Assignment 0: Intro to R and Github

Yuriy Orobtsev, github - yuriy2110

The purpose of this assignment is to be able to create codes using RStudio. Accomplish various tasks from calculation to plotting graphs. As well, knitr packages to format output into pdf.

Link to the document we used:https://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf

ToDo 1

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

```
## [1] 11.11111
```

ToDo 2

```
a <- (2016 -2014)/(2016 - 1998) * 100
a
```

```
## [1] 11.11111
```

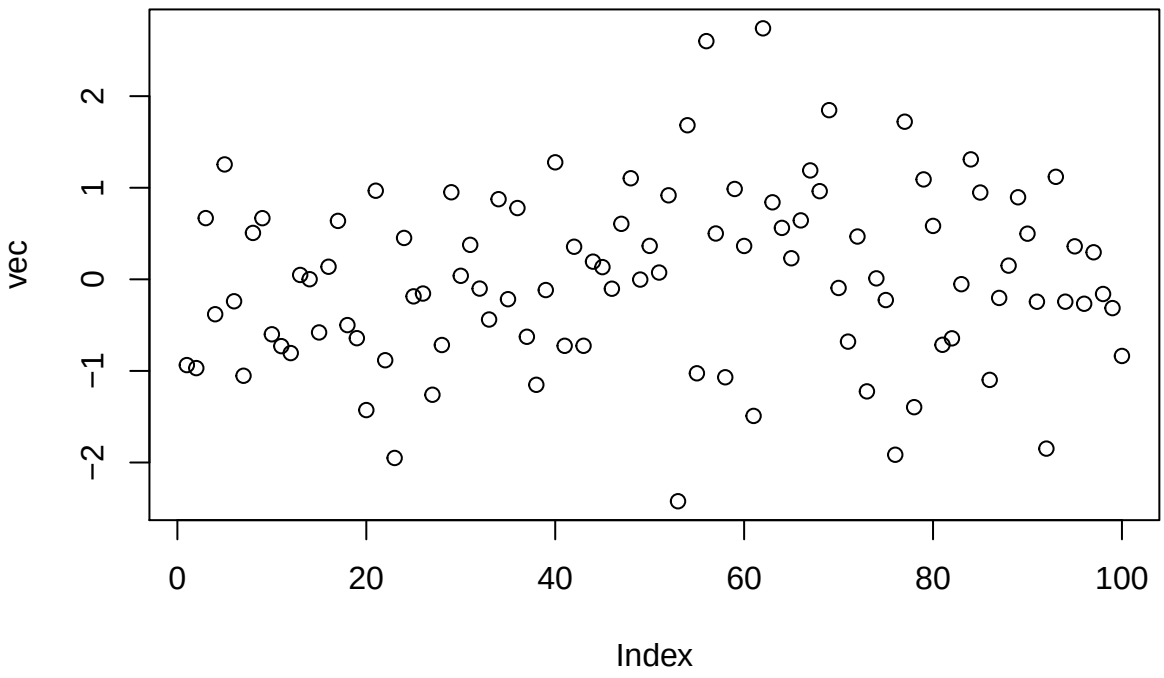
ToDo 3

```
vec <- c(4, 5, 8, 11)
sum(vec)
```

