INTRO TO R FOR DATA SCIENCE ASSIGNMENT2

## SHIVAANI KATRAGADDA

## R00183214

In this assignment, I was given an Assignment2.csv file which contains 43,824 entries, 13 total columns

The thirteen columns in the Assignment2.csv file are named as follows NO, year, month, day, hour, pm2.5, DEWP, TEMP, PRES, cbwd, Iws, Is and Ir

A description of the column headings is as follows:

No: row number

year: year of data in this row

month: month of data in this row

day: day of data in this row

hour: an hour of data in this row

pm2.5: PM2.5 concentration (ug/m^3)

DEWP: Dew Point (â„ƒ)

TEMP: Temperature (â„ƒ)

PRES: Pressure (hPa)

cbwd: Combined wind direction

Iws: Cumulated wind speed (m/s)

Is: Cumulated hours of snow Ir: Cumulated hours of rain

Among all the thirteen columns the column “No” does not have any importance as it just says the number of each entry i.e row number.

# install.packages("readr")  
# install.packages("DataExplorer")  
# install.packages("mice")  
# install.packages("shiny")  
# install.packages("car")  
# install.packages("psych")  
# install.packages("e1071")  
# install.packages("corrplot")  
library(readr)  
library(DataExplorer)  
library(mice)

## Loading required package: lattice

##   
## Attaching package: 'mice'

## The following objects are masked from 'package:base':  
##   
## cbind, rbind

library(shiny)  
library(car)

## Loading required package: carData

## Registered S3 methods overwritten by 'car':  
## method from  
## influence.merMod lme4  
## cooks.distance.influence.merMod lme4  
## dfbeta.influence.merMod lme4  
## dfbetas.influence.merMod lme4

library(psych)

##   
## Attaching package: 'psych'

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

library(e1071)  
library(corrplot)

## corrplot 0.84 loaded

Initially, I loaded both the file by using the read.csv() function which is available in the readr package. contains 43,824 entries, 13 total columns, and it is assigned to Assignment\_2. Then generated the data frame for Assignment\_2 using as.data.frame() function and that data frame is named as Assignment\_2.

I used some functions from the data explore package which will give some idea of a given dataset. Firstly I used the introduce() function will give the outline of the data it tells about the number of rows, columns, missing values, discrete columns, continuous columns, all missing columns, complete rows, total observations, and memory usage.

The plot\_missing() function shows the percentage of missing values of each column present in the dataset.

The missing plot shows the pm2.5 column have missing values so First fill the missing values using imputation technique

After that, I replaced the missing values by using the impute technique by using the mice package.

1.For calculating what percentage of data is missing for each column. I created a function and storing that function in variable p, the function contains the sum of missing values divided by the length of missing value and multiplied by 100 to get the percentage

1. Using apply function we are calling p,i.e function which will give the percentage of missing value

3.md.pattern(md stands for missing data) gives the table as well as a plot for the missing data.

1. There is one more method for missing data patterns for finding missing data is md.pairs.
2. Now we have to start doing Imputation, firstly storing imputed data in the variable impute and the function used in mice, within the data I am taking from 2 to 13 because a first column is just a row number and it does not have any significance or predictive power so I am going to ignore the first column for that I am using square braces and comma which means all the rows but only 2 to 13 columns and specifying how many imputations I want so m=3 and used random seed 123(seed is used here whenever you re-run program the imputed data will not change)
3. Printing impute variable it specifies the number of imputations and imputation methods used for filling the missing values, for pm2.5 column the “PMM” predictive mean matching method is used.

7.The imputepm2.5 lists all the rows where the pm2.5 values are missing and gives estimates for first, second and third imputations

8.we can get complete dataset by using the complete() function in that we can specify which imputation we want to complete the dataset

All the missing data is imputed or not check again whether any missing data is present in the dataset by using plot\_missing()

plot\_histogram() is provided by DataExplorer package which will provide histograms of all columns which are containing the continuous data

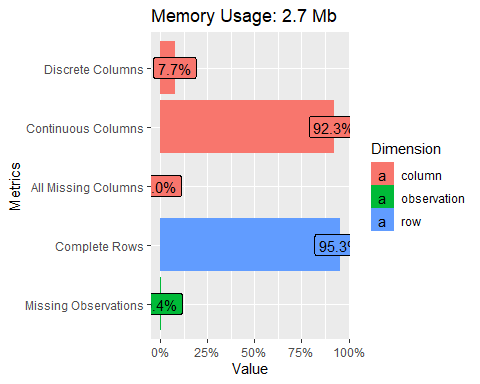
plot\_bar() is provided by DataExplorer package which will provide bar chart for discrete data

I am converted the cbwd which is a factor to numeric for convenience as all the other columns data is numeric. After that, I plot histogram all the columns including cbwd column histograms appears

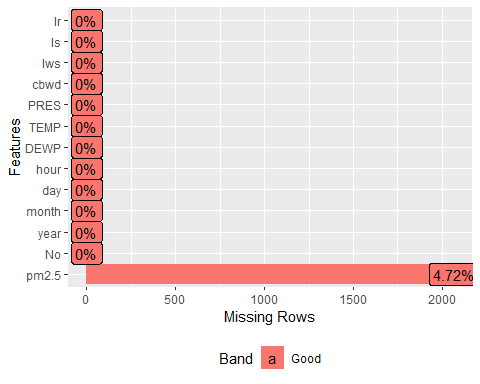
#Reading the Assignmenta.csv file by using read.csv() function and it is named as Assignment\_2  
  
Assignment\_2 <- read.csv("F:/R2 Assignment/Assignment 2.csv")  
  
  
#Generating the dataframe for Assignment\_2 using as.data.frame() function and that dataframe is named as Assignment\_2   
  
Assignment\_2<- as.data.frame(Assignment\_2)  
introduce(Assignment\_2)

## rows columns discrete\_columns continuous\_columns all\_missing\_columns  
## 1 43824 13 1 12 0  
## total\_missing\_values complete\_rows total\_observations memory\_usage  
## 1 2067 41757 569712 2808808

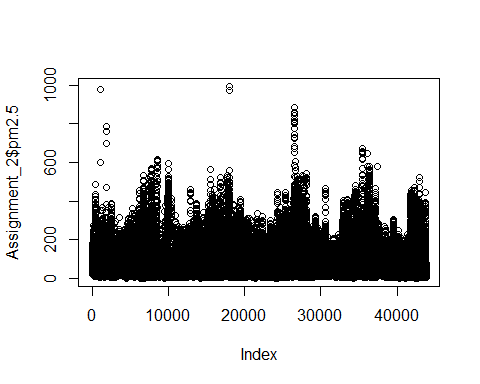
plot\_intro(Assignment\_2)



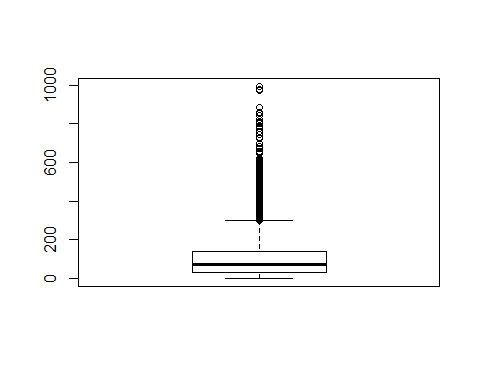
plot\_missing(Assignment\_2)



plot(Assignment\_2$pm2.5)#plot a graph for pm2.5 column of Assignment\_2 dataframe



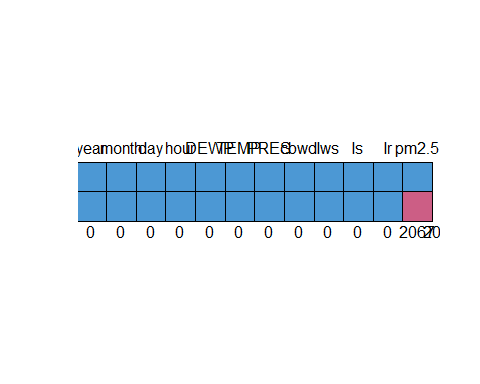
boxplot(Assignment\_2$pm2.5)#boxplot for pm2.5 column of Assignment\_2



p<-function(x){sum(is.na(x))/length(x)\*100}  
apply(Assignment\_2,2,p)

## No year month day hour pm2.5 DEWP TEMP   
## 0.000000 0.000000 0.000000 0.000000 0.000000 4.716594 0.000000 0.000000   
## PRES cbwd Iws Is Ir   
## 0.000000 0.000000 0.000000 0.000000 0.000000

md.pattern(Assignment\_2)



## No year month day hour DEWP TEMP PRES cbwd Iws Is Ir pm2.5   
## 41757 1 1 1 1 1 1 1 1 1 1 1 1 1 0  
## 2067 1 1 1 1 1 1 1 1 1 1 1 1 0 1  
## 0 0 0 0 0 0 0 0 0 0 0 0 2067 2067

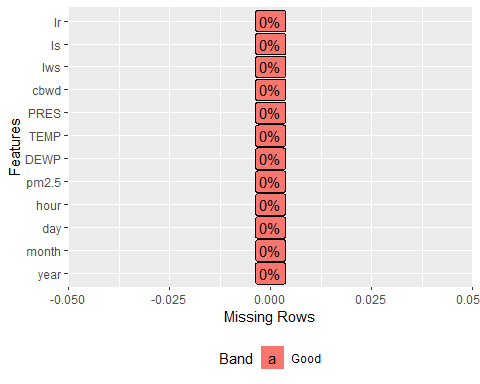
impute<-mice(Assignment\_2[,2:13],m=3,seed=214)

##   
## iter imp variable  
## 1 1 pm2.5  
## 1 2 pm2.5  
## 1 3 pm2.5  
## 2 1 pm2.5  
## 2 2 pm2.5  
## 2 3 pm2.5  
## 3 1 pm2.5  
## 3 2 pm2.5  
## 3 3 pm2.5  
## 4 1 pm2.5  
## 4 2 pm2.5  
## 4 3 pm2.5  
## 5 1 pm2.5  
## 5 2 pm2.5  
## 5 3 pm2.5

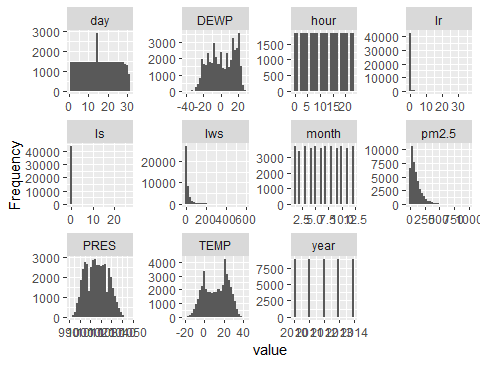
print(impute)

## Class: mids  
## Number of multiple imputations: 3   
## Imputation methods:  
## year month day hour pm2.5 DEWP TEMP PRES cbwd Iws Is Ir   
## "" "" "" "" "pmm" "" "" "" "" "" "" ""   
## PredictorMatrix:  
## year month day hour pm2.5 DEWP TEMP PRES cbwd Iws Is Ir  
## year 0 1 1 1 1 1 1 1 1 1 1 1  
## month 1 0 1 1 1 1 1 1 1 1 1 1  
## day 1 1 0 1 1 1 1 1 1 1 1 1  
## hour 1 1 1 0 1 1 1 1 1 1 1 1  
## pm2.5 1 1 1 1 0 1 1 1 1 1 1 1  
## DEWP 1 1 1 1 1 0 1 1 1 1 1 1

