Freie Universität Berlin

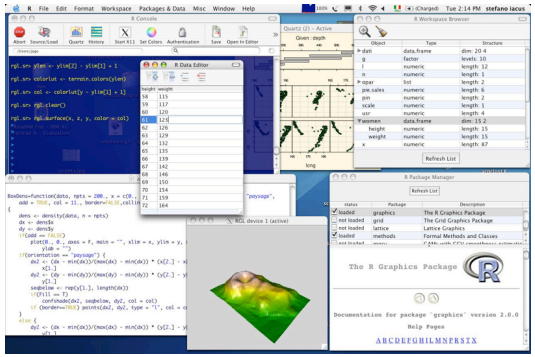
Basics of GNU R -
data types, data manipulations, statistical basics
Introduction into R

This chapter is mainly based on the following text book:
Everitt, B.S., Hothorn, T. (2010) A Handbook of Statistical Analysis using R. 2nd. edition. CRC Press.

What is GNU R?

- Free software environment for statistical computing and graphics
- Website: http://www.r-project.org/
  - Here you will find: R system itself, a collection of add-on packages, manuals, documentation, ...
- GNU project; therefore R is Free Software under the terms of GPL
- Highly extensible through the use of packages for specific functions or specific areas of study
- Over 1,500 user-contributed packages available at R CRAN
- Base distribution already comes with some packages
- Cross-platform application

- Installation is easy and straightforward
- Use manual! *R Installation and Administration* (*http://cran.r-project.org/doc/manuals/R-admin.pdf*)



Using packages in R

- Installing new packages (Internet connection needed)

```
> install.packages("XYZ")
```

- Loading a package

```
> library("XYZ")
```

- Comprehensive list of available packages: http://www.cran.r-project.org/web/packages/

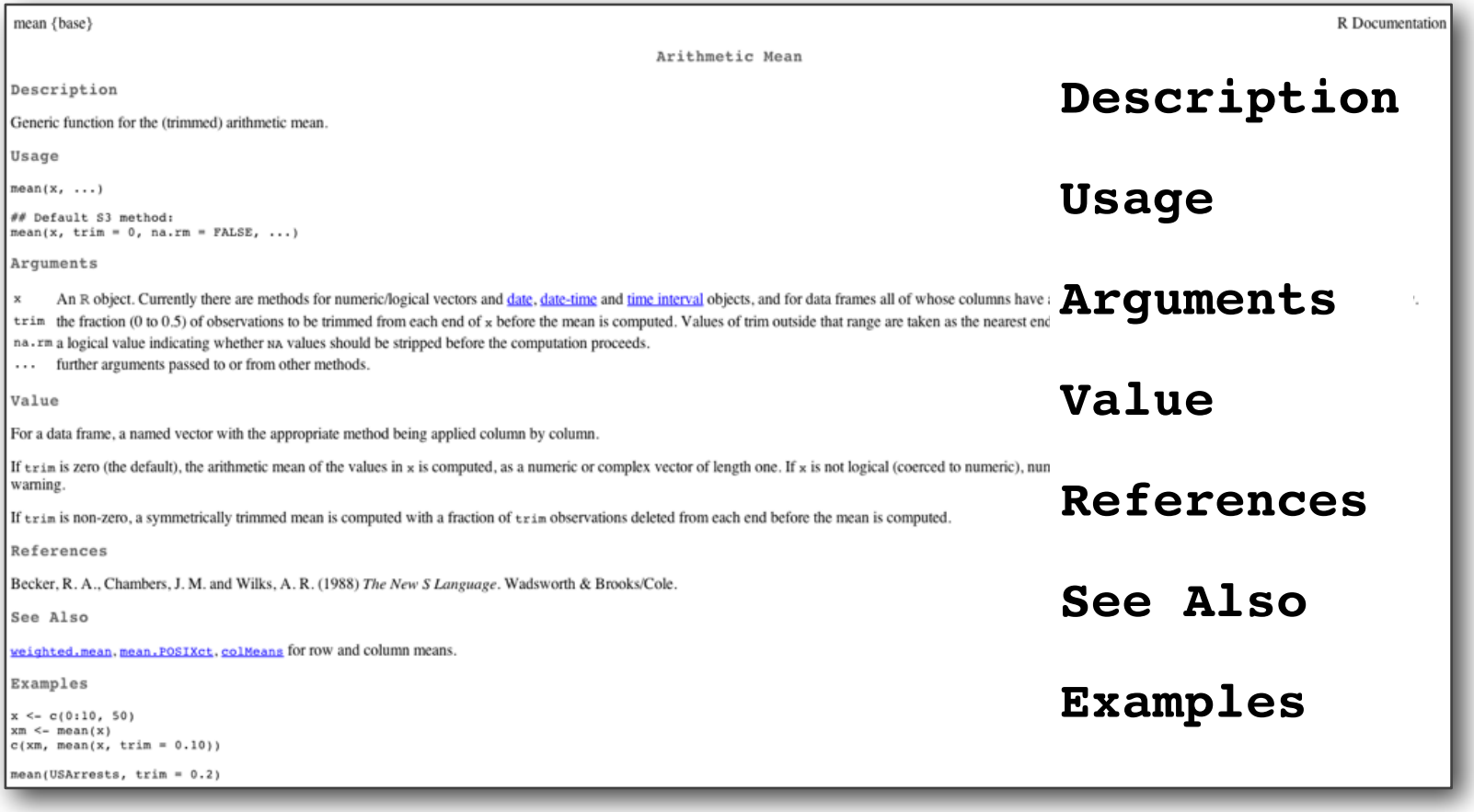
Getting help

- Three different forms of documentation: Online help that comes with the base distribution, electronic manuals, publications such as books or articles (e.g., http://journal.r-project.org)

- Help at command line: `> help("mean")` or `R> ?mean`
- Search through help doc:
```
> help.search("network")
> apropos("network")
> ??("network")
```
- Learn about a library:
```
> help(package=igraph)
> library(help=igraph)
```

- Check out what it can do: `> example(mean)`

Example R command line help



mean {base} R Documentation

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)

## Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)
```

Arguments

x	An R object. Currently there are methods for numeric/logical vectors and date, date-time and time interval objects, and for data frames all of whose columns have
trim	the fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed. Values of trim outside that range are taken as the nearest end
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to or from other methods.

Value

For a data frame, a named vector with the appropriate method being applied column by column.

If trim is zero (the default), the arithmetic mean of the values in x is computed, as a numeric or complex vector of length one. If x is not logical (coerced to numeric), nun warning.

If trim is non-zero, a symmetrically trimmed mean is computed with a fraction of trim observations deleted from each end before the mean is computed.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

See Also

weighted.mean, mean.POSIXct, colMeans for row and column means.

Examples

```
x <- c(0:10, 50)
xm <- mean(x)
c(xm, mean(x, trim = 0.10))

mean(USArrests, trim = 0.2)
```

Description

Usage

Arguments

Value

References

See Also

Examples

General structure of R commands

- R is an expression language with a very simple syntax
- Case sensitive that means "`a`" and "`A`" refer to different variables
- Commands are separated either by a semi-colon "`;`" or by a newline

- Elementary commands consist of either expressions or assignments
  - Expression is a command, it is evaluated, printed, and the value is lost
  - Assignment evaluates an expression and passes the value to a variable

- Elementary commands can be grouped together into one compound expression by braces "`{`" and "`}`"

- Comments can be put anywhere (except inside strings and function definitions) starting with a hashmark "#"

- If a command is not complete, it is indicated by a "+"

Freie Universität Berlin

Basics of GNU R -
data types, data manipulations, statistical basics
Selected elementary functions

This chapter is mainly based on the following text book:
Everitt, B.S., Hothorn, T. (2010) A Handbook of Statistical Analysis using R. 2nd. edition. CRC Press.

A first session

- Your first simple data set, a vector

```
> x <- c(3,5,9)
```

- Accessing elements of a vector

```
> x
> x[2]
> x[1:3]
```

- Important R structures
  - vectors
  - scalars `> x <- 6`
  - character strings `> y <- "ABC"`
  - matrices `> m <- rbind(c(1,4),c(4,9))`
  - lists `> l <- list(u=2, v="abc")`
  - dataframes

```
> d <- dataframe(list(students=c("Bob", "Jill"), ages=c(15,16)))
```

Data objects

- Exemplary data set: Forbes2000
  - Set of 2,000 world leading companies
  - Year of data collection: 2004

- Available via **HSAUR2** package*

- Load data from package (used here for examples):

```
> data("Forbes2000", package="HSAUR2")
```

- List objects that currently stored in global environment:

```
> ls()
```

- Show data

```
> print()
```

- Description of object structure

```
> str()
```

* Everitt, B.S., Hothorn, T. (2010) A Handbook of Statistical Analysis using R. 2nd. edition. CRC Press.

Data frames

- Object that consists of rows and columns
  - Rows: contain different observations from your study, or measurements from your experiment
  - Columns: contain the value of the different variables (list of vectors)
- Possible values are numbers, text, calendar dates, logical variables

- Other ways to explore the object

  Object type: `> class()`

  Dimension of a table: `> dim()`

  Number of rows/columns: `> nrow()` / `> ncol()`

  Names of variables: `> names()`

- Determine the value of a single variable

