SRT411A0

Abhayjot Ressi

February 2, 2017

SRT411: Digital Data Analysis - Winter 2017

Assignment 0

Introduction

My name is Abhayjot Ressi. You can find me on GitHub as AbhayjotRessi. At the time this assignment is being written, I am a second year student in the Informatics and Security Program at Seneca College. This is Assignment 0 for SRT411. This assignment is based on the document "A (Very) Short Introduction to R" by Paul Torfs & Claudia Brauer Can be found here. The main idea of this document is to teach the basics of the R programming language though a tutorial combined with "To Do" sections at the end of each topic. The questions, code and results shown below are the results of my experience with this document. The final To Do is not a normal To Do section but an additional practice of your skill and was part of the requirements of this assignment as set by the professor.

To Do 1

Compute the difference between 2017 and the year you started at this university and divide this by the difference between 2017 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 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

```
collegeyear <- (2017-2015)
birth <- (2017-1997)
collegeyear/birth*100
```

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

To Do 3

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

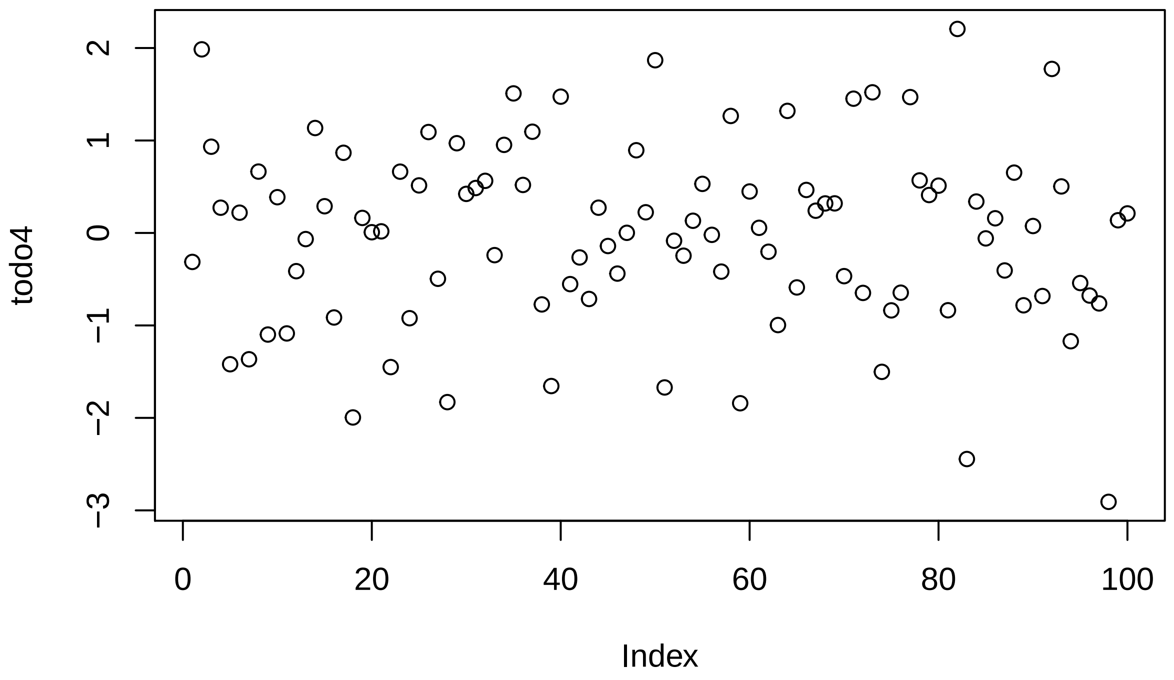
```
sum(x <- c(4,5,8,11))
```

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

To Do 4

Plot 100 normal random numbers

```
todo4 <- rnorm(100)
plot(todo4)
```



To Do 5

Find help for the sqrt function

```
help(sqrt)
```

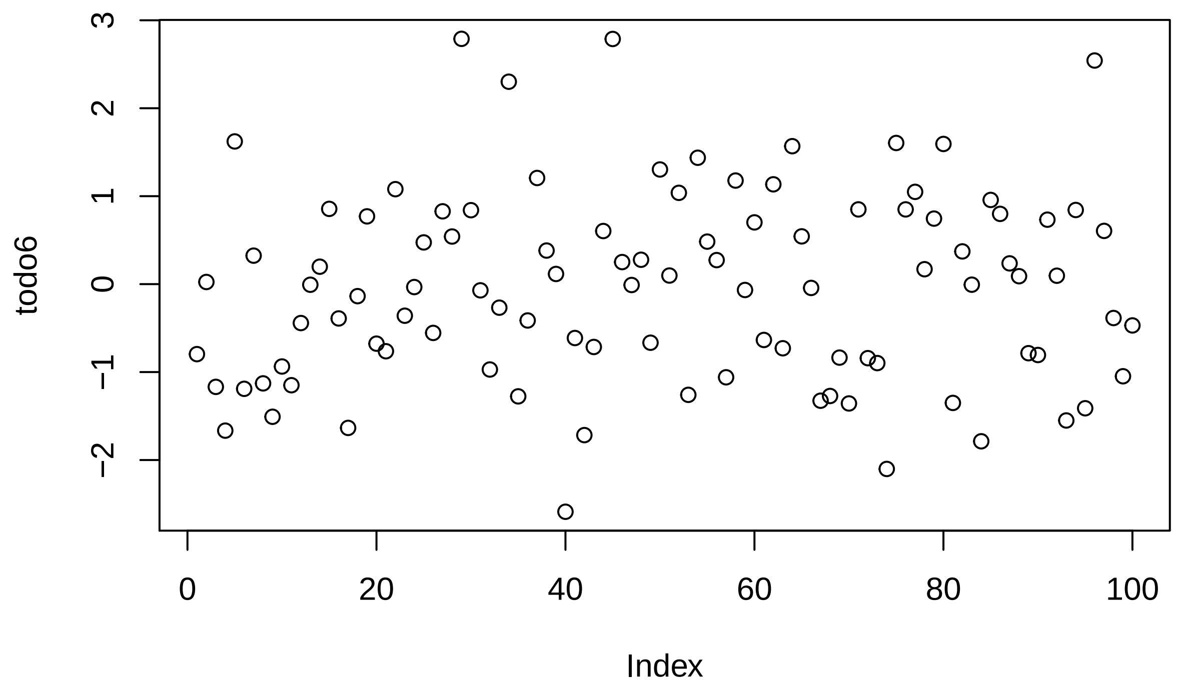
```
## starting httpd help server ...
```

```
##  done
```

To Do 6

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

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



The file “firstscript.R” is located in my working directory. The contents of the that file is the following:

```
#To Do 6

#todo6 <- rnorm(100)
#plot(todo6)
```

To Do 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.

