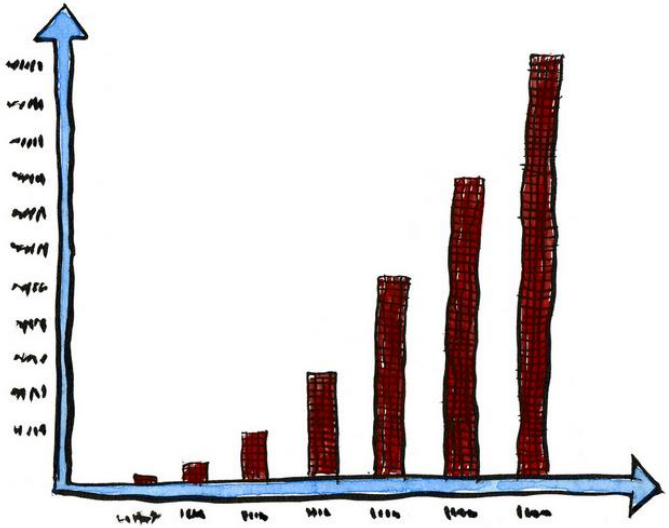
Introduction to R

R basics #1

Outline

- Introduction to R
- Data and Programming
  - Basic calculations
  - Data types, Data structures
  - Practice makes perfect #1
- Data analysis – Statistics ☺
  - Getting started
    - Loading data
    - Selecting data
    - Summary statistics
  - Data visualization
    - Scatterplots ☺



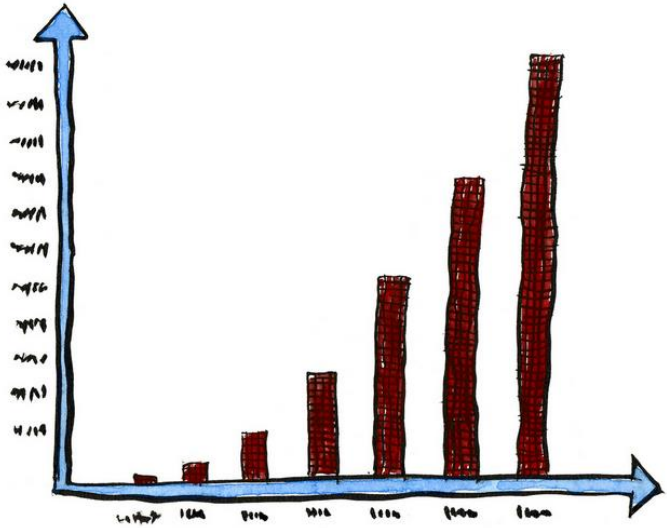
Dramatic increase in the amount of untrue statistics...

Introduction to R

- R: programming language, software environment for statistical computing
  - Open source and free
  - Windows and Mac compatibility
  - Popular statistical package
    - numerous functions and packages
    - support online
  - Efficient

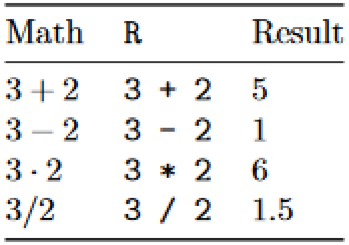
Outline

- Introduction to R
- **Data and Programming**
  - Basic calculations
  - Data types, Data structures
  - Practice makes perfect #1
- Data analysis – Statistics ☺
  - Getting started
    - Loading data
    - Selecting data
    - Summary statistics
  - Data visualization
    - Scatterplots ☺

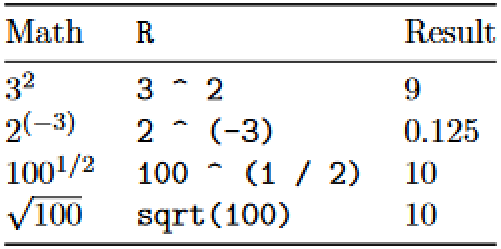


Dramatic increase in the amount of untrue statistics...

