Week 1 Exercises

Jeroen Knoester

Exercise 1 - Variables

a) Write an expression to compute the sum of the numbers 3 and 4

```
sum(3, 4)
```

```
## [1] 7
```

b) Write an expression to determine if the product of 4 and 5 is equal to 21

```
prod(4, 5) == 21
```

```
## [1] FALSE
```

c) Create the variables x,y,z with values (respectively) 2, 3.5, 7

```
x <- 2
y <- 3.5
z <- 7
```

d) Exploit a function of R to investigate the datatypeof the three variables x,y,z

```
typeof(x)
```

```
## [1] "double"
```

```
typeof(y)
```

```
## [1] "double"
```

```
typeof(z)
```

```
## [1] "double"
```

e) Write an expression to compute the multiplication of the three variables x,y,z

```
prod(x,y,z)
```

```
## [1] 49
```

f) Write a Boolean expression (i.e., which result is T or F) which checks if the product of x,y,z is equal to 49

```
prod(x,y,z) == 49
```

```
## [1] TRUE
```

g) Create a variable s with value "Alexander"

```
s <- "Alexander"
```

h) Look for a function returning the length of a string and use it to determine the length of s

```
nchar(s)
```

```
## [1] 9
```

Exercise 2 - Vectors

a) Create a variable ?? (a vector) with the value 1: 10

```
x <- 1:10
```

b) How can you investigate the length of ???

```
length(x)
```

```
## [1] 10
```

c) Multiply x by the number 1.2

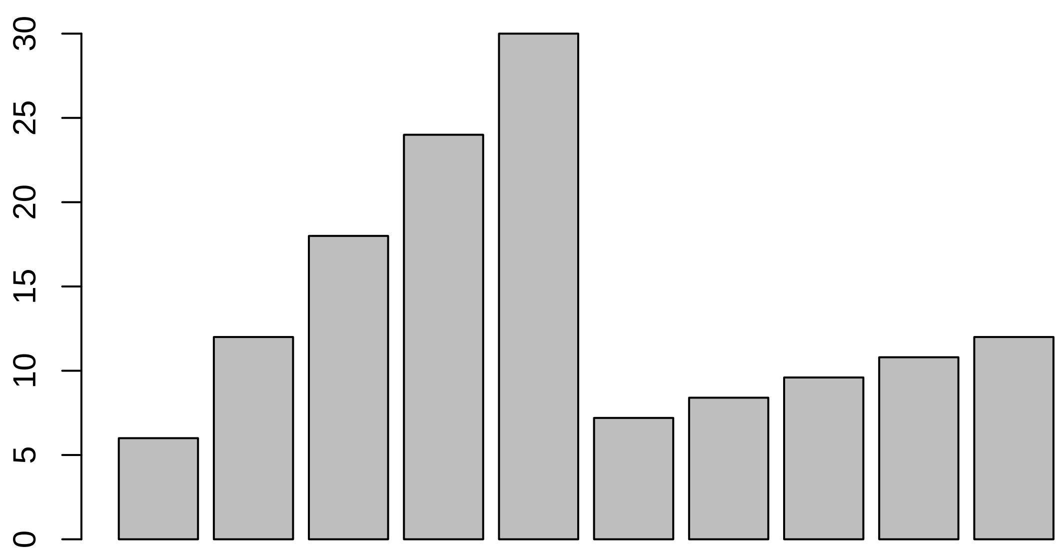
```
x <- x*1.2
```

d) Multiply the first five elements of x by 2

```
x[1:5] <- x[1:5] * 5
```

e) Visualize x using *barplot(x)*

```
barplot(x)
```

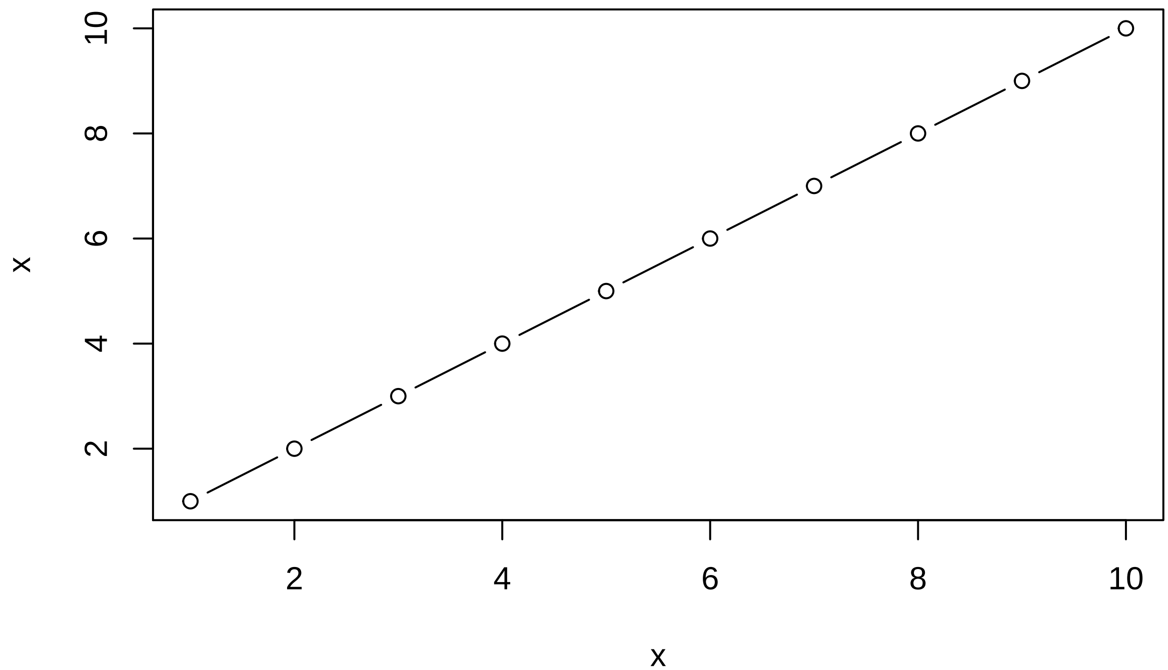


f) Assign to x the values of all integers from 0 to 10

```
x <- 1:10
```

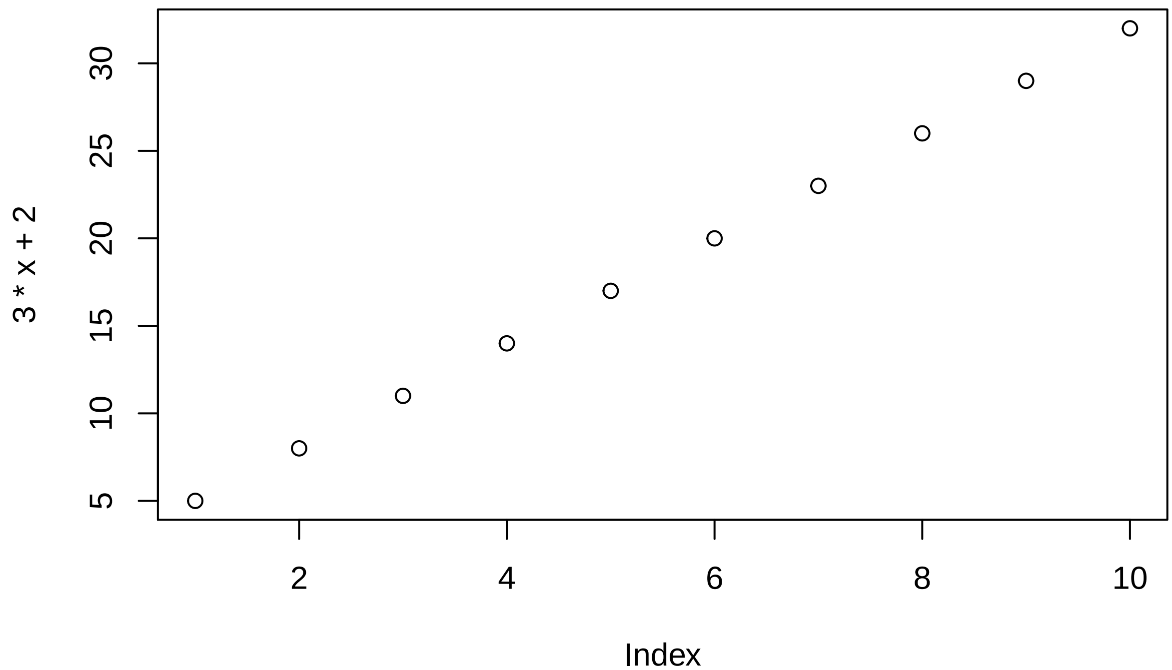
g) Create a graph of x against x using *plot(x,x,type = "b")*

```
plot(x,x,type= "b")
```



