii)* of Lemma 5.1.3 are called "Slutsky's theorem."

5.1.4 Law of Large Numbers (LLN) and Central Limit Theorem (CLT)

So far, we discussed the definitions of the four most important convergence modes, their relations among each other, and basic theorems (CMT and Slutsky) about functionals of stochastic sequences. Though, we still lack of tools that allow us to actually show that a stochastic sequence convergences (in some of the discussed modes) to some limit.

In the following we consider the stochastic sequences $\{\bar{z}_n\}$ of sample means $\bar{z}_n = n^{-1}\sum_{i=1}^n z_i$, where z_i, $i = 1, \ldots, n$, are (scalar, vector, or matrix-valued) *random variables*. Remember: the sample mean $\bar{z}_n$ is an estimator of the deterministic population mean μ.

Weak LLNs, strong LLNs, and CLTs tell us conditions under which arithmetic means $\bar{z}_n = n^{-1}\sum_{i=1}^n z_i$ converge:

$$\begin{aligned} \bar{z}_n &\xrightarrow{\text{p}} \mu \quad \text{(weak LLN)} \\ \bar{z}_n &\xrightarrow{\text{a.s.}} \mu \quad \text{(strong LLN)} \\ \sqrt{n}(\bar{z}_n - \mu) &\xrightarrow{\text{d}} N(0, \sigma^2) \quad \text{(CLT)} \end{aligned}$$

In the following we introduce the most well-known versions of the weak, the strong LLN, and the CLT.

Theorem 5.1.4 (Weak LLN (Chebychev))**.**

$$\textit{If} \quad \lim_{n\to\infty} E(\bar{z}_n) = \mu \quad \textit{and} \quad \lim_{n\to\infty} \text{Var}(\bar{z}_n) = 0 \quad \textit{then} \quad \bar{z}_n \xrightarrow{\text{p}} \mu$$

Proof can be found, for instance, here: https://www.statlect.com/asymptotic-theory/law-of-large-numbers

Theorem 5.1.5 (Strong LLN (Kolmogorov))**.**

$$\textit{If } \{z_i\} \textit{ is an iid sequence} \quad \textit{and} \quad E(z_i) = \mu \quad \textit{then} \quad \bar{z}_n \xrightarrow{\text{a.s.}} \mu$$

Proof can be found, e.g., in *Linear Statistical Inference and Its Applications*, Rao (1973), pp. 112-114.

Note. The weak and the strong LLN for random vectors follow from requiring element-by-element convergence.

Theorem 5.1.6 (CLT (Lindeberg-Levy))**.**

$$\textit{If } \{z_i\} \textit{ is an iid sequence} \quad \textit{and} \quad E(z_i) = \mu \quad \textit{and} \quad \text{Var}(z_i) = \sigma^2 \quad \textit{then}$$

$$\sqrt{n}(\bar{z}_n - \mu) \xrightarrow{\text{d}} N(0, \sigma^2) \quad \textit{as} \quad n \to \infty$$

Proof can be found, e.g., in *Asymptotic Statistics*, van der Vaart (1998), Theorem 2.17.

The Lindeberg-Levy CLT for K-dimensional random vectors follows from our above discussion on "Multivariate Convergence in Distribution." From this we know that if $\bar{z}_n \in \mathbb{R}^K$ and $\mu \in \mathbb{R}^K$, then

$$\sqrt{n}(\bar{z}_n - \mu) \xrightarrow{\text{d}} \mathcal{N}(0, \Sigma) \quad \Leftrightarrow \quad \sqrt{n}(\lambda'\bar{z}_n - \lambda'\mu) \xrightarrow{\text{d}} \mathcal{N}(0, \lambda'\Sigma\lambda),$$

for any $\lambda \in \mathbb{R}^K$.

That is, to apply the Lindeberg-Levy CLT (Theorem 5.1.6) to multivariate (e.g., K-dimensional) stochastic sequences, we need to check whether the univariate stochastic sequence $\{\lambda' z_i\}$ is i.i.d. with $E(\lambda' z_i) = \lambda'\mu$ and $\text{Var}(\lambda' z_i) = \lambda'\Sigma\lambda$ for any $\lambda \in \mathbb{R}^K$. This is the case if the multivariate (K-dimensional) stochastic sequence $\{z_i\}$ is an i.i.d. sequence with $E(z_i) = \mu$ and $\text{Var}(z_i) = \Sigma$.

Note. The LLNs and the CLT are stated with respect to sequences of sample means $\{\bar{z}_n\}$; i.e., the simplest estimators you probably can think of. We will see, however, that this is all we need in order to analyze also more complicated estimators such as the OLS estimator.

5.1.5 Estimators as a Sequences of Random Variables

Our concepts above readily apply to general scalar-valued (univariate) or vector-valued (K-dimensional) estimators, say $\hat{\theta}_n \in \mathbb{R}^K$, that are computed from i.i.d. random samples.

(Weak) Consistency: We say that an estimator $\hat{\theta}_n$ is *(weakly) consistent for* θ if

$$\hat{\theta}_n \xrightarrow{\text{p}} \theta \quad \text{as} \quad n \to \infty$$

Asymptotic Bias: The *asymptotic bias* of an estimator $\hat{\theta}_n$ of some true parameter θ is defined as:

$$\text{ABias}(\hat{\theta}_n) = \lim_{n\to\infty} E(\hat{\theta}_n) - \theta$$

If $\text{ABias}(\hat{\theta}_n) = 0$, then $\hat{\theta}$ is called an **asymptotically unbiased**.

Asymptotic Normality: A consistent estimator $\hat{\theta}_n$ is *asymptotically normal distributed* if

$$\sqrt{n}(\hat{\theta}_n - \theta) \xrightarrow{\text{d}} \mathcal{N}(0, \Sigma) \quad \text{as} \quad n \to \infty$$

where $\lim_{n\to\infty} \text{Var}(\sqrt{n}(\hat{\theta}_n - \theta)) = \lim_{n\to\infty} \text{Var}(\sqrt{n}\hat{\theta}_n) = \Sigma$ as called the asymptotic variance of $\sqrt{n}(\hat{\theta}_n - \theta)$.

$\sqrt{n}$-consistent: Consistent estimators $\hat{\theta}_n \xrightarrow{\text{p}} \theta$ are called *$\sqrt{n}$-consistent* if

$$\sqrt{n}(\hat{\theta}_n - \theta) \xrightarrow{\text{d}} z \quad \text{as} \quad n \to \infty$$

If additionally the random vector z is normal distributed, then $\hat{\theta}_n$ is often called *consistent and asymptotically normal.*

5.2 Asymptotics under the Classic Regression Model

Given the above introduced machinery on asymptotic concepts and results, we can now proof that the OLS estimators $\hat{\beta}$ and s^2_{UB} applied to the classic regression model (defined by Assumptions 1-4 in Chapter 3) are consistent and asymptotically normal distributed estimators as $n \to \infty$. That is, we can drop the unrealistic normality and spherical errors assumption (Assumption 4*), but still use the usual test statistics (t-test, F-test); as long as the sample size n is "large."

However, before we can formally state the asymptotic properties, we first need to adjust our "data generating process" assumption (Assumption 1) such that we can apply Kolmogorov's strong LLN and Lindeberg-Levy's CLT. Second, we need to adjust the rank assumption (Assumption 3), such that the full column rank of X is guaranteed for the limiting case as $n \to \infty$, too. Assumptions 2 and 4 from Chapter 3 are assumed to hold.

Assumption 1*: Data Generating Process (for Asymptotics) Assumption 1 of Chapter 3 applies, but *additionally* we assume that $(\varepsilon_i, X_i) \in \mathbb{R}^{K+1}$ (or equivalently $(Y_i, X_i) \in \mathbb{R}^{K+1}$) is jointly i.i.d. for all $i = 1, \ldots, n$, with existing and finite second moments for X_i and fourth moments for ε_i.

Note 1. The fourth moment of ε_i is actually only needed for Theorem 5.2.4; for the rest two moments are sufficient.

Note 2. The above adjustment of Assumption 1 is far less restrictive than assuming that the error-terms ε_i are i.i.d. normally distributed and independent from X_i (as it's necessary for small sample inference in Chapter 4).

Assumption 3*: Rank Condition (for Asymptotics) The $(K \times K)$ limiting matrix

$$\Sigma_{X'X} := E(S_{X'X}) = E(n^{-1}X'X) = E(X_i X_i')$$

has full rank K. I.e., $\Sigma_{X'X}$ is nonsingular and invertible.

