SRT411A0

Markus Yuzon

March 22, 2019

Introduction

This is assignment0 for SRT411 that instructs us on how to use Rstudio in order to create outputs using basic commands. The goal of this Assignment is to write the codes for the "ToDos" of the document found in this link: https://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf

1

```
((2018-2014)/(2014-1997))*100
```

```
## [1] 23.52941
```

2

```
sy=2018
gy=2014
 by=1997
 a = sy - gy
 b = gy - by
c = a/b
 d = c * 100
 d
```

```
## [1] 23.52941
```

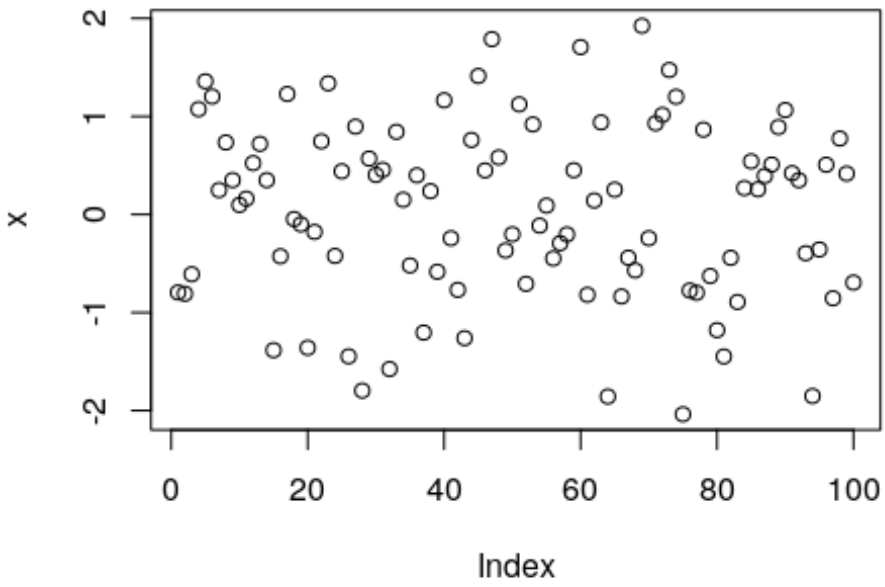
3

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

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

4

```
x= rnorm(100)
plot(x)
```



#5

```
help(sqrt)
```

Miscellaneous Mathematical Functions

Description

abs(x) computes the absolute value of x, sqrt(x) computes the (principal) square root of x, √{x}.

The naming follows the standard for computer languages such as C or Fortran.

Usage

abs(x) sqrt(x) Arguments

x
a numeric or complex vector or array. Details

These are internal generic primitive functions: methods can be defined for them individually or via the Math group generic. For complex arguments (and the default method), z, abs(z) == Mod(z) and sqrt(z) == z^0.5.

abs(x) returns an integer vector when x is integer or logical.

S4 methods

Both are S4 generic and members of the Math group generic.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) The New S Language. Wadsworth & Brooks/Cole.

See Also

Arithmetic for simple, log for logarithmic, sin for trigonometric, and Special for special mathematical functions.

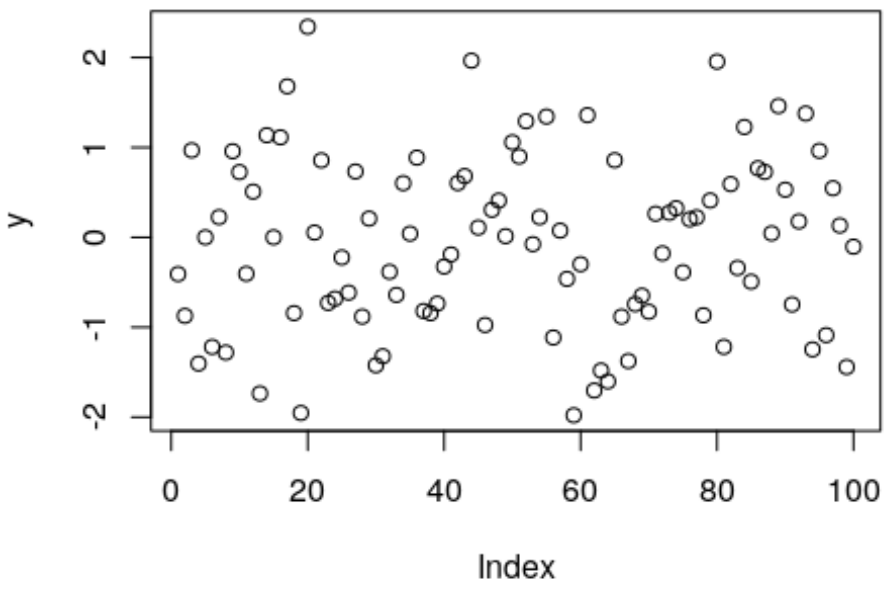
'plotmath' for the use of sqrt in plot annotation.

