Assignment 0

Sepehr Farahmand and Rahil Chadha

2018-02-16

This document is designed to provide a solid introduction to the R language. We will be exploring vectors, matrices, lists and other common data types.We will also be assigning values to variables and creating graphs. Lastly, We will also be exploring ways to script using R such as for loops and if statements.

ToDo 1 - Percentage of life spent at Seneca

```
(2016-2014)/(2014-1991)*100
```

```
## [1] 8.695652
```

ToDo 2 - Percentage of life spent at Seneca with variables

```
a=2014
b=2016
c=1991
(b-a)/(a-c)*100
```

```
## [1] 8.695652
```

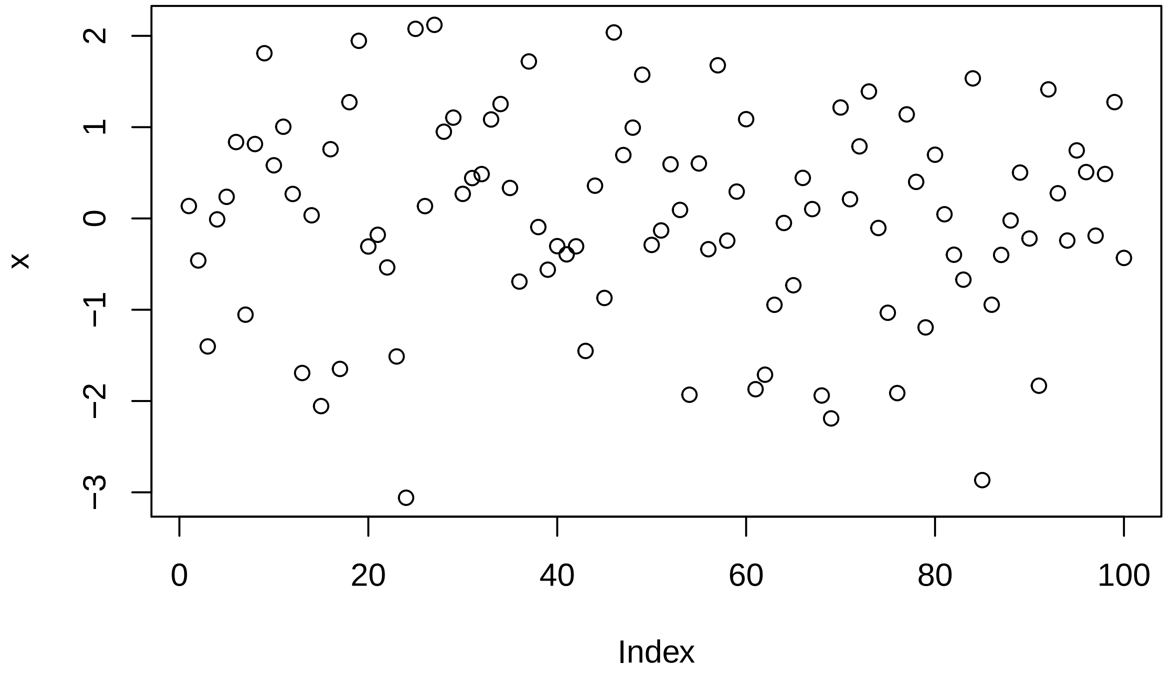
ToDo 3 - Sum of vector

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

