

Applied Statistics in R

Elena Parilina

Master's Program

Game Theory and Operations Research

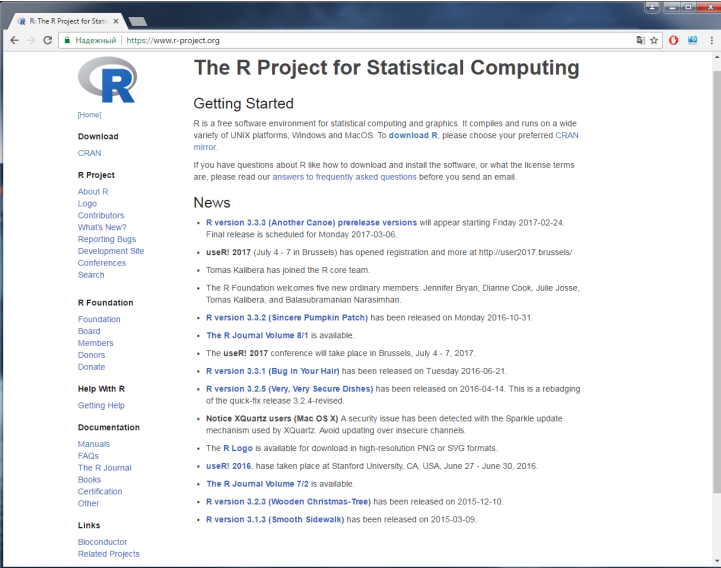
Saint Petersburg State University

2020

References

- ① <https://www.r-project.org/>
- ② Adler J., “R in a Nutshell: A Desktop Quick Reference”, O'Reilly Media, 2010, 640 p.
- ③ Crawley M.J., “The R Book”, Willey, 2007, 950 p.
- ④ Marques de Sa, Joaquim P., “Applied Statistics Using SPSS, STATISTICA, MATLAB and R”, Springer, 2007, 520 p.
- ⑤ James G., Witten D., Hastie T., Tibshirani R., “An Introduction to Statistical Learning with Applications in R”, Springer, 2013, 426 p.
- ⑥ Greene W.H., “Econometric Analysis”, Prentice Hall, 7th edition, 2011, 1188 p.
- ⑦ Maddala G.S., “Introduction to Econometrics”, Macmillan Publishing Company, 2nd edition, 2007, 637 p.

R Project: <http://www.r-project.org/>



The screenshot shows the official website of the R Project for Statistical Computing. The browser window has a title bar that says "R: The R Project for Statistical Computing". The address bar shows the URL "https://www.r-project.org/". The website features a large blue 'R' logo at the top left. Below the logo is a navigation menu with links such as [Home], Download, CRAN, R Project, About R, Logo, Contributors, What's New?, Reporting Bugs, Development Site, Conferences, and Search. The main content area is titled "The R Project for Statistical Computing" and "Getting Started". It provides information about R as a free software environment for statistical computing and graphics, and lists various news items, including the release of R version 3.3.3 (Another Canoe) prerelease versions, the user! 2017 conference, and the release of R version 3.3.2 (Sincere Pumpkin Patch).

The R Project for Statistical Computing

Getting Started

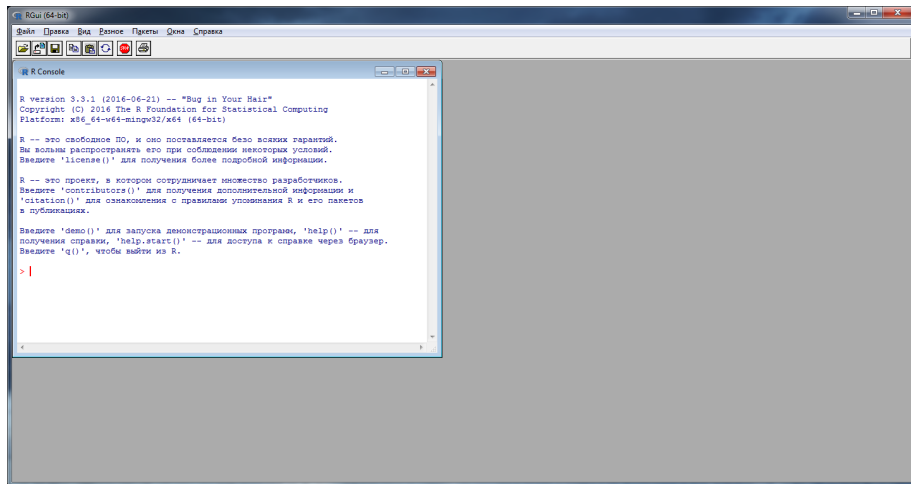
R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To [download R](#), please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

