Assignment 0 Intro to R and Github

Ishmael Guarin and Antonio Milutin; username: iguarin

2018-02-16

This R Markdown document is based on the R Tutorial called "A (very) short introduction to R" by Paul Torfs & Claudia Brauer" (https://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf)

```
(2018-2016)/(2018-1997) * 100
```

```
## [1] 9.52381
```

The above code calculates the percentage of your life you have spent at this University. The year 2016 is the year I started studying at Seneca College and 1997 is my year of birth

```
a=2018
birthyear=1997
startyearcollege=2016
d=100
(a-startyearcollege)/(a-birthyear)*d
```

```
## [1] 9.52381
```

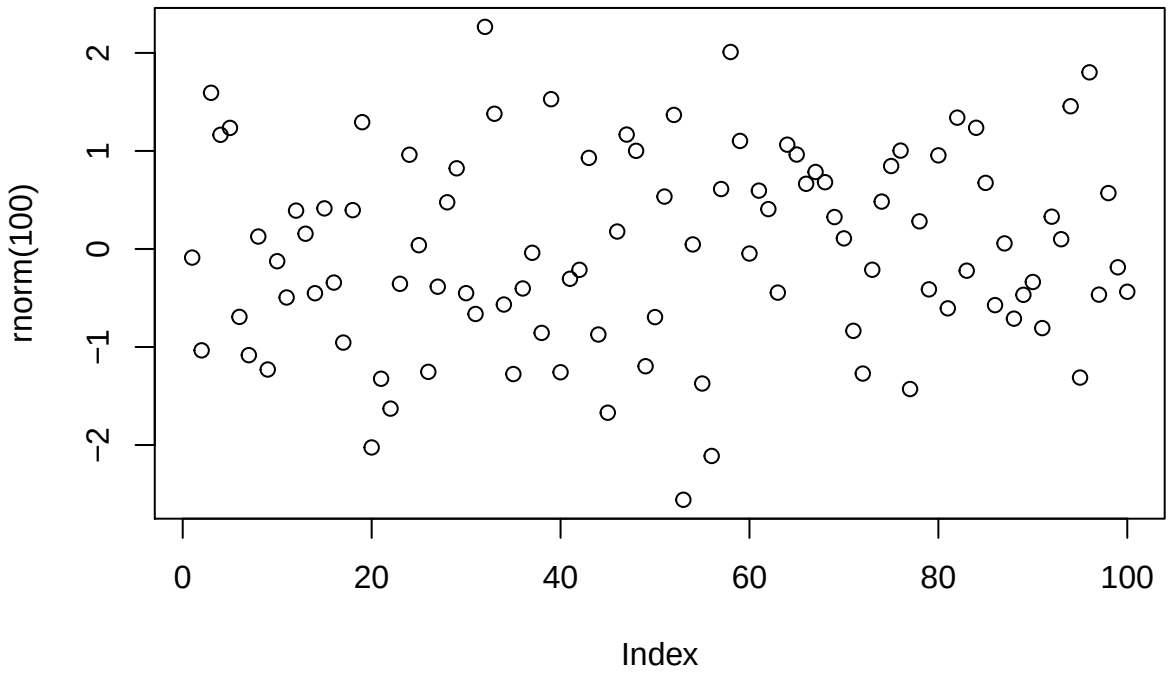
This is the same concept as the previous ToDo but now with the use of variables.

```
b = c(4,5,8,11)
sum(b)
```

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

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

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

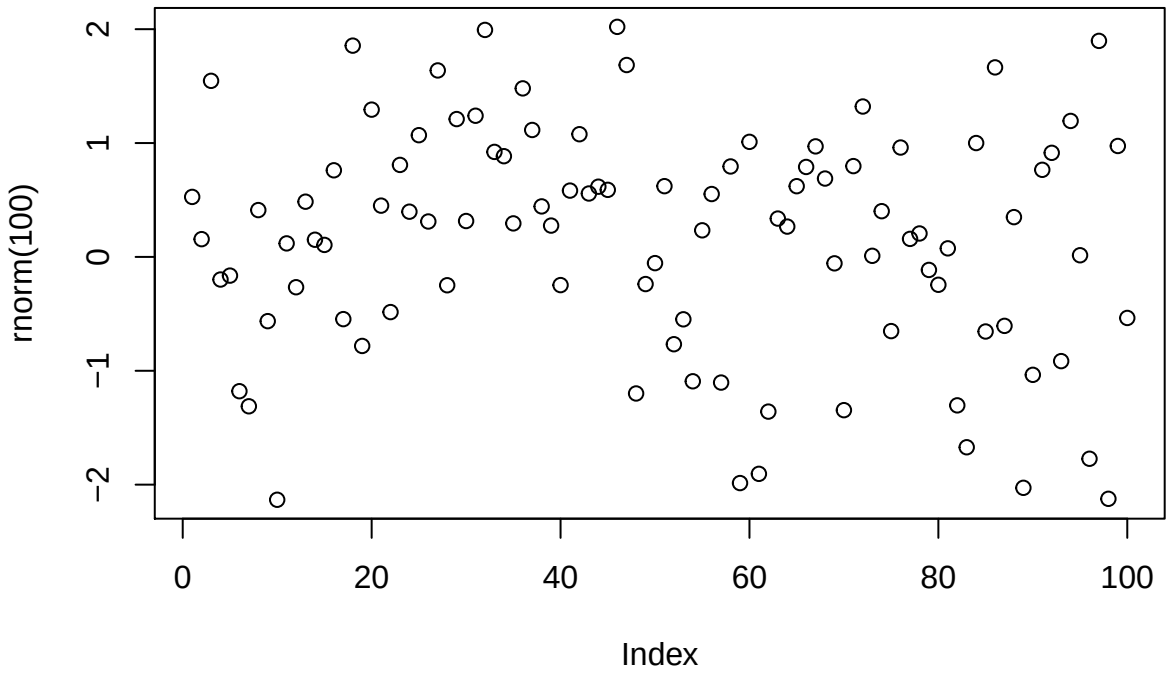


Plot 100 normal random numbers

```
help(sqrt)
```

Finding help for the sqrt function. Alternatively, we can also type ?srt to display the help page. Also, we can run scripts that have R code inside them.

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



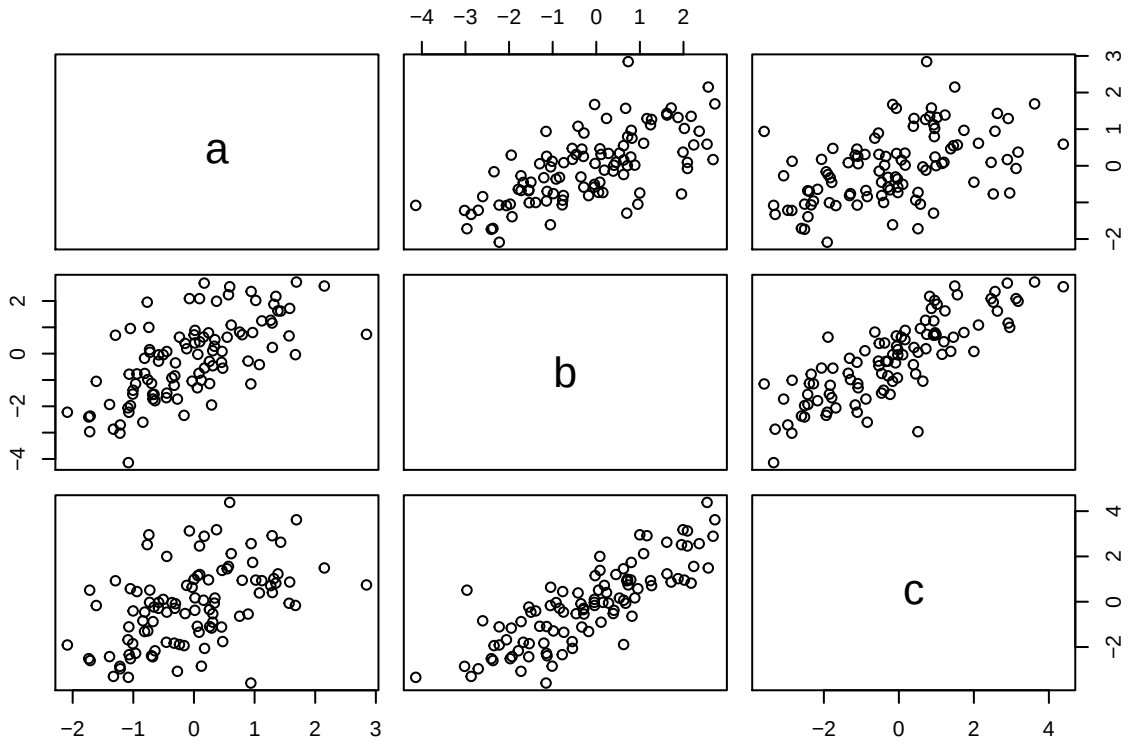
The firstcript.R contains the code above.

