SRT411A0.rmd

Akhilesh Verma, Kulmiye Egeh

February 16, 2018

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

ToDo1

```r
(2016-2014)/(2014-1998)*100
```

```
## [1] 12.5
```

ToDo2

```r
x = (2016-2014)/(2014-1998)*100
x
```

```
## [1] 12.5
```

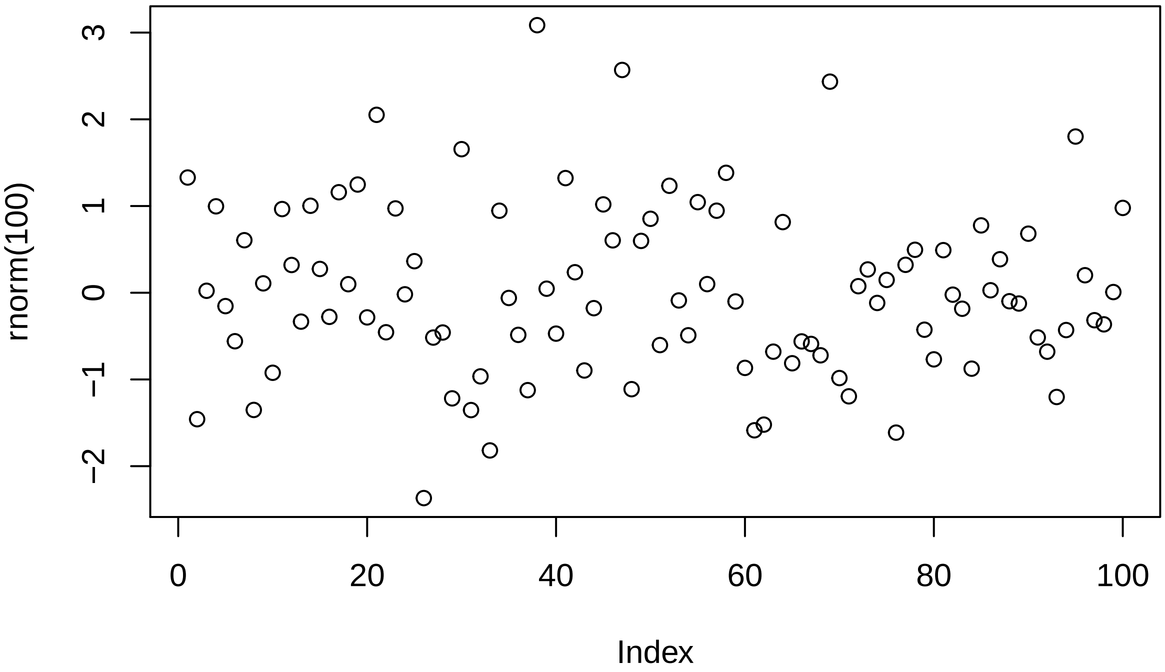
ToDo3

```r
sum(c(4,5,8,11))
```

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

ToDo4