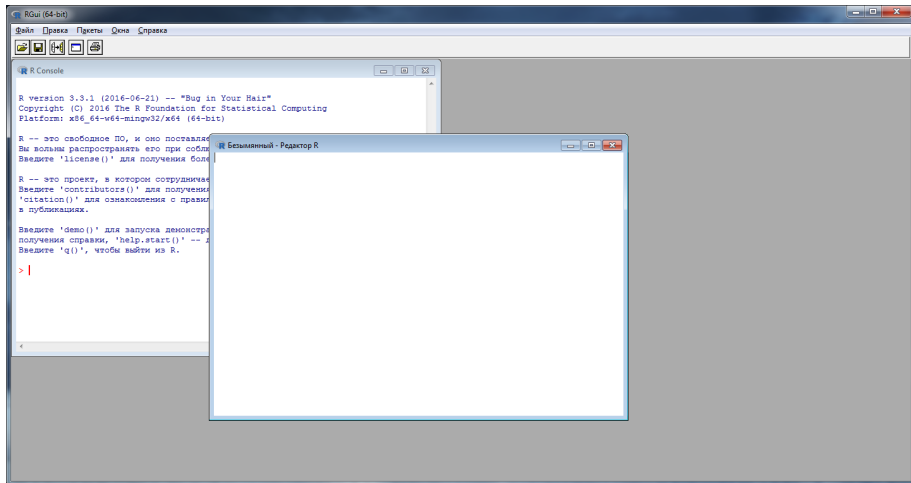
News

- **R version 3.3.3 (Another Canoe) prerelease versions** will appear starting Friday 2017-02-24. Final release is scheduled for Monday 2017-03-06.
- **user! 2017** (July 4 - 7 in Brussels) has opened registration and more at <http://user2017.brussels/>
- Tomas Kalibera has joined the R core team.
- The R Foundation welcomes five new ordinary members: Jennifer Bryan, Dianne Cook, Julie Josse, Tomas Kalibera, and Balasubramanian Narasimhan.
- **R version 3.3.2 (Sincere Pumpkin Patch)** has been released on Monday 2016-10-31.
- **The R Journal Volume 8/1** is available.
- The **user! 2017** conference will take place in Brussels, July 4 - 7, 2017.
- **R version 3.3.1 (Bug in Your Hair)** has been released on Tuesday 2016-06-21.
- **R version 3.2.5 (Very, Very Secure Dishes)** has been released on 2016-04-14. This is a rebadging of the quick-fix release 3.2.4-revised.
- **Notice XQuartz users (Mac OS X)** A security issue has been detected with the Sparkle update mechanism used by XQuartz. Avoid updating over insecure channels.
- The **R Logo** is available for download in high-resolution PNG or SVG formats.
- **user! 2016**, have taken place at Stanford University, CA, USA, June 27 - June 30, 2016.
- **The R Journal Volume 7/2** is available.
- **R version 3.2.3 (Wooden Christmas-Tree)** has been released on 2015-12-10.
- **R version 3.1.3 (Smooth Sidewalk)** has been released on 2015-03-09.

