TS Turing Students Rotterdam

Coding Hours

Introduction to R

Mathijs de Jong

Outline of the Presentation

TS Turing Students Rotterdam

What is R?

Turing Students Rotterdam

- Programming language & free software environment
- Statistical computing & graphics
- First appeared: August 1993
- Written in C, Fortran and R itself

What can R be used for?

TS Turing Students Rotterdam

- Data manipulation
- Data visualization
- Analyzing data
  - Advanced statistical methods
  - Machine learning
- And much more!

Why you should learn R for data science?

TS Turing Students Rotterdam

- R is easy to learn
- Very powerful
- Popular in academia
- Excellent package support
- Open source

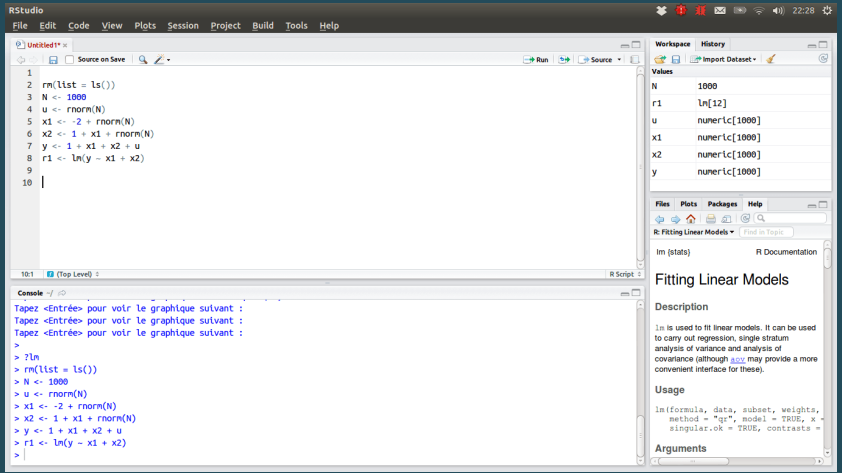
RStudio

Turing Students Rotterdam

- Download RStudio or use RStudio Cloud
- Available on all computers at the EUR
- Preferably work on a Windows computer

RStudio

Turing Students Rotterdam



Learning R with DataCamp

Turing Students Rotterdam

- DataCamp is a website that offers courses for learning R, Python and SQL
- Complete the Introduction to R tutorial

Continuing the Journey with r-statistics

TS Turing Students Rotterdam

- r-statistics is an educational resource for those seeking knowledge related to machine learning and statistical computing in R.
- Read the R Tutorial on r-statistics.
- Continue with the articles about Statistical Tests and Linear Regressions in R

A New Way of Testing

TS Turing Students Rotterdam

1. Import the MASS package
2. Find the maximum price of all cars in the Cars93 data in the MASS package.
3. Check whether the mean of the horsepower of the Cars93 data significantly different from 140.
4. Do the same as for the two questions above for the data on the cars with airbags.
5. Do the same as for the last three questions but now for the cars with more than 4 cylinders that cost less than 20.

Your First Linear Regression in R

TS Turing Students Rotterdam

- Consider the *Boston* data from the MASS package.
- Scatter plot the following columns:
  - crim
  - zn
  - indus
  - age
- Regress crim on zn, indus and age.
- Add a dummy variable for dis larger than 5 in the regression above.
- Plot the results of your regression together with crim in one figure.

Reinvent OLS in R (1/3)

TS Turing Students Rotterdam

1. Write a *Backsubstition*$(\mathbf{A}, \mathbf{b})$ that takes a $m \times n$ matrix $\mathbf{A}$ of full rank and a $m \times 1$ vector $\mathbf{b}$ and returns the unique $n \times 1$ vector $\mathbf{x}$ which solves $\mathbf{Ax} = \mathbf{b}$ using backsubstitution. As a hint, the way to solve it is

$$x_n = b_n / a_{nn}, \qquad x_i = \left(b_i - \sum_{j>i} a_{ij} x_j \right) / a_{ii}, \qquad i \in \{1, 2, \ldots, n-1\}.$$

   It might be easier to first define

$$s = \sum_{j>i} a_{ij} x_j,$$

   and for all $x_n, x_{n-1}, \ldots, x_1$ use the same formula for solving.