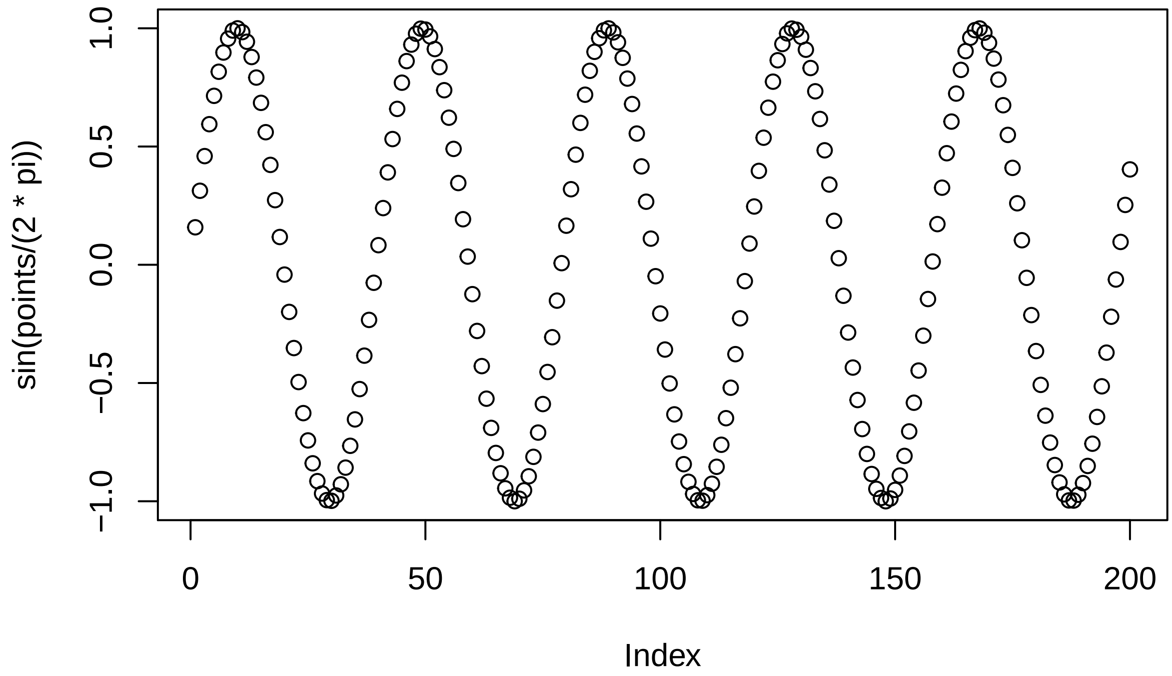
h) Use *plot* to draw the graph of the function $y = 3x + 2$ for values of x between 0 and 10

```
plot(3*x+2)
```



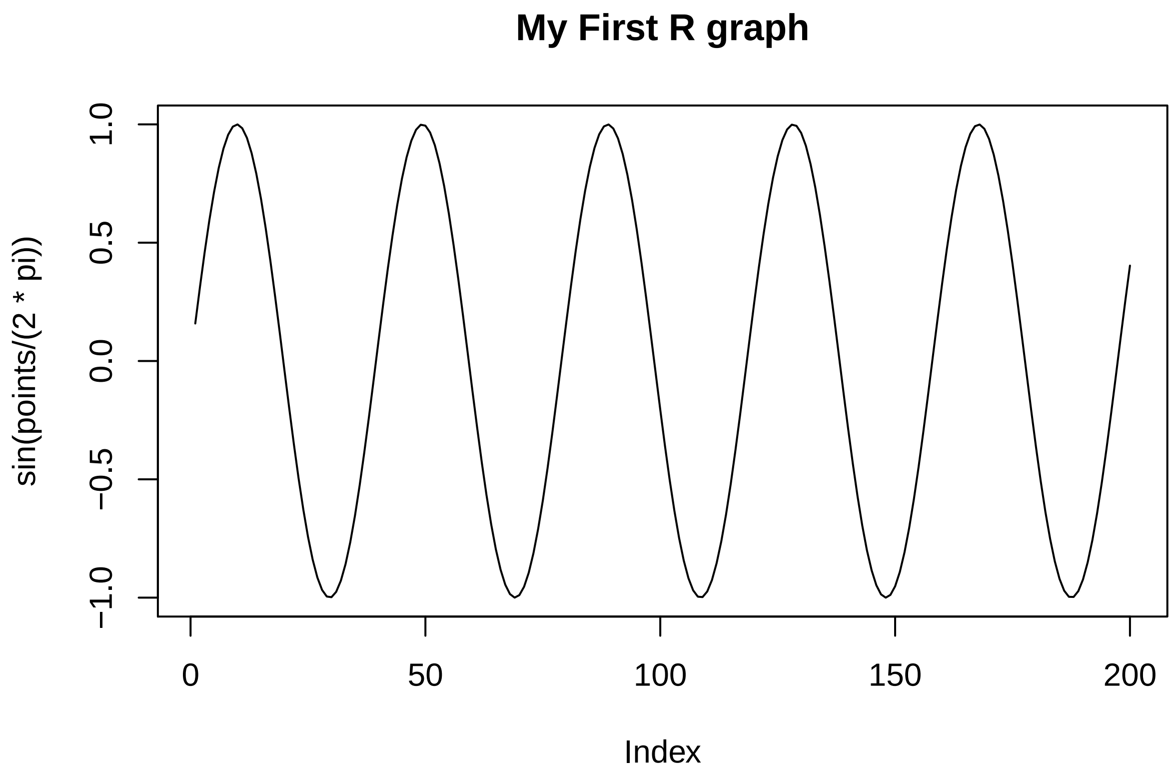
i) Use *plot* to draw the graph of the function $y = sin(x)$ for values of x between 0 and 2 pi (use at least 200 points)

```
points <- 1:200
plot(sin(points / (2 * pi))) # this one is probably wrong
```



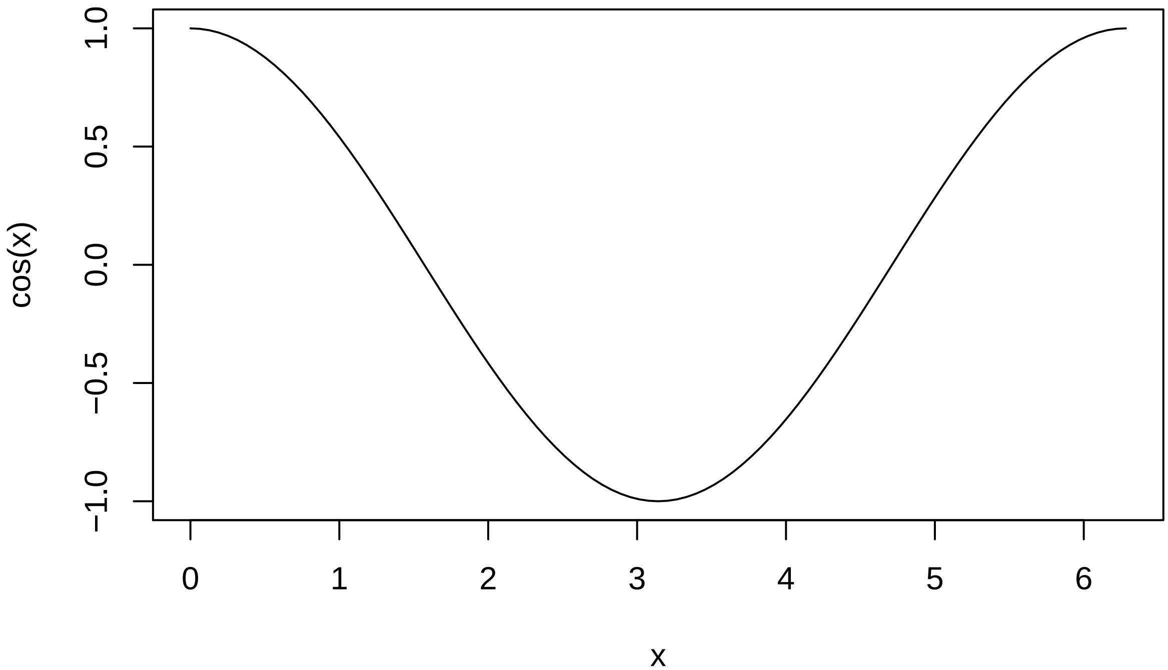
j) Find out how the graph can be made of only one line (without symbols) and give the graph a name (for example *"My First R graph"*)

```
plot(sin(points / (2 * pi)), type="l", main="My First R graph")
```



k) Find out how the function *"curve"* works and use it to plot the values of *cos(x)* between 0 and 2 * pi

```
curve(cos(x), from = 0, to=2*pi)
```


1.0
0.5
0.0
−0.5
−1.0
cos(x)
0
1
2
3
4
5
6
x

Exercise 3 - Factors and dataframes a) Read the dataset "Ozone" (which you can download from N@tschool) into a variable called *"air"*

```
air <- read.csv("ozone.csv")
```

b) Which type of object is air?

```
typeof(air)
```

```
## [1] "list"
```

c) Which are the names of the columns of air?

```
names(air)
```

```
## [1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"
```

d) From how many lines is air composed?

```
lengths(air)
```

```
##   Ozone Solar.R    Wind    Temp   Month     Day
##     153     153     153     153     153     153
```

e) In some cells, you see the value ôNAö. What does this mean?

```
help(NA)
```

f) Try to compute the average ozone level using the simple "mean" command and verify that the result of this operation is NA. Why does this happen? Some values of the set "air" are not available, therefore you can not calculate a *"mean"*

g) Find out how to exclude NA values from the computation of the mean, and compute again the average of the ozone level (obtaining a number).

```
mean(air$Ozone, na.rm = TRUE)
```

```
## [1] 42.12931
```

h) Compute the mean temperature of the dataset.

```
mean(air$Temp)
```

```
## [1] 77.88235
```

i) How many observations of the dataset have a temperature strictly less than 67 and how many have one greater or equal than 67? NB: the sum of these two numbers should match the total lines number (as requested at step d).

```
attach(air)
```

```
## The following objects are masked from air (pos = 3):
##
##     Day, Month, Ozone, Solar.R, Temp, Wind

## The following objects are masked from air (pos = 4):
##
##     Day, Month, Ozone, Solar.R, Temp, Wind

## The following objects are masked from air (pos = 5):
##
##     Day, Month, Ozone, Solar.R, Temp, Wind
```

```
lowtemperatures <- subset(Temp, Temp < 67)
length(lowtemperatures)
```

```
## [1] 21
```

```r
hightemperatures <- subset(Temp, Temp >= 67)
length(hightemperatures)
```

```
## [1] 132
```

j) Create the vector *"hot"* with logical values (T/F), where TRUE corresponds to temperatures strictly higher than 80 Fahrenheit. Use this vector to split the Ozone values in two: the ozone levels of warm days (i.e., when the temperature is greater than 80) and the ozone levels of cold days (i.e., when the temperature is lower or equal to 80).

```r
hot <- Temp > 80
```

k) Determine the mean ozone levels of warm days and cold days

```r
airFiltered <- na.omit(air)
hotdays <- subset(airFiltered, airFiltered$Temp > 80)
colddays <- subset(airFiltered, airFiltered$Temp <= 80)
mean(hotdays$Ozone)
```

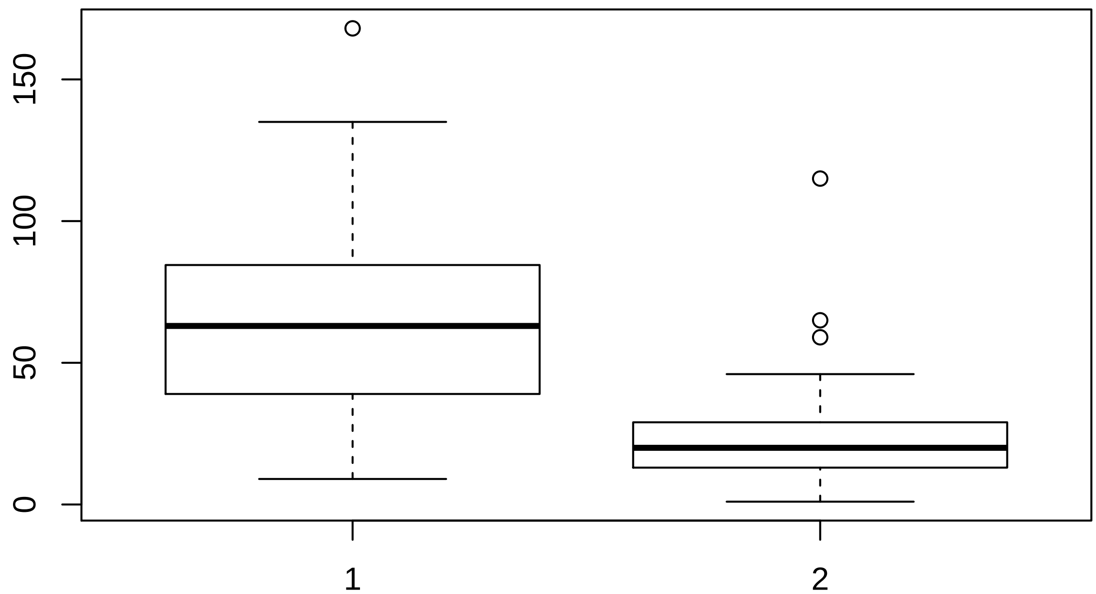
```
## [1] 64.41176
```

```r
mean(colddays$Ozone)
```

```
## [1] 23.13333
```

l) Produce a plot containing the two boxplots of ozone levels (warm days and cold days).

```r
boxplot(hotdays$Ozone, colddays$Ozone)
```


0
50
100
150
1
2