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



ToDo5

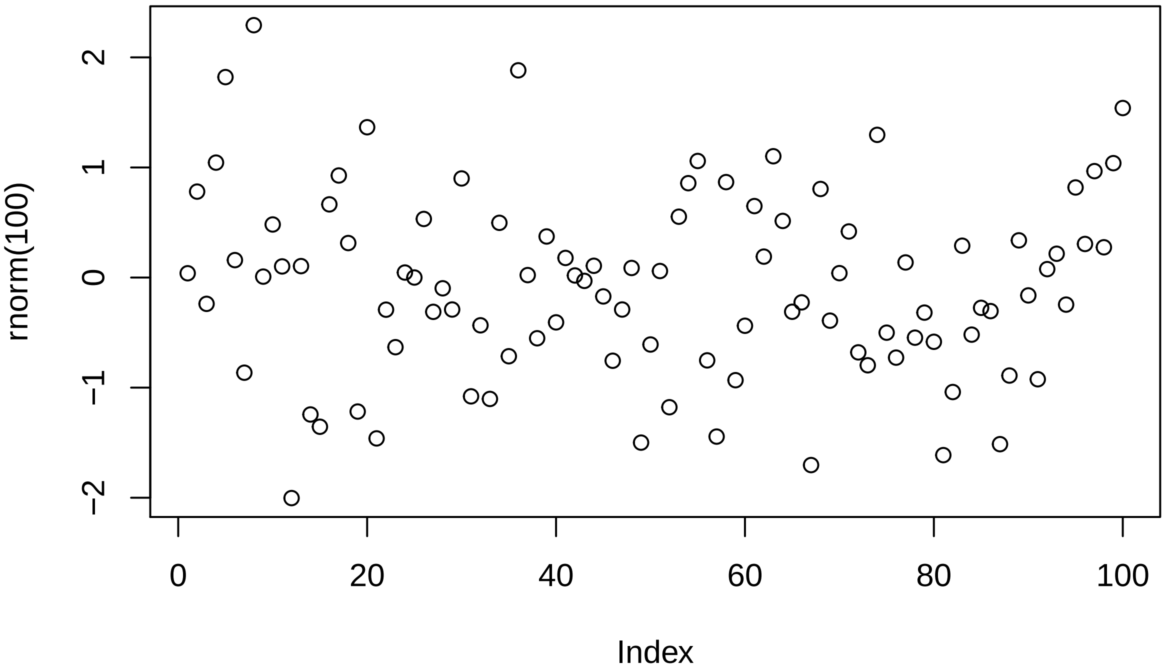
```
?sqrt
```

ToDo6

```
file.create("firstscript.R")
```

```
## [1] TRUE
```

```
writeLines("plot(rnorm(100))","firstscript.R")
source("firstscript.R")
```



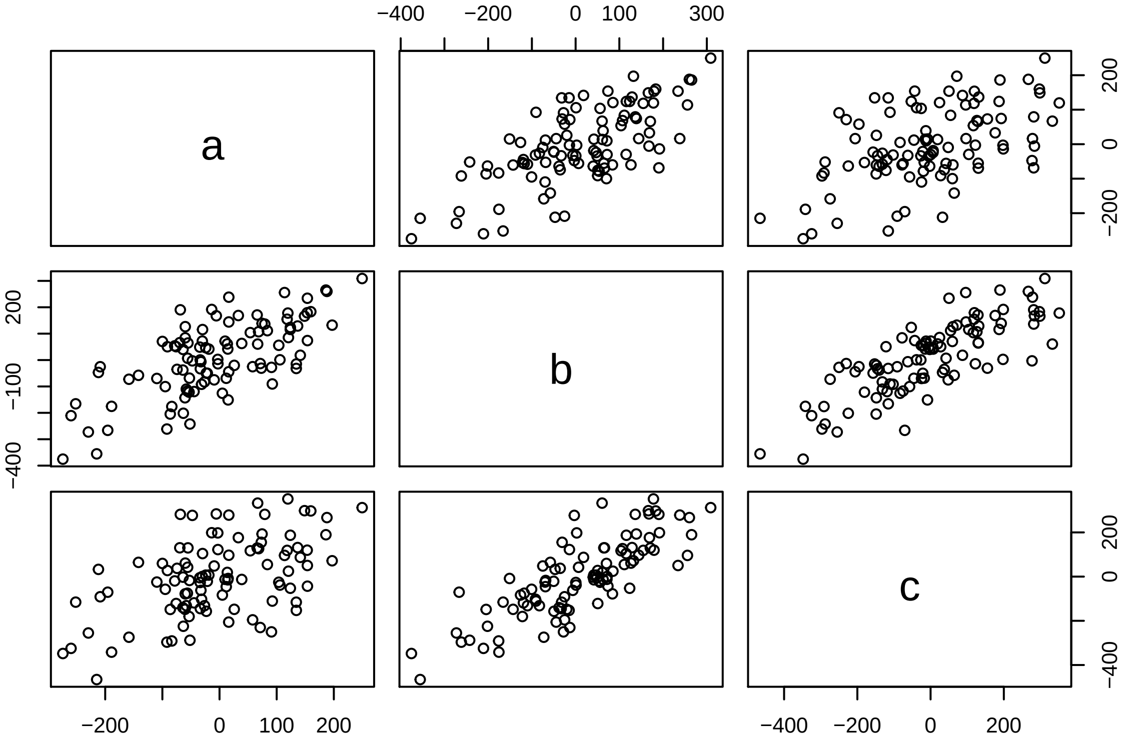
ToDo7

```
p = seq(from = 31, to =60, by = 1)
q = matrix(p, 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
```

ToDo8

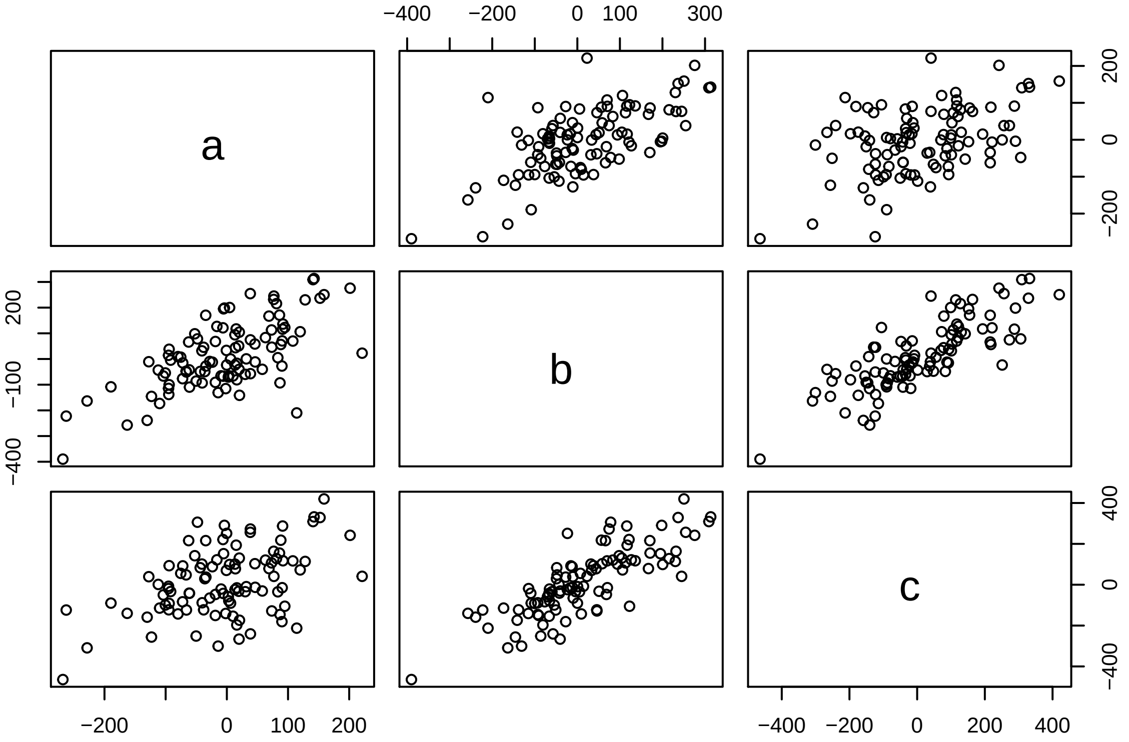
```
x1 = rnorm(100, 1, 100)
x2 = rnorm(100, 1, 100)
x3 = rnorm(100, 1, 100)
t = data.frame(a = x1, b = x1 + x2, c = x1 + x2 + x3)
plot(t)
```