RGui

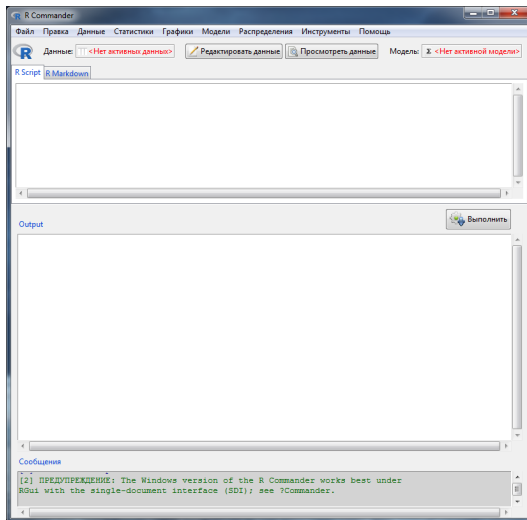


New R script

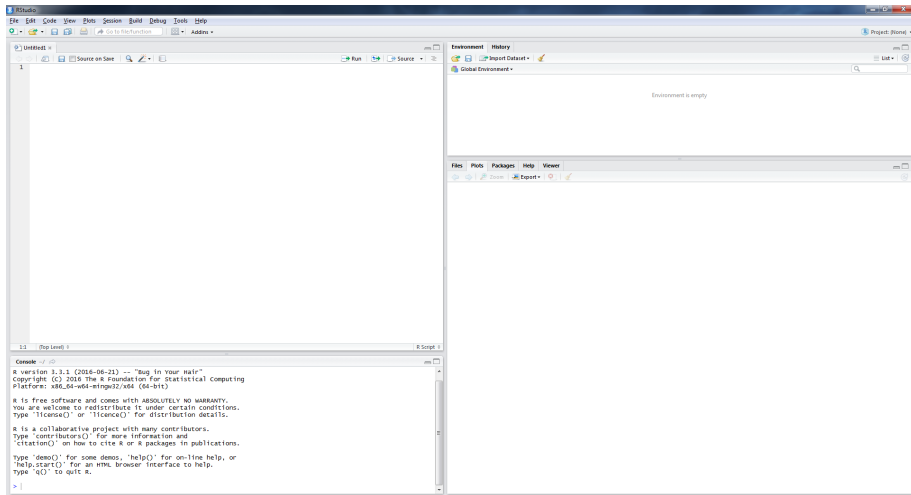


R Commander

```
library(Rcmdr);
```



RStudio: <https://www.rstudio.com>



R packages

- To load a package in R:
`> library()`
- To install a package in R:
`> install.packages()`
- To install several packages in R:
`> install.packages(c("tree", "maptree"))`

Basic operations in R

- To combine a vector in R:

```
> c(1,2,3,4)
[1] 1 2 3 4
```

- Operators with vectors:

```
> c(1,2,3,4)+c(10,20,30,40)
[1] 11 22 33 44
```

```
> c(1,2,3,4)*c(10,20,30,40)
[1] 10 40 90 160
```

```
> 1/c(1,2,3,4)
[1] 1.000 0.500 0.333 0.250
```

```
> c(1,2,3,4)+c(10,100)
[1] 11 102 13 104
```

Variables

- To assign a value to a variable:

```
> x <- 1  
> y <- 2  
> z <- x+y
```

- To refer to a member of a vector:

```
> b <- c(1,2,3,4,5,6,7,8,9,10)  
> b[7]  
[1] 7  
  
> b[1:6]  
[1] 1 2 3 4 5 6
```

Lists in R

- A list is a generic vector containing other objects. For example, the following variable *x* is a list containing copies of three vectors *n*, *s*, *b*, and a numeric value 3:

```
> n = c(2, 3, 5)
> s = c("aa", "bb", "cc", "dd", "ee")
> b = c(TRUE, FALSE, TRUE, FALSE, FALSE)
> x = list(n, s, b, 3) # x contains copies of n, s, b
```

- To access an item in the list:

```
> x[2]
[1] "aa" "bb" "cc" "dd" "ee"
```

Lists in R

- In order to reference a list member directly, use the double square bracket "[[]]" operator. The following object `x[[2]]` is the second member of `x`. In other words, `x[[2]]` is a copy of `s`, but is not a slice containing `s` or its copy:

```
> x[[2]]  
[1] "aa" "bb" "cc" "dd" "ee"
```

Named lists in R

- ```
> v = list(bob=c(2, 3, 5), john=c("aa", "bb"))
> v
$bob
[1] 2 3 5
$john
[1] "aa" "bb"
```
- In order to reference a list member directly, use the double square bracket "[[]]" operator. The following references a member of `v` by name:  

```
> v[["bob"]]
[1] 2 3 5
```
- A named list member can also be referenced directly with the "\$" operator in lieu of the double square bracket operator:  

```
> v$bob
[1] 2 3 5
```

## Data frame

Data frame is a list that contains multiple named vectors that are the same length. Data frame is a lot like a database table. For example, there is a data frame with win/loss results in the National League (NL) East in 2008:

```
> teams <- c("PHI","NYM","FLA","ATL","WSN")
> w <- c(92, 89, 94, 72, 59)
> l <- c(70, 73, 77, 90, 102)
> nleast <- data.frame(teams,w,l)
> nleast
```

|   | teams | w  | l   |
|---|-------|----|-----|
| 1 | PHI   | 92 | 70  |
| 2 | NYM   | 89 | 73  |
| 3 | FLA   | 94 | 77  |
| 4 | ATL   | 72 | 90  |
| 5 | WSN   | 59 | 102 |

## Data frame

- To refer the components of a data frame use "\$" operator:

```
> nleast$w
[1] 92 89 94 72 59
```

- Suppose you want to find the number of losses by Florida Marlins (FLA). You can calculate it like this:

```
> nleast$teams=="FLA"
[1] FALSE FALSE TRUE FALSE FALSE
```

- Then you can use this vector to refer to the right element in the losses vector:

```
> nleast$l[nleast$teams=="FLA"]
[1] 77
```

# Data input

- XLS-file:

```
library(gdata)
data <- read.xls("C:/noname.xls", sheet = 1, header =
TRUE)
```

- XLSX-file:

```
library(readxl)
data <- read_excel("C:/haemolytic.xlsx", sheet = 1,
col_names = TRUE)
```

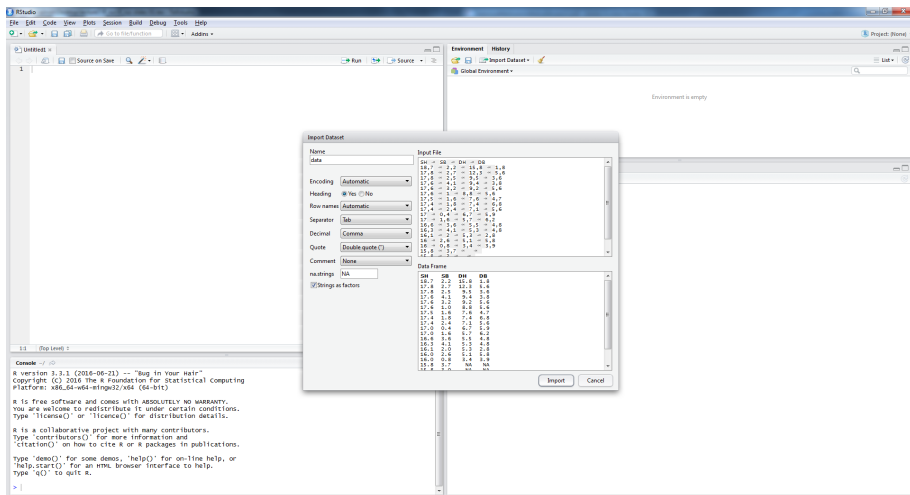
- TXT-file:

```
data <- read.table("C:/noname.txt", header=TRUE,
sep="\t", na.strings="NA", dec=",")
```



# Data input (RStudio)

Tools → Import Dataset → From Local File...



# Data input (RStudio)

The screenshot displays the RStudio environment with the following components:

- Source Editor:** Contains a script with the following R code:

```
> data <- read.delim2("c:/users/admin/desktop/lecture1/R_rus/haenolytic.txt")
> view(data)
```
- Environment:** Shows the 'data' object with 63 observations and 4 variables.
- Console:** Displays the R startup message and the execution of the code in the source editor.
- Data Viewer:** Shows a preview of the data loaded into the 'data' object, displaying columns SN, SB, DH, and DB.

| SN | SB   | DH  | DB   |
|----|------|-----|------|
| 1  | 18.7 | 2.2 | 15.8 |
| 2  | 17.8 | 2.7 | 12.3 |
| 3  | 17.8 | 2.5 | 9.5  |
| 4  | 17.8 | 4.1 | 9.4  |
| 5  | 17.8 | 3.2 | 9.2  |
| 6  | 17.8 | 1.0 | 8.8  |
| 7  | 17.5 | 1.6 | 7.6  |
| 8  | 17.4 | 1.8 | 7.4  |
| 9  | 17.4 | 2.4 | 7.1  |
| 10 | 17.0 | 0.4 | 6.7  |
| 11 | 17.0 | 1.6 | 5.7  |
| 12 | 16.8 | 3.6 | 5.5  |
| 13 | 16.3 | 4.1 | 5.3  |
| 14 | 16.1 | 2.0 | 5.3  |
| 15 | 16.0 | 2.8 | 5.1  |
| 16 | 16.0 | 0.8 | 5.4  |
| 17 | 15.8 | 3.7 | NA   |
| 18 | 15.8 | 2.0 | NA   |
| 19 | 15.8 | 1.7 | NA   |
| 20 | 15.8 | 1.4 | NA   |
| 21 | 15.8 | 2.0 | NA   |
| 22 | 15.8 | 1.6 | NA   |
| 23 | 15.4 | 4.1 | NA   |
| 24 | 15.4 | 2.2 | NA   |

## Missing values: NA

To check if there are NA entries in data:

```
is.na(data$DH)
```

To produce a vector without NA entries:

```
data$DH[! is.na(data$DH)]
```

# Vectors

```
x <- c(1,3,0,2,1,4,2,1,5,0)
```

```
x <- seq(-4,4,0.01)
```

```
length(x)
```

```
table(x)
```

```
summary(x)
```

```
x[4]
```

```
x[c(2,3,6)]
```

```
x[1:3]
```

```
x[-1]
```

```
x[-length(x)]
```

- `max(x)`: maximum value in `x`
- `min(x)`: minimum value in `x`
- `sum(x)`: total of all the values in `x`
- `mean(x)`: arithmetic average of the values in `x`
- `median(x)`: median value in `x`
- `range(x)`: vector of `min(x)` and `max(x)`
- `var(x)`: sample variance of `x`
- `sd`: sample standard deviation of `x`
- `cor(x,y)`: correlation between vectors `x` and `y`
- `sort(x)`: a sorted version of `x`
- `rank(x)`: vector of the ranks of the values in `x`
- `quantile(x)`: vector containing the minimum, lower quartile, median, upper quartile, and maximum of `x`
- `cumsum(x)`: vector containing the sum of all of the elements up to that point
- `cumprod(x)`: vector containing the product of all of the elements up to that point

# Generating random samples

- `rnorm(n,m,sd);`
- `runif(n,a,b);`
- `rweibull(n,shape,scale);`
- `rpois(n,lambda);`
- `rgamma(n,shape,scale);`
- `rbinom(n,size,prob);`
- `rchisq(n,df);`
- `rexp(n,rate);`
- `rf(n,df1,df2);`
- `rt(n,df).`

# PDF, CDF, quantiles

PDF: replace r with d

```
curve(dnorm(x,m=10,sd=2),from=0,to=20,main="Probability
density function N(10,4)")
```

CDF: replace r with p

```
curve(pnorm(x,m=10,sd=2),from=0,to=20,main="Cumulative
distribution function N(10,4)")
```

Quantiles: replace r with q.

# Plot

`hist(x)`: histogram of `x`;

`boxplot(x)`: boxplot of `x`;

`ecdf(x)`: empirical cdf of `x`:

`plot(ecdf(x))`

`plot(x, y)`: `x`-`y` plotting

`plot(ecdf(x), lty=1)`

`lines(ecdf(rnorm(10000, 0, .5))), lty=2)`

`lines`: add connected line segments to a plot

`points(x)`: add points to a plot

`curve(pnorm(x, 0, 1), -4, 4)`

`curve(pnorm(x, 0, 0.5), add=TRUE)`

`curve(expr)`: draw function plots



## Data for seminar 1. Dataset “Babyboom”

- NAME: Time of Birth, Sex, and Birth Weight of 44 Babies
- TYPE: Observational
- SIZE: 44 observations, 4 variables

### **DESCRIPTIVE ABSTRACT:**

The dataset contains the time of birth, sex, and birth weight for each of 44 babies born in one 24-hour period at a Brisbane, Australia, hospital. Also included is the number of minutes since midnight for each birth.

### **SOURCE:**

The data appeared in the Brisbane newspaper “The Sunday Mail” on December 21, 1997.

### **REFERENCE:**

Steele, S. (December 21, 1997), “Babies by the Dozen for Christmas: 24-Hour Baby Boom,” “The Sunday Mail” (Brisbane), p. 7.

# Data for seminar 1. Dataset “Babyboom”

## **VARIABLE DESCRIPTIONS:**

### Columns

1 - 8     Time of birth recorded on the 24-hour clock

9 - 16    Sex of the child (1 = girl, 2 = boy)

17 - 24   Birth weight in grams

25 - 32   Number of minutes after midnight of each birth

Values are aligned and delimited by blanks. There are no missing values.

# Data for seminar 1. Dataset “Babyboom”

## **STORY BEHIND THE DATA:**

Forty-four babies – a new record – were born in one 24-hour period at the Mater Mothers’ Hospital in Brisbane, Queensland, Australia, on December 18, 1997. For each of the 44 babies, “The Sunday Mail” recorded the time of birth, the sex of the child, and the birth weight in grams.