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

ToDo 4 - Plotting random numbers

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



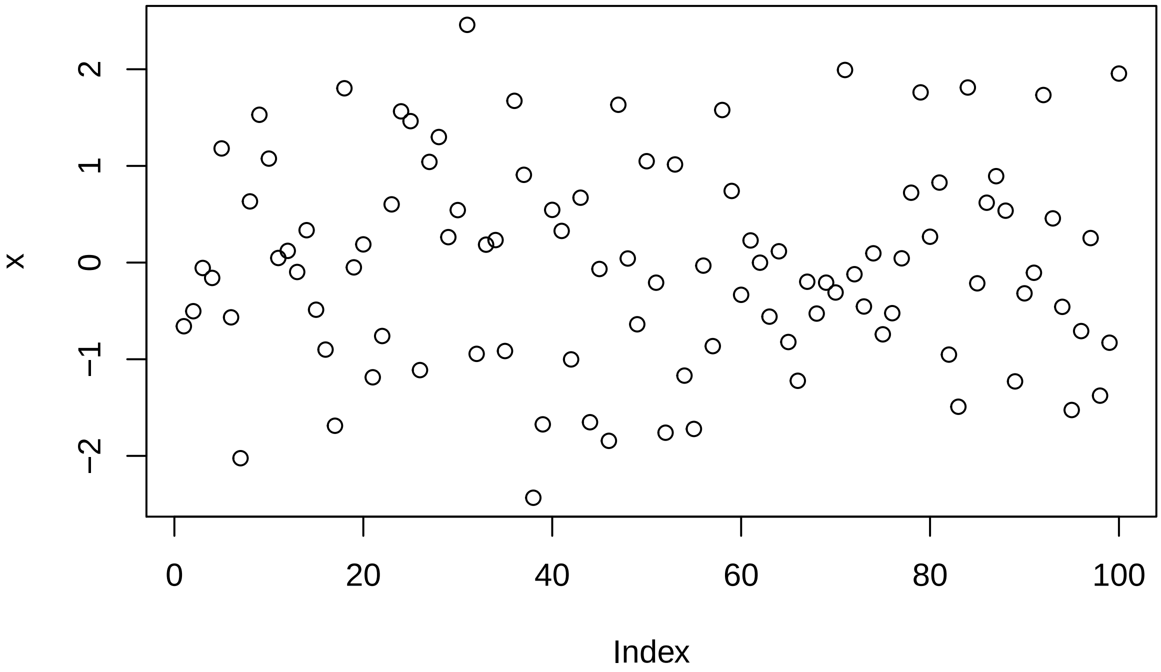
ToDo 5 - Find the help of sqrt

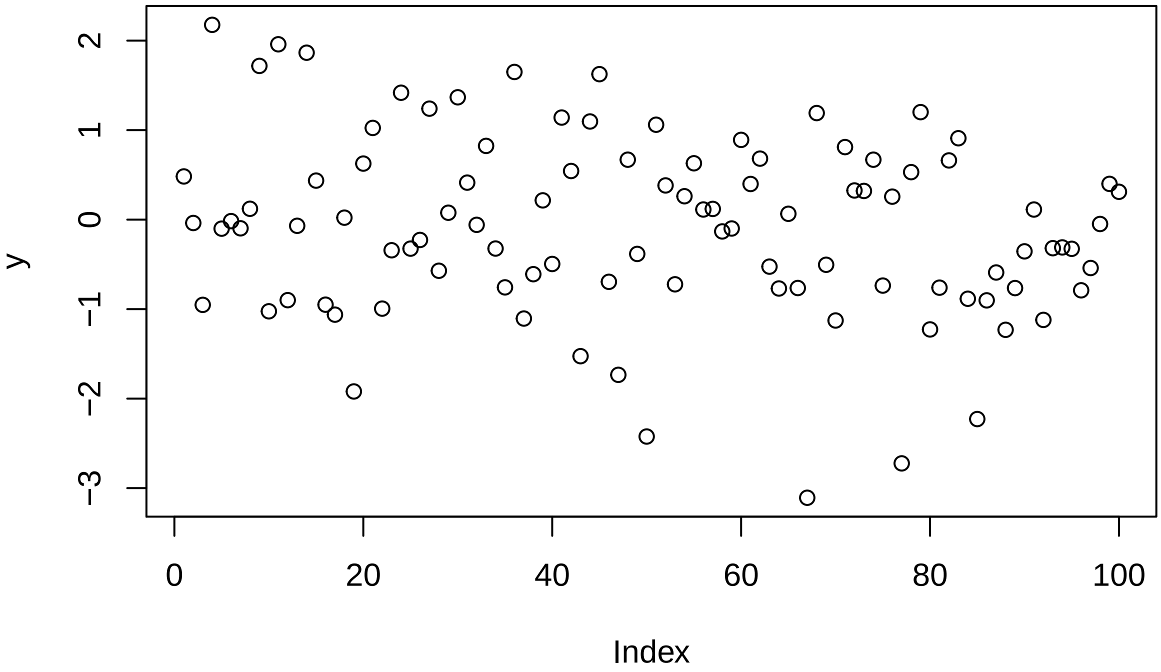
```
help(sqrt)
```

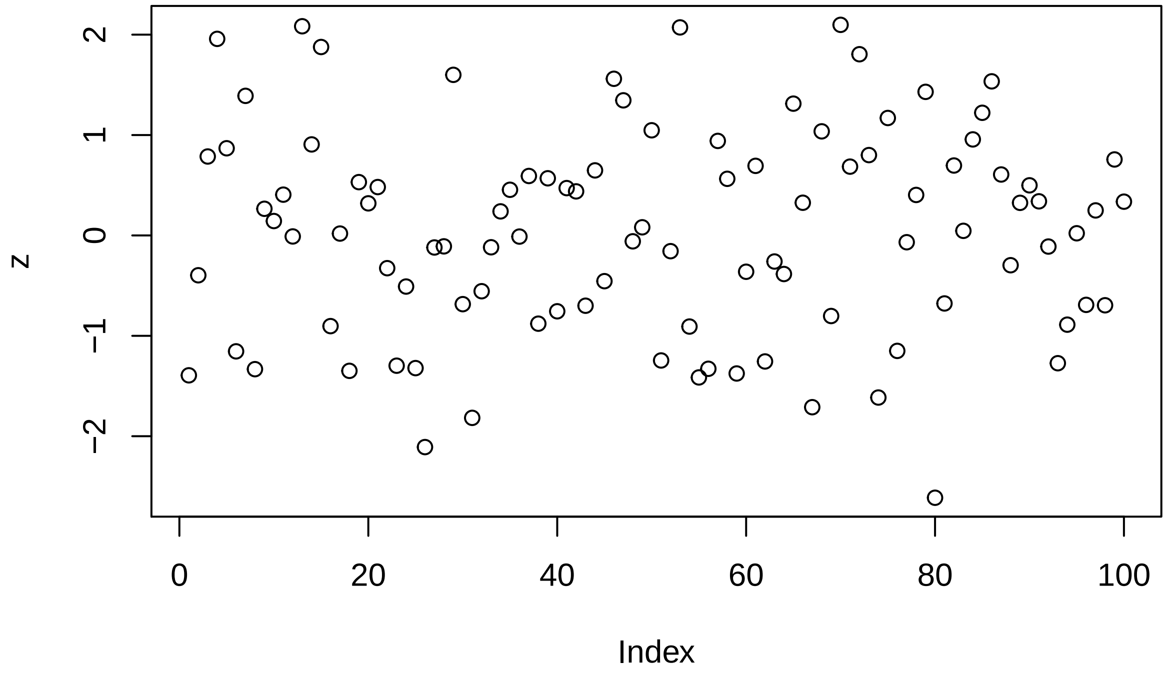
```
## starting httpd help server ... done
```

ToDo 6 - Running an external script

```
source("firstscript.r")
```


x
2
1
0
−1
−2
0
20
40
60
80
100
Index


y
2
1
0
−1
−2
−3
0
20
40
60
80
100
Index



ToDo 7 - Data Structures

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

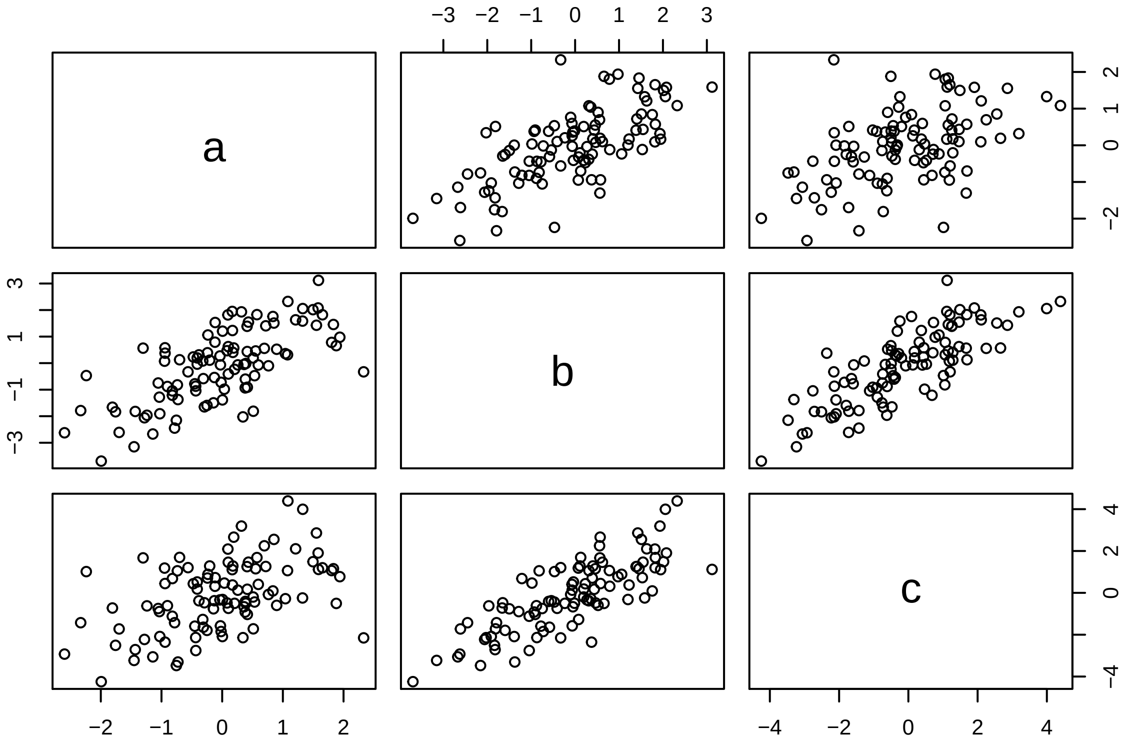
```
##  [1] 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
## [24] 54 55 56 57 58 59 60
```

```
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 - Data Frames

```
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)
```

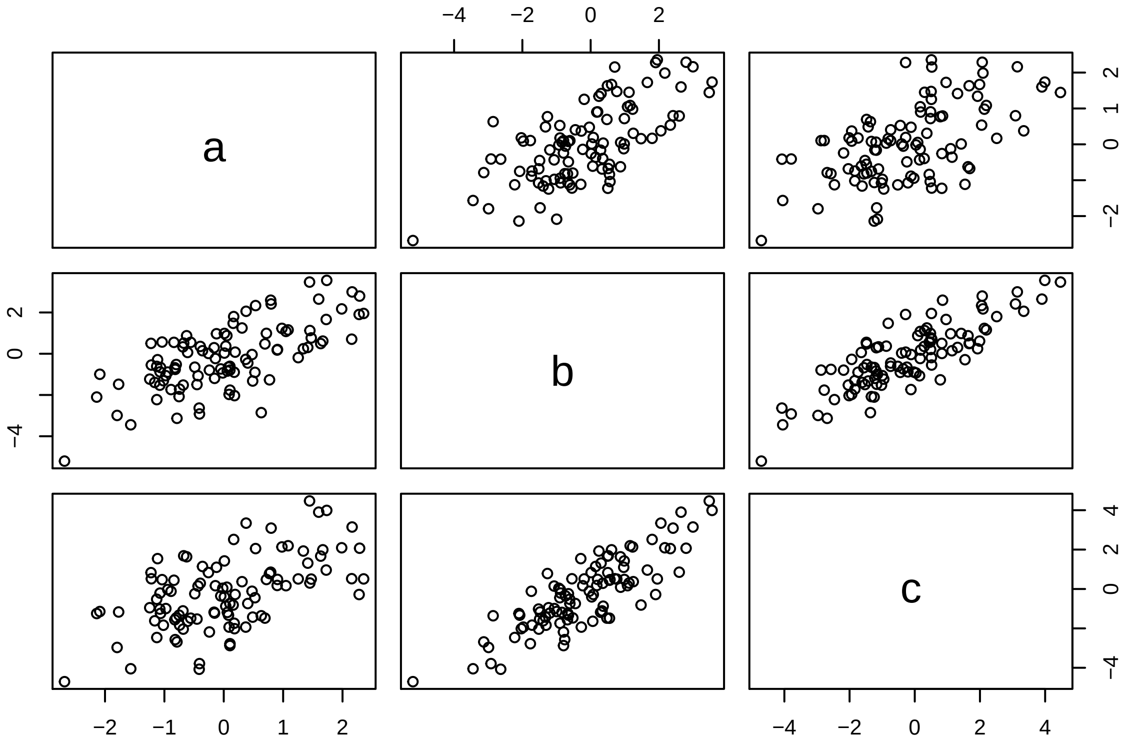