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

ToDo 4

```
vec <- rnorm(100)
plot(vec)
```



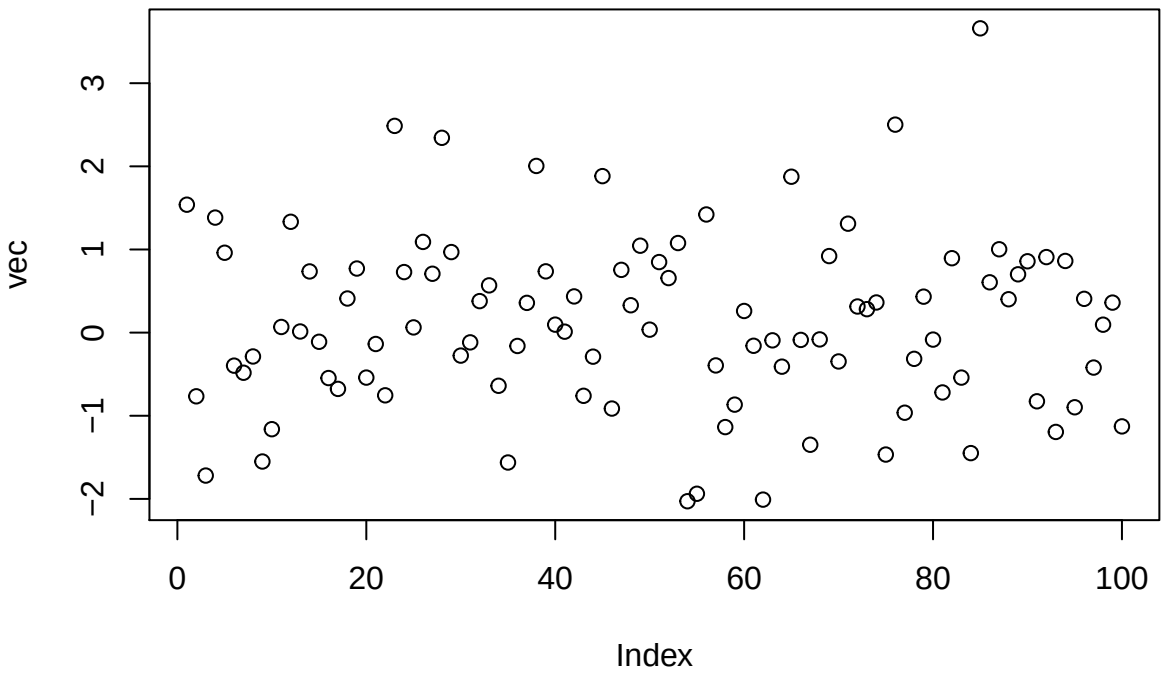
ToDo 5

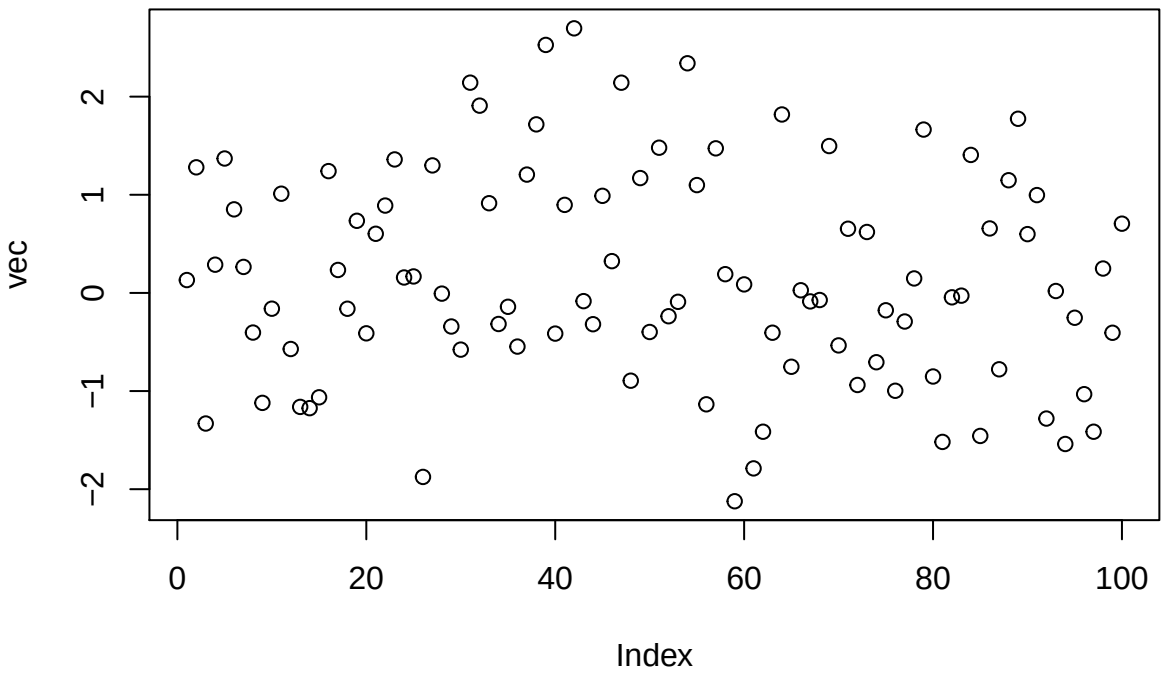
```
help(sqrt)
```

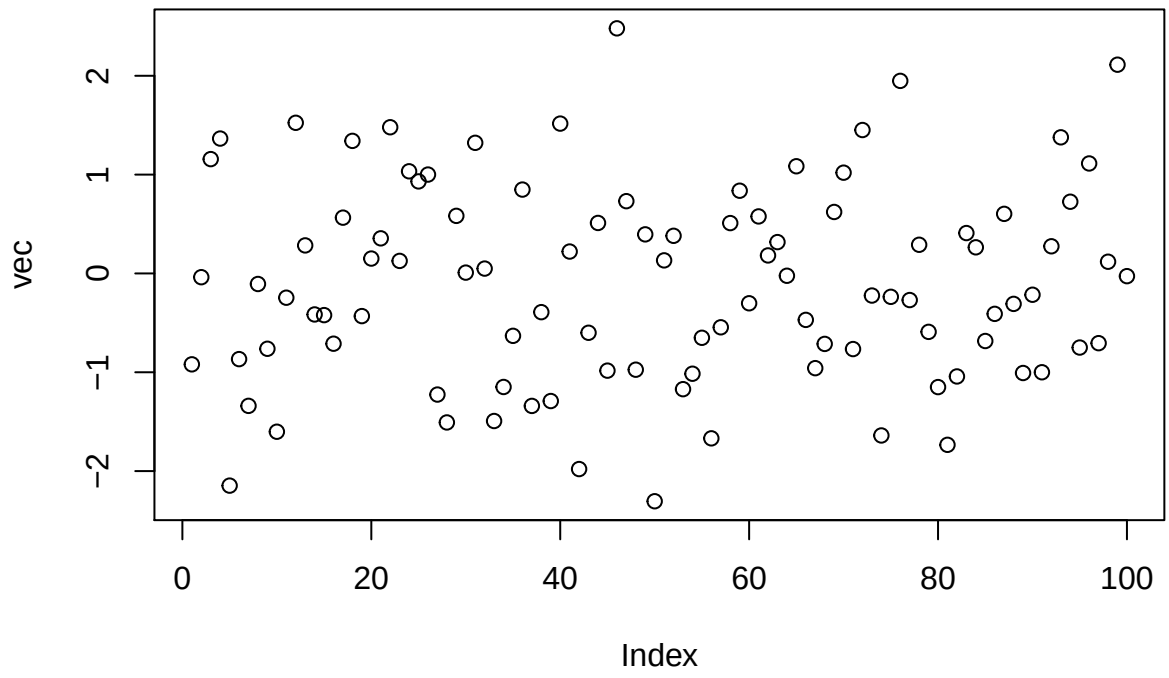
```
## starting httpd help server ... done
```

ToDo 6

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


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


vec
2
1
0
−1
−2
0
20
40
60
80
100
Index



ToDo 7

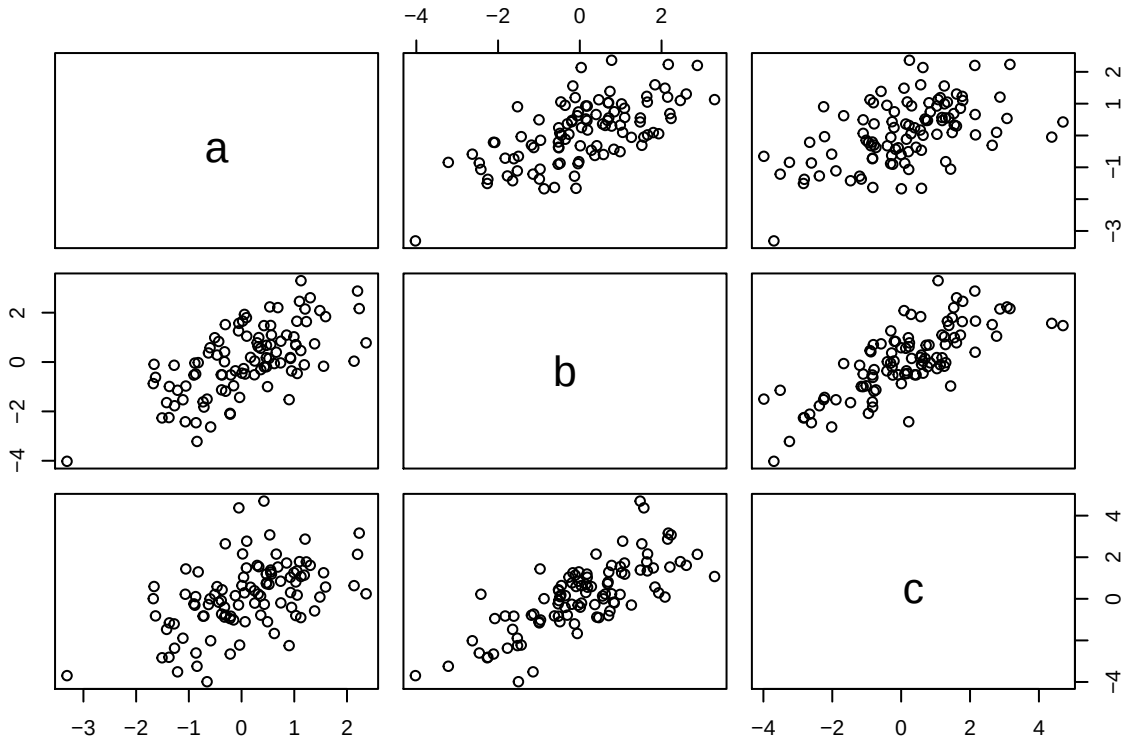
```
p <- 31:60
q <- matrix(p, 6, 5)
q
```

```
##      [,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
```

ToDo 8

```
x1 <- rnorm(100)
x2 <- rnorm(100)
x3 <- rnorm(100)

t <- data.frame(a = c(x1), b = c(x1 + x2), c = c(x1 + x2 + x3))
#sd(t)
plot(t)
```

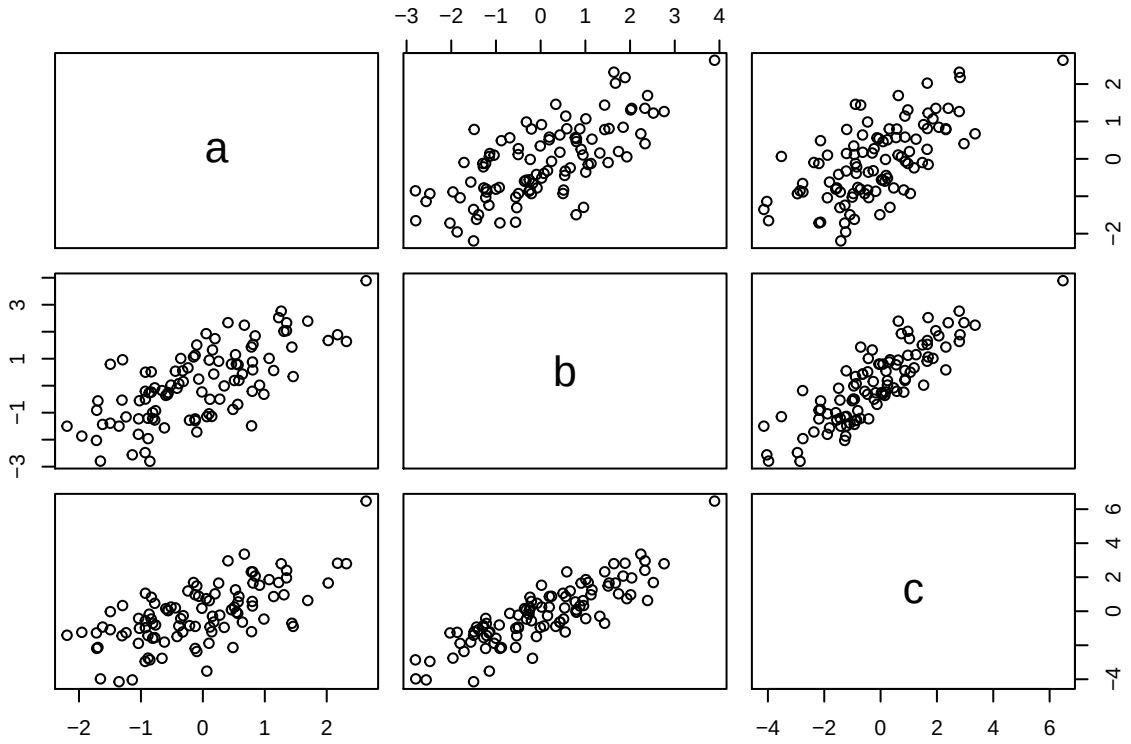


sd kept giving an error: (list) object cannot be coerced to type 'double'

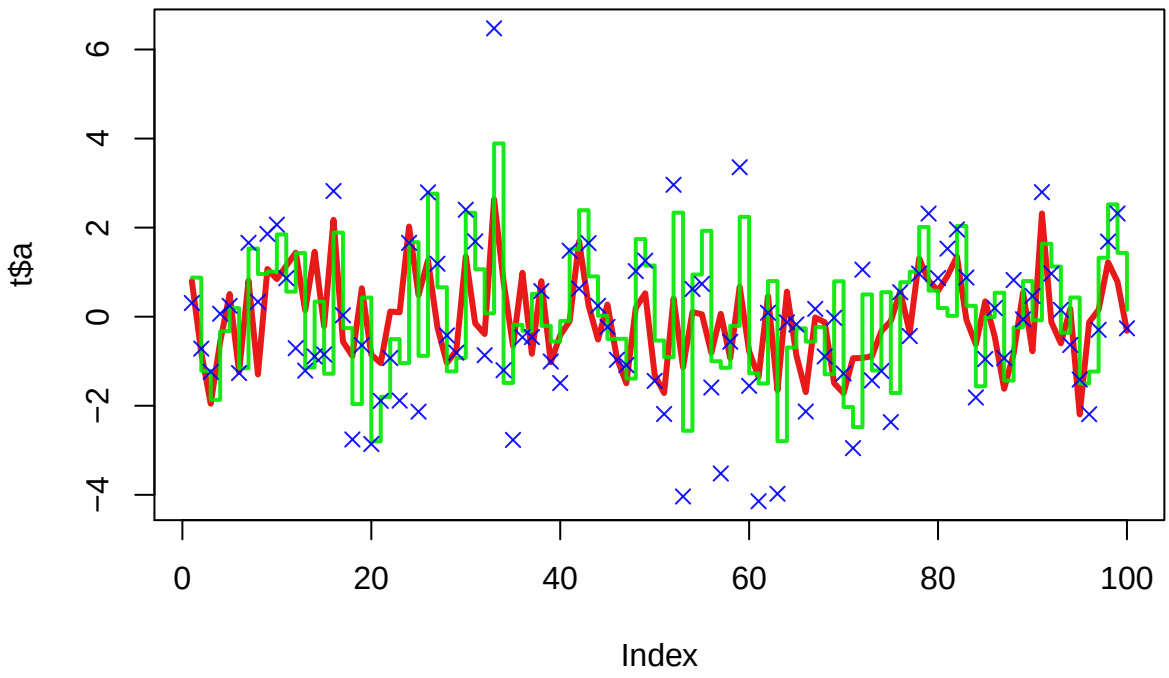
ToDo 9

```
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)
```




```
plot(t$a, type="l", ylim=range(t),
     lwd=3, col=rgb(0.9,0,0,0.9))
lines(t$b, type="s", lwd=2,
      col=rgb(0,0.9,0,0.9))
points(t$c, pch=4, cex=1,
       col=rgb(0,0,1,0.9))
```



lwd signifies the line width in comparison the the default, which is 1. pch signifies which symbol is going to be used for the points on the plot. Cex signifies the scaling for the plot text and symbols. rgb signifies the red, green, blue colour of the points with the final value being the opacity of the colour.

ToDo 10

```
x <-data.frame(read.table(file="tst1.txt", header = TRUE));
x$g = x$g*5
write.table(x, file="tst2.txt", row.names = FALSE)
```

This is the result of running the code: tst2.txt

ToDo 11

```
x <- rnorm(100)
mean(sqrt(x))
```

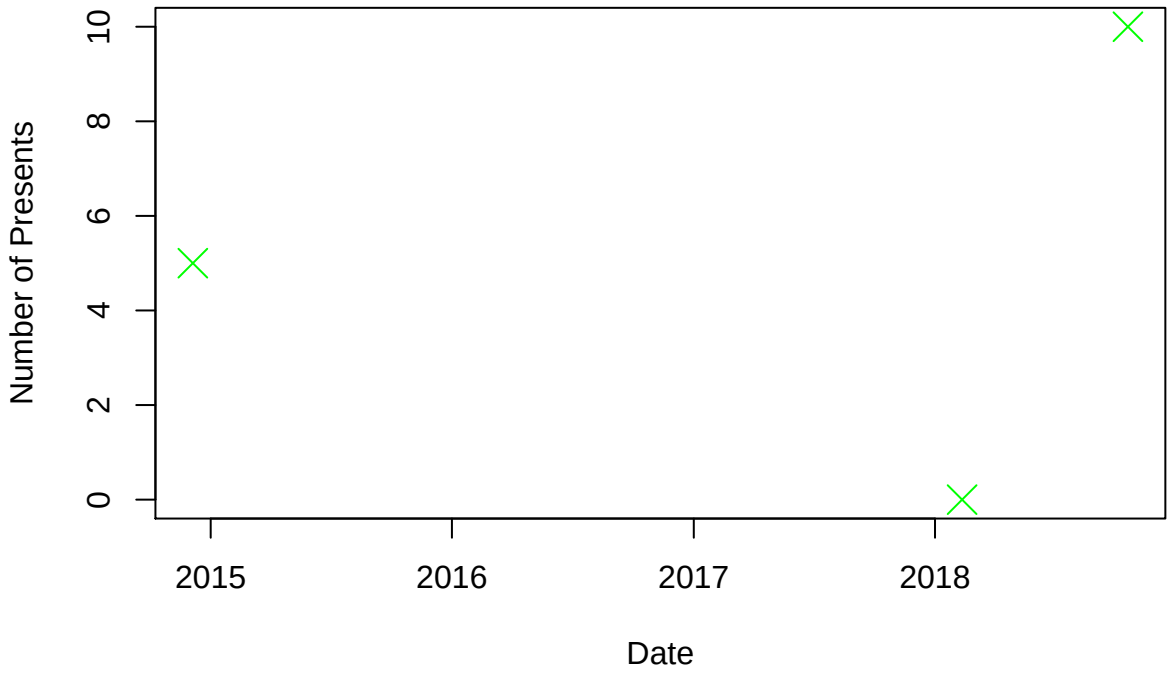
```
## Warning in sqrt(x): NaNs produced
```

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

When trying to sqrt the vector imaginary numbers are produced. Output is In sqrt(x) : NaNs produced So it's imposible to find mean of imaginary numbers

ToDo 12

```
#Vector for the dates.
x <- strptime(c("20180211", "20141205", "20181020"), format ="%Y%m%d")
#Vector for number of presents on each day: 0 today, 5 on Sinterklaas, and 10 on my birthday.
y <- c(0,5,10)
plot(x,y, type="p", pch=4, cex=2, xlab="Date", ylab="Number of Presents", col=rgb(0,1,0,1))
```



ToDo 13

```
x <- c(1:100)
y <- c()
for(i in 1:100)
{
  if (x[i]<5)
  {
    y[i]=x[i]*10;
  }
  else if (x[i]>90)
  {
    y[i]=x[i]*10;
  }
  else
  {
    y[i]=x[i]*0.1;
  }
```

```
}
y
```

```
##   [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 14

```
v=1:100
func = 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)
}
func(arg1=v)
```

```
##   [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 final

```
v=1:100
ifelse(v < 90 | v >90, v * 10, v * 0.1)
```

```
##   [1]   10   20   30   40   50   60   70   80   90  100  110  120  130  140
##  [15]  150  160  170  180  190  200  210  220  230  240  250  260  270  280
##  [29]  290  300  310  320  330  340  350  360  370  380  390  400  410  420
```

```
##  [43]  430  440  450  460  470  480  490  500  510  520  530  540  550  560
##  [57]  570  580  590  600  610  620  630  640  650  660  670  680  690  700
##  [71]  710  720  730  740  750  760  770  780  790  800  810  820  830  840
##  [85]  850  860  870  880  890    9  910  920  930  940  950  960  970  980
##  [99]  990 1000
```