Theorem 5.2.1 (Consistency of $S_{X'X}^{-1}$)**.** *Under Assumption 1* and 3* we have that*

$$\left(\frac{1}{n}\mathrm{X}'\mathrm{X}\right)^{-1} = S_{X'X}^{-1} \quad \xrightarrow{\mathrm{p}} \quad \Sigma_{X'X}^{-1} \quad as \quad n \to \infty$$

Proof is done in the lecture.

Theorem 5.2.2 (Consistency of $\hat{\beta}$)**.** *Under Assumption 1*, 2 and 3* we have that*

$$\hat{\beta}_n \xrightarrow{\mathrm{p}} \beta \quad as \quad n \to \infty$$

Proof is done in the lecture.

Furthermore, we can show that the appropriately scaled (by $\sqrt{n}$) sampling error $\hat{\beta} - \beta$ of the OLS estimator is asymptotically normal distributed.

Theorem 5.2.3 (Sampling error limiting normality (the classic case))**.** *For simplicity, we consider here the special case of spherical error (*$\mathrm{Var}(\varepsilon\,|X) = \sigma^2 I_n$*). Under Assumption 1*, 2 and 3* we then have that*

$$\sqrt{n}(\hat{\beta}_n - \beta) \xrightarrow{\mathrm{d}} \mathcal{N}\left(0, \sigma^2 \Sigma_{X'X}^{-1}\right) \quad as \quad n \to \infty$$

Proof is done in the lecture.

In principle, we can derive the usual test statistics from the latter result. Though, as long as we do not know (we usually don't) σ^2 and $\Sigma_{X'X}$ we need to plug-in the (consistent!) estimators $S_{X'X}^{-1}$ and s_{UB}^2, where the consistency of the former estimator is provided by Theorem 5.2.1 and the consistency of s_{UB}^2 is provided by the following result.

Theorem 5.2.4 (Consistency of s_{UB}^2)**.**

$$s_{UB}^2 \xrightarrow{\text{p}} \sigma^2 \quad as \quad n \to \infty$$

Proof is skipped, but a detailed proof can be found here: https://www.statlect.com/fundamentals-of-statistics/OLS-estimator-properties

5.2.1 The Case of Heteroscedasticity

Theorem 5.2.3 can also be stated and proofed for conditionally heteroscedastic error terms. In this case one gets

$$\sqrt{n}(\hat{\beta}_n - \beta) \xrightarrow{\text{d}} \mathcal{N}\left(0, \Sigma_{X'X}^{-1} E(\varepsilon_i^2 \, X_i X_i') \Sigma_{X'X}^{-1}\right) \quad \text{as} \quad n \to \infty \qquad (5.1)$$

where $\Sigma_{X'X}^{-1} E(\varepsilon_i^2 \, X_i X_i') \Sigma_{X'X}^{-1}$ (i.e., the asymptotic variance of $\sqrt{n}(\hat{\beta}_n - \beta)$) is usually unknown and needs to be estimated from the data by

$$S_{X'X}^{-1} \widehat{E}(\varepsilon_i^2 \, X_i X_i') S_{X'X}^{-1} \xrightarrow{\text{p}} \Sigma_{X'X}^{-1} E(\varepsilon_i^2 \, X_i X_i') \Sigma_{X'X}^{-1} \quad \text{as} \quad n \to \infty,$$

where $\widehat{E}(\varepsilon_i^2\, X_i X_i')$ denotes some consistent estimator of $E(\varepsilon^2\, X_i X_i')$ such as one of the followingt choices:

$$\text{HC0:} \quad \widehat{E}(\varepsilon_i^2\, X_i X_i') = \frac{1}{n}\sum_{i=1}^{n} \hat{\varepsilon}_i^2 X_i X_i'$$

$$\text{HC1:} \quad \widehat{E}(\varepsilon_i^2\, X_i X_i') = \frac{1}{n}\sum_{i=1}^{n} \frac{n}{n-K} \hat{\varepsilon}_i^2 X_i X_i'$$

$$\text{HC2:} \quad \widehat{E}(\varepsilon_i^2\, X_i X_i') = \frac{1}{n}\sum_{i=1}^{n} \frac{\hat{\varepsilon}_i^2}{1-h_i} X_i X_i'$$

$$\text{HC3:} \quad \widehat{E}(\varepsilon_i^2\, X_i X_i') = \frac{1}{n}\sum_{i=1}^{n} \frac{\hat{\varepsilon}_i^2}{(1-h_i)^2} X_i X_i' \quad (\leftarrow \text{Most often used})$$

$$\text{HC4:} \quad \widehat{E}(\varepsilon_i^2\, X_i X_i') = \frac{1}{n}\sum_{i=1}^{n} \frac{\hat{\varepsilon}_i^2}{(1-h_i)^{\delta_i}} X_i X_i'$$

(These are the heteroscedasticity-consistent robust estimators from Chapter 3.6.

Note. In order to show that any of the above versions (HC0-4) of $\widehat{E}(\varepsilon_i^2\, X_i X_i')$ is a consistent estimator of $E(\varepsilon_i^2\, X_i X_i')$ we actually need to assume that the explanatory variables in X have finite *fourth* moments (see Hayashi, 2000, Chapter 2.5). So, for this, we would need to make our Assumption 1^* more restrictive (so far, only two moments are assumed for X).

5.2.2 Hypothesis Testing and Confidence Intervals

From our asymptotic results under the classic regression model (Assumptions 1^*, 2, 3^*, and 4) we get the following results important for testing statistical hypothesis.

5.2.2.1 Robust Hypothesis Testing: Multiple Parameters

Let us reconsider the following system of q-many null hypotheses:

$$H_0 : \underset{(q\times K)}{R}\underset{(K\times 1)}{\beta} - \underset{(q\times 1)}{r} = \underset{(q\times 1)}{0},$$

where the $(q \times K)$ matrix R and the q-vector $r = (r_1, \ldots, r_q)'$ are chosen by the statistician to specify her/his null hypothesis about the unknown true parameter vector β. To make sure that there are no redundant equations, it is required that $\text{rank}(R) = q$.

By contrast to the multiple parameter tests for small samples (see Chapter 4.1), we can work here with a heterosedasticity robust test statistic which is applicable for heteroscedastic error terms:

$$W = n(R\hat{\beta}_n - r)'[R\, S_{X'X}^{-1}\widehat{E}(\varepsilon_i^2\, X_i X_i')S_{X'X}^{-1}\, R']^{-1}(R\hat{\beta}_n - r) \overset{H_0}{\to}_d \chi^2(q) \quad (5.2)$$

as $n \to \infty$. The price to pay is that the distribution of the test statistic under the null hypothesis is only valid asymptotically for large n. That is, the critical values taken from the asymptotic distribution will be useful only for "large" samples sizes. In case of homoscedastic error terms, one can substitute $S_{X'X}^{-1}\widehat{E}(\varepsilon_i^2\, X_i X_i')S_{X'X}^{-1}$ by $s_{UB}^2 S_{X'X}^{-1}$.

Finite-sample correction. In order to improve the finite-sample performance of this test, one usually uses the $F_{q,n-K}$ distribution with q and $n - K$ degrees of freedoms instead of the $\chi^2(q)$ distribution. Asymptotically ($n \to \infty$), $F_{q,n-K}$ is equivalent to $\chi^2(q)$. However, for finite sample sizes n (i.e., the practically relevant case) $F_{q,n-K}$ leads to larger critical values which helps to account for the estimation errors in $S_{X'X}^{-1}\widehat{E}(\varepsilon_i^2\, X_i X_i')S_{X'X}^{-1}$ (or in $s_{UB}^2 S_{X'X}^{-1}$) which are otherwise neglected by the pure asymptotic perspective.

5.2.2.2 Robust Hypothesis Testing: Single Parameters

Let us reconsider the case of hypotheses about only one parameter β_k, with $k = 1, \ldots, K$

$$\begin{aligned} H_0: &\quad \beta_k = r \\ H_A: &\quad \beta_k \neq r \end{aligned}$$

We can selecting the kth diagonal element of the test-statistic in (5.2) and taking the square root yields

$$t = \frac{\sqrt{n}\left(\hat{\beta}_k - r\right)}{\sqrt{\left[S_{X'X}^{-1}\widehat{E}(\varepsilon_i^2\, X_i X_i')S_{X'X}^{-1}\right]_{kk}}} \xrightarrow{H_0}_d \mathcal{N}(0,1).$$

This t test statistic allows for heteroscedastic error terms. In case of homoscedastic error terms, one can substitute $[S_{X'X}^{-1}\widehat{E}(\varepsilon_i^2\, X_i X_i')S_{X'X}^{-1}]_{kk}$ by $s_{UB}^2[S_{X'X}^{-1}]_{kk}$.

Finite-sample correction. In order to improve the finite-sample performance of this t test, one usually uses the $t_{(n-K)}$ distribution with $n-K$ degrees of freedoms instead of the $\mathcal{N}(0,1)$ distribution. Asymptotically ($n \to \infty$), $t_{(n-K)}$ is equivalent to $\mathcal{N}(0,1)$. However, for finite sample sizes n (i.e., the practically relevant case) t_{n-K} leads to larger critical values which helps to account for the estimation errors in $[S_{X'X}^{-1}\widehat{E}(\varepsilon_i^2\, X_i X_i')S_{X'X}^{-1}]_{kk}$ (or in $s_{UB}^2[S_{X'X}^{-1}]_{kk}$) which are otherwise neglected by the pure asymptotic perspective.

5.2.2.3 Robust Confidence Intervals

Following the derivations in Chapter 4.6, but using the expression for the robust standard errors, we get the following heteroscedasticity robust (random) $(1-\alpha)\cdot 100\%$ confidence interval

$$\text{CI}_{1-\alpha} = \left[\hat{\beta}_k \pm t_{1-\alpha/2,n-K}\sqrt{n^{-1}[S_{X'X}^{-1}\widehat{E}(\varepsilon_i^2\, X_i X_i')S_{X'X}^{-1}]_{kk}}\right].$$

Here, the coverage probability is an asymptotic coverage probability with $P(\beta_k \in \text{CI}_{1-\alpha}) \to 1-\alpha$ as $n \to \infty$.

5.3 Practice: Large Sample Inference

Let's apply the above asymptotic inference methods using R. As in Chapter 4.7 we, first, program a function `myDataGenerator()` which allows us to generate data from the following model, i.e., from the following fully specified data generating process:

$$\begin{aligned} Y_i &= \beta_1 + \beta_2 X_{i2} + \beta_3 X_{i3} + \varepsilon_i, \qquad i = 1, \ldots, n \\ \beta &= (\beta_1, \beta_2, \beta_3)' = (2, 3, 4)' \\ X_{i2} &\sim U[-4, 4] \\ X_{i3} &\sim U[-5, 5] \\ \varepsilon_i \,|X_i &\sim U[-0.5|X_{i2}|, 0.5|X_{i2}|], \end{aligned}$$

where (Y_i, X_i) is assumed i.i.d. across $i = 1, \ldots, n$ with X_{i2} and X_{i3} being independent of each other. Note that, by contrast to Chapter 4.7, the error terms are conditionally heteroscedastic ($\text{Var}(\varepsilon_i \,|X_i) = \frac{1}{12} X_{i2}^2$) and not Gaussian.

As a side note: The unconditional variance follows by the law of total variance and is given by $\text{Var}(\varepsilon_i) = E(\text{Var}(\varepsilon_i \,|X_i)) + \text{Var}(E(\varepsilon_i \,|X_i)) = E(\frac{1}{12} X_{i2}^2) + 0 = \frac{1}{(12)}(\frac{1}{(12)}(4 - (-4))^2) = \frac{4}{9}$).

Moreover, by contrast to Chapter 4.7, we here do not need to sample new realizations of $Y_1, \ldots, Y_n$ conditionally on a given data matrix X since the asymptotic normality statement is not conditionally on X. (The small sample normality result in (4.1) in Chapter 4.7 is conditionally on X; however, the large sample normality result in (5.1) is unconditional on X.) Therefore, the option to condition on X is here removed from the R-function `myDataGenerator()`.

```r
## Function to generate artificial data
myDataGenerator <- function(n, beta){
  ##
  X   <- cbind(rep(1, n),
               runif(n, -4, 4),
               runif(n, -5, 5))
  ##
  eps  <- runif(n, -.5 * abs(X[,2]), +.5 * abs(X[,2]))
  Y    <- X %*% beta + eps
  data <- data.frame("Y"=Y,
                     "X_1"=X[,1], "X_2"=X[,2], "X_3"=X[,3])
  ##
  return(data)
}
```

5.3.1 Normally Distributed $\hat{\beta}$ for $n \to \infty$

The above data generating process fulfills our regulatory assumptions Assumption 1*, 2, 3*, and 4. So, by theory, the estimators $\hat{\beta}_k$ should be normal distributed for large sample sizes n – unconditionally on X and even for heteroscedastic error terms.

$$\sqrt{n}\left(\hat{\beta}_k - \beta_k\right) \to_d \mathcal{N}\left(0, \left[\Sigma_{X'X}^{-1} E(\varepsilon_i^2\, X_i X_i') \Sigma_{X'X}^{-1}\right]_{kk}\right)$$

Or:

$$\hat{\beta}_k \to_d \mathcal{N}\left(\beta_k,\ n^{-1}\ \left[\Sigma_{X'X}^{-1} E(\varepsilon_i^2\, X_i X_i') \Sigma_{X'X}^{-1}\right]_{kk}\right)$$

Note: Mathematically, the latter is a bit sloppy since the right hand side of $\to_d$ depends on n, i.e., is not the stable limit object for $n \to \infty$. However, this sloppiness is nevertheless instructive since it gives us the approximative distribution for given largish sample sizes like $n = 100$.

For our above specified data generating process, we have

- From the assumed distributions of X_{i2} and X_{i3} we have that (we used that (i) $E(X^2) = V(X)$ if X has mean zero and (ii) that the variance of uniform distributed random variables $\frac{1}{12}(b-a)^2$):

$$\Sigma_{X'X} = E(S_{X'X}) = E(X_i X_i') = \begin{pmatrix} 1 & 0 & 0 \\ 0 & E(X_{i2}^2) & 0 \\ 0 & 0 & E(X_{i3}^2) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{16}{3} & 0 \\ 0 & 0 & \frac{25}{3} \end{pmatrix}$$

- Moreover, $E(\varepsilon_i^2\, X_i X_i') = E(X_i X_i' E(\varepsilon_i^2\,|X_i)) = E\left(X_i X_i' \left(\frac{1}{12} X_{i2}^2\right)\right)$ such that

$$\begin{aligned} E(\varepsilon_i^2\, X_i X_i') &= \begin{pmatrix} E\left(\frac{1}{12} X_{i2}^2\right) & 0 & 0 \\ 0 & E\left(X_{i2}^2 \cdot \frac{1}{12} X_{i2}^2\right) & 0 \\ 0 & 0 & E\left(X_{i3}^2 \cdot \frac{1}{12} X_{i2}^2\right) \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{12} E\left(X_{i2}^2\right) & 0 & 0 \\ 0 & \frac{1}{12} E\left(X_{i2}^4\right) & 0 \\ 0 & 0 & \frac{1}{12} E\left(X_{i2}^2\right)\, E\left(X_{i3}^2\right) \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{12}\frac{16}{3} & 0 & 0 \\ 0 & \frac{1}{12}\frac{256}{5} & 0 \\ 0 & 0 & \frac{1}{12}\frac{16}{3}\frac{25}{3} \end{pmatrix} = \begin{pmatrix} \frac{4}{9} & 0 & 0 \\ 0 & \frac{64}{15} & 0 \\ 0 & 0 & \frac{100}{27} \end{pmatrix} \end{aligned}$$

Note: For $X \sim U[a,b]$ you can use that $E(X^k) = \frac{b^{k+1}-a^{k+1}}{(k+1)(b-a)}$, $k = 1, 2, \ldots$; see, for instance, Wikipedia.

