SRT411A0

Andres Mollejas

February 2, 2017

Introduction:

For this assignment we had to perform the tasks specified in the document given to us by our professor. This document gave us some of the basic elements we needed to complete the exercises. These exercises gave us the knowledge and confidence to be able to take on most tasks with relative ease. Other help was still needed and is listed in the sources at the bottom of the page.

Here is the link I followed to complete the assignment: https://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf

To Do question 1:

Compute the difference between 2014 and the year you started at this university and divide this by the difference between 2014 and the year you were born. Multiply this with 100 to get the percentage of your life you have spent at this university. Use brackets if you need them.

```
(2017-2015)/(2017-1997)*100
```

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

To Do question 2:

Repeat the previous ToDo, but with several steps in between. You can give the variables any name you want, but the name has to start with a letter.

```
a=(2017-2015)
b=(2017-1997)
a/b*100
```

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

To Do question 3:

Compute the sum of 4, 5, 8 and 11 by first combining them into a vector and then using the function sum.

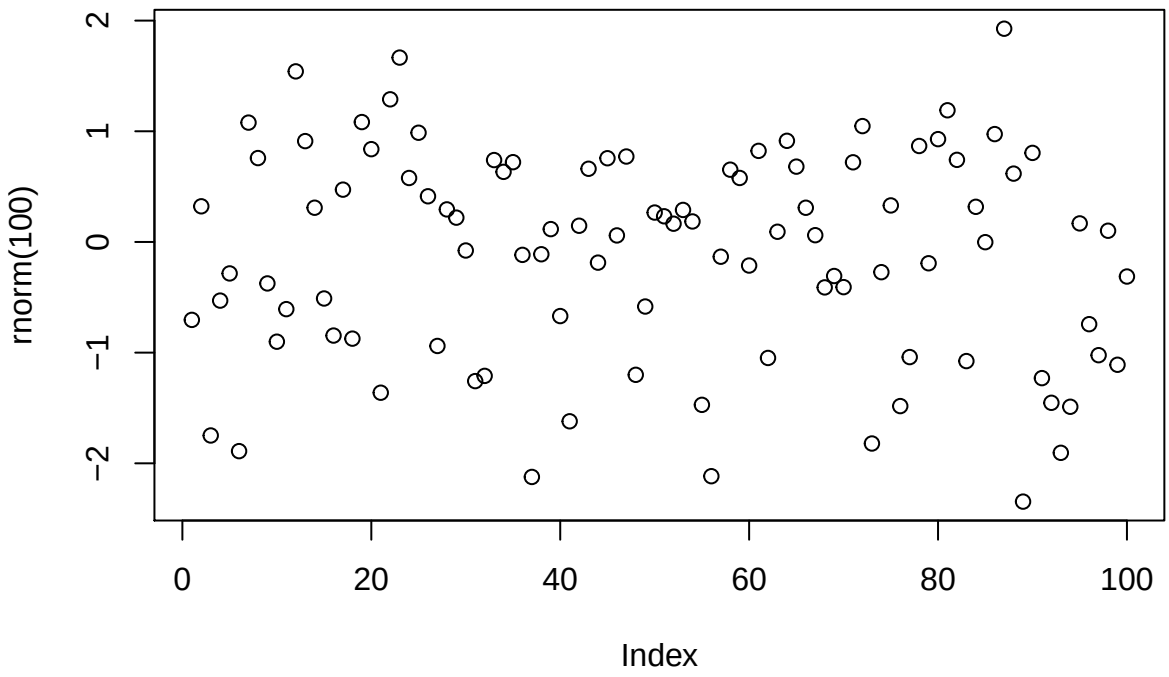
```
sum(TotalSum <- c(4, 5, 8, 11))
```

```
## [1] 28
```

To Do question 4:

Plot 100 normal random numbers:

```r
plot(rnorm(100))
```



To Do question 5:

Find help for the sqrt function.