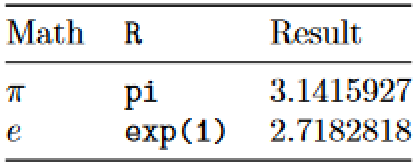
Basics



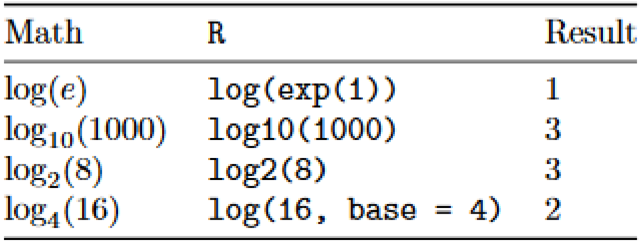
Math	R	Result
$3+2$	`3 + 2`	5
$3-2$	`3 - 2`	1
$3\cdot 2$	`3 * 2`	6
$3/2$	`3 / 2`	1.5



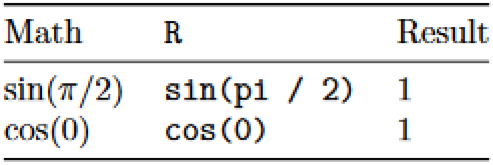
Math	R	Result
3^2	`3 ^ 2`	9
$2^{(-3)}$	`2 ^ (-3)`	0.125
$100^{1/2}$	`100 ^ (1 / 2)`	10
$\sqrt{100}$	`sqrt(100)`	10



Math	R	Result
π	`pi`	3.1415927
e	`exp(1)`	2.7182818



Math	R	Result
$\log(e)$	`log(exp(1))`	1
$\log_{10}(1000)$	`log10(1000)`	3
$\log_2(8)$	`log2(8)`	3
$\log_4(16)$	`log(16, base = 4)`	2

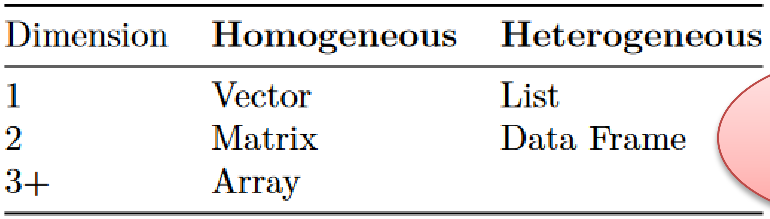


Math	R	Result
$\sin(\pi/2)$	`sin(pi / 2)`	1
$\cos(0)$	`cos(0)`	1

- Datatypes
  - Numeric, Integer, Character, Logical, Complex

    1.5, 1, “BCBL”, {TRUE, FALSE}, 1+2j

- Data structures



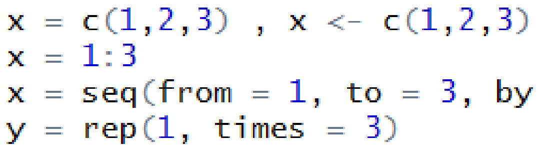
Dimension	**Homogeneous**	**Heterogeneous**
1	Vector	List
2	Matrix	Data Frame
3+	Array	

Elements with different datatype

x = c(1,2,3)
s = c(“Maite”, “Jorge”)
~~s = c(“Maite”, 1)~~ → s = c(“Maite”, “1”)
dFrame = data.frame(name=c(“Maite”, “Jorge”),
rank = c(1,2))

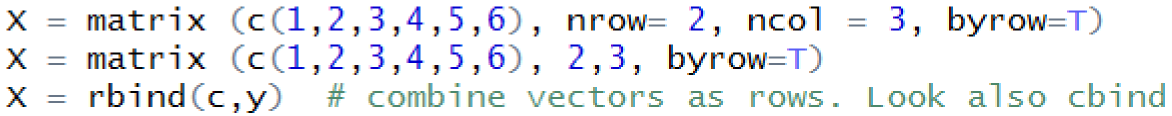
Data structures

- Vector:



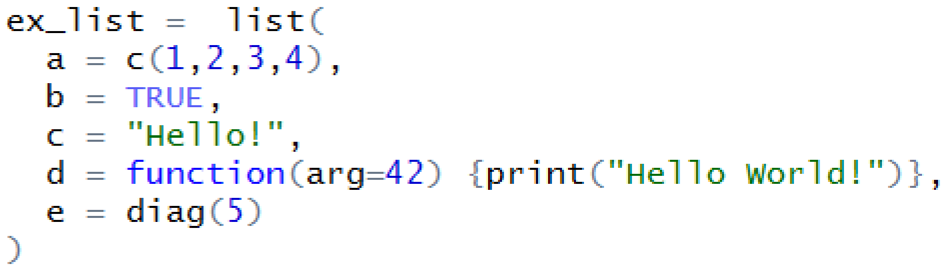
```
x = c(1,2,3) , x <- c(1,2,3)
x = 1:3
x = seq(from = 1, to = 3, by = 1)
y = rep(1, times = 3)
```

- Matrix: matrix (vector, #of rows, #of columns)



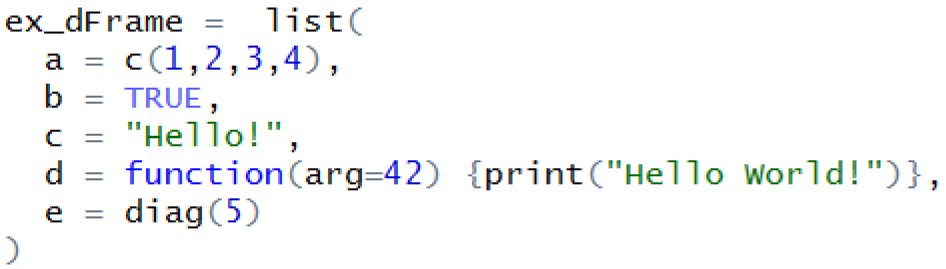
```
X = matrix (c(1,2,3,4,5,6), nrow= 2, ncol = 3, byrow=T)
X = matrix (c(1,2,3,4,5,6), 2,3, byrow=T)
X = rbind(c,y)  # combine vectors as rows. Look also cbind
```

- List



```
ex_list =  list(
  a = c(1,2,3,4),
  b = TRUE,
  c = "Hello!",
  d = function(arg=42) {print("Hello World!")},
  e = diag(5)
)
```

- Dataframe: a list of vectors



```
ex_dFrame =  list(
  a = c(1,2,3,4),
  b = TRUE,
  c = "Hello!",
  d = function(arg=42) {print("Hello World!")},
  e = diag(5)
)
```

Data structures

- Vector:

```
x = c(1,2,3) , x <- c(1,2,3)
x = 1:3
x = seq(from = 1, to = 3, by = 1)
y = rep(1, times = 3)
```

- Matrix: matrix (vector, #of rows, #of columns)

```
X = matrix (c(1,2,3,4,5,6), nrow= 2, ncol = 3, byrow=T)
X = matrix (c(1,2,3,4,5,6), 2,3, byrow=T)
X = rbind(c,y)  # combine vectors as rows. Look also cbind
```

- List

```
ex_list =  list(
  a = c(1,2,3,4),
  b = TRUE,
  c = "Hello!",
  d = function(arg=42) {print("Hello World!")},
  e = diag(5)
)
```

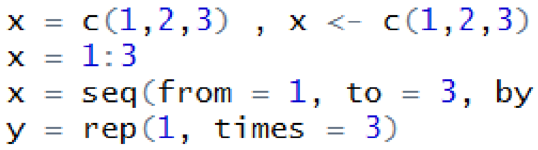
Open
data_structures#1.R
Follow the instructions
and lets “R”un

- Dataframe: a list of vectors

```
ex_dFrame =  list(
  a = c(1,2,3,4),
  b = TRUE,
  c = "Hello!",
  d = function(arg=42) {print("Hello World!")},
  e = diag(5)
)
```

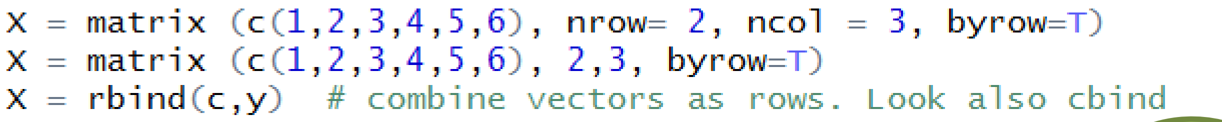
Data structures

- Vector:



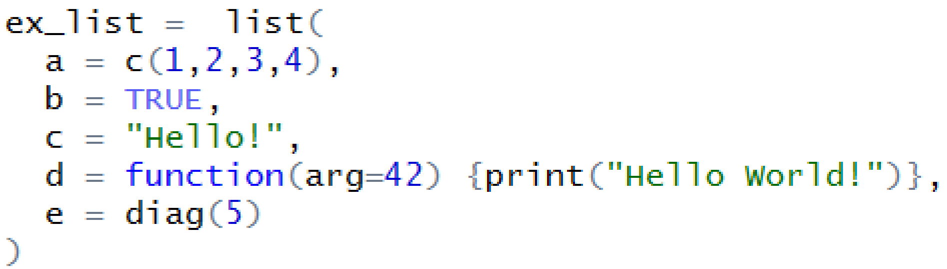
```
x = c(1,2,3) , x <- c(1,2,3)
x = 1:3
x = seq(from = 1, to = 3, by = 1)
y = rep(1, times = 3)
```

- Matrix: matrix (vector, #of rows, #of columns)



```
X = matrix (c(1,2,3,4,5,6), nrow= 2, ncol = 3, byrow=T)
X = matrix (c(1,2,3,4,5,6), 2,3, byrow=T)
X = rbind(c,y)  # combine vectors as rows. Look also cbind
```

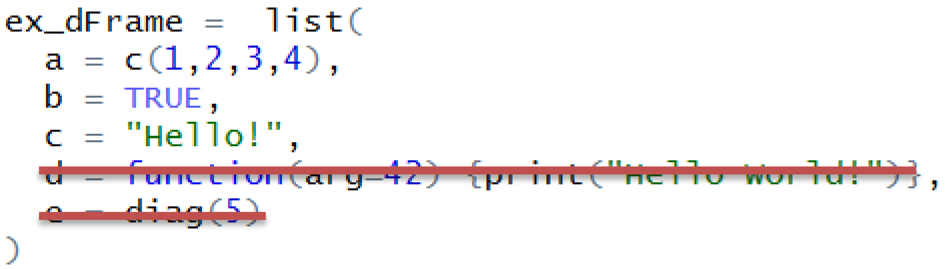
- List



```
ex_list =  list(
  a = c(1,2,3,4),
  b = TRUE,
  c = "Hello!",
  d = function(arg=42) {print("Hello World!")},
  e = diag(5)
)
```

the elements of a data frame must all be vectors, and have the same length

- Dataframe: a list of vectors



```
ex_dFrame =  list(
  a = c(1,2,3,4),
  b = TRUE,
  c = "Hello!",
  ~~d = function(arg=42) {print("Hello World!")}~~,
  ~~e = diag(5)~~
)
```

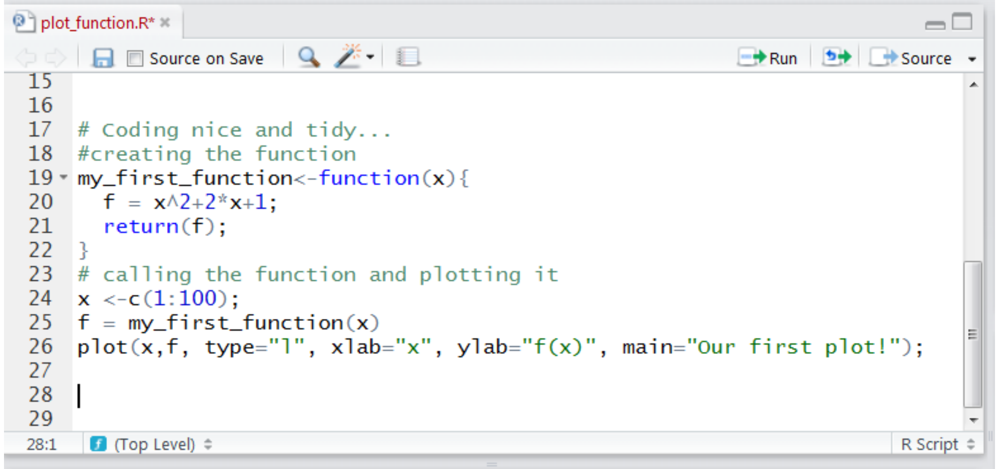
Practice makes perfect#1!

Create an R function!

