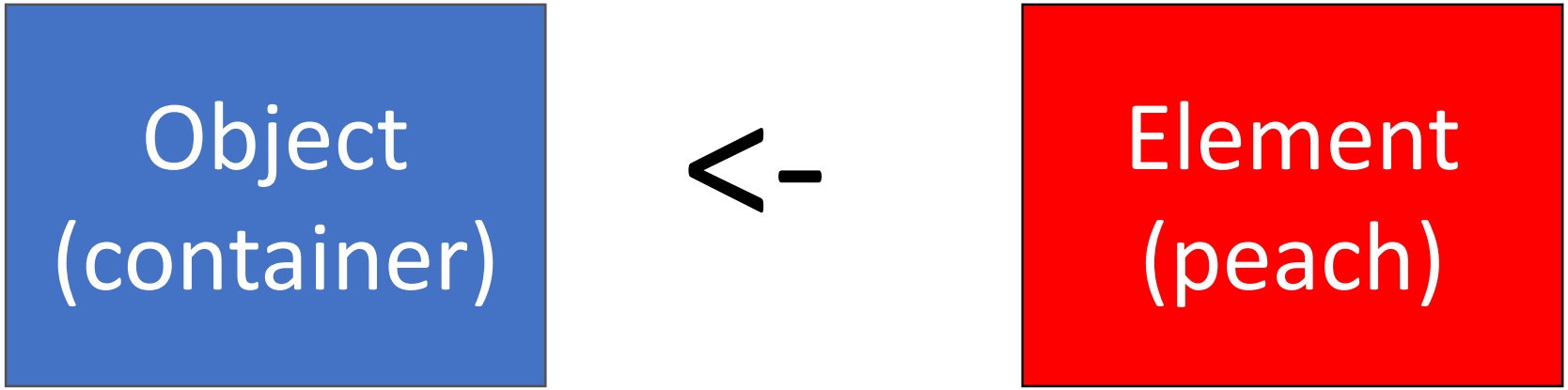
Objects in R

Lukáš Lehotský and Petr Ocelík

Object: what is it?

- **Object** is container
- **Element** is anything in container – a peach
- To reuse elements, they must be **stored as objects**
  - Any name **defined** by user
  - **Remain the same** unless overwritten
  - Must be **removed** by user as well

Object: creating/storing objects



Tips and tricks

- Keyboard hacks
  - For Czech keyboard and Win machine users, **Right Alt** (AltGr) allows you to type some special characters (AltGr + < for <)
- Script hacks
  - **#** allows you to write comments in scripts
  - When writing code, R will automatically add closing bracket, as well as closing quote symbol
  - **Tab** will finish the name of function or argument in R Studio – try typing help(and press Tab

Object types

Object types: vector

Object types: vector

- Vector is the **default object type**
  - Any object without more specific data structure is **vector** – least fancy object type
  - If contains **more than 1 element**, **always** created using function `c()`
  - **Same data class** within vector – otherwise converted to character

```
c(2,3,5)
[1] 2 3 5

vec <- c("aa", "bb", "cc", "dd", "ee")
vec
[1] "aa" "bb" "cc" "dd" "ee"
```

Object types: matrix

Object types: matrix

- 2 dimensions
- **Same data class** within matrix – otherwise converted to character
- Created using function `matrix()`

```
mat <- matrix(data = c(1,2,3,4,5,6,7,8,9,10,11,12),
              nrow = 3,
              ncol = 4)
mat
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

Object types: data frame

Object types: data frame

- 2 dimensions
- Typical **data set** – observations (rows) and variables (columns) – data class **per column**
- Function `data.frame()`
- In fact, it's a set of vectors (columns)
  - Always has to have **same number of elements** in vectors, from which it is created
  - Data classes may be different in each column, but **same within a column**

Object types: data frame