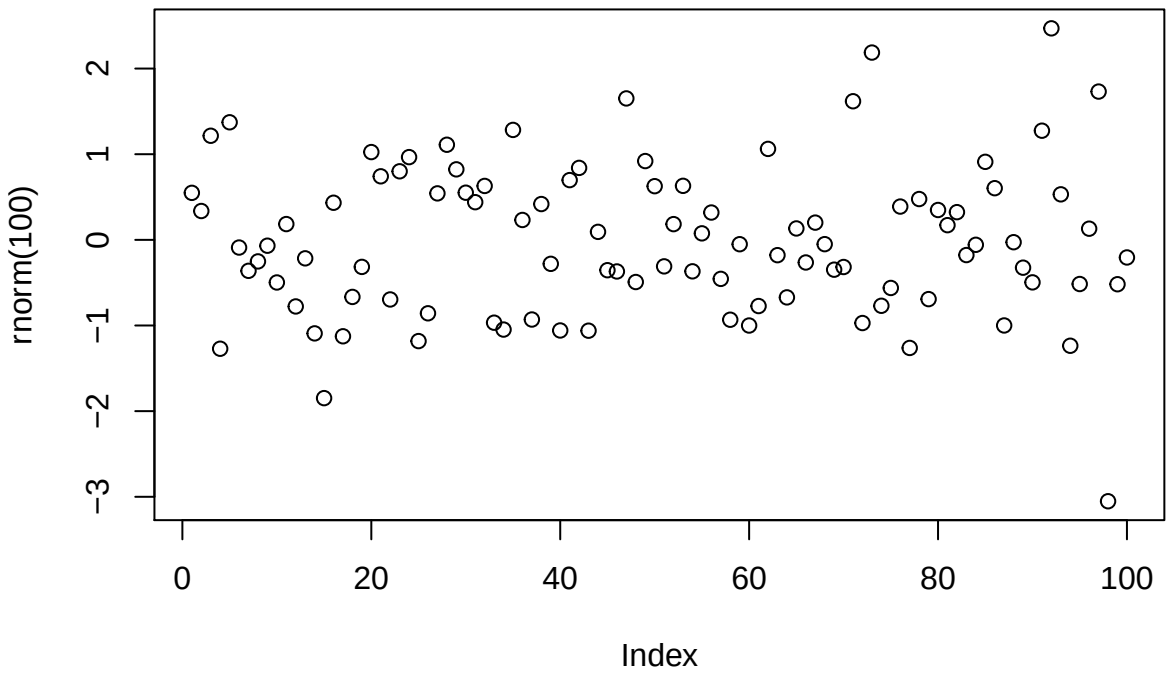
```r
help(sqrt)
```

```
## starting httpd help server ...
##  done
```

To Do question 6:

Make a file called firstscript.R containing Rcode that generates 100 random numbers and plots them, and run this script several times.

```r
source("firstscript.R")
```




```
#to repeat this code one would have to have the cursor on the line of code and keep pressing control +
#to view the contents of firstscript.R please see the attached links I have uploaded
```

To Do question 7:

Put the numbers 31 to 60 in a vector named P and in a matrix with 6 rows and 5 columns named Q. Tip: use the function seq. Look at the different ways scalars, vectors and matrices are denoted in the workspace window.

```
P = 31:60
Q <- matrix(data=P, 6, 5)
Q
```