## **PEDAGOGICAL NOTES:**

The data can be used to demonstrate fitting the binomial distribution (the number of boys/girls born out of 44 births), the geometric distribution (the number of births until a boy or girl is born), the Poisson distribution (births per hour for each hour), and the exponential distribution (times between births). The normal distribution is found to be unsuitable for modeling the birth weights, but better results are obtained when birth weights are separated by sex. The dataset can also be used to illustrate hypothesis tests about proportions, comparisons of birth weights by gender, the runs test of randomness of gender, and skewed data.

## Data for seminar 1. Dataset “Airport”

**NAME:** US AIRPORT STATISTICS

**TYPE:** Census (sort of)

**SIZE:** 135 observations, 7 variables (5 numeric, 2 character)

**DESCRIPTIVE ABSTRACT:**

This dataset consists of all 135 large and medium sized air hubs in the United States as defined by the Federal Aviation Administration.

**SOURCE:**

U.S. Federal Aviation Administration and Research and Special Programs Administration, 'Airport Activity Statistics' (1990).

**SPECIAL NOTES:**

These are the only cities provided in this source. Although, it is not a census of all air hubs, it is a census of all medium and large hubs as classified by FAA.

**STORY BEHIND THE DATA:**

Author, who is very interested in maps, geography, and U.S.

## Data for seminar 1. Dataset “Airport”

transportation system entered this as part of a 'U.S. infrastructure series' (along with highway and others) this summer.

### **VARIABLE DESCRIPTIONS:**

|                                  |               |
|----------------------------------|---------------|
| Airport                          | Columns 1-21  |
| City                             | Columns 22-43 |
| Scheduled departures             | Columns 44-49 |
| Performed departures             | Columns 51-56 |
| Enplaned passengers              | Columns 58-65 |
| Enplaned revenue tons of freight | Columns 67-75 |
| Enplaned revenue tons of mail    | Columns 77-85 |

### **PEDAGOGICAL NOTES:**

It does allow a teacher to use some actual data when teaching concepts of descriptive statistics, such as graphical and numeric descriptions of a set of measurements.

## Data for seminar 1. Dataset “Euroweight”

**NAME:** The Weight of Euro Coins

**TYPE:** Fitting distributions to data

**SIZE:** 2000 observations, 3 variables

**DESCRIPTIVE ABSTRACT:**

In many statistical models the normal distribution of the response is an essential assumption. This paper uses a dataset of 2000 euro coins with information (up to the milligram) about the weight of each coin. As the physical coin production process is subject to a multitude of (very small) variability sources, it seems reasonable to expect that the empirical distribution of the weight of euro coins does agree with the normal distribution. Goodness of fit tests however show that this is not the case. Moreover, some outliers complicate the analysis. Mixtures of normal distributions and skew normal distributions are fitted to the data, revealing that the normality assumption might not hold for those weights.

# Data for seminar 1. Dataset “Euroweight”

## **SOURCE:**

The data were collected by Herman Callaert at Hasselt University in Belgium. The euro coins were “borrowed” at a local bank. Two assistants, Sofie Bogaerts and Saskia Litierie weighted the coins one by one, in laboratory conditions on a weighing scale of the type Sartorius BP 310s.

## **VARIABLE DESCRIPTIONS:**

Columns

|               |                                  |
|---------------|----------------------------------|
| 1 - 8 ID      | this is the case number          |
| 9 - 16 weight | weight of the euro coin in grams |
| 17 batch      | number of the package            |

Values are aligned and tab-delimited. There are no missing values

# Data for seminar 1. Dataset “Euroweight”

## **STORY BEHIND THE DATA:**

Curriculum reform in Flanders (the Flemish part of Belgium) resulted in a significant increase of statistical topics in grades 8-12. A group of university professors and high school teachers took this opportunity for reshaping statistics education towards “more concepts and more real data”. In Belgium, as in many countries in Europe, the introduction of the Euro has had a major impact on the life of people. That’s why it was decided to study a characteristic (the weight) of the “Belgian 1 Euro” coin, and use this dataset in schools.

## **PEDAGOGICAL NOTES:**

Graphical methods can be used, perhaps in conjunction with a simple goodness of fit test. Also the story behind the data gathering process, and the lesson to be learned here, is crucial in any real life statistical experiment. A first year college course could delve a bit deeper, while a full analysis (introducing mixtures and the skew normal) can be revealing in a second course in statistics.