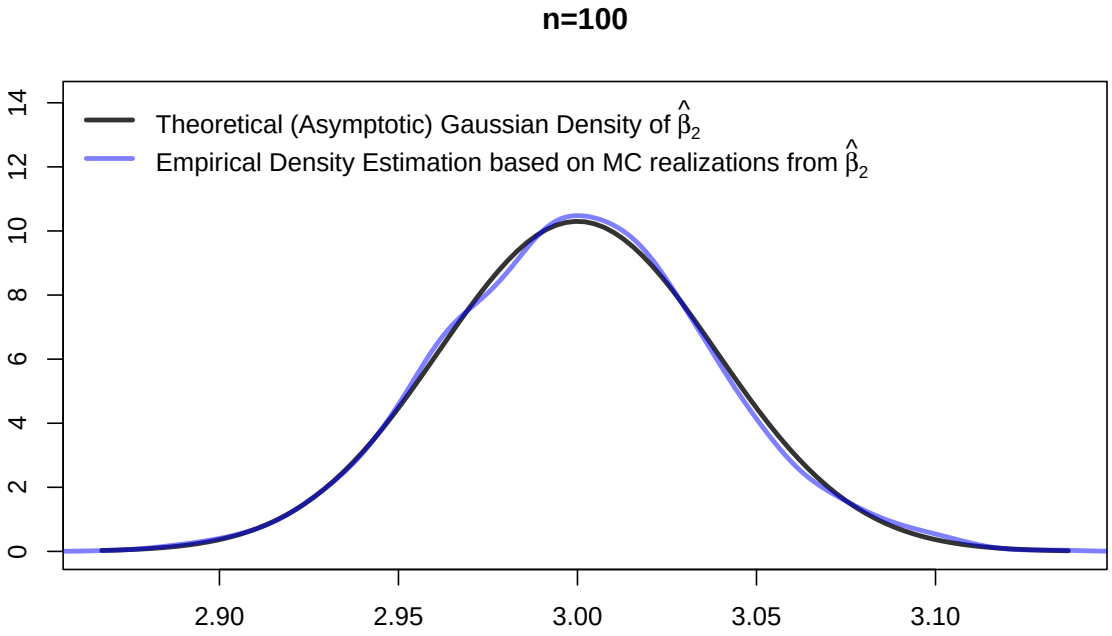
So, for instance, for $\hat{\beta}_2$ we have the following theoretical large sample distribution:

$$\hat{\beta}_2 \to_d \mathcal{N}\left(\beta_2,\ n^{-1}\left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{16}{3} & 0 \\ 0 & 0 & \frac{25}{3} \end{pmatrix}^{-1} \begin{pmatrix} \frac{4}{9} & 0 & 0 \\ 0 & \frac{64}{15} & 0 \\ 0 & 0 & \frac{100}{27} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{16}{3} & 0 \\ 0 & 0 & \frac{25}{3} \end{pmatrix}^{-1}\right]_{22}\right)$$

Let's use a Monte Carlo simulation to check how well this theoretical large sample ($n \to \infty$) distribution of $\hat{\beta}_2$ works as an approximative distribution for a largish sample size of $n = 100$.

```r
set.seed(123)
n           <- 100      # a largish sample size
beta_true   <- c(2,3,4) # true data vector

## Mean and variance of the true asymptotic
## normal distribution of beta_hat_2:
# true mean
beta_true_2     <- beta_true[2]
# true variance
var_true_beta_2 <- (solve(diag(c(1, 16/3, 25/3)))    %*%
                          diag(c(4/9, 64/15, 100/27))%*%
                     solve(diag(c(1, 16/3, 25/3))))[2,2]/n

## Let's generate 5000 realizations from beta_hat_2, and check
## whether their distribution is close to the true normal
## distribution.
## (We don't condition on X since the theoretical limit
## distribution is unconditional on X)
rep        <- 5000 # MC replications
beta_hat_2 <- rep(NA, times=rep)
##
for(r in 1:rep){
    MC_data <- myDataGenerator(n    = n,
                               beta = beta_true)
    lm_obj        <- lm(Y ~ X_2 + X_3, data = MC_data)
    beta_hat_2[r] <- coef(lm_obj)[2]
}

## Compare:
## True beta_2 versus average of beta_hat_2 estimates
```

```r
beta_true_2
#> [1] 3
round(mean(beta_hat_2), 3)
#> [1] 3
## True variance of beta_hat_2 versus
## empirical variance of beta_hat_2 estimates
round(var_true_beta_2, 5)
#> [1] 0.0015
round(var(beta_hat_2), 5)
#> [1] 0.00147

## True normal distribution of beta_hat_2 versus
## empirical density of beta_hat_2 estimates
library("scales")
curve(expr = dnorm(x, mean = beta_true_2,
                   sd=sqrt(var_true_beta_2)),
      xlab="",ylab="", col=gray(.2), lwd=3, lty=1,
xlim=range(beta_hat_2),ylim=c(0,14.1),main=paste0("n=",n))
lines(density(beta_hat_2, bw = bw.SJ(beta_hat_2)),
      col=alpha("blue",.5), lwd=3)
legend("topleft", lty=c(1,1), lwd=c(3,3),
     col=c(gray(.2), alpha("blue",.5)), bty="n", legend=
c(expression(
  "Theoretical (Asymptotic) Gaussian Density of"~hat(beta)[2]),
  expression(
  "Empirical Density Estimation based on MC realizations from"~
  hat(beta)[2])))
```



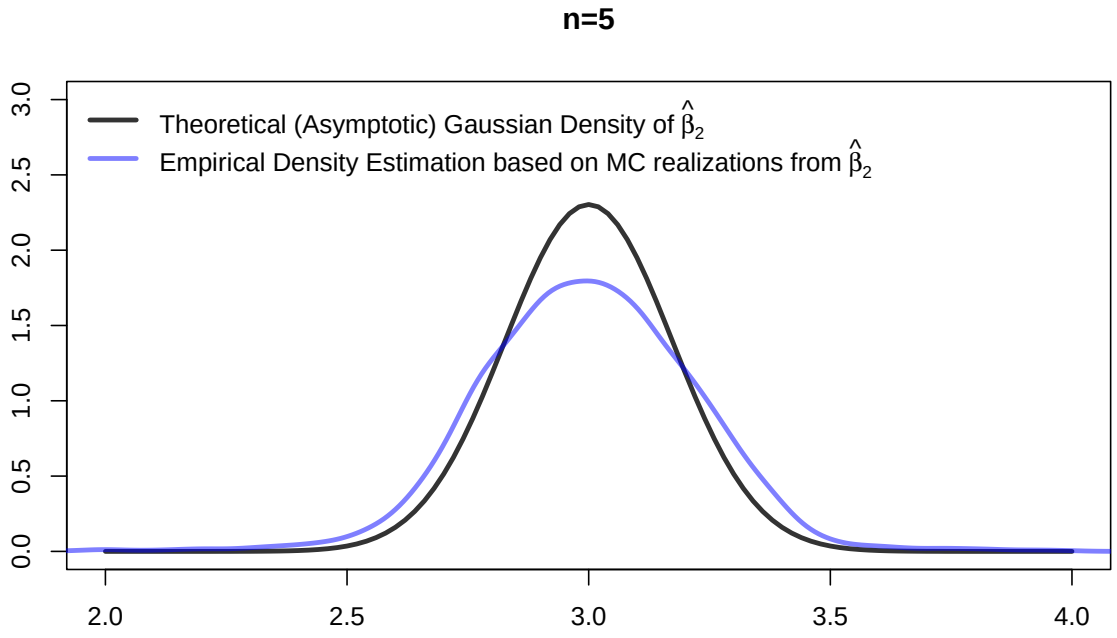
Great! The nonparametric density estimation (estimated via `density()`) computed from the simulated realizations of $\hat{\beta}_2$ is indicating that $\hat{\beta}_2$ is really normally distributed as described by our theoretical result in Theorem 5.2.3 (homoscedastic case) and in Equation (5.1) (heteroscedastic case).

However, is the asymptotic distribution of $\hat{\beta}_2$ also usable for (very) small samples like $n = 5$? Let's check that:

```
set.seed(123)
n           <- 5       # a small sample size
beta_true   <- c(2,3,4) # true data vector

## Mean and variance of the true asymptotic
```

```r
## normal distribution of beta_hat_2:
# true mean
beta_true_2     <- beta_true[2]
# true variance
var_true_beta_2 <- (solve(diag(c(1, 16/3, 25/3)))%*%
                          diag(c(4/9, 64/15, 100/27))%*%
                     solve(diag(c(1, 16/3, 25/3))))[2,2]/n

## Let's generate 5000 realizations from beta_hat_2, and check
## whether their distribution is close to the true normal
## distribution.
## (We don't condition on X since the theoretical limit
## distribution is unconditional on X)
rep        <- 5000 # MC replications
beta_hat_2 <- rep(NA, times=rep)
##
for(r in 1:rep){
    MC_data <- myDataGenerator(n    = n,
                               beta = beta_true)
    lm_obj        <- lm(Y ~ X_2 + X_3, data = MC_data)
    beta_hat_2[r] <- coef(lm_obj)[2]
}

## Compare:
## True beta_2 versus average of beta_hat_2 estimates
beta_true_2
#> [1] 3
round(mean(beta_hat_2), 3)
#> [1] 2.996
## True variance of beta_hat_2 versus
```

```
## empirical variance of beta_hat_2 estimates
round(var_true_beta_2, 5)
#> [1] 0.03
round(var(beta_hat_2), 5)
#> [1] 0.05621

## True normal distribution of beta_hat_2 versus
## empirical density of beta_hat_2 estimates
library("scales")
curve(expr = dnorm(x, mean = beta_true_2,
                   sd=sqrt(var_true_beta_2)),
      xlab="",ylab="", col=gray(.2), lwd=3, lty=1,
xlim=c(2,4), ylim=c(0,3),main=paste0("n=",n))
lines(density(beta_hat_2, bw = bw.SJ(beta_hat_2)),
      col=alpha("blue",.5), lwd=3)
legend("topleft", lty=c(1,1), lwd=c(3,3),
     col=c(gray(.2), alpha("blue",.5)), bty="n", legend=
c(expression(
  "Theoretical (Asymptotic) Gaussian Density of"~hat(beta)[2]),
  expression(
  "Empirical Density Estimation based on MC realizations from"~
  hat(beta)[2])))
```



Not good. The actual distribution has substantially fatter tails. That is, if we would use the quantiles of the asymptotic distribution, we would falsely reject the null-hypothesis too often (probability of type I errors would be larger than the significance level). But asymptotic are kicking in pretty fast here: things become much more reliable already for $n = 10$.

5.3.2 Testing Multiple and Single Parameters

In the following, we do inference about multiple parameters. We test

$$\begin{aligned} H_0 &: \beta_2 = 3 \quad \text{and} \quad \beta_3 = 5 \\ \text{versus} \quad H_A &: \beta_2 \neq 3 \quad \text{and/or} \quad \beta_3 \neq 5. \end{aligned}$$

Or equivalently

$$H_0 : R\beta - r = 0$$
$$H_A : R\beta - r \neq 0,$$

where

$$R = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad r = \begin{pmatrix} 3 \\ 5 \end{pmatrix}.$$

The following R code can be used to test this hypothesis:

```
suppressMessages(library("car")) # for linearHyothesis()
# ?linearHypothesis
library("sandwich")

## Generate data
MC_data <- myDataGenerator(n    = 100,
                           beta = beta_true)

## Estimate the linear regression model parameters
lm_obj <- lm(Y ~ X_2 + X_3, data = MC_data)

vcovHC3_mat <- sandwich::vcovHC(lm_obj, type="HC3")

## Option 1:
car::linearHypothesis(model = lm_obj,
                      hypothesis.matrix = c("X_2=3", "X_3=5"),
                      vcov=vcovHC3_mat)
#> Linear hypothesis test
#>
#> Hypothesis:
#> X_2 = 3
```

```
#> X_3 = 5
#>
#> Model 1: restricted model
#> Model 2: Y ~ X_2 + X_3
#>
#> Note: Coefficient covariance matrix supplied.
#>
#>   Res.Df Df      F    Pr(>F)
#> 1     99
#> 2     97  2 1150.4 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Option 2:
R <- rbind(c(0,1,0),
           c(0,0,1))
car::linearHypothesis(model = lm_obj,
                      hypothesis.matrix = R,
                      rhs = c(3,5),
                      vcov=vcovHC3_mat)
#> Linear hypothesis test
#>
#> Hypothesis:
#> X_2 = 3
#> X_3 = 5
#>
#> Model 1: restricted model
#> Model 2: Y ~ X_2 + X_3
#>
#> Note: Coefficient covariance matrix supplied.
```

```
#>
#>   Res.Df Df      F    Pr(>F)
#> 1     99
#> 2     97  2 1150.4 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is very small and allows us to reject the (false) null-hypothesis at any of the usual significance levels.

Next, we do inference about a single parameter. We test

$$\begin{aligned} & H_0 : \beta_3 = 5 \\ \text{versus} \quad & H_A : \beta_3 \neq 5. \end{aligned}$$

```
# Load libraries
library("lmtest")   # for coeftest()
library("sandwich") # for vcovHC()

## Generate data
n <- 100
MC_data <- myDataGenerator(n    = n,
                           beta = beta_true)

## Estimate the linear regression model parameters
lm_obj <- lm(Y ~ X_2 + X_3, data = MC_data)

## Robust t test

## Robust standard error for \hat{\beta}_3:
SE_rob <- sqrt(vcovHC(lm_obj, type = "HC3")[3,3])
```

```r
## hypothetical (H0) value of \beta_3:
beta_3_H0 <- 5
## estimate for beta_3:
beta_3_hat <- coef(lm_obj)[3]
## robust t-test statistic
t_test_stat <- (beta_3_hat - beta_3_H0)/SE_rob
## p-value
K <- length(coef(lm_obj))
##
p_value <- 2 * min(   pt(q = t_test_stat, df = n - K),
                   1- pt(q = t_test_stat, df = n - K))
p_value
#> [1] 4.330845e-65
```

Again, the p-value is very small and allows us to reject the (false) null-hypothesis at any of the usual significance levels.

Chapter 6

Instrumental Variables Regression

The current version of this chapter is basically completly taken from the free online book: www.econometrics-with-r.org (Hanck et al., 2021)

Regression models may suffer from problems like omitted variables, measurement errors and simultaneous causality. If so, the error term ε_i is correlated with the regressor, X_{ik} say, and the corresponding coefficient of interest, β_k, is estimated **inconsistently**. If one is lucky, one can add, for instance, the omitted variables to the regression to mitigate the risk of biased estimations ("omitted variables bias"). However, if omitted variables cannot be measured or are not available for other reasons, multiple regression cannot solve the problem. The same issue arises if there is "simultaneous causality". When causality runs from X to Y *and vice versa* (e.g. if Y = Demanded quantity of a good and X = Price of this good), there will be an estimation bias that cannot be corrected for by multiple regression.

A general technique for obtaining a consistent estimator of the coefficient of interest is instrumental variables (IV) regression. In this chapter we focus on the IV regression tool called *two-stage least squares* (TSLS). The first sections briefly recap the general mechanics and assumptions of IV regression and show how to perform TSLS estimation using R. Next, IV regression is

used for estimating the elasticity of the demand for cigarettes — a classical example where multiple regression fails to do the job because of simultaneous causality.

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

Consider the simple regression model

$$Y_i = \beta_1 + \beta_2 X_i + \varepsilon_i \quad , \quad i = 1, \ldots, n, \tag{6.1}$$

where the error term ε_i is correlated with the regressor X_i (X is the called "*endogenous*"). In this case Assumption 2 is violated, that is, strict exogeneity and orthogonality between X_i and ε_i do not hold. Therefore, OLS estimation (also maximum likelihood and methods of moments estimation) is inconsistent for the true β_2. In the most simple case, IV regression uses a single instrumental variable Z_i to obtain a consistent estimator for β_2.

Conditions for valid instruments: Z_i must satisfy two conditions to be a valid instrument:

1. **Instrument relevance condition:**
   X_i and its instrument Z_i *must be* correlated: $\rho_{Z,X} \neq 0$.
2. **Instrument exogeneity condition:**
   $E(\varepsilon_i | Z_i) = 0$. As a consequence: The instrument Z_i *must not be* correlated with the error term ε_i: $\rho_{Z,\varepsilon} = 0$.

6.1.1 The Two-Stage Least Squares Estimator

As can be guessed from its name, TSLS proceeds in two stages. In the first stage, the variation in the endogenous regressor X_i is decomposed into a

"problem-free" component that is explained by the (exogenous) instrument Z_i and a problematic component that is correlated with the error ε_i. The second stage uses the problem-free component of the variation in X_i to estimate β_2.

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 that is explained by Z_i while ν_i is the component (an error term) that cannot be explained by Z_i and exhibits correlation with ε_i.

Using the OLS estimates $\hat{\pi}_0$ and $\hat{\pi}_1$ we obtain predicted values $\widehat{X}_i = \hat{\pi}_0 + \hat{\pi}_1 Z_i,\ i = 1, \ldots, n$. If Z_i is a valid instrument, the $\widehat{X}_i$ are problem-free in the sense that $\widehat{X}_i$ is **exogenous** in a regression of Y_i on $\widehat{X}_i$ which is done in the second stage regression. The second stage produces $\hat{\beta}_1^{TSLS}$ and $\hat{\beta}_2^{TSLS}$, the TSLS estimates of β_1 and β_2.

For the case of a single instrument one can show that the TSLS estimator of β_2 is

$$\hat{\beta}_2^{TSLS} = \frac{s_{ZY}}{s_{ZX}} = \frac{\frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \overline{Y})(Z_i - \overline{Z})}{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})(Z_i - \overline{Z})}, \tag{6.2}$$

which is nothing but the ratio of the sample covariance between Z_i and Y_i to the sample covariance between Z_i and X_i.

The estimator in (6.2) is a consistent estimator for β_2 in (6.1) under the assumption that Z_i is a valid instrument. The CLT implies that the distribution of $\hat{\beta}_2^{TSLS}$ can be approximated by a normal distribution if the sample size n is large. This allows us to use t-statistics, F-statistics, confidence intervals, etc.

6.1.2 Application: Demand For Cigarettes (1/2)

The relation between the demand for and the price of commodities is a simple yet widespread problem in economics. Health economics is concerned with the study of how health-affecting behavior of individuals is influenced by the health-care system and regulation policy. Probably the most prominent example in public policy debates is smoking as it is related to many illnesses and negative externalities.

It is plausible that cigarette consumption can be reduced by taxing cigarettes more heavily. The question is by *how much* taxes must be increased to reach a certain reduction in cigarette consumption. Economists use elasticities to answer this kind of question. Since the price elasticity for the demand of cigarettes is unknown, it must be estimated. A simple OLS regression of log quantity on log price cannot be used to estimate the effect of interest since there is simultaneous causality between demand and supply. Instead, IV regression can be used.

We use the data set `CigarettesSW` which comes with the package `AER`. 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. In the following, however, we consider data for the cross section of states in 1995 only – that is, we transform the panel data to a cross-sectional data set. We start by loading the package, attaching the data set. An overview about summary statistics for each of the variables is returned by `summary(CigarettesSW)`. Use `?CigarettesSW` for a detailed description of the variables.