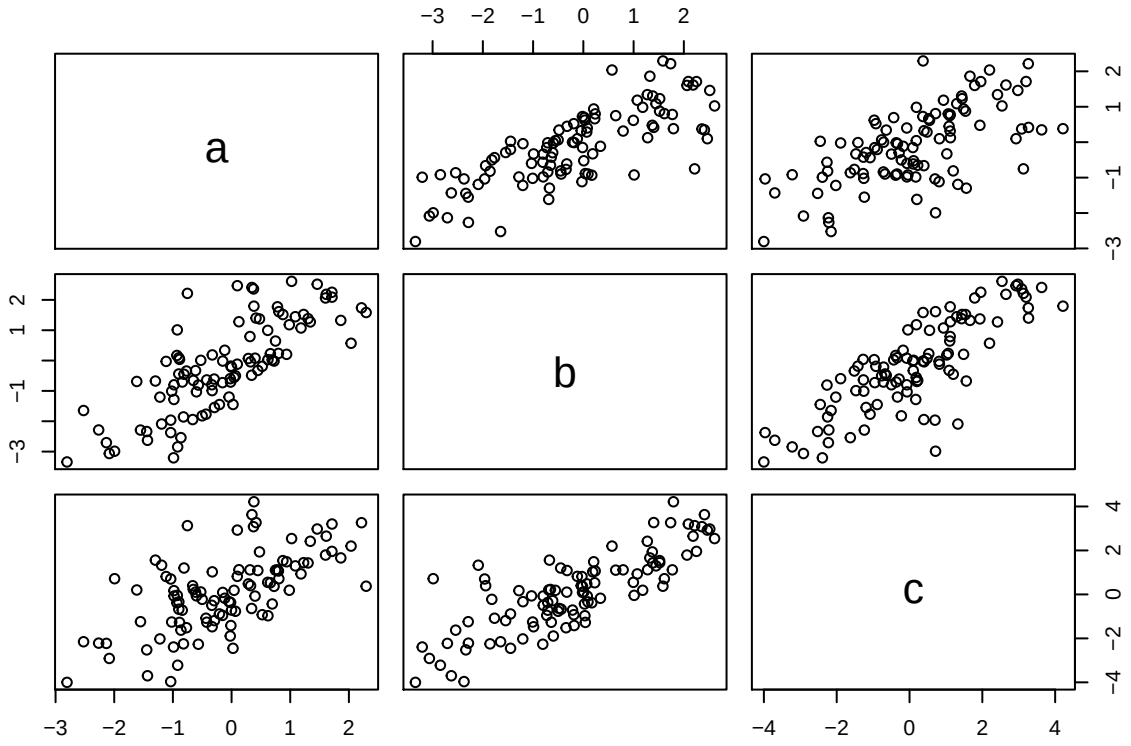
```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   31   37   43   49   55
## [2,]   32   38   44   50   56
## [3,]   33   39   45   51   57
## [4,]   34   40   46   52   58
## [5,]   35   41   47   53   59
## [6,]   36   42   48   54   60
```

To Do question 8:

Make a script file which constructs three random normal vectors of length 100. Call these vectors x1, x2 and x3. Make a data frame called with three columns (called a, b and c)

containing respectively x1, x1+x2 and x1+x2+x3. Call the following functions for this data frame: plot(t) and sd(t). Can you understand the results? Rerun this script a few times.

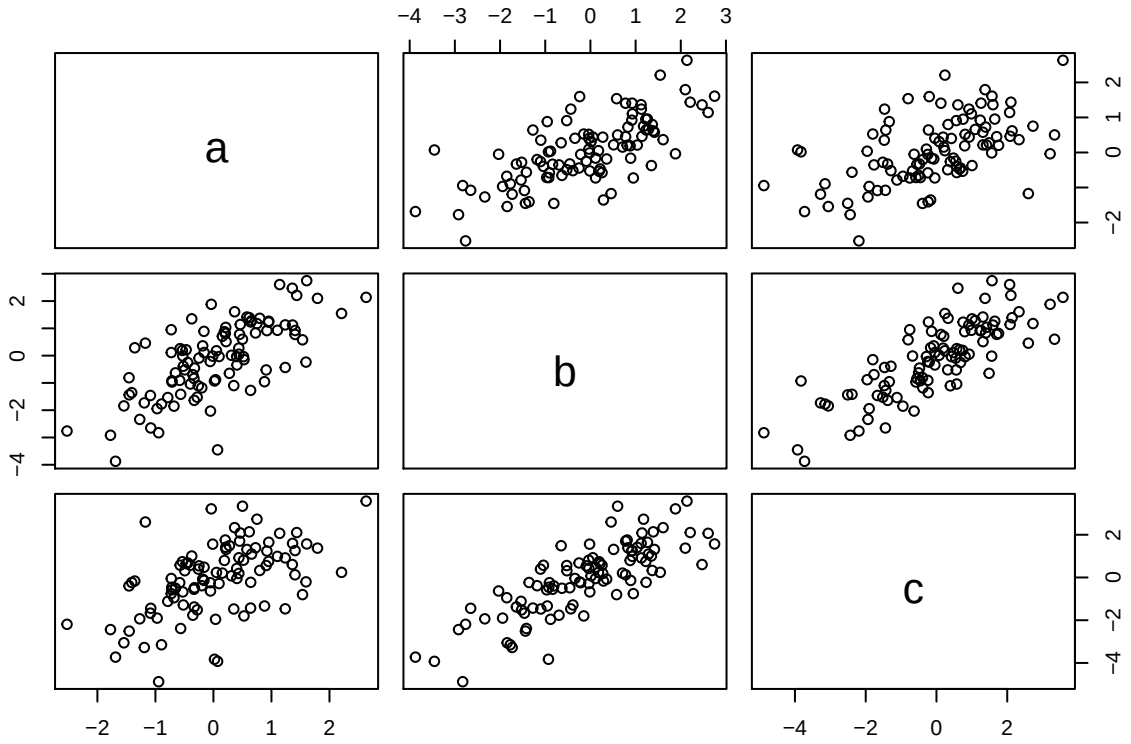
```
x1 <- c(rnorm(100))
x2 <- c(rnorm(100))
x3 <- c(rnorm(100))
t = data.frame(a=x1, b=x1+x2, c=x1+x2+x3)
plot(t)
```



