Assignment 0, SRT411SAA, Prof. Lisa Li

Rohan Saeed

2019-02-05

Introduction

In this assignment I will be completing the exercises in the book **A Short Introduction to R** and publishing the report and code to Github. The document has 14 “To do” sections which are completed and shown according to the book. This document includes both code and also the output of embedded R code. The document has also been uploaded to Github; **account name:** “rsaeed2@myseneca.ca”.

To Do Sections and Results

3.1

Percentage of life spent at Seneca College

```
((2019-2017)/(2019-1995))*100
```

```
## [1] 8.333333
```

3.2

Percentage of life spent at Seneca College (using variables)

```
cur_yr = 2019
start_yr = 2017
brn_yr = 1995
((cur_yr-start_yr)/(cur_yr-brn_yr))*100
```

```
## [1] 8.333333
```

3.4

Calculate sum of a vector

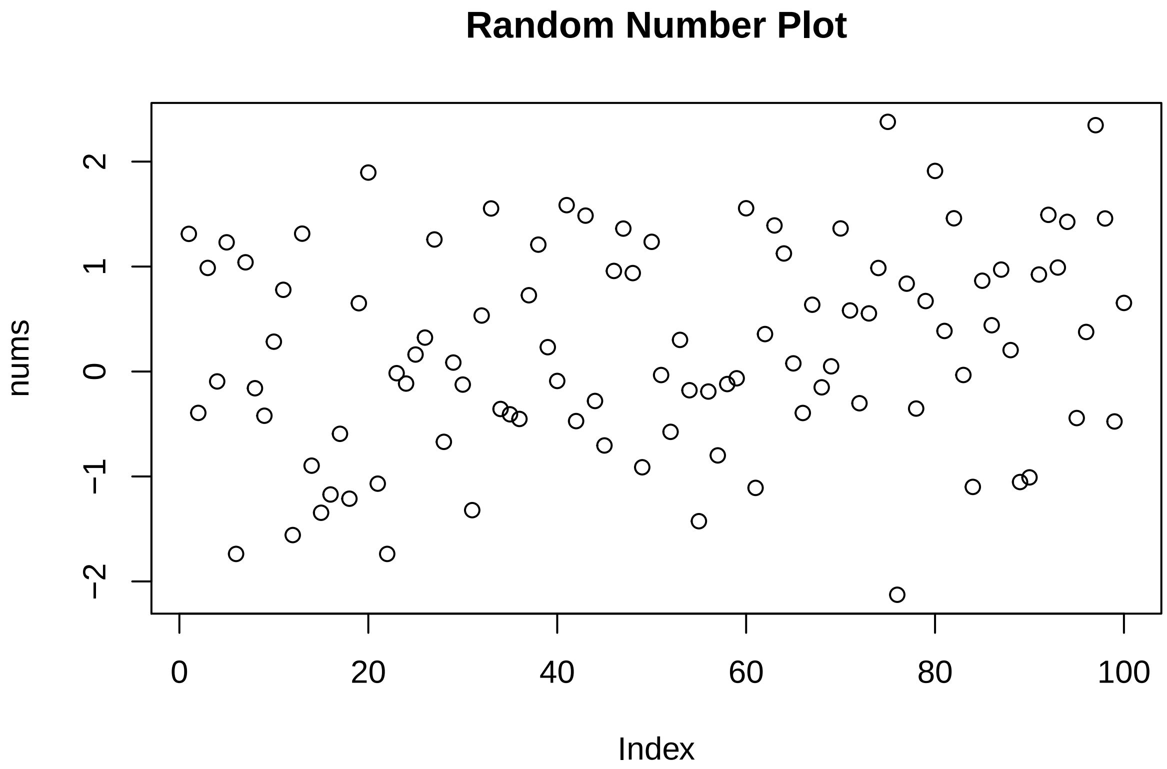
```
nums=c(4,5,8,11)
sum(nums)
```

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

3.5

Plot random numbers

```
nums=rnorm(100)
plot(nums, main='Random Number Plot')
```



4.1

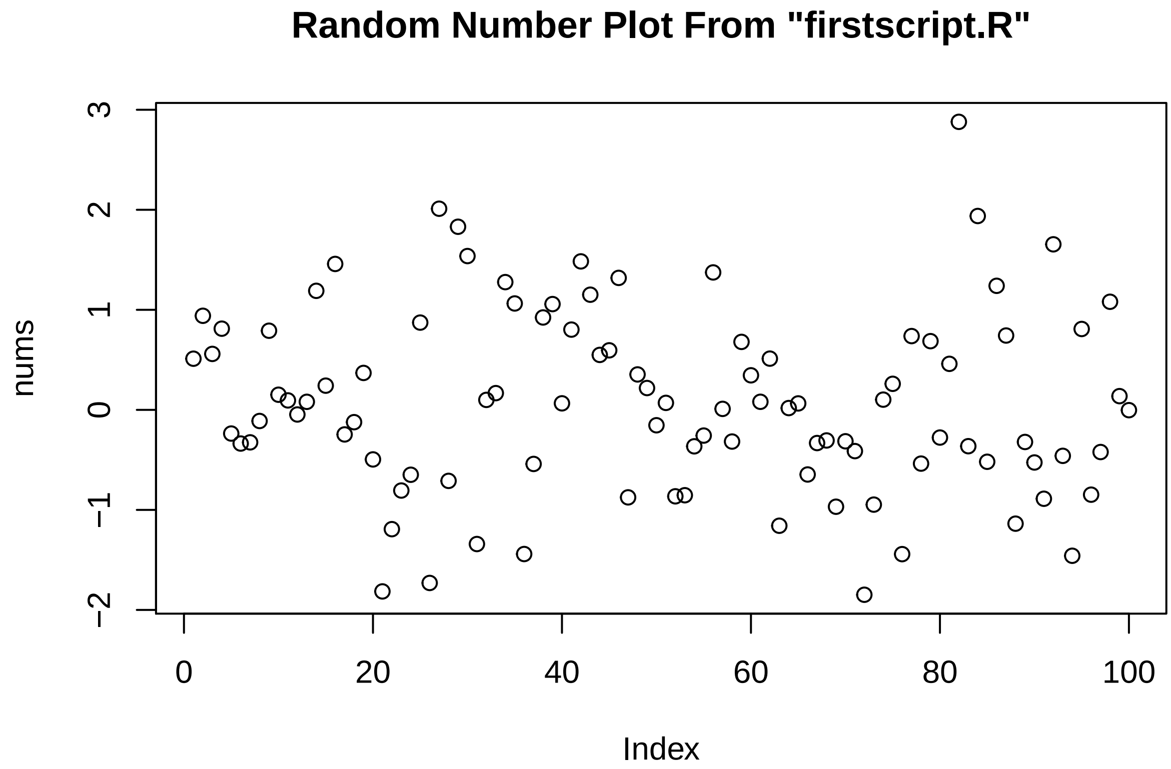
Find help for ‘sqrt’ function

```
help(sqrt)
```

5.1

Make a file called ‘firstscript.R’ containing code that generates 100 random no.s and plots them.

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



6.2

Put the numbers 31 to 60 in a vector named P and in a matrix with 6 rows and 5 columns named Q.

```
Q=matrix(data=c(seq(31,60)),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
```

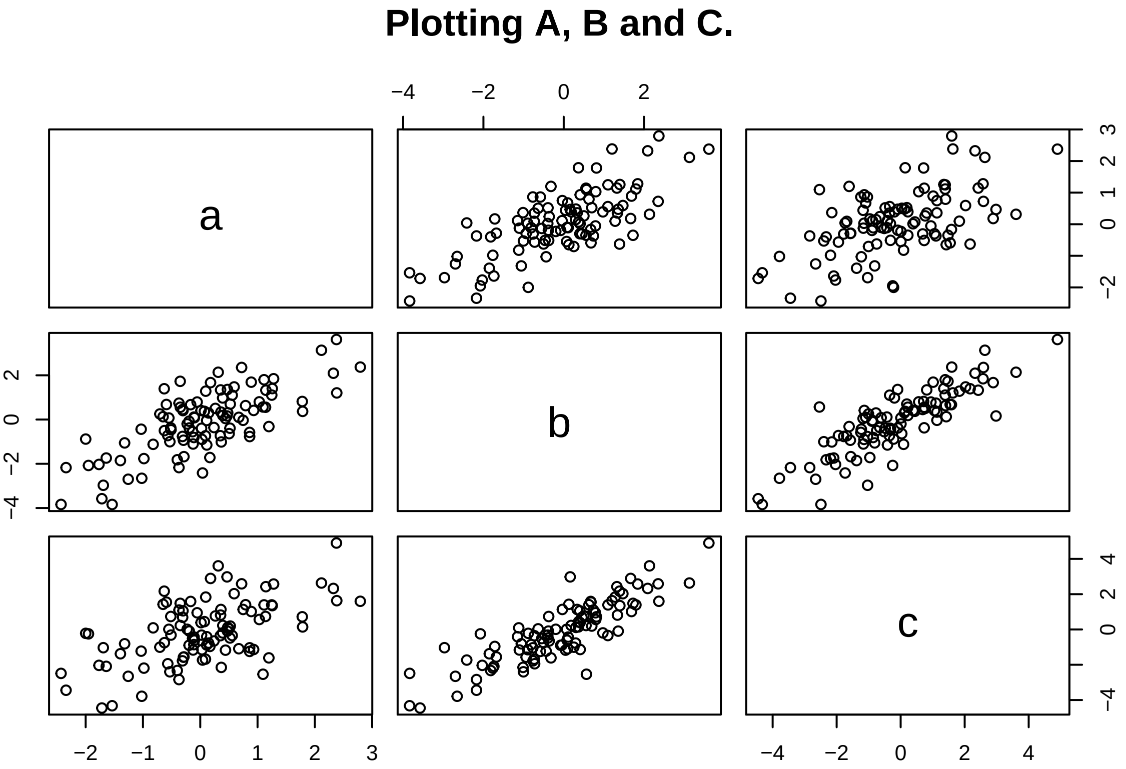
6.3 and 7.1

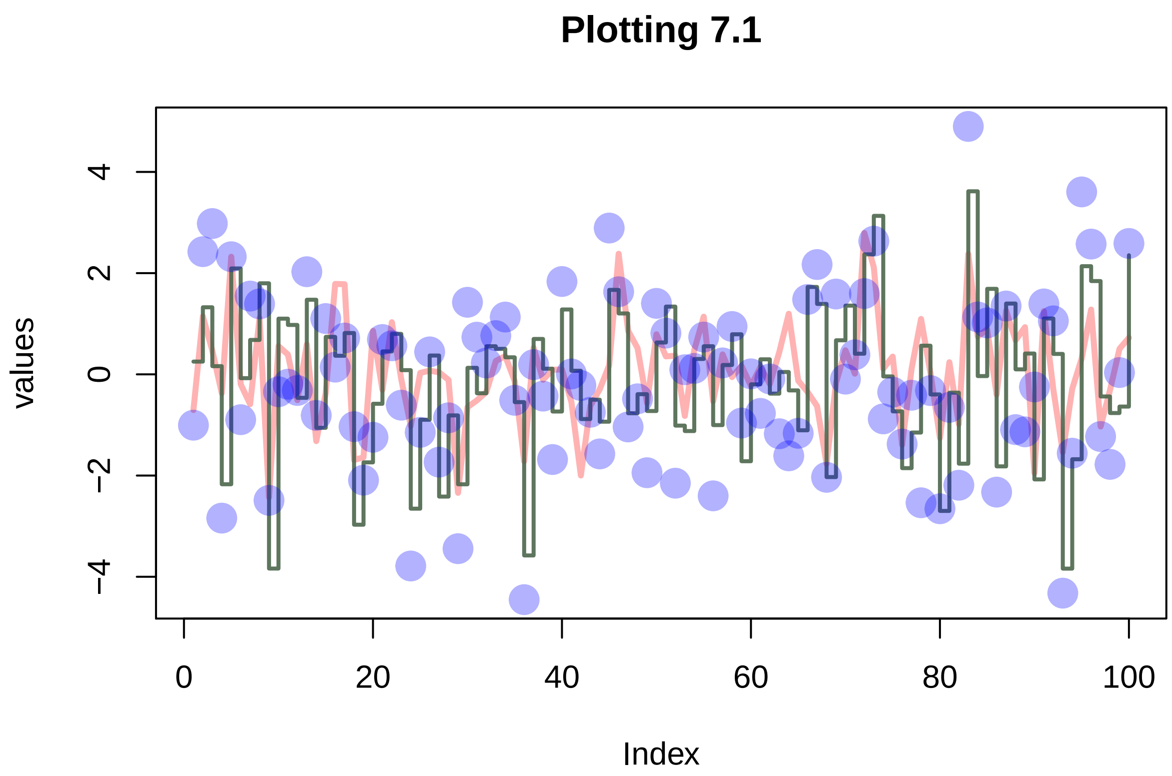
Make a script file which constructs three random normal vectors of length 100. Call these vectors x1, x2 and x3. Make a data frame called t with three columns (called a, b and c) containing respectively x1, x1+x2 and x1+x2+x3. Call the following functions for this data frame: plot(t) and sd(t).

```
x1=rnorm(100)
x2=rnorm(100)
x3=rnorm(100)
t=data.frame(a=c(x1),b=c(x1+x2),c=c(x1+x2+x3))
plot(t, main="Plotting A, B and C.")
sd(t$a)
sd(t$b)
sd(t$c)

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=1, col=rgb(0,0,1,0.3))
```

```
source("secondscript_vectors.R")
```





sd(t$a) [1] 0.9333017

sd(t$b) [1] 1.310654

sd(t$c) [1] 1.720701

RGB tells us what colour the graph should be (choose shade of Red, green and blue to combine).

lwd is line width. pch is which plotting character or symbol you want to use. cex is character or symbol expansion (how big you want your pch to be)

9.1

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

```r
nums=rnorm(100)
mean(sqrt(nums),na.rm=TRUE)
```

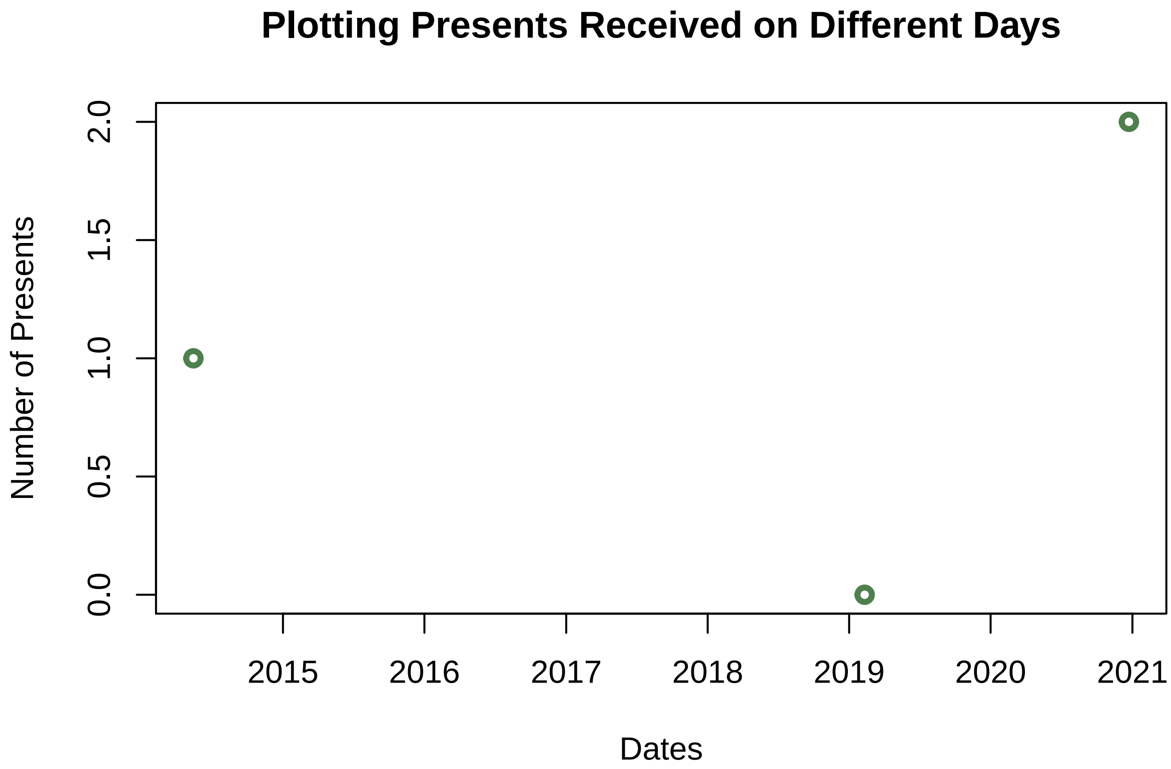
```
## Warning in sqrt(nums): NaNs produced
```

```
## [1] 0.8108468
```

10.1

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.

```r
dates=strptime(c("20190210", "20140515", "20201223"), format="%Y%m%d")
presents = c(0,1,2)
plot(dates,presents,type='p', ylim=range(presents),lwd=3, col=rgb(0.3,0.5,0.3,1), main="Plotting Presen
```



11.2

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.

```
nums=seq(1,100)
for(num in nums)
{
if(num<5|num>90)
{
  nums[num] = num*10
}else{
  nums[num] = num*0.1
}
}
nums
```

```
##   [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
```

11.3

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.

```
function1 = function(nums,low_num,high_num)
{
counter=length(nums)
for(num in 1:counter)
{
if(nums[num]<low_num|nums[num]>high_num)
{
  nums[num] = nums[num]*10
}else{
  nums[num] = nums[num]*0.1
}
}
nums
}

x=c(1,2,3,4,5,20,21,22,23,24,60,70,80,90,100)
low_num=10
high_num=50

function1(x,low_num,high_num)
```

```
##  [1]   10.0   20.0   30.0   40.0   50.0    2.0    2.1    2.2    2.3    2.4
## [11]  600.0  700.0  800.0  900.0 1000.0
```

Extra task

Write code that will prove that footnote true

```
function1 = function(nums,low_num,high_num)
{
counter=seq(1,length(nums))
times_ten=counter[nums<low_num|nums>high_num]
times_less=counter[nums>low_num&nums<high_num]
nums2 = c(nums[times_ten]*10)
nums2 = append(nums2,nums[times_less]*0.1)
nums2
}

x=c(1,2,3,4,5,20,21,22,23,24,60,70,80,90,100)
low_num=10
high_num=50
function1(x,low_num,high_num)
```

```
##  [1]   10.0   20.0   30.0   40.0   50.0  600.0  700.0  800.0  900.0 1000.0
## [11]    2.0    2.1    2.2    2.3    2.4
```