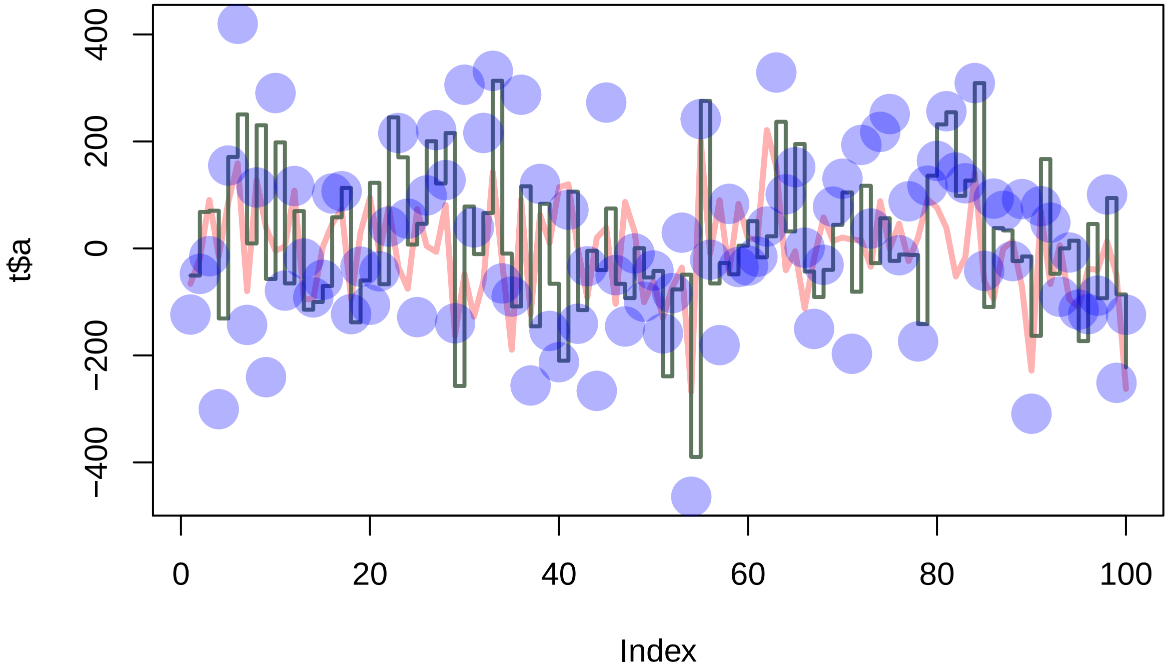
```
#sd(t)
```

ToDo9

```
x1 = rnorm(100, 1, 100)
x2 = rnorm(100, 1, 100)
x3 = rnorm(100, 1, 100)
t = data.frame(a = x1, b = x1 + x2, c = x1 + x2 + x3)
plot(t)
```



```
#sd(t)
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 Function creates colors
#lwd sets the width of the line
#pch specifies symbols to use in the graph
#cex is a FUNCTION which specifies how big the symbols should be compared to default, e.g default = '1'
```

ToDo10

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

```
##    a   g  x
## 1  1  10  3
## 2  2  20  6
## 3  4  40 12
## 4  8  80 24
## 5 16 160 48
## 6 32 320 96
```

```
#Note, you must create a file called tst1.txt to avoid errors upon compiling this code
```

ToDo11

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

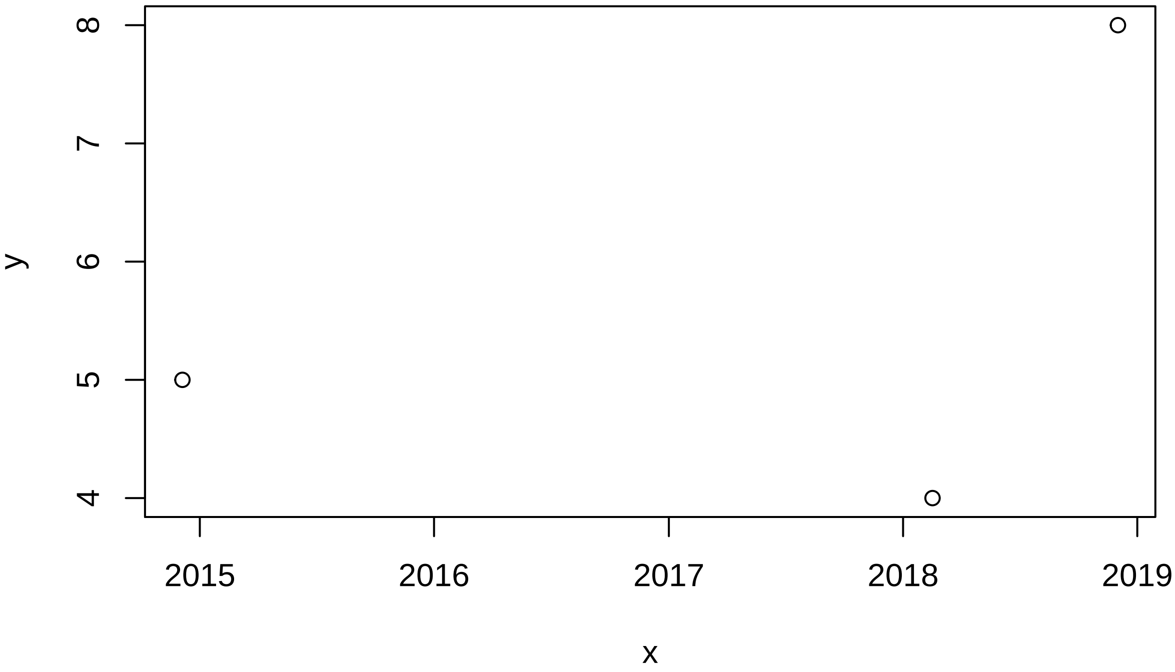
```
## Warning in sqrt(c(rnorm(100))): NaNs produced
```

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

```
#We get 'NaN' for some values because negative random numbers were also generated and they cannot be sq
```

ToDo12

```
x=strptime(c("20180216000000", "20141205000000", "20181202000000"), format="%Y%m%d%H%M%S", tz = "EST")
y = c(4,5,8)
plot(x,y)
```