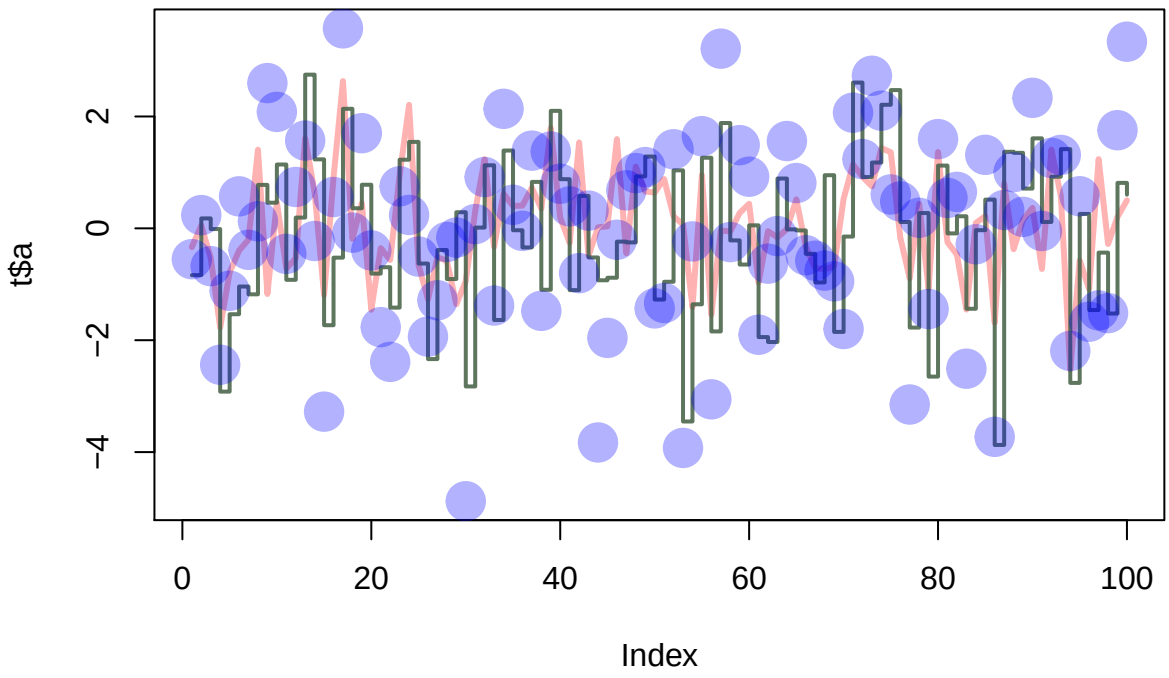
To Do question 9:

Add these lines to the script file of the previous section. Try to find out, either by experimenting or by using the help, what the meaning is of rgb, the last argument of rgb, lwd, pch, cex.

```
x1 <- c(rnorm(100))
x2 <- c(rnorm(100))
x3 <- c(rnorm(100))
t = data.frame(a=x1, b=x1+x2, c=x1+x2+x3)
plot(t)
```




```
plot(t$a, type="l", ylim=range(t),lwd=3, col=rgb(1,0,0,0.3))
lines(t$b, type="s", lwd=2,col=rgb(0.3,0.4,0.3,0.9))
points(t$c, pch=20, cex=4,col=rgb(0,0,1,0.3))
```



