Assignment0

1) TODO

```r
((2018-2014)/(2014-1989))*100
```

[1] 16

2) TODO

```r
ok <-(2018 - 2014)
mk <- (2014 - 1989)
jk <- ((ok/mk)*100)
jk
```
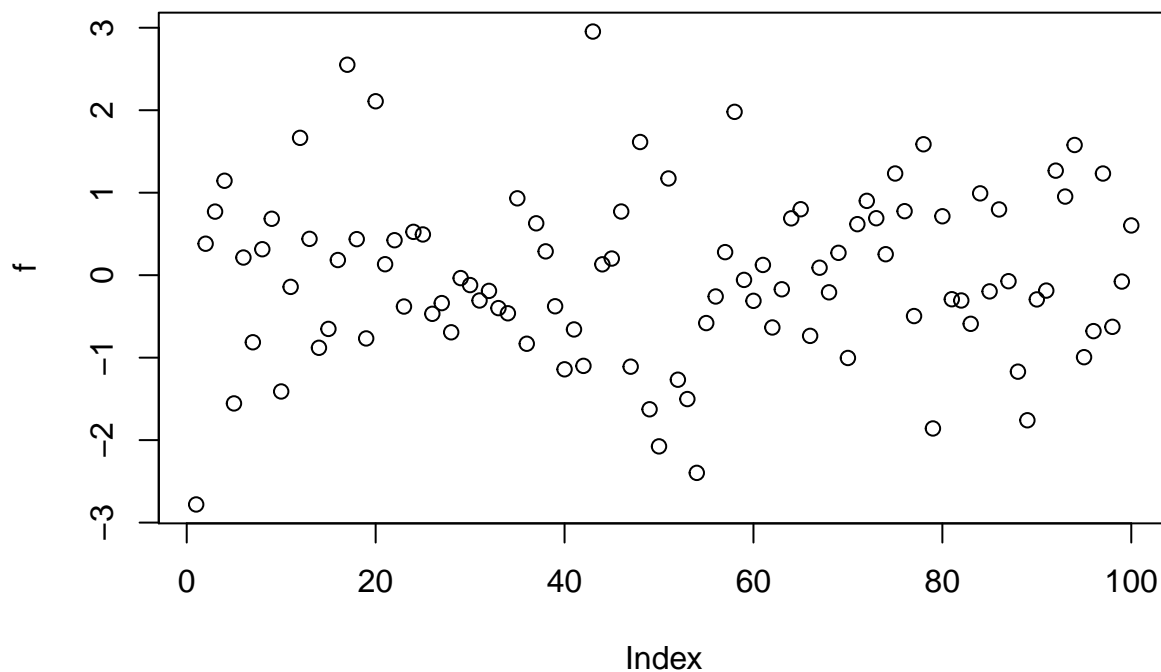
[1] 16

3) TODO

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

[1] 28

4) TODO

```r
f = rnorm(100)
plot(f)
```


5) TODO

```r
help (sqrt)
```

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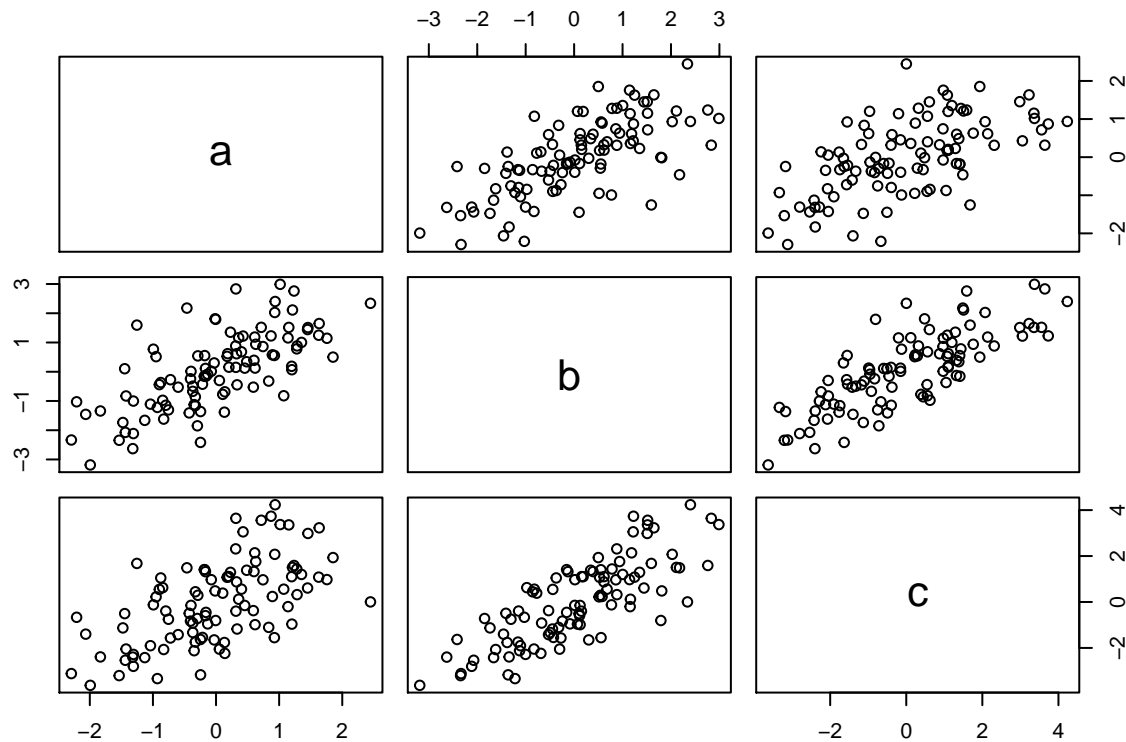
```r
P= seq(from= 31, to= 60)
Q= matrix(data= P, ncol=5, nrow=6)
```

```
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
```

7) TODO

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



```
sd(t$a)
```

```
## [1] 0.997464
```

```
sd(t$b)
```
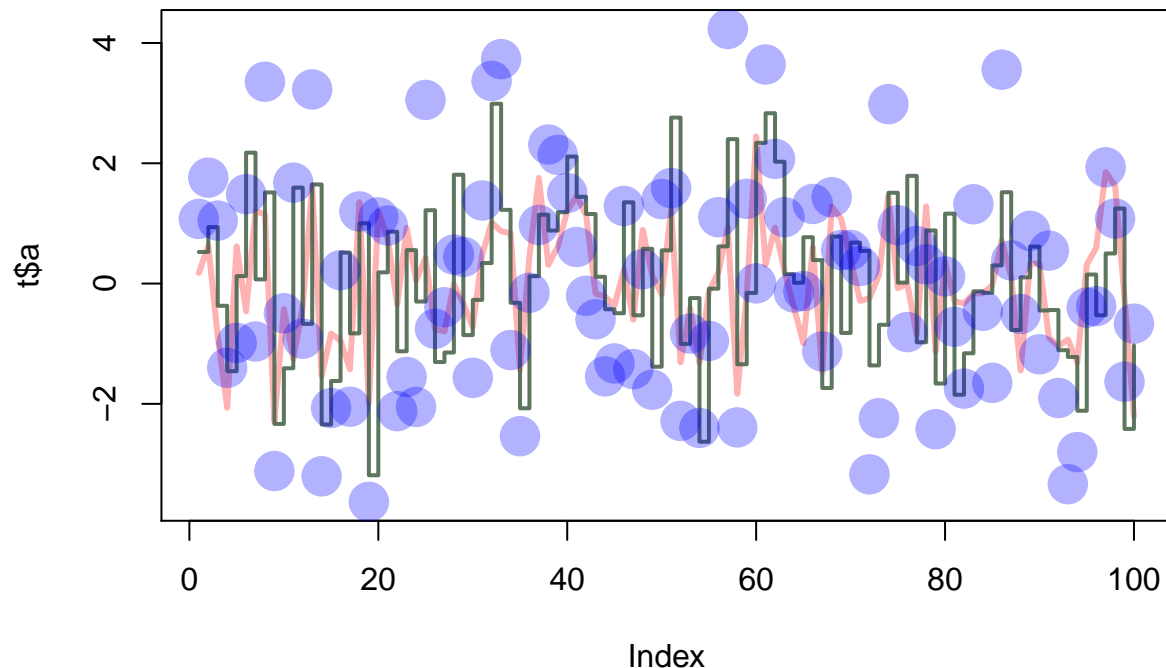
```
## [1] 1.315992
```

```
sd(t$c)
```

```
## [1] 1.801238
```

8) TODO

```
plot(t$a, type="l", ylim=range(t),
     lwd=3, col=rgb(1,0,0,0.3))
```

```r
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))
```


expla-
nation: rgb stands for red, green and blue. This is used to identify colour in the graph.

9) TODO

```r
sqrt (rnorm(100))
```

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

```
##   [1] 0.6840935 0.7835809       NaN 1.0674231 0.8086578 1.0160277       NaN
##   [8] 1.1080336       NaN       NaN       NaN       NaN       NaN 0.6239551
##  [15]       NaN       NaN       NaN       NaN 0.7056253       NaN 1.2984173
##  [22]       NaN       NaN       NaN 0.4562321       NaN       NaN       NaN
##  [29] 1.4113217       NaN 0.3867486       NaN 1.1269338       NaN 1.3023528
##  [36]       NaN       NaN 0.7602840       NaN       NaN       NaN       NaN
##  [43] 0.8410379 0.3377242 0.2663871 1.0702361       NaN       NaN 0.5709774
##  [50] 0.5882289       NaN       NaN 0.5931399 0.1213798       NaN 0.8084115
##  [57]       NaN       NaN 1.1681653 1.0084419 0.6149469       NaN 0.4573452
##  [64] 0.7992273       NaN       NaN       NaN 0.8889342 0.9324759       NaN
##  [71] 0.1623330 1.0481616       NaN       NaN       NaN 0.1716548       NaN
##  [78]       NaN       NaN 0.4283417 0.7421939       NaN 0.1519946       NaN
##  [85]       NaN 1.0391393       NaN 1.1150381       NaN       NaN 1.1225860
##  [92]       NaN       NaN 1.2951793 0.4691251 0.5581713 0.6998251 0.2771367
##  [99]       NaN       NaN
```

When sqrt is executed, NaNs is the output. This stands for not a number.

10) TODO

```r
dates= strptime(c("20190310", "20191224","20191215"),format="%Y%m%d")
presents = c(0,7,5)
plot(dates,presents)
lines(dates,presents)
```
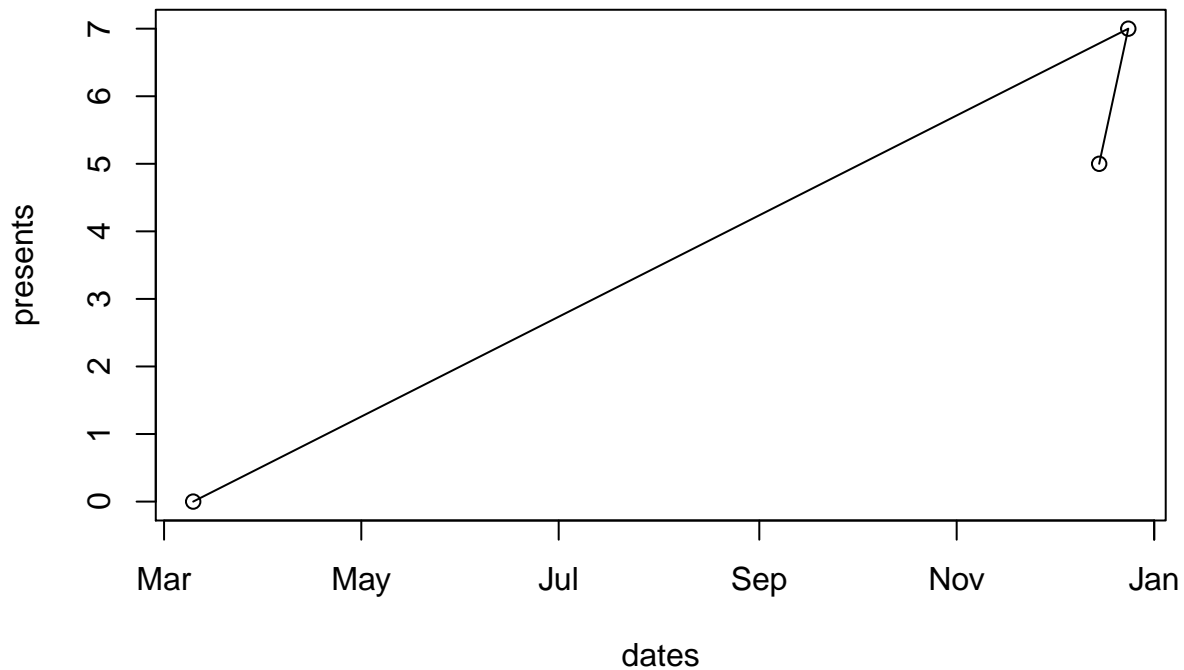
11) TODO

```r
k<- read.table(file="tst1.txt" , header =TRUE)
mytest <- k$g * 5
write.table (mytest, file="tst2.txt")
```

12) TODO

```r
vars = seq(from=1, to=100)
s = c()

for(i in 1:length(vars))
{
  if(vars[i] < 5) {
    s[i] <- (vars[i] *10)
  }else if (vars[i] > 90){
    s[i] <- (vars[i] *10)
  }else{
    s[i] = (vars[i]*0.1)
  }
}
s
```

```
##   [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) TODO

```
functions = function(argv)
{
  s2=c()
  for(i in 1:length(argv))
  {
    if(argv[i] < 5) {
      s2[i] <- (argv[i] *10)
    }else if (argv[i] > 90){
      s2[i] <- (argv[i] *10)
    }else{
      s2[i] = (argv[i]*0.1)
    }
  }
  s2
}

functions(argv = c(7:30))
```

```
##  [1] 0.7 0.8 0.9 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
## [18] 2.4 2.5 2.6 2.7 2.8 2.9 3.0
```