# Your task is to create and plot the following function using R:
f(x)= x^2 + 2 x +1

# Open Practice_makes_perfect#1.R
And follow instructions

Practice makes perfect#1: Solution

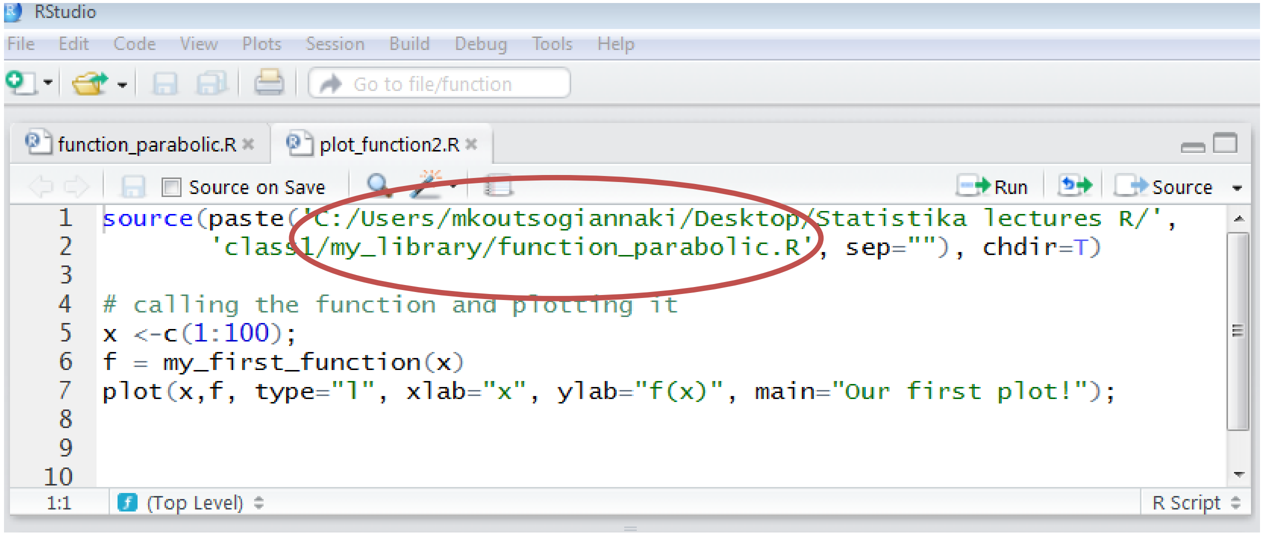


```


# Coding nice and tidy...
#creating the function
my_first_function<-function(x){
  f = x^2+2*x+1;
  return(f);
}
# calling the function and plotting it
x <-c(1:100);
f = my_first_function(x)
plot(x,f, type="l", xlab="x", ylab="f(x)", main="Our first plot!");


```

Function is created and called inside the same R script



```
source(paste('C:/Users/mkoutsogiannaki/Desktop/Statistika lectures R/',
         'class1/my_library/function_parabolic.R', sep=""), chdir=T)

# calling the function and plotting it
x <-c(1:100);
f = my_first_function(x)
plot(x,f, type="l", xlab="x", ylab="f(x)", main="Our first plot!");



```

Function is stored in a separate file and is imported using the source command

Practice makes perfect#@home!

Create an R function!

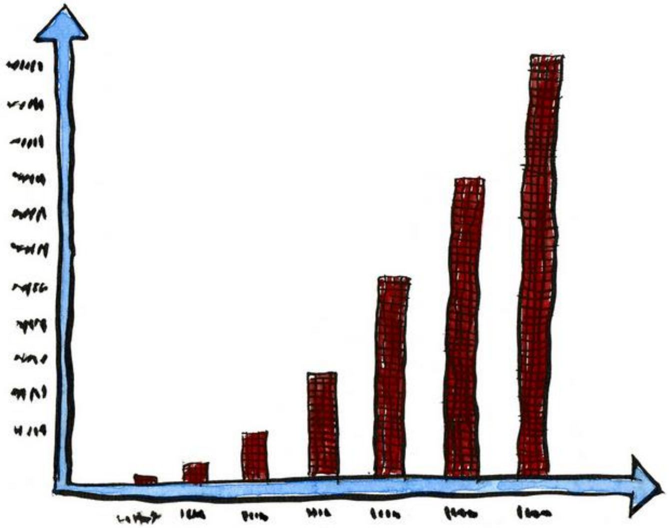
```
# Your task is to create a function that takes two input integers,
multiplies them and prints the result if its odd:
Hint: %% (modulo)

my_function <- function(arg1, arg2, …. ){
# multiply
# if (...) {
  some R code
} else {
  more R code
}
return (…)  # return the result
 }

#Now call the function  you have created.
f = my_function(x) # Then call your function
# print your result
```

Outline

- Introduction to R
- Data and Programming
  - Basic calculations
  - Data types, Data structures
  - Practice makes perfect #1
- Data analysis – Statistics ☺
  - Getting started
    - Loading data
    - Selecting data
    - Summary statistics
  - Data visualization
    - Scatterplots ☺

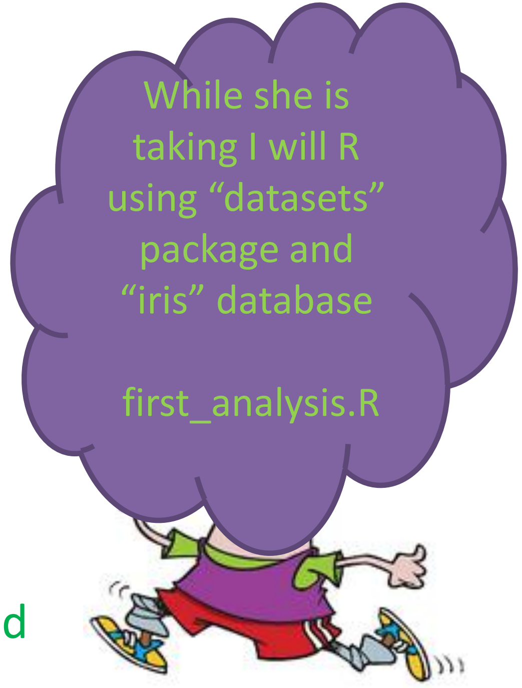


Dramatic increase in the amount of untrue statistics...

Getting started…

- Load data from R (no time to create our own)
  - install.packages(‘package name’)   # do this once
  - library(package_name)   # do this every time..
- Explore data
  - edit(myDataset)
  - head(myDataset)   # the first rows of the dataset (subset)
  - colnames(myDataset)   #names of columns
  - names(myDataset)
  - summary(myDataset)
- Select data
  - myDataset[1,2] # first row, 2nd column
  - myDataset[1,] # whole row
  - myDataset$V1 # whole column
  - subset(myDataset, Vn==‘Male') # select rows
  - subset(myDataset, select=c(V1, V2)) # select columns
  - subset(myDataset, Vn==‘Male' & V2>2.5) # conditioned
  - subset(myDataset, V3!=‘NA’) # remove outliers!

myDataset				
V1	V2	V3	…	Vn
0,1	4	0,8	2	Male
0,2	6	1,2	3	Female
0,8	8		6.3	Male
0,1	1	0,2	1	Male



While she is taking I will R using “datasets” package and “iris” database

first_analysis.R

Summary statistics

- Basic statistics
  - mean(V1), sd(V1)
- Summary: calculate summary statistics
  - summary(myDataset)
  - Groupwise statistics
    - summaryBy(V1+V2~Vn, data=myDataset, FUN=c(mean, var))
    - describeBY(myDataset$V1, myDataset$Vn)
- Scatterplot
  - pairs(myDataset)
  - pairs(Vn~., data=dataset, col=myDataset$Vn) # pair-wise scatterplots colored by class (categorical variable)
  - qplot(V1, V2, colour = Vn, data = myDataset)

Summary data!

Lets do that on iris database!

first_analysis.R

What can you observe? Can you find two variables from which we can predict the Type of Flower?

Material

- DISCOVERING STATISTICS USING R
  - ANDY FIELD, JEREMY MILES, ZOË FIELD
- ? # search for a function in R with installed package
- ?? # packages is missing. Look the package to install for the
  # function you wan
- Search on the internet
- R is huge! You can do same thing with 100 ways (multiple packages)
  Do not get lost! Start from the basics and build slowly your code.