```
sapply(t, sd)
```

```
##        a        b        c
## 1.011183 1.384854 1.689019
```

ToDo 9 - Rgb Graphs

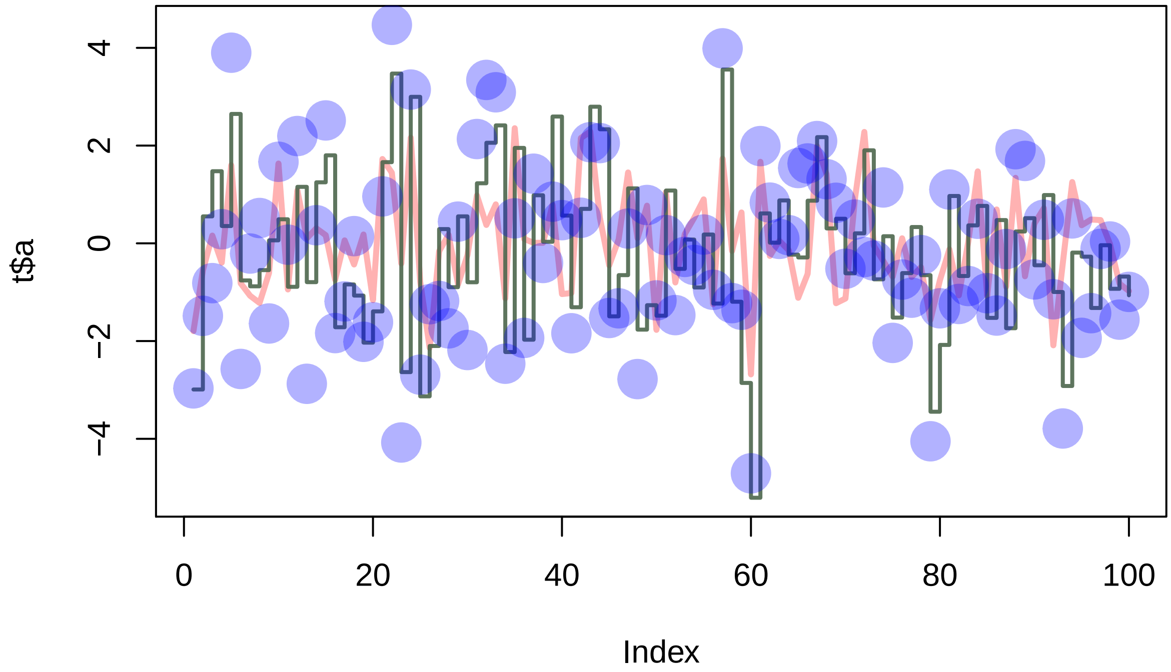
```
x1 = rnorm(100)
x2 = rnorm(100)
x3 = rnorm(100)
t = data.frame(a = c(x1), b = c(x1+x2), c = c(x1+x2+x3))
plot(t)
```



```
sapply(t, sd)
```

```
##        a        b        c
## 1.082245 1.585939 1.804828
```

```
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 means red blue green. The rgb value determines the color of the graph. The final value for rgb is the alpha which is the opaqueness of the colors. Lwd is the line width. Cex is a numerical value giving the amount by which the plotted text should be magnified.Pch specifies a symbol or character to be used as the default plotting point.

ToDo 10 - Writing to text file

```
d2 = read.table(file="tst1.txt", header=TRUE)
d2$g * 5 -> d2$g
write.table(d2, file="tst2.txt", row.names=FALSE)
d2
```

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

ToDo 11 - Using sqrt and mean

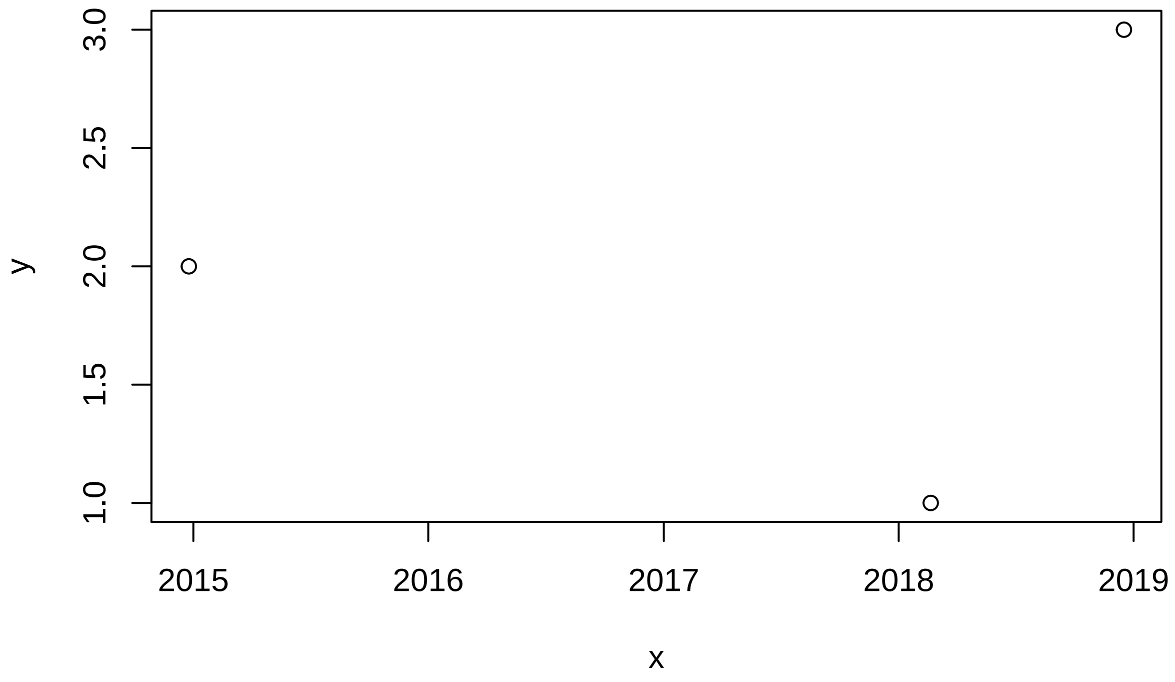
```
x = runif(100, min=1, max=100)
sqrt(x) -> y
mean(y)
```

```
## [1] 6.387533
```

It is not possible to use rnorm to generate your list because you will end up with invalid numbers when you try and square root a negative number. Interestingly enough, the mean is always between 6.3 and 7.2.

ToDo 12 - Time/date vectors and graphs

```
x=strptime( c("20180219180000", "20141225000000", "20181217000000"), format="%Y%m%d%H%M%S")
y=c(1,2,3)
plot(x,y)
```



ToDo 13 - Using if statements and for loops

```
h=seq(from=1, to=100, by=1)
s=c()
for(i in 1:100)
{
  if(h[i]<5)
  {
    s[i]=h[i]*5;
  }
  else if(h[i]>90)
  {
    s[i]=h[i]*10;
  }
  else
  {
    s[i]=h[i]*0.1;
  }
}

s
```

```
##   [1]    5.0   10.0   15.0   20.0    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 - Using functions and arguments

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

```
##   [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 - Avoiding for loops via vector computation

```
w=seq(from=1, to=100, by=1)
w<-ifelse(w<5 | w>90,w*10,w*0.1)
w
```

```
##   [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
```

Acknowledgements

. R Markdown - Dynamic Documents for R (large resource) . Writing reproducible reports in R with markdown, knitr and pandoc . Markdown . knitr with R Markdown . R markdown cheatsheet A (very) short introduction to R ##### The end