To Do question 10:

Make a file called tst1.txt in Notepad from the example in Figure 4 and store it in your working directory. Write a script to read it, to multiply the column called g by 5 and to store it as tst2.txt.

```
eleven <- read.table(file="tst1.txt",header=TRUE)
eleven$g <- eleven$g*5
eleven
```

```
##    a   g  x
## 1  1  10  3
## 2  2  20  6
## 3  4  40 12
## 4  8  80 24
## 5 16 160 48
## 6 32 320 96
```

To Do question 11:

Compute the mean of the square root of a vector of 100 random numbers. What happens?

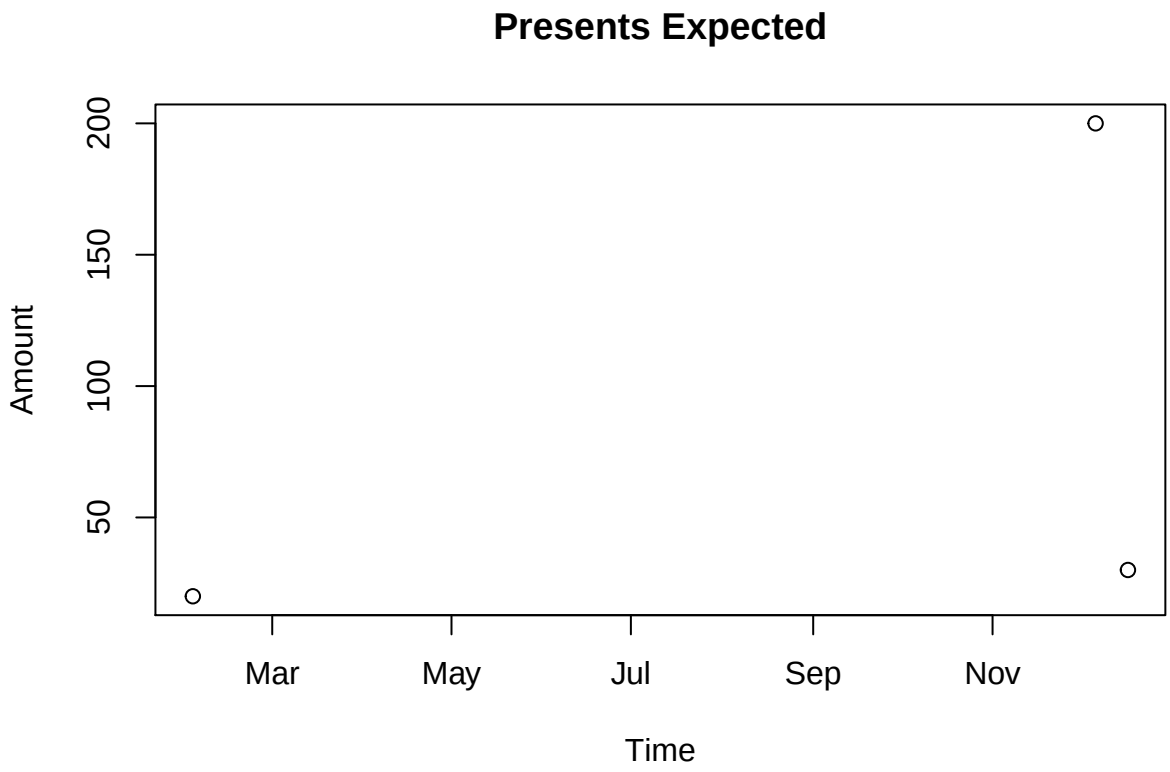
```
ten <- c(rnorm(100))
avgten=mean(ten)
sqrt(avgten)
```

```
## Warning in sqrt(avgten): NaNs produced
## [1] NaN
```

To Do question 12:

Make a graph with on the x-axis: today, Sinterklaas 2014 and your next birthday and on the y-axis the number of presents you expect on each of these days. Tip: make two vectors first.

```
Date1 = strptime(c(20170202, 20171206, 20171217), format = "%Y%m%d")
Date2 = c(20,200, 30)
plot(Date1, Date2, xlab = "Time", ylab = "Amount", main = "Presents Expected")
```