Examples

require(stats) # for spline require(graphics) xx <- -9:9 plot(xx, sqrt(abs(xx)), col = "red") lines(spline(xx, sqrt(abs(xx)), n=101), col = "pink")

6

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



#7

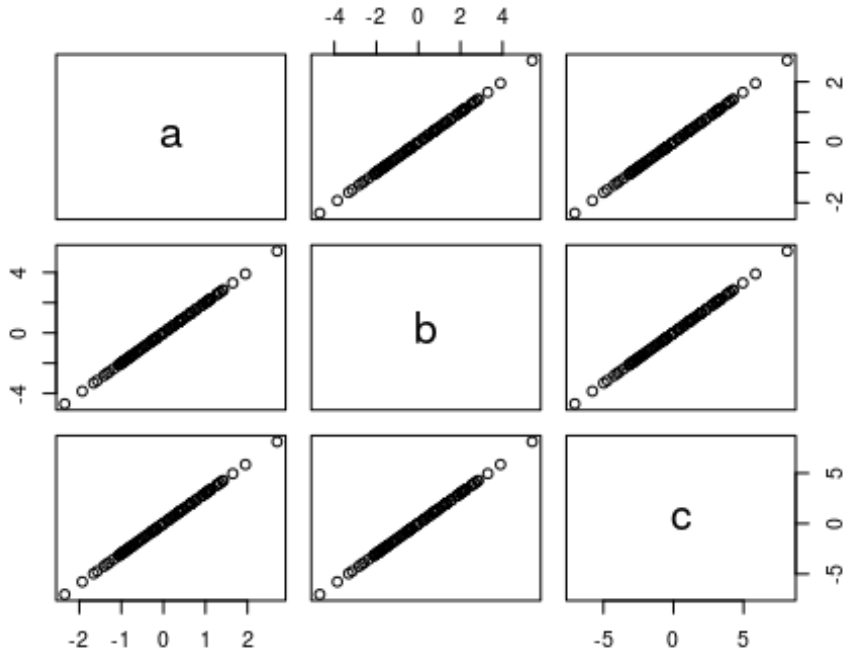
```
P = seq(from=31, to=60, by=1)
Q=matrix(data=(P),ncol = 5,nrow = 6)
Q
```

```
##      [,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
```