```
> class(objectName[,"variableName"])
```

Vector

- Elementary structure for data handling
- Variables with one or more values of the same type
- In R a scalar is a vector of the length 1

- Available functions
  - Length of a vector: `> length()`
  - Minimum of a vector: `> min()`
  - Maximum of a vector: `> max()`
  - Sum of a vector: `> sum()`
  - Mean of a vector: `> mean()`
  - Range of a vector: `> range()`

- Accessing elements in a vector
  `> class(objectName[,"variableName"])[position]`

- Class `factor` vs. class `character`

Basic data manipulation

- Extracting single variables from dataframe

  ```
  > var_name <- objectName$var_name
  ```

  (equals `> var_name <- (objectName[,"variableName"])`)

  - Example

    ```
    > companies <- Forbes2000$name
    ```

- Missing values
  - Special symbol NA, indicating that measurement is not available
  - Example

    ```
    > na_profits <- is.na(Forbes2000$profits)
    > table(na_profits)
    ```

- Selecting a subset

  ```
  > var_name <- subset(objectName, country=="var_name")
  ```

  - Example

    ```
    > dim(subset(Forbes2000, country == "United Kingdom"))
    ```

Handling of missing data

- In statistical data sets you will finde often missing values, which is represented as NA

- How to use NA?
- Example

```
> x <- c(88, NA, 12, 168, 13)
> x(mean)
[1] NA
> x(mean, na.rm=T)
[1] 70.25
```

- NA has not the same meaning as NULL

Simple summary statistics

Minimum	`> min(x)`
Maximum	`> max(x)`
Median	`> median(x)`
Mean	`> mean(x)`
Variance	`> var(x)`
Standard deviation	`> sd(x)`
Quantile	`> quantile(x)` but also `> quantile(x, c(0,.33,.66,1))`

- Another possibility is the generic function `> summary`

- Example: Compare profits in each category

```
> mprofits <- tapply(Forbes2000$profits,
  Forbes2000$category, mean, na.rm=TRUE)
```

Import/export your data

- Most frequent data formats are
  - Comma separated files such as csv
  - Excel files
  - SQL data base engines (Manual R Data Import/Export)

- Simple import by

```
> data <- read.table("fileName", header=TRUE, sep=",",
row.names=1)
```

or

```
> data <- read.csv("fileName", header=TRUE, sep=",")
```

- Data export is straightforward as well

```
> write.table(data, file="", sep=",", row.names=1)
```

or

```
> write.csv("fileName", header=TRUE, sep=",")
```

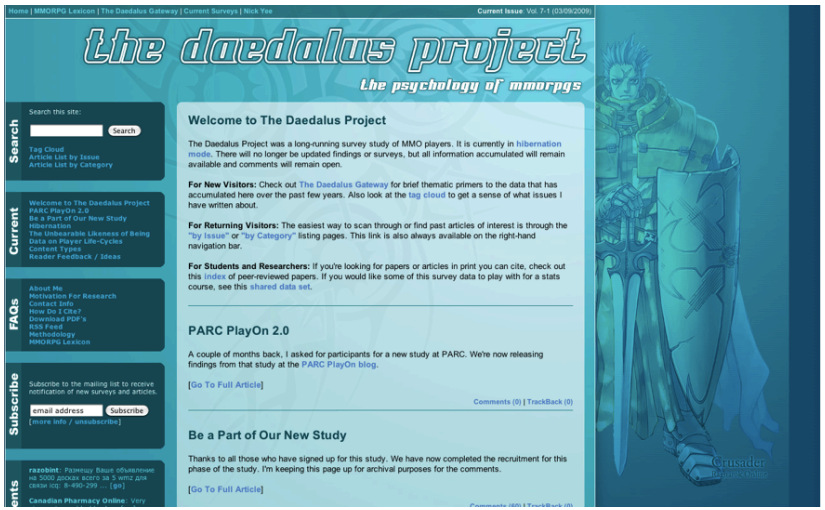
Freie Universität Berlin

Homework assignment 1

The Daedalus Project

- Exploring virtual world of MMORPGs
- Analysis of a dataset posted on “The Daedalus Project” website (http://www.nickyee.com/daedalus/docs/shared-data.php)
- Data set contains results of survey from January 2005
- Dataset for this project contains 3,000 records

- Download data set from class website



Assignment 1: Player of MMOPRG

- The sample data includes two demographic variables that characterize a gamer: age and gender.

- Based on the provided data, please answer the following questions
  - What is the average age of gamers?
  - What proportion of gamers are females?
  - Is the dispersion of the age of female gamers any different from the dispersion of male gamers? (hint: the `tapply` function is very useful here)

- Write a brief paragraph based on your answers defining who the players of MMOPRG are. Include the used R functions in your answer.

- Explain differences between the following functions:
  - `tapply` and `sapply`
  - `read.csv` and `read.table`