SRT411A0

Huda Ali

February 15, 2019

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.

```
(2017-2014)/(2014-1999) *100
```

```
## [1] 20
```

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=2014
b=2017
c=1999
d=(b-a)/(a-c) *100
d
```

```
## [1] 20
```

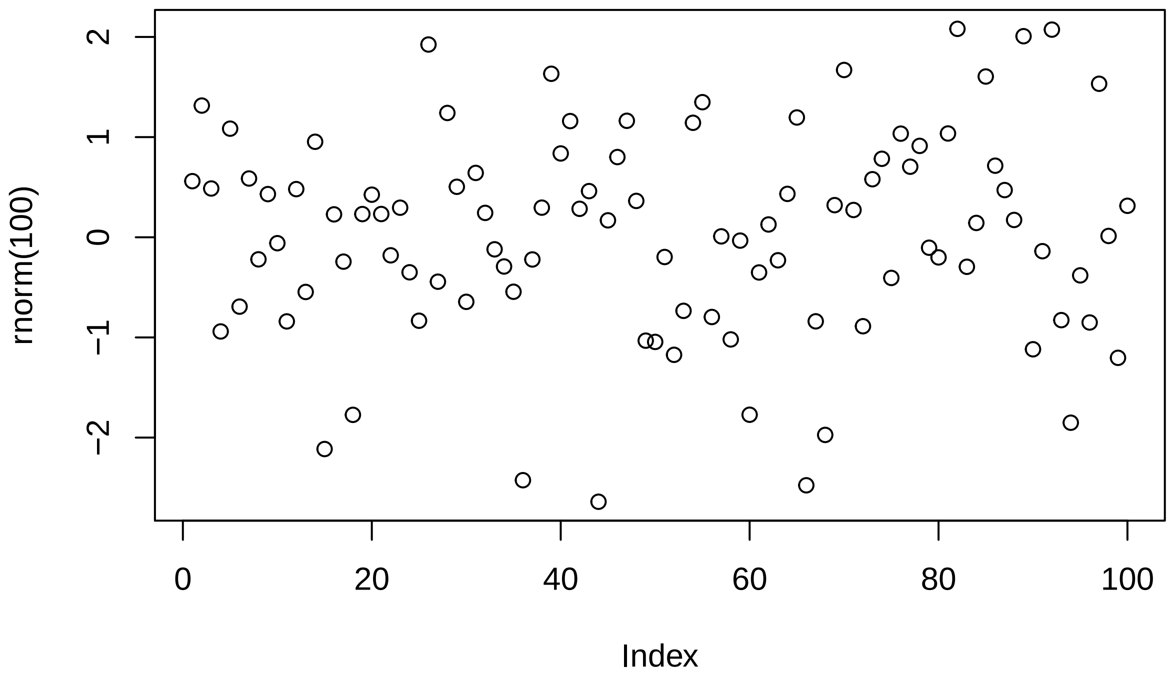
3) Compute the sum of 4, 5, 8 and 11 by first combining them into a vector and then using the function sum.

```
a=c(4,5,8,11)
sum(x=a)
```

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

4) Plot 100 normal random numbers.

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

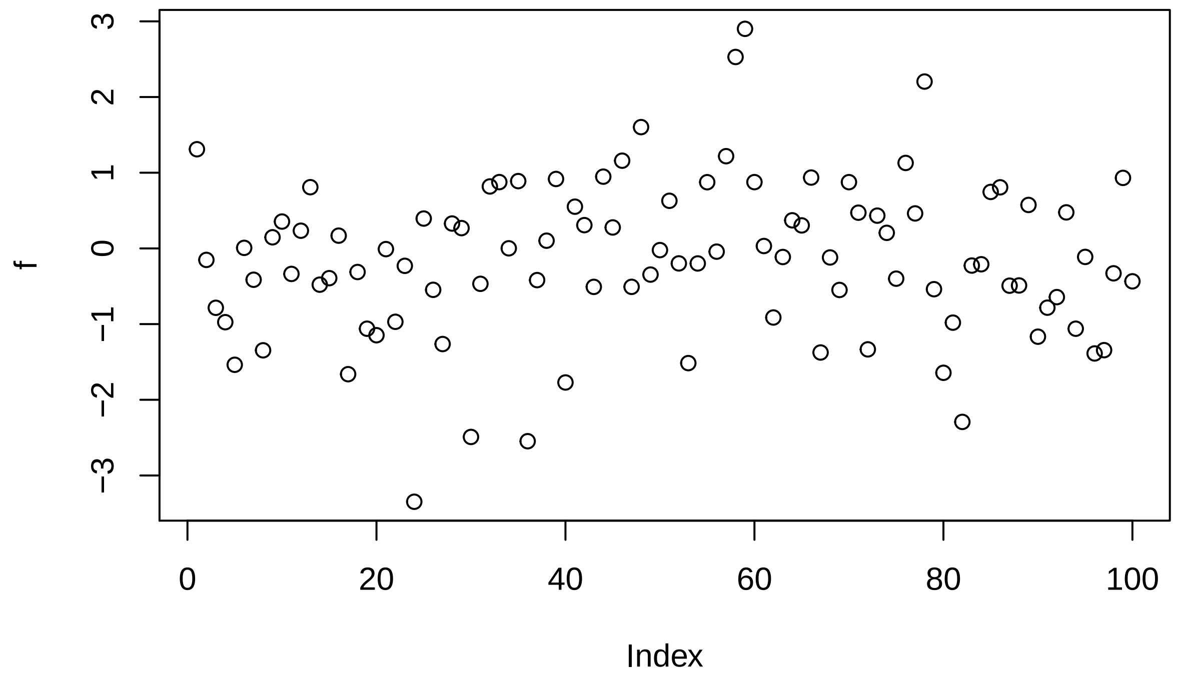


5) Find help for the sqrt function.

```
help(sqrt)
```

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

```
# generate 100 random numbers and plot them
   f = rnorm(100)
   plot(f)
```



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.

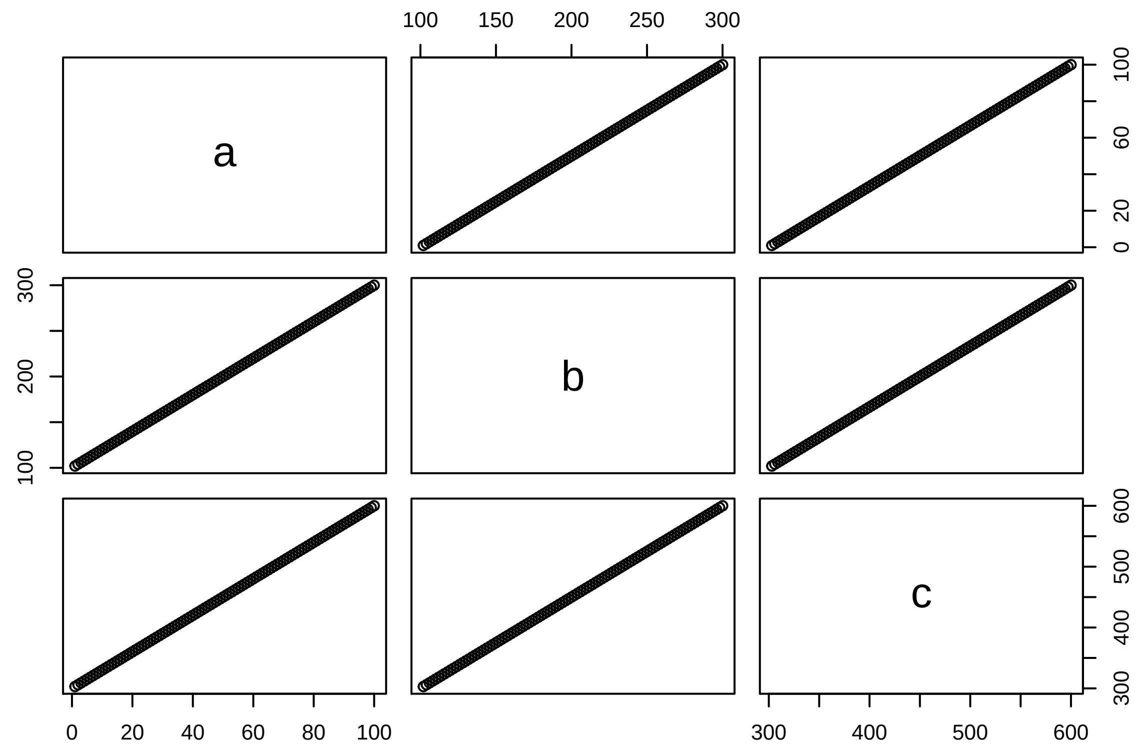
```
P = seq(from=31, to=60, by=1)
Q=matrix(data=P,ncol=5, nrow=6)
Q
```

```
##      [,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
```

8) Make a script file which constructs three random normal vectors of length 100. Call the vectors x1, x2 and x3. Make a data frame called "t" 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.

```
#create 3 vectors of normal numbers with length 100
x1=seq(from=1, to=100, by=1)
x2=seq(from=101, to=200, by=1)
x3=seq(from=201, to=300, by=1)
#create data frame with columns a,b and c
t = data.frame(a = x1,b = x1+x2, c = x1+x2+x3)
#call the plot and sd functions
```

```r
plot(t)
```

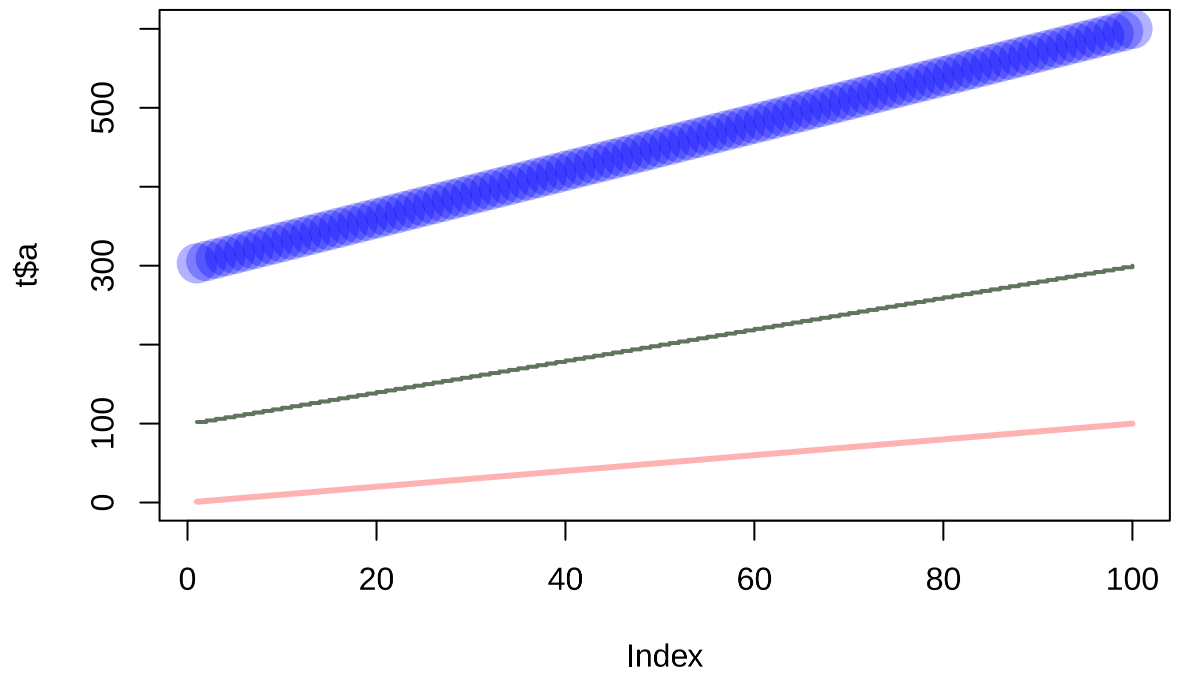


```r
sd=t
```

This graph shows us the out put of column a, then column a and b, and lastly, column a,b and c which is equivalent to the x1, x1+x2, and x1+x2+x3.

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.

```r
#create 3 vectors of normal numbers with length 100
x1=seq(from=1, to=100, by=1)
x2=seq(from=101, to=200, by=1)
x3=seq(from=201, to=300, by=1)
#create data frame with columns a,b and c
t = data.frame(a = x1,b = x1+x2, c = x1+x2+x3)
#call the plot function with the following graph attributes
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))
```



The rgb function is used here to give the different shades of blue, green, and red based on the values provided in the code.

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.

```
#script that reads table in tst1.txt, multiplies column g by 5 and stores in file called tst2.txt
d = read.table(file="tst1.txt",header=TRUE)
write.table(d$g *5, file="tst2.txt",row.names=FALSE)
read.table(file="tst2.txt",header=TRUE)
```

```
##    x
## 1 15
## 2 20
## 3 25
```

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

```
vec1=rnorm(100)
x=sqrt(vec1)
```

```
## Warning in sqrt(vec1): NaNs produced
```

```
mean(x,na.rm=TRUE)
```

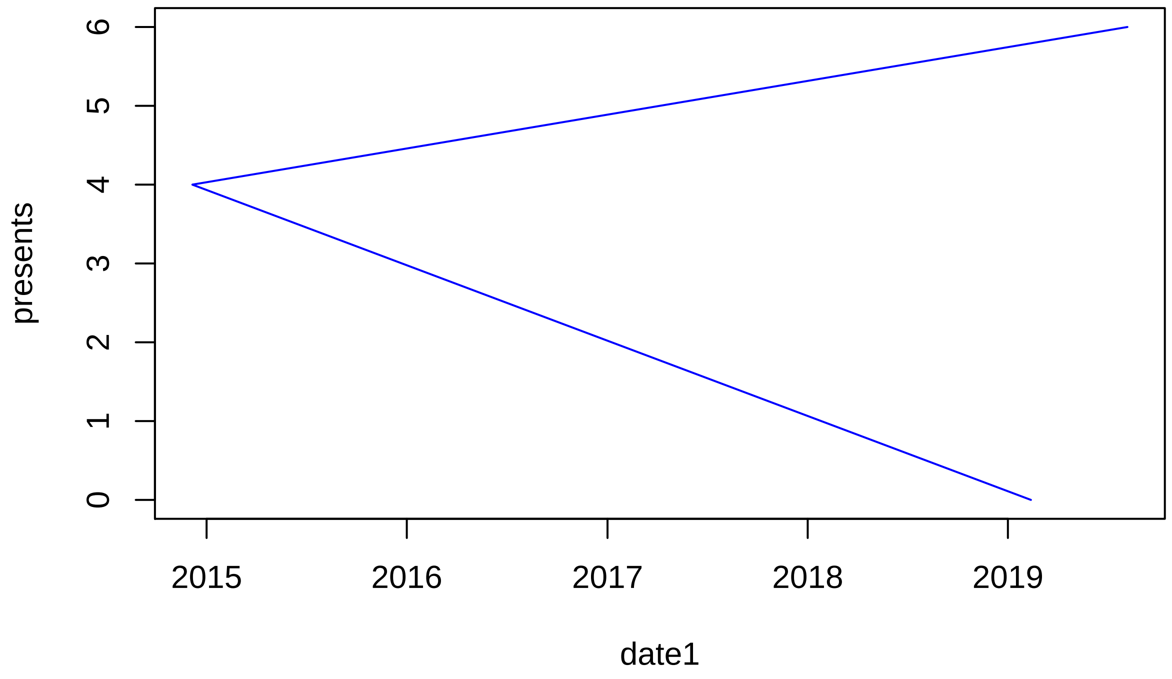
```
## [1] 0.8491423
```

When we calculate the square root and mean of 100 random numbers, we get NaNs, however if we enter "na.rm=TRUE" it will still provide a mean value.

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("20190212000000","20141206000000", "20190807000000"),format="%Y%m%d%H%M%S")
presents=c(0,4,6)
plot(x=date1, y=presents, type="l", col="blue")
```



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.

```
   h = seq(from=1, to=100, by=1)
   s = c()
   for (i in 1:100)
   {
      if ( h[i] < 5 )
      {
        s[i] = h[i] * 10
      }
      else if ( h[i] > 90)
      {
        s[i] = h[i] * 10
      }
      else
      {
        s[i]= h[i]*0.1;
      }
    }
    s
```

```
##   [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
```

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.

```
fun= function(arg1,arg2)
{
  h[i]=arg1[i];
  for(i in length(h))
  {

  }
}
```

15) The final Todo in the document has a footnote. Write code that will prove that footnote true. Footnote: "The ToDo above can be done more easily and quickly without a for-loop but with regular vector-computations."

```
h=c(1:100)
h[1:4] = h[1:4]*10
h[91:100] = h[91:100]*10
h[5:90] = h[5:90]*0.1
h
```

```
##   [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
```