Reinvent OLS in R (2/3)

Turing Students Rotterdam

1. Generate 20 observations from $y = \mathbf{X}\beta + \sigma\varepsilon$ with $\beta = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$, $\mathbf{X} = [1\, u_1\, u_2]$, $u_i \sim \mathcal{N}(0, 1)$ for $i \in \{1, 2\}$, $\varepsilon \in \mathcal{N}(0, 1)$, $\sigma = 0.25$.
2. Estimate the OLS estimator $\hat{\beta}$ using
   - 2.1 using direct matrix multiplication;
   - 2.2 using your elimination + backsubstitution, noting that

$$\mathbf{b} \equiv \mathbf{X}'\mathbf{y} = \mathbf{X}'\mathbf{X}\hat{\beta} \equiv \mathbf{A}\mathbf{x};$$

   - 2.3 using a prepackaged function.

   Write three functions *EstimateMM*(), *EstimateEB*(), *EstimatePF*(), which implement the three options, and check that the results indeed are the same.

Reinvent OLS in R (3/3)

TS Turing Students Rotterdam

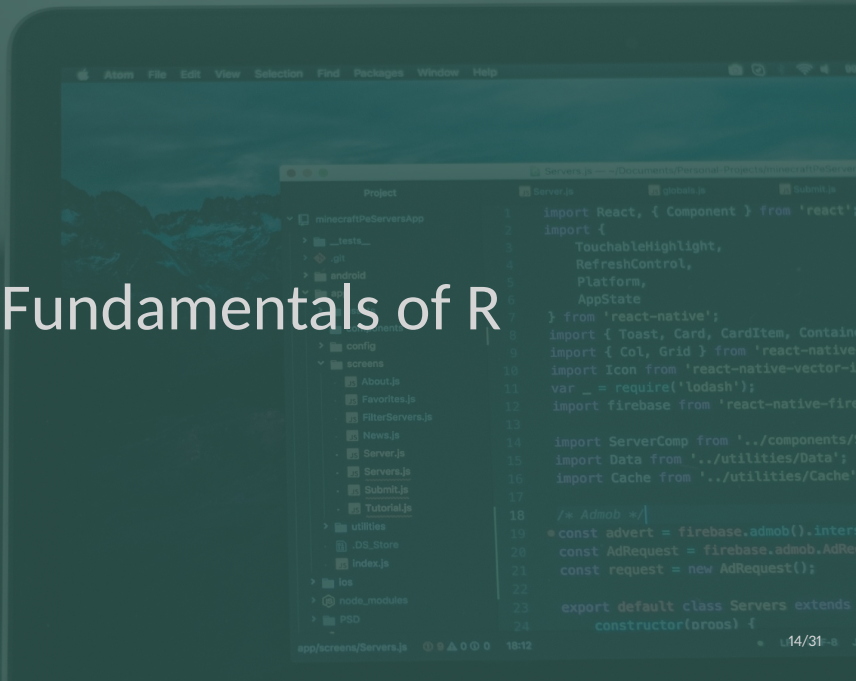
3. Eventually, we might also be interested in

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\hat{\beta}, \qquad \hat{\sigma}^2 = \frac{1}{n-k}\mathbf{e}'\mathbf{e} = \frac{1}{n-k}\sum_{i=1}^{n} e_i^2,$$

$$\hat{\Sigma} = \hat{\sigma^2}(\mathbf{X}'\mathbf{X})^{-1}, \qquad s\left(\hat{\beta}\right) = \operatorname{diag}\left(\hat{\Sigma}\right)^{1/2},$$

with n and k the size of $\mathbf{y}$ and β, respectively. Also the t-statistics, $t = \hat{\beta}_i / s\left(\hat{\beta}_i\right)$, could be of interest. Build a version of your program which also computes $s\left(\hat{\beta}_i\right)$ and the t-value, and outputs this together with $\hat{\beta}$.

Fundamentals of R



Creating variables

Turing Students Rotterdam

```
a <- 10 # assign 10 to 'a'
a = 10  # same as above
10 -> a # assign 10 to 'a'
10 = a  # Wrong!
```

Classes or Data types

TS Turing Students Rotterdam

- Variable types:
  - **character** – String
  - **integer** – Integers
  - **numeric** – Integers + Fractions
  - **logical** – Boolean
- Data types:
  - **vector** – A collection of elements of the same class
  - **matrix** – Simple matrix
  - **data.frame** – Complex table
  - **list** – A collection of elements of different classes

Classes or Data types

Turing Students Rotterdam

```r
a <- 10
class(a)    # numeric
a <- as.character(a)
print(a)    # ???
class(a)    # character
```

What is a R package?

TS Turing Students Rotterdam

- Packages are collections of
  - R functions
  - R data
  - compiled code
- Can be created by anyone
- Academics often create their own packages

Adding packages

Turing Students Rotterdam

- Using the RStudio GUI
- From the R console:

```
install.packages("car")  # install 'car' package
library(car)    # initialize the 'car' package
require(car)    # another way to initialize
```

Getting help

Turing Students Rotterdam

- Build-in support
- Google
- Source code

```
help(merge)  # get help page for 'merge'
?merge  # lookup 'merge' from installed pkgs
??merge  # vague search
example(merge)  # show code examples
help
```

Importing and exporting data

Turing Students Rotterdam

- RStudio GUI
- Using the R console:

```
library(readxl) # initialize the required library
mydata <- read_excel("~/Downloads/mydata.xlsx") # import an Excel file
library(xlsx) # initialize the required library
write.xlsx(mydata, "~/Downloads/mydata.xlsx") # export a variable as an
    Excel file
```

How to create a vector?

Turing Students Rotterdam

- Combine function `c()`
- Only data of one type

```
vec1 <- c(1, 2, 3) # numeric vector
vec2 <- c("a", "b", "c") # character vector
vec3 <- c(1, "a") # error!
```

How to reference elements of a vector?

TS Turing Students Rotterdam

- Elements of a vector can be accessed with its index
- First element has index 1
- Last element has index *length*(*vectorName*)

```
length(vec1)    # 4
print(vec1)     # 1 2 3
print(vec2[3]   # "c"
print(vec2[2:3] # "b" "c"
```

How to manipulate vectors

Turing Students Rotterdam

- Subsetting

```
logic1 <- vec1 < 15  # create a logical vector, TRUE if value < 15
vec1[logic1]  # elements in TRUE positions will be included in subset
vec1[1:2]  # returns elements in 1 & 2 positions.
vec1[c(1,3)]  # returns elements in 1 & 3 positions
vec1[-1]  # returns all elements except in position 1.
```

- Sorting

```
sort(vec1)  # ascending sort
sort(vec1, decreasing = TRUE)  # Descending sort
```

How to create a data frame?

Turing Students Rotterdam

- Data frames are used for storing data tables

```
myDf1 <- data.frame(vec1, vec2)  # make data frame with 2 columns
myDf2 <- data.frame(vec1, vec3, vec4)
myDf3 <- data.frame(vec1, vec2, vec3)
```

- Working with build-in datasets

```
library(datasets) # initialize
library(help=datasets) # display the datasets
```

Basic operations for a data frame

Turing Students Rotterdam

```r
class(airquality)  # get class
sapply(airquality, class)  # get class of all columns
str(airquality)  # structure
summary(airquality)  # summary of airquality
head(airquality)  # view the first 6 obs
fix(airquality)  # view spreadsheet like grid
rownames(airquality)  # row names
colnames(airquality)  # columns names
nrow(airquality)  # number of rows
ncol(airquality)  # number of columns
```

Basic operations for a data frame

Turing Students Rotterdam

```
cbind(myDf1, myDf2)  # columns append DFs with same no. rows
rbind(myDf1, myDf1)  # row append DFs with same no. columns
myDf1$vec1  # vec1 column
myDf1[, 1]  # df[row.num, col.num]
myDf1[, c(1,2)]  # columns 1 and 3
myDf1[c(1:5), c(2)]  # first 5 rows in column 2
```

If-Else condition

Turing Students Rotterdam

```
if(checkConditionIfTrue) {
  ....statements..
  ....statements..
} else {   # place the 'else' in same line as '}'
  ....statements..
  ....statements..
}
```

Loops in R

- Standard for-loop:

```
for(counterVar in c(1:n)){
    .... statements..
}
```

- The apply family
  - **apply()**: Apply FUN through a data frame or matrix by rows or columns.

```
myData <- matrix(seq(1,16), 4, 4)  # make a matrix
apply(myData, 1, FUN=min)  # apply 'min' by rows
# => [1] 1 2 3 4

apply(myData, 2, FUN=min)  # apply 'min' by columns
# => [1] 4 8 12 16
```

Functions in R

Turing Students Rotterdam

- Reusable block of code

```
myfunction <- function(arg1, arg2, ... ){
    .... statements
    return(object)
}
```

Useful Resources

TS Turing Students Rotterdam

- DataCamp R
- Learn R with Swirl
- R-statistics Tutorial
- Statmethods Tutorial
- Linear Regression