

# **STATISTICAL DATA ANALYSIS ASSIGNMENT**

**STUDENT NAME : SHIVAANI KATRAGADDA**

**STUDENT NUMBER : R00183214**

**SUBJECT : STATISTICAL DATA ANALYSIS**

**SUBJECT CODE : STAT9004**

**PROFESSOR : DR.CATHERINE PALMER**

**DATE : 06-05-2020**

**DAY : WEDNESDAY**

In this assignment, I was given two datasets namely treeB.csv and divusaB.csv

The first dataset treeB.csv consists of 60 observations(rows) and 2 variables(columns).The two columns of the dataset are Diam and Vol which measure the diameter (m) and the volume (m<sup>3</sup>) of n tree trunks from the same species.

The second dataset divusaB.csv consists of 77 observations(rows) and 7 variables(columns). The columns of the dataset are as follows  
Year,divorce,unemployed,femlab,marriage,birth,military.

The detailed explanation of the variables are as follows

year - the year from 1920-1996

divorce - per 1000 women aged 15 or more

unemployed - unemployment rate

femlab - percent female participation in labour force aged 16+

marriage - marriages per 1000 unmarried women aged 16+

birth - births per 1000 women aged 15-44

military - military personnel per 1000 population This is a little introduction to the given datasets.

The packages used in the assignment are as follows

```
# install.packages("readr")
# install.packages("DataExplorer")
# install.packages("dplyr")
# install.packages("inspectdf")
# install.packages("caret")
# install.packages("GGally")
# install.packages("skimr")
# install.packages("funModeling")
# install.packages("scatterplot3d")
# install.packages("car")
# install.packages("corrplot")
# install.packages("RColorBrewer")
# install.packages("VIM")
# install.packages("outliers")
# install.packages("psych")
# install.packages("leaps")
library(readr)

## Warning: package 'readr' was built under R version 3.6.3

library(DataExplorer)

## Warning: package 'DataExplorer' was built under R version 3.6.3

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.6.2

##
## 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

library(inspectdf)

## Warning: package 'inspectdf' was built under R version 3.6.3

library(caret)

## Warning: package 'caret' was built under R version 3.6.3

## Loading required package: lattice
```

```
## Loading required package: ggplot2
## Warning: package 'ggplot2' was built under R version 3.6.2
library(GGally)
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg      ggplot2
##
## Attaching package: 'GGally'
## The following object is masked from 'package:dplyr':
##
##   nasa
library(skimr)
## Warning: package 'skimr' was built under R version 3.6.3
library(funModeling)
## Warning: package 'funModeling' was built under R version 3.6.3
## Loading required package: Hmisc
## Warning: package 'Hmisc' was built under R version 3.6.3
## Loading required package: survival
## Warning: package 'survival' was built under R version 3.6.3
##
## Attaching package: 'survival'
## The following object is masked from 'package:caret':
##
##   cluster
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:dplyr':
##
##   src, summarize
## The following objects are masked from 'package:base':
##
##   format.pval, units
```

```
## funModeling v.1.9.3 :)
## Examples and tutorials at livebook.datascienceheroes.com
## / Now in Spanish: librovivodecienciadedatos.ai

##
## Attaching package: 'funModeling'

## The following object is masked from 'package:GGally':
##
##     range01

library(scatterplot3d)
library(car)

## Warning: package 'car' was built under R version 3.6.2
## Loading required package: carData

##
## Attaching package: 'car'

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

library(corrplot)

## Warning: package 'corrplot' was built under R version 3.6.3
## corrplot 0.84 loaded

library(RColorBrewer)
library(VIM)

## Warning: package 'VIM' was built under R version 3.6.2
## Loading required package: colorspace
## Loading required package: grid
## Loading required package: data.table
## Warning: package 'data.table' was built under R version 3.6.2
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## VIM is ready to use.
## Since version 4.0.0 the GUI is in its own package VIMGUI.
```

```
##
##           Please use the package to use the new (and old) GUI.
## Suggestions and bug-reports can be submitted at: https://github.com/alexkow/VIM/issues
##
## Attaching package: 'VIM'
## The following object is masked from 'package:datasets':
##
##     sleep
library(outliers)
library(psych)
## Warning: package 'psych' was built under R version 3.6.3
##
## Attaching package: 'psych'
## The following object is masked from 'package:outliers':
##
##     outlier
## The following object is masked from 'package:car':
##
##     logit
## The following object is masked from 'package:Hmisc':
##
##     describe
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
library(leaps)
## Warning: package 'leaps' was built under R version 3.6.3
```

readr - readxl package useful to get the data from excel to R easily.

DataExplorer - The package scans and analyzes each variable, and visualizes them with typical graphical techniques.

dplyr - dplyr package which is useful for manipulating datasets in R very effectively. This package is used for the function `glimpse()` which shows as much data as possible

inspectdf - inspectdf is a collection of utilities for columns summary, comparison, and visualization of data frames.

caret - Caret(Classification And REgression Training) package contains a set of functions that are useful for creating predictive models

GGally - GGally package which extends ggplot2 by adding several functions to reduce the complexity of combining geoms with transformed data. This package is used for ggpairs and ggcorr.

skimr - skimr is designed to provide summary statistics about variables.

funModeling - This package is used for data cleaning, importance variable analysis, and model performance

scatterplot3d - scatterplot3d returns a list of function closures that can be used to add elements on an existing plot.

car - The package contains mostly functions for applied regression, linear models, and generalized linear models, with an emphasis on regression diagnostics, particularly graphical diagnostic methods

corrplot-The R package corrplot is for visualizing correlation matrices and confidence intervals. It also contains some algorithms to do matrix reordering.

RColorBrewer-RColorBrewer is an R package that contains a ready-to-use color palette for creating beautiful graphics.

VIM- Visualization and Imputation of Missing Values. Description. This package introduces new tools for the visualization of missing or imputed values, which can be used for exploring the data and the structure of the missing or imputed values.

outliers- This package contains a collection of some tests commonly used for identifying outliers.

psych-Functions are primarily for multivariate analysis and scale construction using factor analysis, principal component analysis, cluster analysis, and reliability analysis, although others provide basic descriptive statistics.

leaps - The R package leaps have a function regsubsets that can be used for best subsets, forward selection and backward elimination depending on which approach is considered most appropriate for the application under consideration.

## QUESTION 1

A researcher would like to explore the relationship between the diameter of a tree trunk and the volume of the trunk for a particular species of tree. The tree.xlsx dataset is available on Canvas. The dataset consists of n observations of 2 variables, Diam and Vol which measure the diameter (m) and the volume (m<sup>3</sup>) of n tree trunks from the same species.

The aim of the question is to find out whether the variable Diam, is of use in predicting the value of Vol. Please write up the analysis in the form of a report.

As per the given question the main aim of the question is to find out whether Diam(Diameter) is useful to predict the value of vol(volume).

Therefore I loaded the first dataset treeB.csv by using readr package which will be helpful for reading .csv files. And assigned file to TreeB variable.

```
#Loading the treeb.csv file and storing it in variable TreeB
TreeB<-read.csv("F:/semester2/SDA/treeB.csv")
View(TreeB)
#checking the class of the TreeB
class(TreeB)

## [1] "data.frame"
```

## Question 1(a)

Make a numerical and graphical summary of the data, commenting on the results. Include: boxplots, histograms, scatterplot and the correlation coefficient.

## Solution

I have to find the numerical and graphical summary of the data

So,I will be performing the Exploratory Data Analysis(EDA) to know about the data.

EDA: Exploratory data analysis (EDA) is an approach to the study of data sets, often using visual tools, to outline their basic characteristics. Exploratory Data Analysis performs two main things: 1. It helps clean up a dataset. 2. It gives you a better view of the variables and their relationships. There are mainly three components of exploring data: (i)Understanding your variables (Numerical data Analysis) (ii)Cleaning your dataset (iii)Analyzing relationships between variables (Graphical Data Analysis)

- (i) Understanding your variables (Numerical data Analysis) Firstly I want to know about the given dataset so i will be going through the dataset. In order to understand or to explore dataset i used different packages such as DataExplorer,dplyr,visdat,inspectdf,GGally,skimr,funModeling

```
#Looking in to the data by using dim(which gives dimensions of data),names(column names), head,tail,structure and the summary of the data treeB
dim(TreeB)

## [1] 60  2

names(TreeB)

## [1] "Diam" "Vol"
```

```
head(TreeB)
```

```
##      Diam    Vol
## 1 0.2122 0.4150
## 2 0.1351 0.4030
## 3 0.4656 1.7220
## 4 0.1193 0.0278
## 5 0.3009 0.8801
## 6 0.3787 6.8492
```

```
tail(TreeB)
```

```
##      Diam    Vol
## 55 0.2642 0.8113
## 56 0.2402 0.6318
## 57 0.2040 0.1146
## 58 0.2891 0.7034
## 59 0.1992 0.4490
## 60 0.1642 0.0886
```

```
str(TreeB)
```

```
## 'data.frame':    60 obs. of  2 variables:
## $ Diam: num  0.212 0.135 0.466 0.119 0.301 ...
## $ Vol : num  0.415 0.403 1.722 0.0278 0.8801 ...
```

```
summary(TreeB)
```

```
##      Diam          Vol
## Min.   :0.1060   Min.   :0.0278
## 1st Qu.:0.2067   1st Qu.:0.3309
## Median :0.2750   Median :0.5964
## Mean   :0.2895   Mean    :1.0612
## 3rd Qu.:0.3589   3rd Qu.:1.2084
## Max.   :0.6356   Max.    :6.8492
```

```
glimpse(TreeB)
```

```
## Observations: 60
## Variables: 2
## $ Diam <dbl> 0.2122, 0.1351, 0.4656, 0.1193, 0.3009, 0.3787, 0.2318, 0.205
9...
## $ Vol <dbl> 0.4150, 0.4030, 1.7220, 0.0278, 0.8801, 6.8492, 0.3458, 0.066
6...
```

*#For each variable it returns, Quantity and percentage of zeros, NA values (q\_*  
*NA/p\_na), and infinite values (q\_inf/p\_inf)*

```
df_status(TreeB)
```

```
##   variable q_zeros p_zeros q_na p_na q_inf p_inf   type unique
## 1     Diam      0      0    0    0    0    0 numeric     59
## 2      Vol      0      0    0    0    0    0 numeric     60
```



*#prints a concise statistical summary*

```
describe(TreeB)
```

```
##      vars  n mean   sd median trimmed  mad   min   max range skew kurtosis
se
## Diam    1 60 0.29 0.11   0.27   0.28 0.10 0.11 0.64  0.53 0.83    0.79 0
.01
## Vol     2 60 1.06 1.33   0.60   0.76 0.49 0.03 6.85  6.82 2.56    6.86 0
.17
```

*#provide summary statistics about variables*

```
skim(TreeB)
```

*Data summary*

Name	TreeB
Number of rows	60
Number of columns	2

---


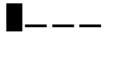
Column type frequency:

numeric	2
---------	---

---

Group variables	None
-----------------	------

**Variable type: numeric**

skim_vari ble	n_missi ng	complete_r ate	mea n	sd	p0	p2 5	p5 0	p7 5	p10 0	hist
Diam	0	1	0.29	0.1	0.1	0.2	0.2	0.3	0.64	
Vol	0	1	1.06	1.3	0.0	0.3	0.6	1.2	6.85	

*#descriptive stastics for the data mean,median,standard deviation,variance,IQ R and normality by using shaprio test*

```
mean(TreeB$Diam)
```

```
## [1] 0.2895333
```

```
mean(TreeB$Vol)
```

```
## [1] 1.061188
```

```
median(TreeB$Diam)
```

```
## [1] 0.27495
```

```
median(TreeB$Vol)
```

```

## [1] 0.59635
sd(TreeB$Diam)
## [1] 0.1109486
sd(TreeB$Vol)
## [1] 1.334955
var(TreeB$Diam)
## [1] 0.01230958
var(TreeB$Vol)
## [1] 1.782104
IQR(TreeB$Diam)
## [1] 0.152125
IQR(TreeB$Vol)
## [1] 0.87755

#NORMALITY
shapiro.test(TreeB$Diam)

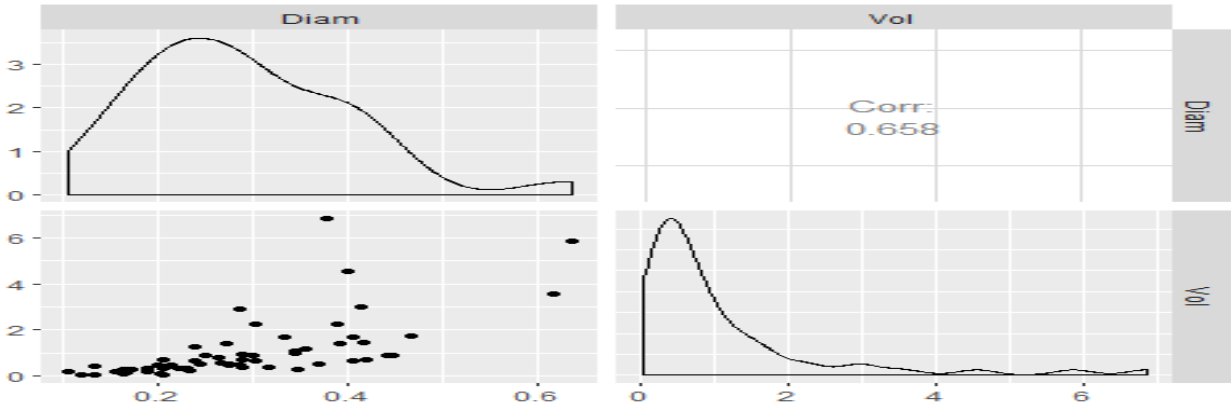
##
##  Shapiro-Wilk normality test
##
## data:  TreeB$Diam
## W = 0.94756, p-value = 0.01198

shapiro.test(TreeB$Vol)

##
##  Shapiro-Wilk normality test
##
## data:  TreeB$Vol
## W = 0.66979, p-value = 2.422e-10

#From the output, the p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution. In other words, we can assume the normality
ggpairs(TreeB)#Make a matrix of plots with data set

```



*#The `ggcorr()` function draws a correlation matrix plot using `ggplot2`.*  
`ggcorr(TreeB, palette = "RdBu", label = TRUE)`



`glimpse`: This makes to see every column in a data frame in a very detailed way.

`df status`: returns the quantity and percentage of zeros for each vector (q zeros and p zeros). Same NA quality metrics (q NA / p na), and limitless quality (q inf / p inf). The last two columns display the form and quantity of unique values in the results. Print this function, and return the results.

`Describe`: This method specifies if the variable is a variable, a dimension, a type, a binary, a discrete number, and a continuous number, and sets out a descriptive statistical description of each.

`Skim`: `skimr` is designed to provide summary statistics about variables.

Now I am performing visualisation of data by using some visulaisation packages to know the data in a better way

*##`Introduce()` function will give the outline of the data it tells about the number of rows,columns,missing values,discrete columns,continous columns,all missing columns,complete rows,total observations and memory usage*  
`introduce(TreeB)`

```
## rows columns discrete_columns continuous_columns all_missing_columns
## 1 60 2 0 2 0
```

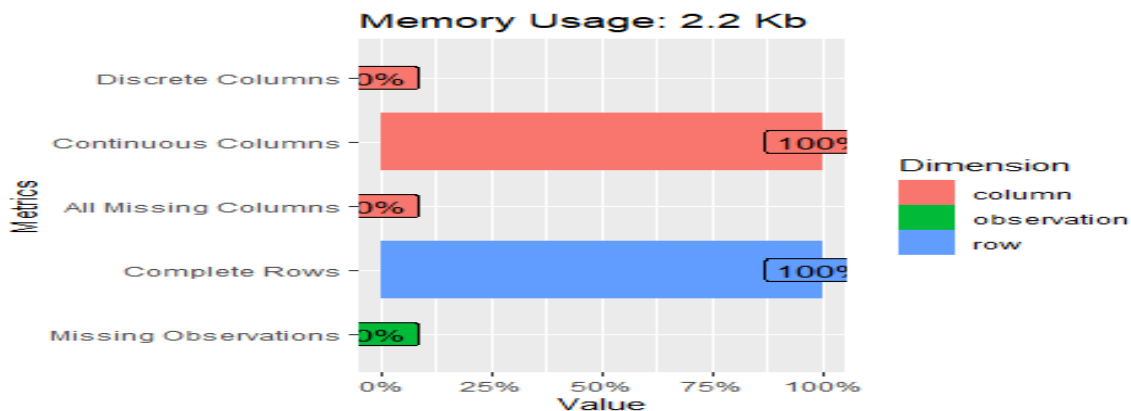
```
## total_missing_values complete_rows total_observations memory_usage
## 1 0 60 120 2296
```

*#It prints the dataset*

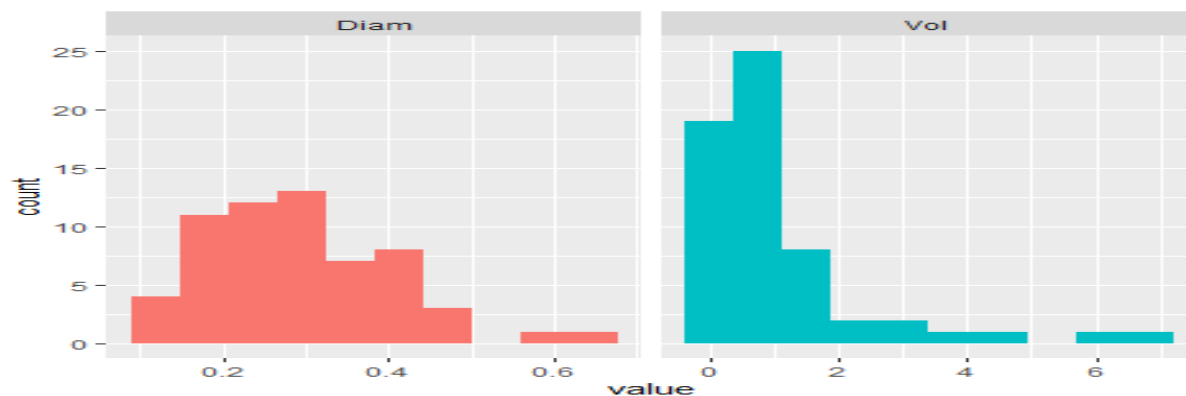
```
plot_str(TreeB)
```

*#plot\_intro() will plot a graph which will give the percentage of discrete columns, continuous columns, All missing columns, complete rows and missing observations*

```
plot_intro(TreeB)
```



```
plot_num(TreeB)
```

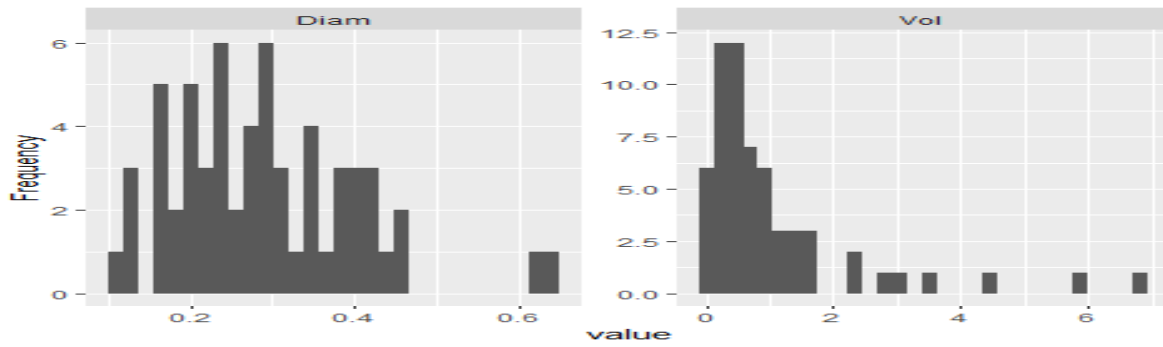


```
profiling_num(TreeB)
```

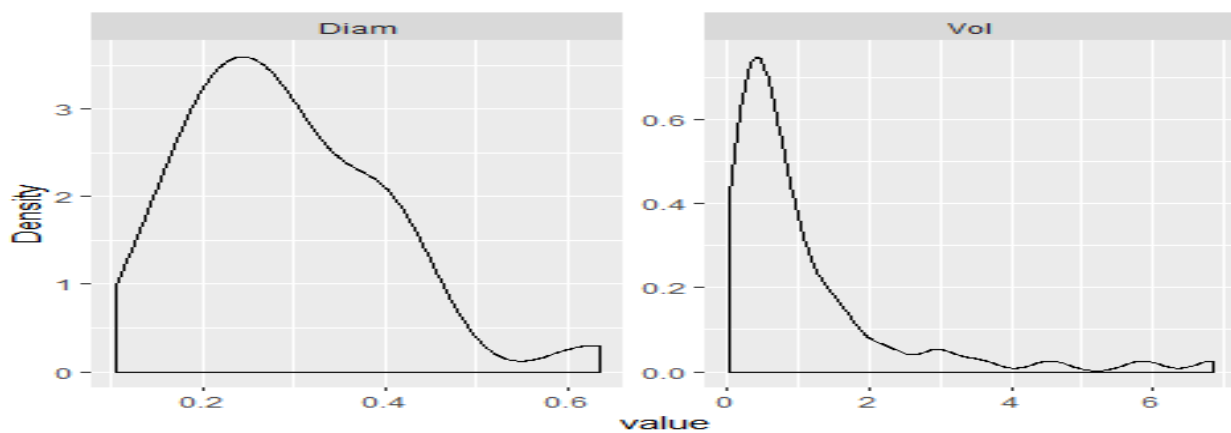
```
## variable      mean  std_dev variation_coef    p_01    p_05    p_25
## 1    Diam 0.2895333 0.1109486    0.3831979 0.113847 0.135005 0.206725
## 2    Vol 1.0611883 1.3349546    1.2579808 0.035470 0.087500 0.330875
##      p_50    p_75    p_95    p_99 skewness kurtosis    iqr
## 1 0.27495 0.358850 0.448785 0.623918 0.8532442 3.916055 0.152125
## 2 0.59635 1.208425 3.598110 6.263330 2.6212871 10.196920 0.877550
##      range_98      range_80
## 1 [0.113847, 0.623918] [0.16414, 0.41556]
## 2 [0.03547, 6.26333] [0.16364, 2.30741]
```

*#prints histogram for numerical data*

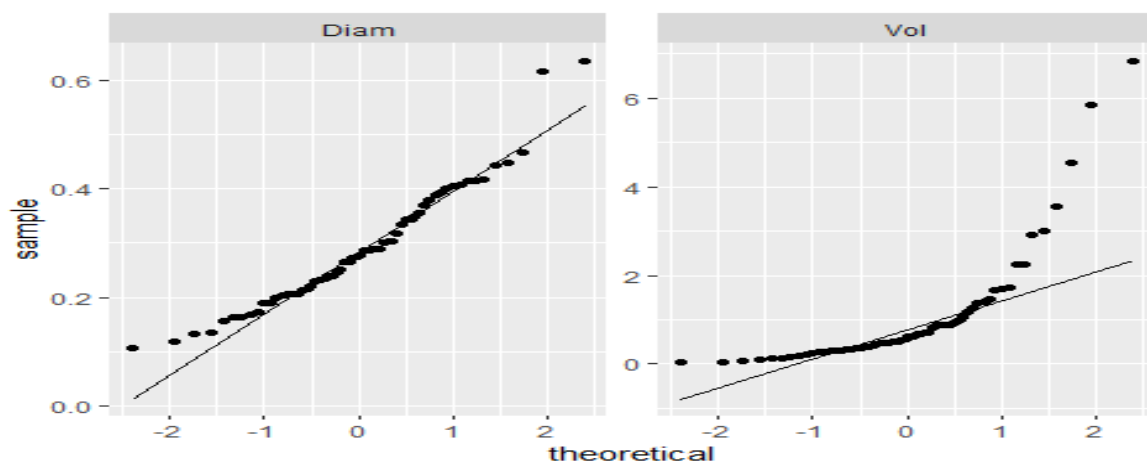
```
plot_histogram(TreeB)
```



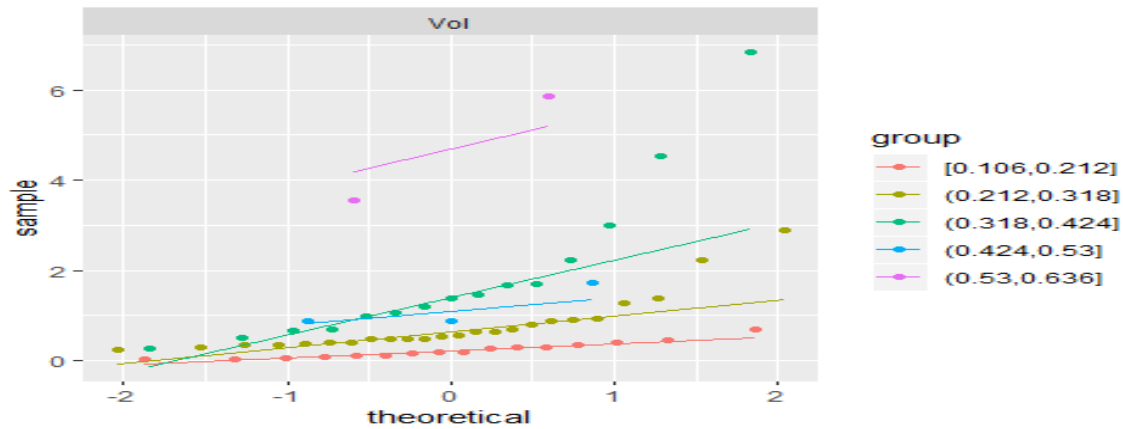
```
#prints density plot
plot_density(TreeB)
```



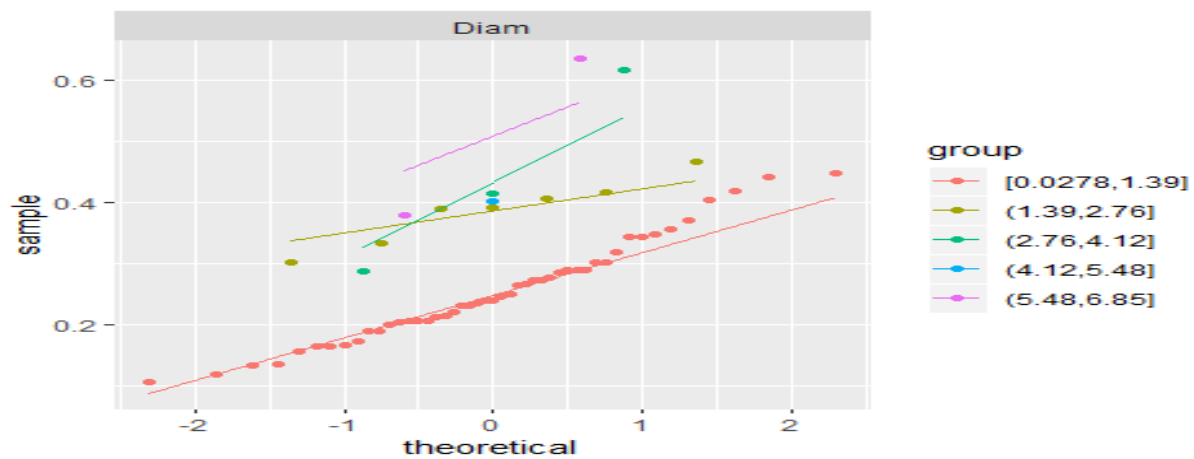
```
#Q-Q plot
qq_data <- TreeB
plot_qq(qq_data) #plotting qq plot for qq_data
```



```
plot_qq(qq_data, by = "Diam")
```



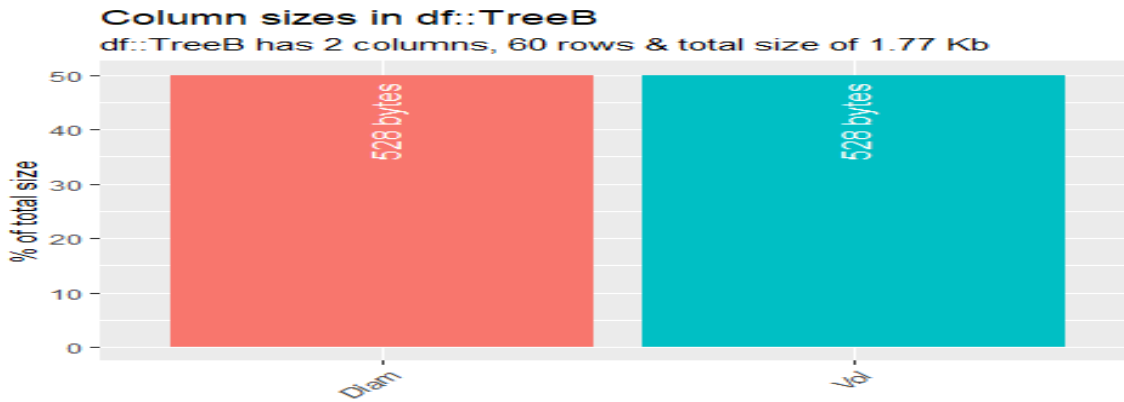
```
plot_qq(qq_data, by = "Vol")
```



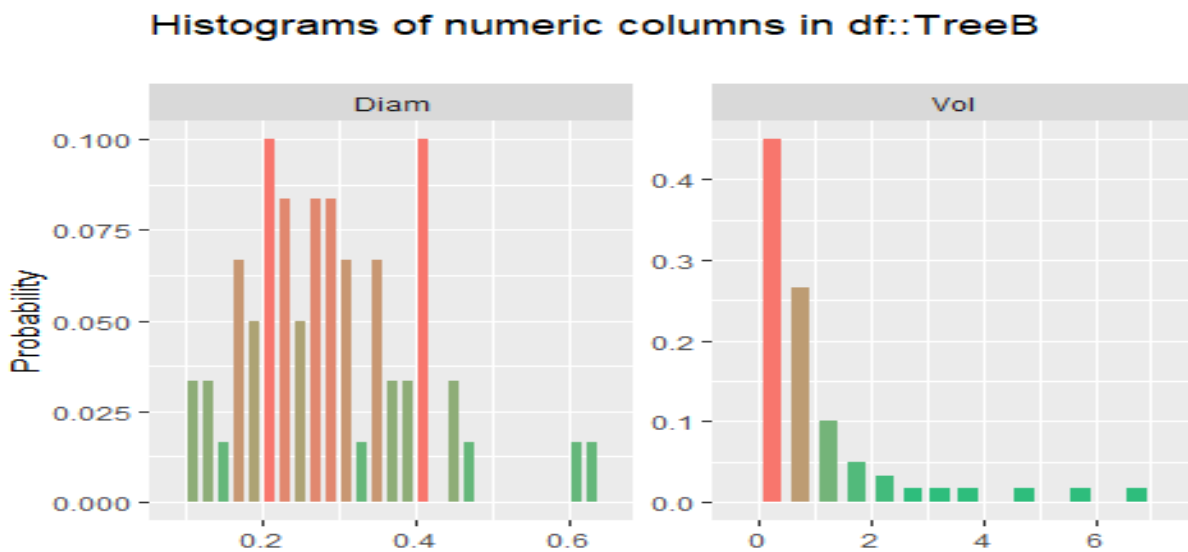
```
#inspect the types of data
a<-inspect_types(TreeB)
show_plot(a)
```



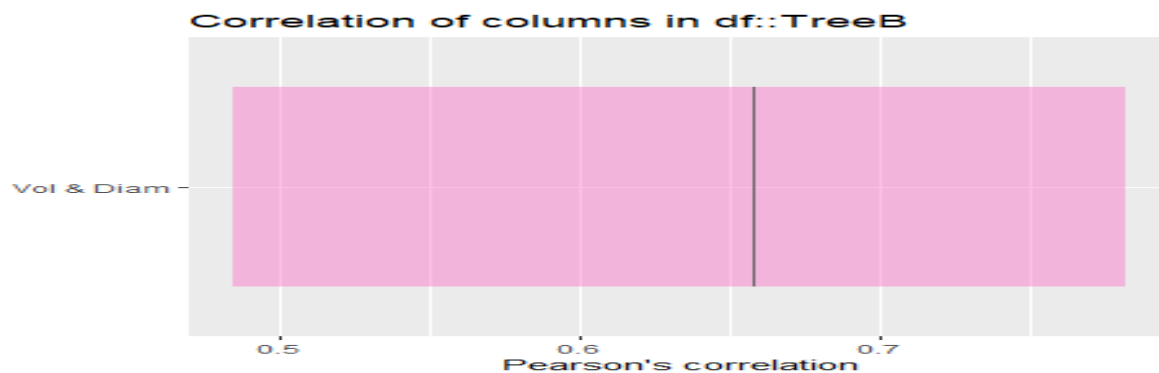
```
#prints the data in the form of graphs and prints column size
b<-inspect_mem(TreeB)
show_plot(b)
```



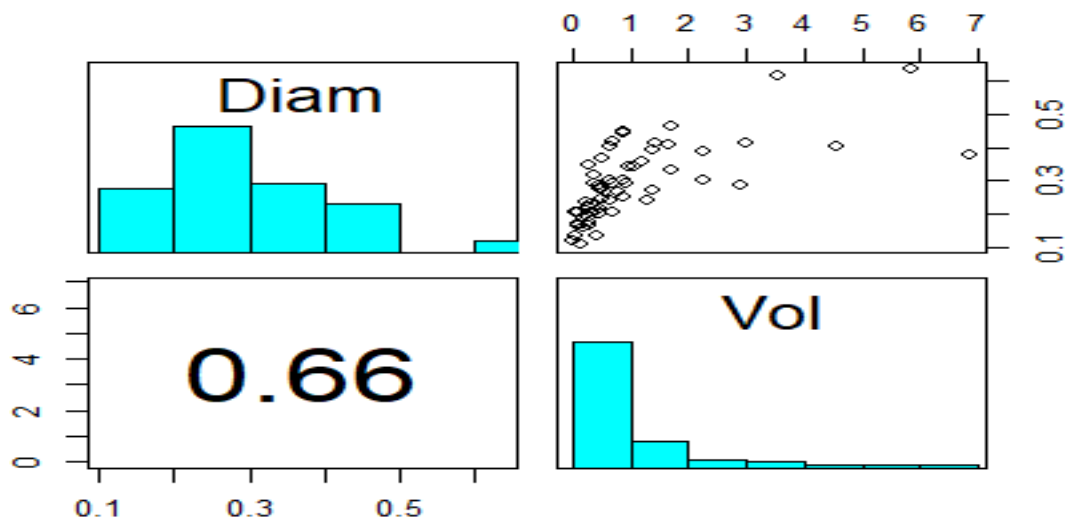
```
#inspects the numerical data
b2<-inspect_num(TreeB)
show_plot(b2)
```



```
#pearson's correlation
b5<-inspect_cor(TreeB)
show_plot(b5)
```



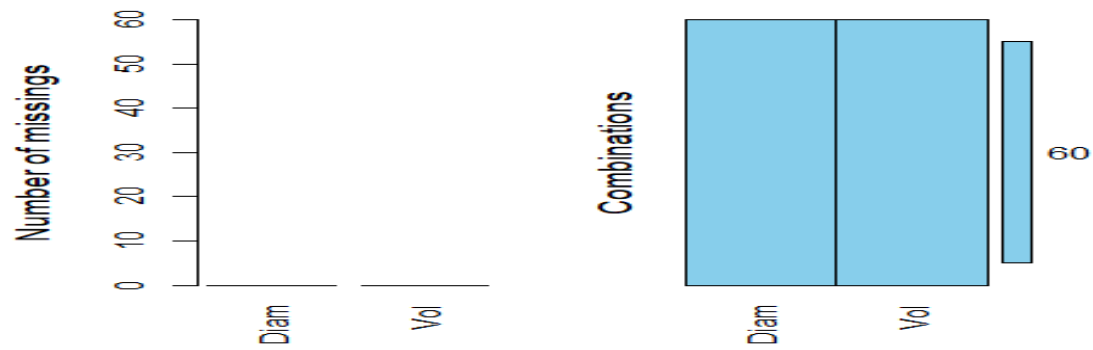
```
## Pairs plot
##function to put histograms on the diagonal
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y, use = "everything"))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste0(prefix, txt)
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}
# this function is taken from the help documentation, it will create
#histograms along the diagonal
panel.hist <- function(x, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5) )
  h <- hist(x, plot = FALSE)
  breaks <- h$breaks; nB <- length(breaks)
  y <- h$counts; y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}
# plot scatter plots for variables 1:7
pairs(TreeB, lower.panel=panel.cor, diag.panel = panel.hist)
```



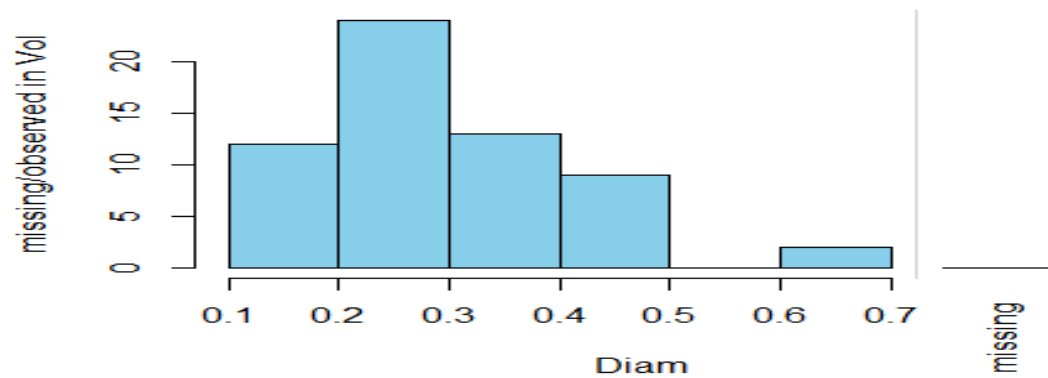
(ii)Cleaning your dataset Looking in to the missing data

```
#shows the amount of missing data for each variable and the frequency of combinations of missing values
aggr(TreeB, prop = FALSE, number=TRUE)
```

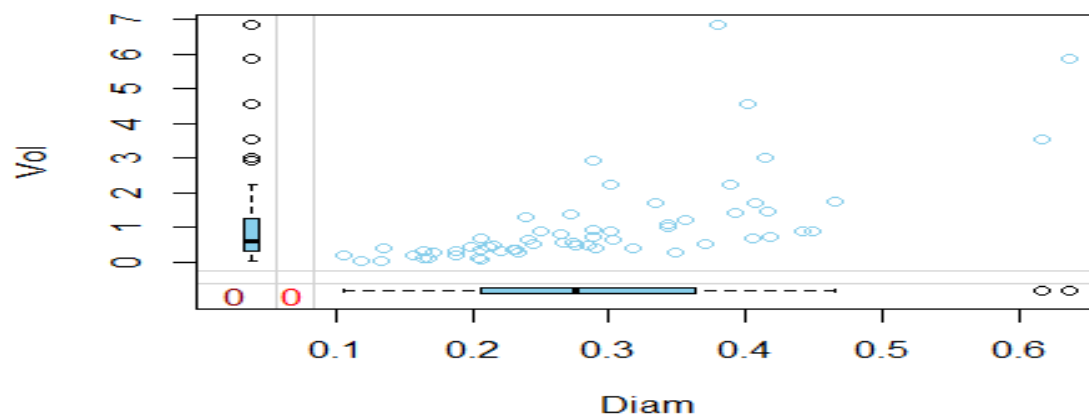




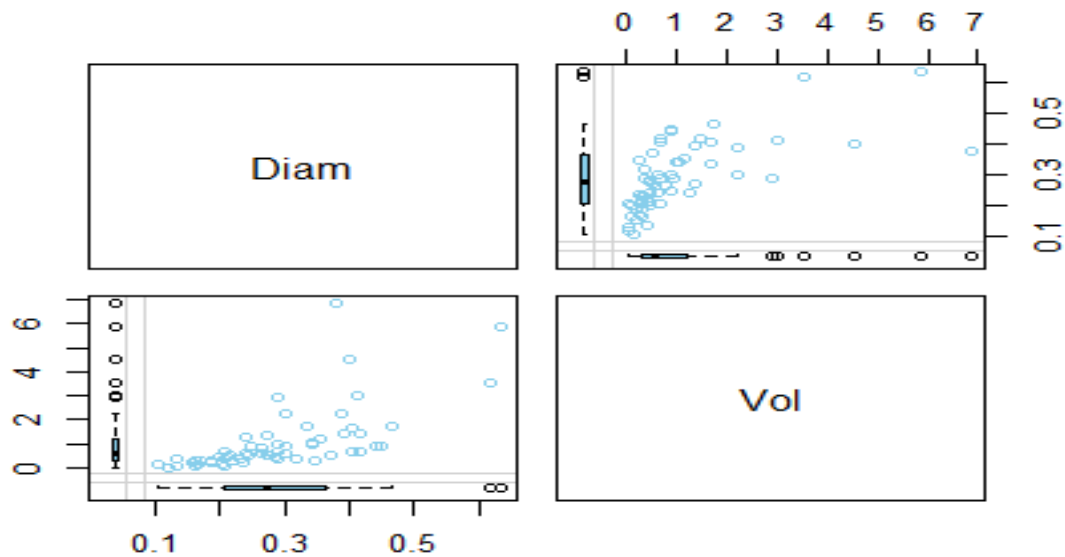
*#creates a barchart showing the values of the variable missing values*  
**barMiss**(TreeB)



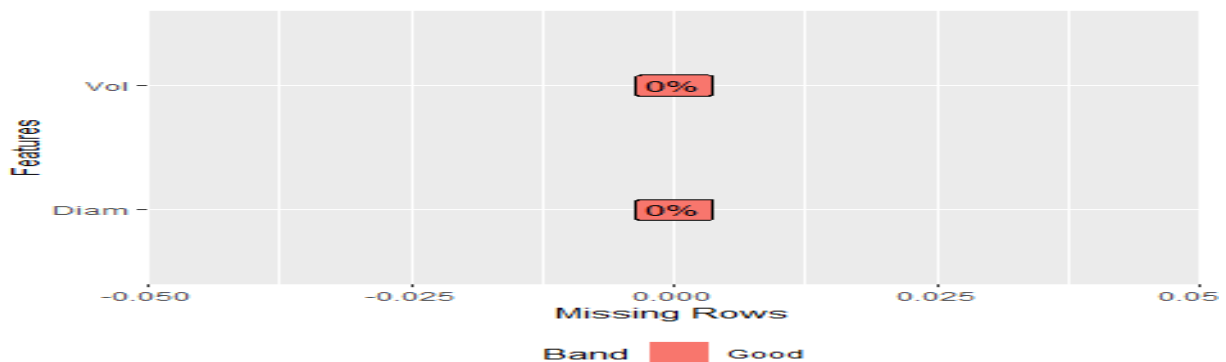
*# Creates a scatterplot between two variables with information about missing values in the margins*  
**marginplot**(TreeB)



```
#Next we create a margin plot
marginmatrix(TreeB)
```



```
#plot_missing() function shows the percentage of missing values of each column present in the dataset
plot_missing(TreeB)
```



```
profile_missing(TreeB)
```

```
## feature num_missing pct_missing
## 1 Diam 0 0
## 2 Vol 0 0
```

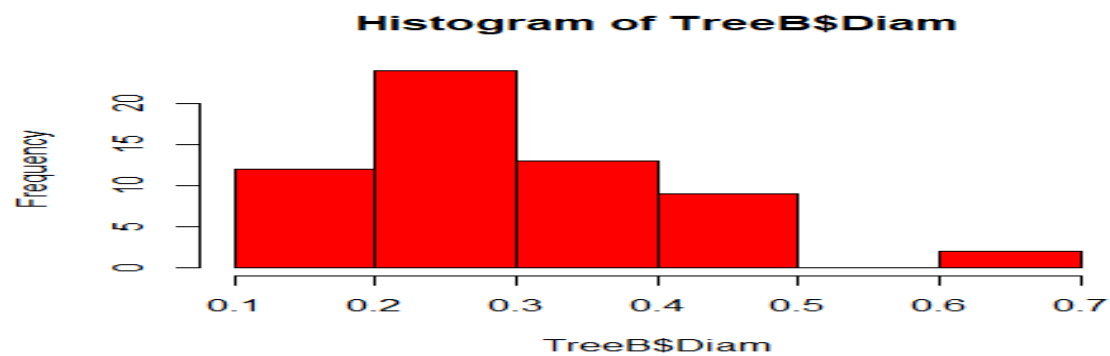
THERE IS NO MISSING DATA so there is no need to clean the data

(iii)Analyzing relationships between variables(graphical data Analysis)

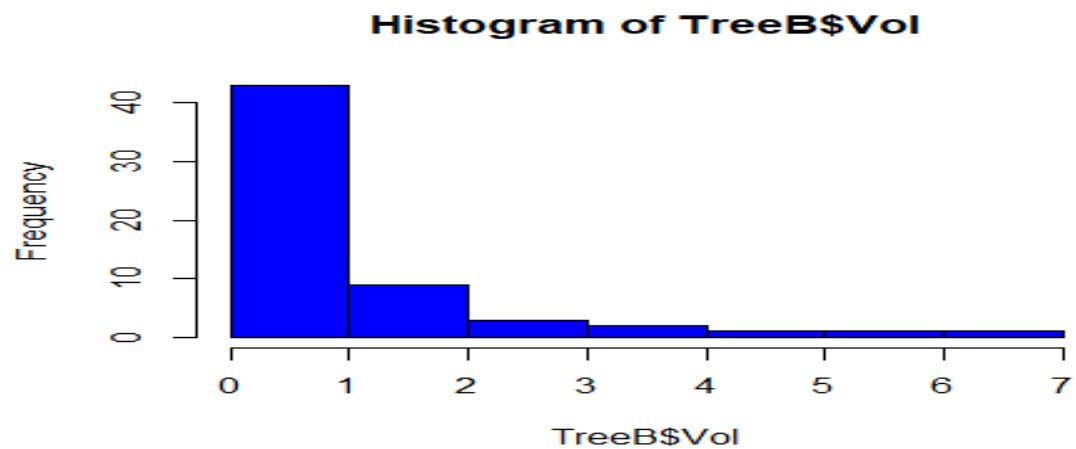
```
#GRAPHICAL SUMMARY
```

```
#Include: boxplots, histograms, scatterplot and the correlation coefficient.
```

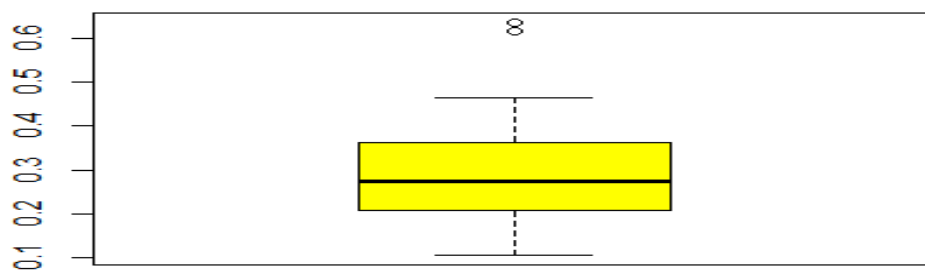
```
#histograms  
hist(TreeB$Diam,col="red")
```



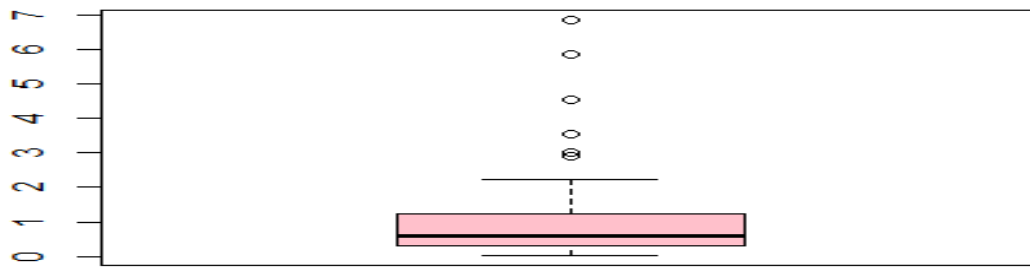
```
hist(TreeB$Vol,col="blue")
```



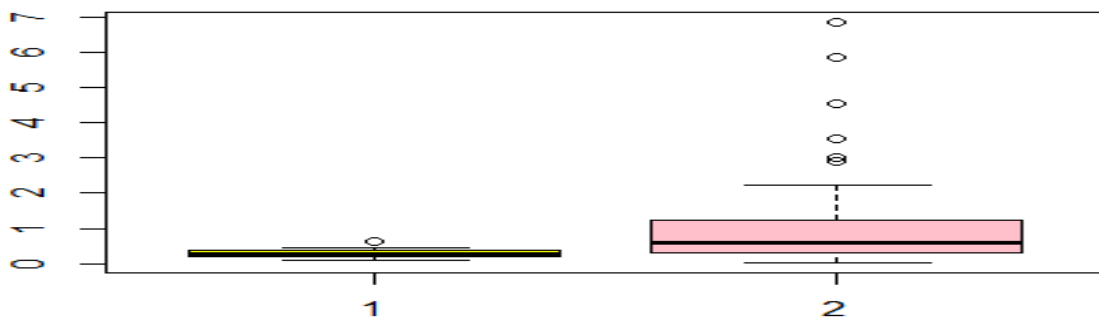
```
#boxplots  
boxplot(TreeB$Diam,col="yellow")
```



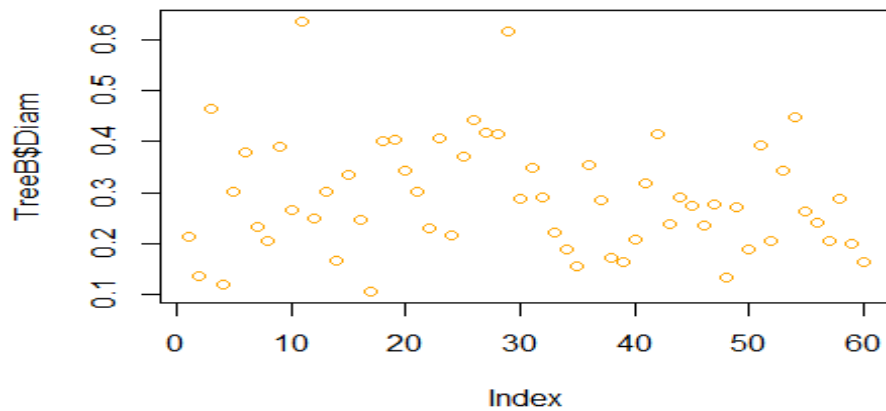
```
boxplot(TreeB$Vol,col="pink")
```



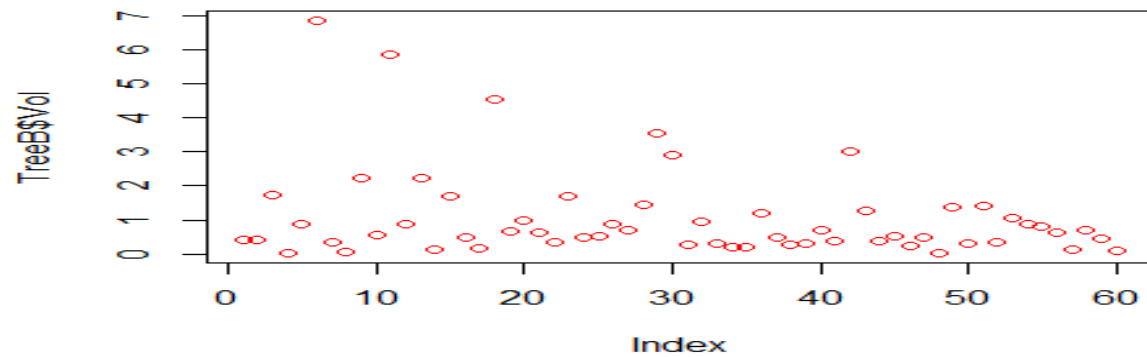
```
boxplot(TreeB$Diam, TreeB$Vol, col=c("yellow", "pink"))
```



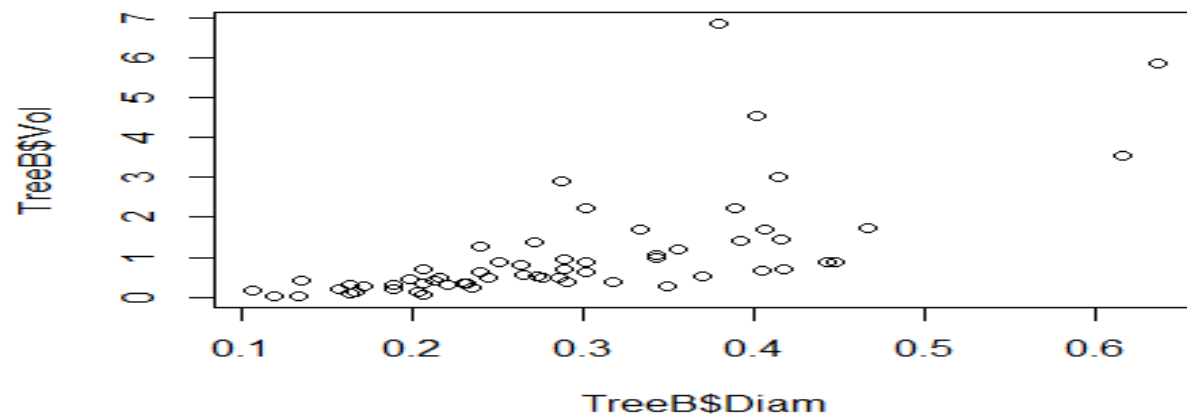
```
#scatterplots
plot(TreeB$Diam, col="orange")
```



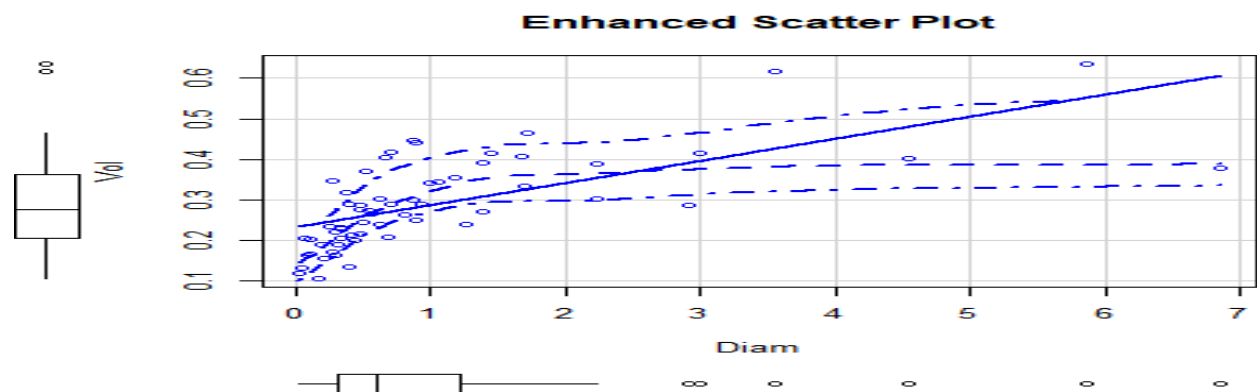
```
plot(TreeB$Vol,col="red")
```



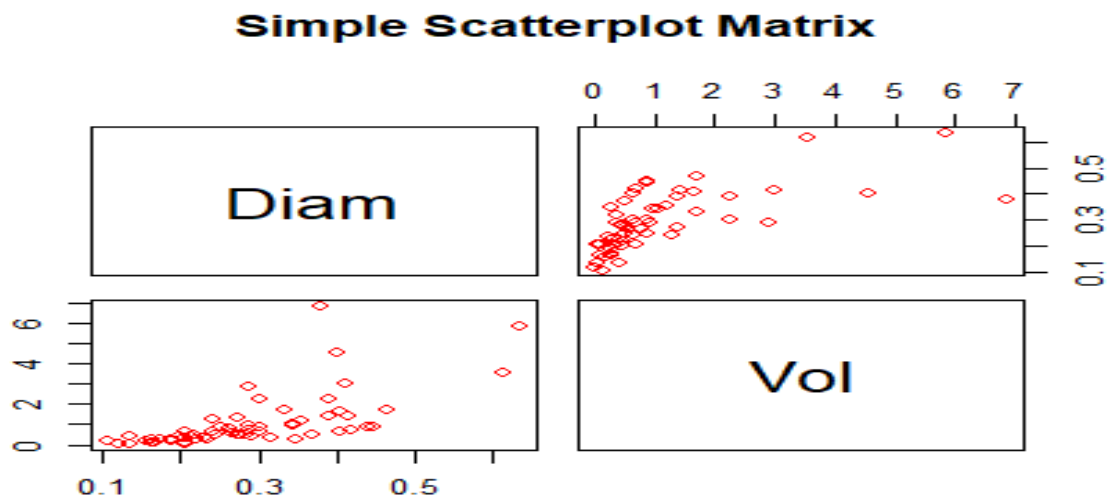
```
plot(TreeB$Diam,TreeB$Vol)
```



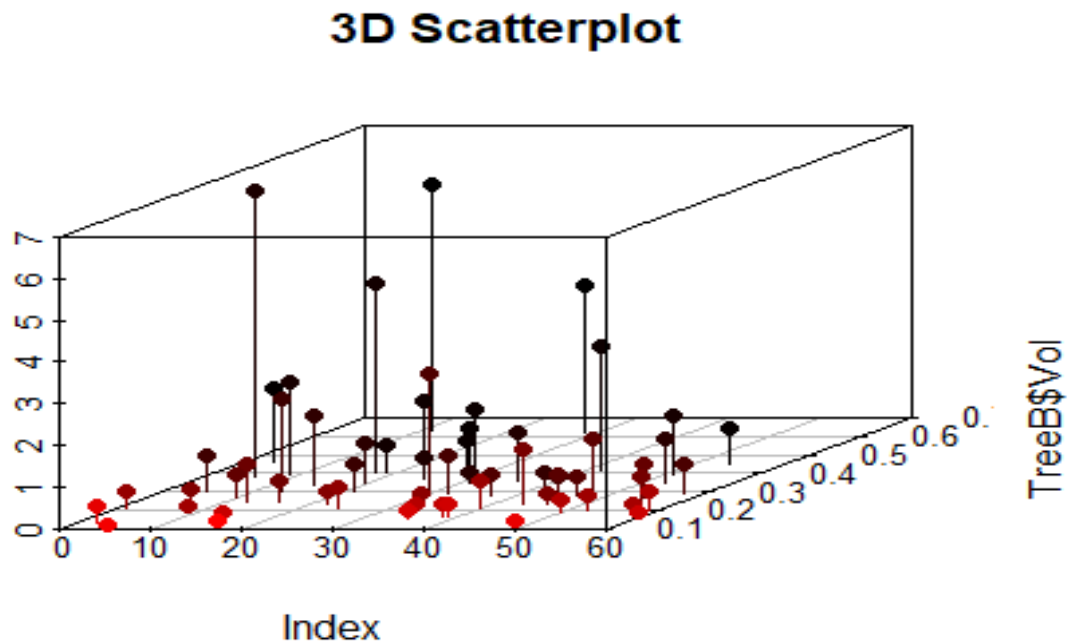
```
scatterplot(Diam~Vol, data=TreeB,
            xlab="Diam", ylab="Vol",
            main="Enhanced Scatter Plot")
```



```
pairs(~Diam+Vol,data=TreeB,
      main="Simple Scatterplot Matrix",col="red")
```

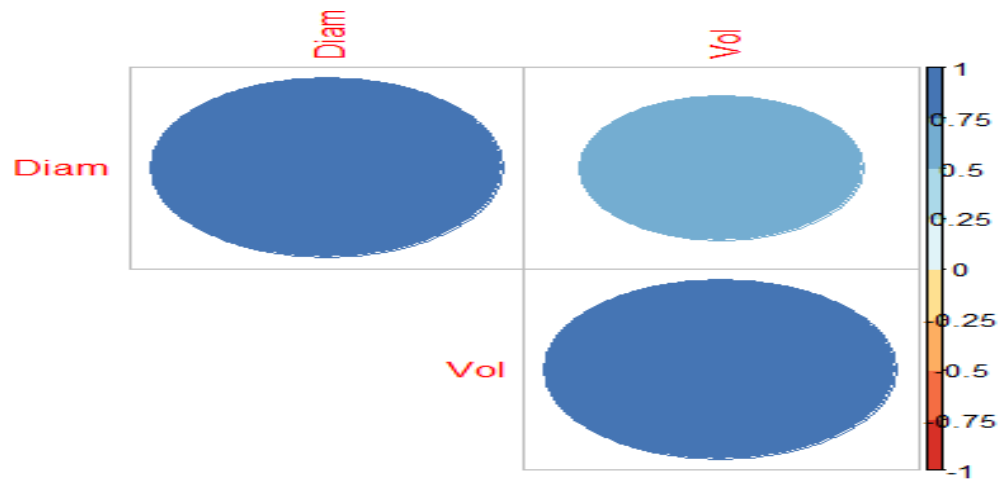


```
scatterplot3d(TreeB$Diam,TreeB$Vol, pch=16, highlight.3d=TRUE,
               type="h", main="3D Scatterplot")
```



```
#correlation coefficient
M <- cor(TreeB)
```

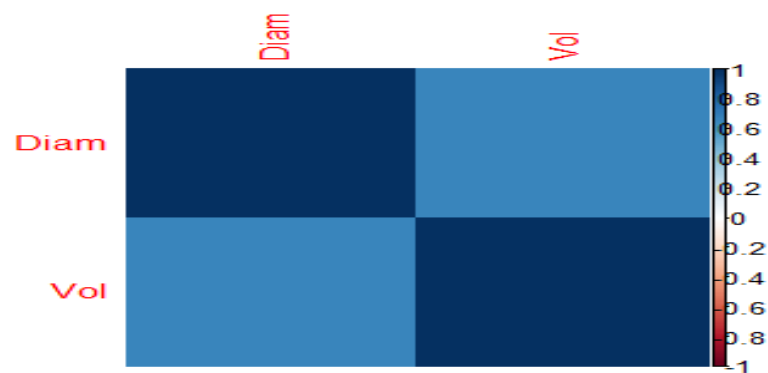
```
corrplot(M, type="upper", order="hclust",
         col=brewer.pal(n=8, name="RdYlBu"))
```



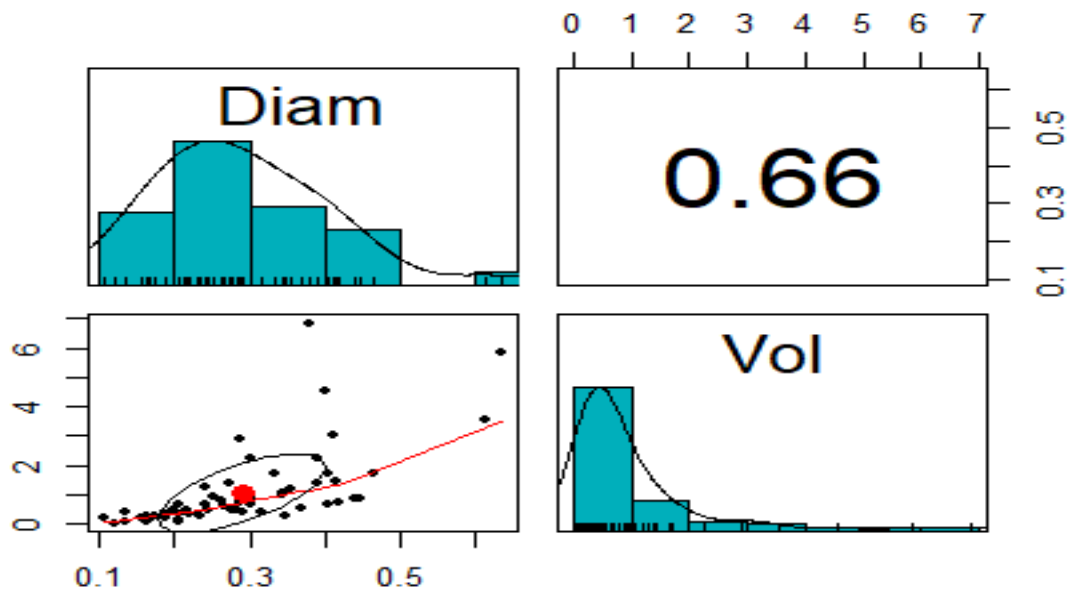
```
corrplot(M, method="number")
```



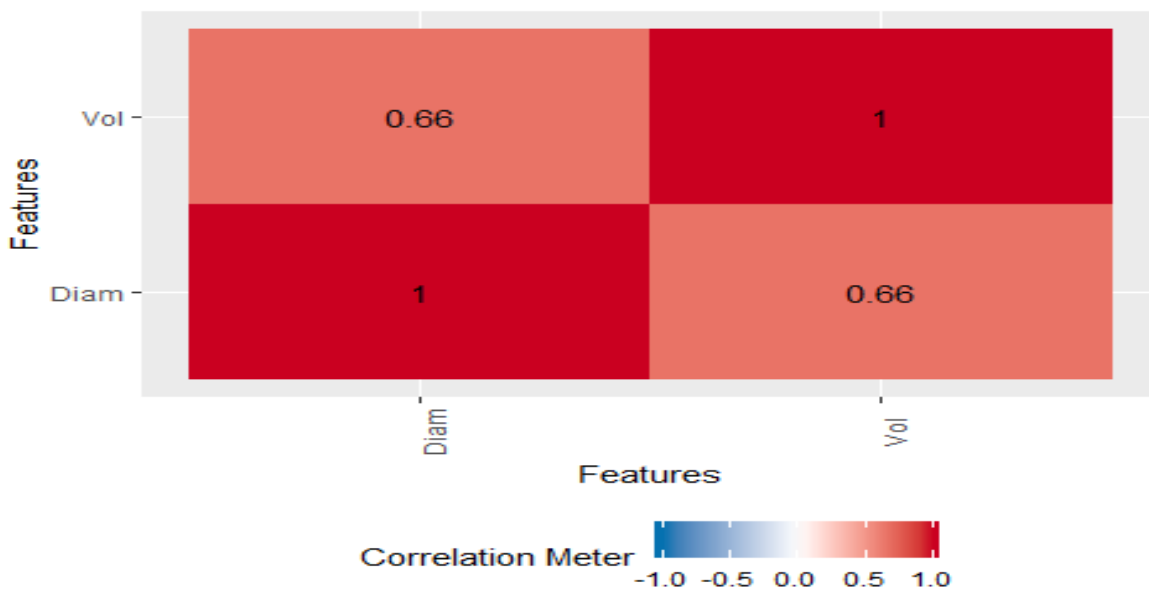
```
corrplot(M, method="color")
```



```
pairs.panels(TreeB,
  method = "pearson", # correlation method
  hist.col = "#00AFBB",
  density = TRUE, # show density plots
  ellipses = TRUE) # show correlation ellipses
```



```
plot_correlation(TreeB)
```



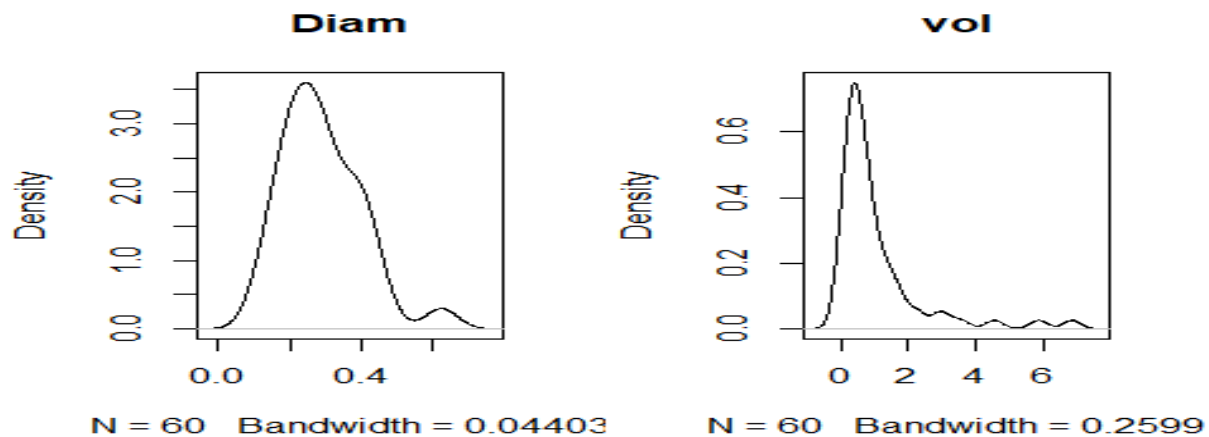
From the density graph I observed that the variables are highly skewed. To correct the skewness we can use log10 transformations



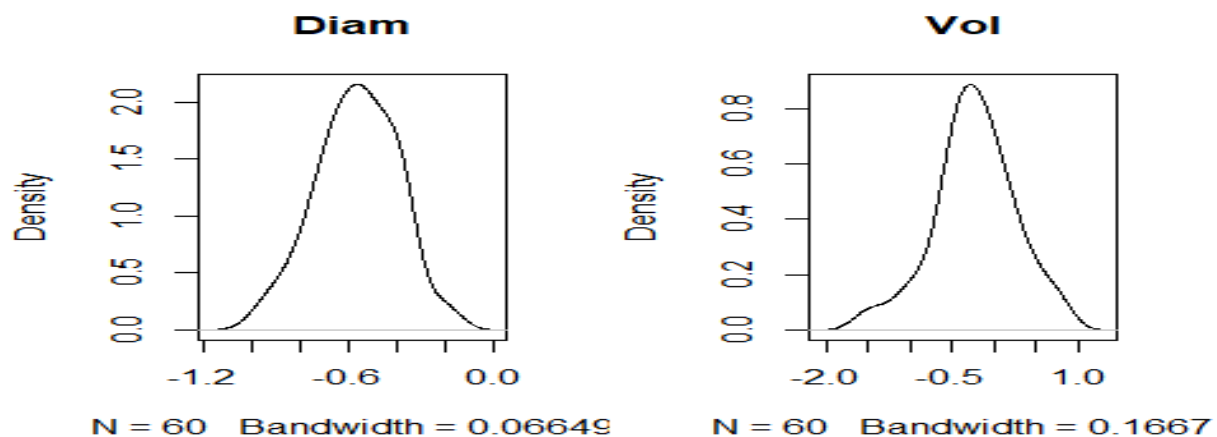
log Transformations - The log transformation can be used to make highly skewed distributions less skewed. This can be valuable both for making patterns in the data more interpretable and for helping to meet the assumptions of inferential statistics.

And from boxplots we can see that there are some outliers. Those outliers will be reduced after applying log transformation because we are reducing highly skewed data to less skewed data

```
par(mfrow=c(1,2))  
plot(density(TreeB$Diam),main="Diam")  
plot(density(TreeB$Vol),main="vol")
```



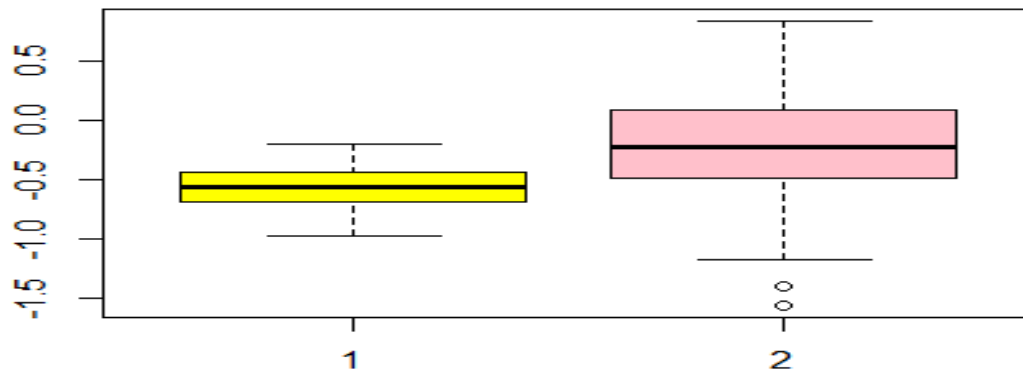
```
# we can apply a log transformation to correct the skew  
TreeB$Diam<-log10(TreeB$Diam)  
TreeB$Vol<-log10(TreeB$Vol)  
# check the distributions after transformation  
par(mfrow=c(1,2))  
plot(density(TreeB$Diam),main="Diam")  
plot(density(TreeB$Vol),main="Vol")
```



Now the highly skewed distribution is less skewed.

After applying log transformation also still there are some outliers present in the Vol, we can see those from the boxplot.

```
boxplot(TreeB$Diam,TreeB$Vol,col=c("yellow","pink"))
```



so I will be performing the grubbs.test to see whether the outliers are significant or not

```
grubbs.test(TreeB$Vol)
```

```
##
##  Grubbs test for one outlier
##
## data:  TreeB$Vol
## G = 2.71570, U = 0.87288, p-value = 0.1551
## alternative hypothesis: lowest value -1.55595520408192 is an outlier
#its is a outlier which is not significant because the p value is greater than 0.05.
```

I performed the grubbs.test for the vol variable the outliers are significant because the p value is greater than 0.05

The numerical and graphical summary of data after applying log transformation

```
#Numerical summary
```

```
df_status(TreeB)
```

```
##  variable q_zeros p_zeros q_na p_na q_inf p_inf    type unique
## 1    Diam         0         0    0    0    0    0 numeric     59
## 2     Vol         0         0    0    0    0    0 numeric     60
```

```
describe(TreeB)
```

```
##      vars  n  mean    sd median trimmed  mad   min   max range  skew kurtos
is
```

```
## Diam      1 60 -0.57 0.17 -0.56 -0.56 0.18 -0.97 -0.20 0.78 -0.17 -0.34
## Vol       2 60 -0.23 0.49 -0.23 -0.21 0.43 -1.56 0.84 2.39 -0.25 0.23
##          se
## Diam 0.02
## Vol  0.06
```

skim(TreeB)

Data summary



Name	TreeB
Number of rows	60
Number of columns	2

Column type frequency:

numeric	2
---------	---

Group variables	None
-----------------	------

Variable type: numeric

skim_variab	n_missi	complete_ra	mea			p2	p5	p7	p10	hist
le	ng	te	n	sd	p0	5	0	5	0	
Diam	0	1	-0.57	0.17	-0.97	0.6	0.5	0.4	0.20	
Vol	0	1	-0.23	0.49	-1.56	0.4	0.2	0.84		

mean(TreeB\$Diam)

```
## [1] -0.5693113
```

mean(TreeB\$Vol)

```
## [1] -0.2256334
```

median(TreeB\$Diam)

```
## [1] -0.5607508
```

median(TreeB\$Vol)

```
## [1] -0.2252588
```

```
sd(TreeB$Diam)
## [1] 0.1675455

sd(TreeB$Vol)
## [1] 0.4898627

var(TreeB$Diam)
## [1] 0.02807149

var(TreeB$Vol)
## [1] 0.2399655

IQR(TreeB$Diam)
## [1] 0.2394582

IQR(TreeB$Vol)
## [1] 0.5627327

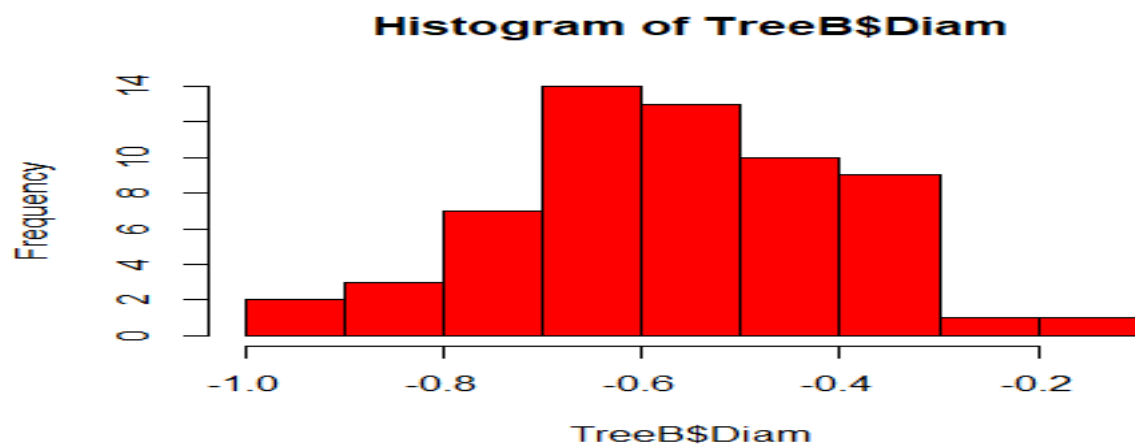
#NORMALITY
shapiro.test(TreeB$Diam)

##
##  Shapiro-Wilk normality test
##
## data:  TreeB$Diam
## W = 0.98879, p-value = 0.8571

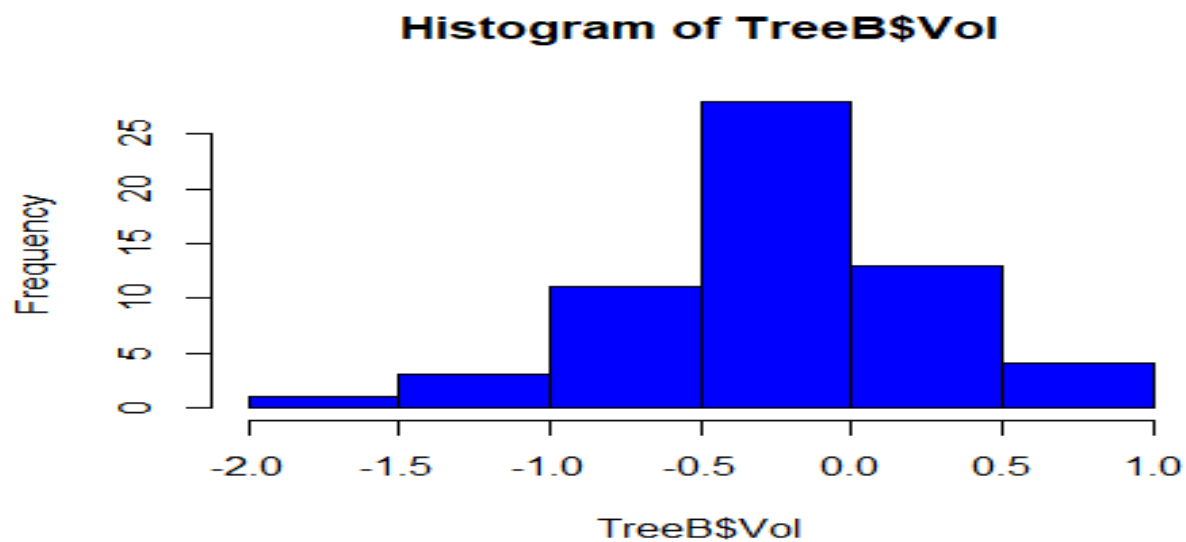
shapiro.test(TreeB$Vol)

##
##  Shapiro-Wilk normality test
##
## data:  TreeB$Vol
## W = 0.98581, p-value = 0.7122

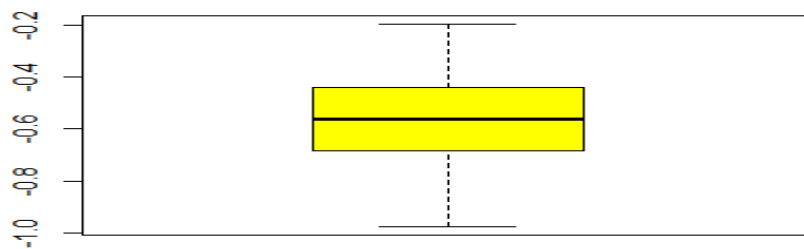
#GRAPHICAL SUMMARY
#Include: boxplots, histograms, scatterplot and the correlation coefficient.
#histograms
hist(TreeB$Diam,col="red")
```



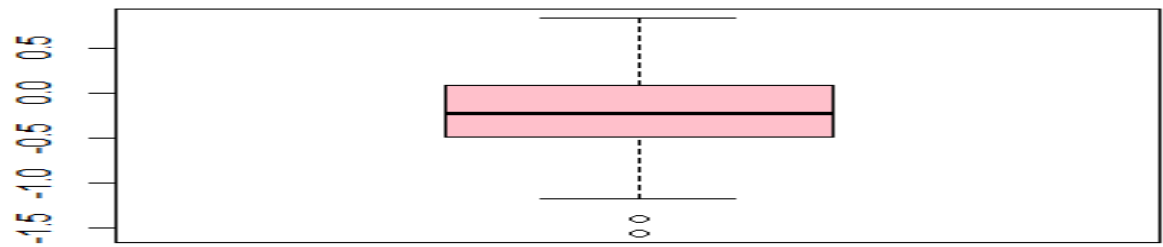
```
hist(TreeB$Vol,col="blue")
```



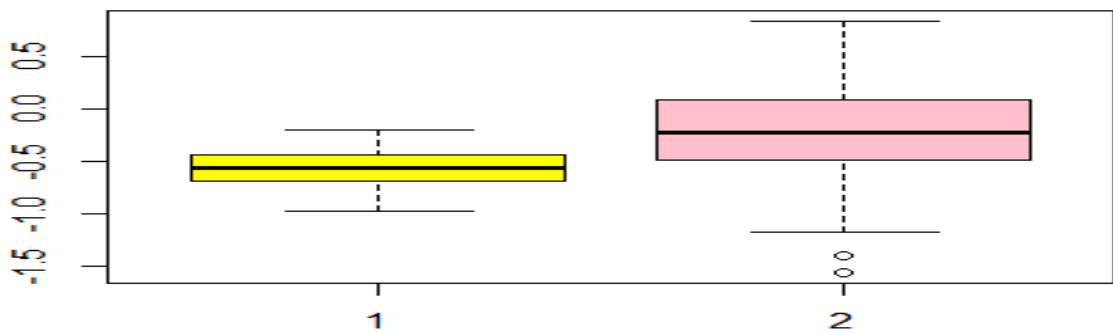
```
#boxplots  
boxplot(TreeB$Diam,col="yellow")
```



```
boxplot(TreeB$Vol,col="pink")
```

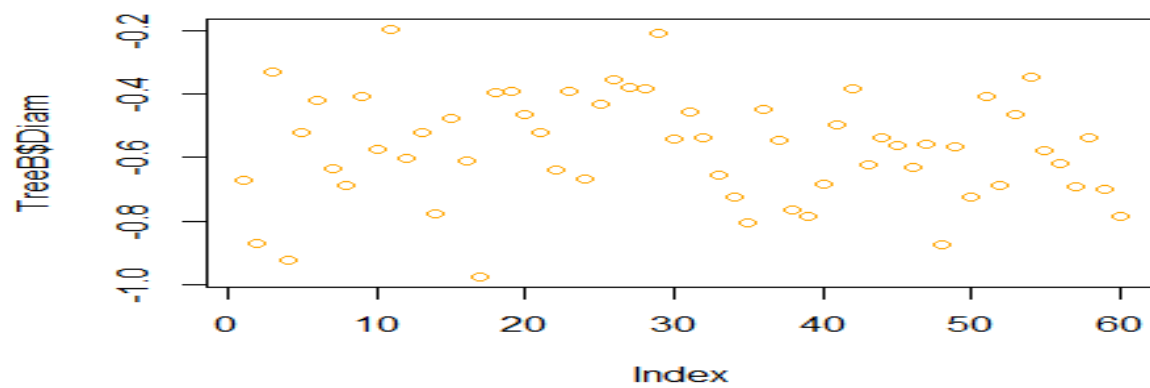


```
boxplot(TreeB$Diam,TreeB$Vol,col=c("yellow","pink"))
```

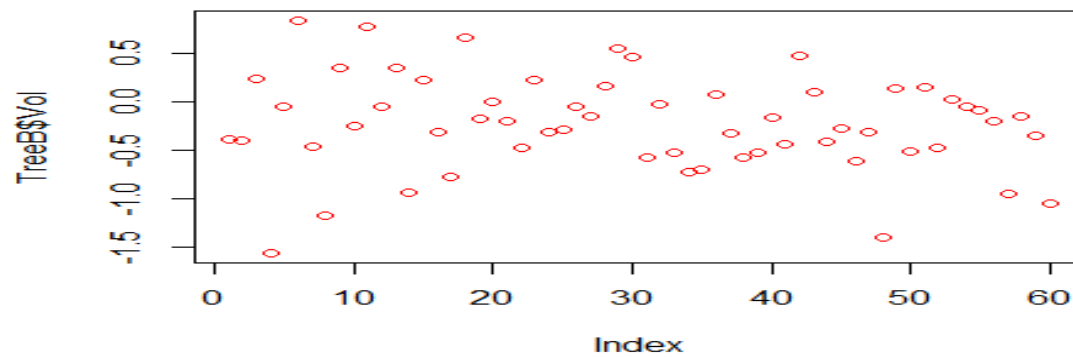


```
#scatterplots
```

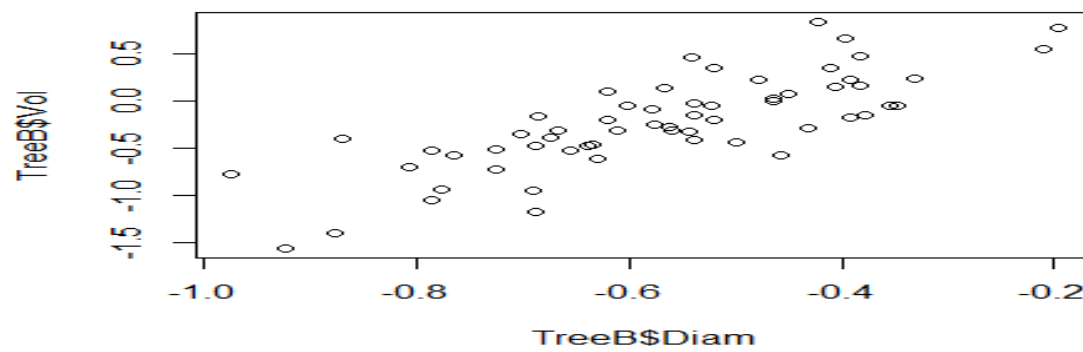
```
plot(TreeB$Diam,col="orange")
```



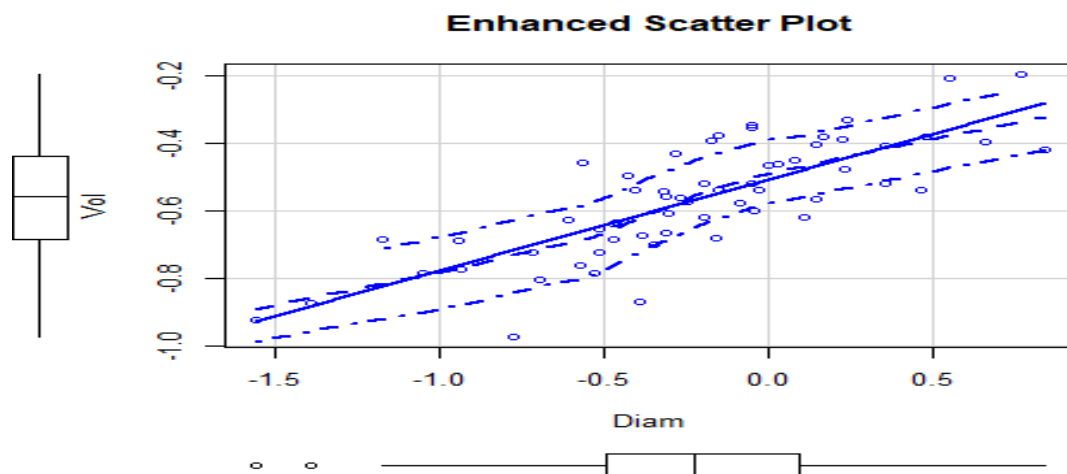
```
plot(TreeB$Vol,col="red")
```



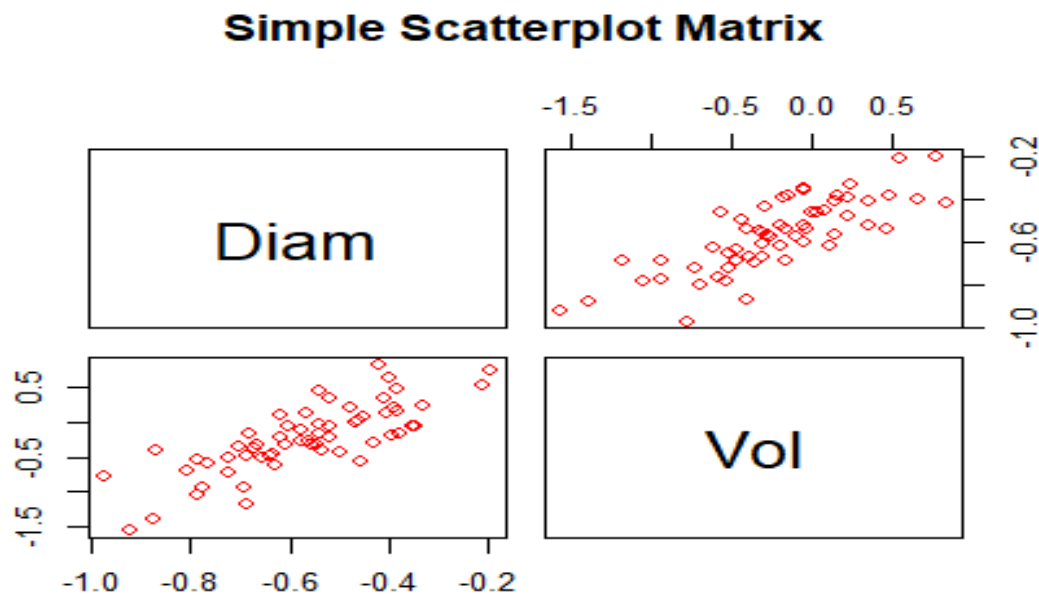
```
plot(TreeB$Diam,TreeB$Vol)
```



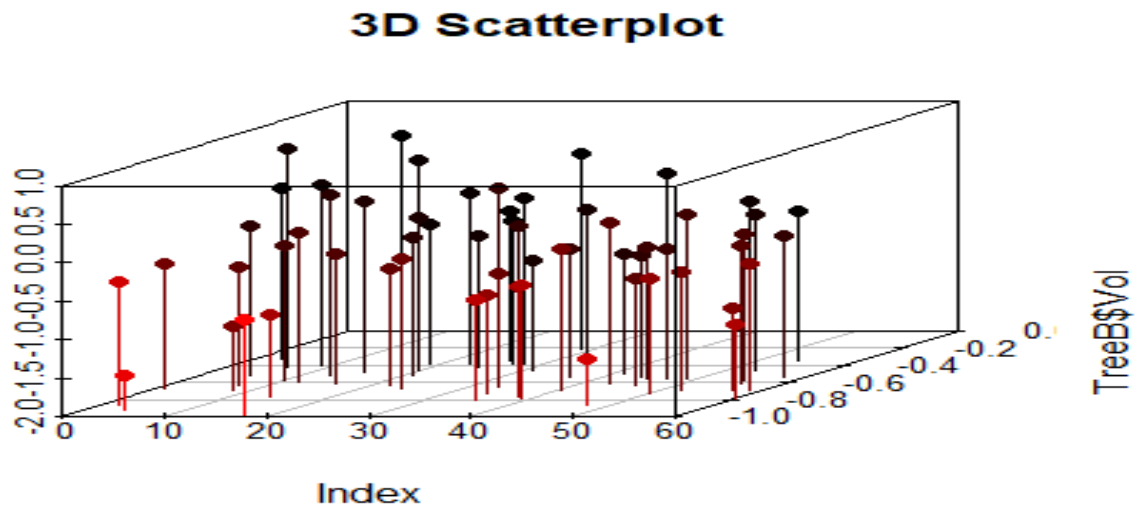
```
scatterplot(Diam~Vol, data=TreeB,  
            xlab="Diam", ylab="Vol",  
            main="Enhanced Scatter Plot")
```



```
pairs(~Diam+Vol,data=TreeB,
      main="Simple Scatterplot Matrix",col="red")
```

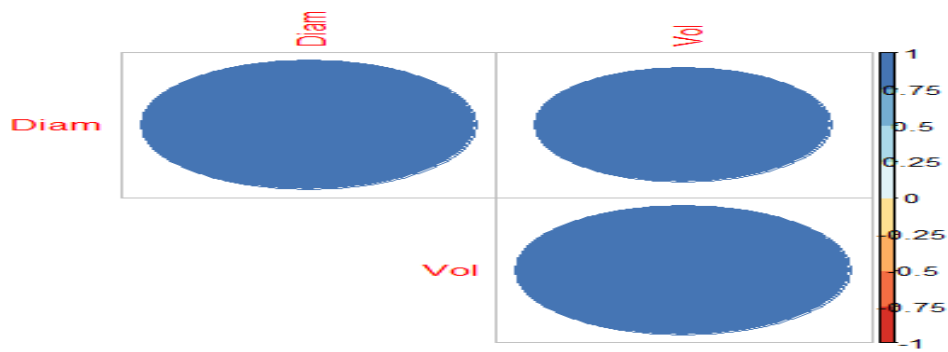


```
scatterplot3d(TreeB$Diam,TreeB$Vol, pch=16, highlight.3d=TRUE,
               type="h", main="3D Scatterplot")
```



```
#corelation coefficient
M <-cor(TreeB)
corrplot(M, type="upper", order="hclust",
          col=brewer.pal(n=8, name="RdYlBu"))
```

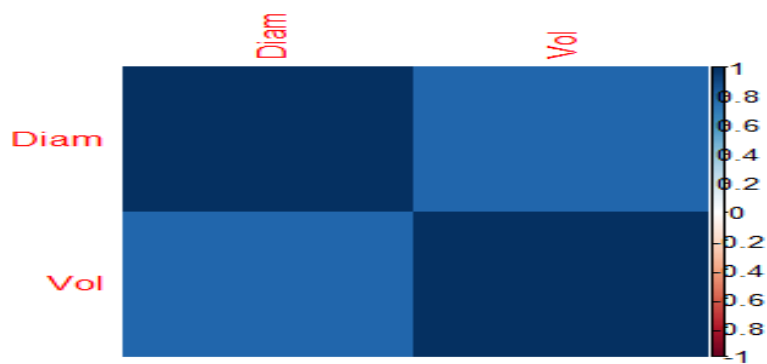




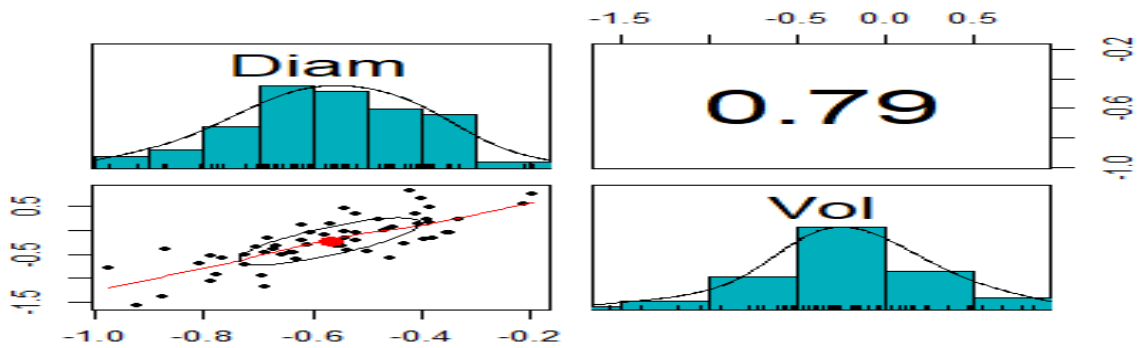
```
corrplot(M, method="number")
```



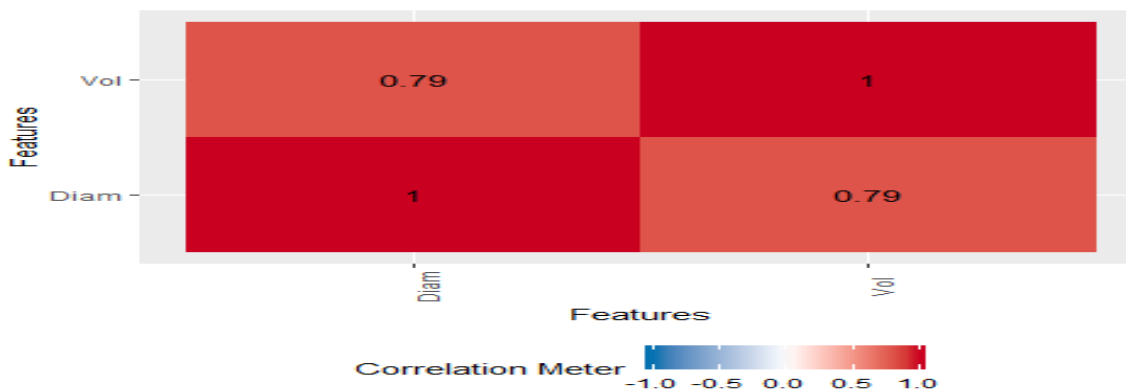
```
corrplot(M, method="color")
```



```
pairs.panels(TreeB,
  method = "pearson", # correlation method
  hist.col = "#00AFBB",
  density = TRUE, # show density plots
  ellipses = TRUE) # show correlation ellipses
```



```
plot_correlation(TreeB)
```



From the plot I observed that the both variables diam and vol both are positive skew before applying log transformations .After that both are symmetrical Distribution.

Before the log transformation the numerical summary of data is as follows: The mean of Diam=0.2895333 and vol=1.061188 The median of Diam=0.27495 and vol=0.59635 The standard deviation of Diam=0.1109486 and vol=1.334955 The variance of Diam=0.01230958 and vol=1.782104 The IQR of Diam=0.152125 and vol=0.87755 The Normality for Diam,From the output, the p-value < 0.05 implying that the distribution of the data are significantly different from normal distribution. The Normality for Vol,From the output, the p-value < 0.05 implying that the distribution of the data are significantly different from normal distribution.

After the log transformation the numerical summary of data is as follows: The mean of Diam=-0.5693113 and vol=-0.2256334 The median of Diam=-0.5607508 and vol=-0.2252588 The standard deviation of Diam=0.1675455 and vol=0.4898627 The variance of Diam=0.02807149 and vol=0.23996554 The IQR of Diam=0.2394582 and vol=0.5627327 The Normality for Diam,From the output, the p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution. The Normality for Vol,From the output, the p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution.

## Question 1(b)

Fit a model of the form  $y = \beta_0 + \beta_1 x + e$  and interpret the value of  $\beta_1$ . Note that you will need to consider the results from your exploratory data analysis in part (a) to fit a valid model. Fit a model of the form  $y = \beta_0 + \beta_1 x + e$  and interpret the value of  $\beta_1$ . Note that you will need to consider the results from your exploratory data analysis in part (a) to fit a valid model.

## Solution

In this question I was asked to build a linear regression model by considering the results from the exploratory data analysis from part(a) to fit the model.

simple linear regression considers a single regressor or explanatory variable  $x$  and a dependent or response variable  $y$ .

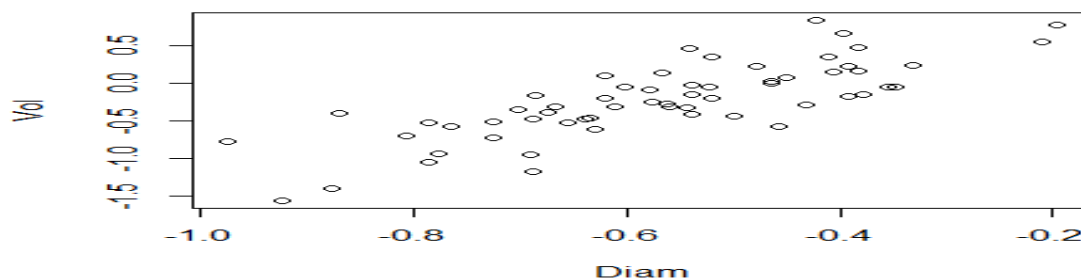
Suppose that the true relationship between  $y$  and  $x$  is a straight line. We assume that each observation,  $y_i$ , can be described by the model:  $y_i = \beta_0 + \beta_1 x_i + e_i$

The regression coefficient  $\beta_0$  represents the intercept of the straight line. The regression coefficient  $\beta_1$  represents the slope of the straight line i.e. the change in  $y$  per unit change in  $x$ . The error term  $e_i$  is a random error with mean zero and (unknown) variance  $\sigma^2$  (square of sigma). The random errors corresponding to different observations are assumed to be independent identically distributed random variables that follow a normal distribution, i.e.  $e \sim N(0, \sigma^2)$ . When fitting a linear model to a set of data, the aim is to calculate the values of  $\beta_0$  and  $\beta_1$  that describe the line of best fit for the data.

Using the function `lm()` we can construct a linear model. We define the model using a formula where the solution variable is divided from the explanatory variables by a `~` on the left-hand side. The data statement is used to say R where the variables used in the formula will be searched for. It is useful to give the model a name so that it is saved as an object. We can then extract information about the linear model from the saved object e.g. the coefficients or residuals.

plot the data and calculate Pearson's correlation between `vol` and `Diam`

```
#Linear regression model  
## Examine the relationship between vol and Diam  
plot(Vol~Diam, data = TreeB)
```



```
cor(TreeB$Vol,TreeB$Diam)
```

```
## [1] 0.791406
```

There appears to be a strong linear relationship between Age and Height. Building the model

```
#fitting the model
```

```
Tree_model<-lm(Vol~Diam, data = TreeB)
```

```
Tree_model
```

```
##
```

```
## Call:
```

```
## lm(formula = Vol ~ Diam, data = TreeB)
```

```
##
```

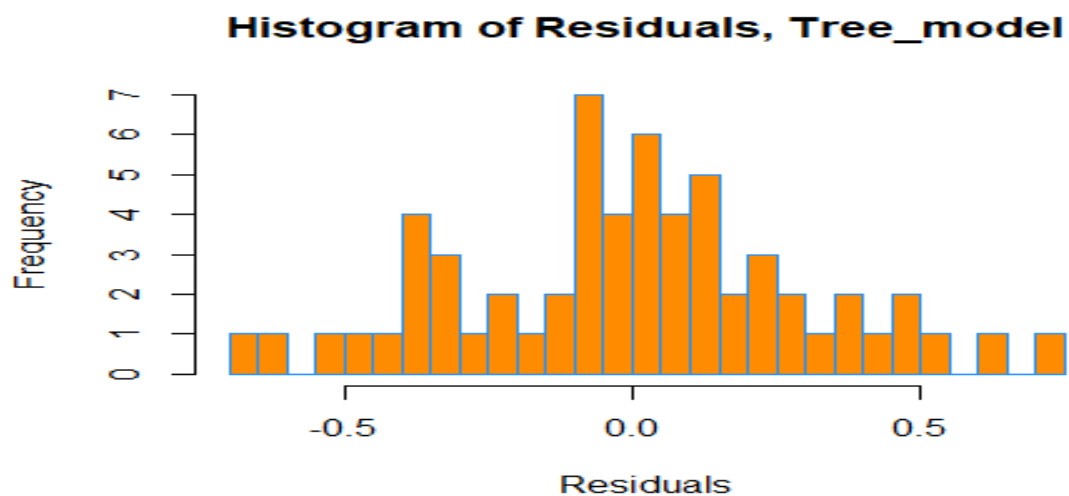
```
## Coefficients:
```

```
## (Intercept)      Diam
```

```
##      1.092      2.314
```

The histogram for the Tree\_model residual

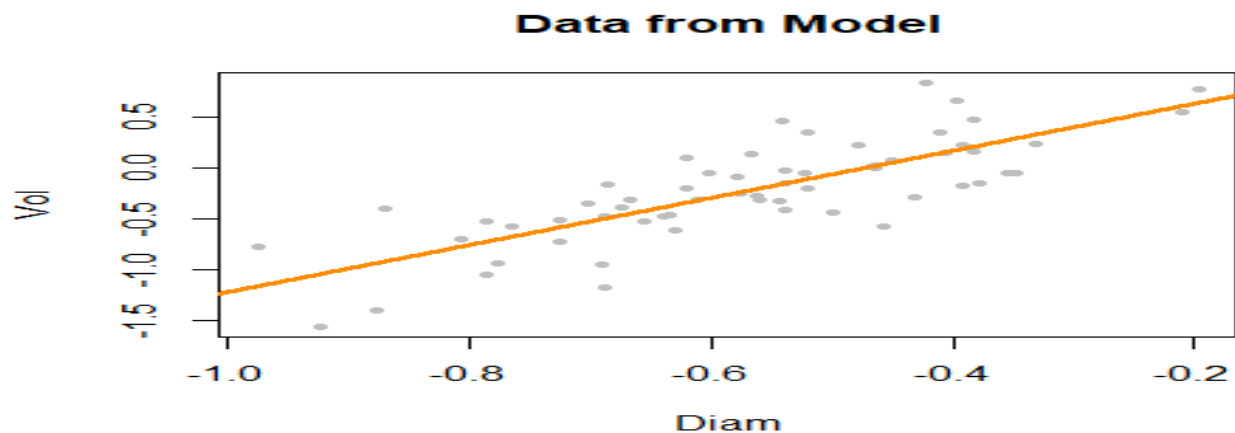
```
hist(resid(Tree_model),  
     xlab = "Residuals",  
     main = "Histogram of Residuals, Tree_model",  
     col = "darkorange",  
     border = "dodgerblue",  
     breaks = 20)
```



scatter plot for the model

```
plot(Vol~Diam ,data = TreeB, col = "grey", pch = 20,  
     main = "Data from Model")
```

```
abline(Tree_model, col = "darkorange", lwd = 3)
```



To create a summary of the fitted model

```
## summary output
summary(Tree_model)
```

```
##
## Call:
## lm(formula = Vol ~ Diam, data = TreeB)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68009 -0.17102  0.00589  0.16179  0.71973
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.0917     0.1392   7.844 1.12e-10 ***
## Diam          2.3139     0.2347   9.860 5.23e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.302 on 58 degrees of freedom
## Multiple R-squared:  0.6263, Adjusted R-squared:  0.6199
## F-statistic: 97.21 on 1 and 58 DF, p-value: 5.225e-14
```

The fitted model is  $\hat{y} = 1.0917 + 2.3139x$ . The slope is significantly different to 0,  $p < 0.001$  indicating that the variable Diam is of value in modeling Vol. The coefficient of determination  $R^2 = 0.6263$  which tells us that 62.63% of the variability in vol can be explained by Diam. The estimate for the error variance  $\sigma^2 = (0.302)^2$

It is possible to extract features of the model such as the model coefficients and the residuals:

```
## extract coefficients and residuals
Tree_model$coefficients

## (Intercept)      Diam
##    1.091685    2.313881
```

Tree\_model\$residuals

```
##           1           2           3           4           5
6
##  0.084193968  0.525179874 -0.087473511 -0.511096324  0.059715838  0.719729
176
##           7           8           9          10          11
12
## -0.083808239 -0.680093438  0.206531741 -0.011511606  0.131339853  0.252138
173
##          13          14          15          16          17
18
##  0.461826187 -0.226620105  0.240702083  0.013178541  0.391013418  0.484077
656
##          19          20          21          22          23
24
## -0.357471761 -0.020175874 -0.089038752 -0.085292782  0.038783586  0.139923
925
##          25          26          27          28          29
30
## -0.375443713 -0.323489215 -0.366640989 -0.046264795 -0.054453267  0.624123
505
##          31          32          33          34          35
36
## -0.600456244  0.127610947 -0.093176555 -0.134035997  0.078324572  0.022451
795
##          37          38          39          40          41
42
## -0.152492482  0.102812728  0.200283109  0.328855785 -0.369522618  0.272256
265
##          43          44          45          46          47
48
##  0.449765577 -0.255646289 -0.061577072 -0.240442378 -0.109967602 -0.455232
041
##          49          50          51          52          53
54
##  0.357037413  0.071372243 -0.007571071  0.028038965  0.009189196 -0.339435
856
##          55          56          57          58          59
60
##  0.155074972  0.142174788 -0.435066950  0.002587768  0.181922229 -0.328718
350
```

To view the ANOVA table we use the `summary.aov()` function.

```
#summary ANOVA table
summary.aov(Tree_model)
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Diam         1  8.867    8.867   97.21 5.23e-14 ***
## Residuals    58  5.290    0.091
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F-statistic is 97.21 with (1,58) degree of freedom and  $p < 0.001$ . As noted above, the variable Diam is of value in modeling vol.

I want to examine the model with the data before log transformation also

*#Fitting the model by using the data before applying log transformations.*

```
Tree<-read.csv("F:/semester2/SDA/treeB.csv")
```

```
Treemodel<-lm(Vol~Diam, data = Tree)
```

```
Treemodel
```

```
##
```

```
## Call:
```

```
## lm(formula = Vol ~ Diam, data = Tree)
```

```
##
```

```
## Coefficients:
```

```
## (Intercept)      Diam
```

```
##      -1.231      7.917
```

```
summary(Treemodel)
```

```
##
```

```
## Call:
```

```
## lm(formula = Vol ~ Diam, data = Tree)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
```

```
## -1.4337 -0.4349 -0.1079  0.2212  5.0821
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  -1.2311      0.3685  -3.341  0.00147 **
```

```
## Diam          7.9171      1.1897   6.655 1.11e-08 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Residual standard error: 1.014 on 58 degrees of freedom
```

```
## Multiple R-squared:  0.433, Adjusted R-squared:  0.4232
```

```
## F-statistic: 44.29 on 1 and 58 DF, p-value: 1.112e-08
```

```
AIC(Tree_model)
```

```
## [1] 30.56669
```

```
AIC(Treemodel)
```

```
## [1] 175.8926
```

The fitted model is  $\hat{y} = -1.2311 + 7.9171x$  The slope is significantly different to 0,  $p < 0.001$  indicating that the variable Diam is of value in modeling Vol. The coefficient of

determination  $R^2=0.433$  which tells us that 43.3% of the variability in vol can be explained by Diam. The estimate for the error variance  $\sigma^2 = (1.014)^2$  From the AIC we can say that the model after log transformations is the best model. Therefore the Tree\_model is the best model(model after applying log transformations )

### Question 1(c)

Calculate a 95% confidence interval for the  $\beta_1$  coefficient.

### Solution

To find the 95% confidence interval i used function `confint()` function for  $\beta_1$  coefficient i.e Diam

```
#the 95% confidence interval for Diam  $\beta_1$  coefficient
confint(Tree_model, 'Diam', level=0.95)

##           2.5 %    97.5 %
## Diam 1.844118 2.783643
```

Therefor for  $\beta_1$  coefficient the 95% confidence interval lies between the value 1.844118(2.5%) and 2.783643(97.5%)

### Question 1(d)

Test the hypothesis:  $H_0: \beta_1=0$   $H_A: \beta_1 \neq 0$  What do the results of the hypothesis test imply for the regression model?

### Solution

Examining the output for `summary(Tree_model)` we see that the t-statistic associated with  $\beta_1$  (Diam) is 6.655, the degree of freedom 58 and the associated p-value is 1.11e-08. The p-value is less than the 5% level of significance. In this instance we reject the null hypothesis at the 5% confidence level and conclude that Diam is strongly associated with Vol. There is enough evidence, based on the given data and a 5% level of significance, we can conclude that there is strong relationship among Diam and Vol.

### Question 1(e)

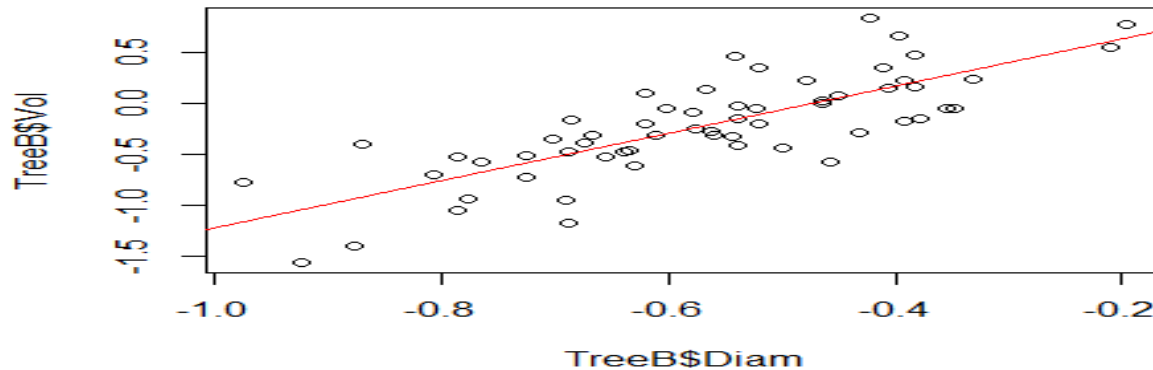
Plot the regression line onto a scatterplot of the data and plot a 95% prediction band



## Solution

Plotting the regression line on the original data

```
#Plotting the regression line on the original data  
plot(TreeB$Vol~TreeB$Diam)  
abline(Tree_model,col="red")
```



The 95% prediction band: 95% confidence interval for fitted value (mean) and predicted value (individual)

```
# 95% confidence interval for fitted value (mean) and predicted value (individual)  
predict(Tree_model, interval = 'confidence', se.fit = T)
```

```
## $fit  
##          fit          lwr          upr  
## 1  -0.46614587 -0.558209546 -0.37408220  
## 2  -0.91987483 -1.080985991 -0.75876366  
## 3   0.32350666  0.187416203  0.45959711  
## 4  -1.04485888 -1.228579732 -0.86113803  
## 5  -0.11518382 -0.196389042 -0.03397859  
## 6   0.11591067  0.011509908  0.22031144  
## 7  -0.37736677 -0.461273962 -0.29345958  
## 8  -0.49643233 -0.591899587 -0.40096508  
## 9   0.14391012  0.035650479  0.25216975  
## 10 -0.23944813 -0.317546438 -0.16134981  
## 11  0.63627605  0.444674997  0.82787710  
## 12 -0.30060638 -0.380124671 -0.22108809  
## 13 -0.11184969 -0.193244424 -0.03045495  
## 14 -0.70445228 -0.829116484 -0.57978808  
## 15 -0.01181431 -0.101121959  0.07749333  
## 16 -0.32130172 -0.401730039 -0.24087340  
## 17 -1.16364098 -1.369447806 -0.95783415  
## 18  0.17340845  0.060915713  0.28590119
```

```

## 19 0.18263813 0.068788527 0.29648774
## 20 0.01611804 -0.076079283 0.10831536
## 21 -0.11051913 -0.191990940 -0.02904731
## 22 -0.38476382 -0.469233892 -0.30029374
## 23 0.18660324 0.072166237 0.30104025
## 24 -0.45203840 -0.542615571 -0.36146123
## 25 0.09119643 -0.009941455 0.19233431
## 26 0.27146179 0.143883162 0.39904043
## 27 0.21489170 0.096189915 0.33359349
## 28 0.20862007 0.090874635 0.32636550
## 29 0.60447360 0.418750487 0.79019671
## 30 -0.16096242 -0.240106986 -0.08181785
## 31 0.03182001 -0.062113075 0.12575309
## 32 -0.15503787 -0.234390834 -0.07568490
## 33 -0.42394986 -0.511770787 -0.33612894
## 34 -0.58515807 -0.692018200 -0.47829795
## 35 -0.77147082 -0.907012562 -0.63592908
## 36 0.05181595 -0.044435153 0.14806705
## 37 -0.16762558 -0.246557007 -0.08869414
## 38 -0.67662690 -0.796938104 -0.55631570
## 39 -0.72752666 -0.855877242 -0.59917608
## 40 -0.49107802 -0.585923437 -0.39623259
## 41 -0.06122055 -0.146106566 0.02366547
## 42 0.20522675 0.087996249 0.32245725
## 43 -0.34452798 -0.426223255 -0.26283270
## 44 -0.15295516 -0.232385585 -0.07352473
## 45 -0.21039397 -0.288503207 -0.13228474
## 46 -0.36615793 -0.449256636 -0.28305922
## 47 -0.20125674 -0.279461424 -0.12305206
## 48 -0.93410780 -1.097752818 -0.77046277
## 49 -0.21665509 -0.294724306 -0.13858587
## 50 -0.58409243 -0.690804894 -0.47737996
## 51 0.15136627 0.042052159 0.26068039
## 52 -0.49643233 -0.591899587 -0.40096508
## 53 0.01787487 -0.074512817 0.11026255
## 54 0.28455963 0.154867325 0.41425193
## 55 -0.24589350 -0.324049741 -0.16773725
## 56 -0.34159517 -0.423116506 -0.26007383
## 57 -0.50574843 -0.602317249 -0.40917961
## 58 -0.15538540 -0.234725659 -0.07604515
## 59 -0.52967589 -0.629182877 -0.43016890
## 60 -0.72384793 -0.851606411 -0.59608945
##
## $se.fit
## [1] 0.04599234 0.08048646 0.06798684 0.09178161 0.04056778 0.05215559
## [7] 0.04191760 0.04769267 0.05408337 0.03901565 0.09571833 0.03972503
## [13] 0.04066245 0.06227862 0.04461551 0.04017966 0.10281512 0.05619811
## [19] 0.05687596 0.04605911 0.04070095 0.04219880 0.05716941 0.04524973
## [25] 0.05052555 0.06373458 0.05929997 0.05882220 0.09278188 0.03953833
## [31] 0.04692624 0.03964244 0.04387279 0.05338422 0.06771272 0.04808426

```

```
## [37] 0.03943185 0.06010399 0.06412023 0.04738202 0.04240659 0.05856496
## [43] 0.04081259 0.03968114 0.03902111 0.04151370 0.03906879 0.08175231
## [49] 0.03900112 0.05331045 0.05461016 0.04769267 0.04615421 0.06479051
## [55] 0.03904459 0.04072570 0.04824298 0.03963609 0.04971081 0.06382443
##
## $df
## [1] 58
##
## $residual.scale
## [1] 0.302019
```

```
predict(Tree_model,interval = 'prediction', se.fit = T)
```

```
## Warning in predict.lm(Tree_model, interval = "prediction", se.fit = T): pr
edictions on current data refer to _future_ responses
```

```
## $fit
##          fit          lwr          upr
## 1  -0.46614587 -1.077672270  0.14538053
## 2  -0.91987483 -1.545530982 -0.29421867
## 3   0.32350666 -0.296178284  0.94319160
## 4  -1.04485888 -1.676715009 -0.41300275
## 5  -0.11518382 -0.725169968  0.49480233
## 6   0.11591067 -0.497594288  0.72941563
## 7  -0.37736677 -0.987718501  0.23298496
## 8  -0.49643233 -1.108480381  0.11561571
## 9   0.14391012 -0.470263284  0.75808351
## 10 -0.23944813 -0.849028442  0.37013219
## 11  0.63627605  0.002083822  1.27046827
## 12 -0.30060638 -0.910370250  0.30915748
## 13 -0.11184969 -0.721861096  0.49816172
## 14 -0.70445228 -1.321728541 -0.08717603
## 15 -0.01181431 -0.622931874  0.59930324
## 16 -0.32130172 -0.931184929  0.28858149
## 17 -1.16364098 -1.802268620 -0.52501333
## 18  0.17340845 -0.441525229  0.78834213
## 19  0.18263813 -0.432545212  0.79782148
## 20  0.01611804 -0.595428496  0.62766457
## 21 -0.11051913 -0.720540825  0.49950257
## 22 -0.38476382 -0.995193181  0.22566555
## 23  0.18660324 -0.428689083  0.80189557
## 24 -0.45203840 -1.063342776  0.15926598
## 25  0.09119643 -0.521761716  0.70415457
## 26  0.27146179 -0.346409651  0.88933324
## 27  0.21489170 -0.401208069  0.83099148
## 28  0.20862007 -0.407296162  0.82453630
## 29  0.60447360 -0.027967617  1.23691481
## 30 -0.16096242 -0.770677662  0.44875283
## 31  0.03182001 -0.579990616  0.64363063
## 32 -0.15503787 -0.764780197  0.45470446
```

```

## 33 -0.42394986 -1.034851925 0.18695220
## 34 -0.58515807 -1.199086329 0.02877018
## 35 -0.77147082 -1.391035492 -0.15190615
## 36 0.05181595 -0.560354854 0.66398675
## 37 -0.16762558 -0.777313190 0.44206204
## 38 -0.67662690 -1.293038779 -0.06021503
## 39 -0.72752666 -1.345557960 -0.10949536
## 40 -0.49107802 -1.103029378 0.12087335
## 41 -0.06122055 -0.671707610 0.54926651
## 42 0.20522675 -0.410591246 0.82104475
## 43 -0.34452798 -0.954579560 0.26552361
## 44 -0.15295516 -0.762707575 0.45679726
## 45 -0.21039397 -0.819975689 0.39918774
## 46 -0.36615793 -0.976399036 0.24408318
## 47 -0.20125674 -0.810850696 0.40833721
## 48 -0.93410780 -1.560421223 -0.30779437
## 49 -0.21665509 -0.826231678 0.39292150
## 50 -0.58409243 -1.197994997 0.02981014
## 51 0.15136627 -0.462993874 0.76572642
## 52 -0.49643233 -1.108480381 0.11561571
## 53 0.01787487 -0.593700395 0.62945013
## 54 0.28455963 -0.333751710 0.90287096
## 55 -0.24589350 -0.855481237 0.36369425
## 56 -0.34159517 -0.951623481 0.26843315
## 57 -0.50574843 -1.117969268 0.10647240
## 58 -0.15538540 -0.765126080 0.45435527
## 59 -0.52967589 -1.142367047 0.08301527
## 60 -0.72384793 -1.341756535 -0.10593932
##
## $se.fit
## [1] 0.04599234 0.08048646 0.06798684 0.09178161 0.04056778 0.05215559
## [7] 0.04191760 0.04769267 0.05408337 0.03901565 0.09571833 0.03972503
## [13] 0.04066245 0.06227862 0.04461551 0.04017966 0.10281512 0.05619811
## [19] 0.05687596 0.04605911 0.04070095 0.04219880 0.05716941 0.04524973
## [25] 0.05052555 0.06373458 0.05929997 0.05882220 0.09278188 0.03953833
## [31] 0.04692624 0.03964244 0.04387279 0.05338422 0.06771272 0.04808426
## [37] 0.03943185 0.06010399 0.06412023 0.04738202 0.04240659 0.05856496
## [43] 0.04081259 0.03968114 0.03902111 0.04151370 0.03906879 0.08175231
## [49] 0.03900112 0.05331045 0.05461016 0.04769267 0.04615421 0.06479051
## [55] 0.03904459 0.04072570 0.04824298 0.03963609 0.04971081 0.06382443
##
## $df
## [1] 58
##
## $residual.scale
## [1] 0.302019

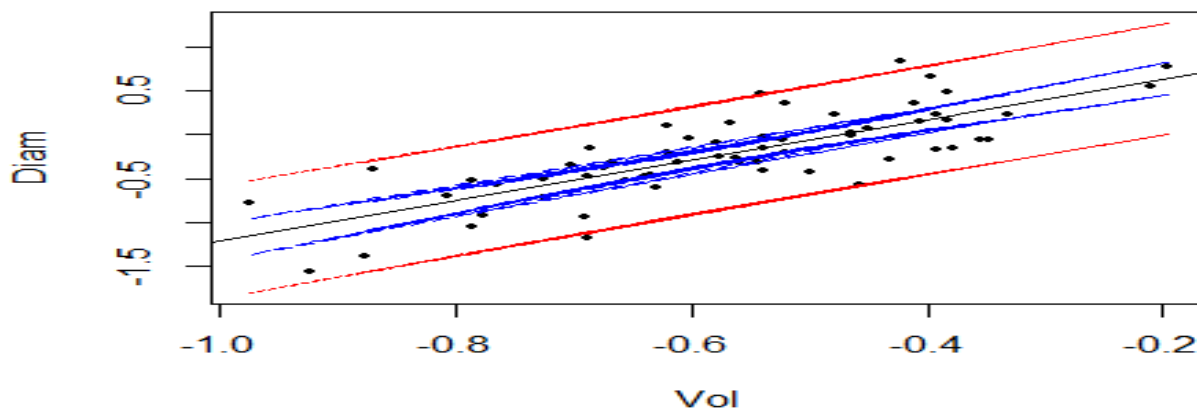
## create a set of 95% confidence interval
dist_ci_band = predict(Tree_model,
                        newdata = data.frame(Diam=TreeB$Diam),

```

```

        interval = "confidence", level = 0.95)
## create a set of 95% prediction interval
dist_pi_band = predict(Tree_model,
                        newdata = data.frame(Diam=TreeB$Diam),
                        interval = "prediction", level = 0.95)
## plot scatter graph of Diam against Vol
plot(TreeB$Vol~TreeB$Diam, xlab = "Vol", ylab = "Diam", pch = 20,
     cex = 0.75, ylim = c(min(dist_pi_band), max(dist_pi_band)))
## add regression line
abline(Tree_model)
lines(TreeB$Diam, dist_ci_band[, "lwr"], col = "blue", lwd = 1, lty = 2)
lines(TreeB$Diam, dist_ci_band[, "upr"], col = "blue", lwd = 1, lty = 2)
lines(TreeB$Diam, dist_pi_band[, "lwr"], col = "red", lwd = 1, lty = 3)
lines(TreeB$Diam, dist_pi_band[, "upr"], col = "red", lwd = 1, lty = 3)

```



Finally plotted these bands onto the scatterplot. The bands are narrowest at the center of the regression line.

## Question 1(f)

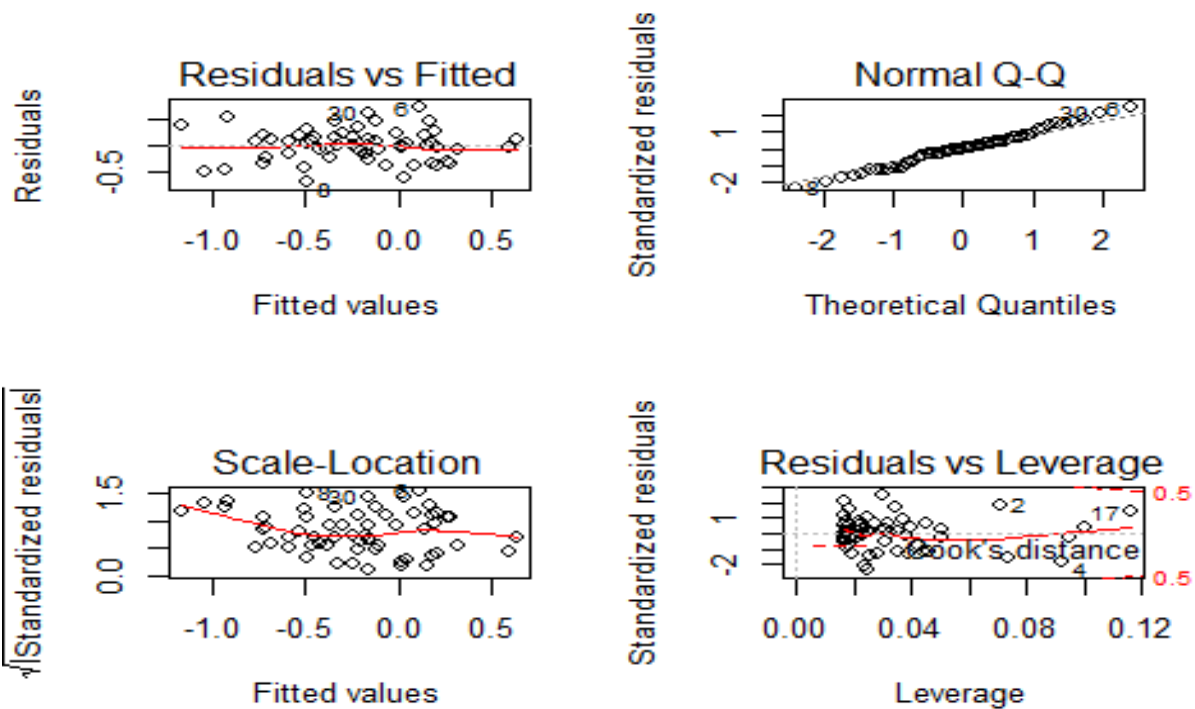
Plot the studentized residuals against the fitted values and identify any outliers.

## Solution

```

## diagnostics
par(mfrow = c(2,2))
#plots for the model
plot(Tree_model)

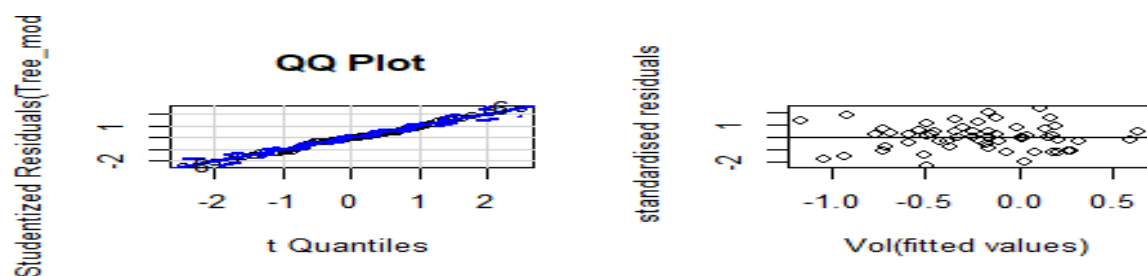
```



```
#Q-Q plot for model
qqPlot(Tree_model,main="QQ Plot")

## [1] 6 8

#studentized residuals
stdres <- rstandard(Tree_model)
#Plot of the studentized residuals against the fitted values
plot(stdres~fitted(Tree_model), xlab = "Vol(fitted values)", ylab = "standard
ised residuals")
abline(0,0)
```



The `plot(Tree_model)` command produces a series of four plots. The first graph (top left) shows a plot of the residuals against the fitted values. The second graph (top right) is the Normal Q-Q plot (quantile to quantile plot) which plots the probability distribution of the standardized residuals with the standard normal probability distribution. The third graph (bottom left) is the Scale – Location plot which shows a plot of the square root of the positive standardised residuals against the fitted values.

The fourth graph shows Cook's distance which measures how influential a point is in the analysis. Cook's distance measures the effect of deleting a given observation on the statistical model. Data points with large residuals (outliers) and/or high leverage may often have a large Cook's distance value and are considered to merit closer examination in the analysis.

Standard residual is defined as the residual divided by the standard deviation of the residuals. The standardized residuals cannot be extracted directly from the fitted model, the way that the residuals can, but they can be calculated using the `rstandard()` function.

From the figure we can say that there are no outliers. An outlier is a point that falls far from the other data points. Therefore from the plot we can say that there are no outliers.

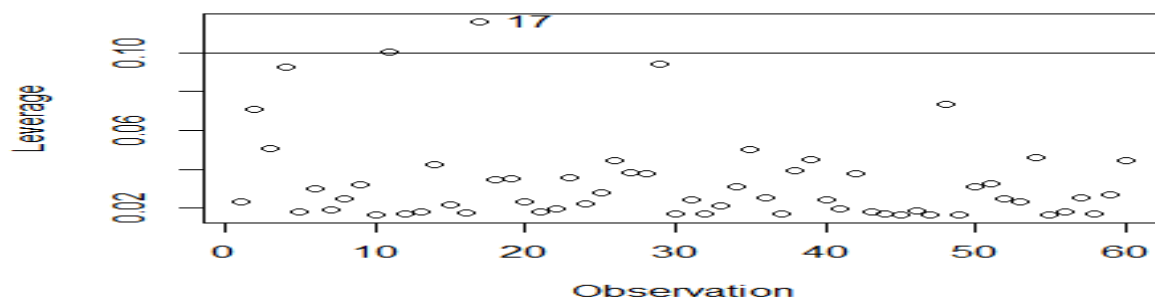
## Question 1(g)

Plot the leverage of each case and identify any observations that have high leverage.

## Solution

Leverage points are those observations made at the independent variables' extreme or outlying values, so that the absence of nearby observations implies that the regression model fitted passes close to the particular observation.

```
## calculate Leverage
h<- lm.influence(Tree_model)$hat
windows(5,5)
plot(h, xlab = "Observation", ylab = "Leverage")
abline(h=0.1)
identify(h,n=1)
```



```
## [1] 17
```

We can calculate and then plot the leverages by using 'hat' matrix. A rule of thumb is that leverages of more than should be looked at more  $2(k+1)/n$  closely, where  $k+1$  represents the number of parameters in the model and  $n$  represents the number of observations. In this case there are three parameters in the model so we need to check that  $h_i = 6/60 = 0.1$ . We can plot the line  $y = 0.1$  onto our plot and then identify points with a leverage greater than this value using the `identify()` function which allows us to identify points on the graph using the mouse. From the plot we can say that the observation 17 have high leverage

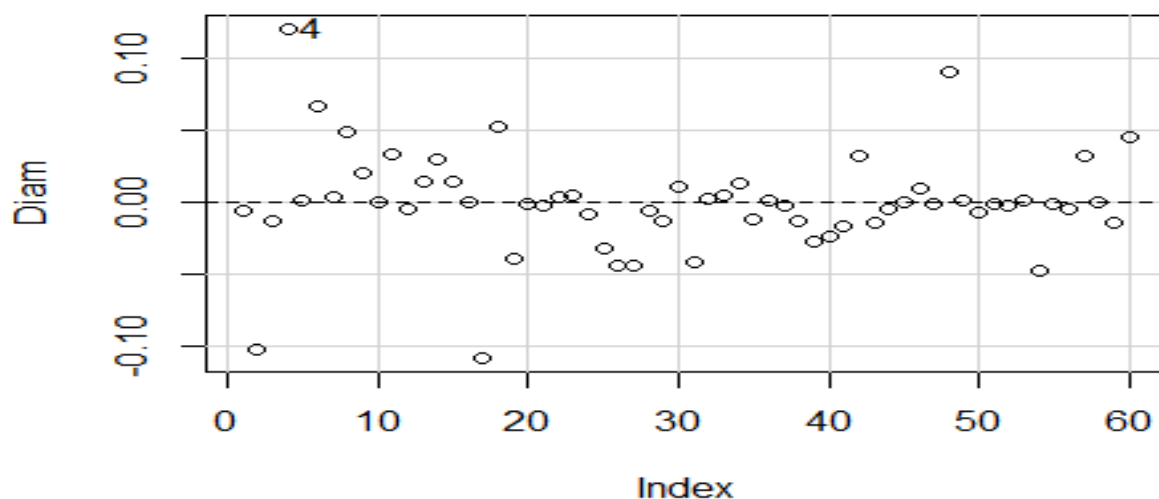
## Question 1(h)

Identify the observation that has the largest influence on the estimate of the  $\beta_1$  coefficient. Explain why this observation has a large influence.

## Solution

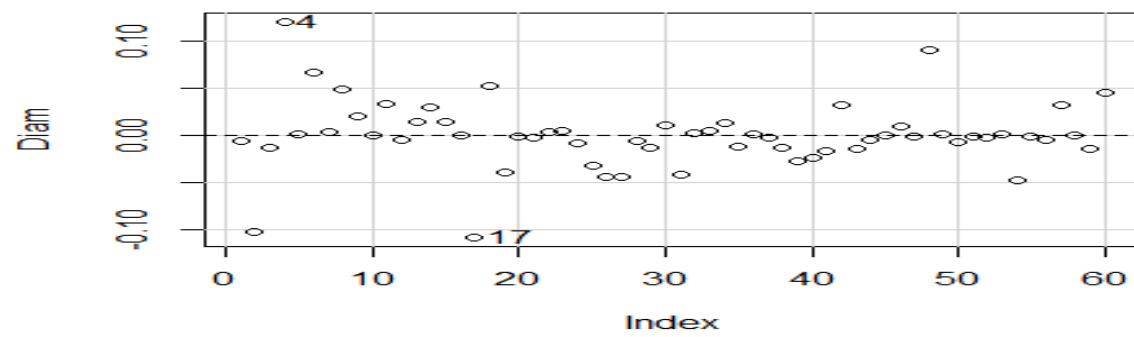
If the values of the parameter change a lot when a point is excluded from the equations, the point is assumed to have influence

```
#influential points  
dfbetaPlots(Tree_model, id.n = 1)
```

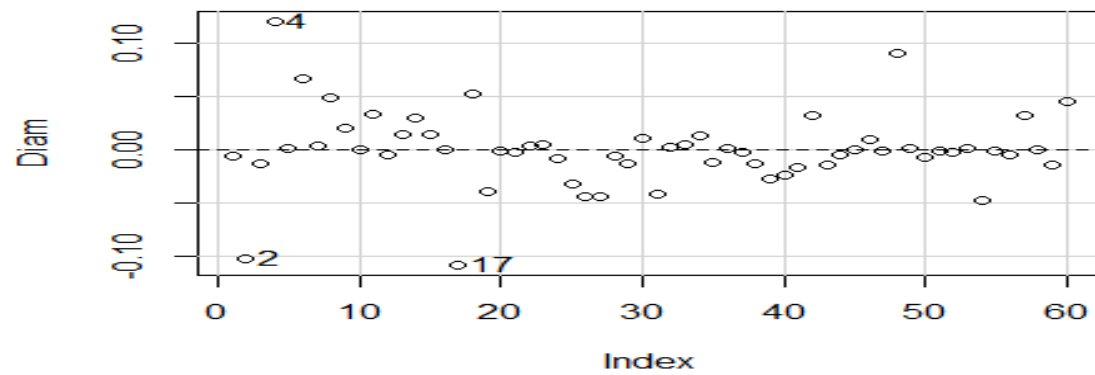


```
dfbetaPlots(Tree_model, id.n = 2)
```

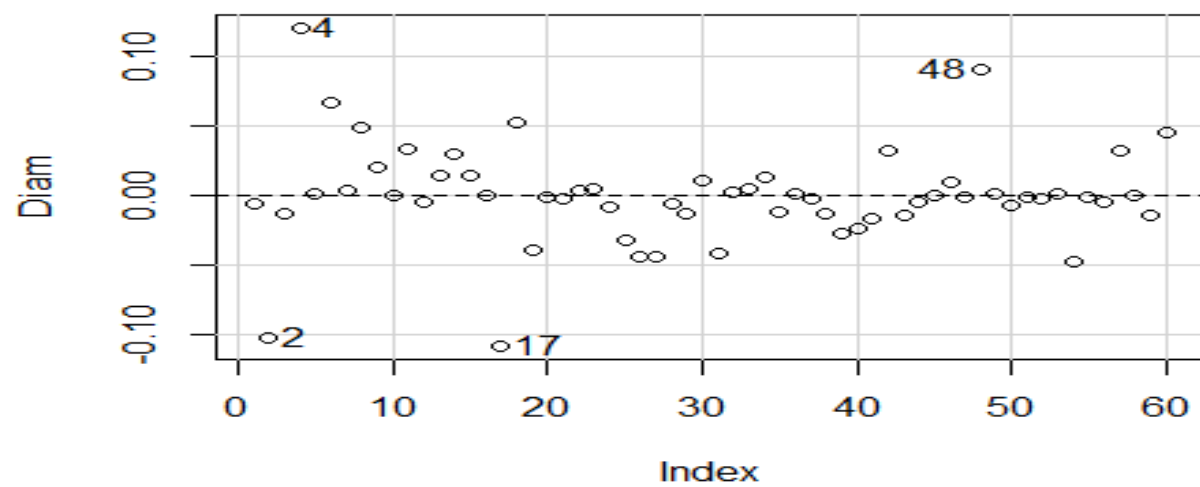




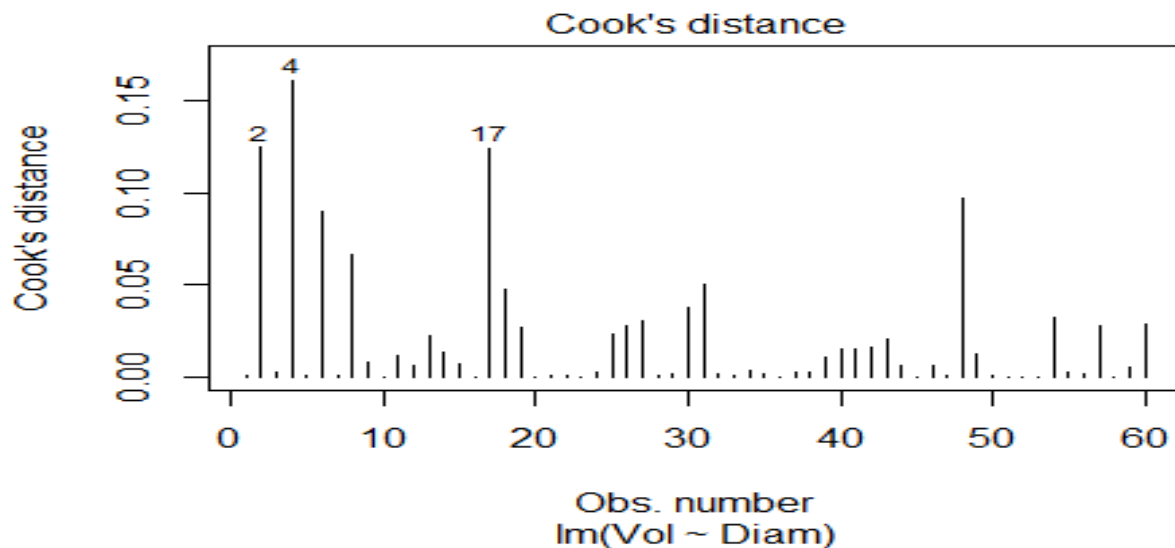
```
dfbetaPlots(Tree_model,id.n = 3)
```



```
dfbetaPlots(Tree_model,id.n = 4)
```



```
# Cook's D plot
# identify D values > 4/(n-k-1)
cutoff <- 4/((nrow(TreeB)-length(Tree_model$coefficients)-2))
plot(Tree_model, which=4, cook.levels=cutoff)
```

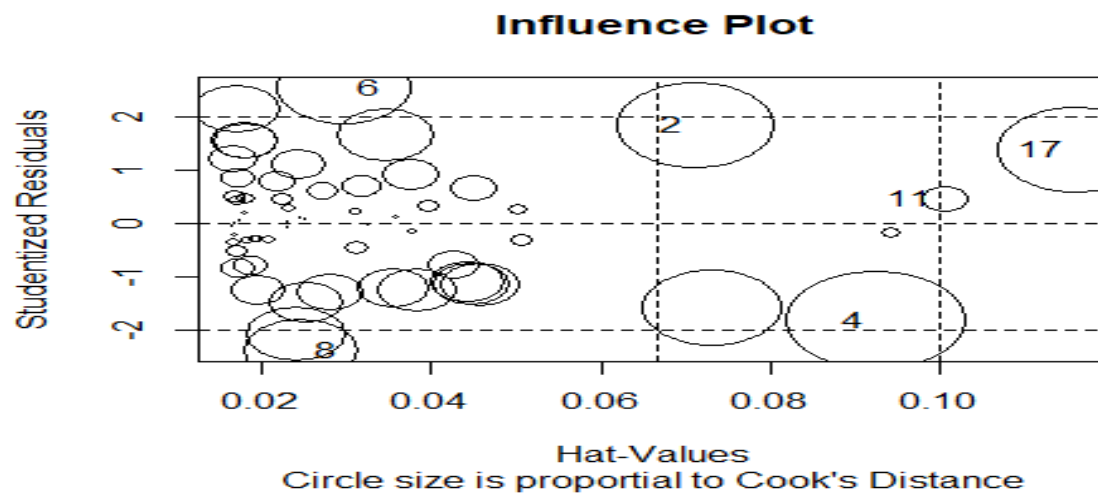


```
# Influence Plot
influencePlot(Tree_model, id.method="identify", main="Influence Plot", sub="C
ircle size is proportional to Cook's Distance" )

## Warning in plot.window(...): "id.method" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "id.method" is not a graphical paramete
r
## Warning in axis(side = side, at = at, labels = labels, ...): "id.method" i
s not
## a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "id.method" i
s not
## a graphical parameter

## Warning in box(...): "id.method" is not a graphical parameter
## Warning in title(...): "id.method" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "id.method" is not
a
## graphical parameter
```



##	StudRes	Hat	CookD
## 2	1.8409220	0.07101943	0.12441752
## 4	-1.8108358	0.09235126	0.16051447
## 6	2.5294964	0.02982176	0.08996424
## 8	-2.3694084	0.02493646	0.06649785
## 11	0.4553653	0.10044346	0.01173706
## 17	1.3878554	0.11588987	0.12425605

From the dfbetaplots we can see the top four observations that has the largest influence on the estimate of the  $\beta_1$  coefficient. The cooks plot provides the three observations that has **the largest influence. The observations 4,17 and 2 have the largest influence** An **significant** observation is a function of two factors: (1) how much the effect of the observation on the predictor variable varies from the mean of the predictor variable, and (2) the distance between the observer's expected score and its actual score. Observations have significant impact because of these reasons.

Therefore I can say that Diam(Diameter) is useful to predict the value of vol(volume).

#####

## QUESTION 2

The div dataset is available on Blackboard. The dataset consists of 77 observations and 7 variables.

Firstly I loaded the first dataset divusaB.csv by using readr package which will be helpful for reading .csv files.And assigned file to divusaB variable.

```
#Loading the divusaB.csv file and storing it in variable divusaB
divusaB<-read.csv("F:/semester2/SDA/divusaB.csv")
View(divusaB)
#checking the class of divusaB
class(divusaB)

## [1] "data.frame"
```

Please answer the questions below and write up the analysis in the form of a report.

## Question 2(a)

Make a numerical and graphical summary of the data, commenting on the results. Include: boxplots, histograms, scatterplots and correlation coefficients.

### Solution

I have to find the numerical and graphical summary of the data So I will be following the same steps which I have followed in Question1 So,I will be performing the Exploratory Data Analysis(EDA) to know about the data. STEPS IN EDA are as follows (i)Understanding your variables (Numerical data Analysis) (ii)Cleaning your dataset (iii)Analyzing relationships between variables (Graphical data Analysis)

- (i) Understanding your variables (Numerical data Analysis) Firstly I want to know about the given dataset so I will be going through the dataset. In order to understand or to explore dataset

*#Looking into the data by using dim,names,head,tail,str,summary,df\_status,describe,skim*

```
dim(divusaB)
```

```
## [1] 77 7
```

```
names(divusaB)
```

```
## [1] "year"      "divorce"   "unemployed" "femlab"    "marriage"
## [6] "birth"     "military"
```

```
head(divusaB)
```

```
##   year divorce unemployed femlab marriage  birth military
## 1 1932  13.536      8.951 39.174   66.128  85.732    1.856
## 2 1933   8.734      6.771 23.990   77.576 104.122    2.331
## 3 1922  18.159      7.371 45.922   64.847  84.774    7.672
## 4 1923  18.565      5.161 38.477   63.876  95.591    9.211
## 5 1931  13.505     17.959 35.689   66.151  78.464    3.515
## 6 1921  20.481     22.416 50.033   64.878  86.429    5.381
```

```
tail(divusaB)
```

```
##   year divorce unemployed femlab marriage  birth military
## 72 1984   9.636     10.424 28.728   76.408 104.096    9.639
## 73 1982  21.097      5.799 52.379   62.116  72.397    3.767
## 74 1985  12.907      9.777 34.156   77.235 108.854   10.625
## 75 1978  14.868      8.973 35.984   70.472  85.000   10.799
## 76 1995  23.482      4.792 51.720   61.160  72.075   10.448
## 77 1987  16.817      5.262 47.378   67.408  83.341   10.192
```

```
str(divusaB)
```

```
## 'data.frame': 77 obs. of 7 variables:
## $ year : int 1932 1933 1922 1923 1931 1921 1924 1925 1926 1930 ...
## $ divorce : num 13.54 8.73 18.16 18.57 13.51 ...
## $ unemployed: num 8.95 6.77 7.37 5.16 17.96 ...
## $ femlab : num 39.2 24 45.9 38.5 35.7 ...
## $ marriage : num 66.1 77.6 64.8 63.9 66.2 ...
## $ birth : num 85.7 104.1 84.8 95.6 78.5 ...
## $ military : num 1.86 2.33 7.67 9.21 3.52 ...
```

```
summary(divusaB)
```

```
##      year      divorce      unemployed      femlab
## Min.   :1920   Min.    : 2.603   Min.    : 3.190   Min.    :10.51
## 1st Qu.:1939   1st Qu.:10.266   1st Qu.: 5.801   1st Qu.:30.00
## Median :1958   Median :14.302   Median : 7.285   Median :38.34
## Mean   :1958   Mean    :13.971   Mean    : 8.454   Mean    :37.05
## 3rd Qu.:1977   3rd Qu.:17.394   3rd Qu.: 8.951   3rd Qu.:44.11
## Max.    :1996   Max.    :24.182   Max.    :25.365   Max.    :61.96
##      marriage      birth      military
## Min.   :49.49   Min.    : 72.08   Min.    : 1.856
## 1st Qu.:66.15   1st Qu.: 84.36   1st Qu.: 7.601
## Median :70.60   Median : 91.47   Median :10.300
## Mean   :71.18   Mean    : 91.54   Mean    :13.096
## 3rd Qu.:76.41   3rd Qu.: 99.40   3rd Qu.:13.655
## Max.    :87.08   Max.    :111.17   Max.    :86.275
```

```
glimpse(divusaB)
```

```
## Observations: 77
## Variables: 7
## $ year      <int> 1932, 1933, 1922, 1923, 1931, 1921, 1924, 1925, 1926, 1
9...
## $ divorce   <dbl> 13.536, 8.734, 18.159, 18.565, 13.505, 20.481, 18.541,
1...
## $ unemployed <dbl> 8.951, 6.771, 7.371, 5.161, 17.959, 22.416, 4.239, 15.4
5...
## $ femlab    <dbl> 39.174, 23.990, 45.922, 38.477, 35.689, 50.033, 40.664,
...
## $ marriage  <dbl> 66.128, 77.576, 64.847, 63.876, 66.151, 64.878, 75.446,
...
## $ birth     <dbl> 85.732, 104.122, 84.774, 95.591, 78.464, 86.429, 90.208
,...
## $ military  <dbl> 1.856, 2.331, 7.672, 9.211, 3.515, 5.381, 16.776, 8.911
,...
```

```
df_status(divusaB)#funmodeling
```

```
##      variable q_zeros p_zeros q_na p_na q_inf p_inf      type unique
## 1      year      0      0      0      0      0      0 integer      77
```

```
## 2    divorce      0      0      0      0      0      0 numeric    77
## 3 unemployed      0      0      0      0      0      0 numeric    77
## 4    femlab       0      0      0      0      0      0 numeric    77
## 5   marriage      0      0      0      0      0      0 numeric    77
## 6     birth       0      0      0      0      0      0 numeric    77
## 7   military      0      0      0      0      0      0 numeric    77
```

**describe**(divusaB)

```
##          vars  n    mean    sd  median trimmed   mad     min     max range
## year          1 77 1958.00 22.37 1958.00 1958.00 28.17 1920.00 1996.00 76.00
## divorce        2 77   13.97  4.71   14.30   14.09  4.91    2.60   24.18 21.58
## unemployed     3 77    8.45  4.39    7.29    7.60  2.29    3.19   25.36 22.17
## femlab         4 77   37.05  9.96   38.34   37.22 10.61   10.51   61.96 51.44
## marriage       5 77   71.18  7.35   70.60   71.17  7.43   49.49   87.08 37.60
## birth          6 77   91.54  9.91   91.47   91.45 11.14   72.08  111.17 39.10
## military       7 77   13.10 14.01   10.30   10.51  4.78    1.86   86.28 84.42
##          skew kurtosis   se
## year        0.00    -1.25 2.55
## divorce     -0.21    -0.41 0.54
## unemployed  2.04     4.01 0.50
## femlab      -0.19    -0.31 1.14
## marriage    -0.07    -0.21 0.84
## birth       0.08    -0.82 1.13
## military    3.95    16.23 1.60
```

**skim**(divusaB)

### Data summary

Name	divusaB
Number of rows	77
Number of columns	7

### Column type frequency:

numeric	7
---------	---

Group variables	None
-----------------	------

## Variable type: numeric

skim_var iable	n_mis sing	complete _rate	mean	sd	p0	p25	p50	p75	p100	hist
year	0	1	1958 .00	22. 37	1920 .00	1939 .00	1958 .00	1977 .00	1996 .00	
divorce	0	1	13.9 7	4.7 1	2.60	10.2 7	14.3 0	17.3 9	24.1 8	
unemplo yed	0	1	8.45	4.3 9	3.19	5.80	7.29	8.95	25.3 6	
femlab	0	1	37.0 5	9.9 6	10.5 1	30.0 0	38.3 4	44.1 1	61.9 6	
marriage	0	1	71.1 8	7.3 5	49.4 9	66.1 5	70.6 0	76.4 1	87.0 8	
birth	0	1	91.5 4	9.9 1	72.0 8	84.3 6	91.4 7	99.4 0	111. 17	
military	0	1	13.1 0	14. 01	1.86	7.60	10.3 0	13.6 5	86.2 8	

*#Descriptive stastics mean,median,standard deviation,variance,IQR,shaprio tes  
t for normality*

```
mean(divusaB$year)
```

```
## [1] 1958
```

```
mean(divusaB$divorce)
```

```
## [1] 13.97126
```

```
mean(divusaB$unemployed)
```

```
## [1] 8.453558
```

```
mean(divusaB$femlab)
```

```
## [1] 37.05366
```

```
mean(divusaB$marriage)
```

```
## [1] 71.1777
```

```
mean(divusaB$birth)
```

```
## [1] 91.54112
```

```
mean(divusaB$military)
```

```
## [1] 13.09644
```

```
median(divusaB$year)
```

```
## [1] 1958
median(divusaB$divorce)
## [1] 14.302
median(divusaB$unemployed)
## [1] 7.285
median(divusaB$femlab)
## [1] 38.34
median(divusaB$marriage)
## [1] 70.601
median(divusaB$birth)
## [1] 91.472
median(divusaB$military)
## [1] 10.3
sd(divusaB$year)
## [1] 22.37186
sd(divusaB$divorce)
## [1] 4.714373
sd(divusaB$unemployed)
## [1] 4.392784
sd(divusaB$femlab)
## [1] 9.961989
sd(divusaB$marriage)
## [1] 7.350366
sd(divusaB$birth)
## [1] 9.914067
sd(divusaB$military)
## [1] 14.01326
var(divusaB$year)
```



```
## [1] 500.5
var(divusaB$divorce)
## [1] 22.22531
var(divusaB$unemployed)
## [1] 19.29655
var(divusaB$femlab)
## [1] 99.24123
var(divusaB$marriage)
## [1] 54.02787
var(divusaB$birth)
## [1] 98.28873
var(divusaB$military)
## [1] 196.3714
IQR(divusaB$year)
## [1] 38
IQR(divusaB$divorce)
## [1] 7.128
IQR(divusaB$unemployed)
## [1] 3.15
IQR(divusaB$femlab)
## [1] 14.11
IQR(divusaB$marriage)
## [1] 10.257
IQR(divusaB$birth)
## [1] 15.037
IQR(divusaB$military)
## [1] 6.054
#NORMALITY
shapiro.test(divusaB$year)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  divusaB$year
## W = 0.95491, p-value = 0.008161

shapiro.test(divusaB$divorce)

##
##  Shapiro-Wilk normality test
##
## data:  divusaB$divorce
## W = 0.98964, p-value = 0.7921

shapiro.test(divusaB$unemployed)

##
##  Shapiro-Wilk normality test
##
## data:  divusaB$unemployed
## W = 0.75615, p-value = 5.572e-10

shapiro.test(divusaB$femlab)

##
##  Shapiro-Wilk normality test
##
## data:  divusaB$femlab
## W = 0.9911, p-value = 0.8742

shapiro.test(divusaB$marriage)

##
##  Shapiro-Wilk normality test
##
## data:  divusaB$marriage
## W = 0.9884, p-value = 0.7145

shapiro.test(divusaB$birth)

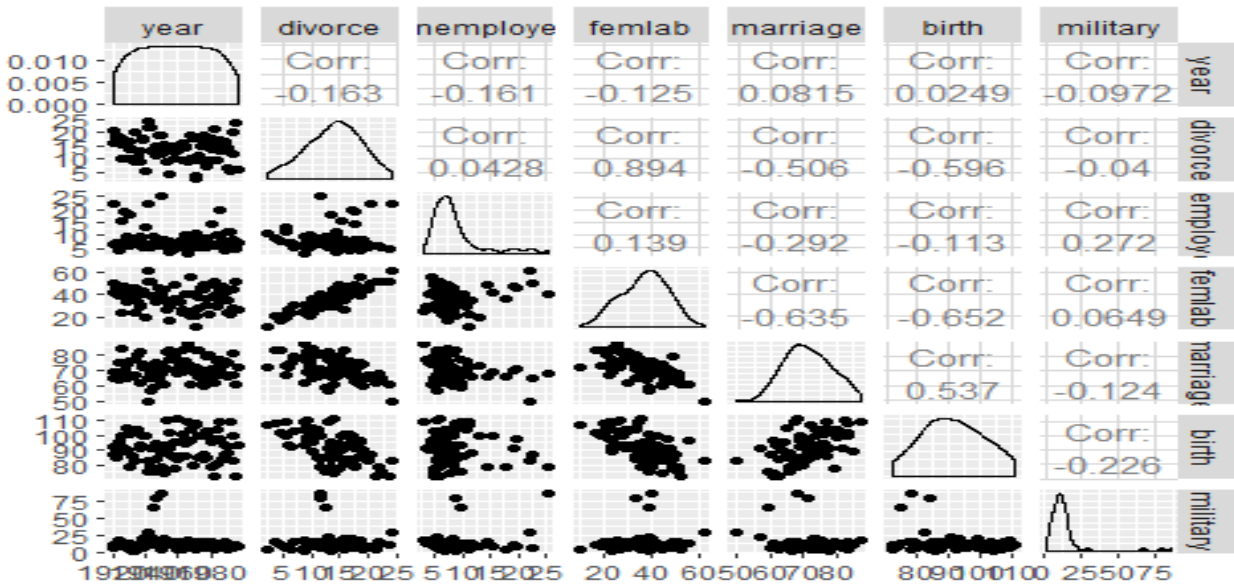
##
##  Shapiro-Wilk normality test
##
## data:  divusaB$birth
## W = 0.98049, p-value = 0.2842

shapiro.test(divusaB$military)

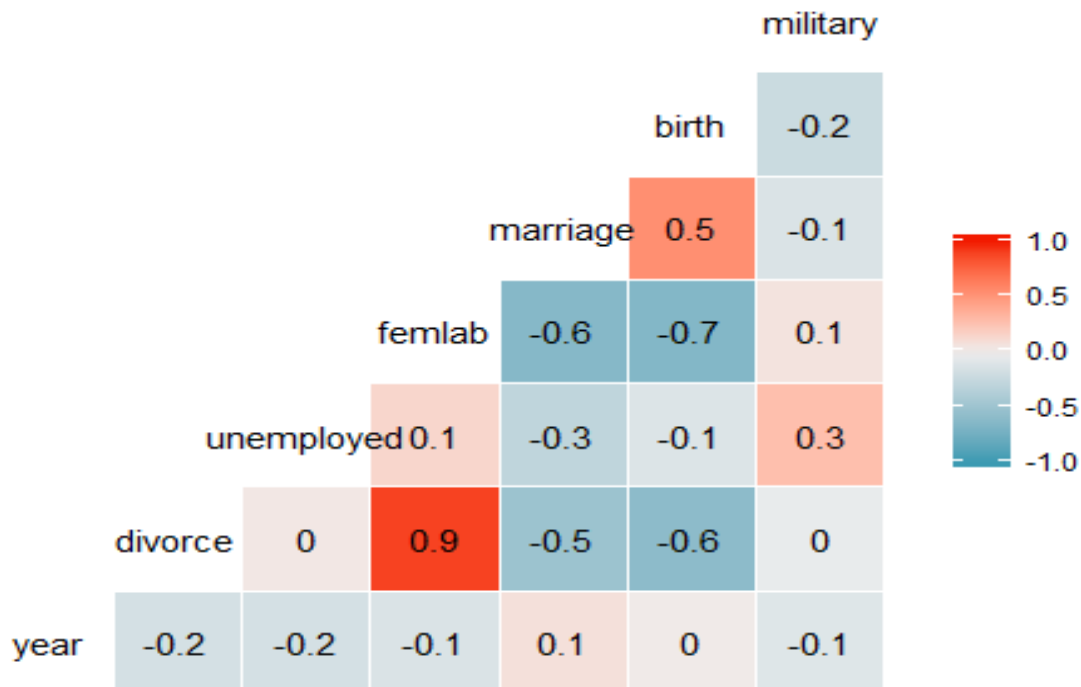
##
##  Shapiro-Wilk normality test
##
```

```
## data: divusaB$military
## W = 0.49914, p-value = 8.888e-15
```

```
ggpairs(divusaB)#Make a matrix of plots with data set
```



```
#The ggcorr() function draws a correlation matrix plot using ggplot2.
ggcorr(divusaB, palette = "RdBu", label = TRUE)
```



Now I am performing visualization of data by using some visualization packages to know the data in a better way

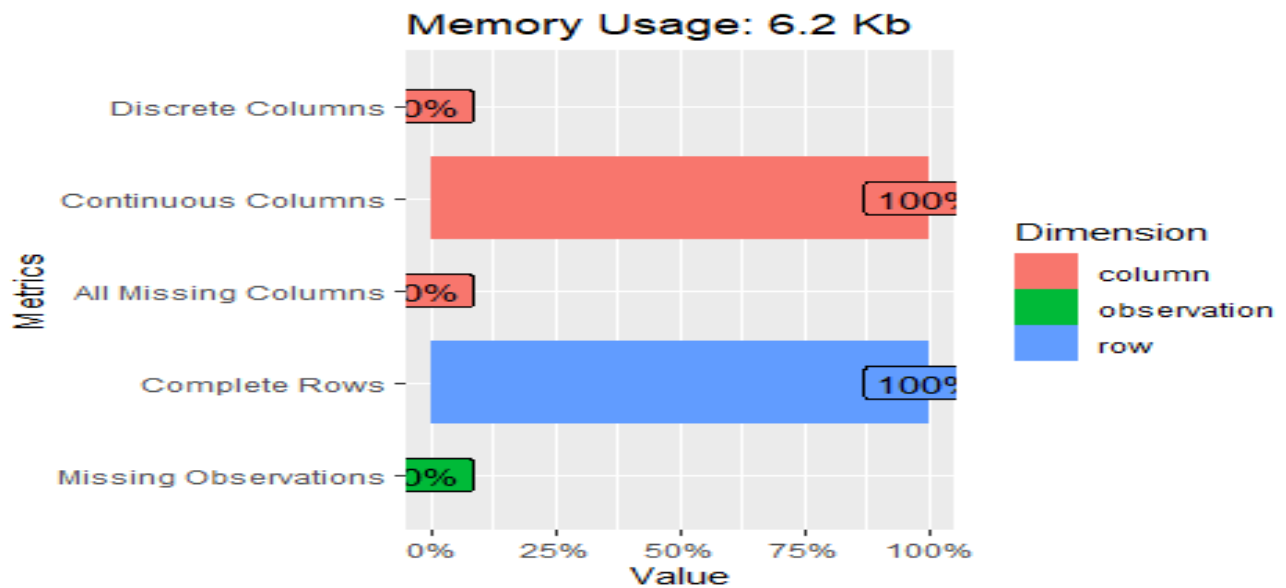
*#Gives the information about dataset in the form of plots*

```
introduce(divusaB)
```

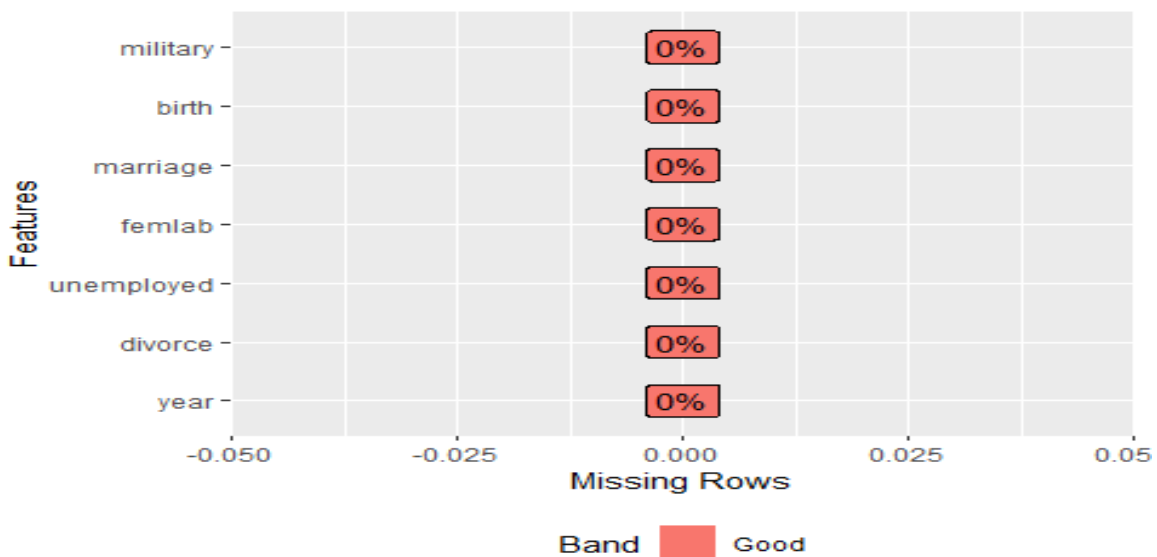
```
## rows columns discrete_columns continuous_columns all_missing_columns
## 1 77 7 0 7 0
## total_missing_values complete_rows total_observations memory_usage
## 1 0 77 539 6336
```

```
plot_str(divusaB)
```

```
plot_intro(divusaB)
```



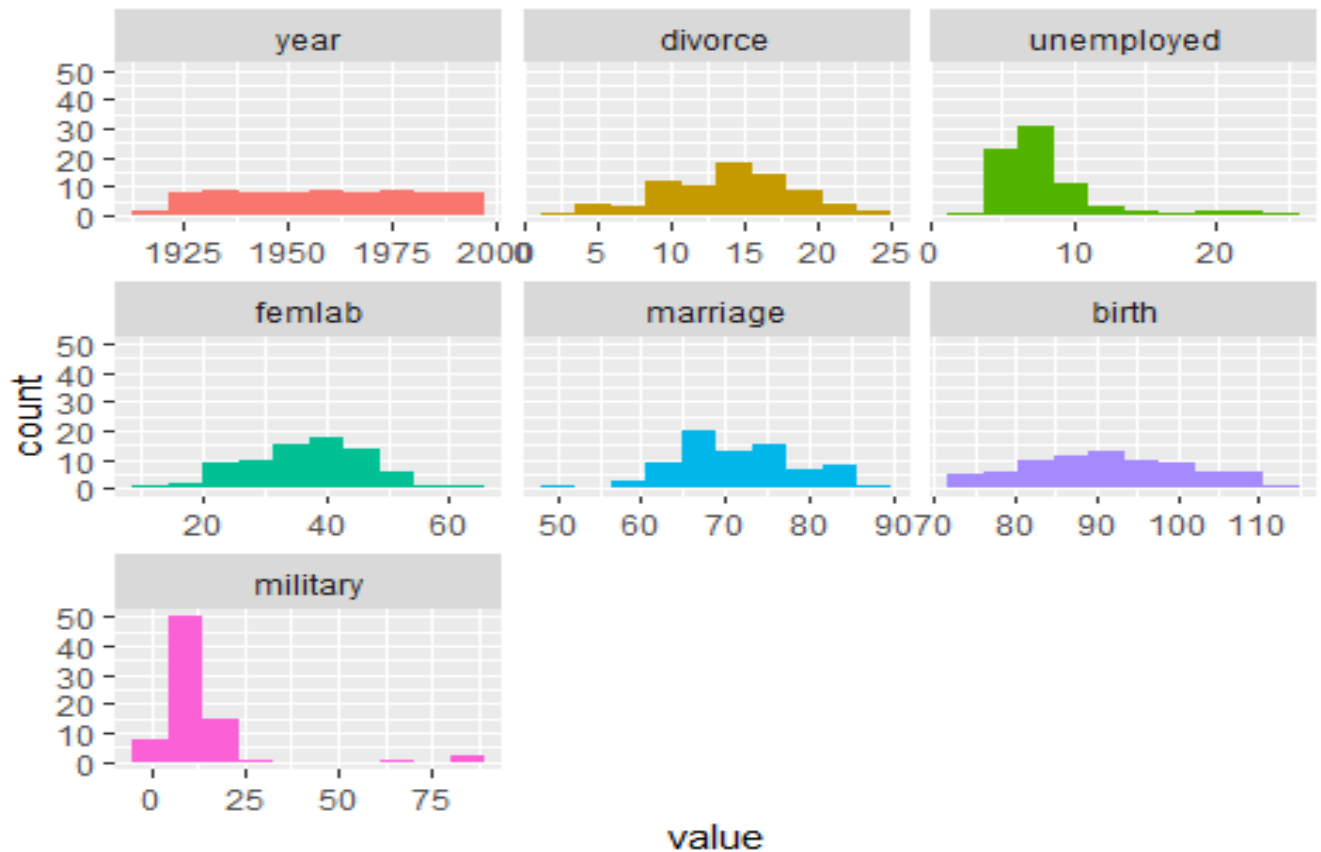
```
plot_missing(divusaB)
```



```
profile_missing(divusaB)
```

```
##      feature num_missing pct_missing
## 1      year           0           0
## 2    divorce           0           0
## 3 unemployed           0           0
## 4     femlab           0           0
## 5   marriage           0           0
## 6     birth           0           0
## 7   military           0           0
```

```
plot_num(divusaB)
```



```
profiling_num(divusaB)
```

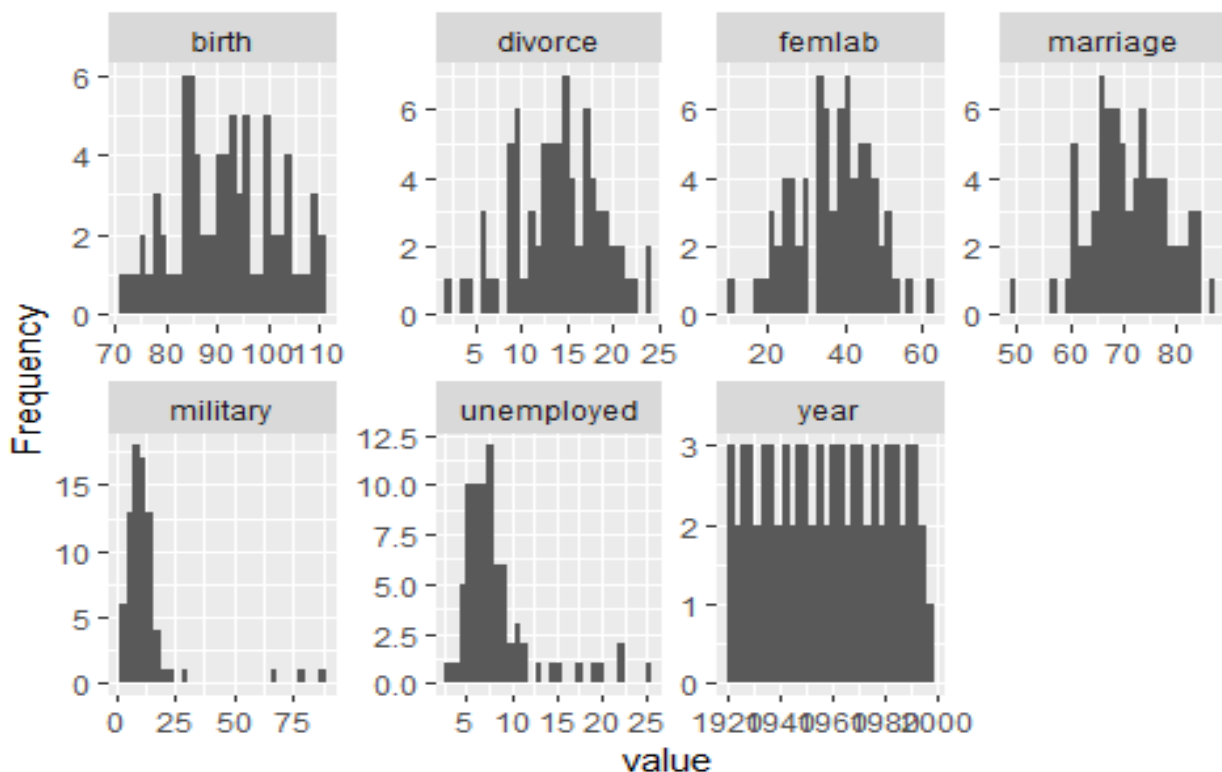
```
##      variable      mean  std_dev variation_coef      p_01      p_05
p_25
## 1      year 1958.000000 22.371857      0.01142587 1920.76000 1923.8000 193
9.000
## 2    divorce   13.971260  4.714373      0.33743363   3.46560   5.6918   1
0.266
## 3 unemployed    8.453558  4.392784      0.51963727   3.59052   4.6150
5.801
## 4     femlab   37.053662  9.961989      0.26885302  15.69772  20.9186   3
0.003
## 5   marriage   71.177701  7.350366      0.10326781  55.33040  60.7190   6
```

```

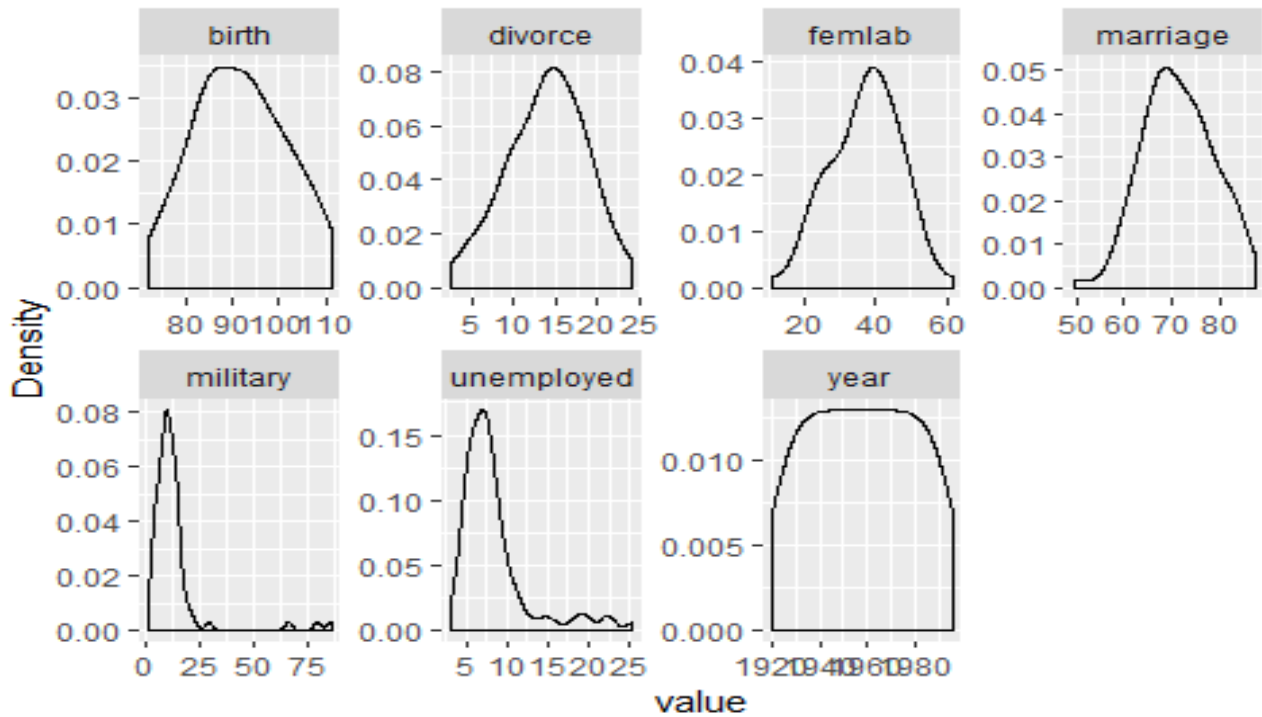
6.151
## 6      birth      91.541117  9.914067      0.10830179  72.31972  75.1532  8
4.359
## 7    military     13.096442 14.013259      1.07000507   2.21700   3.4994
7.601
##      p_50      p_75      p_95      p_99      skewness      kurtosis      iqr
## 1 1958.000 1977.000 1992.2000 1995.24000 0.00000000 1.799595 38.000
## 2  14.302  17.394  21.2404  23.65000 -0.21697132 2.660623  7.128
## 3   7.285   8.951  19.3506  23.12376  2.08497588 7.193909  3.150
## 4  38.340  44.113  51.5960  57.37496 -0.19444528 2.762845 14.110
## 5  70.601  76.408  83.1756  85.27800 -0.06763021 2.866814 10.257
## 6  91.472  99.396 108.9434 110.24252  0.08177618 2.235953 15.037
## 7  10.300  13.655  23.8058  81.47484  4.02831437 19.734729  6.054
##      range_98      range_80
## 1 [1920.76, 1995.24] [1927.6, 1988.4]
## 2 [ 3.4656, 23.65] [ 8.1204, 19.4946]
## 3 [ 3.59052, 23.12376] [ 5.1196, 13.6166]
## 4 [15.69772, 57.37496] [23.698, 48.7944]
## 5 [ 55.3304, 85.278] [ 61.886, 81.8898]
## 6 [72.31972, 110.24252] [78.4568, 104.3936]
## 7 [ 2.217, 81.47484] [ 4.724, 17.344]

```

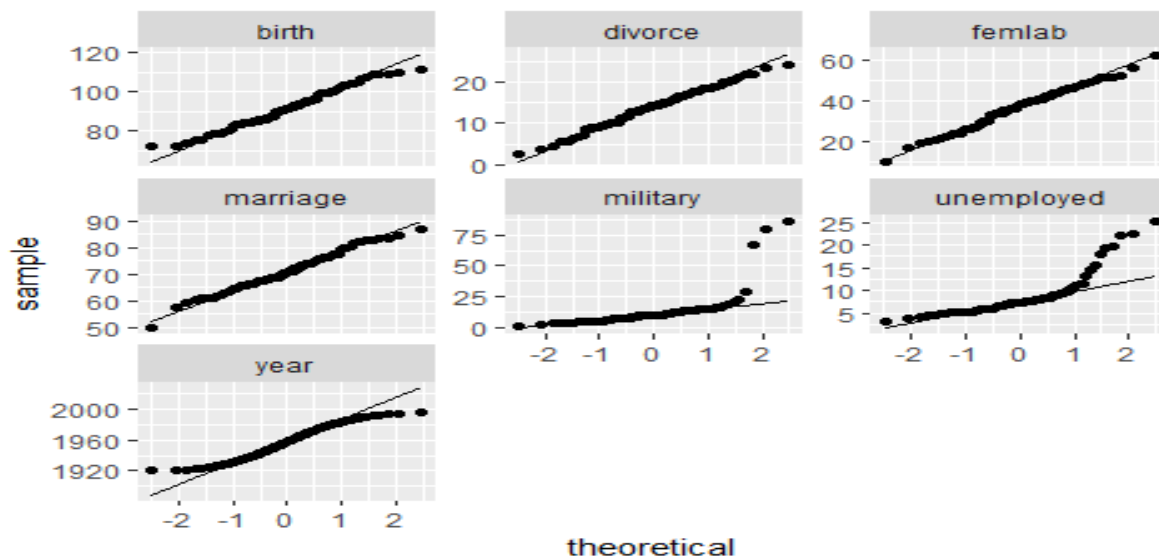
*#histogram,density and Q-Q plots*  
`plot_histogram(divusaB)`



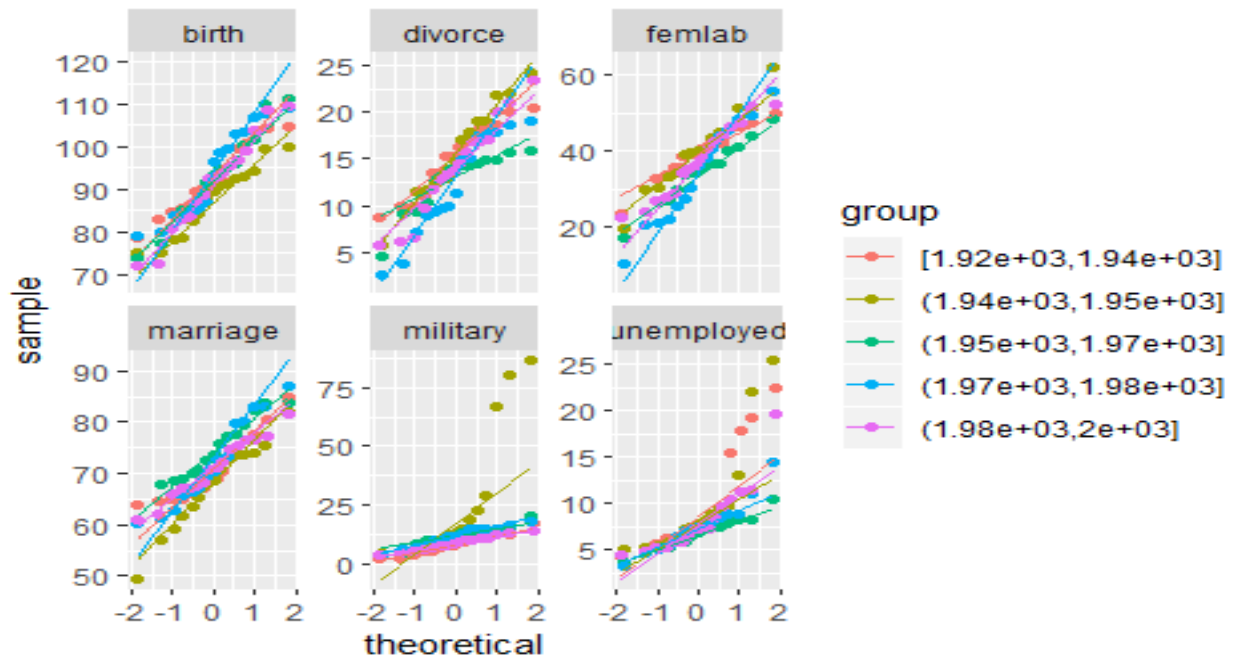
```
plot_density(divusaB)
```



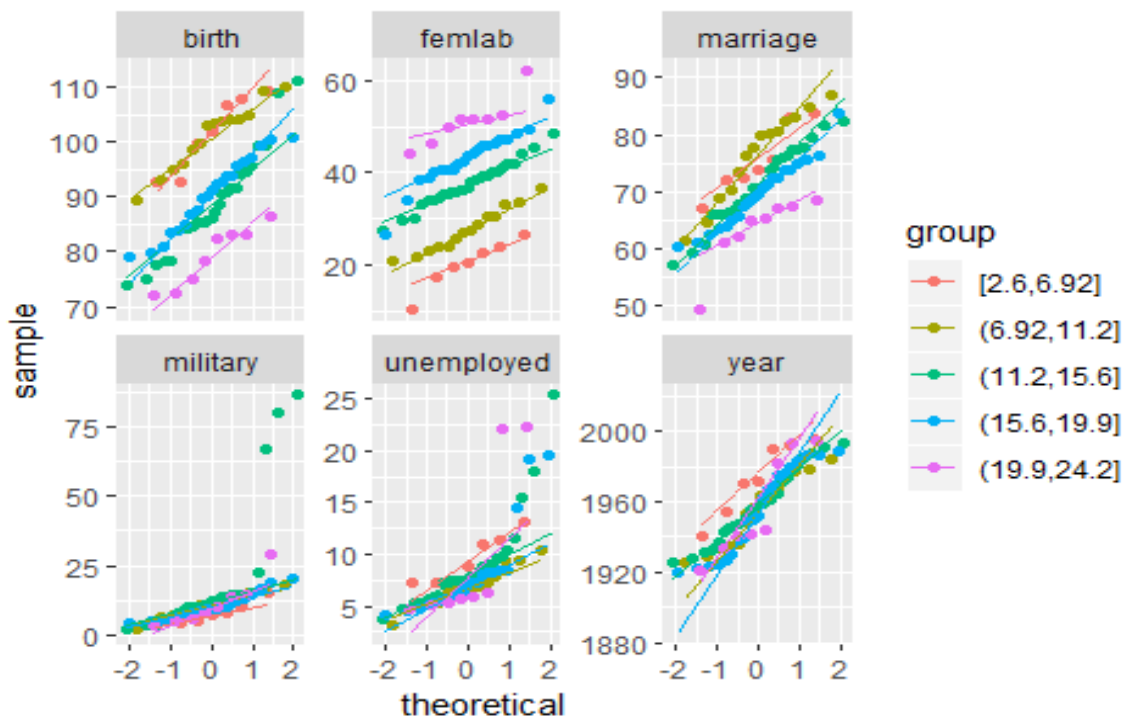
```
qq_data <- divusaB
plot_qq(qq_data) #plotting qq plot for qq_data
```



```
plot_qq(qq_data, by = "year")
```

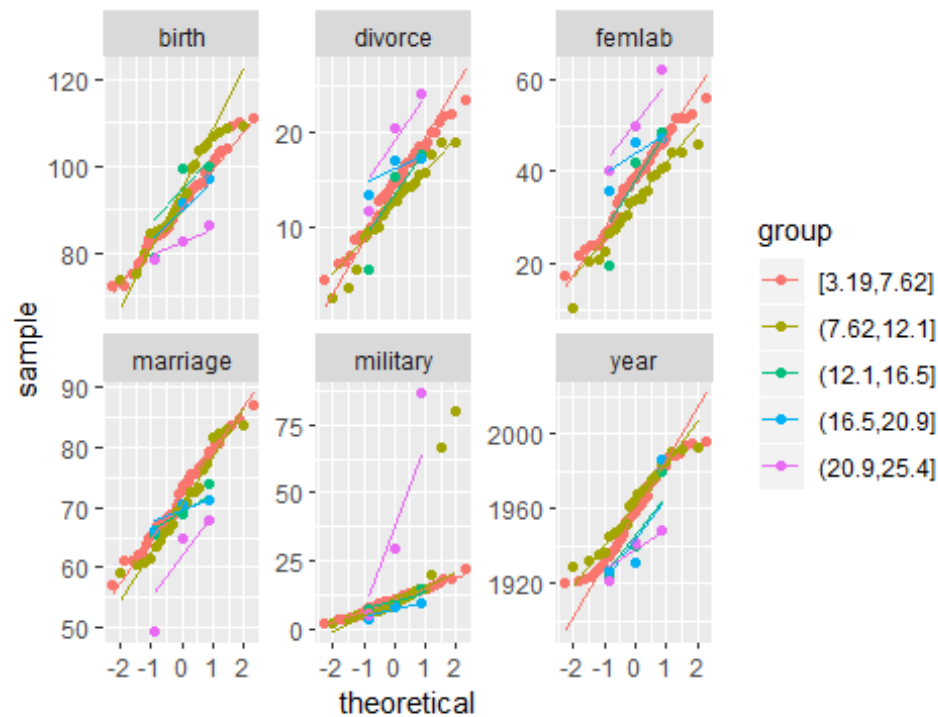


```
plot_qq(qq_data, by = "divorce")
```

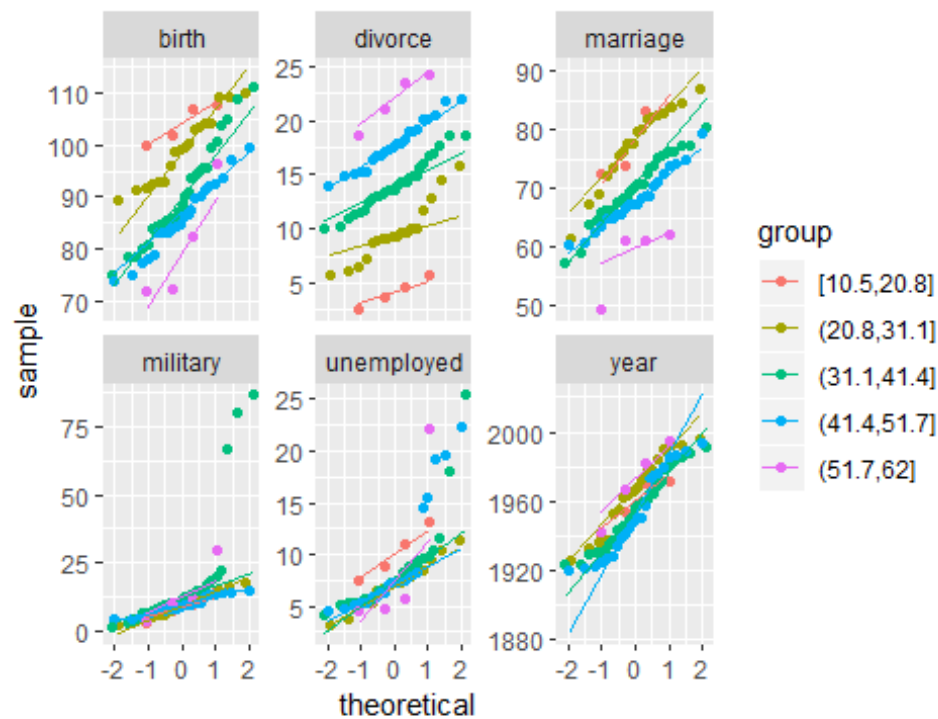


```
plot_qq(qq_data, by = "unemployed")
```

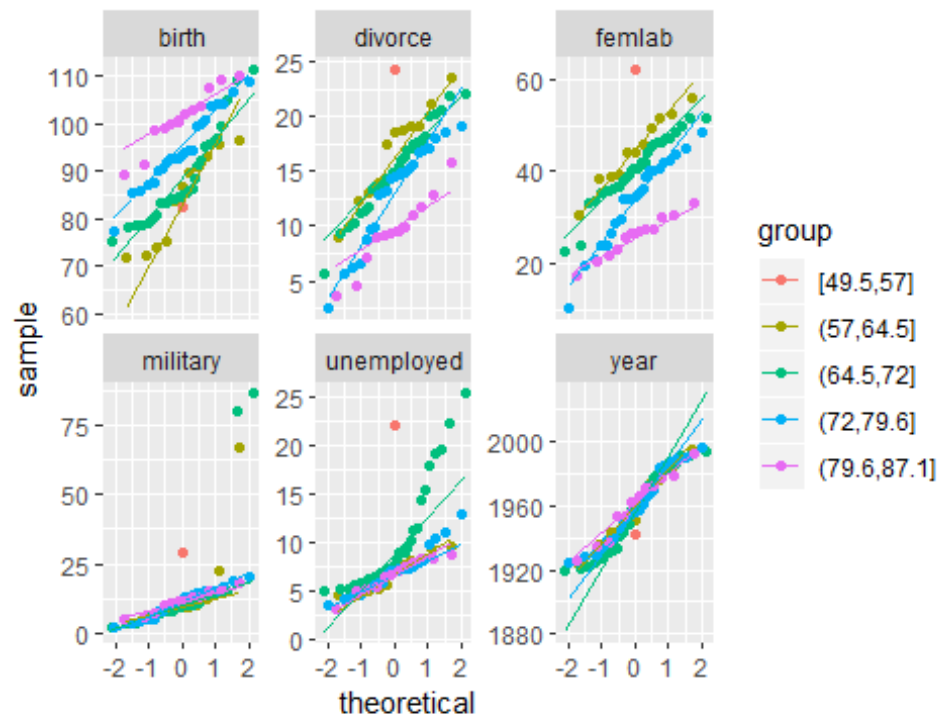




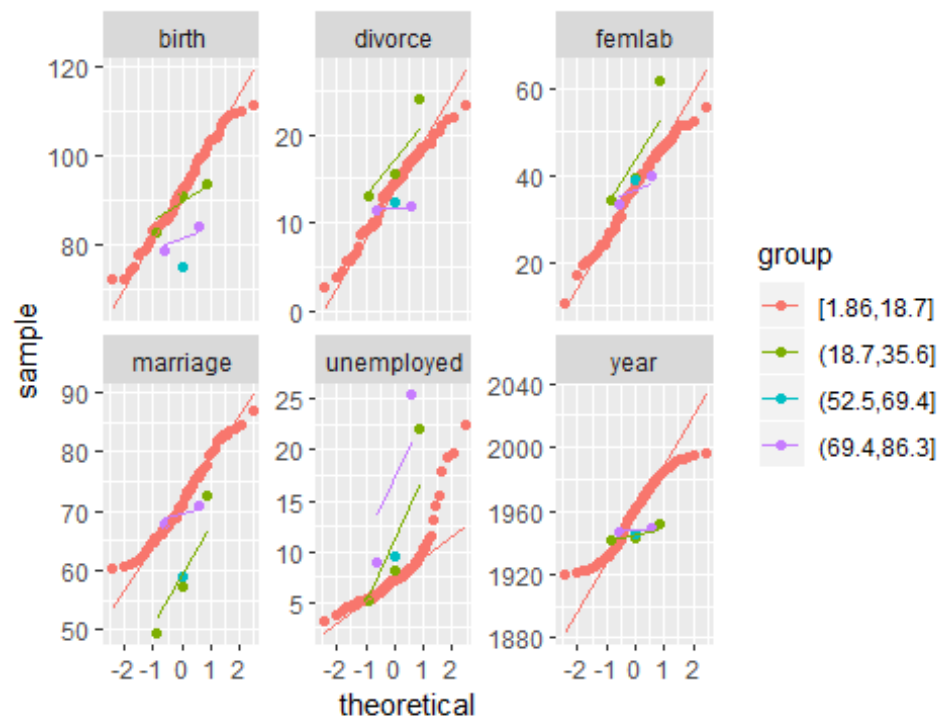
```
plot_qq(qq_data, by = "femlab")
```



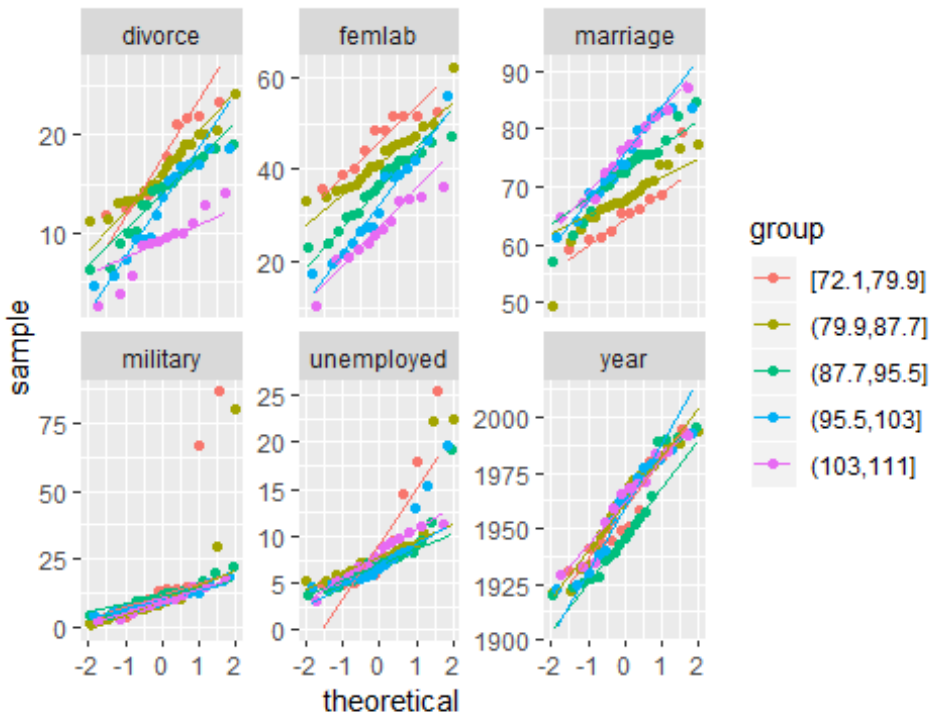
```
plot_qq(qq_data, by = "marriage")
```



```
plot_qq(qq_data, by = "military")
```

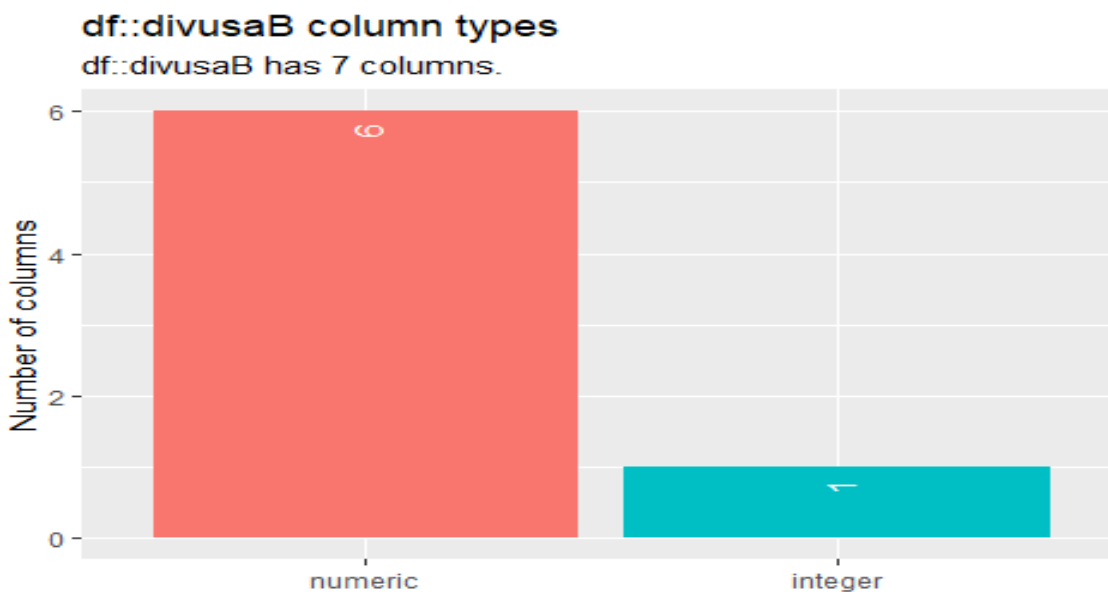


```
plot_qq(qq_data, by = "birth")
```



# Gives information about the type,size, numerical data and pearson's correlation of dataset

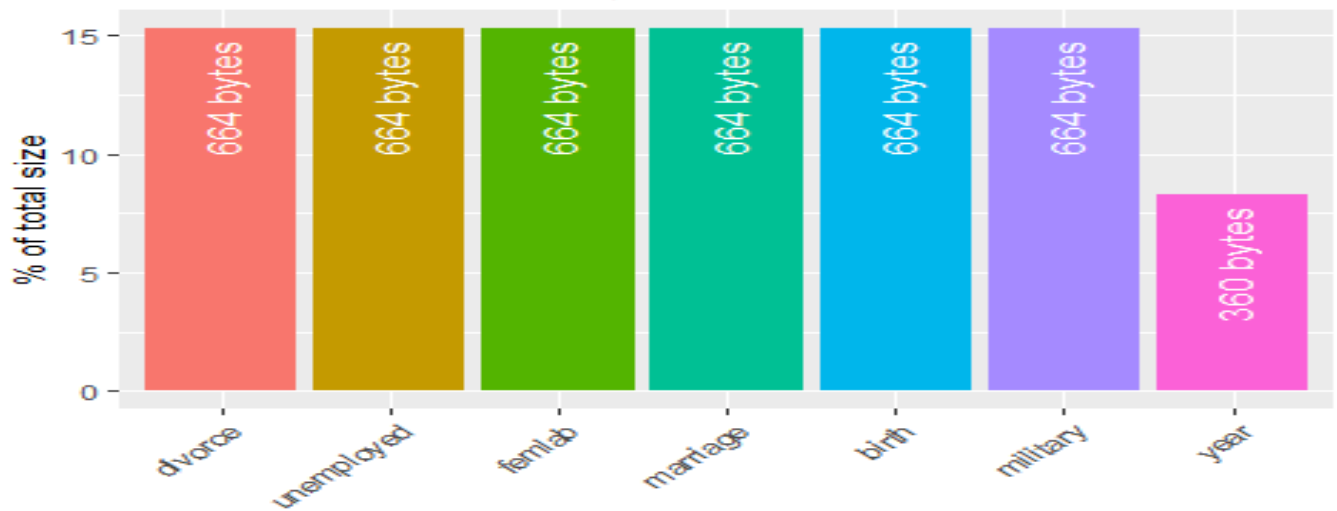
```
a<-inspect_types(divusaB)
show_plot(a)
```



```
b<-inspect_mem(divusaB)
show_plot(b)
```

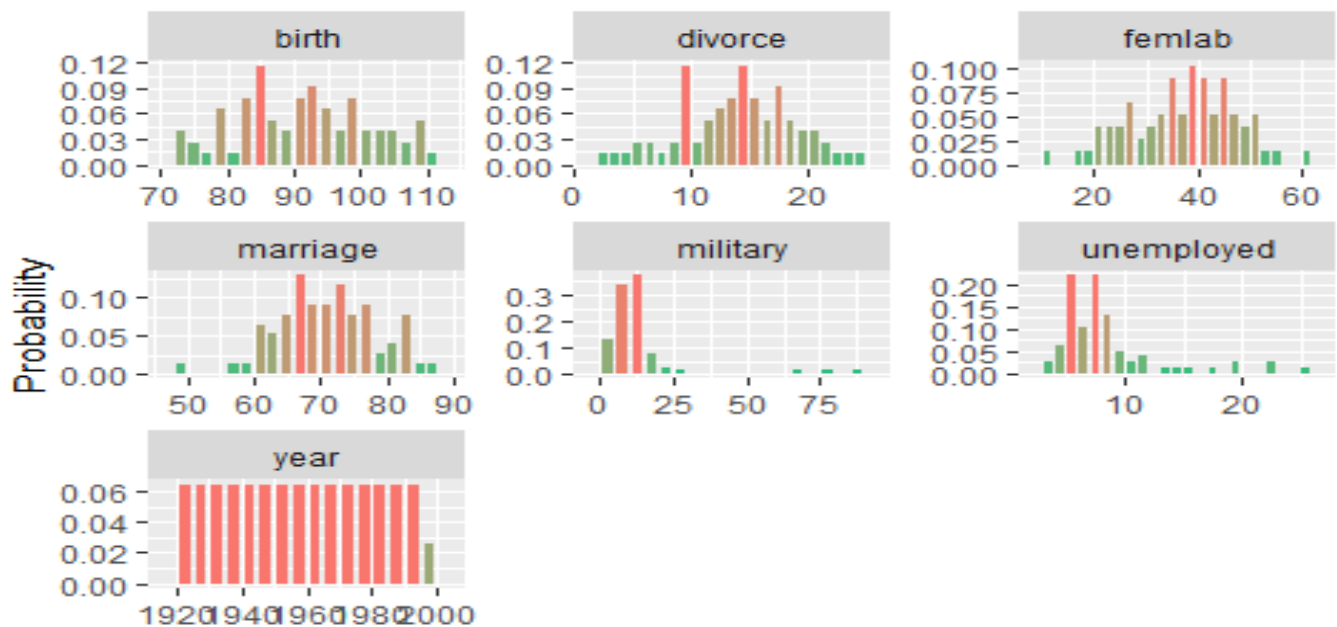
## Column sizes in df::divusaB

df::divusaB has 7 columns, 77 rows & total size of 5.37 Kb

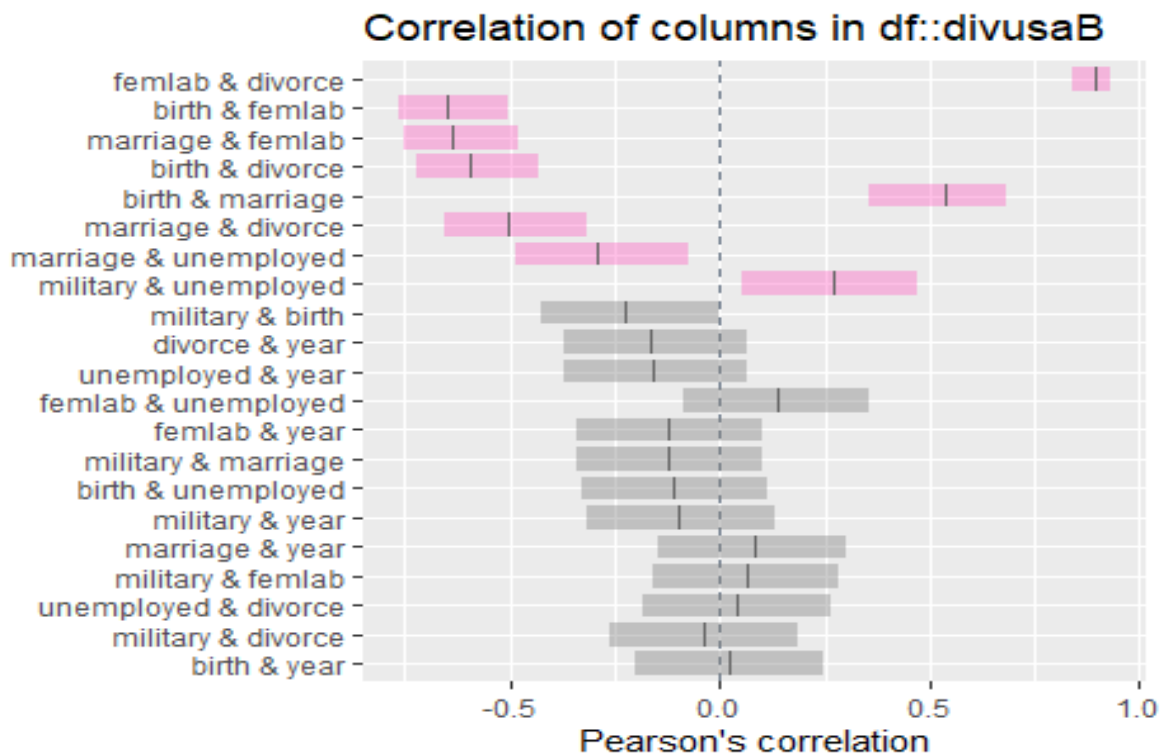


```
b2<-inspect_num(divusaB)
show_plot(b2)
```

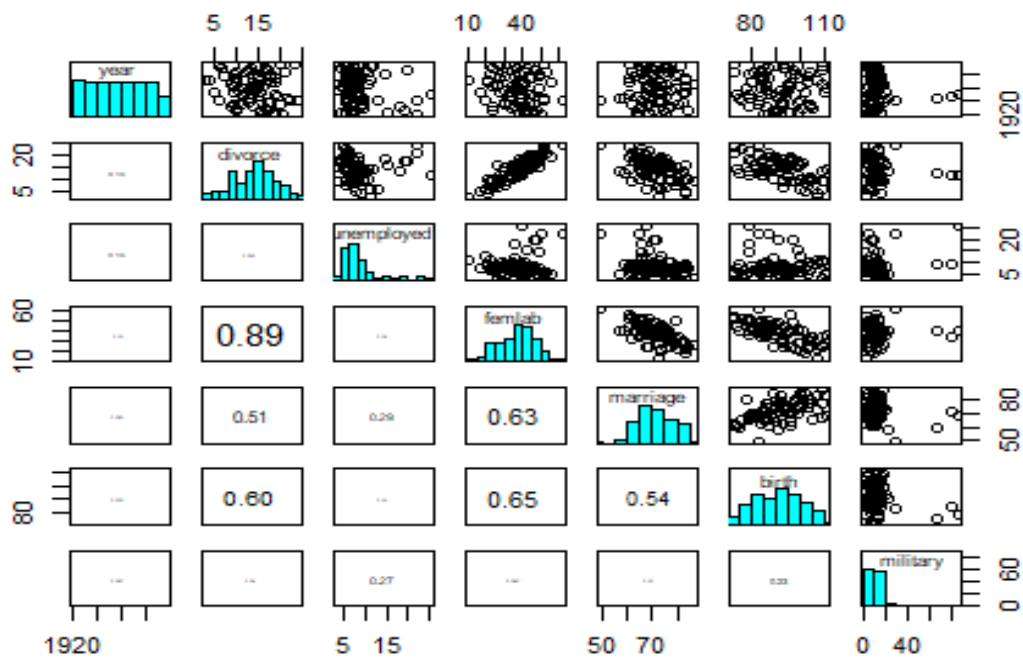
## Histograms of numeric columns in df::divusaB



```
b5<-inspect_cor(divusaB)
show_plot(b5)
```



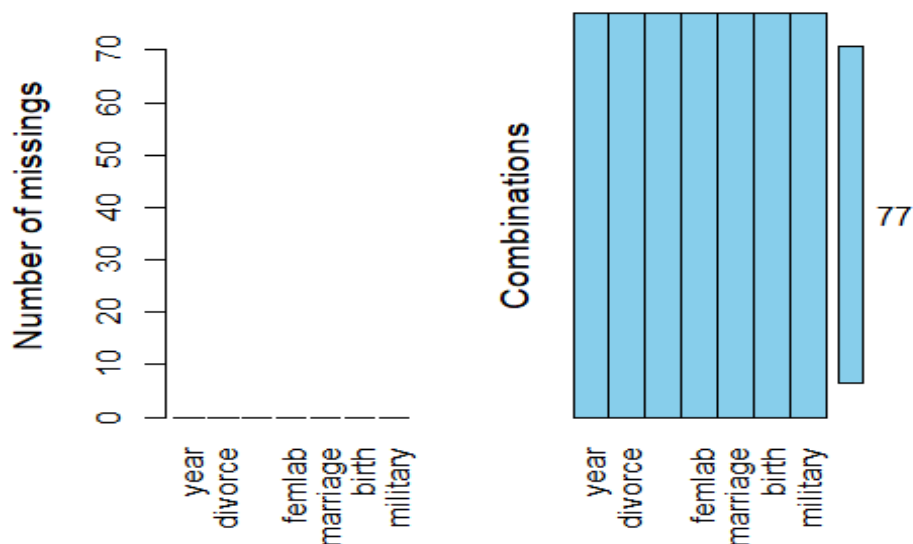
```
## Pairs plot
##function to put histograms on the diagonal
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y, use = "everything"))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste0(prefix, txt)
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}
# this function is taken from the help documentation, it will create
#histograms along the diagonal
panel.hist <- function(x, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5) )
  h <- hist(x, plot = FALSE)
  breaks <- h$breaks; nB <- length(breaks)
  y <- h$counts; y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}
# plot scatter plots for variables 1:7
#windows(10,10)
pairs(divusaB, lower.panel=panel.cor, diag.panel = panel.hist)
```



(ii) Cleaning your dataset Looking in to the missing data

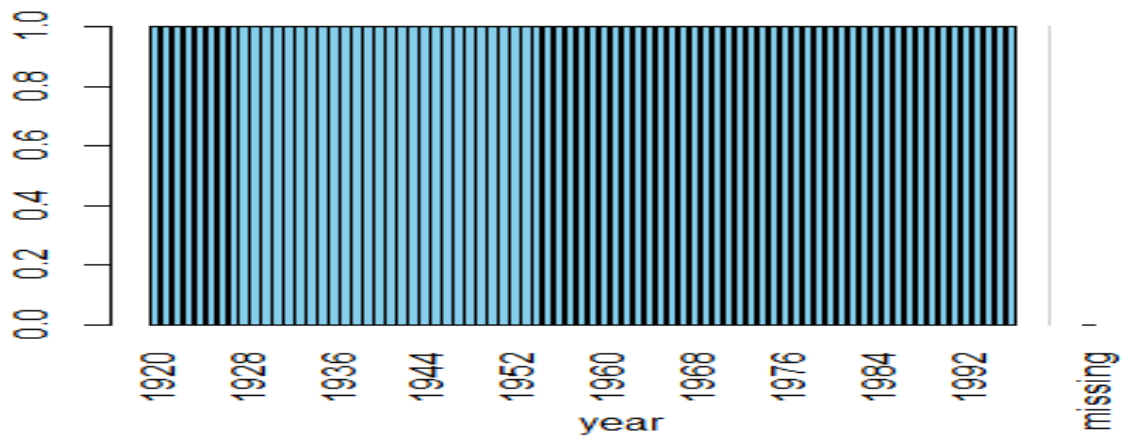
*#shows the amount of missing data for each variable and the frequency of combinations of missing values*

```
aggr(divusaB, prop = FALSE, number=TRUE)
```

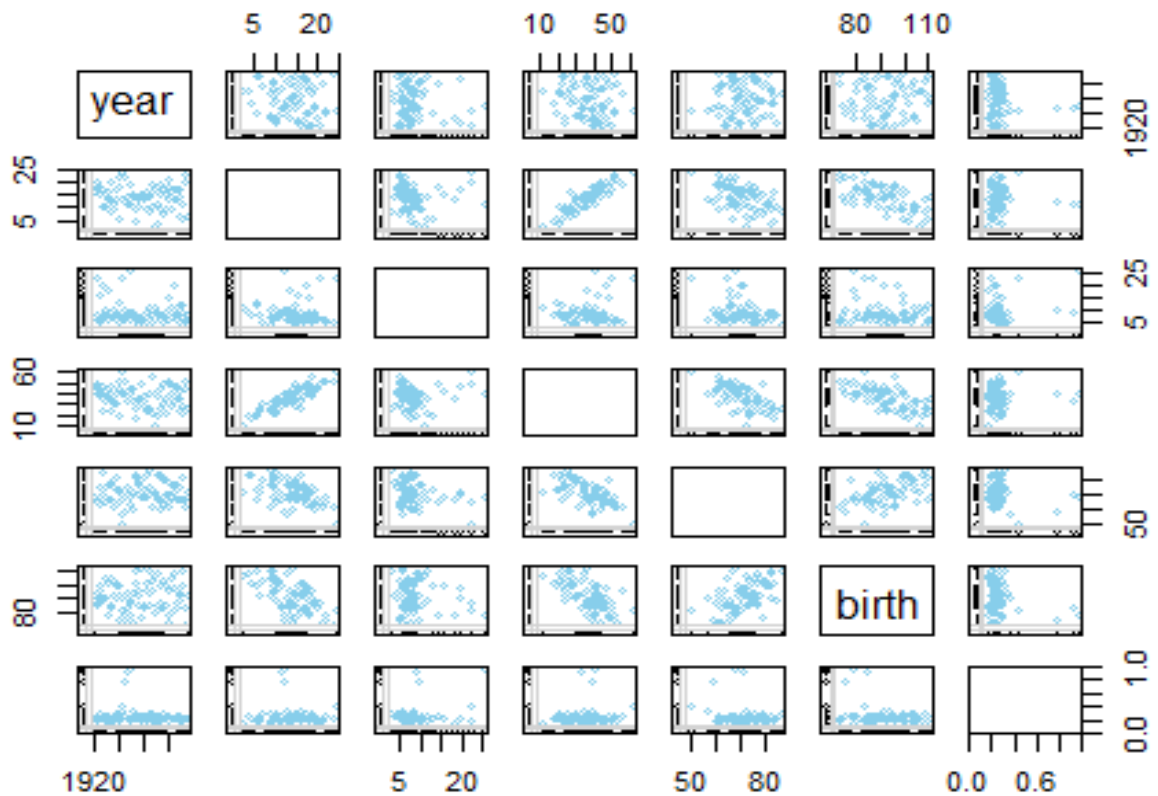


*#creates a barchart showing the values that are missing*

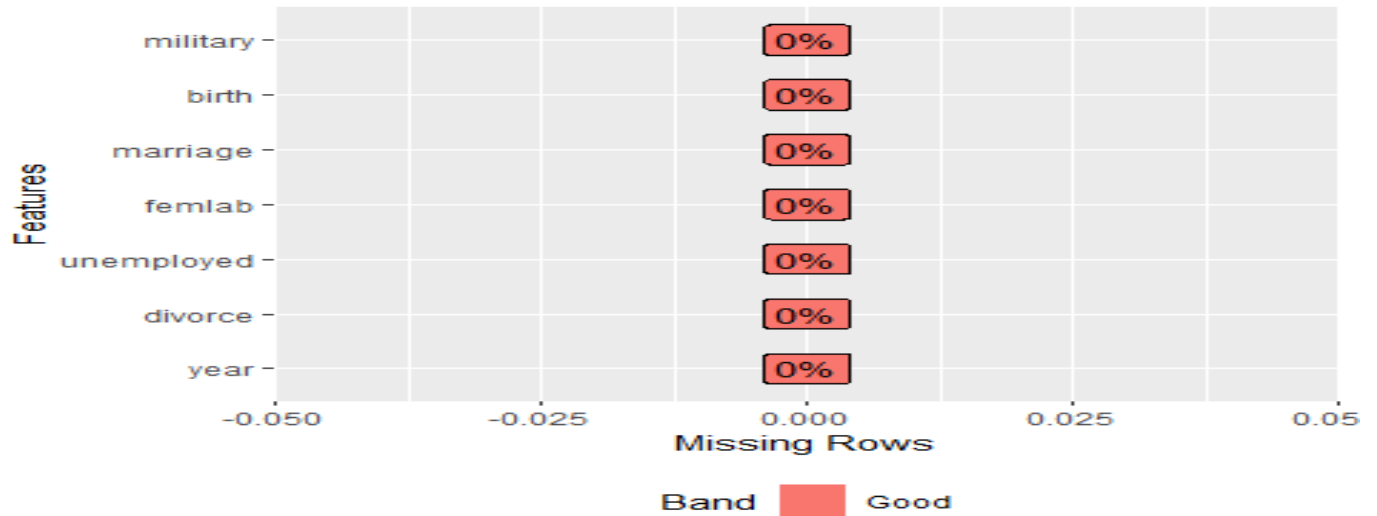
```
barMiss(divusaB)
```



*#Next we create a margin plot*  
`marginmatrix(divusaB)`



`plot_missing(divusaB)`



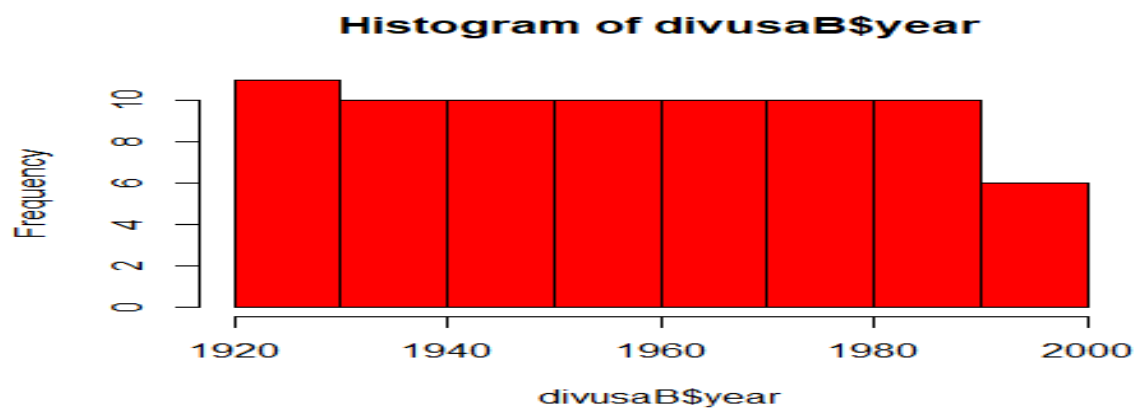
```
profile_missing(divusaB)
```

```
##      feature num_missing pct_missing
## 1      year           0           0
## 2    divorce           0           0
## 3 unemployed           0           0
## 4     femlab           0           0
## 5   marriage           0           0
## 6     birth           0           0
## 7   military           0           0
```

THERE IS NO MISSING DATA so there is no need to clean the data

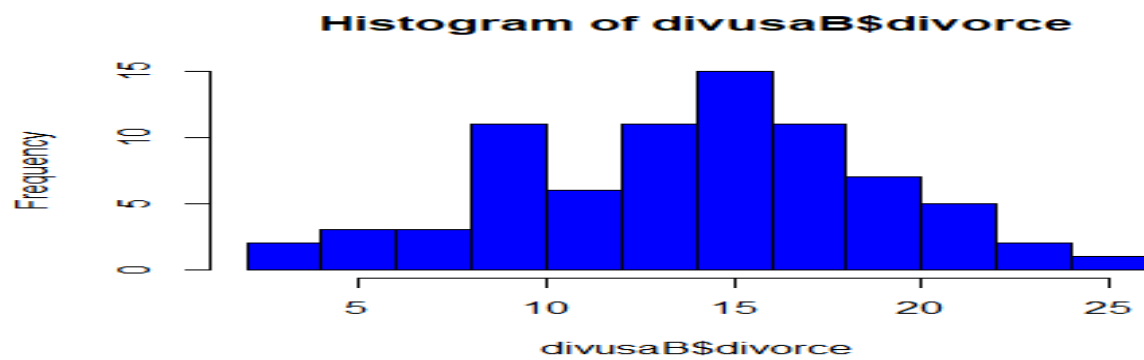
(iii) Analyzing relationships between variables (graphical data Analysis)

```
#PLOTting THE PLoTs
#GRAPHICAL SUMMARY
#Include: boxplots, histograms, scatterplot and the correlation coefficient.
#histograms
hist(divusaB$year, col="red")
```

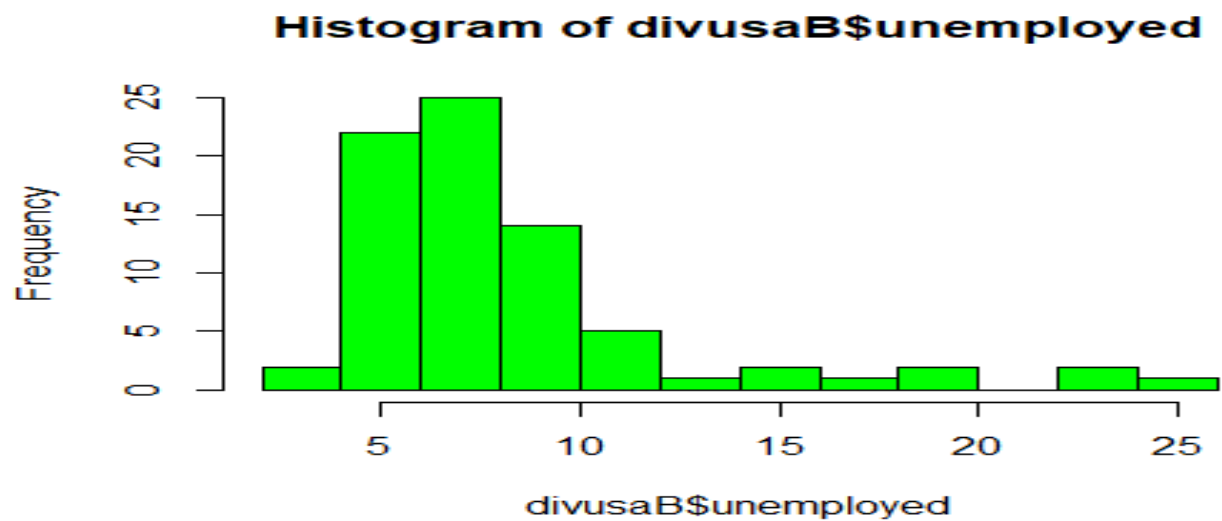




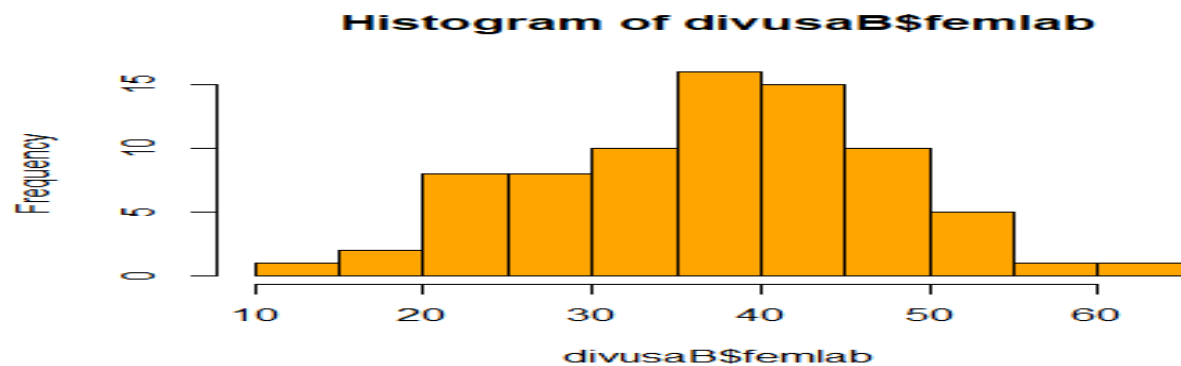
```
hist(divusaB$divorce,col="blue")
```



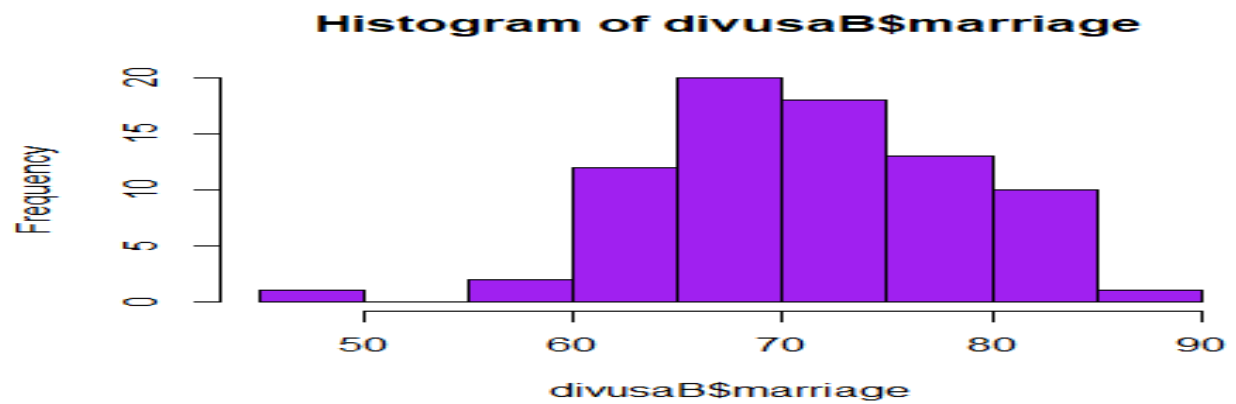
```
hist(divusaB$unemployed,col="green")
```



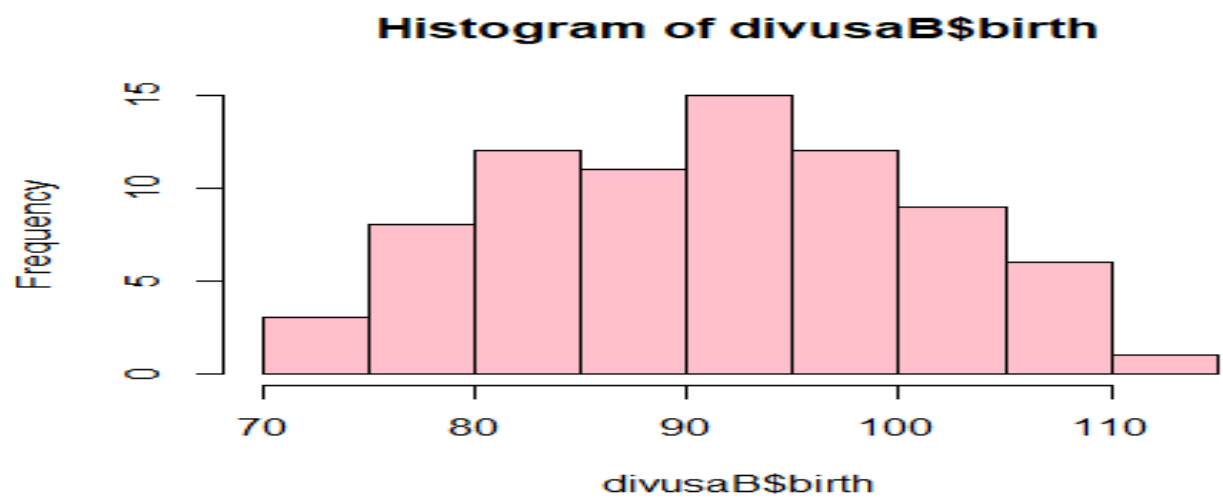
```
hist(divusaB$femlab,col="orange")
```



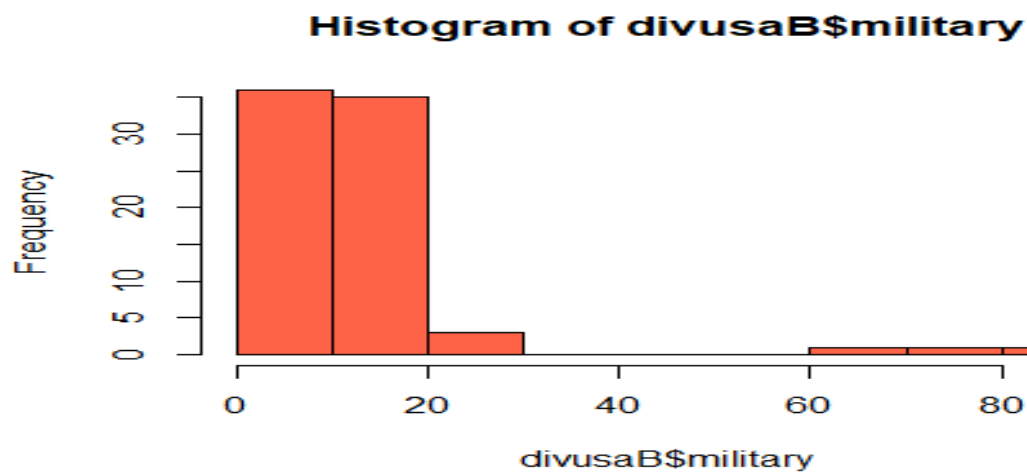
```
hist(divusaB$marriage,col="purple")
```



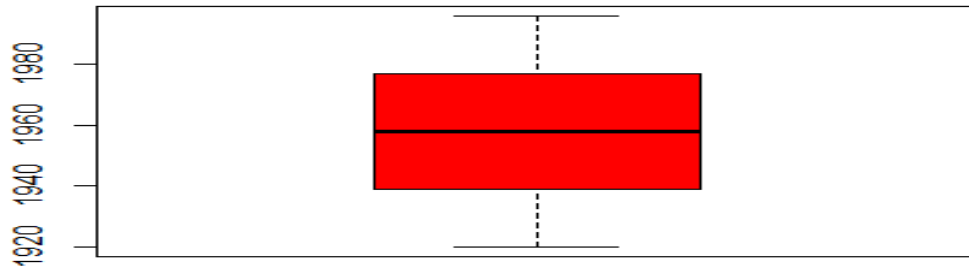
```
hist(divusaB$birth,col="pink")
```



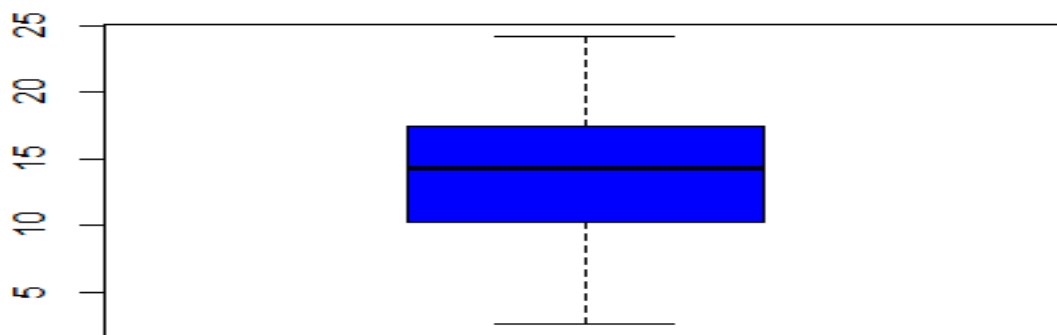
```
hist(divusaB$military,col="tomato")
```



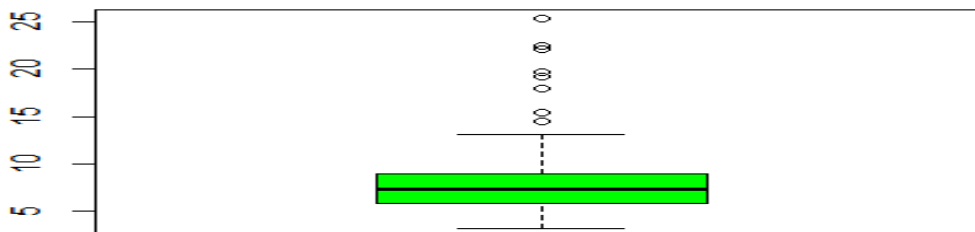
```
#boxplots  
boxplot(divusaB$year,col="red")
```



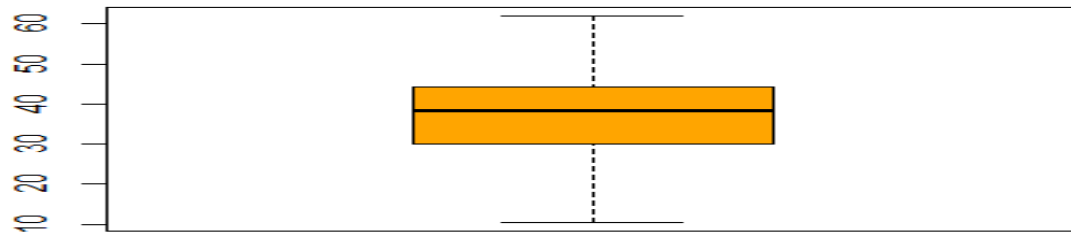
```
boxplot(divusaB$divorce,col="blue")
```



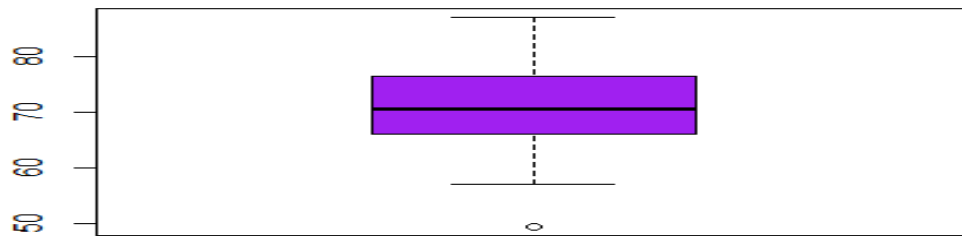
```
boxplot(divusaB$unemployed,col="green")
```



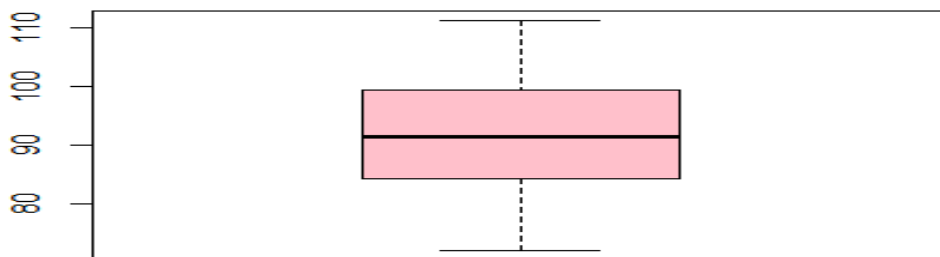
```
boxplot(divusaB$femlab,col="orange")
```



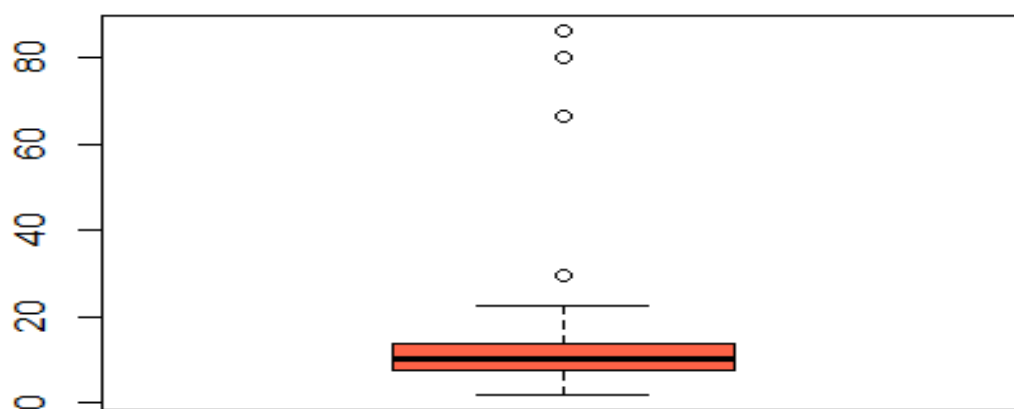
```
boxplot(divusaB$marriage,col="purple")
```



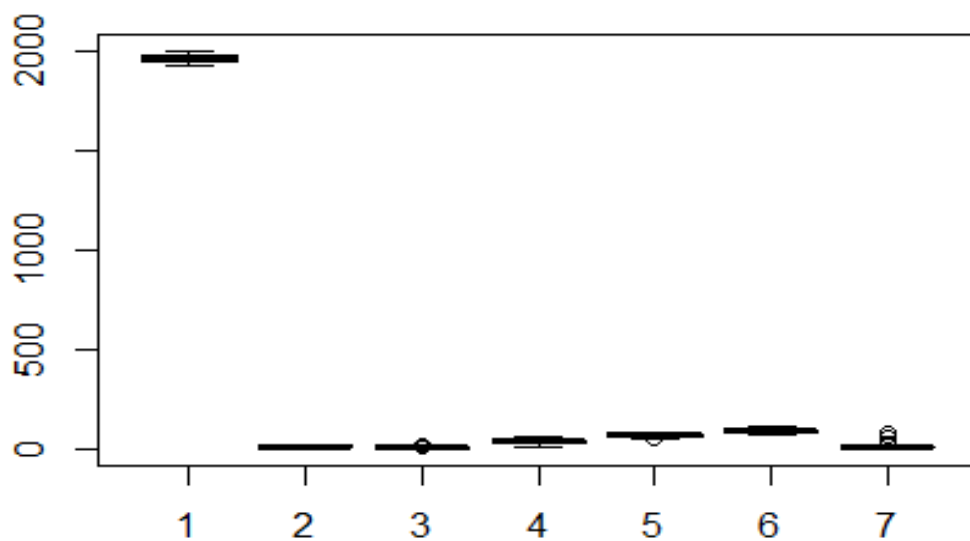
```
boxplot(divusaB$birth,col="pink")
```



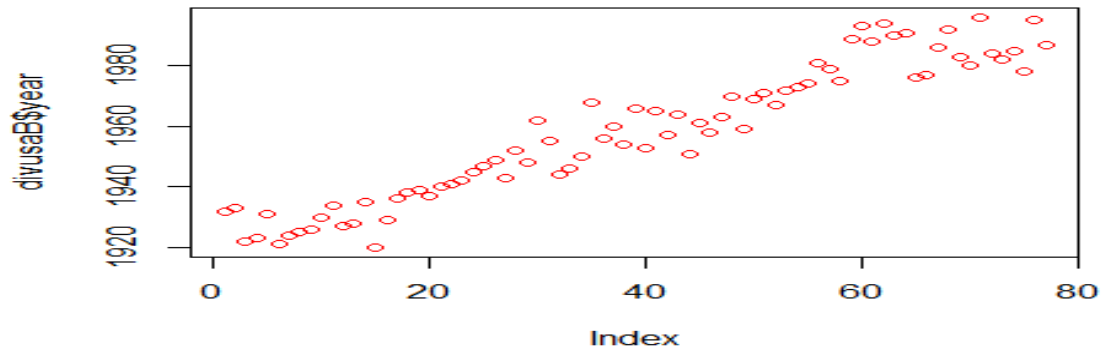
```
boxplot(divusaB$military,col="tomato")
```



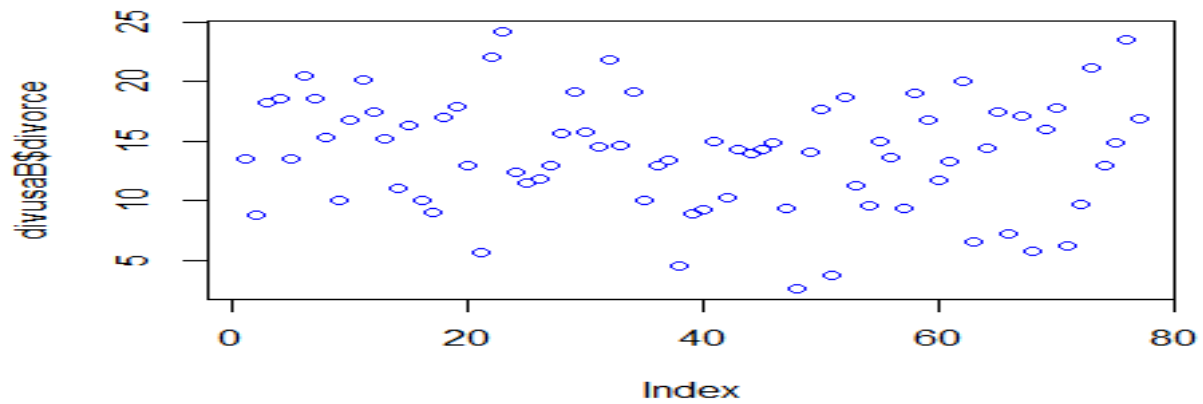
```
boxplot(divusaB$year,divusaB$divorce,divusaB$unemployed,divusaB$femlab,divusaB$marriage,divusaB$birth,divusaB$military,col=terrain.colors(7))
```



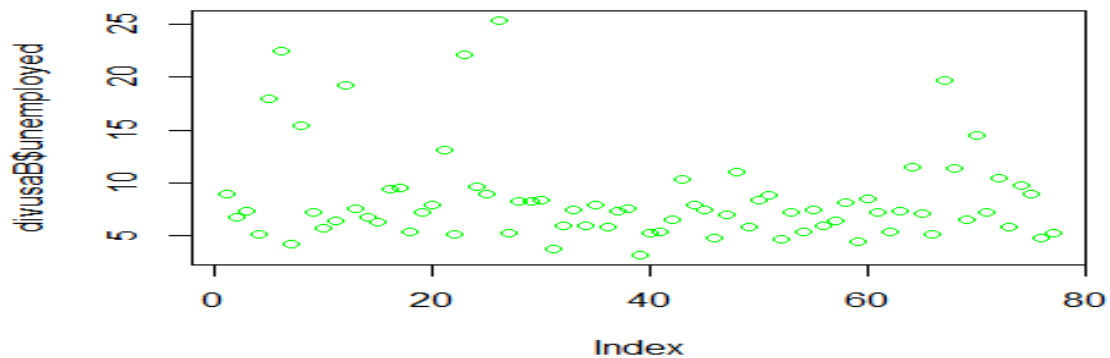
```
#scatterplots  
plot(divusaB$year,col="red")
```



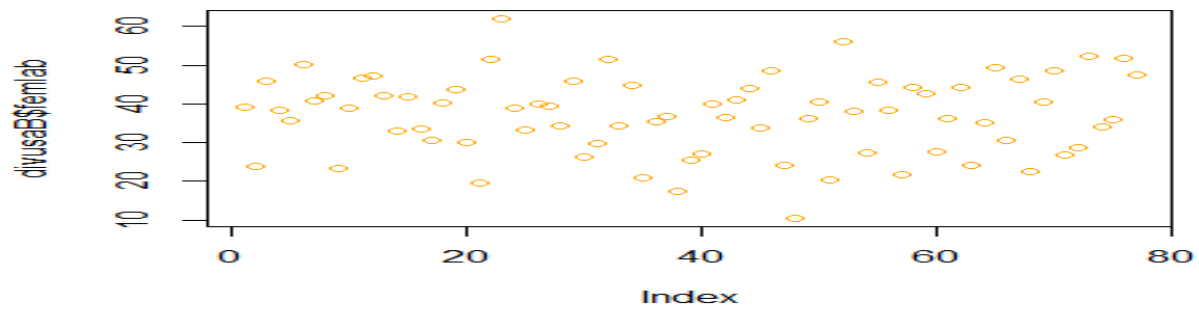
```
plot(divusaB$divorce,col="blue")
```



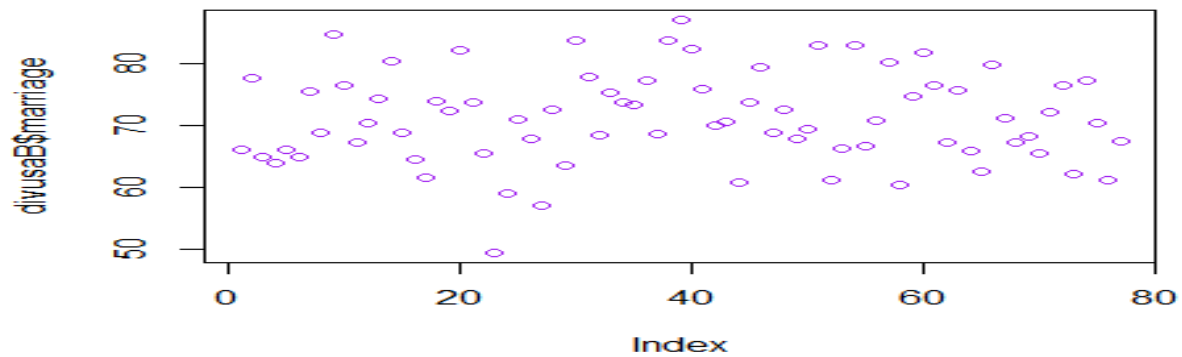
```
plot(divusaB$unemployed,col="green")
```



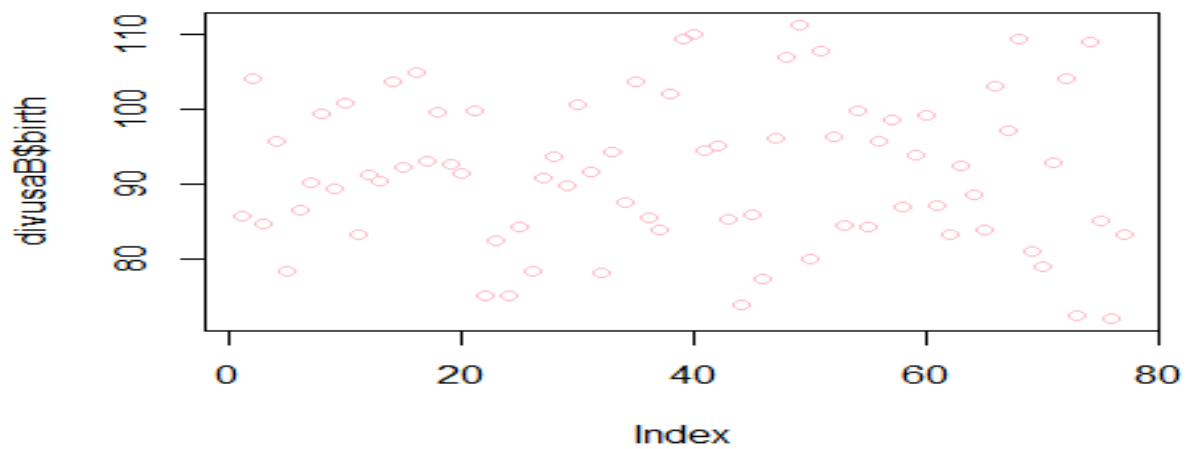
```
plot(divusaB$femlab,col="orange")
```



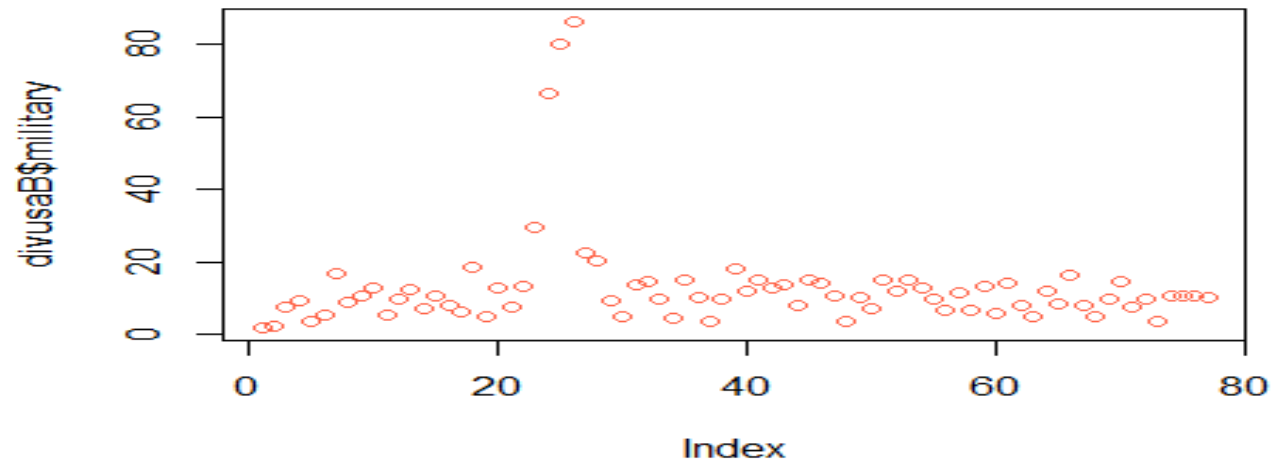
```
plot(divusaB$marriage,col="purple")
```



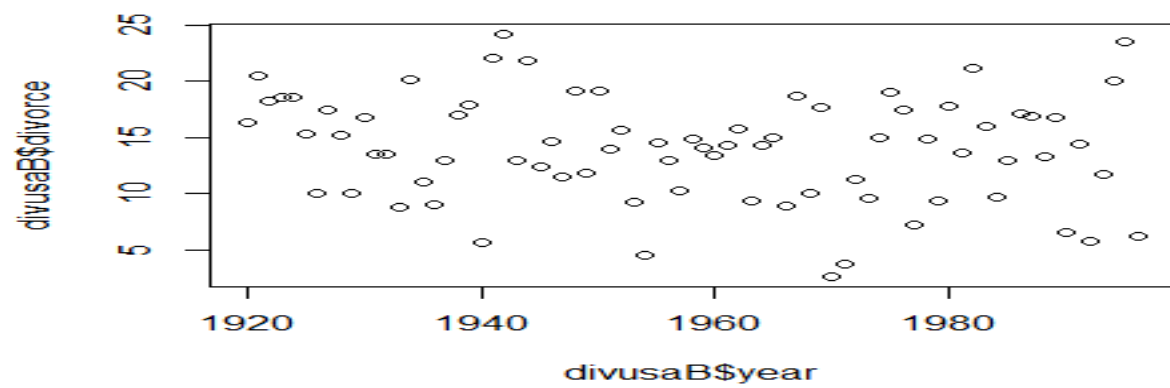
```
plot(divusaB$birth,col="pink")
```



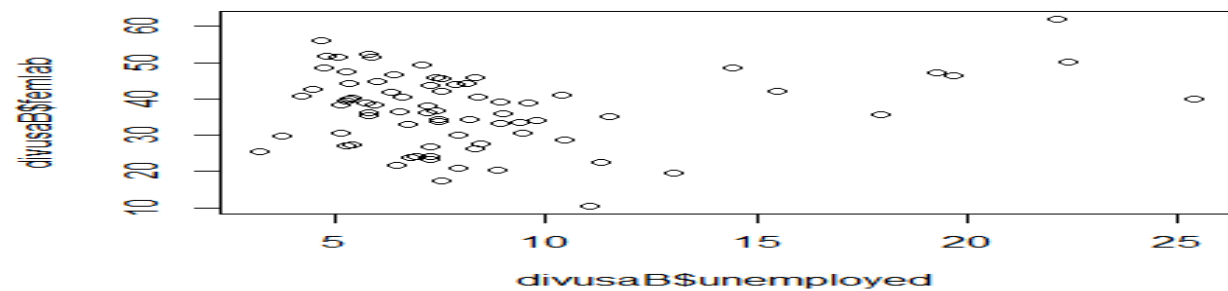
```
plot(divusaB$military,col="tomato")
```



```
plot(divusaB$year,divusaB$divorce)
```

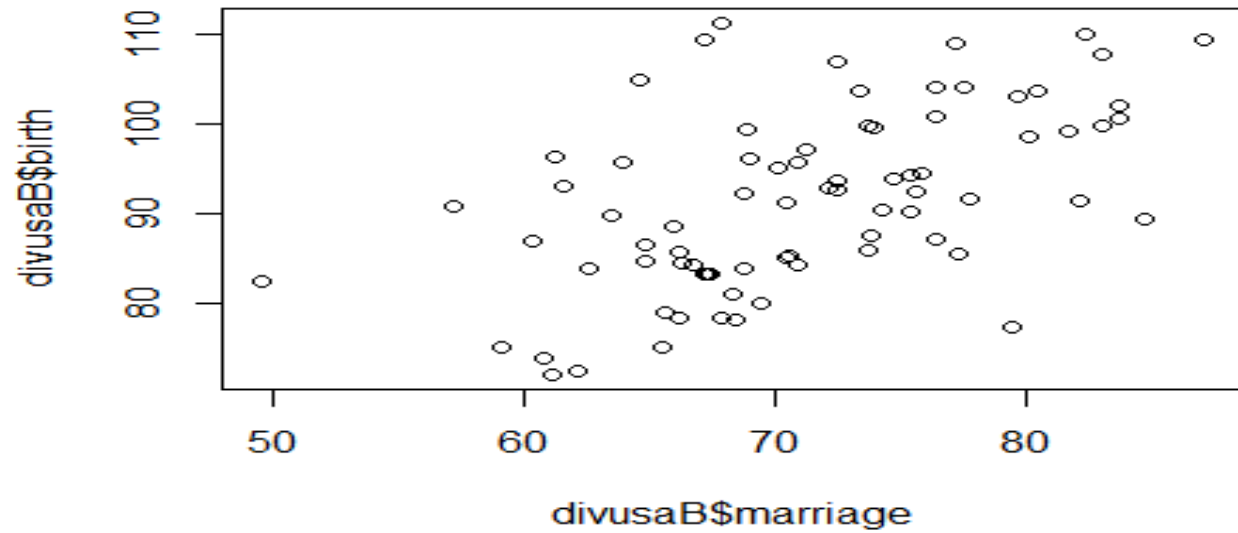


```
plot(divusaB$unemployed,divusaB$femlab)
```

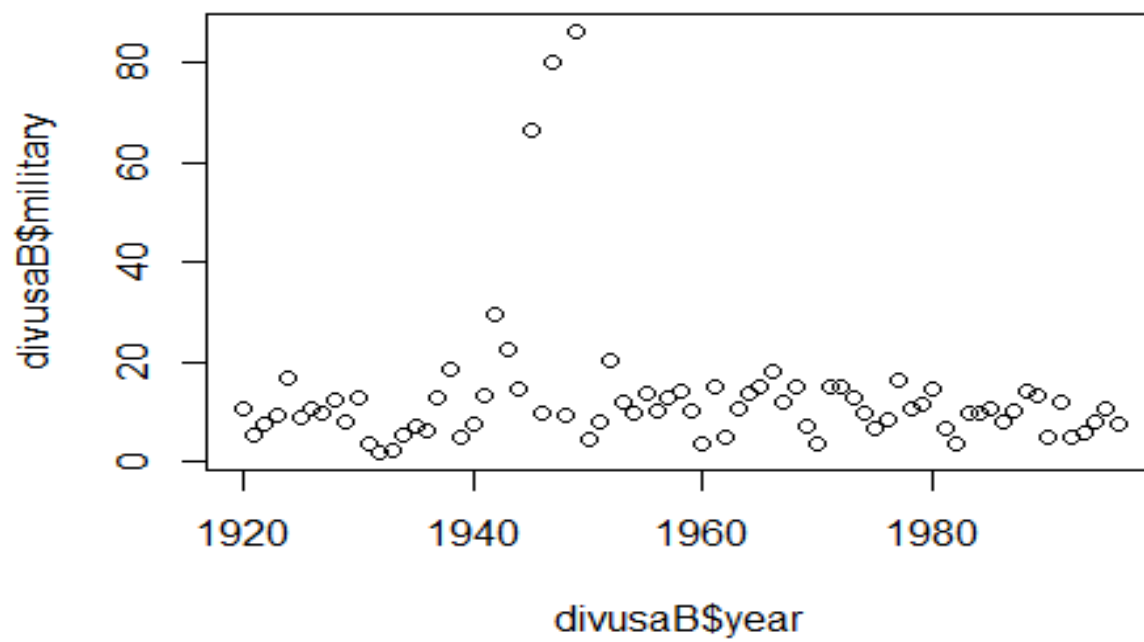




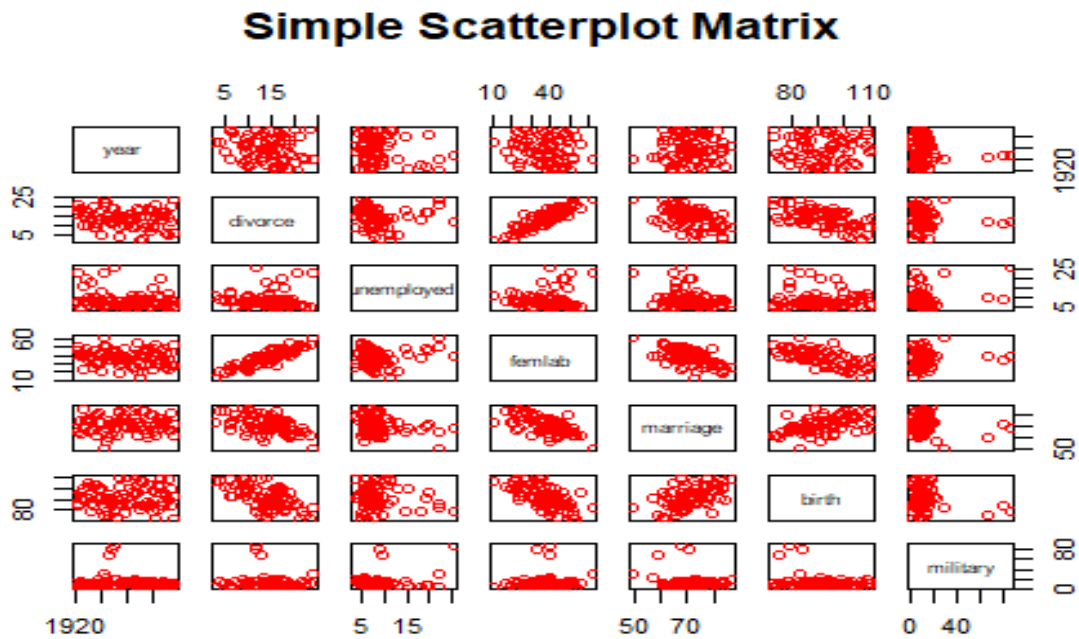
```
plot(divusaB$marriage,divusaB$birth)
```



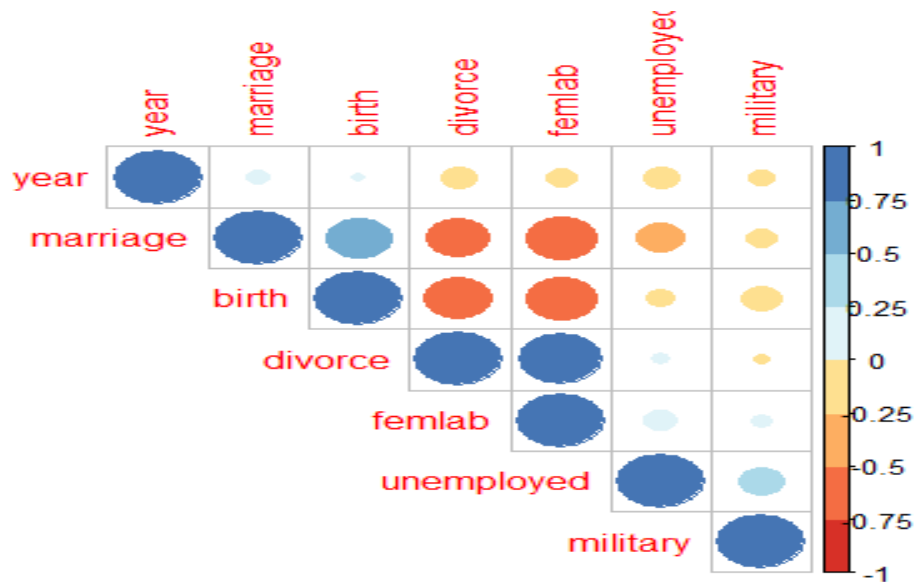
```
plot(divusaB$year,divusaB$military)
```



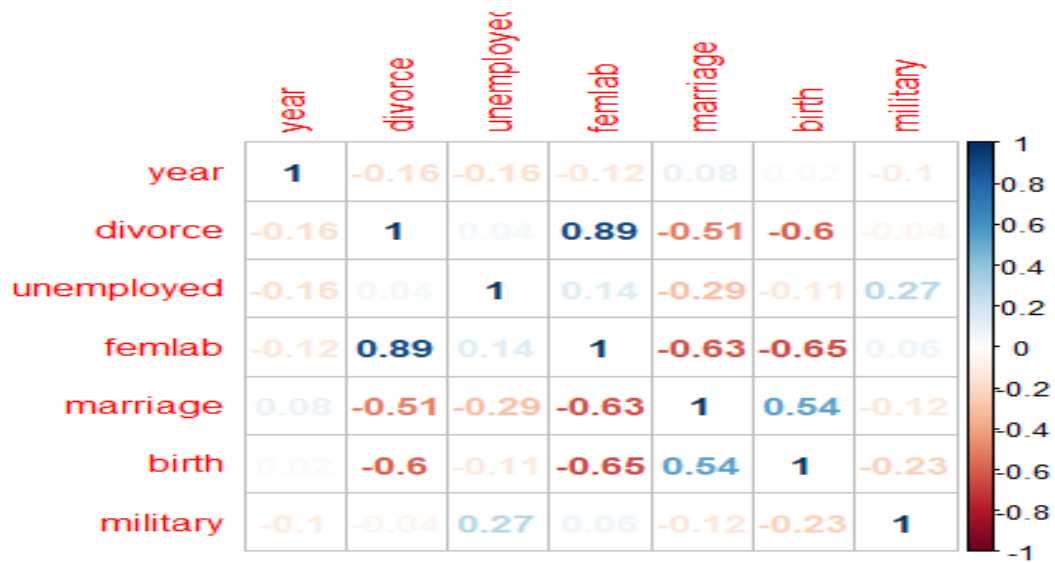
```
pairs(divusaB,main="Simple Scatterplot Matrix",col="red")
```



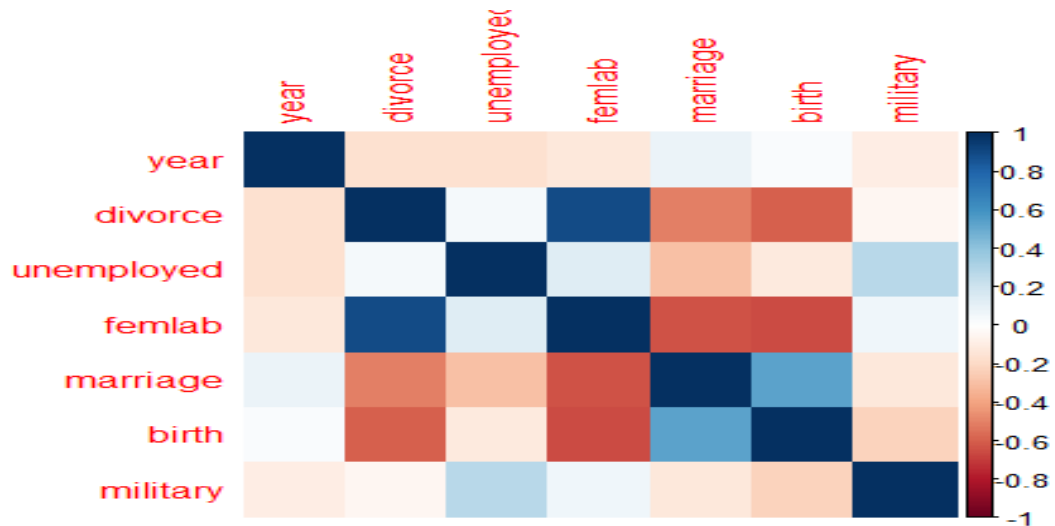
```
#corelation coefficient
M <-cor(divusaB)
corrplot(M, type="upper", order="hclust",
          col=brewer.pal(n=8, name="RdYlBu"))
```



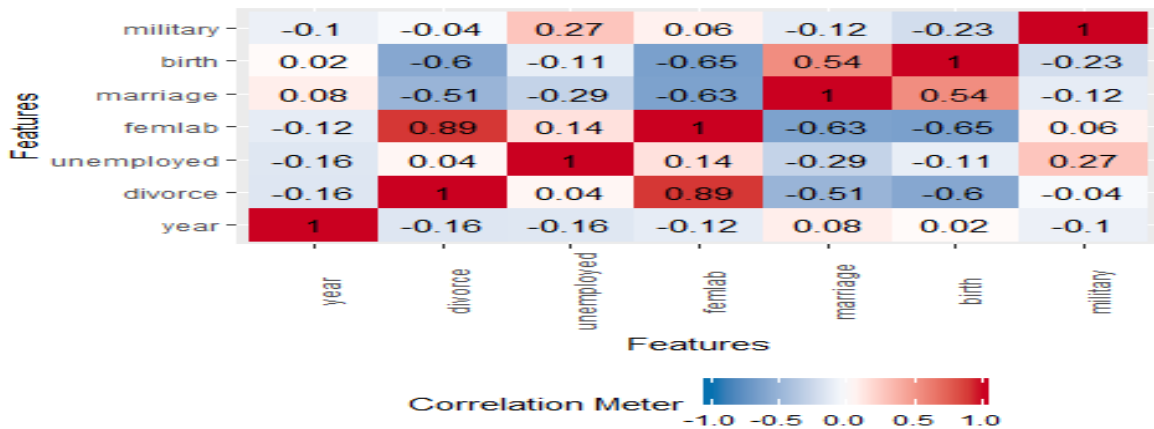
```
corrplot(M, method="number")
```



```
corrplot(M, method="color")
```



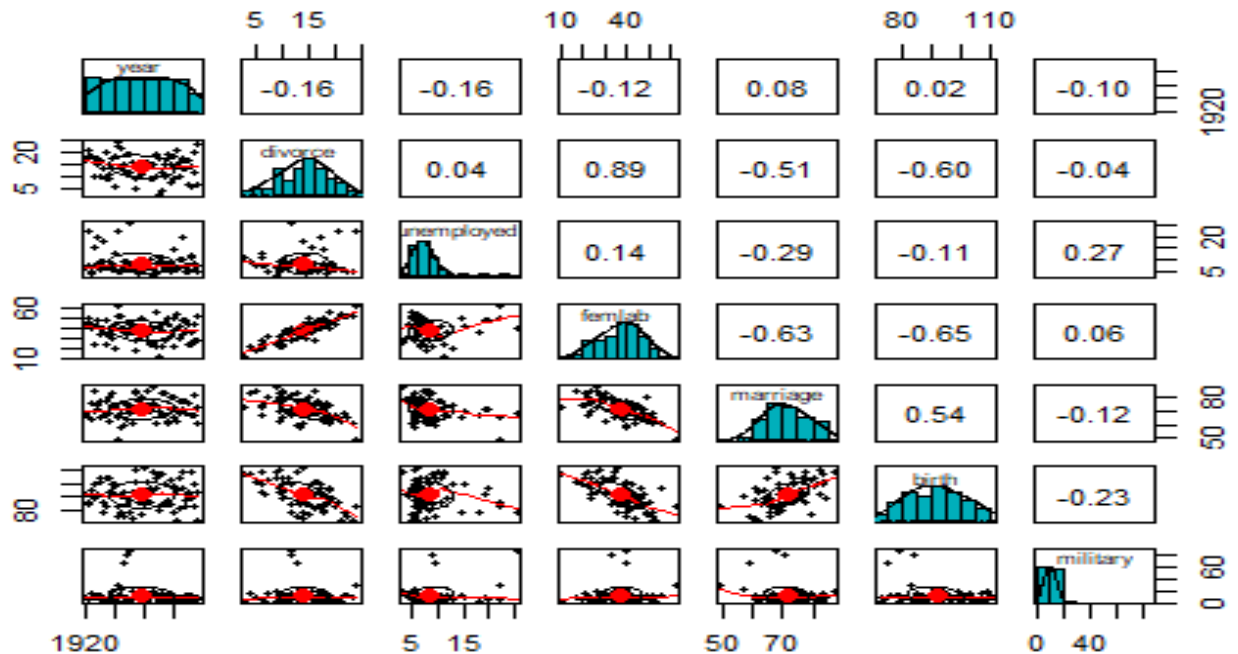
```
plot_correlation(divusaB)
```



```

pairs.panels(divusaB,
  method = "pearson", # correlation method
  hist.col = "#00AFBB",
  density = TRUE, # show density plots
  ellipses = TRUE) # show correlation ellipses

```



From the density graph I observed that the variables are less skewed. so there is no need to apply log transformation. But there are some outliers in the data so I will be applying grubbs.test to see whether the outliers are significant or not

*#its is a outlier which is not significant because the p value is greater than 0.05.*

```
grubbs.test(divusaB$year)
```

```
##
```

```
## Grubbs test for one outlier
```

```
##
```

```
## data: divusaB$year
```

```
## G = 1.69856, U = 0.96154, p-value = 1
```

```
## alternative hypothesis: highest value 1996 is an outlier
```

*#its is a outlier which is not significant because the p value is greater than 0.05.*

```
grubbs.test(divusaB$divorce)
```

```
##
```

```
## Grubbs test for one outlier
```

```
##
```

```

## data:  divusaB$divorce
## G = 2.41140, U = 0.92248, p-value = 0.5472
## alternative hypothesis: lowest value 2.603 is an outlier

#its is a outlier which is significant because the p value is less than 0.05.
grubbs.test(divusaB$unemployed)

##
## Grubbs test for one outlier
##
## data:  divusaB$unemployed
## G = 3.84982, U = 0.80242, p-value = 0.001973
## alternative hypothesis: highest value 25.365 is an outlier

#its is a outlier which is not significant because the p value is greater than 0.05.
grubbs.test(divusaB$femlab)

##
## Grubbs test for one outlier
##
## data:  divusaB$femlab
## G = 2.66419, U = 0.90538, p-value = 0.2503
## alternative hypothesis: lowest value 10.513 is an outlier

#its is a outlier which is not significant because the p value is greater than 0.05.
grubbs.test(divusaB$marriage)

##
## Grubbs test for one outlier
##
## data:  divusaB$marriage
## G = 2.9511, U = 0.8839, p-value = 0.09343
## alternative hypothesis: lowest value 49.486 is an outlier

#its is a outlier which is not significant because the p value is greater than 0.05.
grubbs.test(divusaB$birth)

##
## Grubbs test for one outlier
##
## data:  divusaB$birth
## G = 1.98010, U = 0.94773, p-value = 1
## alternative hypothesis: highest value 111.172 is an outlier

#its is a outlier which is significant because the p value is less than 0.05.
grubbs.test(divusaB$military)

##
## Grubbs test for one outlier

```

```
##  
## data:  divusaB$military  
## G = 5.22209, U = 0.63646, p-value = 2.513e-07  
## alternative hypothesis: highest value 86.275 is an outlier
```

This is about the numerical and graphical summary of the data

The data is symmetrical Distributed there is no high skewness so I did not apply log transformations for this dataset. The numerical summary of data is as follows: The mean of year=1958, divorce=13.97126,unemployed=8.453558,femlab=37.05366,marriage=71.1777,birth=91.54112,military=13.09644

The median of year=1958,divorce=14.302,unemployed=7.285,femlab=38.34,marriage=70.601,birth=91.472,military=10.3

The standard deviation of year=22.37186,divorce=4.714373,unemployed=4.392784,femlab=9.961989,marriage=7.350366,birth=9.914067,military=14.01326

The IQR of year=38,divorce=7.128,unemployed=3.15,femlab=14.11,marriage=10.257,birth=15.037,military=6.054

The Normality for year,From the output, the p-value < 0.05 implying that the distribution of the data are significantly different from normal distribution.

The Normality for divorce,From the output, the p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution.

The Normality for unemployed,From the output, the p-value , <0.05 implying that the distribution of the data are significantly different from normal distribution.

The Normality for femlab,From the output, the p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution.

The Normality for marriage,From the output, the p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution.

The Normality for birth,From the output, the p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution.

The Normality for military,From the output, the p-value < 0.05 implying that the distribution of the data are significantly different from normal distribution.

## Question 2(b)

Fit the model :  $y = \beta_0 + \beta_1 \text{unemployed} + \beta_2 \text{femlab} + \beta_3 \text{marriage} + \beta_4 \text{birth} + \beta_5 \text{military} + e$

## Solution

In this question I was asked to build a multiple linear regression model to fit the model.

Simple linear regression is an extension of multiple regression. It is used anytime we have to estimate a variable's value dependent on two or more other variables. The variable that we want to measure is called the dependent variable (or sometimes the predictor of outcome, target or criterion).

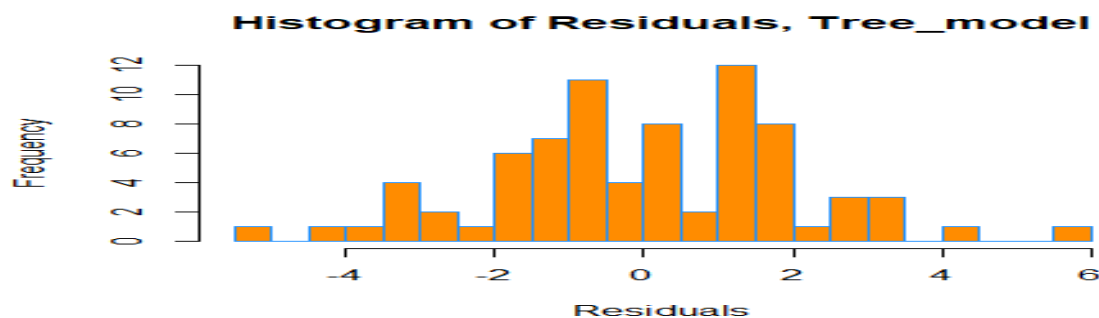
The equation of multiple linear regression is  $y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik} + e_i$  where  $y_i$  represents the  $i$ th observation of the response variable and  $x_{ji}$  represents the  $i$ th observation of the  $j$ th explanatory variable.

```
#fit the model
m1 <- lm(divorce ~ unemployed + femlab + marriage + birth + military, data =
divusaB)
m1

##
## Call:
## lm(formula = divorce ~ unemployed + femlab + marriage + birth +
##     military, data = divusaB)
##
## Coefficients:
## (Intercept)    unemployed      femlab    marriage      birth    milita
ry
##   -2.34394      -0.04380      0.43393      0.05947     -0.03510     -0.031
52
```

The histogram for the Tree\_model residual

```
hist(resid(m1),
     xlab = "Residuals",
     main = "Histogram of Residuals, Tree_model",
     col = "darkorange",
     border = "dodgerblue",
     breaks = 20)
```



To create a summary of the fitted model

```
## summary output
```

```
summary(m1)
```

```
##
## Call:
## lm(formula = divorce ~ unemployed + femlab + marriage + birth +
##     military, data = divusaB)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.2328 -1.2259  0.0223  1.4396  5.7199
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.34394     5.08225  -0.461   0.6461
## unemployed  -0.04380     0.05880  -0.745   0.4587
## femlab       0.43393     0.03545  12.242 <2e-16 ***
## marriage     0.05947     0.04445   1.338   0.1852
## birth       -0.03510     0.03337  -1.052   0.2965
## military    -0.03152     0.01820  -1.732   0.0877 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.074 on 71 degrees of freedom
## Multiple R-squared:  0.8193, Adjusted R-squared:  0.8065
## F-statistic: 64.37 on 5 and 71 DF, p-value: < 2.2e-16
```

The fitted model is  $\hat{y} = -2.34394 - 0.04380X_1 + 0.43393X_2 + 0.05947X_3 - 0.03510X_4 - 0.03152X_5$   
The coefficient of determination  $R^2 = 0.8193$  which tells us that 81.93% of the variability in vol can be explained by Diam. The estimate for the error variance  $\sigma^2 = (2.074)^2$  It is possible to extract features of the model such as the model coefficients and the residuals:

```
## extract coefficients and residuals
```

```
m1$coefficients
```

```
## (Intercept) unemployed      femlab      marriage      birth      military
## -2.34393626 -0.04380484  0.43392905  0.05946538 -0.03509919 -0.03151601
```

```
m1$residuals
```

```
##           1           2           3           4           5           6
## -1.59141228  0.07955643  0.26007010  4.28577521  0.08024084  1.44127412
##           7           8           9          10          11          12
##  2.63384948 -0.22982286  0.99456962  1.78893888  1.63872630 -0.63556260
##          13          14          15          16          17          18
## -1.23981050 -1.62860814  0.26715394 -1.76403755 -1.65956426  1.84748314
##          19          20          21          22          23          24
##  0.73678975  1.31154471 -0.57914709  1.43959869  1.48863890 -0.55627931
##          25          26          27          28          29          30
##  1.00980373 -0.69908706 -1.10277787  3.10709505  1.65134717  5.71987851
##          31          32          33          34          35          36
```



```
## 3.20027987 1.17586272 1.59545562 1.03966527 3.27012867 -1.07393922
##          37          38          39          40          41          42
## -0.96914606 -1.40769598 -0.48677553 -0.63945488 -0.57098896 -3.38489467
##          43          44          45          46          47          48
## -1.48802413 -3.22872621 1.38392634 -5.23283938 1.09946648 0.41327110
##          49          50          51          52          53          54
## 1.11626177 1.74597439 -3.07309734 -2.89978327 -3.10509074 -0.74946808
##          55          56          57          58          59          60
## -2.76071926 -1.02652266 1.51565923 2.23090290 0.15978299 1.23494057
##          61          62          63          64          65          66
## -0.83393249 2.62156251 -2.35972928 1.53046290 -1.79427481 -4.05885242
##          67          68          69          70          71          72
## -0.36858701 -1.22587633 0.02225011 -0.89904043 -3.56153851 -0.61551833
##          73          74          75          76          77
## -0.06776377 0.42067195 1.12362254 2.81518944 -1.92928264
```

To view the ANOVA table we use the `summary.aov()` function.

```
#summary ANOVA table
summary.aov(m1)

##          Df Sum Sq Mean Sq F value Pr(>F)
## unemployed  1    3.1      3.1    0.720 0.3990
## femlab      1 1359.8  1359.8  316.255 <2e-16 ***
## marriage    1    6.2      6.2    1.451 0.2324
## birth       1    1.8      1.8    0.426 0.5161
## military    1   12.9     12.9    2.999 0.0877 .
## Residuals   71   305.3      4.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Question 2b(i)

Interpret the coefficient for femlab.

## Solution

The coefficient value of femlab is 0.43393x2 Therefore for every unit change in x2 the value of y will be increasing by 0.43393 amount provided all the other variables are constant An increase in percent female participation in labour force aged 16+ will be increased by 0.43393 adjusting for all the other variables

## Question 2b(ii)

Calculate the variance inflation factors for this model and discuss their implications for collinearity in the model.

## Solution

The variance inflation factor calculates how much an independent variable's activity (variance) is impacted by its interaction/correlation with other independent variables or increased. Inflation variables with variance make a simple calculation of how much a variable contributes to the standard regression error.

Collinearity is a condition of the strong correlation between some of the independent variables. Collinearity tends to inflate the variance of at least one predicted coefficient of regression,  $\beta_j$ . That can cause the wrong sign to have at least some regression coefficients.

```
#variance inflation factors of model m1  
vif(m1)
```

```
## unemployed      femlab    marriage      birth    military  
##    1.179226    2.203993    1.886869    1.934978    1.149686
```

The VIFs lie between 1 and 2 indicating that collinearity is not having a large impact on the coefficient estimates for this model.

## Question 2b(iii)

By fitting alternative models, determine whether collinearity has an impact on the coefficient estimates.

## Solution

```
#Building different models  
m1 <- lm(divorce ~ unemployed + femlab , data = divusaB)  
summary(m1)  
  
##  
## Call:  
## lm(formula = divorce ~ unemployed + femlab, data = divusaB)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -4.3960 -1.2998  0.1313  1.5842  6.3567   
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)      
## (Intercept)  -1.15887    0.98826   -1.173    0.245      
## unemployed   -0.08970    0.05537   -1.620    0.109      
## femlab        0.42880    0.02442   17.562 <2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 2.1 on 74 degrees of freedom
```

```
## Multiple R-squared:  0.8069, Adjusted R-squared:  0.8016
## F-statistic: 154.6 on 2 and 74 DF,  p-value: < 2.2e-16

vif(m1)

## unemployed      femlab
##    1.019846    1.019846

m11<- lm(divorce ~ unemployed + femlab + marriage , data = divusaB)
summary(m11)

##
## Call:
## lm(formula = divorce ~ unemployed + femlab + marriage, data = divusaB)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.0270 -1.2794 -0.1113  1.4633  5.9506
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.90386    4.09827  -1.441   0.154
## unemployed  -0.07153    0.05728  -1.249   0.216
## femlab       0.45219    0.03126  14.464 <2e-16 ***
## marriage     0.05233    0.04387   1.193   0.237
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.094 on 73 degrees of freedom
## Multiple R-squared:  0.8106, Adjusted R-squared:  0.8028
## F-statistic: 104.1 on 3 and 73 DF,  p-value: < 2.2e-16

vif(m11)

## unemployed      femlab      marriage
##    1.097518    1.681636    1.802716

m111 <- lm(divorce ~ unemployed + femlab + marriage + birth, data = divusaB)
summary(m111)

##
## Call:
## lm(formula = divorce ~ unemployed + femlab + marriage + birth,
##      data = divusaB)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.247 -1.318 -0.022  1.441  5.946
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.00884    5.05928  -0.792   0.431
```

```
## unemployed -0.07046    0.05753 -1.225    0.225
## femlab      0.44127    0.03568 12.369 <2e-16 ***
## marriage    0.05844    0.04506  1.297    0.199
## birth      -0.02114    0.03283 -0.644    0.522
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.102 on 72 degrees of freedom
## Multiple R-squared:  0.8116, Adjusted R-squared:  0.8012
## F-statistic: 77.56 on 4 and 72 DF,  p-value: < 2.2e-16

vif(m111)

## unemployed      femlab      marriage      birth
##    1.098441    2.172455    1.886536    1.822018
```

For all the three models The VIFs lie between 1 and 2 indicating that collinearity is not having a large impact on the coefficient estimates for this model.

## Question 2b(iv)

Create a partial regression plot to examine relationship between birth and divorce adjusted for unemployed, femlab, marriage and military.

## Solution

A partial regression plot show the effect of adding another variable to a model already having one or more independent variables.

The steps for creating a partial regression plot to show the relationship between birth and divorce adjusted for unemployed, femlab, marriage and military. are shown below.

Calculate the residuals for the regression model of divorce ~

unemployed+femlab+marriage+military Calculate the residuals for the regression model of

birth ~ unemployed+femlab+marriage+military Plot the residuals of divorce ~

unemployed+femlab+marriage+military (y-axis) against the residuals of birth ~

unemployed+femlab+marriage+military (x-axis).

We can fit a regression line to the two sets of residuals, the slope of the regression line measures the effect of birth and divorce adjusted for unemployed, femlab, marriage and military.

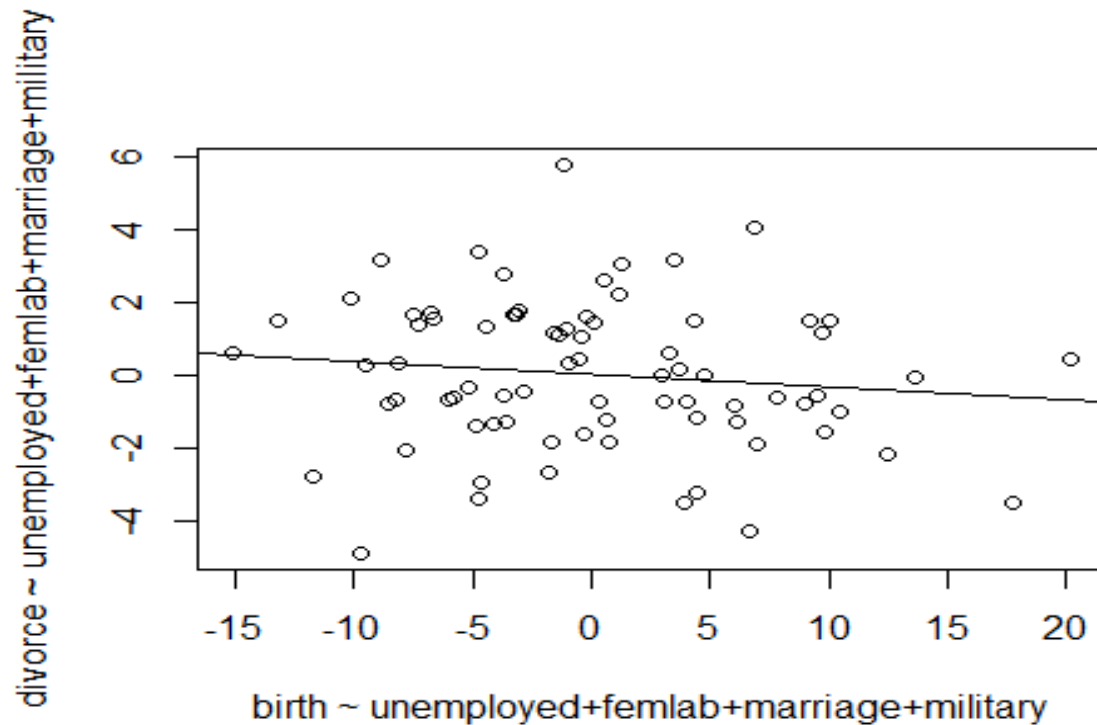
```
# Create a partial regression plot
m_divorce<-lm(divorce ~ unemployed+femlab+marriage+military, data = divusaB)
# first create the model
m_birth <-lm(birth ~ unemployed+femlab+marriage+military, data = divusaB)
# plot residuals
plot(m_divorce$res ~ m_birth$res, xlab = "birth ~ unemployed+femlab+marriage+
military",
```

```

ylab = "divorce ~ unemployed+femlab+marriage+military")

# fit a regression model to the residuals
m_res <-lm(m_divorce$res ~ m_birth$res)
abline(m_res)

```



```

summary(m_res)

##
## Call:
## lm(formula = m_divorce$res ~ m_birth$res)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.2328 -1.2259  0.0223  1.4396  5.7199
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.701e-17  2.299e-01   0.000    1.000
## m_birth$res  -3.510e-02  3.247e-02  -1.081    0.283
##
## Residual standard error: 2.018 on 75 degrees of freedom
## Multiple R-squared:  0.01534,    Adjusted R-squared:  0.002211
## F-statistic: 1.168 on 1 and 75 DF,  p-value: 0.2832

```

## Question 2b(v)

Test the hypothesis:

$H_0: \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0$   $H_A$ : at least one of the  $\beta_i \neq 0$  What do the results of the hypothesis test imply for the regression model?

## Solution

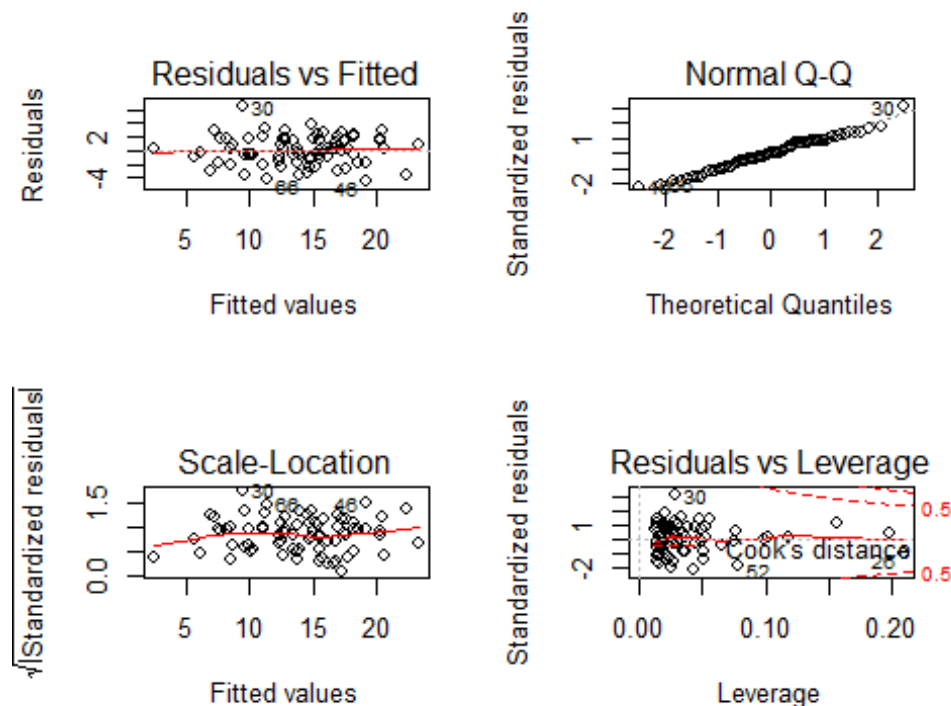
Examining the output for `summary(m1)` we see that the global F-statistic is  $F(5, 71) = 64.37$  and  $p = < 2.2e-16$ , this F-statistic compares the fitted model to the null model (also called the intercept only model). In this instance may reject the null hypothesis at the 1% confidence level and conclude that at least one of the predictors is associated with divorce.

## Question 2b(vi)

Assess the fit of the model using diagnostic plots, commenting on the assumptions of the regression model and influential points.

## Solution

```
#Diagnostic  
par(mfrow=c(2,2))  
plot(m1)
```



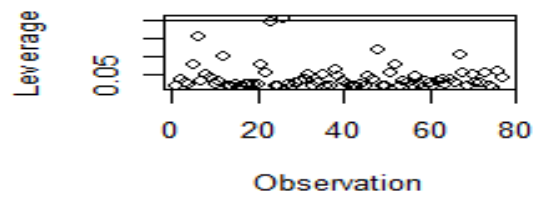
```

h<- lm.influence(m1)$hat
plot(h, xlab = "Observation", ylab = "Leverage")
abline(h=0.2)
identify(h,n=4)

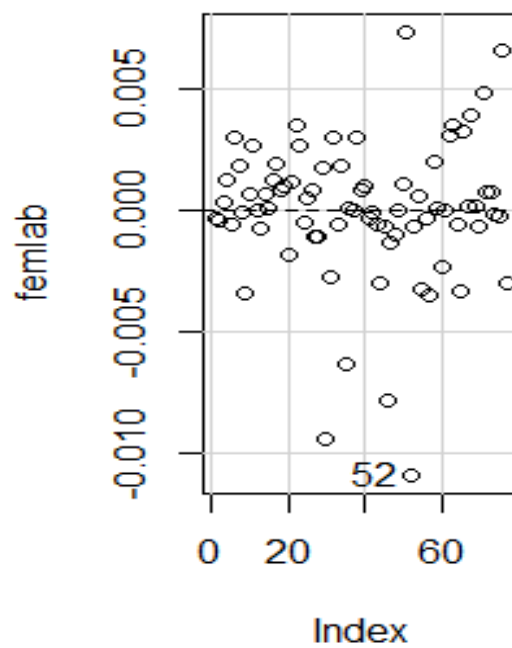
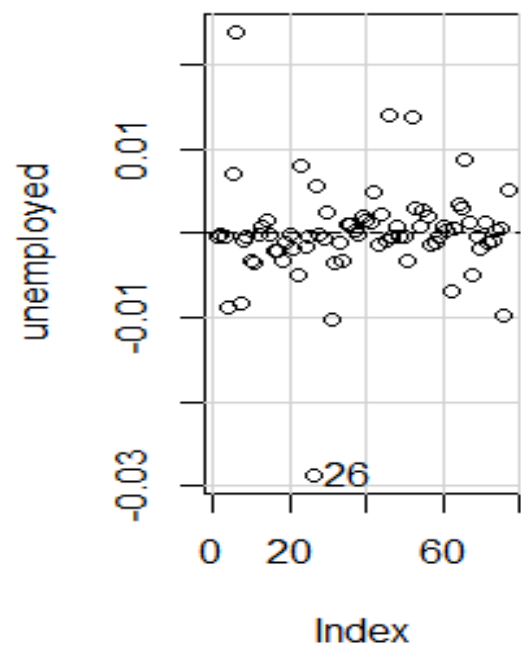
## integer(0)

dfbetaPlots(m1,id.n = 1)

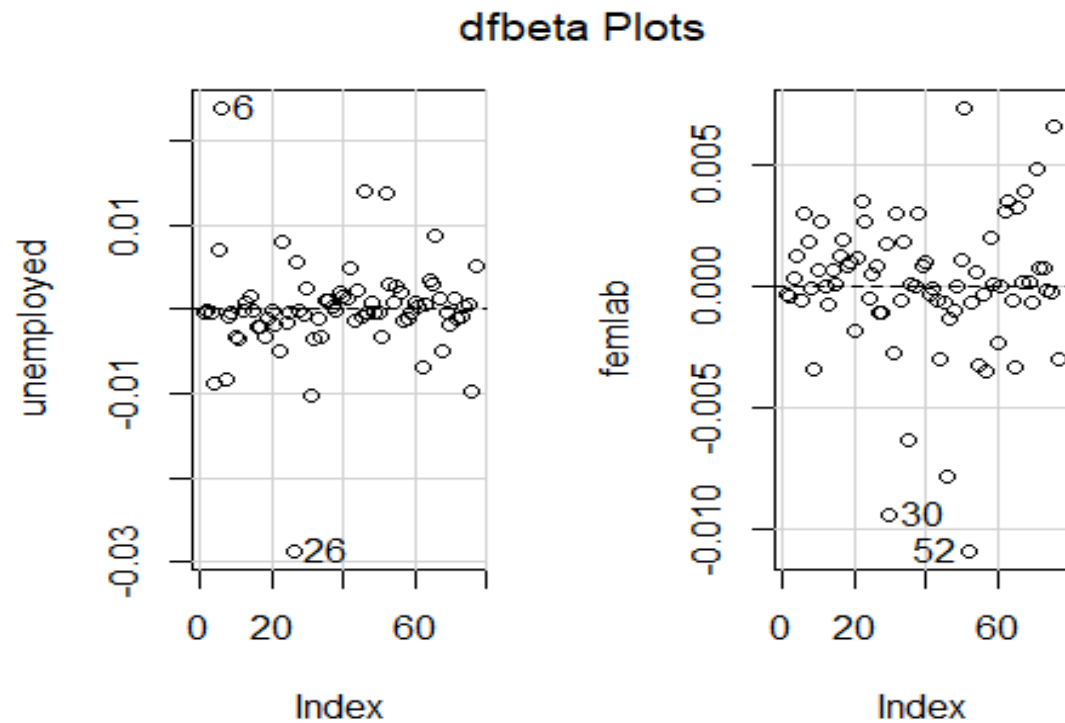
```



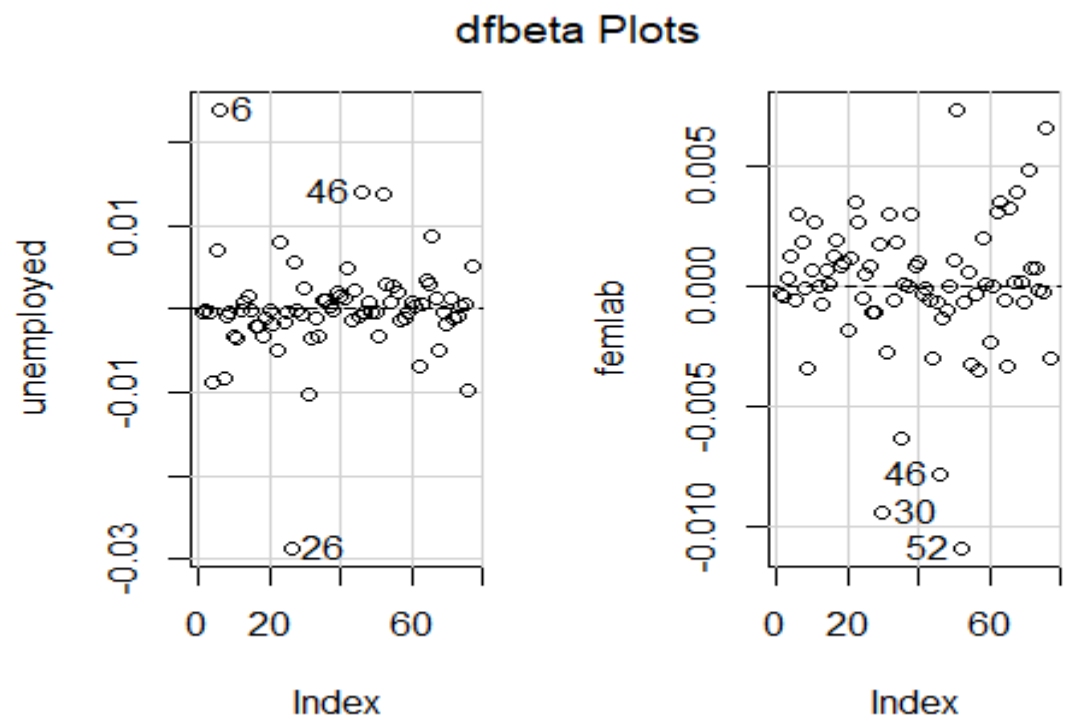
### dfbeta Plots



```
dfbetaPlots(m1,id.n = 2)
```

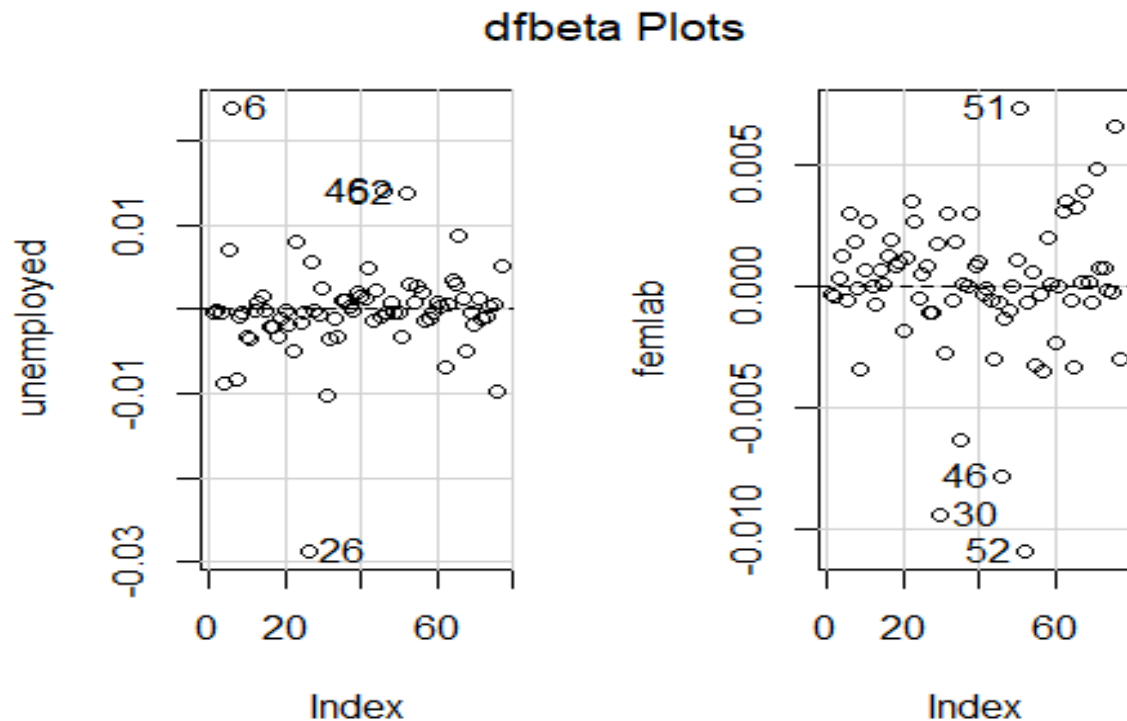


```
dfbetaPlots(m1,id.n = 3)
```





```
dfbetaPlots(m1,id.n = 4)
```



I have plotted the model m1 which gives the diagnostic plots are Residuals vs Fitted, Normal Q-Q, Scale\_Location and Residuals Vs Leverage. For Leverage the observation 23,24,25,26 have high leverage And also we can see the top four observations influential points of each variable by using dfbetaPlots.

Assumptions of Regression Model: Regression diagnostics are used to test the conclusions of the model and to analyze whether or not there are findings with a significant, undue effect on the study.

Again, linear regression principles are: linearity: The relationship is linear between X and the mean of Y. Homoscedasticity: The residual variance is the same on every X-value. Independence: Observations are independent of each other. Normality: Y is usually distributed for every fixed value of the X.

Influential observations: An influential observation is defined as the one that alters the line's slope. Influential points thus have a major impact on the model's fit. One approach for defining important points is to compare the model's match with and without each observation. Varying prominent outliers is the best technique. Outliers will affect calculations of parameters (e.g., mean), and even influence the square sums. Summaries of squares are used to determine the standard error, and if the square sums are skewed, the default error is likely too. Getting a skewed standard error is very poor since it is used to measure intervals of confidence around our estimation of parameters.

## Question 2(c)

Use an F-test to compare the full model to the model including all variables except unemployment.

### Solution

An F-test is any statistical test in which the test statistic has an F-distribution under the null hypothesis. It is most often used when comparing statistical models that have been fitted to a data set, in order to identify the model that best fits the population from which the data were sampled.

```
#performing F-test
#m1 <- lm(divorce ~ unemployed + femlab + marriage + birth + military, data =
divusaB)
#summary(m1)
m2<- lm(divorce ~ femlab + marriage + birth + military, data = divusaB)
summary(m2)

##
## Call:
## lm(formula = divorce ~ femlab + marriage + birth + military,
##     data = divusaB)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.1716 -1.2334  0.0473  1.2957  5.6087
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.08311     4.96903   -0.620    0.537
## femlab         0.43414     0.03534   12.286 <2e-16 ***
## marriage      0.06808     0.04279    1.591    0.116
## birth        -0.03735     0.03313   -1.127    0.263
## military     -0.03506     0.01751   -2.002    0.049 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.067 on 72 degrees of freedom
## Multiple R-squared:  0.8179, Adjusted R-squared:  0.8077
## F-statistic: 80.82 on 4 and 72 DF,  p-value: < 2.2e-16

anova(m2,m1)

## Analysis of Variance Table
##
## Model 1: divorce ~ femlab + marriage + birth + military
## Model 2: divorce ~ unemployed + femlab
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
```

```
## 1      72 307.66
## 2      74 326.24 -2    -18.576 2.1736 0.1212
```

The result of the test is as follows Res.Df RSS Df Sum of Sq F Pr(>F) 1 71 305.28 2 72 307.66 -1 -2.3864 0.555 0.4587 The residual sum of square for model1(m2) is 305.28 and model2(m1) is 307.66 with the difference -2.3864 and the p value is 0.4587. So we would fail to reject the null hypothesis and we can conclude that the data does not support the alternative hypothesis that there is difference in fit of two model. So we will conclude that the larger model does not fit better than the smaller model.

## Question 2(d)

Compare the predictive accuracy of the two models from part (c) using 50 repeats of 10-fold cross validation.

## Solution

Cross-validation is a technique that is used to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it.

In the k-fold cross-validation, the sample is partitioned into k equal size subsamples randomly. One subsample is used as the validation dataset for model testing among many of the k subsamples, and hence the majority of the k-1 subsamples are used as training data. The cross-validation method is replicated k-times, with each of the k subsamples being used as validation data exactly once. They will then combine the k outcomes from the folds to provide a single calculation. The benefit of this process is that all observations are used for both the training and testing, and that each observation is used only once for validation.

Below are the steps to perform k-fold cross-validation: 1. Split your entire dataset into k" folds" randomly 2. For each k-fold in the dataset, build the model on k – 1 folds of the dataset. Then, evaluate the model to check for kth fold performance 3. Record the error that is observed on each of the predictions 4. Repeat until every k-fold performs as the test set. 5. The average of the k recorded errors is called the cross-validation error and will serve as the performance metric for the model

```
#10-fold cross-validation
# define training control
train_control <- trainControl(method="repeatedcv", number=10, repeats=50)
# train the model
modell1 <- train(divorce~., data=divusaB, trControl=train_control, method="lm"
)
# summarize results
print(modell1)

## Linear Regression
##
```

```
## 77 samples
## 6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 50 times)
## Summary of sample sizes: 69, 69, 70, 69, 69, 70, ...
## Resampling results:
##
##    RMSE      Rsquared   MAE
##    2.070999   0.8259243   1.722525
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

The result of model is as follows RMSE Rsquared MAE  
 2.071201 0.8230761 1.718682 Now, The two models from part(c) are m1 <- lm(divorce ~ unemployed + femlab + marriage + birth + military, data = divusaB) m2<- lm(divorce ~ femlab + marriage + birth + military, data = divusaB) Now I am going to apply k-fold cross-validation for both the models

```
#performing 10-fold cross-validation
m1 <- train(divorce ~ unemployed + femlab + marriage + birth + military, data = divusaB, trControl=train_control, method="lm")
# summarize results
print(m1)

## Linear Regression
##
## 77 samples
## 5 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 50 times)
## Summary of sample sizes: 69, 70, 69, 69, 70, 69, ...
## Resampling results:
##
##    RMSE      Rsquared   MAE
##    2.072958   0.8215972   1.718948
##
## Tuning parameter 'intercept' was held constant at a value of TRUE

m2 <- train(divorce~femlab + marriage + birth + military, data=divusaB, trControl=train_control, method="lm")
# summarize results
print(m2)

## Linear Regression
##
## 77 samples
## 4 predictor
##
## No pre-processing
```

```
## Resampling: Cross-Validated (10 fold, repeated 50 times)
## Summary of sample sizes: 69, 69, 70, 69, 70, 69, ...
## Resampling results:
##
##      RMSE      Rsquared   MAE
##  2.06147  0.8208524  1.721713
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

The result from model m1 is as follows RMSE Rsquared MAE  
 2.070958 0.825239 1.721155 The result for the model m2 is as follows RMSE Rsquared  
 MAE  
 2.062907 0.8234187 1.721076 From both the models m1 and m2 we can see that the RMSE  
 values are m1=2.07 and m2=2.06 There is only slight difference between m1 and m2 of 0.1.  
 Therefore we can say that model2(m2) performance is better when compare to  
 model1(m1)

## Question 2(e)

Can you suggest any improvements to the model?

## Solution

I want to build two models first and would like to draw conclusion by looking in to those  
 models Model1 without marriage

```
#Linear model without marriage
divusa.lm1 <- lm(divorce ~unemployed+femlab+birth+military, data = divusaB)
summary(divusa.lm1)

##
## Call:
## lm(formula = divorce ~ unemployed + femlab + birth + military,
##     data = divusaB)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.4529 -1.1823 -0.0144  1.4987  6.1649
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.97202    3.94846   0.499   0.6190
## unemployed   -0.06428    0.05708  -1.126   0.2639
## femlab        0.41332    0.03210  12.876 <2e-16 ***
## birth        -0.02582    0.03282  -0.787   0.4340
## military     -0.03119    0.01830  -1.705   0.0926 .
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.085 on 72 degrees of freedom
## Multiple R-squared:  0.8147, Adjusted R-squared:  0.8044
## F-statistic: 79.15 on 4 and 72 DF,  p-value: < 2.2e-16

vif(divusa.lm1)

## unemployed      femlab      birth      military
##  1.099367    1.787795    1.851476    1.149483
```

In the absence of marriage variable in the dataset, birth variable is no longer significant due to high p-value. VIF value of femlab has decreased to 1.787795. R2 value has decreased to 0.8147. Residual standard error(S) also increased to 2.085. This means range increases for predicted interval and confidence interval. Model2 without birth

```
#Linear model birth
divusa.lm2 <- lm(divorce ~ unemployed+femlab+marriage+military, data = divusa
B)
summary(divusa.lm2)

##
## Call:
## lm(formula = divorce ~ unemployed + femlab + marriage + military,
##     data = divusaB)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.8912 -1.2820 -0.0224  1.4666  5.7596
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.55150     4.06844  -1.365   0.177
## unemployed  -0.04939     0.05860  -0.843   0.402
## femlab       0.45208     0.03098  14.590 <2e-16 ***
## marriage     0.04975     0.04351   1.143   0.257
## military    -0.02689     0.01767  -1.522   0.132
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.075 on 72 degrees of freedom
## Multiple R-squared:  0.8165, Adjusted R-squared:  0.8063
## F-statistic: 80.07 on 4 and 72 DF,  p-value: < 2.2e-16

vif(divusa.lm2)

## unemployed      femlab      marriage      military
##  1.169590    1.681645    1.805443    1.082569
```

In the absence of birth variable in the dataset, unemployed variables are no longer significant due to high p-value. There is a significant decrease in intercept( $\beta_0$ ). VIF value of

femlab has decreased to 1.681645. R2 value has decreased to 0.8165. Residual standard error(S) also increased to 2.075. This means range increases for predicted interval and confidence interval.

Conclusion: Removal of variables has not improved the model. In fact, it has an adverse effect on parameters R2, Residual standard error(S) and intercept( $\beta_0$ ).

#\*\*\*\*\* #

## QUESTION 3

### Question 3(a)

Please write two or three paragraphs on the concept of step-wise regression, including a brief description of three different types of step-wise regression and an overview of the different criteria that can be used to decide whether an explanatory variable is to be retained in the final model. Finally, select and discuss one problem associated with step-wise regression.

### Solution

Step-wise regression: In statistics, stepwise regression is a method of fitting regression models, where an automated technique is used to choose predictive variables. In each step, a variable is considered dependent on some fixed requirement for addition to or subtraction from the set of explanatory variables. This typically takes the form of a series of F-tests or t-tests, but other methods, such as modified R2, Akaike information criterion, Bayesian information criterion, Mallows' Cp, Release, or false discovery risk, might be appropriate. The repetitive procedure of fitting the final model chosen accompanied by providing estimates and confidence intervals without adjusting them to represent the model building process calls preventing further use of stepwise model building entirely or at least ensuring that model complexity is properly reflected.

Three different types of step-wise regression:

#### 1. Forward (Step-Up) Selection:

Often this approach is used to provide an initial sampling of the candidate variables when there is a large number of variables. For example, suppose you have fifty to one hundred variables to choose from, well outside the realm of any reasonable regression technique. Using this forward selection technique to extract the best ten to fifteen variables would be a logical method and then apply the all-possible algorithm to the variables in this subset. This technique is also a safe choice in cases where multicollinearity is a problem.

The forward selection method is simple to define. You start with no variables in the model for the candidates. Pick an increasing element with the largest R-Squared. Select the candidate variable at each step that most increases R-Squared. Avoid introducing variables until there are no large variables left. Note that this can not be removed until a component has joined the model.

#### 2. Backward (Step-Down) Selection:

This approach is less common because it begins with a model that involves all candidate variables. However, you still keep a significant value of R-Squared as it works its way down instead of upwards. The problem is that the models chosen by this method that contain variables not necessarily needed. The programmer sets the amount of sense that variables can enter the model

The process of backward selection starts with all candidate variables in the process. The variable which is the least significant is omitted at each step. This method proceeds until there are no nonsignificant variables left. The developer determines the amount of importance at which variables can be eliminated from the model.

### 3.Stepwise Selection:

Stepwise regression is a variation of the sorting strategies forwards and backward. At one time it was very common, but the method for Multivariate Variable Selection mentioned in a later chapter would often do at least as well and typically better. Stepwise regression is a modification of the forward selection such that all candidate variables in the model are tested at each step in which a variable has been introduced to see if their importance has been decreased below the defined tolerance point. If a component with no importance is detected, it is excluded from the formula. Stepwise regression needs two stages of significance: one for variables added, and one for variables omitted. The cutoff probability of adding variables will be greater than the cutoff probability of eliminating variables in order to prevent the process from forming an infinite loop.

And the limitation with the algorithms are :

Each of the sample sizes is the biggest limitation of such techniques. A strong rule of thumb is that for every element in the applicant set, you have at least five observations. You will have 250 observations if you have 50 variables. With fewer data per item, these search procedures that match the randomness inherent in most datasets and will result in spurious models. It is a key point. To see what can happen when the sample sizes are too small, create a series of random numbers of 30 measurements for 20 variables. Run each of these procedures and see what a glorious RSquared value is obtained, even though its potential value is zero.

One difficulty associated with step-wise regression is that p-values are too small Overall, p-values are the probability of having a test statistic at least as serious as the one you get where the null hypothesis is true. If  $H_0$  is real, there should be a uniform distribution to the p-value.

But after continuous selection (or indeed, after a number of other methods to sample filtering), the p-values of those terms remaining in the model don't really have that property, even though we assume the null hypothesis is valid.

It is because we select variables that have or appear to have low p-values (depending on the particular parameters we used). It implies that usually, the p-values of the variables left in the formula are much less than they would be if we were to match a single sample. Remember that on average, filtering will choose models that tend to match much more



than the true model, if the model class contains the true model, or if the model class is robust enough to similarly approach the true model.

### Question 3(b)

Select a step-wise regression function from the statistical software R and give a brief description on how the algorithm is implemented. Include a description of the type of step-wise regression and the criterion used to determine whether a variable is included in the final model.

### Solution

There are so many functions and R packages for computing stepwise regression. These include: `stepAIC()` which is available in the MASS package, which chooses the best model by AIC. It has an option called `direction`, which takes the following values: i) “both” (both forward and backward selection); ii) “backward” (for backward selection) and iii) “forward” (for forwarding selection). It returns the best final model. The `stepAIC()` is implemented in the following way

```
library(MASS)

##
## Attaching package: 'MASS'

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

#I am loading swiss dataset which is available in R
data(swiss)
# Fit the full model
full.model <- lm(Fertility ~., data = swiss)
# Stepwise regression model
step.model <- stepAIC(full.model, direction = "both",
                     trace = FALSE)

summary(step.model)

##
## Call:
## lm(formula = Fertility ~ Agriculture + Education + Catholic +
##      Infant.Mortality, data = swiss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.6765  -6.0522   0.7514   3.1664  16.1422
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    62.10131     9.60489   6.466 8.49e-08 ***
```

```
## Agriculture      -0.15462    0.06819   -2.267   0.02857 *
## Education        -0.98026    0.14814   -6.617  5.14e-08 ***
## Catholic          0.12467    0.02889    4.315  9.50e-05 ***
## Infant.Mortality 1.07844    0.38187    2.824   0.00722 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.168 on 42 degrees of freedom
## Multiple R-squared:  0.6993, Adjusted R-squared:  0.6707
## F-statistic: 24.42 on 4 and 42 DF,  p-value: 1.717e-10
```

I have used the dataset swiss which is available in R. I built the normal linear model by using all the explanatory variable and considered Fertility as the dependent variable and then I applied stepwise regression model by using stepAIC() function. In this example, both backward and forward selection is used.

### Question 3(c)

- Use 10-fold cross validation with step-wise regression to select a final model based on minimising the RMSE. Does using cross-validation prevent spurious conclusions?

### Solution

In order to build the model by using 10-fold cross-validation with step-wise regression, I need to generate my own data first. Generating the data: We begin by creating a single data set with 15 predictors, 5 of which are linearly related to the outcome  $Y$  and 10 of which are noise. An important consideration in regression analysis is the signal to noise ratio i.e. the magnitude of the effect size (the  $\beta$  coefficients) to the variance. To start, set the  $\beta$  coefficients that are linearly related to the outcome to 0.5 and set the variance of the error term to 1. We create a data set with 300 observations.

Let  $k$  represent the number of explanatory variables  $X_j, j = 1 \dots k$

Each explanatory variable is drawn from a Standard Normal distribution  $X_j \sim N(0,1)$ .

Let  $k_{ln}$  represent the number of explanatory variables that are linearly related to the response variable  $Y$ .

Let  $B_{mag}$  represent the magnitude of the population  $\beta$  coefficients that are linearly related to the response  $Y$ .

The  $i_{th}$  observation of the response variable is defined to be  $[Y_i = \beta_0 + \beta_1 X_{i1} + \dots + \beta_k X_{ik} + e_i]$  where  $e_i \sim N(0,1)$

```
set.seed(2) # so can reproduce results
k = 15      # 15 explanatory variables
k_ln = 5    # 5 explanatory variables are linearly related to Y
B_mag = 0.5 # the magnitude of the explanatory variables that are linearly related to Y
n = 300     # number of observations in data set
```

```

#create a vector containing the population beta coefficients
B <- c(rep(B_mag,k_ln),rep(0,k-k_ln))
X <- matrix(rnorm(n*k), nrow=n) # create the explanatory variables
Y <- X%*%B + matrix(rnorm(n),nrow=n) # create the response variable
# combine the response and explanatory variables in a single dataframe
DF <- cbind(Y,X)
DF<-as.data.frame(DF)
# assign column names
colnames(DF)<-c("Y","X1","X2","X3","X4","X5","X6","X7","X8","X9","X10","X11",
               "X12","X13","X14","X15")

```

Now I use 10-fold cross-validation to estimate the average prediction error (RMSE) of each of the 15 models. The RMSE statistical metric is used to compare the 15 models and automatically choose the best model. The best is defined as the model that minimizes the RMSE.

```

# Set seed for reproducibility
set.seed(123)
# Set up repeated k-fold cross-validation
train.control <- trainControl(method = "cv", number = 10)
# Train the model
step.model <- train(Y ~., data = DF,
                    method = "leapBackward",
                    tuneGrid = data.frame(nvmax = 1:15),
                    trControl = train.control)
#printing the result
step.model$results

```

##	nvmax	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
## 1	1	1.5036451	0.09761691	1.1863496	0.14850986	0.08129504	0.08817272
## 2	2	1.3511870	0.26074545	1.0697143	0.14737624	0.14877213	0.13652943
## 3	3	1.3017278	0.32335861	1.0636101	0.12498193	0.11836366	0.12176649
## 4	4	1.2071628	0.42167259	0.9904012	0.09242621	0.10236222	0.07710407
## 5	5	0.9710298	0.60995634	0.7887817	0.10388615	0.10885257	0.09796131
## 6	6	0.9754062	0.60828358	0.7871348	0.10384811	0.10862900	0.08912724
## 7	7	0.9834076	0.60140037	0.7914725	0.10932110	0.11018991	0.09770902
## 8	8	0.9905276	0.59291384	0.7958144	0.10771889	0.10872477	0.09729921
## 9	9	0.9897257	0.59399292	0.7913583	0.10747869	0.10644178	0.09834255
## 10	10	0.9847047	0.59837768	0.7880585	0.11011439	0.10680983	0.10158833
## 11	11	0.9840741	0.59905540	0.7886560	0.11473575	0.10948880	0.10415889
## 12	12	0.9880610	0.59559026	0.7918650	0.11477845	0.10761186	0.10559052
## 13	13	0.9879634	0.59544226	0.7927730	0.11330007	0.10713066	0.10624851
## 14	14	0.9880880	0.59527676	0.7927779	0.11452586	0.10813084	0.10775830
## 15	15	0.9882385	0.59523566	0.7931251	0.11439986	0.10807706	0.10774557

The output above shows different metrics and their standard deviation for comparing the accuracy of the 15 best models. The Columns are:

nvmax: Number of variables within the model

RMSE and MAE are totally different metrics that measure the prediction error of every model. The lower the RMSE and MAE, the higher the model.

R-squared indicates the correlation between the observed outcome values and the values predicted by the model. The higher the R squared, the best the model.

The best model among the 15 models is

```
#printing the best model  
step.model$bestTune
```

```
##    nvmax  
## 5      5
```

This indicates that the best model is the one with nvmax = 5 variables. The function summary() reports the best set of variables for each model size, up to the best 5-variables model.

```
#printing the summary of all variables until the best model  
summary(step.model$finalModel)
```

```
## Subset selection object  
## 15 Variables (and intercept)  
##      Forced in Forced out  
## X1      FALSE      FALSE  
## X2      FALSE      FALSE  
## X3      FALSE      FALSE  
## X4      FALSE      FALSE  
## X5      FALSE      FALSE  
## X6      FALSE      FALSE  
## X7      FALSE      FALSE  
## X8      FALSE      FALSE  
## X9      FALSE      FALSE  
## X10     FALSE      FALSE  
## X11     FALSE      FALSE  
## X12     FALSE      FALSE  
## X13     FALSE      FALSE  
## X14     FALSE      FALSE  
## X15     FALSE      FALSE  
## 1 subsets of each size up to 5  
## Selection Algorithm: backward  
##      X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15  
## 1 ( 1 ) "*" " " " " " " " " " " " " " " " " " " " "  
## 2 ( 1 ) "*" " " " " "*" " " " " " " " " " " " " " " "  
## 3 ( 1 ) "*" "*" " " "*" " " " " " " " " " " " " " " "  
## 4 ( 1 ) "*" "*" "*" "*" " " " " " " " " " " " " " " "  
## 5 ( 1 ) "*" "*" "*" "*" "*" " " " " " " " " " " " " " " "
```

An asterisk specifies that the given variable is included in the corresponding model.

The regression coefficients of the final model (id = 5) can be accessed as follow:

```
#printing the coefficients  
coef(step.model$finalModel, 4)  
  
## (Intercept)          X1          X2          X3          X4  
##  0.1348693    0.5721939    0.5233610    0.5086602    0.5650929
```

This is the procedure to select a model based on minimizing the RMSE by using 10-fold cross-validation with step-wise regression.

if the particular training set has some spurious correlation with those test points, the model will have difficulties determining which correlations are real and which are spurious, because even though the training set doesn't change but the test set changes, Therefore, less correlated training folds means that the model will be fit for multiple unique datasets. Cross-validation is a technique that allows us to produce a test set like scoring metrics using the training set. That is, it allows us to simulate the effects of "going out of sample" using just our training data, so we can get a sense of how well our model generalizes.