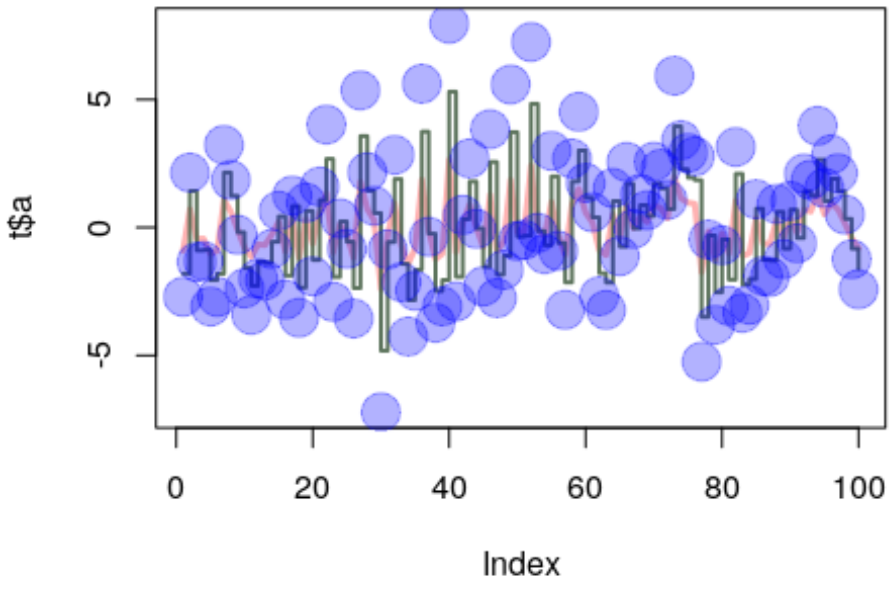
8

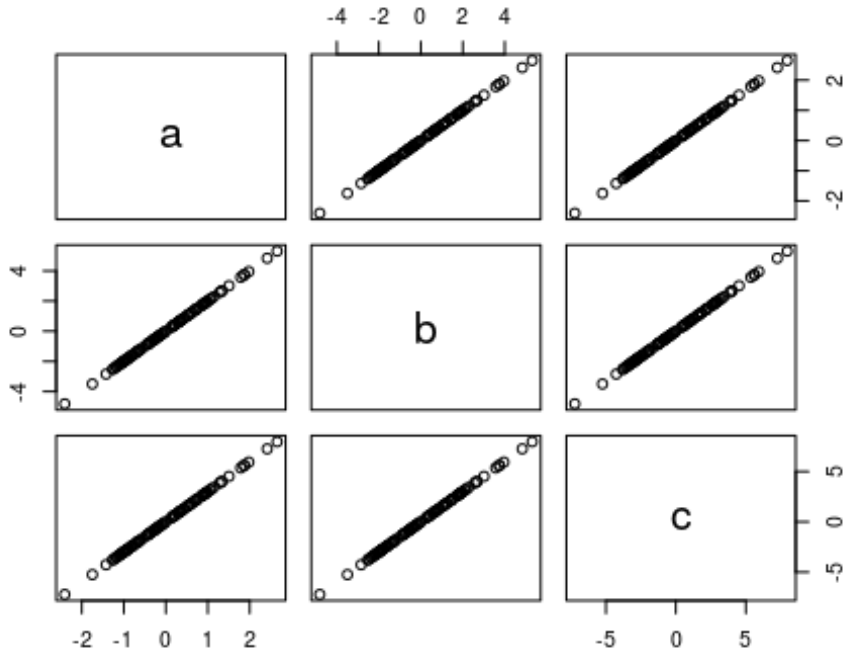
```
source("dataframes.R")
```



9

```
source("df2.R")
```





rgb is for the color option for the plots. parameters will be set per color on the rgb function lwd can be a vector, its first element will apply to lines but the whole vector to symbols pch is plotting character either used as a single or integer code of a set of graphics symbols cex is the character/symbol expansion: numerical vector #10

```
source("multi.R")
r
```

```
##   [1] -0.90266661  0.71533448 -0.44895244 -0.43693861 -1.02937931
##   [6] -0.90388439  1.07670776  0.61921133 -0.09457086 -0.79210683
##  [11] -1.13898841 -0.66554564 -0.68733678 -0.27968553  0.21730639
##  [16] -0.94049152  0.41630521 -1.17593285  0.32302769 -0.63421182
##  [21]  0.53267399  1.34064376 -0.96855291  0.11793747 -0.27847083
##  [26] -1.18472078  1.78685004  0.72015505  0.28258723 -2.40778312
##  [31] -0.28567612  0.94650923 -0.71078615 -1.42097540 -0.82389467
##  [36]  1.87314207 -0.12078083 -1.24174037 -1.02625654  2.65388626
##  [41] -0.96411406  0.16378173  0.89933296 -0.01837126 -0.76613056
##  [46]  1.28011215 -0.91996881 -0.55200242  1.86303895 -0.17885389
##  [51] -0.16032315  2.41728490 -0.06971651 -0.34686871  0.99941692
##  [56] -0.30347917 -1.07298475  0.89835289  1.50645957  0.58058551
##  [61]  0.19900383 -0.89471658 -1.07483576  0.51553487 -0.37339267
##  [66]  0.85433878 -0.01881560  0.43674019  0.22415928  0.85697199
##  [71]  0.76514874  0.36068879  1.97750213  1.13799858  0.97359184
##  [76]  0.92613075 -1.74826214 -0.15587299 -1.26719088 -0.23178791
##  [81] -1.02048391  1.04984827 -1.11303019 -1.00578145  0.36887073
##  [86] -0.63215570 -0.64351225  0.30639017 -0.41300542  0.34699674
##  [91] -0.20676854  0.70371469  0.62097562  1.32346505  0.51575033
##  [96]  0.95269678  0.70630915  0.16989574 -0.41442925 -0.80531651
```