```
# load the data set and get an overview
library("AER")
data("CigarettesSW")
summary(CigarettesSW)
```

```
#>      state      year         cpi          population          packs
#>  AL     : 2   1985:48   Min.   :1.076   Min.   :  478447   Min.   : 49.2
#>  AR     : 2   1995:48   1st Qu.:1.076   1st Qu.: 1622606   1st Qu.: 92.4
#>  AZ     : 2              Median :1.300   Median : 3697472   Median :110.1
#>  CA     : 2              Mean   :1.300   Mean   : 5168866   Mean   :109.1
#>  CO     : 2              3rd Qu.:1.524   3rd Qu.: 5901500   3rd Qu.:123.5
#>  CT     : 2              Max.   :1.524   Max.   :31493524   Max.   :197.9
#>  (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
#>
```

We are interested in estimating β_2 in

$$\log(Q_i^{cigarettes}) = \beta_1 + \beta_2 \log(P_i^{cigarettes}) + \varepsilon_i, \tag{6.3}$$

where $Q_i^{cigarettes}$ is the number of cigarette packs per capita sold and $P_i^{cigarettes}$ is the after-tax average real price per pack of cigarettes in state $i = 1, \ldots, n = 48$.

The instrumental variable we are going to use for instrumenting the endogenous regressor $\log(P_i^{cigarettes})$ is $SalesTax$, the portion of taxes on cigarettes arising from the general sales tax. $SalesTax$ is measured in dollars per pack. The idea is that:

1. $SalesTax$ is a relevant instrument as it is included in the after-tax average price per pack.

2. Also, it is plausible that $SalesTax$ is exogenous since the sales tax does not influence quantity sold directly but indirectly through the price.

In the following, we perform some transformations in order to obtain deflated cross section data for the year 1995. We also compute the sample correlation between the sales tax and price per pack. The sample correlation is a consistent estimator of the population correlation. The estimate of approximately 0.614 indicates that $SalesTax$ and $P_i^{cigarettes}$ exhibit positive correlation which meets our expectations: higher sales taxes lead to higher prices. However, a correlation analysis like this is not sufficient for checking whether the instrument is relevant. We will later come back to the issue of checking whether an instrument is relevant and exogenous; see Chapter 6.3.

```
# 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 first stage regression is

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

We estimate this model in R using `lm()`. In the second stage we run a regression of $\log(Q_i^{cigarettes})$ on $\log(\widehat{P_i^{cigarettes}})$ to obtain $\widehat{\beta}_1^{TSLS}$ and $\widehat{\beta}_2^{TSLS}$.

```
# 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 is

$$\widehat{\log(P_i^{cigarettes})} = \underset{(0.03)}{4.62} + \underset{(0.005)}{0.031}\, SalesTax_i$$

which predicts the relation between sales tax price per cigarettes to be positive. How much of the observed variation in $\log(P^{cigarettes})$ is explained by the instrument $SalesTax$? This can be answered by looking at the regression's R^2 which states that about 47% of the variation in after tax prices is explained by the variation of the sales tax across states.

```
# inspect the R^2 of the first stage regression
summary(cig_s1)$r.squared
#> [1] 0.4709961
```

We next store $\widehat{\log(P_i^{cigarettes})}$, 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
```

Next, we run the second stage regression which gives us the TSLS estimates we seek.

```
# run the stage 2 regression
cig_s2 <- lm(log(c1995$packs) ~ lcigp_pred)
coeftest(cig_s2, vcov = vcovHC, type = "HC1")
#>
#> t test of coefficients:
#>
#>              Estimate Std. Error t value  Pr(>|t|)
#> (Intercept)   9.7199     1.5971  6.0859 2.153e-07 ***
#> lcigp_pred   -1.0836     0.3337 -3.2472  0.002178 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Thus estimating the model (6.3) using TSLS yields

$$\widehat{\log(Q_i^{cigarettes})} = \underset{(1.60)}{9.72} - \underset{(0.33)}{1.08}\log(P_i^{cigarettes}), \tag{6.4}$$

The function `ivreg()` from the package `AER` carries out TSLS procedure automatically. It is used similarly as `lm()`. Instruments can be added to the usual specification of the regression formula using a vertical bar separating the model equation from the instruments. Thus, for the regression at hand the correct formula is `log(packs) ~ log(rprice) | salestax`.

```
# 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 find that the coefficient estimates coincide for both approaches.

Two Notes on the Computation of TSLS Standard Errors:

1. We have demonstrated that running the individual regressions for each stage of TSLS using ‘lm()‘ leads to the same coefficient estimates as when using ‘ivreg()‘. However, the standard errors reported for the second-stage regression, e.g., by ‘coeftest()‘ or ‘summary()‘, are **invalid**: neither adjusts for using predictions from the first-stage regression as regressors in the second-stage regression. Fortunately, ‘ivreg()‘ performs the necessary adjustment automatically. This is another advantage over manual step-by-step estimation which we have done above for demonstrating the mechanics of the procedure.

2. Just like in multiple regression it is important to compute heteroskedasticity-robust standard errors as we have done above using ‘vcovHC()‘.

The TSLS estimate for β_2 in (6.4) suggests that an increase in cigarette prices by one percent reduces cigarette consumption by roughly 1.08 percentage points, which is fairly elastic. However, we should keep in mind that this estimate might not be trustworthy even though we used IV estimation: there still might be a bias due to **omitted variables**. Thus a multiple IV regression approach is needed to reduce the risk of omitted variable biases.

6.2 The General IV Regression Model

The simple IV regression model is easily extended to a multiple regression model which we refer to as the general IV regression model. In this model we distinguish between four types of variables: the dependent variable, included exogenous variables, included endogenous variables and instrumental variables:

$$Y_i = \beta_1 + \beta_2 X_{2i} + \cdots + \beta_K X_{Ki} + \beta_{K+1} W_{1i} + \cdots + \beta_{K+r} W_{ri} + \varepsilon_i, \qquad (6.5)$$

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

- Y_i is the dependent variable
- $\beta_1, \ldots, \beta_{K+r}$ are $K + r$ unknown regression coefficients
- $X_{2i}, \ldots, X_{Ki}$ are $K - 1$ endogenous regressors
- $W_{1i}, \ldots, W_{ri}$ are r exogenous regressors which are uncorrelated with ε_i
- ε_i is the error term
- $Z_{1i}, \ldots, Z_{mi}$ are m instrumental variables

The coefficients are **overidentified** if $m > (K-1)$. If $m < (K-1)$, the coefficients are **underidentified** and when $m = (K-1)$ they are **exactly identified**. For estimation of the IV regression model we require exact identification or overidentification.

Estimating regression models with TSLS using multiple instruments by means of `ivreg()` is straightforward. There are, however, some subtleties in correctly specifying the regression formula. Assume that you want to estimate the model

$$Y_i = \beta_1 + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 W_{1i} + \varepsilon_i$$

where X_{2i} and X_{3i} are endogenous regressors that shall be instrumented by Z_{1i}, Z_{2i} and Z_{3i}, and where W_{1i} is an exogenous regressor. Say the corresponding data is available in a `data.frame` with column names `y`, `x2`, `x3`, `w1`, `z1`, `z2` and `z3`. It might be tempting to specify the argument `formula` in your call of `ivreg()` as `y ~ x2 + x3 + w1 | z1 + z2 + z3` which is, however, **wrong**. As explained in the documentation of `ivreg()` (see `?ivreg`), 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 ~ x2 + x3 + w1 | w1 + z1 + z2 + z3`, where `w1` is "instrumenting itself".

Similarly to the simple IV regression model, the general IV model (6.5) can be estimated using the two-stage least squares estimator:

- **First-stage regression(s):**
  Run an OLS regression for each of the endogenous variables $(X_{2i}, \ldots, X_{Ki})$ on all instrumental variables $(Z_{1i}, \ldots, Z_{mi})$, all exogenous variables $(W_{1i}, \ldots, W_{ri})$ **and an intercept**. Compute the fitted values $(\widehat{X}_{2i}, \ldots, \widehat{X}_{Ki})$.

- **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 $\widehat{\beta}_1^{TSLS}, \ldots, \widehat{\beta}_{K+r}^{TSLS}$, the TSLS estimates of the model coefficients.

In the general IV regression model, the instrument relevance and instrument exogeneity assumptions are equivalent to the case of the simple regression model with a single endogenous regressor and only one instrument. That is, for $Z_{1i}, \ldots, Z_{mi}$ to be a set of valid instruments, the following two conditions must be fulfilled:

1. **Instrument Relevance**
   If there are $K-1$ endogenous variables, r exogenous variables and $m \geq K-1$ instruments and the $\widehat{X}_{2i}^*, \ldots, \widehat{X}_{Ki}^*$ are the predicted values from the $K-1$ population first stage regressions, it must hold that

   $$(\widehat{X}_{2i}^*, \ldots, \widehat{X}_{Ki}^*, W_{1i}, \ldots, W_{ri}, 1)$$

   are not perfectly multicollinear, where "1" denotes the constant regressor (intercept) which equals 1 for all observations.
   Explanations: Let's say there is only one endogenous regressor X_i. If all the instruments $Z_{1i}, \ldots, Z_{mi}$ are irrelevant, all the $\widehat{X}_i^*$ are just the mean of X such that there is perfect multicollinearity with the constant intercept 1.

2. **Instrument Exogeneity**
   $E(\varepsilon_i \,|Z_{1i}, \ldots, Z_{im}) = 0$. Consequently, all m instruments must be uncorrelated with the error term,

   $$\rho_{Z_1,\varepsilon} = 0, \ldots, \rho_{Z_m,\varepsilon} = 0.$$

One can show that if the IV regression assumptions hold, the TSLS estimator in (6.5) is consistent and normally distributed when the sample size n is large.

That is, if we have valid instruments, we obtain valid statistical inference using t-tests, F-tests and confidence intervals for the model coefficients.

6.2.1 Application: Demand for Cigarettes (2/2)

The estimated elasticity of the demand for cigarettes in (6.1) is 1.08. Although (6.1) was estimated using IV regression it is plausible that this IV estimate is biased. The TSLS estimator is inconsistent for the true β_2 if the instrument (here: the real sales tax per pack) is invalid, i.e., if the instrument correlates with the error term. This is likely to be the case here since there are economic factors, like state income, which impact the demand for cigarettes and correlate with the sales tax. States with high personal income tend to generate tax revenues by income taxes and less by sales taxes. Consequently, state income should be included in the regression model.

$$\log(Q_i^{cigarettes}) = \beta_1 + \beta_2 \log(P_i^{cigarettes}) + \beta_3 \log(income_i) + \varepsilon_i \tag{6.6}$$

Before estimating (6.6) using `ivreg()` we define *income* as real per capita income `rincome` and append it to the data set `CigarettesSW`.

```
# add real income to the dataset (cpi: consumer price index)
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

$$\widehat{\log(Q_i^{cigarettes})} = \underset{(1.26)}{9.42} - \underset{(0.37)}{1.14}\log(P_i^{cigarettes}) + \underset{(0.31)}{0.21}\log(income_i). \tag{6.7}$$

In the following we add the cigarette-specific taxes ($cigtax_i$) as a further instrumental variable and estimate again using TSLS.

```
# 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
```

Using the two instruments $salestax_i$ and $cigtax_i$ we have $m = 2$ for one endogenous regressor – so the coefficient on the endogenous regressor $\log(P_i^{cigarettes})$ is *overidentified*. The TSLS estimate of (6.6) is

$$\widehat{\log(Q_i^{cigarettes})} = \underset{(0.96)}{9.89} - \underset{(0.25)}{1.28}\log(P_i^{cigarettes}) + \underset{(0.25)}{0.28}\log(income_i). \tag{6.8}$$

Should we trust the estimates presented in (6.7) or rather rely on (6.8)? The estimates obtained using both instruments are more precise since in (6.8) all standard errors reported are smaller than in (6.7). In fact, the standard error for the estimate of the demand elasticity is only two thirds of the standard error when the sales tax is the only instrument used. This is due to more information being used in estimation (6.8). *If* the instruments are valid, (6.8) can be considered more reliable.

However, without insights regarding the validity of the instruments it is not sensible to make such a statement. This stresses why checking instrument validity is essential. Chapter 6.3 briefly discusses guidelines in checking instrument validity and presents approaches that allow to test for instrument relevance and exogeneity under certain conditions. These are then used in an application to the demand for cigarettes in Chapter 6.4.

6.3 Checking Instrument Validity

6.3.1 Instrument Relevance

Instruments that explain little variation in the endogenous regressor X_i are called *weak instruments*. Weak instruments provide little information about the variation in X_i that is exploited by IV regression to estimate the effect of interest: the estimate of the coefficient on the endogenous regressor is estimated inaccurately. Moreover, weak instruments cause the distribution of the estimator to deviate considerably from a normal distribution even in large samples such that the usual methods for obtaining inference about the true coefficient on X_i may produce wrong results.

A Rule of Thumb for Checking for Weak Instruments: Consider the case of a single endogenous regressor X_i and m instruments $Z_{1i}, \dots, Z_{mi}$. If the coefficients on all instruments in the population first-stage regression of a TSLS estimation are zero, the instruments do not explain any of the variation in the X_i which clearly violates assumption that instruments must be relevant. Although the latter case is unlikely to be encountered in practice, we should ask ourselves to what extent the assumption of instrument relevance should be fulfilled. While this is hard to answer for general IV regression, in the case of a *single* endogenous regressor X_i one may use **the following rule of thumb**: Compute the F-statistic which corresponds to the hypothesis that the coefficients on $Z_{1i}, \dots, Z_{mi}$ are all zero in the first-stage regression. If the F-statistic is less than 10, the instruments are "weak" such that the TSLS estimate of the coefficient on X_i is probably biased and no valid statistical inference about its true value can be made.

This rule of thumb is easily implemented in R. Run the first-stage regression using `lm()` and subsequently compute the heteroskedasticity-robust F-statistic by means of `linearHypothesis()`. This is part of the application to the demand for cigarettes discussed in Chapter 6.4.

If Instruments are Weak

There are two ways to proceed if instruments are weak:

1. Discard the weak instruments and/or find stronger instruments. While the former is only an option if the unknown coefficients remain identified when the weak instruments are discarded, the latter can be very difficult and even may require a redesign of the whole study.
2. Stick with the weak instruments but use methods that improve upon TSLS in this scenario, for example limited information maximum likelihood estimation. (Out of the scope of this course.)
3. Use tests that allow for inferences robust to weak instruments: **Anderson-Rubin test**

6.3.2 Instrument Validity

If there is correlation between an instrument and the error term, IV regression is not consistent. The overidentifying restrictions test (also called the J-test) is an approach to test the hypothesis that **additional** instruments are exogenous. For the J-test to be applicable there need to be more instruments than endogenous regressors.

The J-Statistic (or Sargan-Hansen test) Take $\hat{\varepsilon}_i^{TSLS}$, $i = 1, \ldots, n$, the residuals of the TSLS estimation of the general IV regression model (6.5). Run the OLS regression

$$\hat{\varepsilon}_i^{TSLS} = \delta_0 + \delta_1 Z_{1i} + \cdots + \delta_m Z_{mi} + \delta_{m+1} W_{1i} + \cdots + \delta_{m+r} W_{ri} + e_i \qquad (6.9)$$

and test the joint hypothesis

$$H_0 : \delta_1 = \ldots \delta_m = 0$$

which states that all instruments are exogenous. This can be done using the corresponding F-statistic by computing

$$J = mF.$$

This test is the overidentifying restrictions test and the statistic is called the J-statistic with

$$J \xrightarrow{H_0}_d \chi^2_{m-(K-1)} \quad \text{as} \quad n \to \infty$$

under the **assumption of homoskedasticity**. The degrees of freedom $m - (K - 1)$ state the degree of overidentification since this is the number of instruments m minus the number of endogenous regressors $K - 1$.

It is important to note that the J-statistic is only $\chi^2_{m-(K-1)}$ distributed when the error term ε_i in the regression (6.9) is homoskedastic. A discussion of the heteroskedasticity-robust J-statistic is beyond the scope of this chapter. The application in the next section shows how to apply the J-test using `linearHypothesis()`.

6.4 Application to the Demand for Cigarettes

Are the general sales tax and the cigarette-specific tax valid instruments? If not, TSLS is not helpful to estimate the demand elasticity for cigarettes discussed in Chapter 6.2. As discussed in Chapter 6.1, both variables are likely to be relevant but whether they are exogenous is a different question.

One can argue that cigarette-specific taxes could be endogenous because there might be state specific historical factors like economic importance of the tobacco farming and cigarette production industry that lobby for low cigarette specific taxes. Since it is plausible that tobacco growing states have higher rates of smoking than others, this would lead to endogeneity of

cigarette specific taxes. If we had data on the size on the tobacco and cigarette industry, we could solve this potential issue by including the information in the regression. Unfortunately, this is not the case.

However, since the role of the tobacco and cigarette industry is a factor that can be assumed to differ across states but not over time we may exploit the panel structure of `CigarettesSW`. Alternatively, a (non-panel) regression using data on *changes* between two time periods eliminates such state specific and time invariant effects. Next, we consider such changes in variables between 1985 and 1995. That is, we are interested in estimating the *long-run elasticity* of the demand for cigarettes.

The model to be estimated by TSLS using the general sales tax and the cigarette-specific sales tax as instruments hence is

$$\begin{aligned}\log(Q_{i,1995}^{cigarettes}) - \log(Q_{i,1985}^{cigarettes}) = \beta_1 + \beta_2 \left[\log(P_{i,1995}^{cigarettes}) - \log(P_{i,1985}^{cigarettes})\right] \\ + \beta_3 \left[\log(income_{i,1995}) - \log(income_{i,1985})\right] + \varepsilon_i \,.\end{aligned} \tag{6.10}$$

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 - c198

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

We now perform three different IV estimations of (6.10) using `ivreg()`:

1. TSLS using only the difference in the sales taxes between 1985 and 1995 as the instrument.

2. TSLS using only the difference in the cigarette-specific sales taxes 1985 and 1995 as the 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)
```

As usual we use `coeftest()` in conjunction with `vcovHC()` to obtain robust coefficient summaries for all models.

```
# 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

# 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 proceed by generating a tabulated summary of the estimation results using `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,
  header = FALSE,
  type = "latex",
  omit.table.layout = "n",
  digits = 3,
  column.labels = c("IV: salestax", "IV: cigtax",
                    "IVs: salestax, cigtax"),
  dep.var.labels.include = FALSE,
  dep.var.caption =
"Dependent Variable: 1985-1995 Difference in Log per Pack Price",
  se = rob_se)
```

Table 6.1: TSLS Estimates of the Long-Term Elasticity of the Demand for Cigarettes using Panel Data

	Dep. variable: 1985-95 diff in log price/pack		
	IV: salestax	IV: cigtax	IVs: salestax, cigtax
	(1)	(2)	(3)
pricediff	-0.938^{***}	-1.343^{***}	-1.202^{***}
	(0.208)	(0.229)	(0.197)
incomediff	0.526	0.428	0.462
	(0.339)	(0.299)	(0.309)
Constant	-0.118^{*}	-0.017	-0.052
	(0.068)	(0.067)	(0.062)
Observations	48	48	48
R^2	0.550	0.520	0.547
Adjusted R^2	0.530	0.498	0.526
Residual Std. Error (df = 45)	0.091	0.094	0.091

Table 6.1 reports negative estimates of the coefficient on `pricediff` that are quite different in magnitude. Which one should we trust? This hinges on the validity of the instruments used. To assess this we compute F-statistics for the first-stage regressions of all three models to check instrument relevance.

```
# 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
```

```
# 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
```

All F-statistics are larger than 10; so, the rule of thumb for detecting weak instruments would suggest that the instruments are not weak.

Next, we also conduct the overidentifying restrictions test for model three which is the only model where the coefficient on the difference in log prices is overidentified ($m = 2$, $(K-1) = 1$) such that the J-statistic can be computed. To do this we take the residuals stored in `cig_ivreg_diff3` and regress them on both instruments and the presumably exogenous regressor `incomediff`. We again use `linearHypothesis()` to test whether the coefficients on both instruments are zero which is necessary for the exogeneity assumption to be fulfilled. Note that with `test = "Chisq"` we 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 + c
#>
```

```
#>   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: In this case the p-value reported by `linearHypothesis()` is wrong because the degrees of freedom are set to 2. This differs from the degree of overidentification ($m - (K - 1) = 2 - (2 - 1) = 1$) so the J-statistic is χ^2_1 distributed instead of following a χ^2_2 distribution as assumed defaultly by `linearHypothesis()`. We may compute the correct p-value using `pchisq()`.

```
# compute correct p-value for J-statistic
pchisq(cig_OR_test[2, 5], df = 1, lower.tail = FALSE)
#> [1] 0.02636406
```

Since this value is smaller than 0.05 we reject the hypothesis that both instruments are exogenous at the level of 5%. This means one of the following:

1. The sales tax is an invalid instrument for the per-pack price.
2. The cigarettes-specific sales tax is an invalid instrument for the per-pack price.
3. Both instruments are invalid.

One can argue that the assumption of instrument exogeneity is more likely to hold for the general sales tax such that the IV estimate of the long-run elasticity of demand for cigarettes we consider the most trustworthy is -0.94, the TSLS estimate obtained using the general sales tax as the only instrument.

The interpretation of this estimate is that over a 10-year period, an increase in the average price per package by one percent is expected to decrease

consumption by about 0.94 percentage points. This suggests that, in the long run, price increases can reduce cigarette consumption considerably.

Bibliography

Cribari-Neto, F. (2004). Asymptotic inference under heteroskedasticity of unknown form. *Computational Statistics & Data Analysis*, 45(2):215–233.

Hanck, C., Arnold, M., Gerber, A., and Schmelzer, M. (2021). Introduction to econometrics with R.

Hayashi, F. (2000). *Econometrics*. Princeton University Press.

Kleiber, C. and Zeileis, A. (2008). *Applied Econometrics with R*. Springer.

Lindsay, B. G. and Basak, P. (2000). Moments determine the tail of a distribution (but not much else). *The American Statistician*, 54(4):248–251.

Long, J. S. and Ervin, L. H. (2000). Using heteroscedasticity consistent standard errors in the linear regression model. *The American Statistician*, 54(3):217–224.

MacKinnon, J. G. and White, H. (1985). Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics*, 29(3):305–325.

White, H. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, pages 817–838.