ToDo13

```
x = seq(from = 1, to = 100)
s = c()
for (i in 1:length(x)){
  if(x[i] < 5){
    s[i] = x[i]*5
  }
  else if (x[i] > 90){
    s[i] = x[i]*5
  }
  else{
    s[i] = x[i]*.5
  }
}
s
```

```
##   [1]   5.0  10.0  15.0  20.0   2.5   3.0   3.5   4.0   4.5   5.0   5.5
##  [12]   6.0   6.5   7.0   7.5   8.0   8.5   9.0   9.5  10.0  10.5  11.0
##  [23]  11.5  12.0  12.5  13.0  13.5  14.0  14.5  15.0  15.5  16.0  16.5
##  [34]  17.0  17.5  18.0  18.5  19.0  19.5  20.0  20.5  21.0  21.5  22.0
##  [45]  22.5  23.0  23.5  24.0  24.5  25.0  25.5  26.0  26.5  27.0  27.5
##  [56]  28.0  28.5  29.0  29.5  30.0  30.5  31.0  31.5  32.0  32.5  33.0
##  [67]  33.5  34.0  34.5  35.0  35.5  36.0  36.5  37.0  37.5  38.0  38.5
##  [78]  39.0  39.5  40.0  40.5  41.0  41.5  42.0  42.5  43.0  43.5  44.0
##  [89]  44.5  45.0 455.0 460.0 465.0 470.0 475.0 480.0 485.0 490.0 495.0
## [100] 500.0
```

ToDo14

```
fun1 = function(arg1, arg2)
{
  x = seq(from=arg1, to=arg2)
  s = c()
  for(i in 1:length(x))
  {
    if(x[i]<5)
    {
      s[i]=x[i]*10
    }
    else if(x[i]>90)
    {
      s[i]=x[i]*10
    }
    else(s[i]=x[i]*0.1)
    {

    }
  }
  s
}
### Samples:
fun1(10, 30)
```

```
##  [1] 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6
## [18] 2.7 2.8 2.9 3.0
```

```
fun1(1,100)
```

```
##   [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
```

```
fun1(40,120)
```

```
##  [1]    4.0    4.1    4.2    4.3    4.4    4.5    4.6    4.7    4.8    4.9
## [11]    5.0    5.1    5.2    5.3    5.4    5.5    5.6    5.7    5.8    5.9
## [21]    6.0    6.1    6.2    6.3    6.4    6.5    6.6    6.7    6.8    6.9
## [31]    7.0    7.1    7.2    7.3    7.4    7.5    7.6    7.7    7.8    7.9
## [41]    8.0    8.1    8.2    8.3    8.4    8.5    8.6    8.7    8.8    8.9
## [51]    9.0  910.0  920.0  930.0  940.0  950.0  960.0  970.0  980.0  990.0
## [61] 1000.0 1010.0 1020.0 1030.0 1040.0 1050.0 1060.0 1070.0 1080.0 1090.0
## [71] 1100.0 1110.0 1120.0 1130.0 1140.0 1150.0 1160.0 1170.0 1180.0 1190.0
## [81] 1200.0
```

ToDo15

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

```
##   [1]    5.0   10.0   15.0   20.0   12.5   15.0   17.5   20.0   22.5   25.0
##  [11]   27.5   30.0   32.5   35.0   37.5   40.0   42.5   45.0  475.0  500.0
##  [21]  525.0  550.0  575.0  600.0  625.0  650.0  675.0  700.0  725.0  750.0
##  [31]  775.0  800.0  825.0  850.0  875.0  900.0  925.0  950.0  975.0 1000.0
##  [41] 1025.0 1050.0 1075.0 1100.0 1125.0 1150.0 1175.0 1200.0 1225.0 1250.0
##  [51] 1275.0 1300.0 1325.0 1350.0 1375.0 1400.0 1425.0 1450.0 1475.0 1500.0
##  [61] 1525.0 1550.0 1575.0 1600.0 1625.0 1650.0 1675.0 1700.0 1725.0 1750.0
##  [71] 1775.0 1800.0 1825.0 1850.0 1875.0 1900.0 1925.0 1950.0 1975.0 2000.0
##  [81] 2025.0 2050.0 2075.0 2100.0 2125.0 2150.0 2175.0 2200.0 2225.0 2250.0
##  [91] 2275.0 2300.0 2325.0 2350.0 2375.0 2400.0 2425.0 2450.0 2475.0 2500.0
```