```
cars <- c("BMW","Audi","VW")
type <- c("3","A4","Passat")
price <- c(1200000,1164000,950500)
consumption <- c(6.2,5.9,5.9)

mydf <- data.frame(cars,type,price,consumption)
mydf
```

```
      cars  type     price      consumption
1     BMW   3        1200000    6.2
2     Audi  A4       1164000    5.9
3     VW    Passat   950500     5.9
```

Object types: list

Object types: list

- Heterogeneous – objects **nested** within an object
- Function `list()` creates nested structure

```
num.vector <- c(2, 3, 5)
char.vector <- c("aa", "bb", "cc", "dd", "ee")
mylist <- list(num.vector, char.vector, 3)
mylist
[[1]]
[1] 2 3 5

[[2]]
[1] "aa" "bb" "cc" "dd" "ee"

[[3]]
[1] 3
```

Libraries/packages

- Libraries combine together many pre-defined functions according to problem at hand
  - Most libraries download and install automatically from the integrated R repository (called CRAN) – only a proper **library name** needs to be entered
- As of September 17, 2019, 14925 libraries available (43 packages published on Sep. 17 alone)

Libraries/packages

Libraries/packages

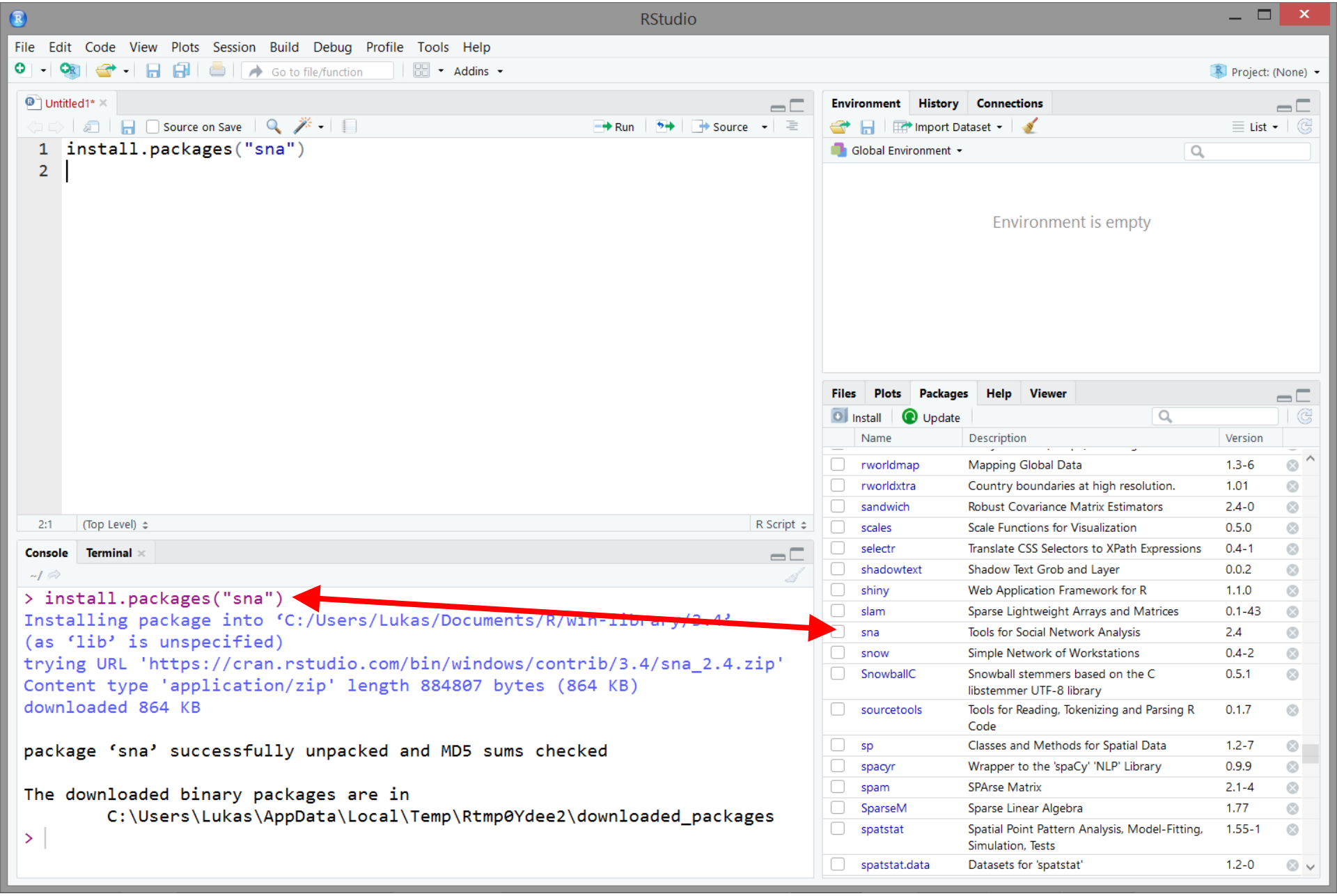
- If libraries are not available, they have to be installed

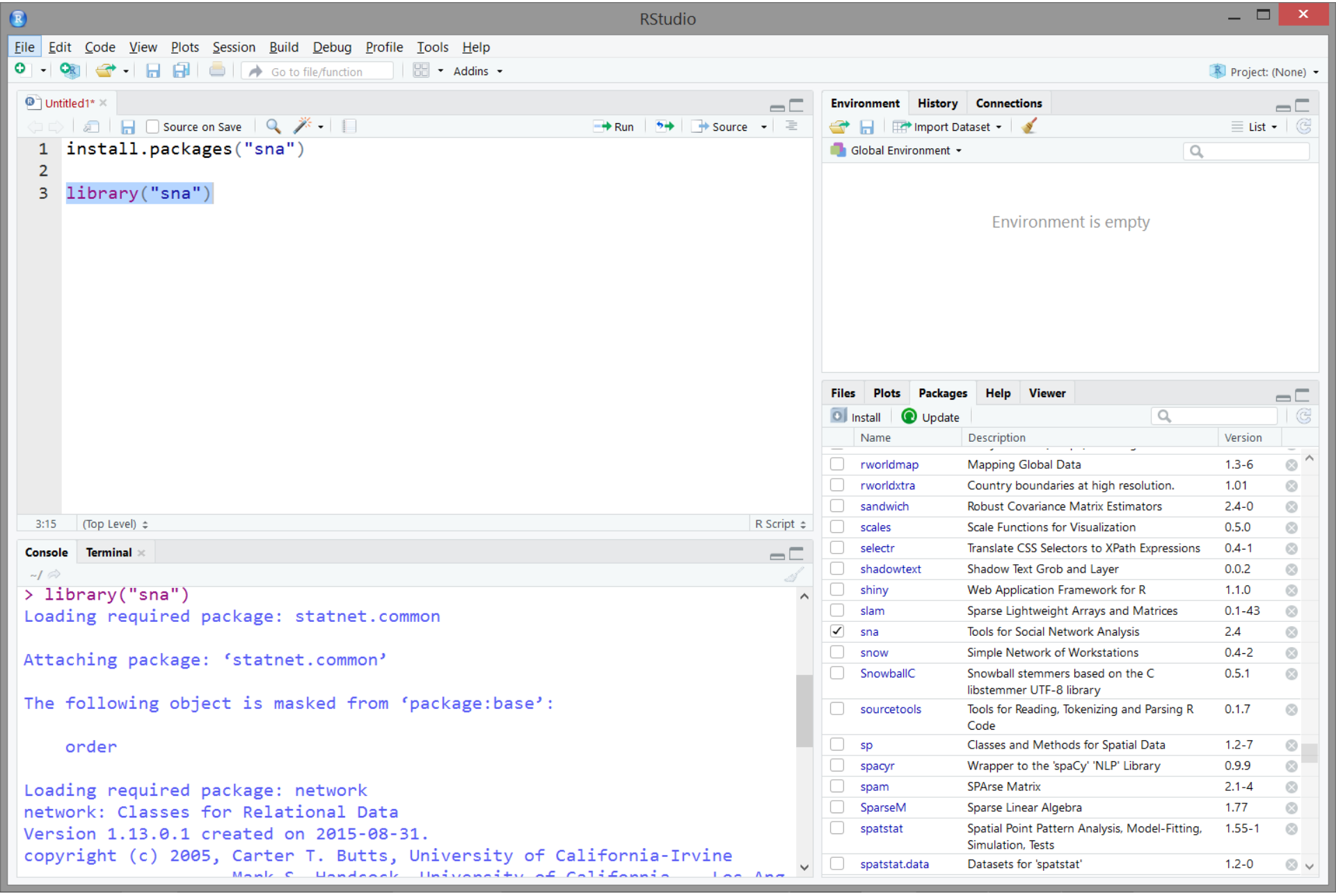
```
install.packages("sna")
```

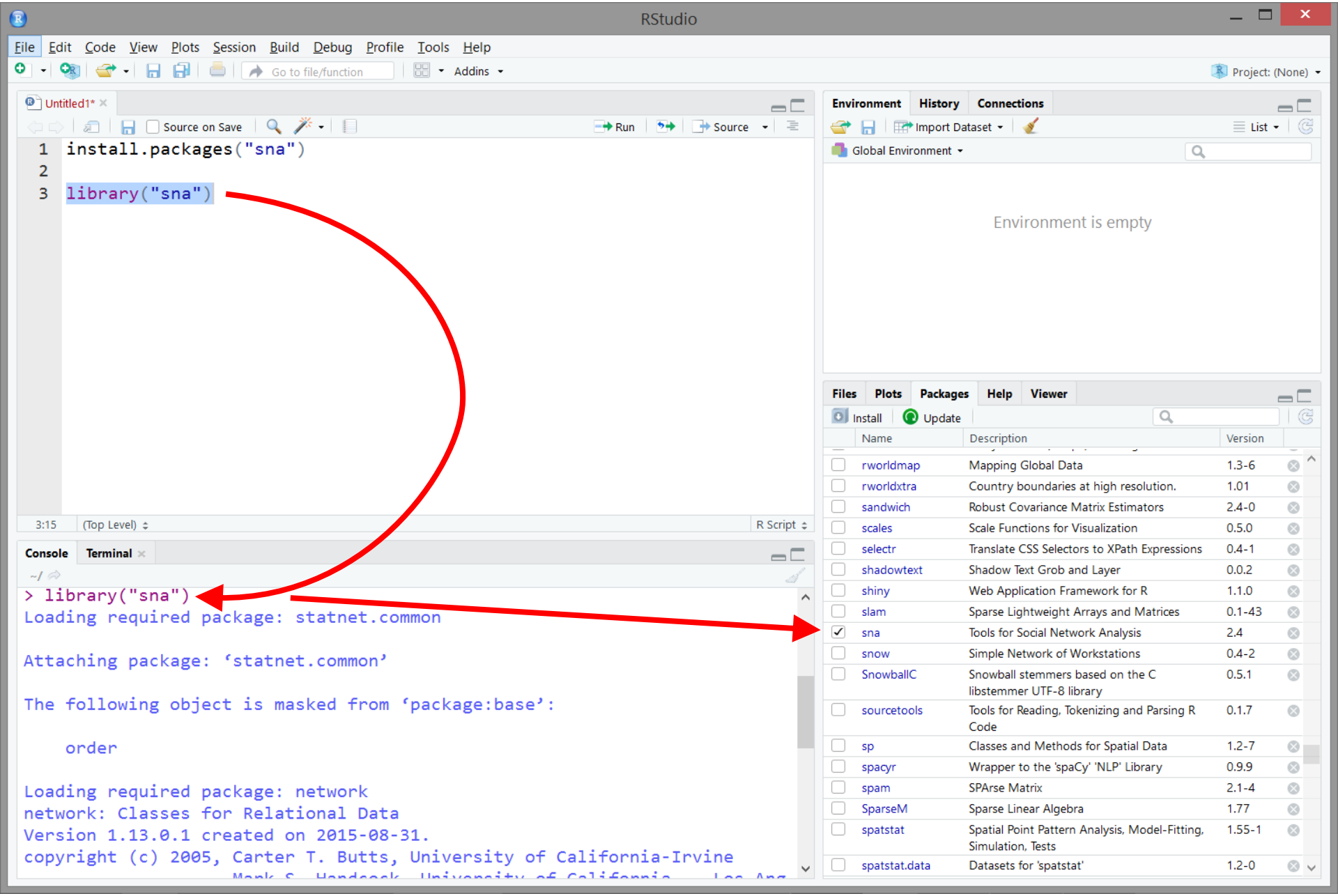
- Libraries must be loaded

```
library("sna")

library(sna)
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Untitled1*
Source on Save
Run
Source
1 install.packages("sna")
2
2:1
(Top Level)
R Script
Console
Terminal
~/
> install.packages("sna")
Installing package into 'C:/Users/Lukas/Documents/R/win-library/3.4'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/sna_2.4.zip'
Content type 'application/zip' length 884807 bytes (864 KB)
downloaded 864 KB
package 'sna' successfully unpacked and MD5 sums checked
The downloaded binary packages are in
C:\Users\Lukas\AppData\Local\Temp\Rtmp0Ydee2\downloaded_packages
>
Environment
History
Connections
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
Install
Update
Name
Description
Version
rworldmap Mapping Global Data 1.3-6
rworldxtra Country boundaries at high resolution. 1.01
sandwich Robust Covariance Matrix Estimators 2.4-0
scales Scale Functions for Visualization 0.5.0
selectr Translate CSS Selectors to XPath Expressions 0.4-1
shadowtext Shadow Text Grob and Layer 0.0.2
shiny Web Application Framework for R 1.1.0
slam Sparse Lightweight Arrays and Matrices 0.1-43
sna Tools for Social Network Analysis 2.4
snow Simple Network of Workstations 0.4-2
SnowballC Snowball stemmers based on the C libstemmer UTF-8 library 0.5.1
sourcetools Tools for Reading, Tokenizing and Parsing R Code 0.1.7
sp Classes and Methods for Spatial Data 1.2-7
spacyr Wrapper to the 'spaCy' 'NLP' Library 0.9.9
spam SPArse Matrix 2.1-4
SparseM Sparse Linear Algebra 1.77
spatstat Spatial Point Pattern Analysis, Model-Fitting, Simulation, Tests 1.55-1
spatstat.data Datasets for 'spatstat' 1.2-0


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Untitled1*
Source on Save
Run
Source
1 install.packages("sna")
2
3 library("sna")
3:15
(Top Level)
R Script
Console
Terminal
~/
> library("sna")
Loading required package: statnet.common
Attaching package: 'statnet.common'
The following object is masked from 'package:base':
order
Loading required package: network
network: Classes for Relational Data
Version 1.13.0.1 created on 2015-08-31.
copyright (c) 2005, Carter T. Butts, University of California-Irvine
Environment
History
Connections
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
Install
Update
Name
Description
Version
rworldmap Mapping Global Data 1.3-6
rworldxtra Country boundaries at high resolution. 1.01
sandwich Robust Covariance Matrix Estimators 2.4-0
scales Scale Functions for Visualization 0.5.0
selectr Translate CSS Selectors to XPath Expressions 0.4-1
shadowtext Shadow Text Grob and Layer 0.0.2
shiny Web Application Framework for R 1.1.0
slam Sparse Lightweight Arrays and Matrices 0.1-43
sna Tools for Social Network Analysis 2.4
snow Simple Network of Workstations 0.4-2
SnowballC Snowball stemmers based on the C libstemmer UTF-8 library 0.5.1
sourcetools Tools for Reading, Tokenizing and Parsing R Code 0.1.7
sp Classes and Methods for Spatial Data 1.2-7
spacyr Wrapper to the 'spaCy' 'NLP' Library 0.9.9
spam SPArse Matrix 2.1-4
SparseM Sparse Linear Algebra 1.77
spatstat Spatial Point Pattern Analysis, Model-Fitting, Simulation, Tests 1.55-1
spatstat.data Datasets for 'spatstat' 1.2-0

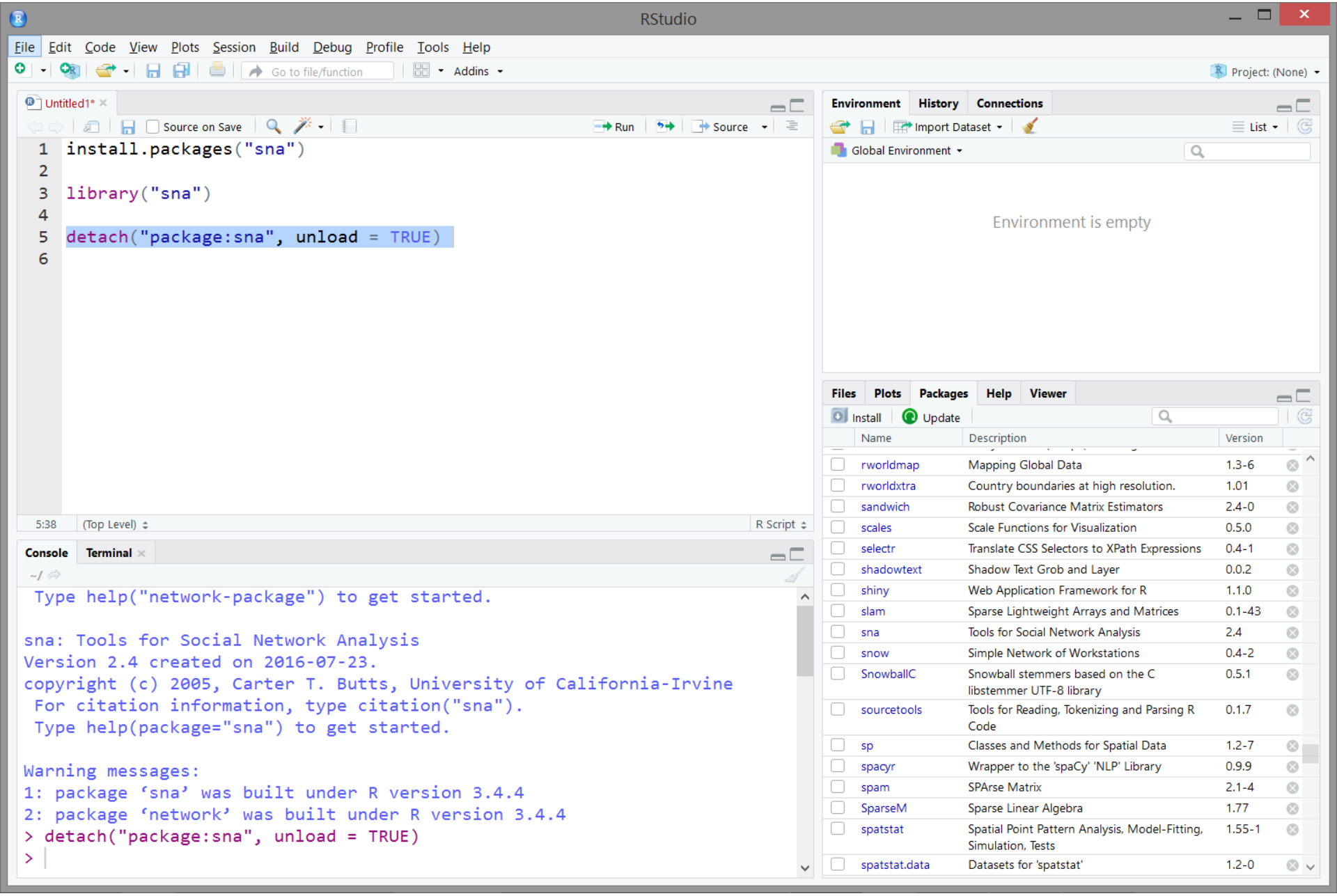

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Untitled1*
Source on Save
Run
Source
1 install.packages("sna")
2
3 library("sna")
3:15
(Top Level)
R Script
Console
Terminal
~/
> library("sna")
Loading required package: statnet.common
Attaching package: 'statnet.common'
The following object is masked from 'package:base':
order
Loading required package: network
network: Classes for Relational Data
Version 1.13.0.1 created on 2015-08-31.
copyright (c) 2005, Carter T. Butts, University of California-Irvine
Environment
History
Connections
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
Install
Update
Name
Description
Version
rworldmap Mapping Global Data 1.3-6
rworldxtra Country boundaries at high resolution. 1.01
sandwich Robust Covariance Matrix Estimators 2.4-0
scales Scale Functions for Visualization 0.5.0
selectr Translate CSS Selectors to XPath Expressions 0.4-1
shadowtext Shadow Text Grob and Layer 0.0.2
shiny Web Application Framework for R 1.1.0
slam Sparse Lightweight Arrays and Matrices 0.1-43
sna Tools for Social Network Analysis 2.4
snow Simple Network of Workstations 0.4-2
SnowballC Snowball stemmers based on the C libstemmer UTF-8 library 0.5.1
sourcetools Tools for Reading, Tokenizing and Parsing R Code 0.1.7
sp Classes and Methods for Spatial Data 1.2-7
spacyr Wrapper to the 'spaCy' 'NLP' Library 0.9.9
spam SPArse Matrix 2.1-4
SparseM Sparse Linear Algebra 1.77
spatstat Spatial Point Pattern Analysis, Model-Fitting, Simulation, Tests 1.55-1
spatstat.data Datasets for 'spatstat' 1.2-0

Libraries/packages

- Libraries **don't** load/unload **automatically**
  - Any time R is started, libraries have to be **reloaded**
  - Packages may be unloaded when necessary

```
library("sna")

detach("package:sna", unload = TRUE)
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Untitled1*
Source on Save
Run
Source
1 install.packages("sna")
2
3 library("sna")
4
5 detach("package:sna", unload = TRUE)
6
5:38
(Top Level)
R Script
Console
Terminal
~/
Type help("network-package") to get started.
sna: Tools for Social Network Analysis
Version 2.4 created on 2016-07-23.
copyright (c) 2005, Carter T. Butts, University of California-Irvine
For citation information, type citation("sna").
Type help(package="sna") to get started.
Warning messages:
1: package 'sna' was built under R version 3.4.4
2: package 'network' was built under R version 3.4.4
> detach("package:sna", unload = TRUE)
>
Environment
History
Connections
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
Install
Update
Name
Description
Version
rworldmap Mapping Global Data 1.3-6
rworldxtra Country boundaries at high resolution. 1.01
sandwich Robust Covariance Matrix Estimators 2.4-0
scales Scale Functions for Visualization 0.5.0
selectr Translate CSS Selectors to XPath Expressions 0.4-1
shadowtext Shadow Text Grob and Layer 0.0.2
shiny Web Application Framework for R 1.1.0
slam Sparse Lightweight Arrays and Matrices 0.1-43
sna Tools for Social Network Analysis 2.4
snow Simple Network of Workstations 0.4-2
SnowballC Snowball stemmers based on the C libstemmer UTF-8 library 0.5.1
sourcetools Tools for Reading, Tokenizing and Parsing R Code 0.1.7
sp Classes and Methods for Spatial Data 1.2-7
spacyr Wrapper to the 'spaCy' 'NLP' Library 0.9.9
spam SPArse Matrix 2.1-4
SparseM Sparse Linear Algebra 1.77
spatstat Spatial Point Pattern Analysis, Model-Fitting, Simulation, Tests 1.55-1
spatstat.data Datasets for 'spatstat' 1.2-0

Object analysis

- Data **classes** and **object types**
  - Function `class()` returns data class/object type of object
- Object **structure** (organization of object type)
  - Function `str()`
- Object **dimensions**
  - One dimension - `length()`
  - Two and more dimensions – `dim()`, `nrow()`, `ncol()`

Object analysis