11

```
ss = rnorm(100)
sq = sqrt(ss)
```

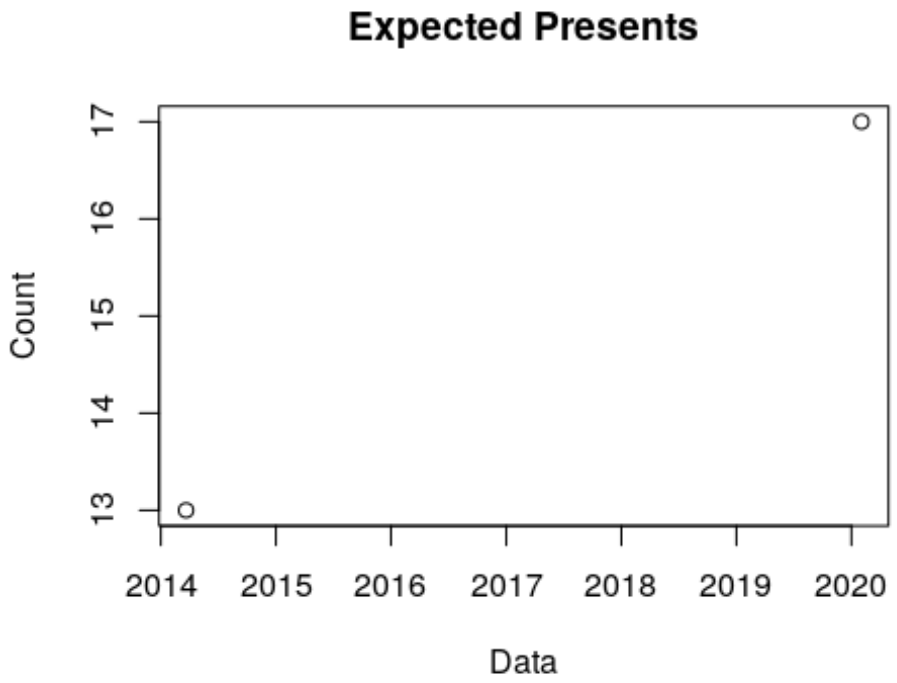
```
## Warning in sqrt(ss): NaNs produced
```

```
mean(sq)
```

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

Value cannot be produced. #12

```
data1 = (strptime(c("20140322","20200202"),format = "%Y%m%d"))
data2=c(13,17)
plot(data1,data2, xlab = "Data",ylab = "Count", main = "Expected Presents")
```



#13

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

```
##   [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
```

14

```
j=1:150
ar = function(arg1)
{
     l = length(arg1)
     for (i in 1:100) {
         if (arg1[i] <5 | arg1[i] > 90)
         {
             arg1[i] = arg1[i] * 10
         }else
         {
             arg1[i] = arg1[i] * 0.1
         }
     }
     return(arg1)
 }
ar(arg1=j)
```

```
##   [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
## [101]  101.0  102.0  103.0  104.0  105.0  106.0  107.0  108.0  109.0
110.0
## [111]  111.0  112.0  113.0  114.0  115.0  116.0  117.0  118.0  119.0
120.0
```

```
## [121]  121.0  122.0  123.0  124.0  125.0  126.0  127.0  128.0  129.0
130.0
## [131]  131.0  132.0  133.0  134.0  135.0  136.0  137.0  138.0  139.0
140.0
## [141]  141.0  142.0  143.0  144.0  145.0  146.0  147.0  148.0  149.0
150.0
```

15

```
jj=j
jk = (jj[jj<5] * 10)
jk1 = jj[5:90] * 0.1
jk3 = jj[jj > 90] * 10
pf = c(jk,jk1,jk3)
pf
```

```
##   [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
## [101] 1010.0 1020.0 1030.0 1040.0 1050.0 1060.0 1070.0 1080.0 1090.0
1100.0
## [111] 1110.0 1120.0 1130.0 1140.0 1150.0 1160.0 1170.0 1180.0 1190.0
1200.0
## [121] 1210.0 1220.0 1230.0 1240.0 1250.0 1260.0 1270.0 1280.0 1290.0
1300.0
## [131] 1310.0 1320.0 1330.0 1340.0 1350.0 1360.0 1370.0 1380.0 1390.0
1400.0
## [141] 1410.0 1420.0 1430.0 1440.0 1450.0 1460.0 1470.0 1480.0 1490.0
1500.0
```