Homework 1

Group 20

Student 1 Varun Ozarkar

Student 2 Venkata Revanth Kota

857-271-6470 (Tel of Student 1)

857-544-2329 (Tel of Student 2)

ozarkar.v@husky.neu.edu

kota.v@husky.neu.edu

Percentage of Effort Contributed by Student 1: 50

Percentage of Effort Contributed by Student 2: 50

Signature of Student 1: Varun Ozarkar

Signature of Student 2: Venkata Revanth Kota

Submission Date: 9/29/2017

```r
#install.packages("rlang")
#install.packages("gclus")
#install.packages("car")
#install.packages("MASS")
#install.packages("psych")
#install.packages("dataQualityR")
#install.packages("scatterplot3d")
#install.packages("plotrix")
#install.packages("tidyverse")
#install.packages("sm")
library(plotrix)

## Warning: package 'plotrix' was built under R version 3.3.3

library(scatterplot3d)

## Warning: package 'scatterplot3d' was built under R version 3.3.3

library(dataQualityR)

## Warning: package 'dataQualityR' was built under R version 3.3.2

library(psych)

## Warning: package 'psych' was built under R version 3.3.3

##
## Attaching package: 'psych'

## The following object is masked from 'package:plotrix':
##
##     rescale

library(MASS)

## Warning: package 'MASS' was built under R version 3.3.3

library(car)

## Warning: package 'car' was built under R version 3.3.3

##
## Attaching package: 'car'

## The following object is masked from 'package:psych':
##
##     logit

library(gclus)

## Warning: package 'gclus' was built under R version 3.3.3
```

```
## Loading required package: cluster

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.3.3

##
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':
##
##     %+%, alpha

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.3.3

## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr

## Warning: package 'tibble' was built under R version 3.3.3

## Warning: package 'tidyr' was built under R version 3.3.3

## Warning: package 'readr' was built under R version 3.3.3

## Warning: package 'purrr' was built under R version 3.3.3

## Warning: package 'dplyr' was built under R version 3.3.3

## Conflicts with tidy packages --------------------------------------------
-

## %+%():    ggplot2, psych
## alpha():  ggplot2, psych
## filter(): dplyr, stats
## lag():    dplyr, stats
## recode(): dplyr, car
## select(): dplyr, MASS
## some():   purrr, car

library(dplyr)
library(rlang)

## Warning: package 'rlang' was built under R version 3.3.3

##
## Attaching package: 'rlang'

## The following objects are masked from 'package:purrr':
##
```

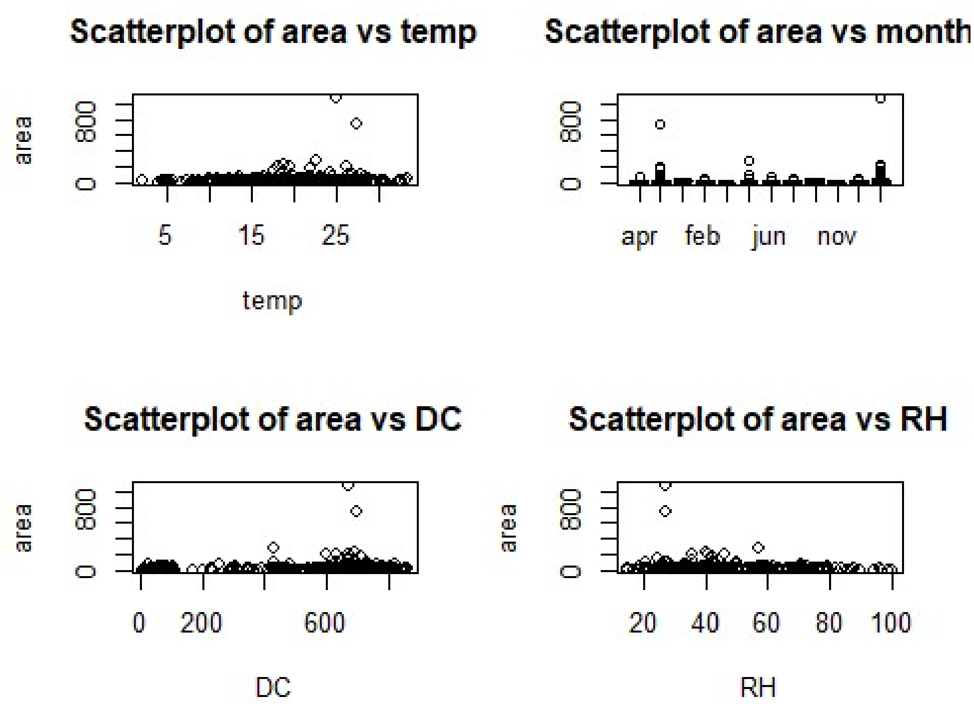
```
##     %@%, %||%, as_function, flatten, flatten_chr, flatten_dbl,
##     flatten_int, flatten_lgl, invoke, list_along, modify, prepend,
##     rep_along, splice

## The following object is masked from 'package:tibble':
##
##     has_name

library(sm)

## Warning: package 'sm' was built under R version 3.3.3

## Package 'sm', version 2.2-5.4: type help(sm) for summary information

##
## Attaching package: 'sm'

## The following object is masked from 'package:MASS':
##
##     muscle

setwd("C://Users//varun//Desktop//SEM4//Data mining//HW1")
```

Problem1

```
ff<-read.csv("forestfires.csv")
#View(ff)
```

Q1)a

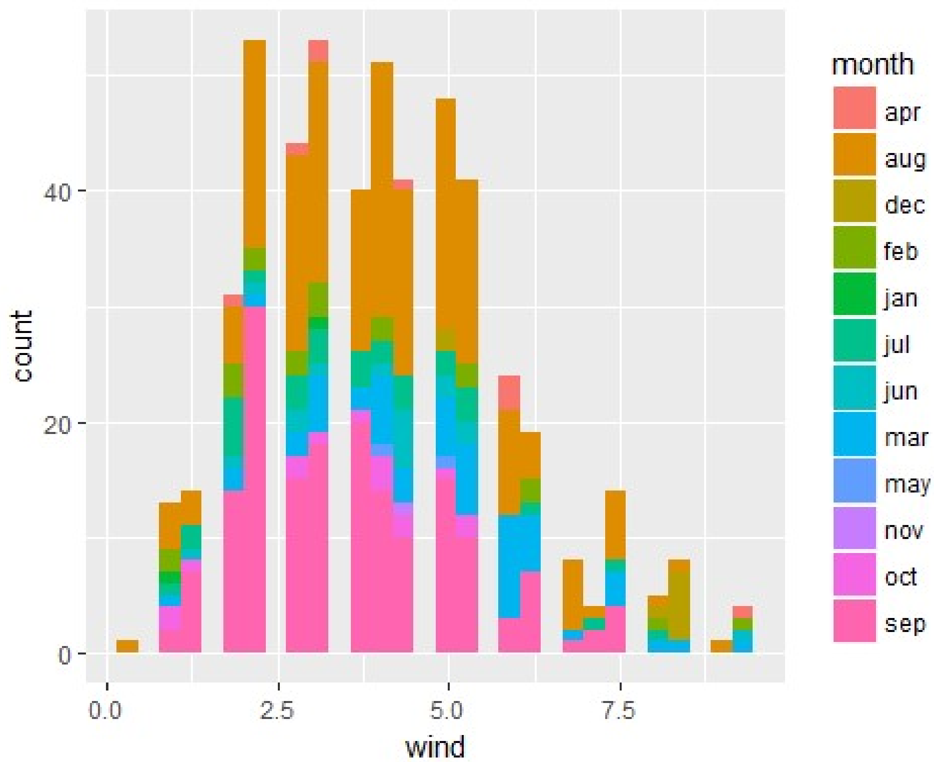
```
attach(ff) #attaching dataset ff ie. forstfires.csv
opar <- par(no.readonly=TRUE) #no.readonly=TRUE option produces a list of
current graphical settings that can be modified
par(mfrow=c(2,2)) #display the figures in the row(2), column(2) specification
by adding graphical parameters
plot(temp,area ,main="Scatterplot of area vs temp") #scatterplot of area vs
month
plot(month,area,main="Scatterplot of area vs month") #scatterplot of area vs
month
plot(DC,area,main="Scatterplot of area vs DC") #scatterplot of area vs DC
plot(RH,area,main="Scatterplot of area vs RH") #scatterplot of area vs RH
```




```
par(opar)
detach(ff) #detaching the dataset

#.  Here from Area vs Temp scatter plot we can say that higher the
temperature, higher are the incidents of forest fire. We can also interpret
that as the temperature goes on increasing, there is a chance that greater
forest area is burnt in the fire.
#.  From area vs month scatter plot we can see that there are higher
incidents of forest fires in August and September months as they are high
summer months. So we can justify the area vs temp plot by stating that higher
the temperatures (predominantly in summers), higher the incidents of forest
fire.
#.   From Area vs DC index we can see that higher the DC Index, Higher is the
chance for forest fire.
#.  Here in Area vs R.H graph we can say that relative humidity has no effect
in predicting the occurrence of  forest fire incidents .
```

Q1)b

```
ggplot(data = ff)+
  geom_histogram(mapping = aes(fill = month, x = wind),bins = 30) #plot a
histogram with 30 bins for wind and fill the bars with months
```

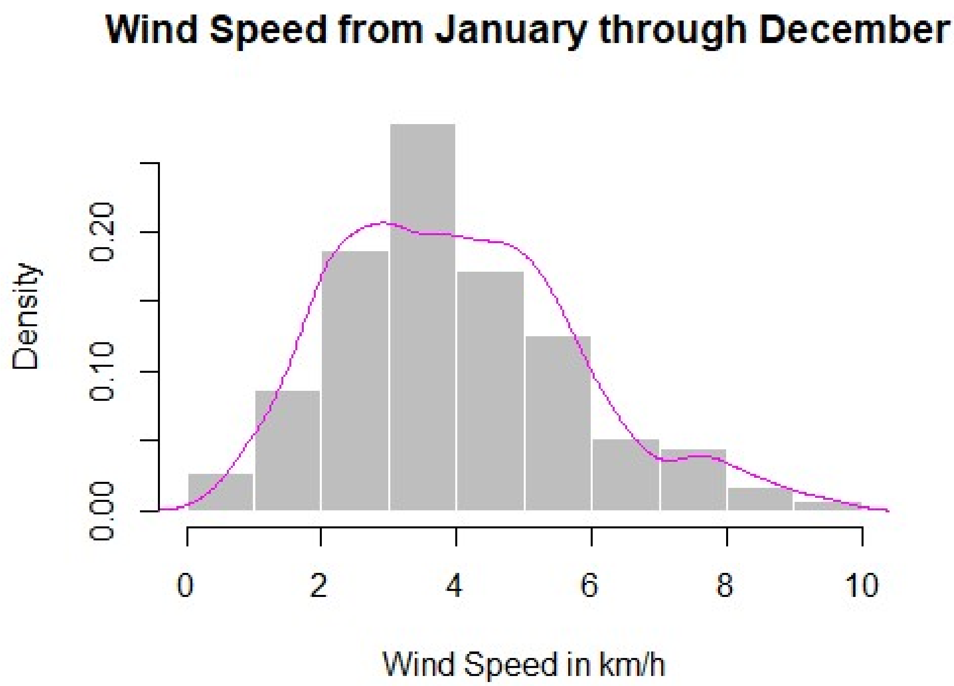


Q1)c

```
summary(ff$wind) #generate summary statistics

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.400   2.700   4.000   4.018   4.900   9.400
```

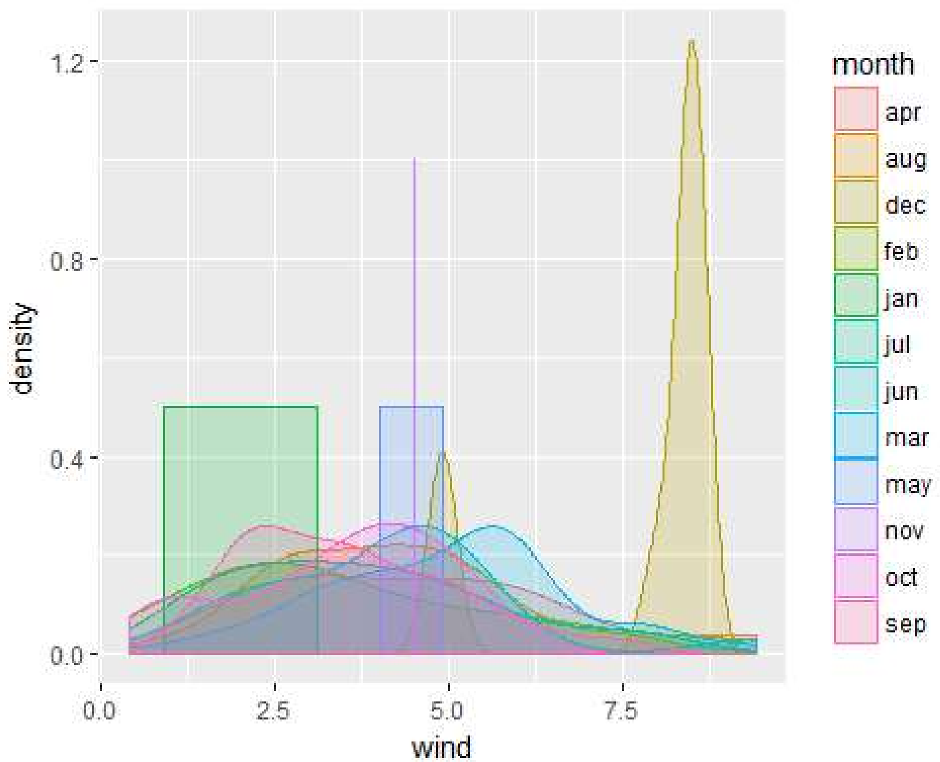
Q1)d

```
x<-ff$wind #assigning the wind column from ff dataset to an object 'x'
hist(x,xlab = "Wind Speed in km/h",main = " Wind Speed from January through
December", col = "Grey",border = FALSE,probability = TRUE) #create a
histogram for 'x' with the specified ordinates and abscissa with the color of
the bins specified as 'grey'
lines(density(x),col="Magenta") #add a density line of color 'magenta'
```



Q1)e

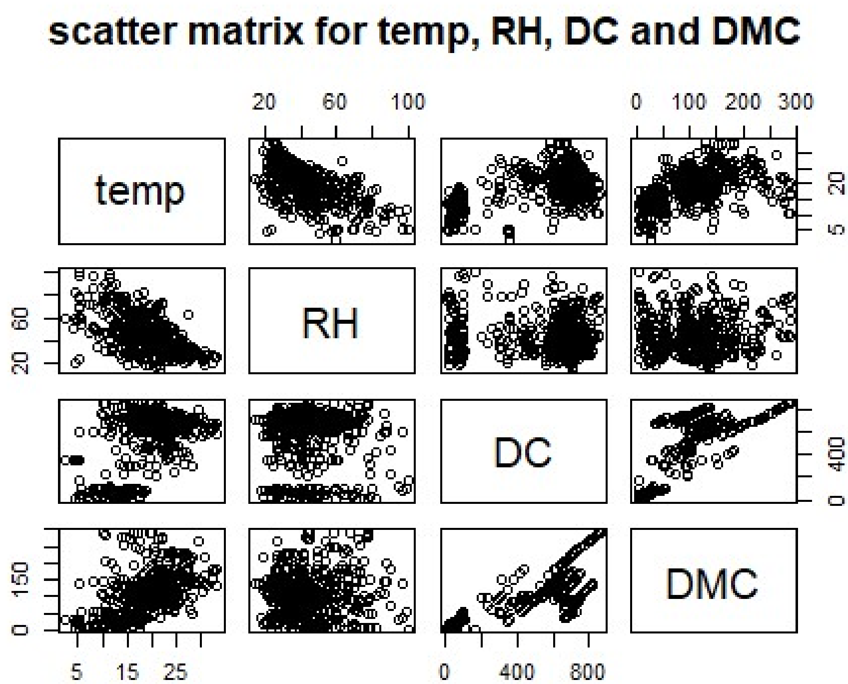
```
ggplot(ff, aes(wind, fill = month, color = month))+
  geom_density(alpha = 1/5) #geom_density displays a density kernel for the continuous wind speed data and we fill the plot with months
```




```
#Interpretation
#for months oct, nov, dec and jan there are sudden spikes in wind speed densities
```

Q1)f

```
pairs(~temp+RH+DC+DMC, data=ff,
      main="scatter matrix for temp, RH, DC and DMC") #produces a scatter plot matrix for the variables: temp, RH, DC, DMC with the mentioned title in main
```

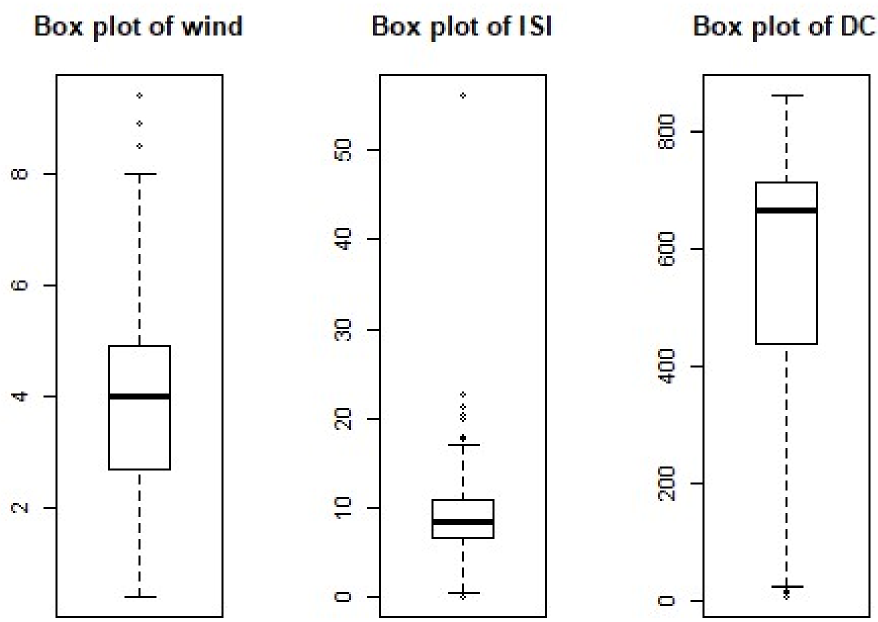



```
#Interpretation
#temp vs RH - Inversely proportional
#temp vs DC - weak non linear correlation
#temp vs DMC - weak non linear correlation
#RH vs DC - No correlation at all
#RH vs DMC - No correlation at all
#DC vs DMC - Directly proportional

cor(ff[c("temp", "RH", "DC", "DMC")])

##            temp          RH          DC        DMC
## temp  1.0000000 -0.52739034  0.49620805 0.46959384
## RH   -0.5273903  1.00000000 -0.03919165 0.07379494
## DC    0.4962081 -0.03919165  1.00000000 0.68219161
## DMC   0.4695938  0.07379494  0.68219161 1.00000000
```

Q1)g

```
attach(ff) #attach dataset
opar <- par(no.readonly=TRUE)
#no.readonly=TRUE option produces a list of current graphical settings that
can be modified
par(mfrow=c(1,3))
#display the figures in the row(1), column(3) specification by adding
graphical parameters
boxplot(ff$wind, main="Box plot of wind") #boxplot of wind
```

```r
boxplot(ff$ISI, main="Box plot of ISI") #boxplot of ISI
boxplot(ff$DC, main="Box plot of DC") #boxplot of DC
```

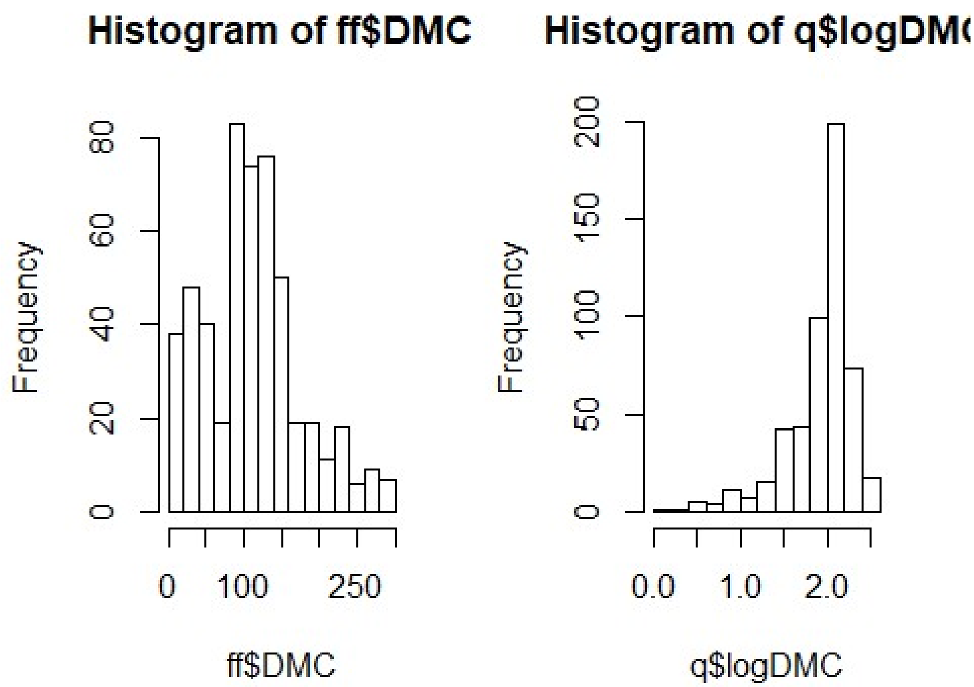



```r
par(opar)
detach(ff) #detaching dataset

#from the boxplots we can say there are anomalies.
#But the given boxplots are not sufficient to make any kind of
interpretation.
#Transforming these variables can give us a better insight
```

Q1)h

```r
q <- mutate(ff, logDMC = log10(DMC)) #assign the log10(DMC) to object 'q'

## Warning: package 'bindrcpp' was built under R version 3.3.3

attach(ff) #attach dataset 'ff'
attach(q) #attach dataset 'q'

## The following objects are masked from ff:
##
##     area, day, DC, DMC, FFMC, ISI, month, rain, RH, temp, wind, X,
##     Y

opar <- par(no.readonly=TRUE)
par(mfrow=c(1,2))
```

```
#display the figures in the row(1), column(2) specification by adding
graphical parameter
hist(ff$DMC) #histogram of DMC
hist(q$logDMC) #histogram of logDMC
```




```
par(opar)
detach(ff) #detach dataset 'ff'
detach(q) #detach dataset 'q'

#the histogram of 'DMC' does not give us a clear idea about the data
distribution though we can see that majority of data points lie in a range of
100 to 175 with an increase in distribution from 0 to 100 and then decrease
from 100.
# But upon transforming the DMC Variable on a logarithamic scale we can see
that the data is left skewed.
```

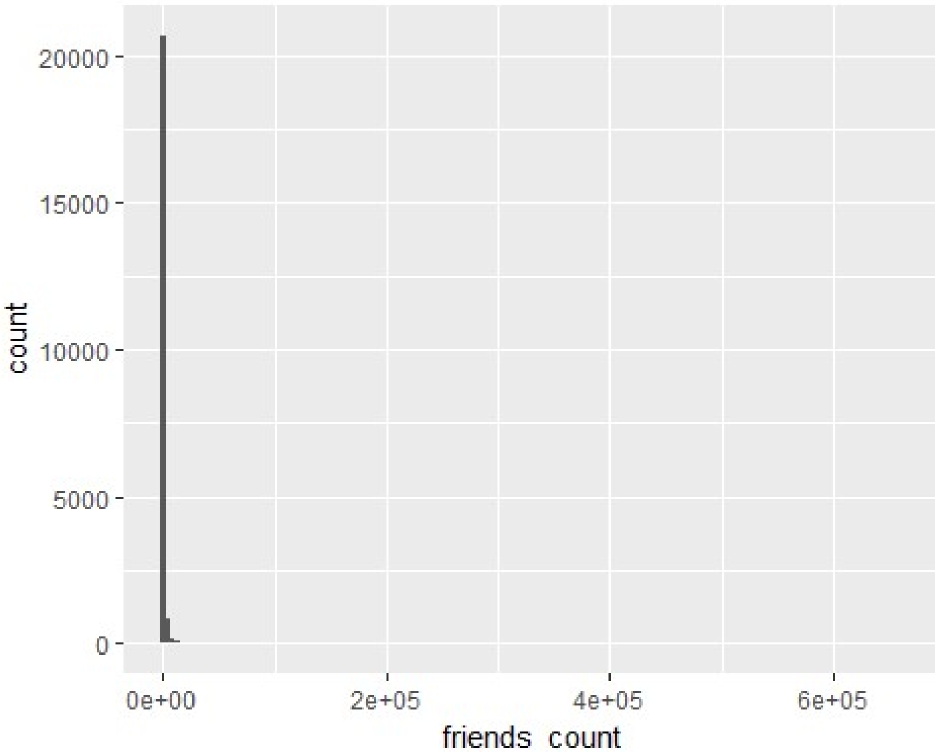
Problem 2

```
t<-read.csv("M01_quasi_twitter.csv") #reads the M01_quasi_twitter.csv file
and assigns it to object 't'
```

Q2)a

```
describe(t$friends_count) #describes the data distribution of object 't'

##    vars     n    mean      sd median trimmed    mad min    max  range
## X1    1 21916 1057.91 8125.05    324  496.01 370.65 -84 660549 660633
```

```
##      skew kurtosis    se
## X1 52.72  3523.12 54.88

#

ggplot(data = t)+
  geom_histogram(mapping = aes(x = friends_count), bins = 150) #plots a
histogram with friends_count as x-axis
```




```
#from the plot as we move along the friend count variable, we can say the
data is right skewed
```

Q2)b

```
summary(t$friends_count) #computes summary statistics for 'friends_count'

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     -84     123     324    1058     849  660500
```

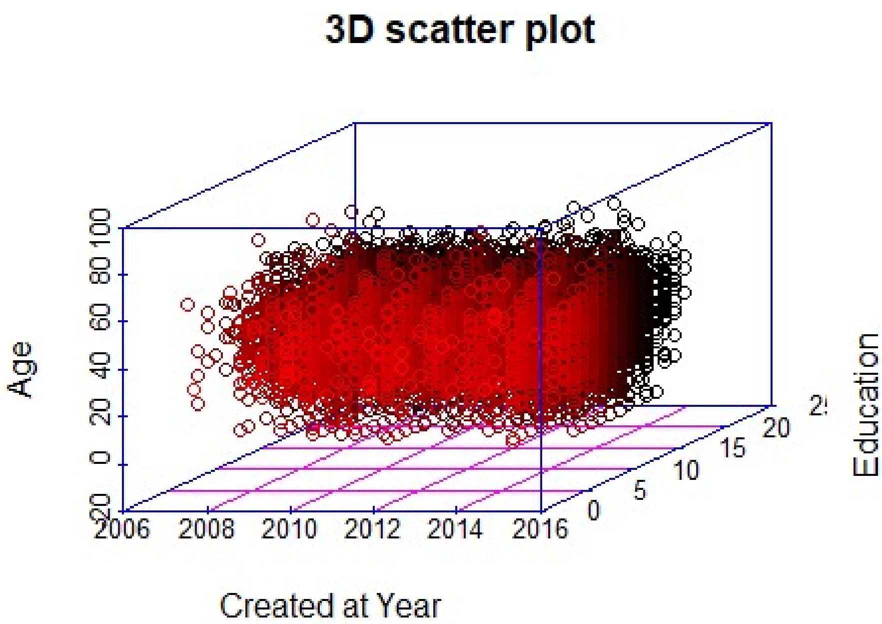
Q2)c

```
select(t,friends_count) %>% filter(NA) #diplays NA values in 'friends_count'
variable

## [1] friends_count
## <0 rows> (or 0-length row.names)
```

```r
#the friends_count has no NA values

select(t, friends_count) %>%
  filter(friends_count < 0) #displays values less than '0' in the
friends_count variable

##   friends_count
## 1           -84

#the friends_count variable has '-84' as one of the values. friends_count
cannot be negative hence we can say that this particular variable lacks
quality and we need to clean the friends_count variable before moving to
analysis part.

guess_parser(t$friends_count) #gives the type/class of the variable
'friends_count'

## [1] "integer"

# As the friends_quality variable is an integer, we can say that there are no
decimals points in the data
#The friends_count variable has integer values. As we checked for values
below '0', we can say that the apart from one anomaly (-84 value) the data
quality of friend_count variable is good.
```

Q2)d

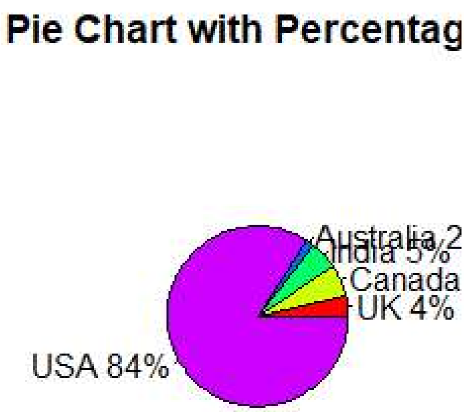
```r
attach(t)
scatterplot3d(created_at_year, education, age,
col.axis="blue",col.grid="magenta",xlab = "Created at Year",ylab =
"Education",zlab = "Age", main = "3D scatter plot", highlight.3d = T)
```




```
#Produces a 3D scatter plot on 't' dataset for variables: created_at_year,
education, age with the title:"3D scatter plot"

detach(t)
```

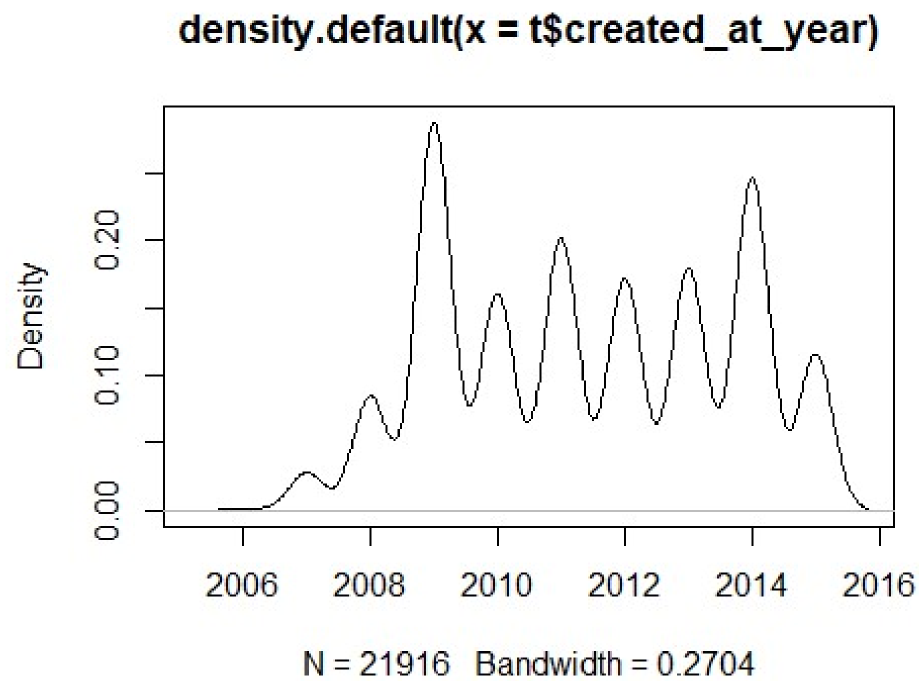
Q2)e

```
par(mfrow=c(1,2)) #display the 2 figures in the 1 row(1),2 column(2)
specification by adding graphical parameter
slices <- c(650, 1000, 900, 300, 14900) #stores the tweeter accounts values
as a vector in object 'slices'
lbls <- c("UK", "Canada", "India", "Australia", "USA") #stores the labels as
a vector in object 'lbls'
pct <- round(slices/sum(slices)*100) #calculates twitter accounts percentages
for each country and stores them in object 'pct'
lbls2 <- paste(lbls, " ", pct, "%", sep="") #pastes 'lbls' and 'pct'together
as a vector and stores in object 'lbls2' representing countries along with
their twitter accounts percentages.
pie(slices, labels=lbls2, col=rainbow(length(lbls2)),
    main="Pie Chart with Percentages") #plots a simple pie chart and colors
the 'lbls2'
pie3D(slices, labels=lbls2,explode=0.1,
      main="3D Pie Chart ") #pie 3D protrudes 2D pie chart into 3D pie chart
```



Q2)f

```
d <- density(t$created_at_year) #kernel density plot for variable
'created_at_year'
plot(d)
```




```
#the density for year 2009 is highest i.e. the maximum no. of twitter
accounts were created
#in the year 2009 for the date range and countries considered in the dataset

# Here from the graph we can say that the created_at_year variables does not
follow a specific pattern of increase or deecrease rather wavey.
```

Problem 3

```
rd <- read_csv("raw_data.csv") #reading the dataset and storing it in object
'rd'

## Parsed with column specification:
## cols(
##   A = col_double(),
##   B = col_double(),
##   C = col_integer(),
##   D = col_integer()
## )
```

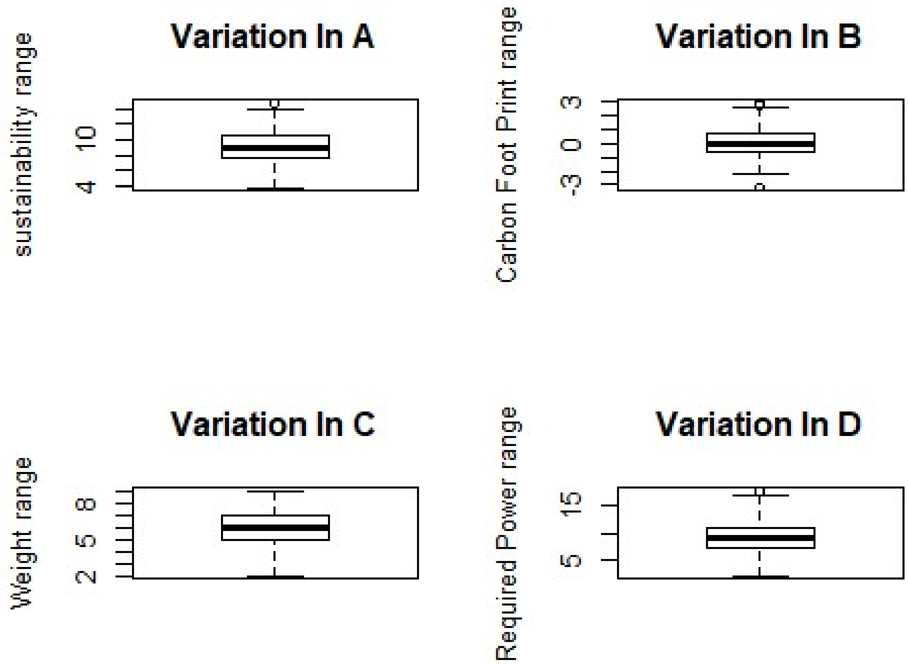
Q3)a

```
d1 <- mutate(rd, nma = (A-mean(A))/sd(A, na.rm = FALSE),
             nmb =(B-mean(B))/sd(B, na.rm = FALSE),
             nmc = (C-mean(C))/sd(C, na.rm = FALSE),
             nmd = (D-mean(D))/sd(D, na.rm = FALSE)) #Here we used mutate
function to add new normalized columns nma,nmb,nmc,nmd to the old data frame
```

```r
# Here we can see that we normalized the columns by substracting the
observations from their mean value and dividing them by their standard
deviation.

Ndata <- select(d1, -A,-B,-C,-D) #creates a new dataframe that consists of
only normalized variables
head(Ndata, n = 10 ) #Displayes the first 10 observations of the newly
created Ndata data frame.

## # A tibble: 10 x 4
##            nma        nmb        nmc         nmd
##          <dbl>      <dbl>      <dbl>       <dbl>
##  1 -0.46047167 -0.6870000 -0.2019694 -0.29312326
##  2  0.82780052 -0.7467798  0.4705888 -0.29312326
##  3 -0.18769316  0.7693173  0.4705888 -1.25008451
##  4 -1.41378095  1.5532638 -0.2019694  0.34485090
##  5  0.15837732  0.9970078  0.4705888 -0.29312326
##  6 -0.03285735  0.6893851  0.4705888  0.98282506
##  7  1.47453577  1.3112562 -2.8922024  0.34485090
##  8  0.25416645  0.4010108  0.4705888  0.34485090
##  9  0.08135825  2.2222747  1.1431470  0.02586382
## 10  0.66259177 -0.9918466 -1.5470859 -1.25008451
```

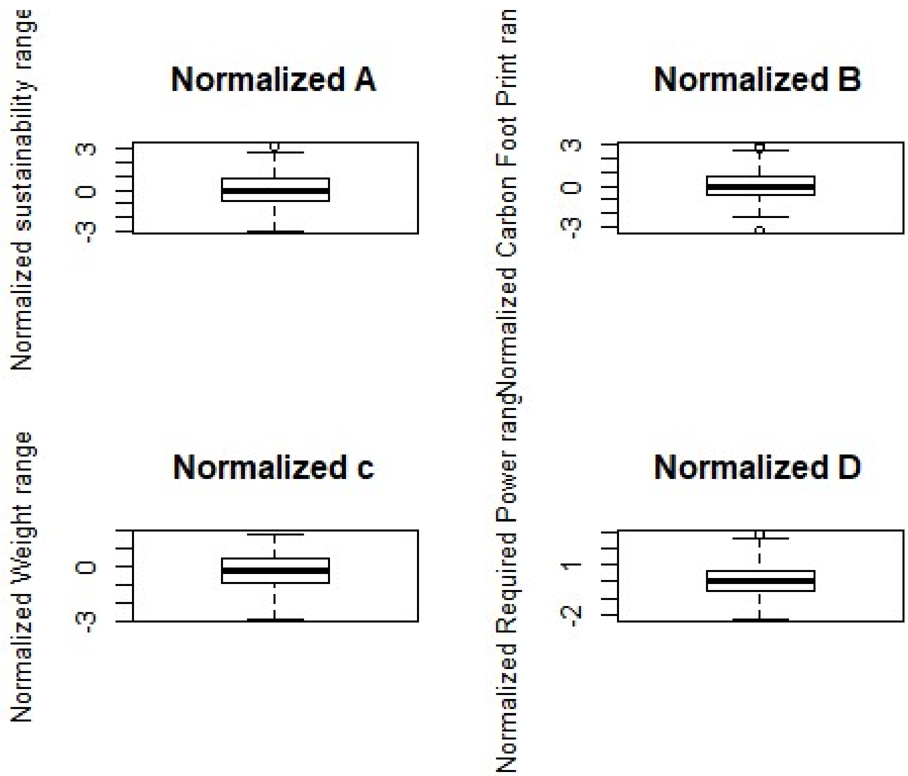
Q3)b

```r
par(mfrow = c(2,2))
boxplot(rd$A, main = "Variation In A", ylab = "sustainability range")
#boxplot for sustainability range
boxplot(rd$B, main = "Variation In B", ylab = "Carbon Foot Print range")
#boxplot for Carbon Foot Print range
boxplot(rd$C, main = "Variation In C", ylab = "Weight range") #boxplot for
Weight range
boxplot(rd$D, main = "Variation In D", ylab = "Required Power range")
#boxplot for Required Power range
```



Q3)c

```
par(mfrow = c(2,2))
boxplot(Ndata$nma, main = "Normalized A", ylab = "Normalized sustainability range") #boxplot for sustainability range of the normalized data
boxplot(Ndata$nmb,main = "Normalized B"  , ylab ="Normalized Carbon Foot Print range") #boxplot for Carbon Foot Print range of the normalized data
boxplot(Ndata$nmc, main = "Normalized c"  , ylab ="Normalized Weight range") #boxplot for Weight range of the normalized data
boxplot(Ndata$nmd, main = "Normalized D"  , ylab ="Normalized Required Power range") #Required Power range range of the normalized data
```

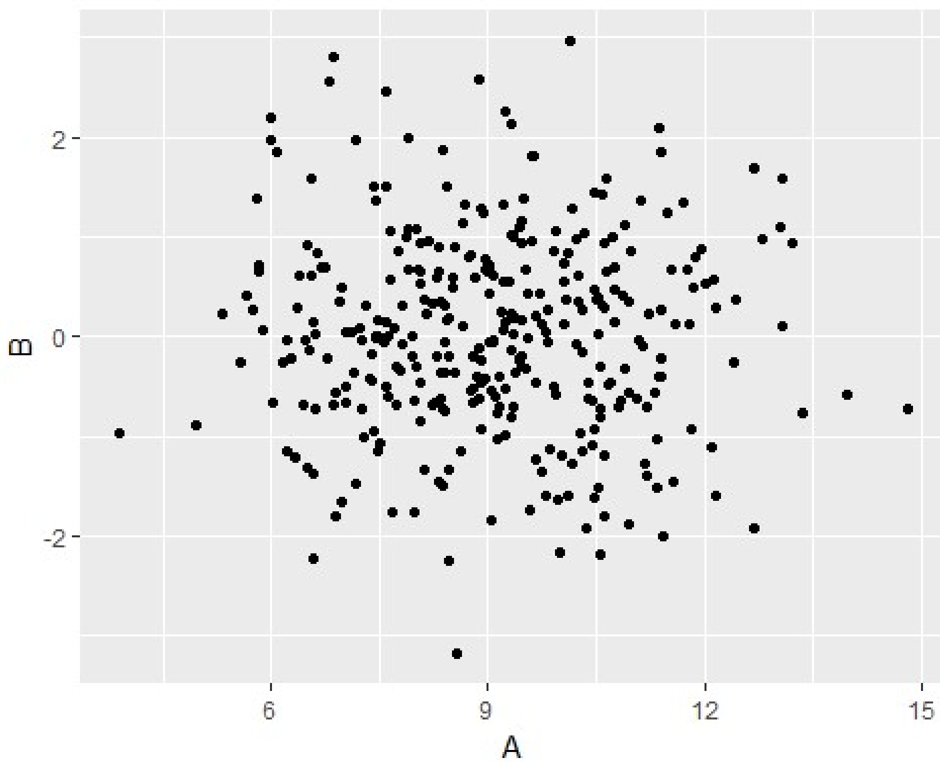


Q3)d

```
#From c plot we can, see that the boxplots from normalized data look almost the same and mean of the normalized data is around 0 for all the box plots which we could not infer from the original plot from B.
```

Q3)e

```
ggplot(data = rd)+
  geom_point(mapping = aes(x = A, y = B)) #plots geomp_point for variables 'A' and 'B'
```




```r
#we can interpret from the plot  there is no correlation

cor(rd[c("A", "B")]) #displays correlation between variables 'A' and 'B'

##             A           B
## A  1.00000000 -0.03059086
## B -0.03059086  1.00000000

#variables 'A' and 'B' have a correlation of only -0.03 which is very weak
```