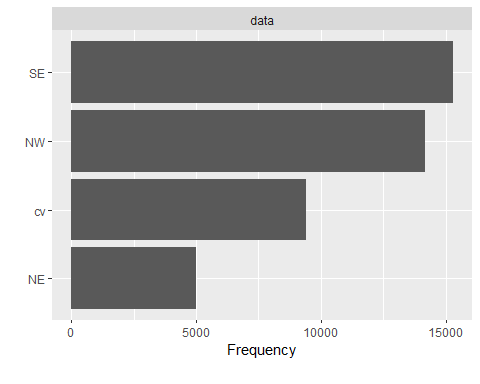
Assignment\_2<-complete(impute,1)  
plot\_missing(Assignment\_2)



plot\_histogram(Assignment\_2)



plot\_bar(Assignment\_2$cbwd)



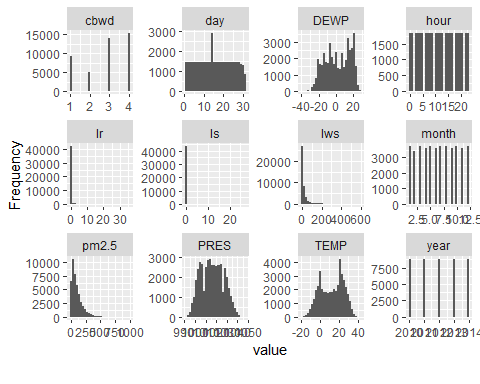
sort(table(Assignment\_2$cbwd))

##   
## NE cv NW SE   
## 4997 9387 14150 15290

class(Assignment\_2$cbwd)

## [1] "factor"

Assignment\_2$cbwd<-as.numeric(Assignment\_2$cbwd)  
plot\_histogram(Assignment\_2)

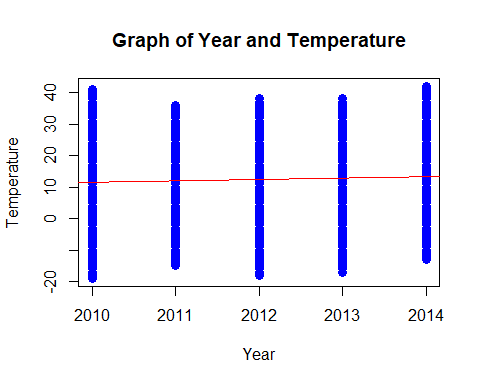


### QUESTION(1)

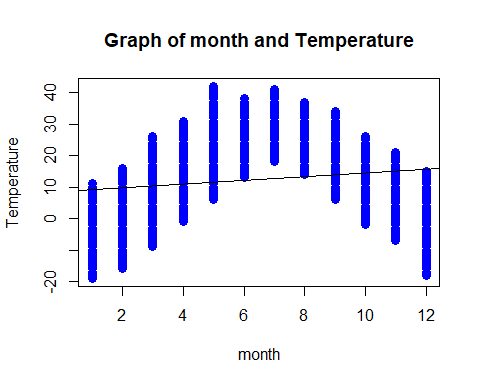
1. You must apply a linear regression model with TEMP as your y variable and an appropriate combination of other x variables from the dataset. You should justify your choice in terms of the number of variables.

Created various linear regression models by taking TEMP as y variable and approximate combination of other x variables.I did simple linear regression model for TEMPyear,TEMPmonth,TEMPday,TEMPhour,TEMPpm2.5,TEMPPRES,TEMP~DEWP and TEMP~cbwd the columns Iws,Ir,Is does not have that much significance and build multiple linear regression with different variables taking as x variable

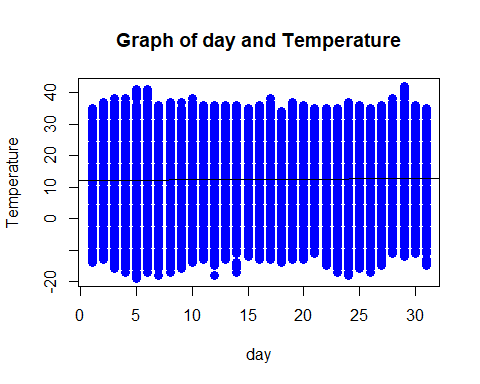
##   
## Call:  
## lm(formula = TEMP ~ year, data = Assignment\_2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -30.663 -10.841 1.159 10.552 29.337   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -777.90458 82.84015 -9.390 <2e-16 \*\*\*  
## year 0.39282 0.04117 9.541 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 12.19 on 43822 degrees of freedom  
## Multiple R-squared: 0.002073, Adjusted R-squared: 0.00205   
## F-statistic: 91.03 on 1 and 43822 DF, p-value: < 2.2e-16



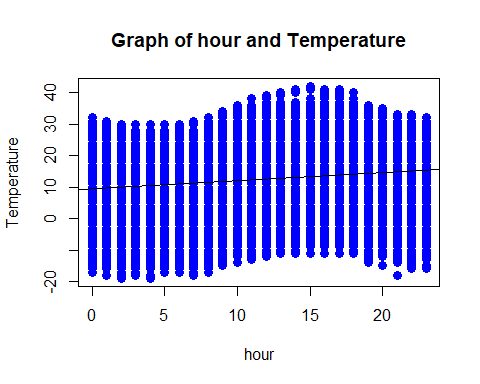
##   
## Call:  
## lm(formula = TEMP ~ month, data = Assignment\_2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -33.744 -10.329 1.061 10.265 30.468   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.52351 0.12287 69.37 <2e-16 \*\*\*  
## month 0.60167 0.01665 36.13 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 12.02 on 43822 degrees of freedom  
## Multiple R-squared: 0.02893, Adjusted R-squared: 0.02891   
## F-statistic: 1306 on 1 and 43822 DF, p-value: < 2.2e-16



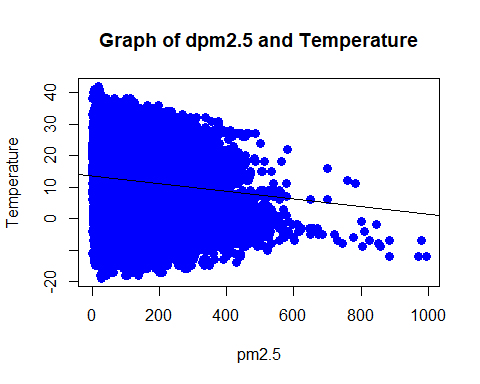
##   
## Call:  
## lm(formula = TEMP ~ day, data = Assignment\_2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -31.23 -10.64 1.32 10.50 29.28   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 12.126026 0.119334 101.614 < 2e-16 \*\*\*  
## day 0.020505 0.006622 3.097 0.00196 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 12.2 on 43822 degrees of freedom  
## Multiple R-squared: 0.0002188, Adjusted R-squared: 0.000196   
## F-statistic: 9.589 on 1 and 43822 DF, p-value: 0.001958



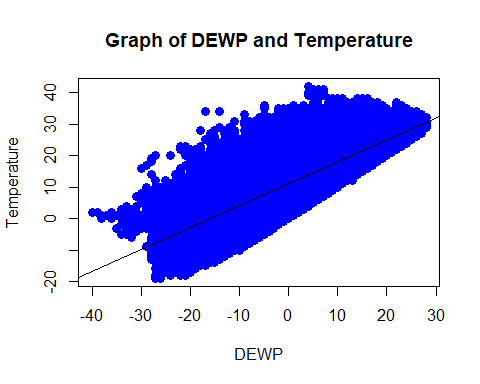
##   
## Call:  
## lm(formula = TEMP ~ hour, data = Assignment\_2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -32.961 -10.730 1.304 10.574 28.626   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 9.407348 0.111714 84.21 <2e-16 \*\*\*  
## hour 0.264450 0.008323 31.77 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 12.06 on 43822 degrees of freedom  
## Multiple R-squared: 0.02252, Adjusted R-squared: 0.0225   
## F-statistic: 1010 on 1 and 43822 DF, p-value: < 2.2e-16



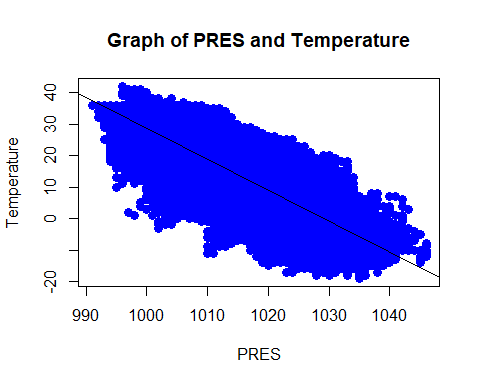
##   
## Call:  
## lm(formula = TEMP ~ pm2.5, data = Assignment\_2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -32.332 -10.505 1.174 10.446 28.569   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 13.6652714 0.0850208 160.73 <2e-16 \*\*\*  
## pm2.5 -0.0123421 0.0006304 -19.58 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 12.15 on 43822 degrees of freedom  
## Multiple R-squared: 0.008671, Adjusted R-squared: 0.008648   
## F-statistic: 383.3 on 1 and 43822 DF, p-value: < 2.2e-16



##   
## Call:  
## lm(formula = TEMP ~ DEWP, data = Assignment\_2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -14.728 -4.848 -1.031 4.273 34.666   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 11.181992 0.033223 336.6 <2e-16 \*\*\*  
## DEWP 0.696950 0.002284 305.2 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.9 on 43822 degrees of freedom  
## Multiple R-squared: 0.68, Adjusted R-squared: 0.68   
## F-statistic: 9.313e+04 on 1 and 43822 DF, p-value: < 2.2e-16



##   
## Call:  
## lm(formula = TEMP ~ PRES, data = Assignment\_2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -29.7805 -4.1393 0.3271 4.9325 21.8069   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.011e+03 3.246e+00 311.4 <2e-16 \*\*\*  
## PRES -9.821e-01 3.193e-03 -307.6 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.864 on 43822 degrees of freedom  
## Multiple R-squared: 0.6834, Adjusted R-squared: 0.6834   
## F-statistic: 9.46e+04 on 1 and 43822 DF, p-value: < 2.2e-16



##   
## Call:  
## lm(formula = TEMP ~ year + month + hour + day + pm2.5, data = Assignment\_2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -33.658 -10.168 0.899 10.326 27.472   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -7.672e+02 8.035e+01 -9.548 < 2e-16 \*\*\*  
## year 3.844e-01 3.994e-02 9.626 < 2e-16 \*\*\*  
## month 5.941e-01 1.638e-02 36.273 < 2e-16 \*\*\*  
## hour 2.612e-01 8.158e-03 32.016 < 2e-16 \*\*\*  
## day 2.845e-02 6.441e-03 4.416 1.01e-05 \*\*\*  
## pm2.5 -1.161e-02 6.161e-04 -18.838 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 11.82 on 43818 degrees of freedom  
## Multiple R-squared: 0.06129, Adjusted R-squared: 0.06119   
## F-statistic: 572.2 on 5 and 43818 DF, p-value: < 2.2e-16

##   
## Call:  
## lm(formula = TEMP ~ year + month + hour + day + DEWP, data = Assignment\_2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -17.543 -4.559 -0.653 3.954 33.208   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -7.656e+02 4.461e+01 -17.163 < 2e-16 \*\*\*  
## year 3.847e-01 2.217e-02 17.355 < 2e-16 \*\*\*  
## month -8.841e-02 9.349e-03 -9.457 < 2e-16 \*\*\*  
## hour 2.953e-01 4.529e-03 65.202 < 2e-16 \*\*\*  
## day -1.215e-02 3.564e-03 -3.409 0.000653 \*\*\*  
## DEWP 7.050e-01 2.235e-03 315.458 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.562 on 43818 degrees of freedom  
## Multiple R-squared: 0.7107, Adjusted R-squared: 0.7107   
## F-statistic: 2.153e+04 on 5 and 43818 DF, p-value: < 2.2e-16

##   
## Call:  
## lm(formula = TEMP ~ year + month + hour + day + PRES, data = Assignment\_2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -30.140 -3.943 0.465 4.528 20.846   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 377.199229 44.634713 8.451 <2e-16 \*\*\*  
## year 0.304548 0.022111 13.774 <2e-16 \*\*\*  
## month 0.422339 0.009082 46.501 <2e-16 \*\*\*  
## hour 0.204313 0.004520 45.205 <2e-16 \*\*\*  
## day 0.010741 0.003553 3.023 0.0025 \*\*   
## PRES -0.966873 0.003053 -316.698 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.544 on 43818 degrees of freedom  
## Multiple R-squared: 0.7123, Adjusted R-squared: 0.7122   
## F-statistic: 2.169e+04 on 5 and 43818 DF, p-value: < 2.2e-16

##   
## Call:  
## lm(formula = TEMP ~ year + month + hour + day + cbwd, data = Assignment\_2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -33.365 -10.094 1.062 10.142 30.786   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -8.432e+02 7.979e+01 -10.569 < 2e-16 \*\*\*  
## year 4.196e-01 3.965e-02 10.581 < 2e-16 \*\*\*  
## month 6.369e-01 1.629e-02 39.089 < 2e-16 \*\*\*  
## hour 2.274e-01 8.181e-03 27.789 < 2e-16 \*\*\*  
## day 1.743e-02 6.370e-03 2.736 0.00623 \*\*   
## cbwd 1.589e+00 5.014e-02 31.688 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 11.73 on 43818 degrees of freedom  
## Multiple R-squared: 0.07489, Adjusted R-squared: 0.07479   
## F-statistic: 709.4 on 5 and 43818 DF, p-value: < 2.2e-16

##   
## Call:  
## lm(formula = TEMP ~ month, data = Assignment\_2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -33.744 -10.329 1.061 10.265 30.468   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.52351 0.12287 69.37 <2e-16 \*\*\*  
## month 0.60167 0.01665 36.13 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 12.02 on 43822 degrees of freedom  
## Multiple R-squared: 0.02893, Adjusted R-squared: 0.02891   
## F-statistic: 1306 on 1 and 43822 DF, p-value: < 2.2e-16

##   
## Call:  
## lm(formula = TEMP ~ year + month + hour + day + pm2.5 + PRES,   
## data = Assignment\_2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -29.5873 -3.8853 0.4124 4.4246 19.8341   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.122e+02 4.337e+01 9.504 < 2e-16 \*\*\*  
## year 2.916e-01 2.149e-02 13.574 < 2e-16 \*\*\*  
## month 4.106e-01 8.828e-03 46.514 < 2e-16 \*\*\*  
## hour 1.991e-01 4.393e-03 45.330 < 2e-16 \*\*\*  
## day 2.596e-02 3.465e-03 7.493 6.87e-14 \*\*\*  
## pm2.5 -1.691e-02 3.318e-04 -50.960 < 2e-16 \*\*\*  
## PRES -9.742e-01 2.970e-03 -328.040 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.358 on 43817 degrees of freedom  
## Multiple R-squared: 0.7284, Adjusted R-squared: 0.7283   
## F-statistic: 1.958e+04 on 6 and 43817 DF, p-value: < 2.2e-16

##   
## Call:  
## lm(formula = TEMP ~ hour, data = Assignment\_2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -32.961 -10.730 1.304 10.574 28.626   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 9.407348 0.111714 84.21 <2e-16 \*\*\*  
## hour 0.264450 0.008323 31.77 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 12.06 on 43822 degrees of freedom  
## Multiple R-squared: 0.02252, Adjusted R-squared: 0.0225   
## F-statistic: 1010 on 1 and 43822 DF, p-value: < 2.2e-16

## [1] 343517.1

## [1] 342321.4

## [1] 343598.4

## [1] 342609.8

## [1] 343226.4

## [1] 293670.7

## [1] 293203

## [1] 340844

## [1] 289260.9

## [1] 289022

## [1] 340204.6

## [1] 280061.4

## [1] 286500.6

## [1] 339919.3

#12,13 models are the best among all the models

### Question(2)

2.You should build a Shiny app or dashboard allowing a scatterplot for any combination of variables to be displayed. Additionally, you should be able to generate histograms, boxplots etc. of your data in this app.

shiny is a package used to easily build interactive web apps straight from R To generate shiny we should follow three steps. They are as follows 1.Ui 2.server 3.shinyApp(UI, server)

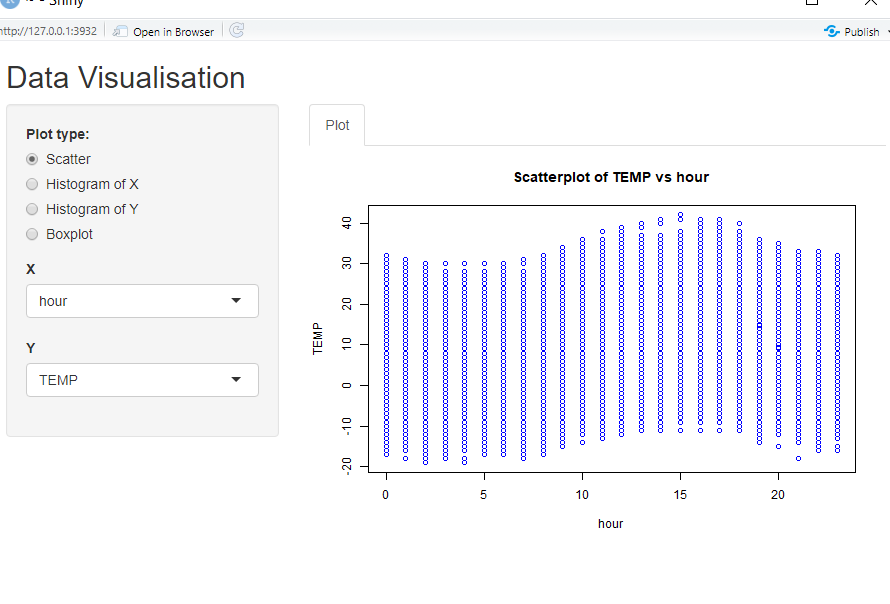
Ui Initially, we should create UI function i.e user interface, Generally user interface consists of title panel, sidebar layout, sidebar panel, main panel For this app, I gave the title as Data Visualisation. In the sidebar panel, I defined radio buttons for a scatterplot, histogram of x, histogram of y, boxplot and dropdown list menu for the x and y variables In the main panel, I defined tabsetpanel as plot all the plots output will appear in this tab

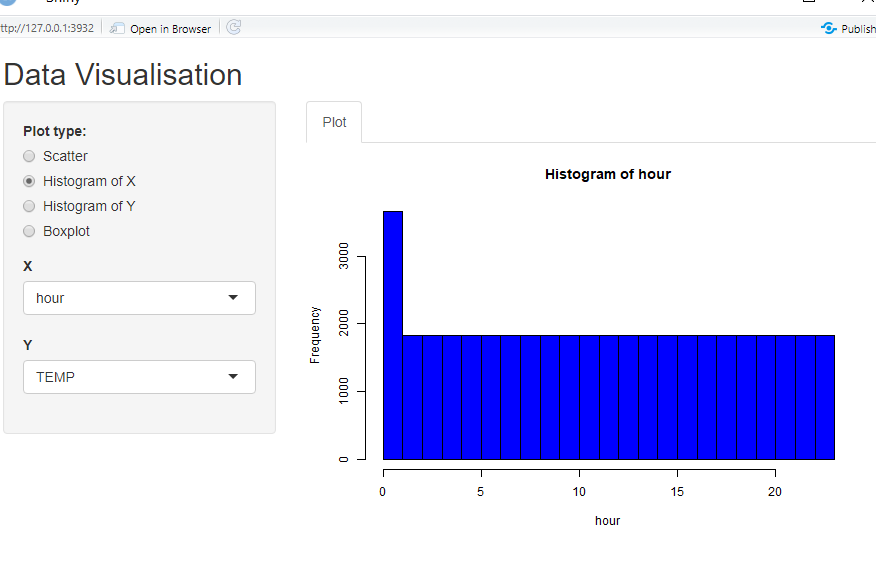
server in server function, I created outputtype is equal to “scat” the scatterplot will be executed or plotted on plot tab if the input$type is equal to "Histogram of x" the Histogram for x variable will be generated on plot tab if the input$type is equal to “Histogram of y” the Histogram for y variable will be generated on plot tab if the input$type is equal to “box” the boxplot will be generated on the plot tab

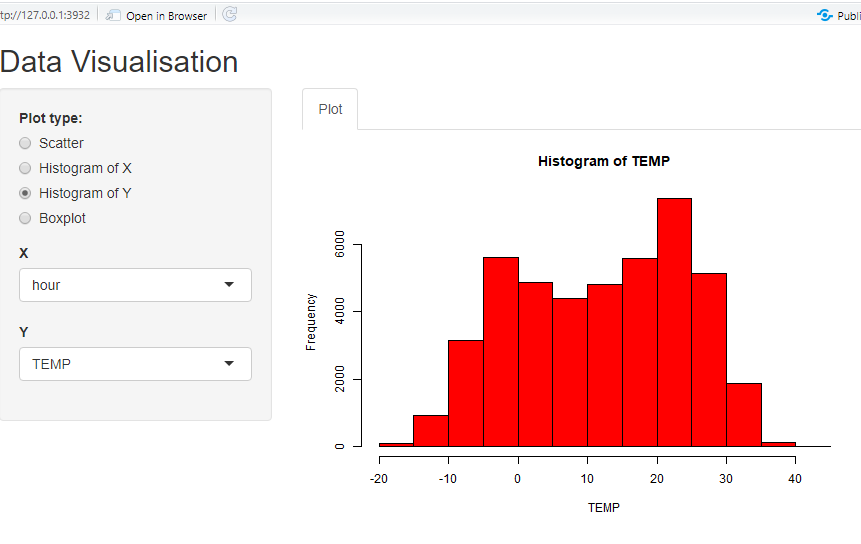
shinyAPP(UI, server) we will call the ui and server functions by using shinyAPP() function and generate the App

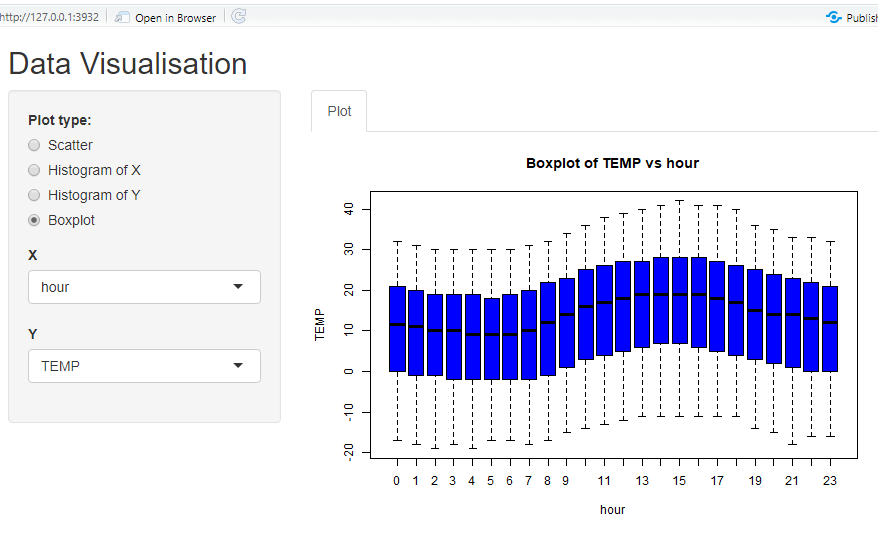
## PhantomJS not found. You can install it with webshot::install\_phantomjs(). If it is installed, please make sure the phantomjs executable can be found via the PATH variable.

Shiny applications not supported in static R Markdown documents









### Question(3)

1. You should include the ability to fit a linear regression model to the scatterplots generated in (2). The chart should include the fitted line and a table with the slope and intercept should be present within the Shiny App or dashboard.

To generate shiny we should follow three steps. They are as follows 1.Ui 2.server 3.shinyApp(UI, server)

Ui Initially, we should create UI function i.e user interface, Generally, the user interface consists of title panel, sidebar layout, sidebar panel, main panel.

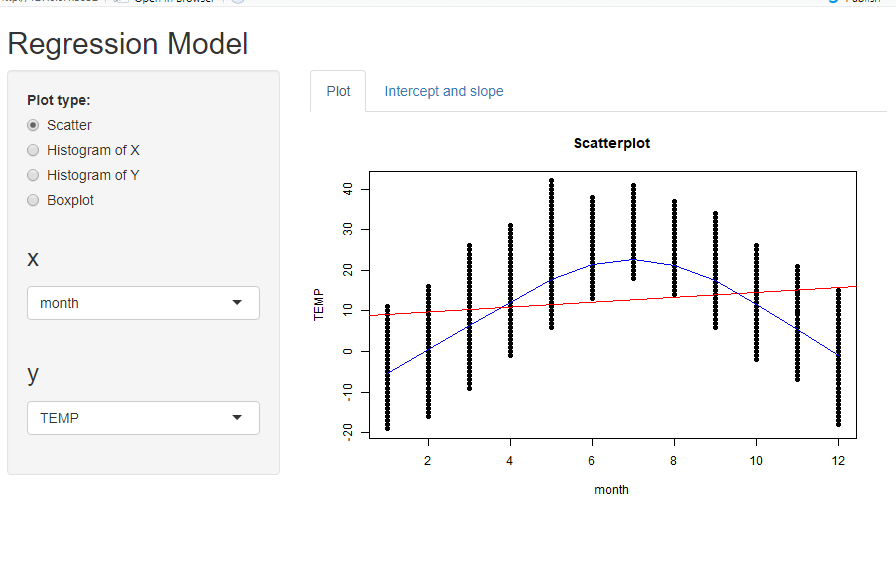
For this app, I gave the title as Regression ModelIn the sidebar panel, I defined radio buttons for a scatterplot, histogram of x, histogram of y, boxplot and dropdown list menu for the x and y variables.

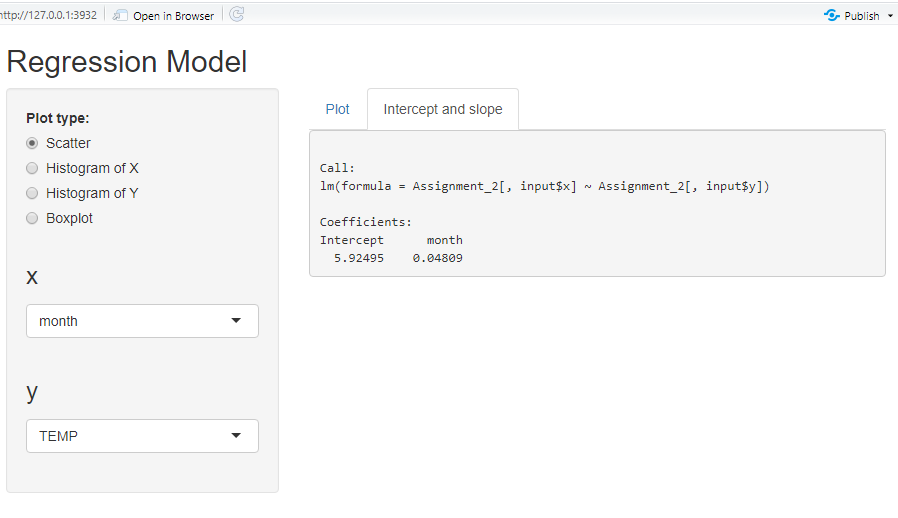
In the main panel, I defined tabsetpanel as plot in this tab all the plots output will appear in this tab and Intercept and slope tab the intercept and slope related to the x variable will appear here

Server In server function, I created outputplotxy all the plot will store in that variable if the input$type is equal to "scat" the scatterplot with the linear regression model line will be executed or plotted on plot tab if the input$type is equal to “Histogram of x” the Histogram for x variable will be generated on plot tab if the input$type is equal to "Histogram of y" the Histogram for y variable will be generated on plot tab if the input$type is equal to “box” the boxplot will be generated on the plot tab

shinyAPP(UI, server) we will call the UI and server functions by using shinyAPP() function and generate the App

Shiny applications not supported in static R Markdown documents





### Question(4)

4)Using Monte Carlo simulations, you should attempt to predict the temperature in subsequent years. This should be done using at least two different models (i.e. different collections of variables in part 1). You should clearly state which performs best.

In a simulation, we set the ground rules of a random process and then the computer uses random numbers to generate an outcome that adheres to those rules

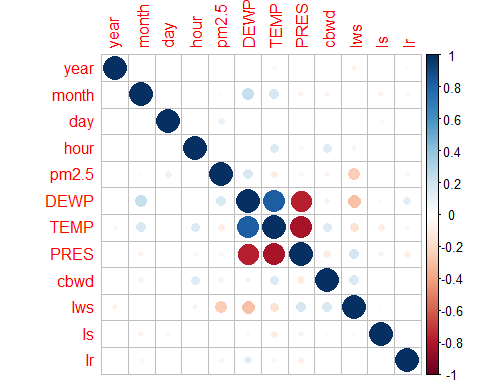
Monte Carlo simulations are used to model the probability of different outcomes in a process that cannot easily be predicted due to the intervention of random variables. It is a technique used to understand the impact of risk and uncertainty in prediction and forecasting models.

In this, I considered two best linear regression models which I did in question 1 and I found the mean absolute deviation for both the models and then I found model coefficients .later I found the stimulation for the model And then plotted the histograms for actual variables and stimulated values and both the histograms are nearly equal

For model one: I used the linear regression model TEMP~year+month+hour+day+pm2.5+DEWP I found the mean mean absolute deviation (mad) for the model TEMP~year+month+hour+day+pm2.5+DEWP and storing in disp variable And the model coefficients for the model TEMP~year+month+hour+day+pm2.5+DEWP and stored in the model considering Ysim variable and assigned NULL to it and performing the Monte Carlo simulation by running the loop 500 times After that generating the histograms for actual value i.e Assignment\_2$TEMP and hist(ysim) And checking both are simulated distributed and nearly the same.

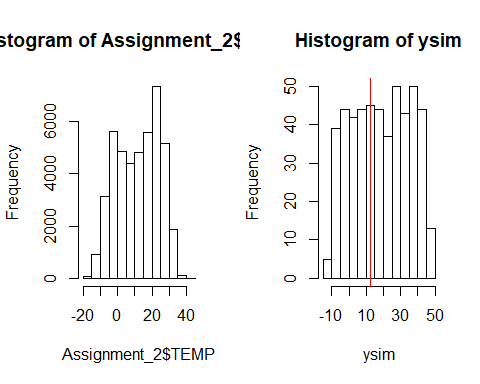
For model two: I used the linear regression model TEMP~year+month+hour+day+pm2.5+PRES I found the mean mean absolute deviation (mad) for the model TEMP~year+month+hour+day+pm2.5+PRESand storing in disp1 variable And the model coefficients for the model TEMP~year+month+hour+day+pm2.5+PRESDEWP and stored in model1 considering Ysim1 variable and assigned NULL to it and performing the Monte Carlo simulation by running the loop 500 times After that generating the histograms for actual value i.e Assignment\_2$TEMP and hist(ysim1) And checking both are simulated distribution and nearly the same.

Among both the models TEMP~year+month+hour+day+pm2.5+DEWP i.e first model is best and its performance is good, while plotting histograms of actual and stimulated values both the histograms are distribution was stimulated and nearly both are equal.This is the reson why the first model performance is good



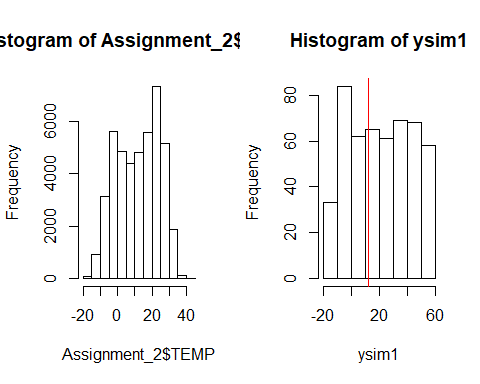
## [1] 5.732931

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -715.57910890 4.016322e+01 -17.816777 9.313742e-71  
## year 0.36142679 1.996160e-02 18.106102 5.270119e-73  
## month -0.14355730 8.434729e-03 -17.019788 9.466773e-65  
## hour 0.28803247 4.078379e-03 70.624252 0.000000e+00  
## day 0.01488688 3.219548e-03 4.623904 3.776577e-06  
## pm2.5 -0.03165471 3.128588e-04 -101.178930 0.000000e+00  
## DEWP 0.74157632 2.044396e-03 362.736143 0.000000e+00



## [1] 6.159766

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 412.22629890 4.337397e+01 9.504001 2.118050e-21  
## year 0.29164160 2.148508e-02 13.574143 6.929892e-42  
## month 0.41061031 8.827709e-03 46.513802 0.000000e+00  
## hour 0.19911679 4.392642e-03 45.329621 0.000000e+00  
## day 0.02596043 3.464758e-03 7.492712 6.873165e-14  
## pm2.5 -0.01690903 3.318069e-04 -50.960432 0.000000e+00  
## PRES -0.97424640 2.969904e-03 -328.039733 0.000000e+00



###Question(5)

5)We now consider two linear models: one where we consider y as temperature and x as pressure and a second with y as temperature and both pressure and wind speed as x variables. You should apply both models and calculate the ESS statistics on the data (some notes on this statistic can be found here: <https://www.graphpad.com/guides/prism/7/curvefitting/reg_howtheftestworks.htm?toc=0&printWindow> ). You are required to generate a distribution for this test statistic by simulation; you may assume the errors and residuals are normal. You should clearly state whether this model is an appropriate fit to the data based on your simulations

Initially, I consider Assignment\_2PRES as PRES And Assignment\_2$Iws as Iws I created model coefficient i.e TEMP~PRES and assigned it to model1 And then the mean absolute deviation (mad) for the model TEMP~PRES and storing in disp variable Now I created a model coefficient for TEMP~PRES+Iws and assigned it to model2 And then the mean absolute deviation (mad) for the model TEMP~PRES+Iws and storing in disp 1 variable Now find the sum of the square value for model1 and model2 The degree of freedom is defined as the number of data points minus the number of parameters So for model 1, there are two parameters so the degree of freedom is nrow(Assignment\_2)-2 For model2 there are three parameters so a degree of freedom is nrow(Assignment\_2)-3 The formula for the extra sum of squares of f\_test is F=((ss1-ss2)/ss2)/((df1-df2)/df2) The F value for the models 0.2776 The value of F is less than 1 so the simple model is correct And then I considered an empty vector and assigning it to variable a And considering rsss1,rsss2,ysim1,ysim2 and assigning NULL to them and applying the stimulation for both model 1 and model2 and calculating inside the loop for 500 times and plotting the histograms. The simple model is correct to fit for the data based on the stimulation

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 1010.6609064 3.245637284 311.3906 0  
## PRES -0.9820598 0.003192955 -307.5708 0

## [1] 6.701746

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 1.010347e+03 3.2999749899 306.1679928 0.0000000  
## PRES -9.817424e-01 0.0032492860 -302.1409711 0.0000000  
## Iws -3.515686e-04 0.0006671768 -0.5269497 0.5982312

## [1] 6.709858

## [1] 43822

## [1] 43821

## [1] 0.277676