```
length(1:30)
[1] 30

dim(mydf)
[1] 3 4

length(mydf)
[1] 4

nrow(mydf)
[1] 3

ncol(mydf)
[1] 4
```

Object analysis

```
class(mydf)
[1] "data.frame"

str(mydf)
'data.frame':     3 obs. of  4 variables:
 $ cars       : chr  "BMW" "Audi" "VW"
 $ type       : chr  "3" "A4" "Passat"
 $ price      : num  1200000 1164000 950500
 $ consumption: num  6.2 5.9 5.9
```

Object analysis

- More sophisticated function `describe()` included in **library "psych"**

```
str(mydf)
'data.frame': 3 obs. of  4 variables:
 $ cars       : chr  "BMW" "Audi" "VW"
 $ type       : chr  "3" "A4" "Passat"
 $ price      : num  1200000 1164000 950500
 $ consumption: num  6.2 5.9 5.9

describe(mydf)
             vars   n      mean          sd          median
cars*        1      3      2             1.00        2.0
type*        2      3      2             1.00        2.0
price        3      3      1104833       134863.20   1164000.0
consumption  4      3      6             0.17        5.9
```

Object analysis

- Data frame or matrix **preview**
  - Function `head()` and `tail()` returns first or last 5 rows of the data frame
- Object **names**
  - Vector or list names may be accessed using `names()`
  - Data frame or matrix row and column names may be accessed using `colnames()` and `rownames()`

```
head(mydf)

tail(mydf)

colnames(mydf)
```

Object analysis

- **Visual inspection** of data is possible using function `View()`
  - Useful especially in visual inspection of matrices or data frames
  - There's inconsistency in implementation – function `view()` does not exist
- Manual edit of the data (Excel-like) is possible – function `fix()` – **not advised** (replicability)

```
View(mydf)

fix(mydf)
```

Object analysis: useful functions

```
class() # explore elements' data class
length() # explore number of first dim. of object
dim() # explore dimensions of two-dimensional obj.
nrow() # number of rows
ncol() # number of columns

head() # first few rows of data
tail() # last few rows of data
str() # explore structure of object

names() # names in the named vector - one dimension
rownames() # names of rows - two dimensions
colnames() # names of columns - two dimensions
```

Practice 1

- Create a data frame with the following structure

name	age	sex	econ_scale	soc_scale
Jose	17	male	-3	-7.2
Sara	22	female	0.6	0.2
Maria	21	female	2	0
Frank	21	male	-3	0.5
John	18	male	3.1	3

Practice 2

- Install library “psych”
- Enable the library
- Summarize and explore the data frame created in the Practice 1
  - Find and use all available functions (creative task)

Practice 3

- Type `eurodist` in your script
- Create an object from the eurodist dataset
- Visually inspect the data source
- Figure out the type of the data source (creative task)

Practice 4

- Create a matrix with following parameters
  - 6 by 6 layout
  - Contains numbers 0 and 1 in a 0 1 0 1 0 1 0 1 pattern (creative task)
  - The pattern is organized by rows

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    1    0    1    0    1
[2,]    0    1    0    1    0    1
[3,]    0    1    0    1    0    1
[4,]    0    1    0    1    0    1
[5,]    0    1    0    1    0    1
[6,]    0    1    0    1    0    1
```