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

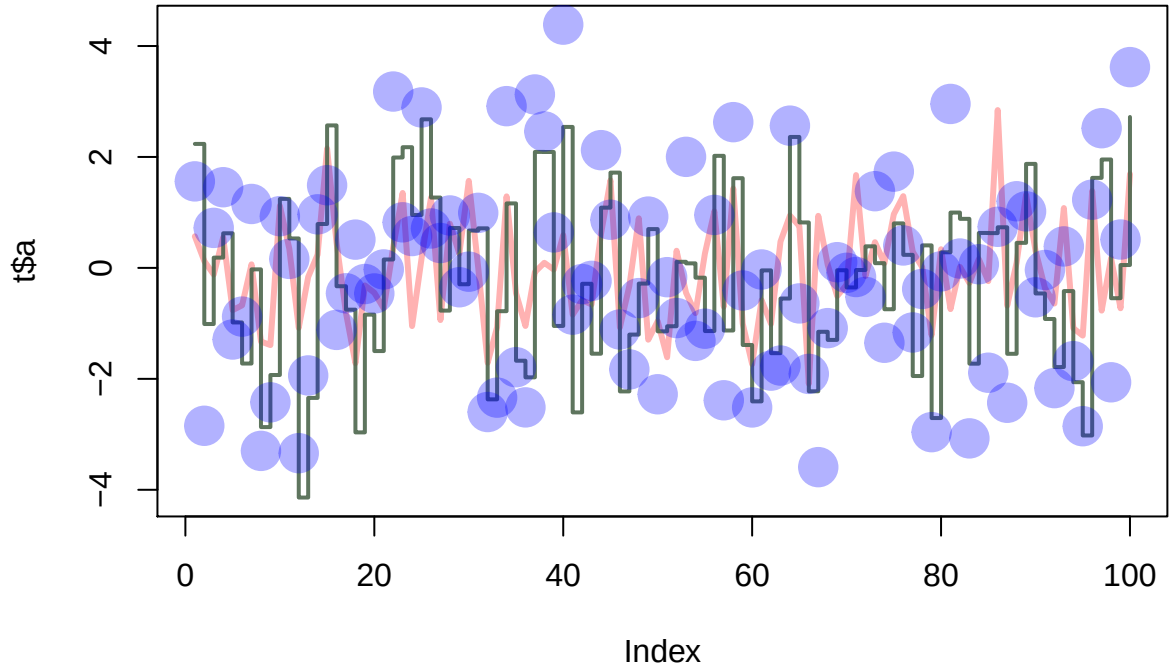
This produces a sequence of numbers from 31 to 60 with an increment of 1. Then we created a matrix called Q which arranges the numbers in 5 columsna nd 6 rows.

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



This code assignes 100 random numbers to x1, x2, and x3. Then we created a data frame called t with three columns.

```
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))
```



rgb stands for red, blue and green. The three values of rgb represent the value of each colour. lwd represent the width of the line(thickness).cex tells you how big to magnify text. pch option is used to specify what symbols to use when plotting points.

```
read_tst1txt = read.table( file="tst1.txt", header = TRUE)
write.table(read_tst1txt$g * 5, file="tst2.txt", row.names=FALSE)
read.table(file="tst2.txt")
```

```
##    V1
## 1   x
## 2  10
## 3  20
## 4  40
## 5  80
## 6 160
## 7 320
```

We made a file called tst1.txt that has columns "a", "g" and "x". We read the file and multiplied the column "g" by 5 and stored the results to a file called tst2.txt.

```
mean(sqrt(rnorm(100)))
```

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

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

rnorm might have generated some NaN numbers.

```
date1=strptime( c("20180216","20181220"),format="%Y%m%d")
present=c(10,2)
date1
```

```
## [1] "2018-02-16 EST" "2018-12-20 EST"
```

Presents the relationship of the current date to my next birthday and the number of presents.

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

We generated a set of numbers from 1 to 100 then we multiplied the elements which are smaller than 5 and larger than 90 with 10. Other elements are multiplied with 0.1.

```
k = 1:10
function1 = function(arg1)
{
l = length(arg1)
for( i in 1:l)
{
if(arg1[i] < 5 | arg1[i] >90)
{
arg1[i] = arg1[i] * 10
} else
{
arg1[i] = arg1[i] *0.1
}
}
return(arg1)
}
function1(arg1=k)
```

```
##  [1] 10.0 20.0 30.0 40.0  0.5  0.6  0.7  0.8  0.9  1.0
```

This is the same as the previous ToDo but now we can specify the values with the variable "k".

References: http://rmarkdown.rstudio.com/ http://nicercode.github.io/guides/reports/ http://kbroman.org/knitr_knutshell/pages/markdown.html http://kbroman.org/knitr_knutshell/pages/Rmarkdown.html https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf