Name: Preyash Date: 30th September, 2022

Reg No. : 20BPS1022

Data Cleaning, Imputation and Statistics R code

Module -03

```
a <- 33
b <- 200
if (b > a) {
  print("b is greater than a")
}
```

```
## [1] "b is greater than a"
```

```
a <- 33
b <- 33
if (b > a) {
  print("b is greater than a")
}else if(a == b){
  print("a and b are equal")
}
```

```
## [1] "a and b are equal"
```

```
a <- 200
b <- 33
if (b > a) {
  print("b is greater than a")
}else if(a == b){
  print("a and b are equal")
}else{
  print("a is greater than b")
}
```

```
## [1] "a is greater than b"
```

```
x <- 41
if (x > 10) {
  print("Above ten")
  if (x > 20) {
    print("and also above 20!")
  } else {
    print("but not above 20.")
  }
} else {
```

```
  print("below 10.")
}
```

```
## [1] "Above ten"
## [1] "and also above 20!"
```

```
a <- 200
b <- 33
c <- 500
if (a > b & c > a) {
  print("Both conditions are true")
}
```

```
## [1] "Both conditions are true"
```

```
a <- 200
b <- 33
c <- 500
if (a > b | a > c) {
  print("At least one of the conditions are true")
}
```

```
## [1] "At least one of the conditions are true"
```

```
i<- 1
while(i<6){
  print(i)
  i <- i+1
  if(i==4){
    break
  }
}
```

```
## [1] 1
## [1] 2
## [1] 3
```

```
i<- 1
while(i<6){
  i <- i+1
  if(i==3){
    next
  }
  print(i)
}
```

```
## [1] 2
## [1] 4
## [1] 5
## [1] 6
```

```r
dice <- 1
while (dice <= 6)
  { if (dice < 6) {
    print("No")
  } else {
    print("Yes!")
  }
dice <- dice + 1
}
```

```
## [1] "No"
## [1] "No"
## [1] "No"
## [1] "No"
## [1] "No"
## [1] "Yes!"
```

```r
for(x in 1:10){
  print(x)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

```r
fruits <- list("banana","apple","cherry")

for(x in fruits){
  print(x)
}
```

```
## [1] "banana"
## [1] "apple"
## [1] "cherry"
```

```r
dice <- c(1,2,3,4,5,6)
for(x in dice){
  print(x)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
```

```
## [1] 5
## [1] 6
```

```
fruits <- list("banana","apple","cherry")

for(x in fruits){
  if(x == "cherry"){
    break
  }
  print(x)
}
```

```
## [1] "banana"
## [1] "apple"
```

```
fruits <- list("banana","apple","cherry")

for(x in fruits){
  if(x == "apple"){
    next
  }
  print(x)
}
```

```
## [1] "banana"
## [1] "cherry"
```

```
dice <- 1:6
for (x in dice) {
  if (x == 6) {
    print(paste("The dice number is", x, "Yes"))
  } else {
    print(paste("The dice number is", x, "No"))
  }
}
```

```
## [1] "The dice number is 1 No"
## [1] "The dice number is 2 No"
## [1] "The dice number is 3 No"
## [1] "The dice number is 4 No"
## [1] "The dice number is 5 No"
## [1] "The dice number is 6 Yes"
```

```
adj <- list("red", "big", "tasty")
fruits <- list("apple", "banana", "cherry")
for (x in adj) {
  for (y in fruits) {
    print(paste(x, y))
  }
}
```

```
## [1] "red apple"
## [1] "red banana"
## [1] "red cherry"
## [1] "big apple"
## [1] "big banana"
## [1] "big cherry"
## [1] "tasty apple"
## [1] "tasty banana"
## [1] "tasty cherry"

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.1.3

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

data<-data.frame(x1=1:6, x2=c(1,2,2,3,1,2),x3=c("f","d","j","e","a","d"))
data

##   x1 x2 x3
## 1  1  1  f
## 2  2  2  d
## 3  3  2  j
## 4  4  3  e
## 5  5  1  a
## 6  6  2  d

arrange(data,x3)

##   x1 x2 x3
## 1  5  1  a
## 2  2  2  d
## 3  6  2  d
## 4  4  3  e
## 5  1  1  f
## 6  3  2  j

filter(data,x2==2)

##   x1 x2 x3
## 1  2  2  d
## 2  3  2  j
## 3  6  2  d
```

```
mutate(data,x4=x1+x2)
```

```
##   x1 x2 x3 x4
## 1  1  1  f  2
## 2  2  2  d  4
## 3  3  2  j  5
## 4  4  3  e  7
## 5  5  1  a  6
## 6  6  2  d  8
```

```
getwd()
```

```
## [1] "D:/SEM5/LAB/CSE3505/LAB_30th_Sept"
```

```
setwd("D:\\SEM5\\LAB\\CSE3505\\LAB_30th_Sept")
getwd()
```

```
## [1] "D:/SEM5/LAB/CSE3505/LAB_30th_Sept"
```

```
data1=read.csv("D:\\SEM5\\LAB\\CSE3505\\LAB_30th_Sept\\Datasample.txt",header
=FALSE,sep=" ")
data1
```

```
##    V1 V2 V3
## 1 100  A  a
## 2 200  B  b
## 3 300  C  c
## 4 400  D  d
## 5 500  E  e
## 6 600  F  f
```

```
data2=read.csv("D:\\SEM5\\LAB\\CSE3505\\LAB_30th_Sept\\events.csv")
data2
```

```
##   student_id student_name event1 event2 event3 event4 sum
## 1          1      Preyash     10   10.0     10    9.0  39
## 2          2       Spidey      8    9.0      9   10.0  36
## 3          3        Lynda      9    7.0      9    4.0  29
## 4          4        Disha      6    8.0      8    7.0  29
## 5          5      Arshiya      7    8.0      9    8.0  32
## 6          0            0      8    8.4      9    7.6   0
```

```
x=c(1,3,4,5,10)
y=c(2,4,6,8,10)
z=c(10,12,14,16,18)
data3=cbind(x,y,z)
data3
```

```
##       x  y  z
## [1,]  1  2 10
## [2,]  3  4 12
## [3,]  4  6 14
```

```
## [4,]  5  8 16
## [5,] 10 10 18

write.csv(data3,file="D:\\SEM5\\LAB\\CSE3505\\LAB_30th_Sept\\practice.csv",ro
w.names=FALSE)
df=data.frame("name"=c("a","b","c"),"language"=c("r","p","j"),age=c(22,25,28)
)
df

##   name language age
## 1    a        r  22
## 2    b        p  25
## 3    c        j  28

write.table(df,file="D:\\SEM5\\LAB\\CSE3505\\LAB_30th_Sept\\p_text.txt",sep="
\t",row.names=TRUE,col.names=NA)

data(mtcars)
cars=mtcars
mtcars

##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
```

```
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

```
tail(mtcars)
```

```
##                  mpg cyl  disp  hp drat    wt qsec vs am gear carb
## Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.7  0  1    5    2
## Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.9  1  1    5    2
## Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.5  0  1    5    4
## Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.5  0  1    5    6
## Maserati Bora  15.0   8 301.0 335 3.54 3.570 14.6  0  1    5    8
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.6  1  1    4    2
```

```
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
names(mtcars)
```

```
##  [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
## [11] "carb"
```

```
rownames(mtcars)
```

```
##  [1] "Mazda RX4"           "Mazda RX4 Wag"       "Datsun 710"
##  [4] "Hornet 4 Drive"      "Hornet Sportabout"   "Valiant"
##  [7] "Duster 360"          "Merc 240D"           "Merc 230"
## [10] "Merc 280"            "Merc 280C"           "Merc 450SE"
## [13] "Merc 450SL"          "Merc 450SLC"         "Cadillac Fleetwood"
## [16] "Lincoln Continental" "Chrysler Imperial"   "Fiat 128"
## [19] "Honda Civic"         "Toyota Corolla"      "Toyota Corona"
## [22] "Dodge Challenger"    "AMC Javelin"         "Camaro Z28"
## [25] "Pontiac Firebird"    "Fiat X1-9"           "Porsche 914-2"
## [28] "Lotus Europa"        "Ford Pantera L"      "Ferrari Dino"
## [31] "Maserati Bora"       "Volvo 142E"
```

```
dim(mtcars)
```

```
## [1] 32 11
```

```r
sub1=cbind(mtcars$mpg,mtcars$cyl)
sub2=mtcars[4:8]
sub2
```

```
##                      hp drat    wt  qsec vs
## Mazda RX4           110 3.90 2.620 16.46  0
## Mazda RX4 Wag       110 3.90 2.875 17.02  0
## Datsun 710           93 3.85 2.320 18.61  1
## Hornet 4 Drive      110 3.08 3.215 19.44  1
## Hornet Sportabout   175 3.15 3.440 17.02  0
## Valiant             105 2.76 3.460 20.22  1
## Duster 360          245 3.21 3.570 15.84  0
## Merc 240D            62 3.69 3.190 20.00  1
## Merc 230             95 3.92 3.150 22.90  1
## Merc 280            123 3.92 3.440 18.30  1
## Merc 280C           123 3.92 3.440 18.90  1
## Merc 450SE          180 3.07 4.070 17.40  0
## Merc 450SL          180 3.07 3.730 17.60  0
## Merc 450SLC         180 3.07 3.780 18.00  0
## Cadillac Fleetwood  205 2.93 5.250 17.98  0
## Lincoln Continental 215 3.00 5.424 17.82  0
## Chrysler Imperial   230 3.23 5.345 17.42  0
## Fiat 128             66 4.08 2.200 19.47  1
## Honda Civic          52 4.93 1.615 18.52  1
## Toyota Corolla       65 4.22 1.835 19.90  1
## Toyota Corona        97 3.70 2.465 20.01  1
## Dodge Challenger    150 2.76 3.520 16.87  0
## AMC Javelin         150 3.15 3.435 17.30  0
## Camaro Z28          245 3.73 3.840 15.41  0
## Pontiac Firebird    175 3.08 3.845 17.05  0
## Fiat X1-9            66 4.08 1.935 18.90  1
## Porsche 914-2        91 4.43 2.140 16.70  0
## Lotus Europa        113 3.77 1.513 16.90  1
## Ford Pantera L      264 4.22 3.170 14.50  0
## Ferrari Dino        175 3.62 2.770 15.50  0
## Maserati Bora       335 3.54 3.570 14.60  0
## Volvo 142E          109 4.11 2.780 18.60  1
```

```r
sub3=mtcars[,c(2,6,8)]
sub3
```

```
##                     cyl    wt vs
## Mazda RX4             6 2.620  0
## Mazda RX4 Wag         6 2.875  0
## Datsun 710            4 2.320  1
## Hornet 4 Drive        6 3.215  1
## Hornet Sportabout     8 3.440  0
## Valiant               6 3.460  1
## Duster 360            8 3.570  0
## Merc 240D             4 3.190  1
```

```
## Merc 230              4 3.150  1
## Merc 280              6 3.440  1
## Merc 280C             6 3.440  1
## Merc 450SE            8 4.070  0
## Merc 450SL            8 3.730  0
## Merc 450SLC           8 3.780  0
## Cadillac Fleetwood    8 5.250  0
## Lincoln Continental   8 5.424  0
## Chrysler Imperial     8 5.345  0
## Fiat 128              4 2.200  1
## Honda Civic           4 1.615  1
## Toyota Corolla        4 1.835  1
## Toyota Corona         4 2.465  1
## Dodge Challenger      8 3.520  0
## AMC Javelin           8 3.435  0
## Camaro Z28            8 3.840  0
## Pontiac Firebird      8 3.845  0
## Fiat X1-9             4 1.935  1
## Porsche 914-2         4 2.140  0
## Lotus Europa          4 1.513  1
## Ford Pantera L        8 3.170  0
## Ferrari Dino          6 2.770  0
## Maserati Bora         8 3.570  0
## Volvo 142E            4 2.780  1
```

```
sub4=mtcars[c("mpg","cyl")]
sub4
```

```
##                      mpg cyl
## Mazda RX4           21.0   6
## Mazda RX4 Wag       21.0   6
## Datsun 710          22.8   4
## Hornet 4 Drive      21.4   6
## Hornet Sportabout   18.7   8
## Valiant             18.1   6
## Duster 360          14.3   8
## Merc 240D           24.4   4
## Merc 230            22.8   4
## Merc 280            19.2   6
## Merc 280C           17.8   6
## Merc 450SE          16.4   8
## Merc 450SL          17.3   8
## Merc 450SLC         15.2   8
## Cadillac Fleetwood  10.4   8
## Lincoln Continental 10.4   8
## Chrysler Imperial   14.7   8
## Fiat 128            32.4   4
## Honda Civic         30.4   4
## Toyota Corolla      33.9   4
## Toyota Corona       21.5   4
```

```
## Dodge Challenger    15.5   8
## AMC Javelin         15.2   8
## Camaro Z28          13.3   8
## Pontiac Firebird    19.2   8
## Fiat X1-9           27.3   4
## Porsche 914-2       26.0   4
## Lotus Europa        30.4   4
## Ford Pantera L      15.8   8
## Ferrari Dino        19.7   6
## Maserati Bora       15.0   8
## Volvo 142E          21.4   4
```

```
sub5=subset(mtcars,mpg>18)
sub5
```

```
##                    mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Fiat 128          32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic       30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla    33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona     21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Pontiac Firebird  19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9         27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2     26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa      30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ferrari Dino      19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Volvo 142E        21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

```
#table() Function
df=data.frame("Name"=c("abc","cde","def"),"Gender"=c("Male","Female","Male"))
df
```

```
##   Name Gender
## 1  abc   Male
## 2  cde Female
## 3  def   Male
```

```
table(df)
```

```
##      Gender
## Name  Female Male
##   abc      0    1
```

```
##   cde      1    0
##   def      0    1

x=c(1,2,NA,10,3)
is.na(x)

## [1] FALSE FALSE  TRUE FALSE FALSE

is.nan(x)

## [1] FALSE FALSE FALSE FALSE FALSE

x=c(1,2,NaN,NA,4)
is.na(x)

## [1] FALSE FALSE  TRUE  TRUE FALSE

is.nan(x)

## [1] FALSE FALSE  TRUE FALSE FALSE

#removing missing values
x=c(1,2,NA,4,NA,5)
bad=is.na(x)
x[!bad]

## [1] 1 2 4 5

x

## [1]  1  2 NA  4 NA  5

data=data.frame(x1=c(7,2,1,NA,9),x2=c(1,3,1,9,NA),x3=c(NA,8,8,NA,5))
data

##   x1 x2 x3
## 1  7  1 NA
## 2  2  3  8
## 3  1  1  8
## 4 NA  9 NA
## 5  9 NA  5

complete.cases(data)

## [1] FALSE  TRUE  TRUE FALSE FALSE

data_complete=data[complete.cases(data),]
data_complete

##   x1 x2 x3
## 2  2  3  8
## 3  1  1  8
```

```r
x=c(1,2,NA,4,NA,5)
y=c("a","b",NA,"d",NA,"f")
good=complete.cases(x,y)
good
```

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

```r
x[good]
```

```
## [1] 1 2 4 5
```

```r
y[good]
```

```
## [1] "a" "b" "d" "f"
```

```r
#airquality dataset
airquality[1:6,]
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

```r
summary(airquality)
```

```
##      Ozone           Solar.R           Wind             Temp      
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00  
##  1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00  
##  Median : 31.50   Median :205.0   Median : 9.700   Median :79.00  
##  Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88  
##  3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00  
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00  
##  NA's   :37       NA's   :7                                       
##      Month            Day      
##  Min.   :5.000   Min.   : 1.0  
##  1st Qu.:6.000   1st Qu.: 8.0  
##  Median :7.000   Median :16.0  
##  Mean   :6.993   Mean   :15.8  
##  3rd Qu.:8.000   3rd Qu.:23.0  
##  Max.   :9.000   Max.   :31.0  
##
```

```r
mean(airquality$Ozone,na.rm=TRUE)
```

```
## [1] 42.12931
```

```r
good=complete.cases(airquality)
airquality[good,][1:6,]
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 7    23     299  8.6   65     5   7
## 8    19      99 13.8   59     5   8
```

```
#na.omit
x=c(1,24,NA,6,NA,9)
x=na.omit(x)
x
```

```
## [1]  1 24  6  9
## attr(,"na.action")
## [1] 3 5
## attr(,"class")
## [1] "omit"
```

```
#data imputation
data=data.frame(marks1=c(NA,22,NA,49,75),marks2=c(81,14,NA,61,12),marks3=c(78
.5,19.325,NA,28,48.002))
data
```

```
##   marks1 marks2 marks3
## 1     NA     81 78.500
## 2     22     14 19.325
## 3     NA     NA     NA
## 4     49     61 28.000
## 5     75     12 48.002
```

```
#impute manually(method 1)
data$marks1[is.na(data$marks1)]=mean(data$marks1,na.rm=TRUE)
data
```

```
##     marks1 marks2 marks3
## 1 48.66667     81 78.500
## 2 22.00000     14 19.325
## 3 48.66667     NA     NA
## 4 49.00000     61 28.000
## 5 75.00000     12 48.002
```

```
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 4.1.3
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.1.3

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units

#using Hmisc
impute(data$marks2,median)

##     1     2     3     4     5
##  81.0  14.0 37.5*  61.0  12.0

#impute with a specific constant value
impute(data$marks3,2000)

##         1         2         3         4         5
##    78.500    19.325 2000.000*    28.000    48.002

#impute the entire dataset
all_column_median=apply(data,2,median,na.rm=TRUE)
#imputing median value with NA
for(i in colnames(data))
  data[,i][is.na(data[,i])]=all_column_median[i]
data

##     marks1 marks2 marks3
## 1 48.66667   81.0 78.500
## 2 22.00000   14.0 19.325
## 3 48.66667   37.5 38.001
## 4 49.00000   61.0 28.000
## 5 75.00000   12.0 48.002
```

Module-04

```
#Module 4
library(MASS)

## Warning: package 'MASS' was built under R version 4.1.3

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

painters

##                 Composition Drawing Colour Expression School
## Da Udine                 10       8     16          3      A
## Da Vinci                 15      16      4         14      A
## Del Piombo                8      13     16          7      A
## Del Sarto                12      16      9          8      A
## Fr. Penni                 0      15      8          0      A
## Guilio Romano            15      16      4         14      A
## Michelangelo              8      17      4          8      A
## Perino del Vaga          15      16      7          6      A
## Perugino                  4      12     10          4      A
## Raphael                  17      18     12         18      A
## F. Zucarro               10      13      8          8      B
## Fr. Salviata             13      15      8          8      B
## Parmigiano               10      15      6          6      B
## Primaticcio              15      14      7         10      B
## T. Zucarro               13      14     10          9      B
## Volterra                 12      15      5          8      B
## Barocci                  14      15      6         10      C
## Cortona                  16      14     12          6      C
## Josepin                  10      10      6          2      C
## L. Jordaens              13      12      9          6      C
## Testa                    11      15      0          6      C
## Vanius                   15      15     12         13      C
## Bassano                   6       8     17          0      D
## Bellini                   4       6     14          0      D
## Giorgione                 8       9     18          4      D
## Murillo                   6       8     15          4      D
## Palma Giovane            12       9     14          6      D
## Palma Vecchio             5       6     16          0      D
## Pordenone                 8      14     17          5      D
## Tintoretto               15      14     16          4      D
## Titian                   12      15     18          6      D
## Veronese                 15      10     16          3      D
## Albani                   14      14     10          6      E
## Caravaggio                6       6     16          0      E
```

```
## Corregio                  13      13     15         12      E
## Domenichino               15      17      9         17      E
## Guercino                  18      10     10          4      E
## Lanfranco                 14      13     10          5      E
## The Carraci               15      17     13         13      E
## Durer                      8      10     10          8      F
## Holbein                    9      10     16         13      F
## Pourbus                    4      15      6          6      F
## Van Leyden                 8       6      6          4      F
## Diepenbeck                11      10     14          6      G
## J. Jordaens               10       8     16          6      G
## Otho Venius               13      14     10         10      G
## Rembrandt                 15       6     17         12      G
## Rubens                    18      13     17         17      G
## Teniers                   15      12     13          6      G
## Van Dyck                  15      10     17         13      G
## Bourdon                   10       8      8          4      H
## Le Brun                   16      16      8         16      H
## Le Suer                   15      15      4         15      H
## Poussin                   15      17      6         15      H

painters$School

##  [1] A A A A A A A A A A B B B B B B C C C C C C D D D D D D D D D E E E
E E E
## [39] E F F F F G G G G G G H H H H
## Levels: A B C D E F G H

help(painters)

## starting httpd help server ... done

summary(painters)

##   Composition       Drawing          Colour        Expression
School
##  Min.   : 0.00   Min.   : 6.00   Min.   : 0.00   Min.   : 0.000   A
:10
##  1st Qu.: 8.25   1st Qu.:10.00   1st Qu.: 7.25   1st Qu.: 4.000   D
:10
##  Median :12.50   Median :13.50   Median :10.00   Median : 6.000   E      :
7
##  Mean   :11.56   Mean   :12.46   Mean   :10.94   Mean   : 7.667   G      :
7
##  3rd Qu.:15.00   3rd Qu.:15.00   3rd Qu.:16.00   3rd Qu.:11.500   B      :
6
##  Max.   :18.00   Max.   :18.00   Max.   :18.00   Max.   :18.000   C      :
6
##                                                                   (Other):
8
```

```r
#frequency distribution of qualitative data
school=painters$School
school.freq=table(school)
school.freq
```

```
## school
##  A  B  C  D  E  F  G  H
## 10  6  6 10  7  4  7  4
```

```r
cbind(school.freq)
```

```
##   school.freq
## A          10
## B           6
## C           6
## D          10
## E           7
## F           4
## G           7
## H           4
```

```r
#relative frequency distribution of the painter schools
school.relfreq=school.freq/nrow(painters)
school.relfreq
```

```
## school
##         A          B          C          D          E          F
G
## 0.18518519 0.11111111 0.11111111 0.18518519 0.12962963 0.07407407
0.12962963
##          H
## 0.07407407
```

```r
#enhanced
old=options(digits=1)
school.relfreq
```

```
## school
##    A    B    C    D    E    F    G    H
## 0.19 0.11 0.11 0.19 0.13 0.07 0.13 0.07
```

```r
options(old)
cbind(school.relfreq)
```

```
##   school.relfreq
## A     0.18518519
## B     0.11111111
## C     0.11111111
## D     0.18518519
## E     0.12962963
## F     0.07407407
```

```
## G     0.12962963
## H     0.07407407

options(old)

#quantitative data
faithful

##     eruptions waiting
## 1       3.600      79
## 2       1.800      54
## 3       3.333      74
## 4       2.283      62
## 5       4.533      85
## 6       2.883      55
## 7       4.700      88
## 8       3.600      85
## 9       1.950      51
## 10      4.350      85
## 11      1.833      54
## 12      3.917      84
## 13      4.200      78
## 14      1.750      47
## 15      4.700      83
## 16      2.167      52
## 17      1.750      62
## 18      4.800      84
## 19      1.600      52
## 20      4.250      79
## 21      1.800      51
## 22      1.750      47
## 23      3.450      78
## 24      3.067      69
## 25      4.533      74
## 26      3.600      83
## 27      1.967      55
## 28      4.083      76
## 29      3.850      78
## 30      4.433      79
## 31      4.300      73
## 32      4.467      77
## 33      3.367      66
## 34      4.033      80
## 35      3.833      74
## 36      2.017      52
## 37      1.867      48
## 38      4.833      80
## 39      1.833      59
## 40      4.783      90
## 41      4.350      80
```

```
## 42      1.883      58
## 43      4.567      84
## 44      1.750      58
## 45      4.533      73
## 46      3.317      83
## 47      3.833      64
## 48      2.100      53
## 49      4.633      82
## 50      2.000      59
## 51      4.800      75
## 52      4.716      90
## 53      1.833      54
## 54      4.833      80
## 55      1.733      54
## 56      4.883      83
## 57      3.717      71
## 58      1.667      64
## 59      4.567      77
## 60      4.317      81
## 61      2.233      59
## 62      4.500      84
## 63      1.750      48
## 64      4.800      82
## 65      1.817      60
## 66      4.400      92
## 67      4.167      78
## 68      4.700      78
## 69      2.067      65
## 70      4.700      73
## 71      4.033      82
## 72      1.967      56
## 73      4.500      79
## 74      4.000      71
## 75      1.983      62
## 76      5.067      76
## 77      2.017      60
## 78      4.567      78
## 79      3.883      76
## 80      3.600      83
## 81      4.133      75
## 82      4.333      82
## 83      4.100      70
## 84      2.633      65
## 85      4.067      73
## 86      4.933      88
## 87      3.950      76
## 88      4.517      80
## 89      2.167      48
## 90      4.000      86
## 91      2.200      60
```

```
## 92      4.333      90
## 93      1.867      50
## 94      4.817      78
## 95      1.833      63
## 96      4.300      72
## 97      4.667      84
## 98      3.750      75
## 99      1.867      51
## 100     4.900      82
## 101     2.483      62
## 102     4.367      88
## 103     2.100      49
## 104     4.500      83
## 105     4.050      81
## 106     1.867      47
## 107     4.700      84
## 108     1.783      52
## 109     4.850      86
## 110     3.683      81
## 111     4.733      75
## 112     2.300      59
## 113     4.900      89
## 114     4.417      79
## 115     1.700      59
## 116     4.633      81
## 117     2.317      50
## 118     4.600      85
## 119     1.817      59
## 120     4.417      87
## 121     2.617      53
## 122     4.067      69
## 123     4.250      77
## 124     1.967      56
## 125     4.600      88
## 126     3.767      81
## 127     1.917      45
## 128     4.500      82
## 129     2.267      55
## 130     4.650      90
## 131     1.867      45
## 132     4.167      83
## 133     2.800      56
## 134     4.333      89
## 135     1.833      46
## 136     4.383      82
## 137     1.883      51
## 138     4.933      86
## 139     2.033      53
## 140     3.733      79
## 141     4.233      81
```

```
## 142     2.233      60
## 143     4.533      82
## 144     4.817      77
## 145     4.333      76
## 146     1.983      59
## 147     4.633      80
## 148     2.017      49
## 149     5.100      96
## 150     1.800      53
## 151     5.033      77
## 152     4.000      77
## 153     2.400      65
## 154     4.600      81
## 155     3.567      71
## 156     4.000      70
## 157     4.500      81
## 158     4.083      93
## 159     1.800      53
## 160     3.967      89
## 161     2.200      45
## 162     4.150      86
## 163     2.000      58
## 164     3.833      78
## 165     3.500      66
## 166     4.583      76
## 167     2.367      63
## 168     5.000      88
## 169     1.933      52
## 170     4.617      93
## 171     1.917      49
## 172     2.083      57
## 173     4.583      77
## 174     3.333      68
## 175     4.167      81
## 176     4.333      81
## 177     4.500      73
## 178     2.417      50
## 179     4.000      85
## 180     4.167      74
## 181     1.883      55
## 182     4.583      77
## 183     4.250      83
## 184     3.767      83
## 185     2.033      51
## 186     4.433      78
## 187     4.083      84
## 188     1.833      46
## 189     4.417      83
## 190     2.183      55
## 191     4.800      81
```

```
## 192     1.833      57
## 193     4.800      76
## 194     4.100      84
## 195     3.966      77
## 196     4.233      81
## 197     3.500      87
## 198     4.366      77
## 199     2.250      51
## 200     4.667      78
## 201     2.100      60
## 202     4.350      82
## 203     4.133      91
## 204     1.867      53
## 205     4.600      78
## 206     1.783      46
## 207     4.367      77
## 208     3.850      84
## 209     1.933      49
## 210     4.500      83
## 211     2.383      71
## 212     4.700      80
## 213     1.867      49
## 214     3.833      75
## 215     3.417      64
## 216     4.233      76
## 217     2.400      53
## 218     4.800      94
## 219     2.000      55
## 220     4.150      76
## 221     1.867      50
## 222     4.267      82
## 223     1.750      54
## 224     4.483      75
## 225     4.000      78
## 226     4.117      79
## 227     4.083      78
## 228     4.267      78
## 229     3.917      70
## 230     4.550      79
## 231     4.083      70
## 232     2.417      54
## 233     4.183      86
## 234     2.217      50
## 235     4.450      90
## 236     1.883      54
## 237     1.850      54
## 238     4.283      77
## 239     3.950      79
## 240     2.333      64
## 241     4.150      75
```

```
## 242     2.350      47
## 243     4.933      86
## 244     2.900      63
## 245     4.583      85
## 246     3.833      82
## 247     2.083      57
## 248     4.367      82
## 249     2.133      67
## 250     4.350      74
## 251     2.200      54
## 252     4.450      83
## 253     3.567      73
## 254     4.500      73
## 255     4.150      88
## 256     3.817      80
## 257     3.917      71
## 258     4.450      83
## 259     2.000      56
## 260     4.283      79
## 261     4.767      78
## 262     4.533      84
## 263     1.850      58
## 264     4.250      83
## 265     1.983      43
## 266     2.250      60
## 267     4.750      75
## 268     4.117      81
## 269     2.150      46
## 270     4.417      90
## 271     1.817      46
## 272     4.467      74
```

```
head(faithful)
```

```
##   eruptions waiting
## 1     3.600      79
## 2     1.800      54
## 3     3.333      74
## 4     2.283      62
## 5     4.533      85
## 6     2.883      55
```

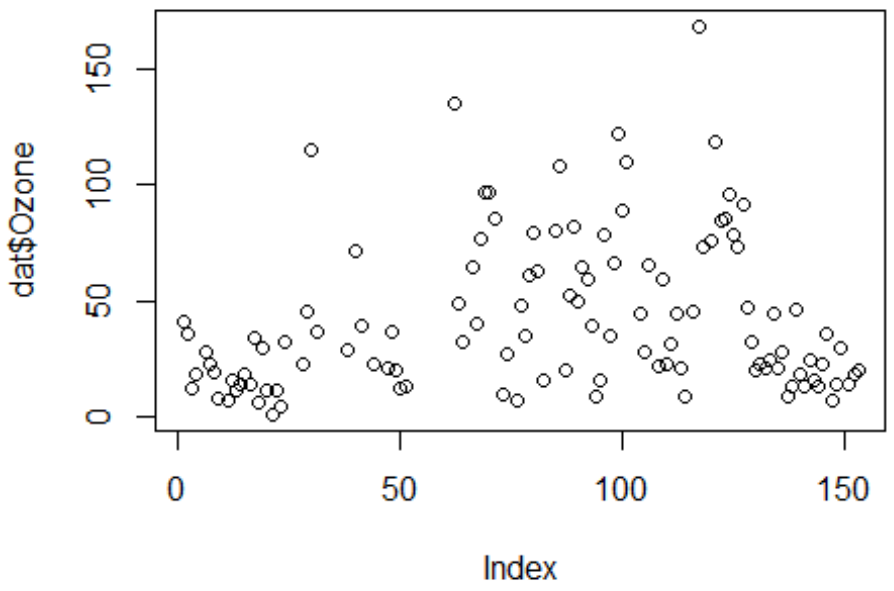
```
#range of eruption duration
duration=faithful$eruptions
range(duration)
```

```
## [1] 1.6 5.1
```

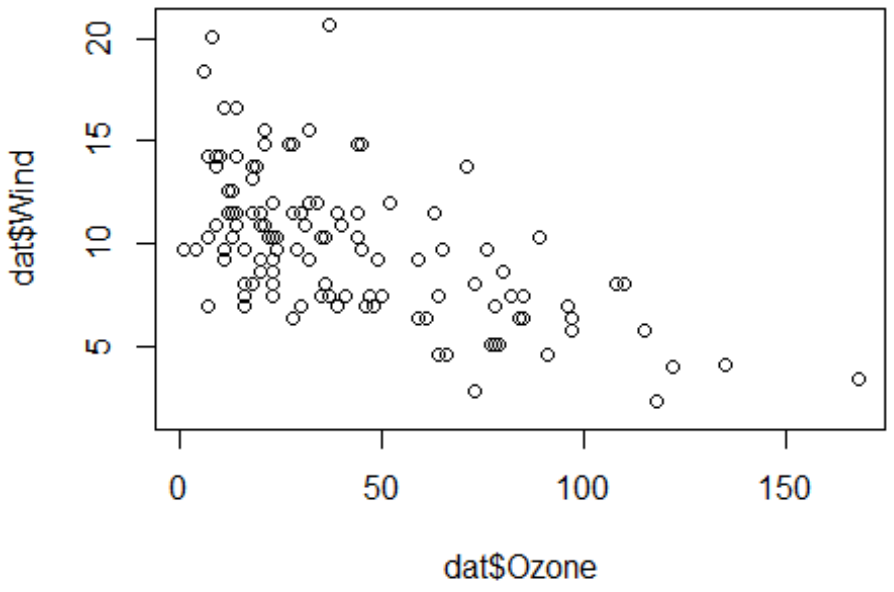
```
#break the range into non-overlapping sub-intervals
breaks=seq(1.5,5.5, by=0.5)
breaks
```

```
## [1] 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5

#CUT METHOD
duration.cut=cut(duration,breaks,right=FALSE)
#frequency of eruptions in each sub-interval with table function
duration.freq=table(duration.cut)

#cumulative frequency
duration.cumfreq=cumsum(duration.freq)
duration.cumfreq

## [1.5,2) [2,2.5) [2.5,3) [3,3.5) [3.5,4) [4,4.5) [4.5,5) [5,5.5)
##      51      92      97     104     134     207     268     272

#mean
mean(duration)

## [1] 3.487783

#median
median(duration)

## [1] 4

#quartiles
quantile(duration)

##      0%     25%     50%     75%    100%
## 1.60000 2.16275 4.00000 4.45425 5.10000

#percentile
quantile(duration,c(.32,.57,.98))

##     32%     57%     98%
## 2.39524 4.13300 4.93300

#range
max(duration)-min(duration)

## [1] 3.5

#interquartile range
IQR(duration)

## [1] 2.2915

#variance
var(duration)

## [1] 1.302728

#standard deviation
sd(duration)
```

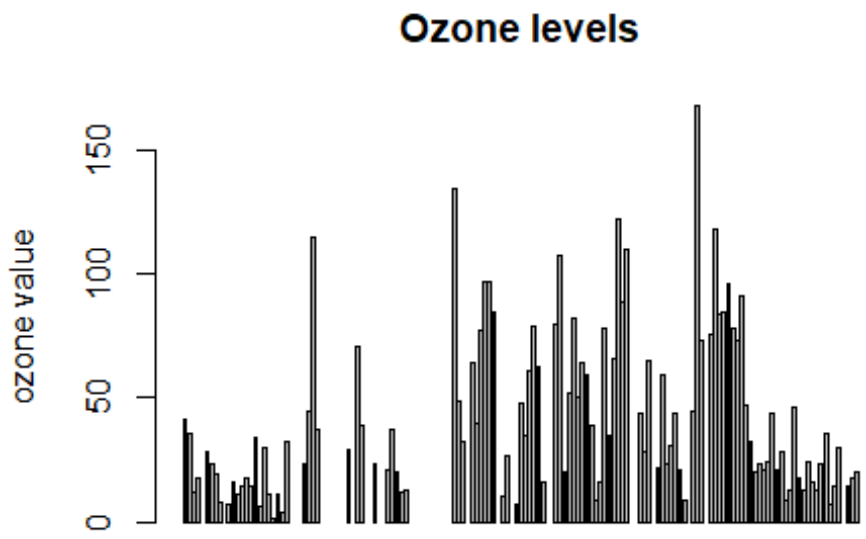
```
## [1] 1.141371

library(e1071)

## Warning: package 'e1071' was built under R version 4.1.3

##
## Attaching package: 'e1071'

## The following object is masked from 'package:Hmisc':
##
##     impute

#central moment
moment(duration,order=3,center=TRUE)

## [1] -0.6149059

#skewness
skewness(duration)

## [1] -0.4135498

#kurtosis
kurtosis(duration)

## [1] -1.511605

#visualisation
dat=airquality
plot(dat$Ozone)
```



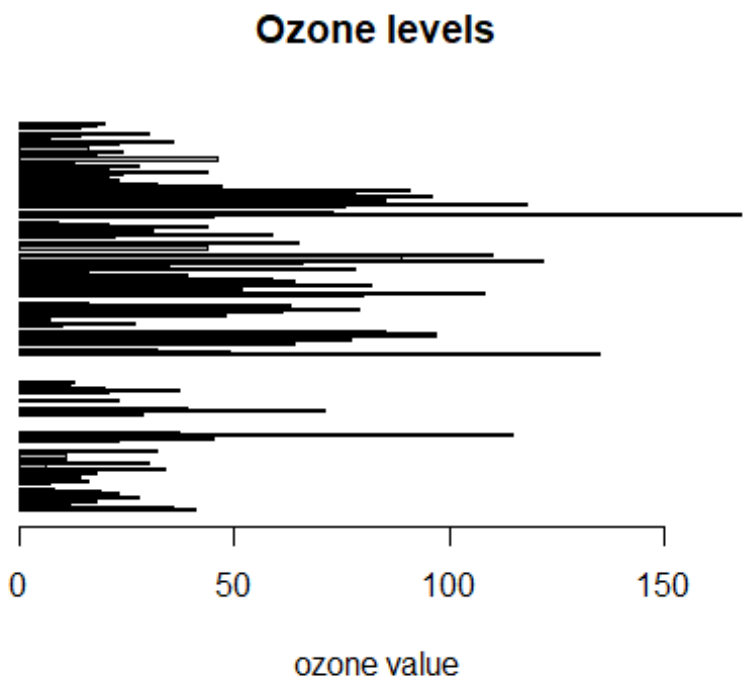

```
plot(dat$Ozone,dat$Wind)
```



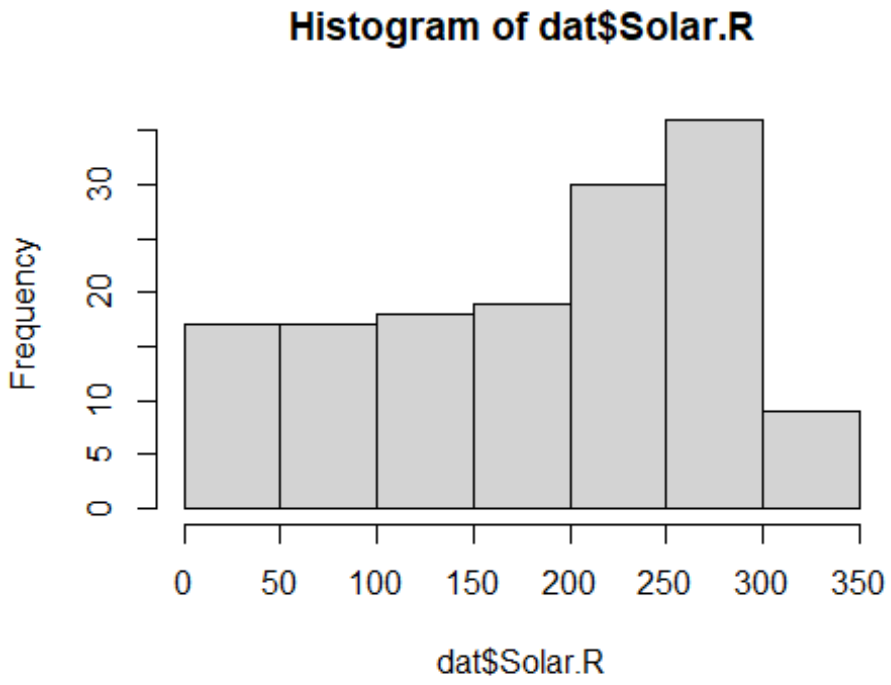
```r
#bar plot
barplot(dat$Ozone,main='Ozone levels',ylab='ozone value')
```



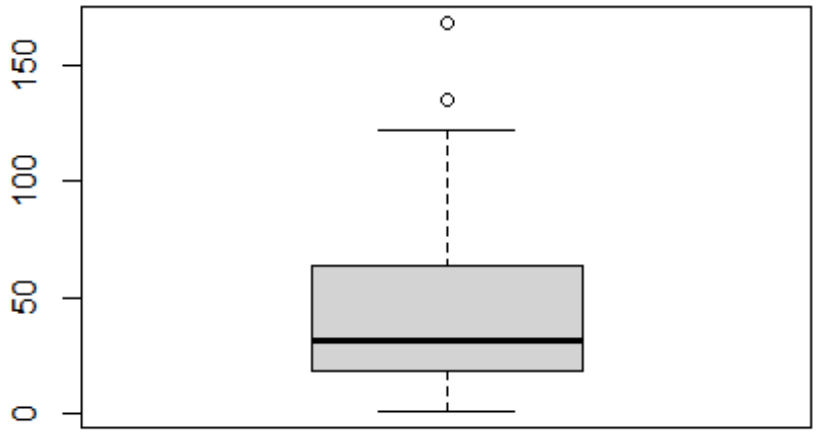

```r
barplot(dat$Ozone,main='Ozone levels',xlab='ozone value',horiz=TRUE)
```



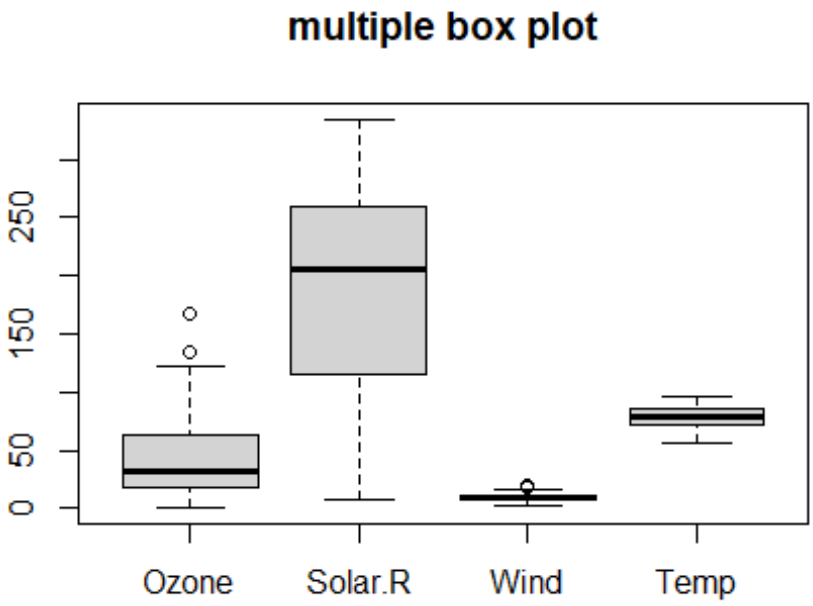

```
#histogram
hist(dat$Solar.R)
```



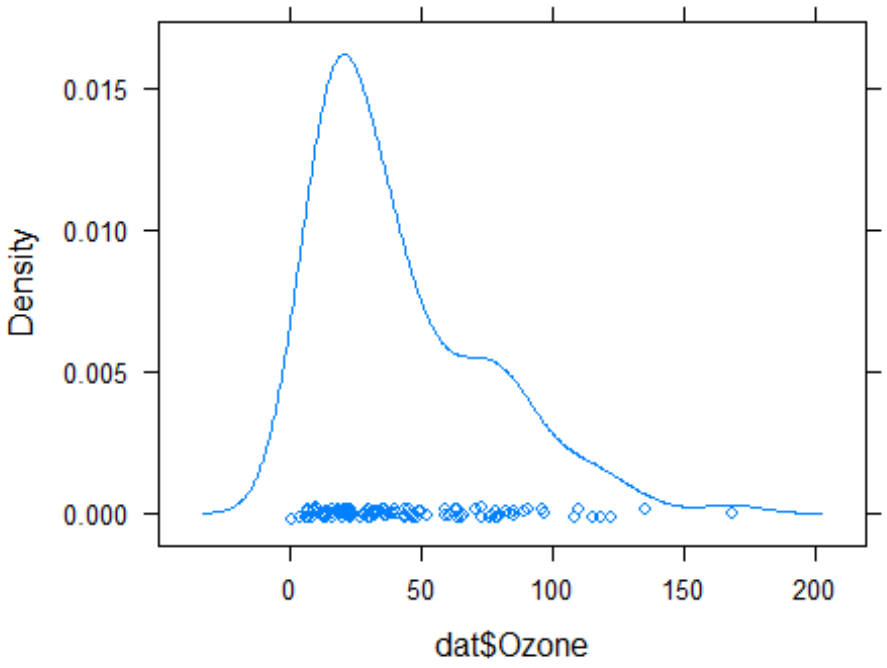
```r
#boxplot
boxplot(dat$Ozone)
```



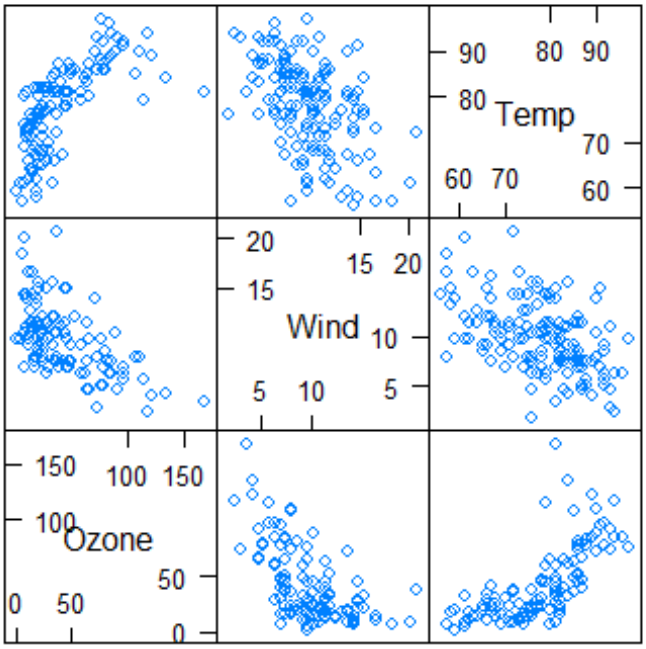

```r
boxplot(dat[,1:4],main='multiple box plot')
```




```
#lattice graphs
library(lattice)
densityplot(dat$Ozone)
```

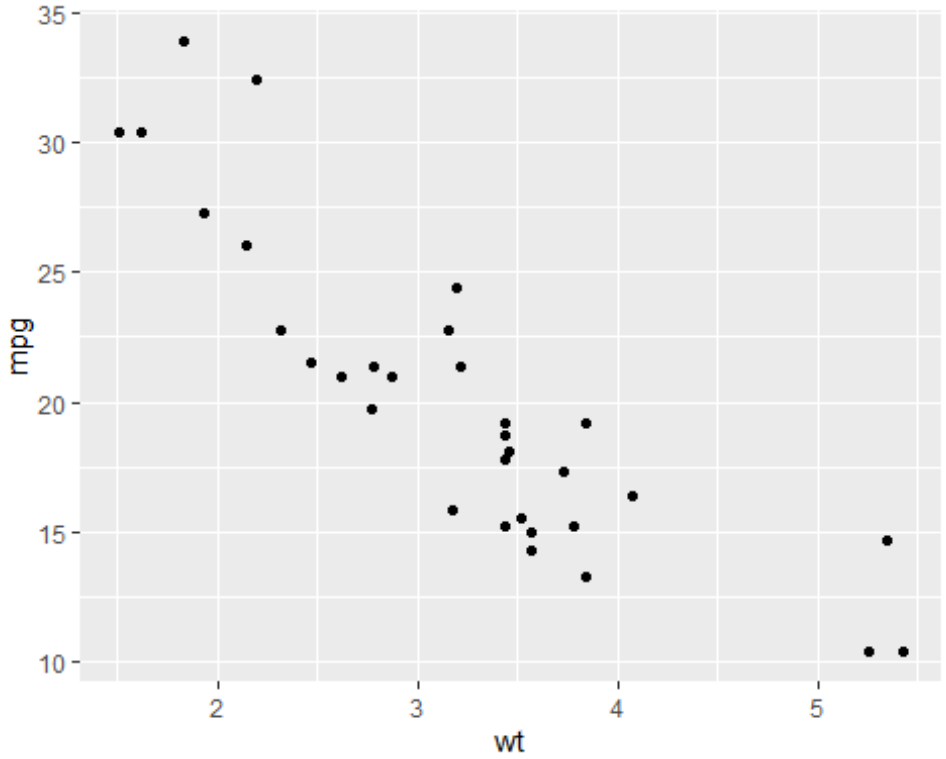


```
splom(dat[c(1,3,4)])
```

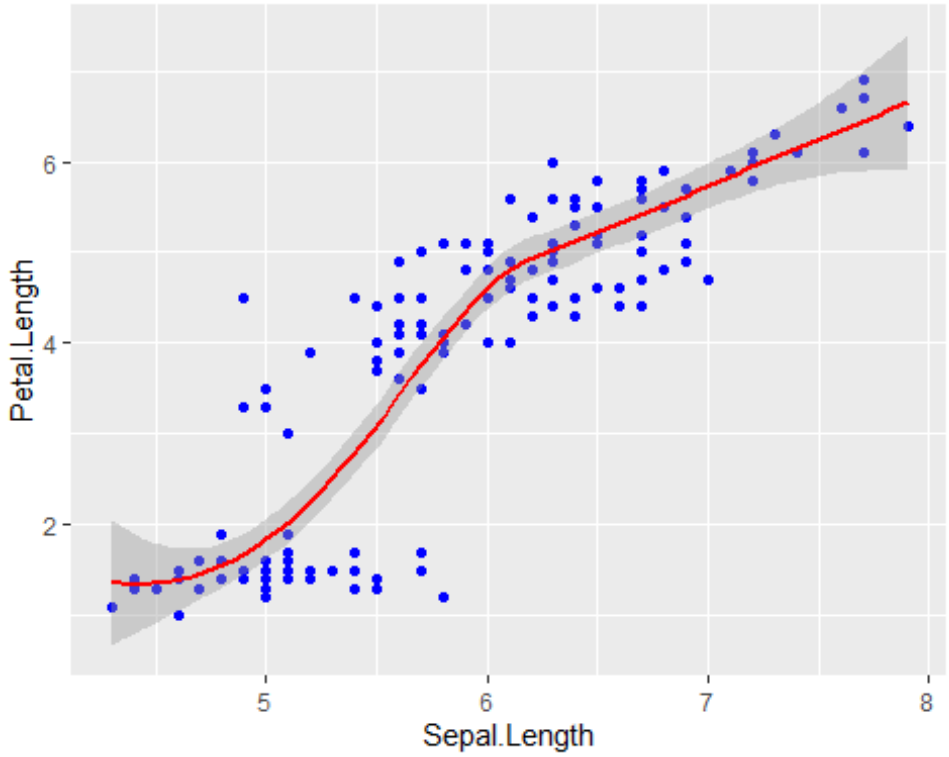


Scatter Plot Matrix

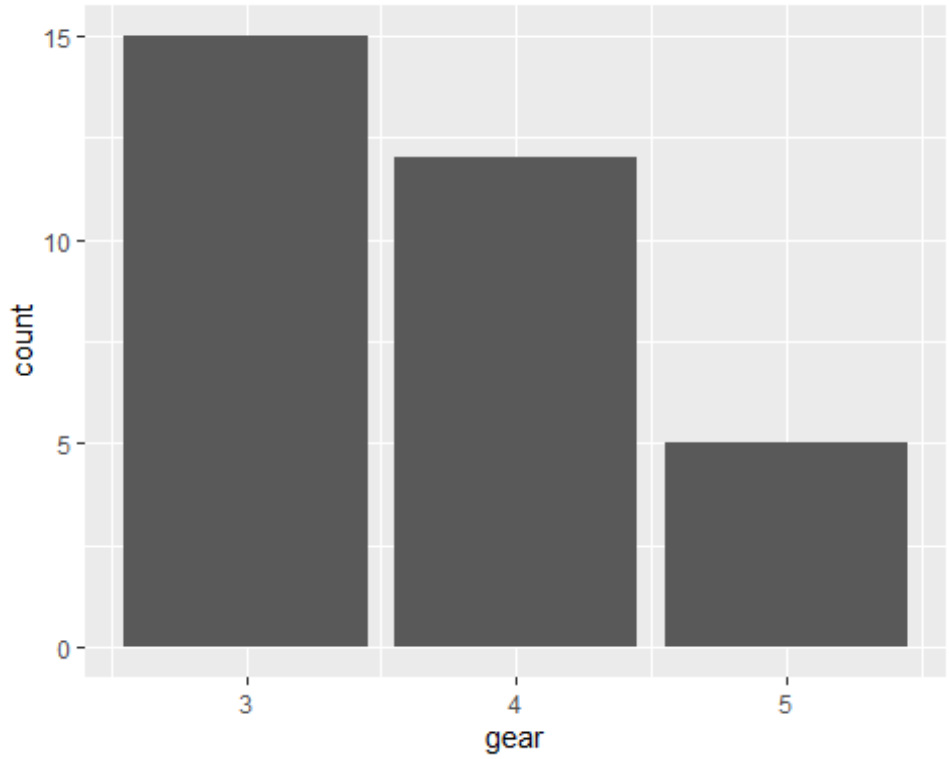
```
#ggplot
library(ggplot2)
ggplot(data = mtcars,
mapping = aes(x=wt,y=mpg))+geom_point()
```




```
ggplot(iris,aes(x=Sepal.Length,y=Petal.Length,col=Species))+geom_point(color=
"blue")+geom_smooth(color="red")

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

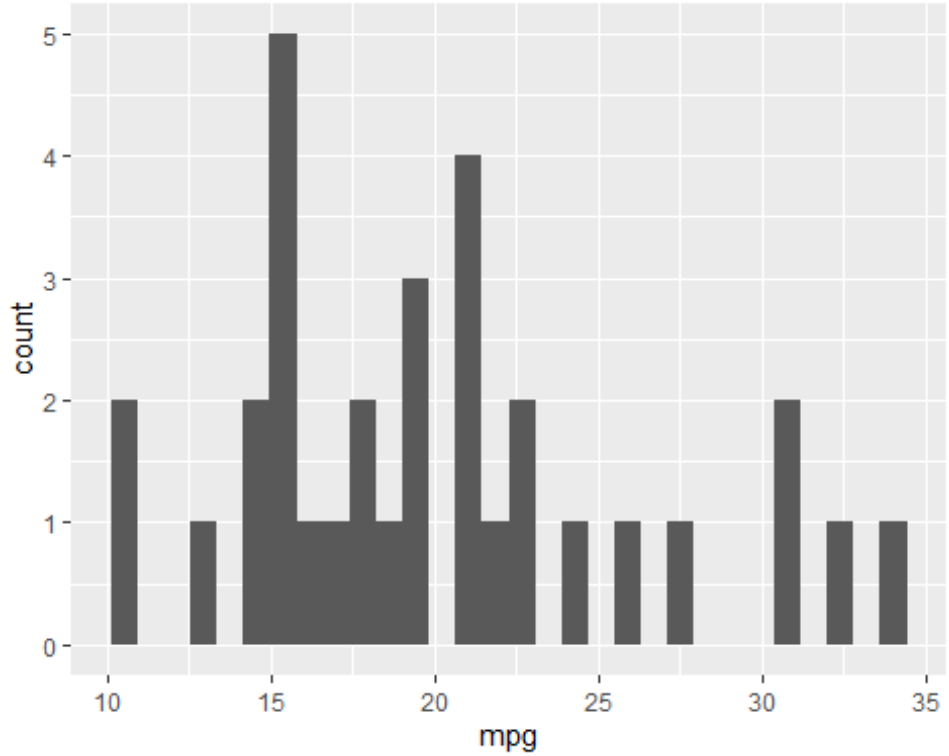



```
ggplot(mtcars,aes(x=gear))+geom_bar()
```

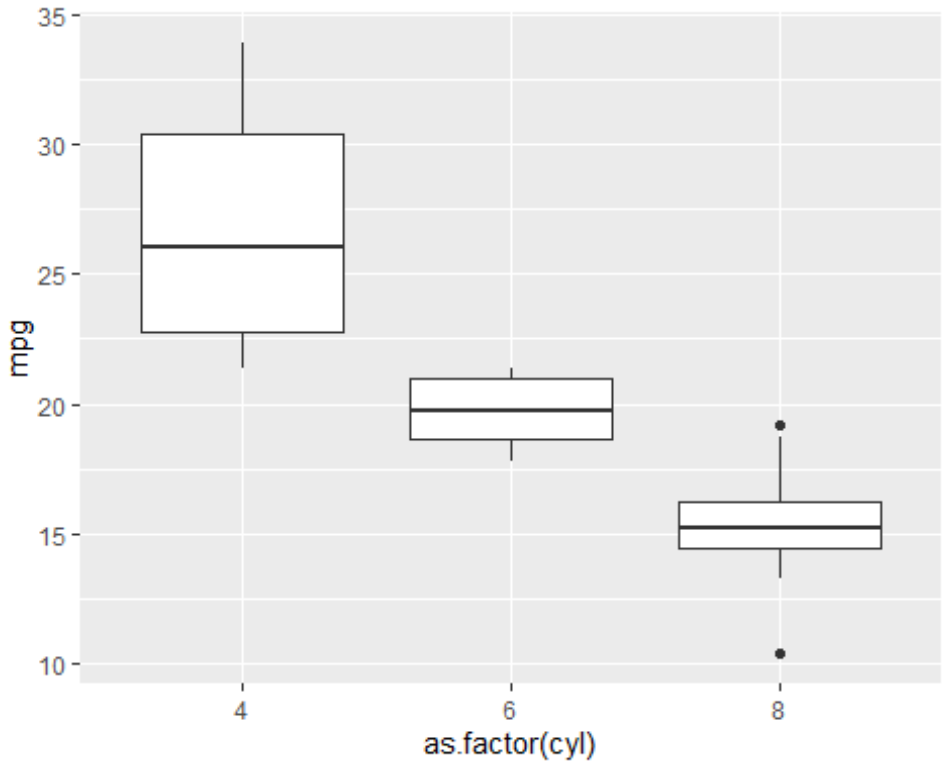


```r
#histogram
ggplot(mtcars,aes(x=mpg))+geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
#boxplot
ggplot(mtcars,aes(x=as.factor(cyl),y=mpg))+geom_boxplot()
```


35
30
25
20
15
10
mpg
4
6
8
as.factor(cyl)