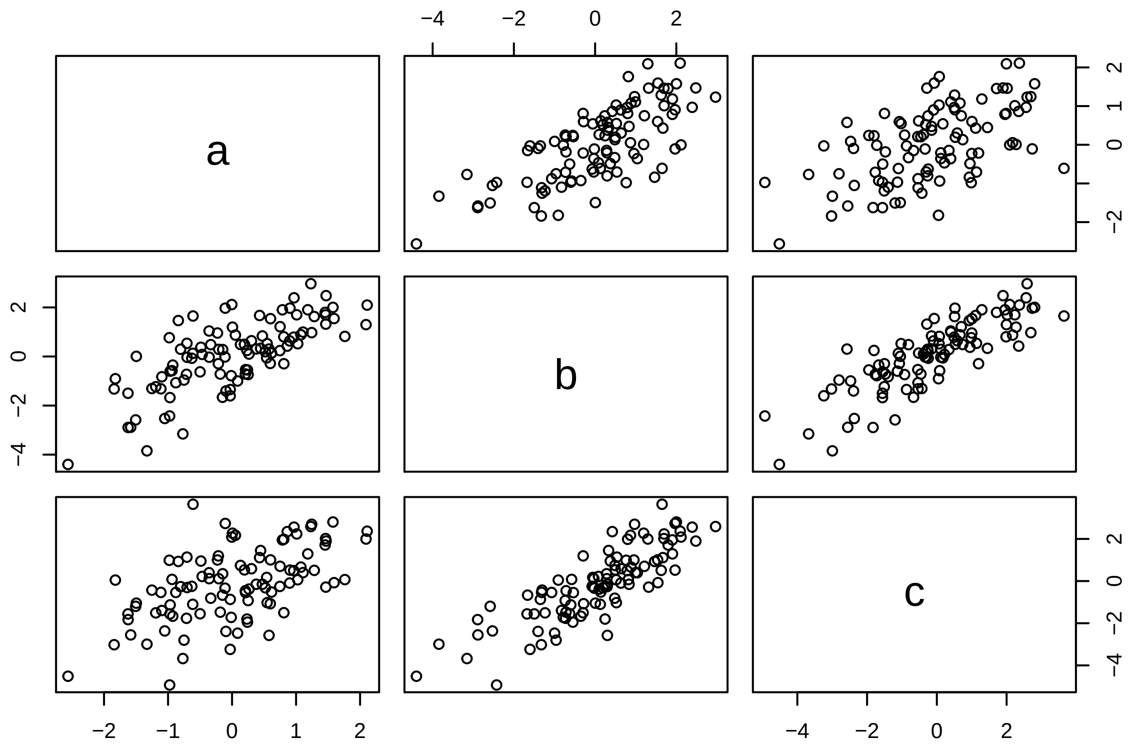
```
P <- seq(31:60)
Q <- matrix(data=P,ncol=5,nrow=6)
Q
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    7   13   19   25
## [2,]    2    8   14   20   26
## [3,]    3    9   15   21   27
## [4,]    4   10   16   22   28
## [5,]    5   11   17   23   29
## [6,]    6   12   18   24   30
```

To Do 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 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.

```
source("todo8.R")
```



The file "todo8.R" is located in my working directory. The contents of the that file is the following:

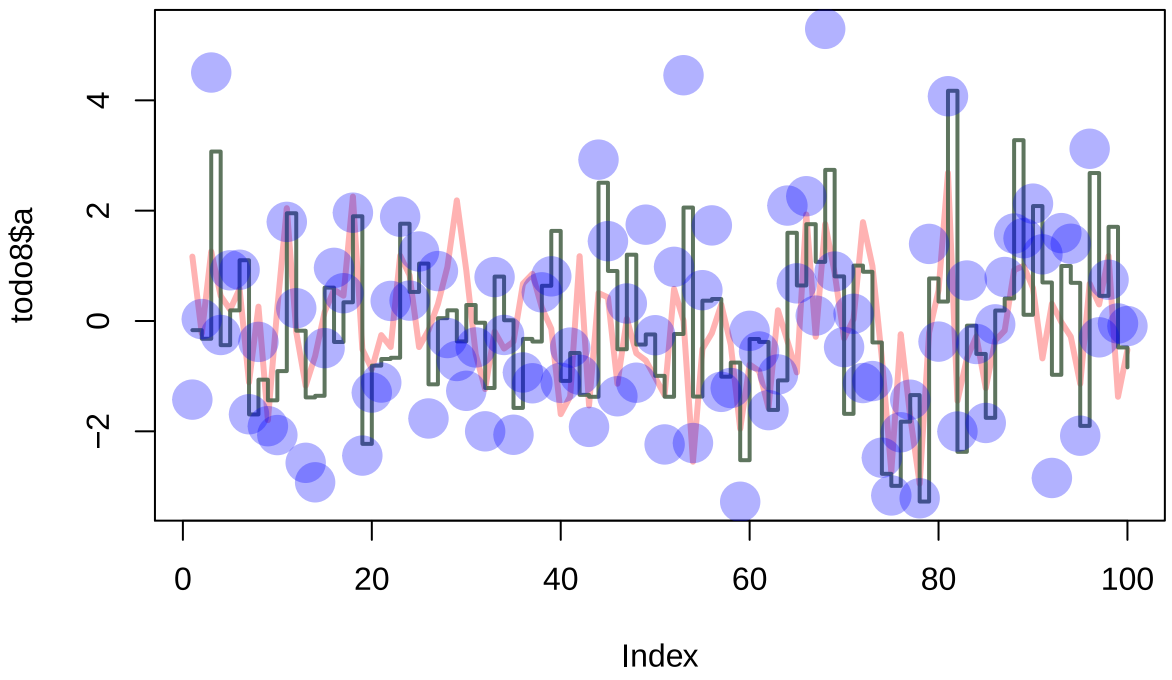
```
#TO DO 8

#x1 <- c(rnorm(100))
#x2<- c(rnorm(100))
#x3<- c(rnorm(100))
#todo8 = data.frame(a=x1,b=x1+x2,c=x1+x2+x3)
#plot(todo8)
```

To Do 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

```
source("todo9.R")
```



The file "todo9.R" is located in my working directory. The contents of the that file is the following:

```
#TO DO 9

#x1 <- c(rnorm(100))
#x2<- c(rnorm(100))
#x3<- c(rnorm(100))
#todo8 = data.frame(a=x1,b=x1+x2,c=x1+x2+x3)
#plot(todo8$a, type="l", ylim=range(todo8),lwd=3, col=rgb(1,0,0,0.3))
#lines(todo8$b, type="s", lwd=2,col=rgb(0.3,0.4,0.3,0.9))
#points(todo8$c, pch=20, cex=4,col=rgb(0,0,1,0.3))
#?rgb
```

To Do 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.

```
todo10 <- read.table(file="tst1.txt", header=TRUE)
todo10$g <- todo10$g*5
write.table(todo10,file="tst2.txt", row.names=FALSE)
```

The contents of tst1.txt:

```
#a g x
#1 2 3
#2 4 6
```

```
#4 8 12
#8 16 24
#16 32 48
#32 64 96
```

The contents of tst2.txt after running the script:

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

To Do 11

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

```
todo11 <- mean(sqrt(c(rnorm(100))))
```

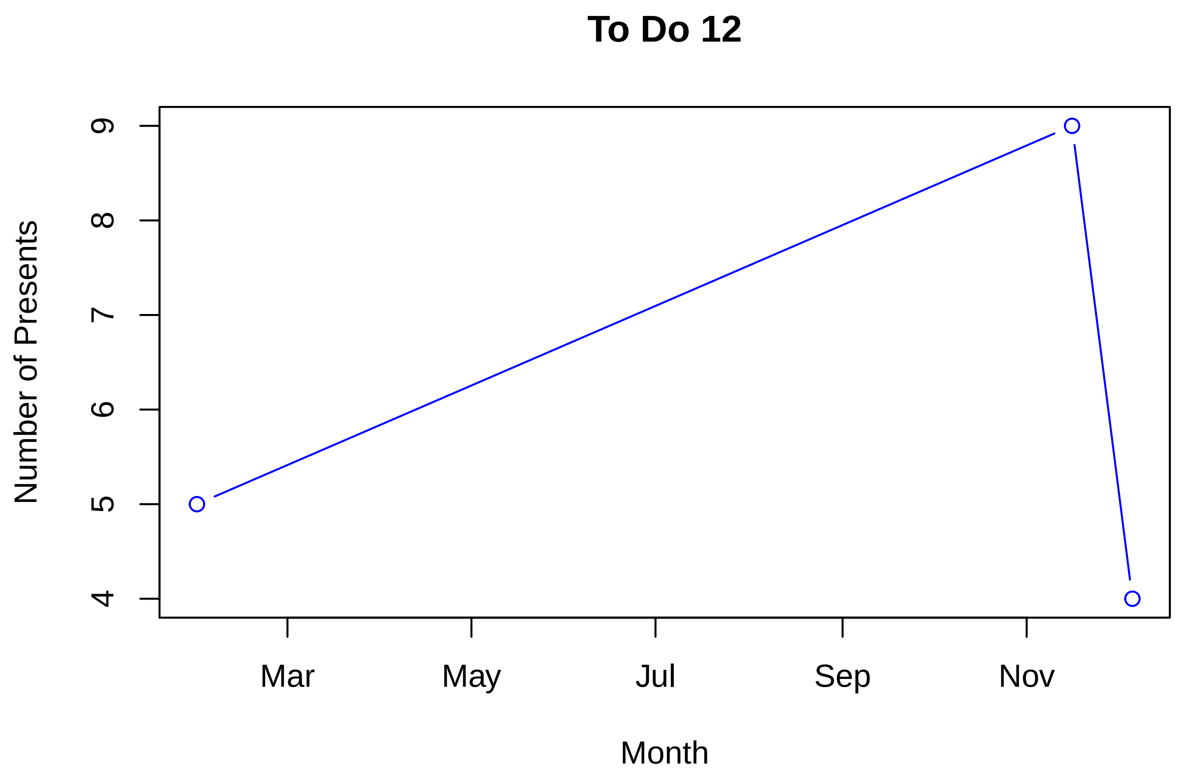
```
## Warning in sqrt(c(rnorm(100))): NaNs produced
```

The code above produces a warning message saying NaNs produced. Based on the To Do asking what happens, I believe that the warning message is to be expected.

To Do 12

Make a graph with on the x-axis: today, Sinterklaas 2017 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.

```
birthklass=strptime( c("20170130","20171116","20171206"),format="%Y%m%d")
presentsreceived=c(5,9,4)
plot( birthklass,presentsreceived, type = "b", col = "Blue", xlab = "Month",ylab="Number of Presents",ma
```



To Do 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

```
todo13 = 1:100
for(i in 1:100)
{
  if (todo13[i] < 5 | todo13[i] > 90)
  {
    todo13[i] = todo13[i] * 10
  } else
  {
    todo13[i] = todo13[i] * 0.1
  }
}
todo13
```

```
##   [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 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
functionfortodo14 = function(value1)
{
  l = length(value1)
  for(i in 1:l)
  {
    if (value1[i] < 5 | value1[i] > 90)
    {
      value1[i] = value1[i] * 10
    } else
    {
      value1[i] = value1[i] * 0.1
    }
  }
  return (value1)
}
functionfortodo14(value1=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 15 (Footnote)

Actually, people often use more for-loops than necessary. The ToDo above (ToDo 14) can be done more easily and quickly without a for-loop but with regular vector computations.

```
todo15 <- c(1:100)
ifelse(todo15 < 5 | x > 90, x * 10, x * 0.1)
```

```
##   [1]  40.0  50.0  80.0 110.0   0.4   0.5   0.8   1.1   0.4   0.5   0.8
##  [12]   1.1   0.4   0.5   0.8   1.1   0.4   0.5   0.8   1.1   0.4   0.5
##  [23]   0.8   1.1   0.4   0.5   0.8   1.1   0.4   0.5   0.8   1.1   0.4
##  [34]   0.5   0.8   1.1   0.4   0.5   0.8   1.1   0.4   0.5   0.8   1.1
##  [45]   0.4   0.5   0.8   1.1   0.4   0.5   0.8   1.1   0.4   0.5   0.8
##  [56]   1.1   0.4   0.5   0.8   1.1   0.4   0.5   0.8   1.1   0.4   0.5
##  [67]   0.8   1.1   0.4   0.5   0.8   1.1   0.4   0.5   0.8   1.1   0.4
##  [78]   0.5   0.8   1.1   0.4   0.5   0.8   1.1   0.4   0.5   0.8   1.1
##  [89]   0.4   0.5   0.8   1.1   0.4   0.5   0.8   1.1   0.4   0.5   0.8
## [100]   1.1
```

Sources and Acknowledgement

Short Intro to R

An Introduction to R

R Markdown Cheatsheet

R Markdown Database

Nice R Code

Markdown

R Markdown

Sharing Data Science Portfolio

StackOverflow - Global Variables in R

Plotting Date-Time Graphs

Basic Dataframe Manipulations in R