SRT411A0

Volodymyr Nahirnyi, Github username: VolodymyrNahirnyi

February 16, 2018

This assignment is based on the excercises from "A (very) short introduction to R", available at

https://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf

RStudio was used to write the code, the document was created using Knitr engine that generated a Markdown report. This report was then converted to PDF with Pandoc converter. This documends includes several R code chunks and their results (data or charts). The additional resources that were used are:

https://rmarkdown.rstudio.com/

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

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

ToDo 1

```
(2016-2014)/(2014-2000)*100
```

```
## [1] 14.28571
```

ToDo 2

```
a <- 2016
b <- 2014
c <- 2000
d <- 100
(2016-2014)/(2014-2000)*100
```

```
## [1] 14.28571
```

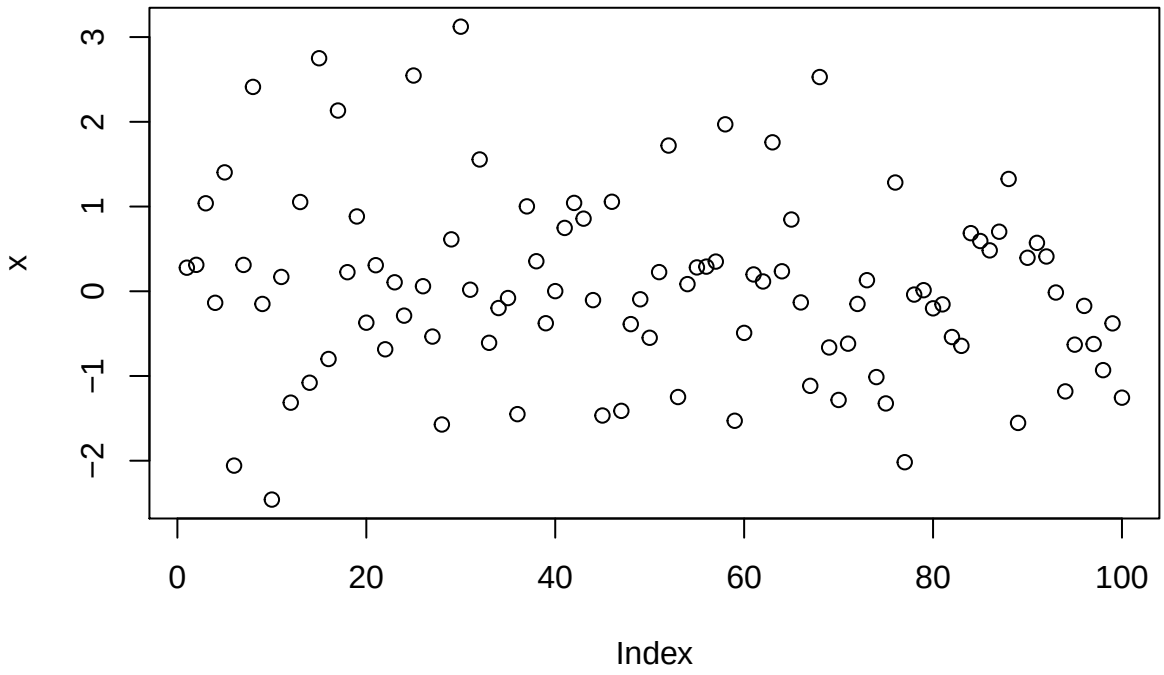
ToDo 3

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

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

ToDo 4

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

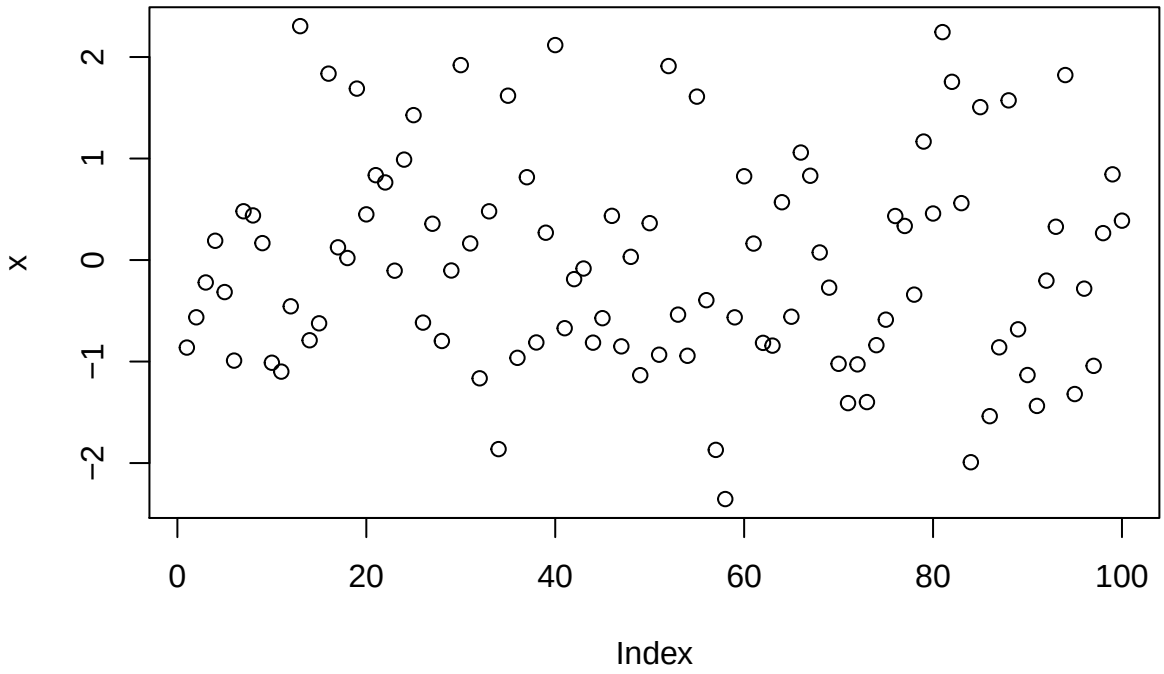


ToDo 5

```
help(sqrt)
```

ToDo 6

```
x <- rnorm(100)
write(plot(x), file="filescript.R")
```



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

ToDo 7

```
P = seq(from=31, to=60, by=1)
P
```

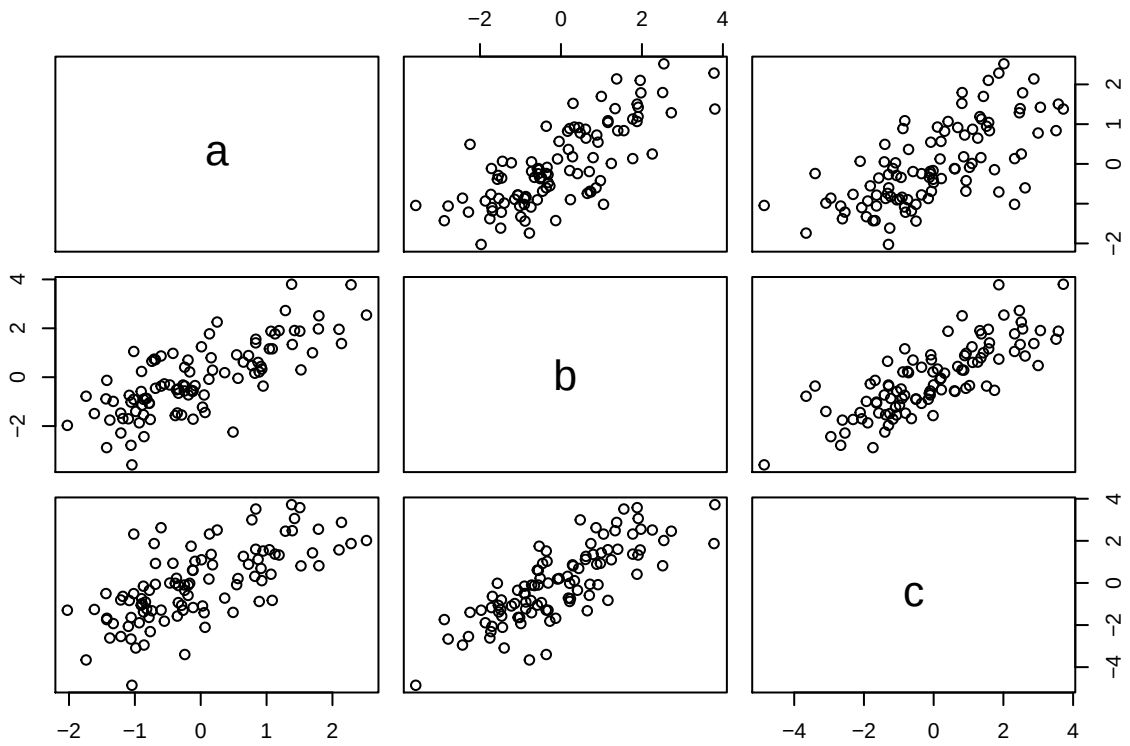
```
##  [1] 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
## [24] 54 55 56 57 58 59 60
```

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

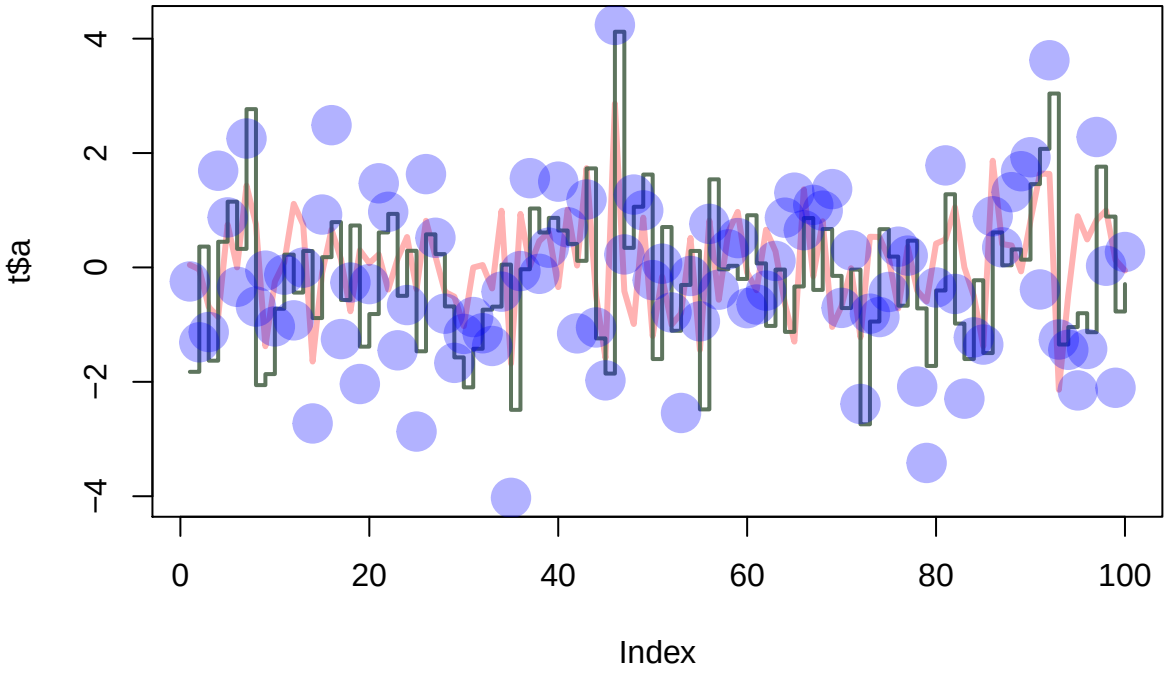
ToDo 8

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



ToDo 9

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



ToDo 10

```
d <- data.frame(a = c(1,2,4,8,16,32), g = c(2,4,8,16,32,64), x = c(3,6,12,24,48,96))
write.table(d, file="tst1.txt", row.names=FALSE)
d2 <- read.table(file="tst1.txt", header=TRUE)
d2$g <- d2$g * 5
write.table(d2, file="tst2.txt", row.names=FALSE)
```

ToDO 11

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

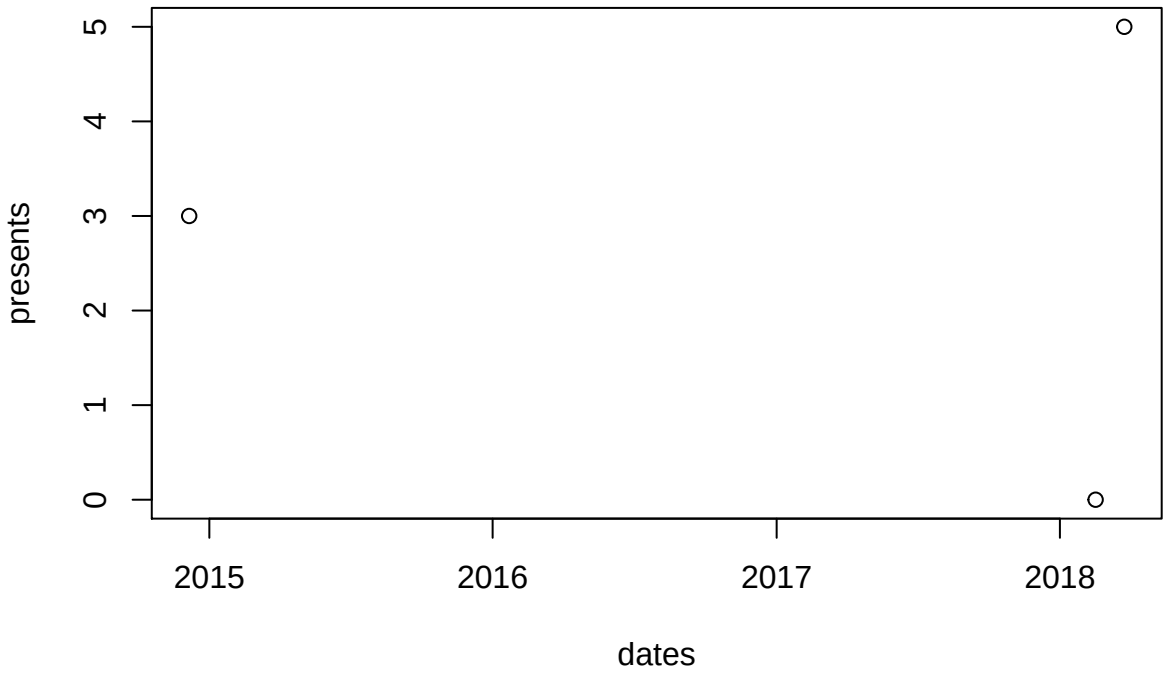
```
## Warning in sqrt(v): NaNs produced
```

```
## [1] NaN
```

There is no square root of a negative number, therefore negative numbers from this vector are not available and we get this output.

ToDo 12

```
dates <- strptime( c("20180216", "20141206", "20180325"), format="%Y%m%d")
presents <- c(0,3,5)
plot(dates,presents)
```



ToDo 13

```
v <- seq(from=1, to=100)
  for(e in v)
  {
    if(e < 5 | e > 90)
    {v[e] <- e * 10}
    else{
      v[e] <- e * 0.1
    }
  }
v
```

```
##   [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
```

ToDo 14

```
fun = function(arg1)
  {
  l = length(arg1)
  for(e  in 1:l)
    {
      if(e < 5 | e > 90){
        arg1[e] <- e * 10
      }
      else{
        arg1[e] <- e * 0.1
      }
    }
  return(arg1)
  }
fun(arg1 = v)
```

```
##   [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
```

ToDo 15