##To Do question 13: #####Make a vector from 1 to 100. Make a for-loop which runs through the whole vector. Multiply the elements which are smaller than 5 and larger than 90 with 10 and the other elements with 0.1.

```
twelve = 1:100

for (i in 1:100) {
   if (twelve[i] < 5 | twelve[i] > 90) {
      twelve[i] = twelve[i] * 10 }
   else {
      twelve[i] = twelve[i] * 0.1}
}
twelve
```

```
##   [1]   10.0   20.0   30.0   40.0    0.5    0.6    0.7    0.8    0.9    1.0
##  [11]    1.1    1.2    1.3    1.4    1.5    1.6    1.7    1.8    1.9    2.0
##  [21]    2.1    2.2    2.3    2.4    2.5    2.6    2.7    2.8    2.9    3.0
##  [31]    3.1    3.2    3.3    3.4    3.5    3.6    3.7    3.8    3.9    4.0
##  [41]    4.1    4.2    4.3    4.4    4.5    4.6    4.7    4.8    4.9    5.0
##  [51]    5.1    5.2    5.3    5.4    5.5    5.6    5.7    5.8    5.9    6.0
##  [61]    6.1    6.2    6.3    6.4    6.5    6.6    6.7    6.8    6.9    7.0
##  [71]    7.1    7.2    7.3    7.4    7.5    7.6    7.7    7.8    7.9    8.0
##  [81]    8.1    8.2    8.3    8.4    8.5    8.6    8.7    8.8    8.9    9.0
##  [91]  910.0  920.0  930.0  940.0  950.0  960.0  970.0  980.0  990.0 1000.0
```

To Do question 14:

Write a function for the previous ToDo, so that you can feed it any vector you like (as argument). Use a for-loop in the function to do the computation with each element. Use the standard R function length in the specification of the counter.

```
Todo14=1:100
fun14 = function(arg1) {
  len = length(arg1)
  for(i in 1:len) {
    if (arg1[i] < 5 | arg1[i] > 90) {
      arg1[i] = arg1[i] * 10
      }
    else{
      arg1[i] = arg1[i] * 0.1
      }
  }
  return (arg1)
}
fun14(arg1=Todo14)
```

```
##   [1]   10.0   20.0   30.0   40.0    0.5    0.6    0.7    0.8    0.9    1.0
##  [11]    1.1    1.2    1.3    1.4    1.5    1.6    1.7    1.8    1.9    2.0
##  [21]    2.1    2.2    2.3    2.4    2.5    2.6    2.7    2.8    2.9    3.0
##  [31]    3.1    3.2    3.3    3.4    3.5    3.6    3.7    3.8    3.9    4.0
##  [41]    4.1    4.2    4.3    4.4    4.5    4.6    4.7    4.8    4.9    5.0
##  [51]    5.1    5.2    5.3    5.4    5.5    5.6    5.7    5.8    5.9    6.0
##  [61]    6.1    6.2    6.3    6.4    6.5    6.6    6.7    6.8    6.9    7.0
##  [71]    7.1    7.2    7.3    7.4    7.5    7.6    7.7    7.8    7.9    8.0
##  [81]    8.1    8.2    8.3    8.4    8.5    8.6    8.7    8.8    8.9    9.0
##  [91]  910.0  920.0  930.0  940.0  950.0  960.0  970.0  980.0  990.0 1000.0
```

To Do question 15:

Actually, people often use more for-loops than necessary. The ToDo above can be done more easily and quickly without a for-loop but with regular vectorcomputations.

```
Todo15 <- function(fun15) {
  return(ifelse
         (
           (fun15 < 5 | fun15 > 90)
           ,fun15*10,fun15*0.1
```

```
            )
          )
}
```

Sources

http://stackoverflow.com/questions/10933945/how-to-calculate-the-euclidean-norm-of-a-vector-in-r

https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf

https://nicercode.github.io/guides/reports/

http://kbroman.org/knitr_knutshell/pages/markdown.html

http://kbroman.org/knitr_knutshell/pages/Rmarkdown.html

https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf