J.N.T.U.H. UNIVERSITY COLLEGE OF ENGINEERING, SCIENCE & TECHNOLOGY HYDERABAD

KUKATPALLY, HYDERABAD – 500 085

Certificate

Certified that this is the bonafide record of the practical work done during

the academic year **2024-2025** *by*

Name **Thakur Pratyush Singh**

Roll Number **21011A0550** *Class* **B.Tech IV Year I Semester**

in the Laboratory of **Data Analytics**

of the Department of **Computer Science and Engineering**

Signature of the Staff Member *Signature of the Head of the Department*

Date of Examination **18/11/2024**

Signature of the Examiner/s

Internal Examiner *External Examiner*

J.N.T.U.H. UNIVERSITY COLLEGE OF ENGINEERING, SCIENCE & TECHNOLOGY HYDERABAD

KUKATPALLY, HYDERABAD – 500 085

Name **Thakur Pratyush Singh** *Roll Number* **21011A0550**

Class **CSE-Regular** *Year* **IV** *Laboratory* **Data Analytics**

List of Experiments

S.No.	Name of the Experiment	Date of Experiment	Page Number	Marks	Remarks
1.	Demonstrate Data Cleaning – missing values	24/07/24	1-2		
2.	Implement Data Normalization (Min-Max and Z-Score)	31/07/24	3		
3.	Implement Attribute Subset Selection for Data Reduction	07/08/24	4-6		
4.	Demonstrate Outlier Detection	14/08/24	7-8		
5.	MongoDB Installation and basic commands	21/08/24	9-12		
6.	Perform Analytics on any standard data set	04/09/24	13-16		
7.	Implement Linear Regression	18/09/24	17-18		
8.	Implement Logistic Regression	25/09/24	19-22		
9.	Construct Decision Tree for weather data set	16/10/24	23-24		
10.	Analyze Time-Series Data	23/10/24	25-26		
11.	Work on any Data Visualization tools	30/10/24	27-29		

1. Demonstrate Data Cleaning – missing values

```
library(tidyverse)

x <- sample(1:21, 20, replace = TRUE)
y <- sample(1:10, 20, replace = TRUE)

for(i in 1:20){
  a <- x[i]
  b <- y[i]
  mtcars[a,b] = NA
}

which(is.na(mtcars))

sum(is.na(mtcars))

na.exclude(mtcars)

View(mtcars)
dispna <- apply(mtcars["disp"],2,mean,na.rm=TRUE)
View(dispna)

newcars <- mtcars %>%
  mutate(disp=ifelse(is.na(disp), dispna, disp))
View(newcars)
```

Output:

Name	Type	Value
dispna	double [1]	235.7226
disp	double [1]	235.7226

Honda Civic	30.4	4	*NA*	52	4.93	1.615	18.52	*NA*	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	*NA*	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2

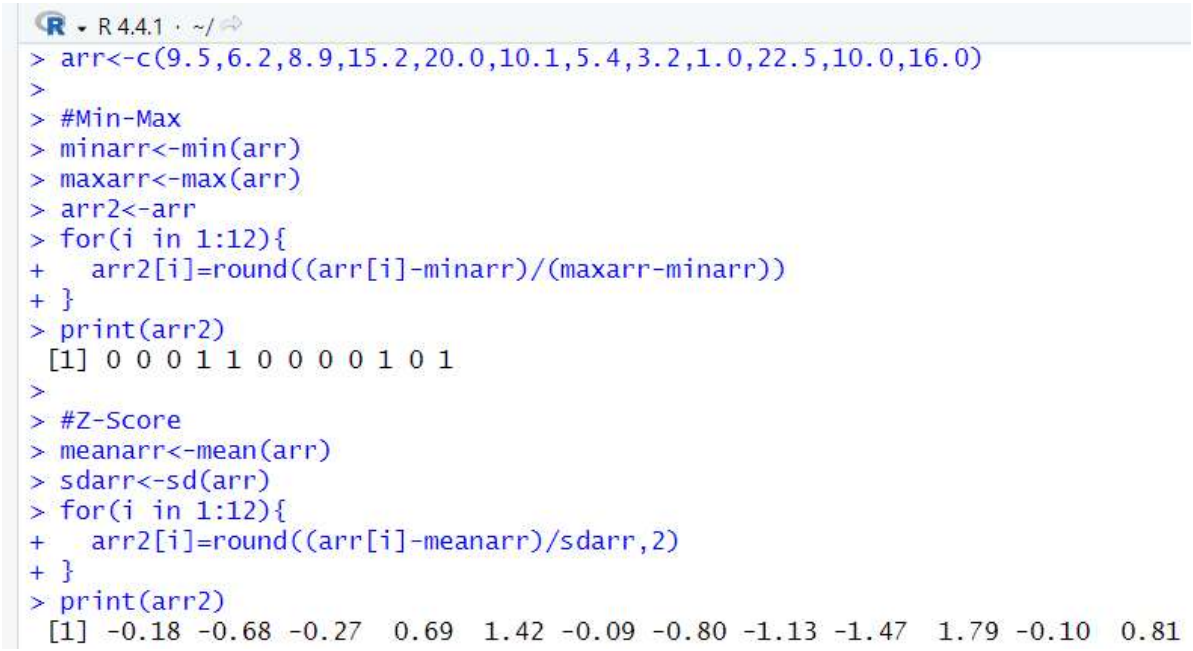
Dodge Challenger	15.5	8	318.0000	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0000	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0000	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0000	175	3.08	3.845	17.05	0	0	3	2
Fiat X1-9	27.3	4	79.0000	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3000	91	4.43	2.140	16.70	0	1	5	2

```
R 4.4.1 · ~/
> library(tidyverse)
>
> x <- sample(1:21, 20, replace = TRUE)
> y <- sample(1:10, 20, replace = TRUE)
>
> for(i in 1:20){
+   a <- x[i]
+   b <- y[i]
+   mtcars[a,b] = NA
+ }
>
> which(is.na(mtcars))
 [1]  21  40  83 108 134 173 200 226 228 237 243 268 270 271 289 292 295 299
>
> sum(is.na(mtcars))
[1] 18
>
> na.exclude(mtcars)
                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82  0  0    3    4
Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
Fiat 128          32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Toyota Corolla    33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
Dodge Challenger  15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
AMC Javelin       15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
Camaro Z28        13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
Pontiac Firebird  19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
Fiat X1-9         27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
Porsche 914-2     26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
Lotus Europa      30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
Ford Pantera L    15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
Ferrari Dino      19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
Maserati Bora     15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
Volvo 142E        21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

2. Implement Data Normalization (Min-Max and Z-Score)

```
arr<-c(9.5,6.2,8.9,15.2,20.0,10.1,5.4,3.2,1.0,22.5,10.0,16.0)

#Min-Max
minarr<-min(arr)
maxarr<-max(arr)
arr2<-arr
for(i in 1:12){
  arr2[i]=round((arr[i]-minarr)/(maxarr-minarr))
}
print(arr2)

#Z-Score
meanarr<-mean(arr)
sdarr<-sd(arr)
for(i in 1:12){
  arr2[i]=round((arr[i]-meanarr)/sdarr,2)
}
print(arr2)
```

Output:



3. Implement Attribute Subset Selection for Data Reduction

```
library(leaps)

View(Titanic)
sum(is.na(Titanic))
Titanic <- na.omit(Titanic)
dim(Titanic)

fwd <- regsubsets(Freq~., data = Titanic, nvmax = 19, method = "forward")
summary(fwd)
coef(fwd, 3)

bwd <- regsubsets(Freq~., data = Titanic, nvmax = 19, method = "backward")
summary(bwd)
coef(bwd, 3)

full <- regsubsets(Freq~., data = Titanic, nvmax = 19)
summary(full)
coef(full, 3)
```

Output:

	Class	Sex	Age	Survived	Freq
1	1st	Male	Child	No	0
2	2nd	Male	Child	No	0
3	3rd	Male	Child	No	35
4	Crew	Male	Child	No	0
5	1st	Female	Child	No	0
6	2nd	Female	Child	No	0
7	3rd	Female	Child	No	17
8	Crew	Female	Child	No	0
9	1st	Male	Adult	No	118
10	2nd	Male	Adult	No	154

```
R 4.4.1 · ~/
> library(leaps)
Warning message:
package 'leaps' was built under R version 4.4.2
>
> View(Titanic)
> sum(is.na(Titanic))
[1] 0
> Titanic <- na.omit(Titanic)
> dim(Titanic)
[1] 4 2 2 2
>
> fwd <- regsubsets(Freq~., data = Titanic, nvmax = 19, method = "forward")
> summary(fwd)
Subset selection object
Call: regsubsets.formula(Freq ~ ., data = Titanic, nvmax = 19, method = "forward")
6 Variables  (and intercept)
           Forced in Forced out
Class2nd       FALSE      FALSE
Class3rd       FALSE      FALSE
ClassCrew      FALSE      FALSE
SexFemale      FALSE      FALSE
AgeAdult       FALSE      FALSE
SurvivedYes    FALSE      FALSE
1 subsets of each size up to 6
Selection Algorithm: forward
         Class2nd Class3rd ClassCrew SexFemale AgeAdult SurvivedYes
1  ( 1 ) " "      " "      " "       " "       "*"      " "
2  ( 1 ) " "      " "      " "       "*"       "*"      " "
3  ( 1 ) " "      " "      " "       "*"       "*"      "*"
4  ( 1 ) " "      " "      "*"       "*"       "*"      "*"
5  ( 1 ) " "      "*"      "*"       "*"       "*"      "*"
6  ( 1 ) "*"      "*"      "*"       "*"       "*"      "*"
> coef(fwd, 3)
(Intercept)   SexFemale    AgeAdult SurvivedYes
    70.5625    -78.8125    123.9375    -48.6875
> bwd <- regsubsets(Freq~., data = Titanic, nvmax = 19, method = "backward")
> summary(bwd)
Subset selection object
Call: regsubsets.formula(Freq ~ ., data = Titanic, nvmax = 19, method = "backward")
6 Variables  (and intercept)
           Forced in Forced out
Class2nd       FALSE      FALSE
Class3rd       FALSE      FALSE
ClassCrew      FALSE      FALSE
SexFemale      FALSE      FALSE
AgeAdult       FALSE      FALSE
SurvivedYes    FALSE      FALSE
1 subsets of each size up to 6
Selection Algorithm: backward
         Class2nd Class3rd ClassCrew SexFemale AgeAdult SurvivedYes
1  ( 1 ) " "      " "      " "       " "       "*"      " "
2  ( 1 ) " "      " "      " "       "*"       "*"      " "
3  ( 1 ) " "      " "      " "       "*"       "*"      "*"
4  ( 1 ) " "      " "      "*"       "*"       "*"      "*"
5  ( 1 ) " "      "*"      "*"       "*"       "*"      "*"
6  ( 1 ) "*"      "*"      "*"       "*"       "*"      "*"
> coef(bwd, 3)
(Intercept)   SexFemale    AgeAdult SurvivedYes
    70.5625    -78.8125    123.9375    -48.6875
```

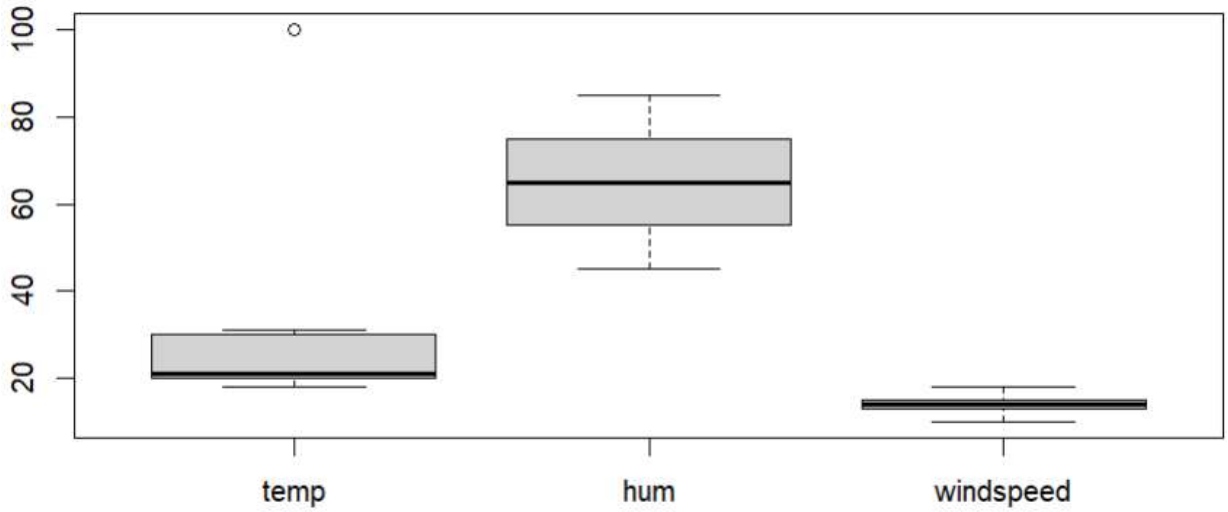
```
> full <- regsubsets(Freq~., data = Titanic, nvmax = 19)
> summary(full)
Subset selection object
Call: regsubsets.formula(Freq ~ ., data = Titanic, nvmax = 19)
6 Variables  (and intercept)
           Forced in Forced out
Class2nd       FALSE      FALSE
Class3rd       FALSE      FALSE
ClassCrew      FALSE      FALSE
SexFemale      FALSE      FALSE
AgeAdult       FALSE      FALSE
SurvivedYes    FALSE      FALSE
1 subsets of each size up to 6
Selection Algorithm: exhaustive
         Class2nd Class3rd ClassCrew SexFemale AgeAdult SurvivedYes
1  ( 1 ) " "      " "      " "       " "       "*"      " "
2  ( 1 ) " "      " "      " "       "*"       "*"      " "
3  ( 1 ) " "      " "      " "       "*"       "*"      "*"
4  ( 1 ) " "      " "      "*"       "*"       "*"      "*"
5  ( 1 ) " "      "*"      "*"       "*"       "*"      "*"
6  ( 1 ) "*"      "*"      "*"       "*"       "*"      "*"
> coef(full, 3)
(Intercept)   SexFemale    AgeAdult SurvivedYes
    70.5625    -78.8125    123.9375    -48.6875
```

4. Demonstrate Outlier Detection

```
day <- data.frame(
  temp = c(20,21,22,20,19,30,31,100,18,33),
  hum = c(55,60,65,50,45,70,75,80,85,200),
  windspeed = c(12,15,14,16,14,10,18,13,15,50)
)
View(day)

boxplot(day[,c('temp','hum','windspeed')])

for(i in c('hum','windspeed')){
  data <- unlist(day[i])
  newdata <- data[data %in% boxplot.stats(data)$out]
  data[data %in% newdata] <- NA
  day[[i]] <- data
}

day <- drop_na(day)

boxplot(day[,c('temp','hum','windspeed')])
```

Output:

	temp	hum	windspeed
1	20	55	12
2	21	60	15
3	22	65	14
4	20	50	16
5	19	45	14
6	30	70	10
7	31	75	18
8	100	80	13
9	18	85	15


100
80
60
40
20
temp
hum
windspeed

5. MongoDB Installation and basic commands

MongoDB Installation:

Step 1: Download MongoDB Community Server

a. Visit the MongoDB Download Center
   - Go to MongoDB Community Server Download.

b. Select the Installer:
   - **Version**: Choose the latest stable version
   - **Platform**: Windows
   - **Package**: MSI

c. Download the MSI Installer

Step 2: Install MongoDB

a. Run the MongoDB Installer
   - Locate the downloaded .msi file and double-click it.

b. Follow the Installation Wizard
   - **License Agreement**: Accept the License Agreement.
   - **Setup Type**: Select Complete.

c. Component Selection: Ensure the following options are checked:
   - **MongoDB Server**: Core database server.
   - **MongoDB Shell (mongosh)**: Interactive shell for MongoDB.

d. Service Configuration:
   - Choose Run Service as Network Service User (default).
   - Set the Data Directory and Log Directory (defaults are recommended).

e. Finish Installation:
   - Click Install and wait for the process to complete.
   - Once done, click Finish.

Step 3: Add MongoDB to System PATH

a. Edit Environment Variables
   - Press Win + S, type Environment Variables, and select 'Edit' the system environment variables.
   - In the System Properties window, click Environment Variables.

b. Update PATH
   - Under System variables, find and select the 'Path' variable.
   - Click Edit > New.
   - Add the MongoDB 'bin' directory.
   - Click OK to save changes.

Step 4: Verify Installation

a. Check MongoDB Server Version
   - Run ‘mongod --version’

b. Check MongoDB Shell Version
   - Run ‘mongosh --version’

c. Connect to MongoDB via MongoDB Shell
   - In Command Prompt, type ‘mongosh’
   - This will open an interactive shell session connected to your local MongoDB instance.

Basic Commands:

```
test> use blog
switched to db blog
blog> db.createCollection("posts")
{ ok: 1 }
blog> db.createCollection("users")
{ ok: 1 }
blog> db.posts.insertOne({
... title: "Introduction to MongoDB",
... content: "MongoDB is a NoSQL database.",
... author: "John Doe",
... tags: ["mongodb", "nosql", "database"]
... })
{
  acknowledged: true,
  insertedId: ObjectId('66f371af3edbac1d74c73bf8')
}
```

```
blog> db.users.insertMany([ { username: "johndoe"
... , email: "johndoe@ex.com",
... age: 30
... },
... {
... username: "jane",
... email: "jane@ex.com",
... age: 28
... }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66f372723edbac1d74c73bf9'),
    '1': ObjectId('66f372723edbac1d74c73bfa')
  }
}
blog> db.users.updateOne(
...     { username: "johndoe" },
...     { $set: { age: 31 } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
blog> db.posts.updateMany(
...     { tags: "mongodb" },
...     { $addToSet: { tags: "database" } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
```

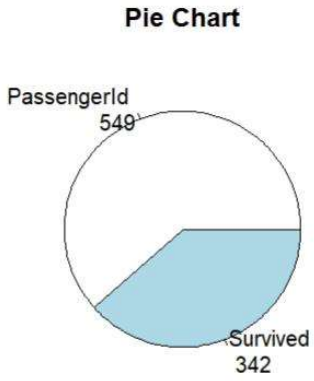
```
blog> db.users.deleteOne({ username: "janedoe" })
{ acknowledged: true, deletedCount: 0 }
blog> db.posts.deleteMany({ author: "John Doe" })
{ acknowledged: true, deletedCount: 1 }
blog> db.users.drop()
true
blog> db.posts.find()

blog> show collections
posts
blog>
```

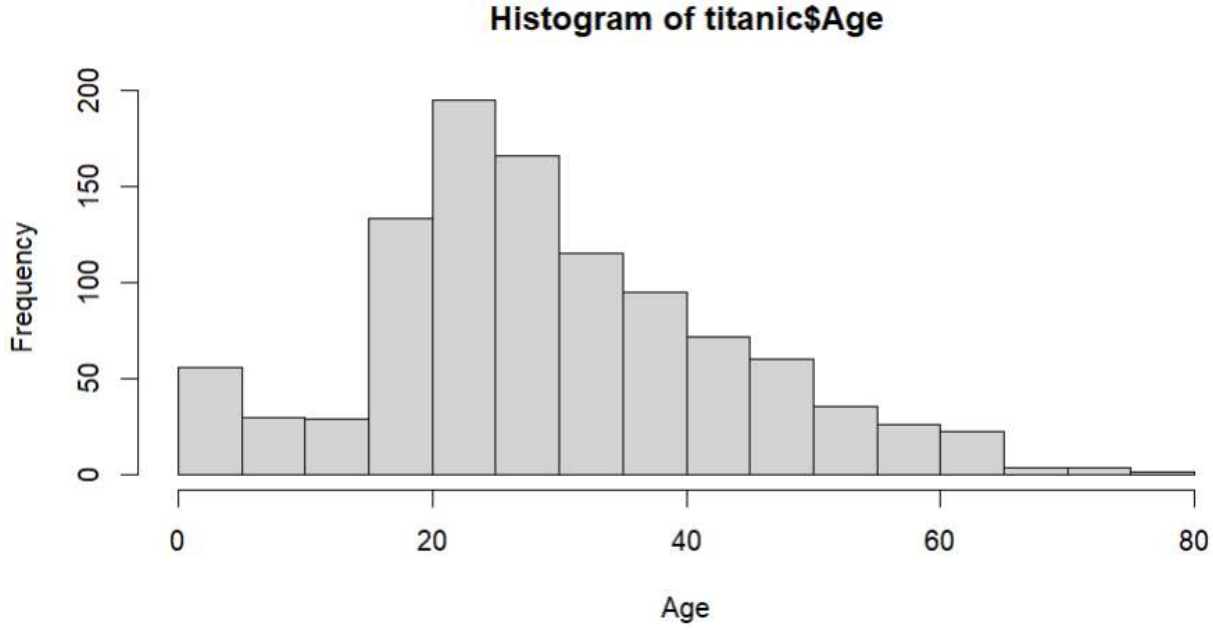
6. Perform Analytics on any standard data set

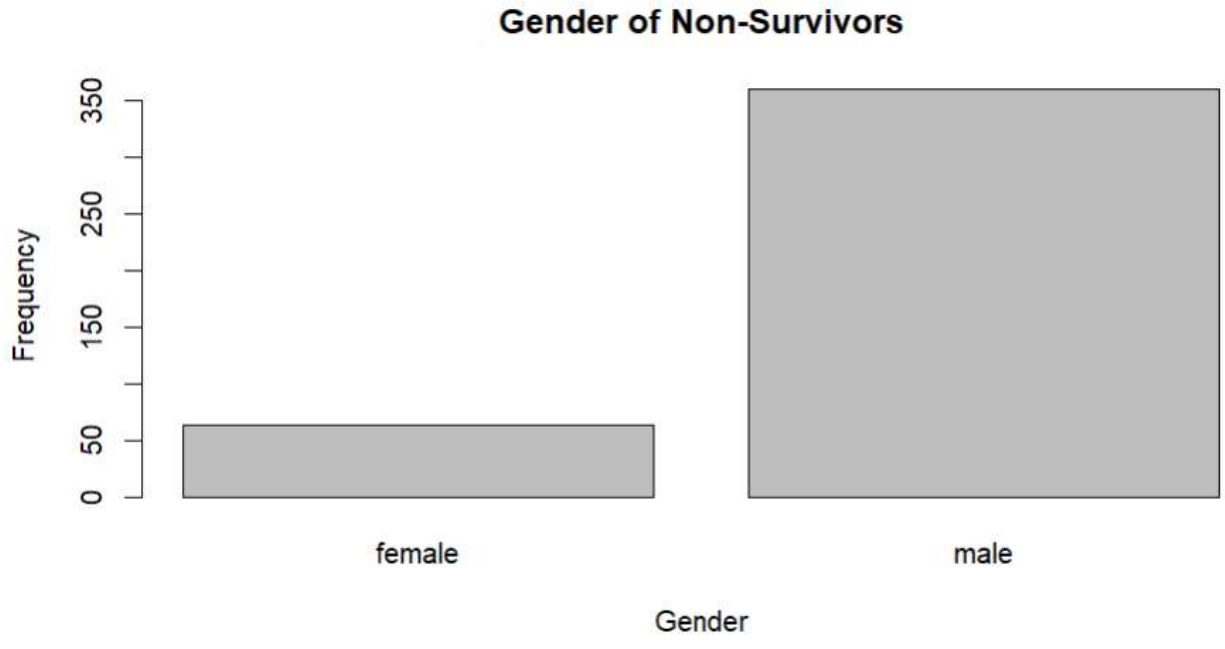
```
library(tidyverse)
library(titanic)

data("titanic_train")
data("titanic_test")
titanic_test$Survived <- NA
titanic <- rbind(titanic_train, titanic_test)
head(titanic)

titanic$Sex = as.factor(titanic$Sex)
titanic$Survived = as.factor(titanic$Survived)

summary(titanic)

dropnull_titanic <- titanic[rowSums(is.na(titanic)) <= 0, ]
survived_list <- dropnull_titanic[dropnull_titanic$Survived == 1, ]
notsurvived_list <- dropnull_titanic[dropnull_titanic$Survived == 0, ]

mytable <- table(titanic$Survived)
lbls <- paste(names(titanic), "\n", mytable, sep = " ")

pie(mytable, labels = lbls, main = 'Pie Chart')

hist(titanic$Age, xlab = 'Age', ylab = 'Frequency')

barplot(table(notsurvived_list$Sex), main = 'Gender of Non-Survivors', xlab = 'Gender', ylab =
'Frequency')

temp <- density(table(survived_list$Fare))
plot(temp, type = 'n', main = 'Fare Charged')
polygon(temp, col = 'lightgray', border = 'gray')

boxplot(titanic$Fare, main = 'Fare')
```

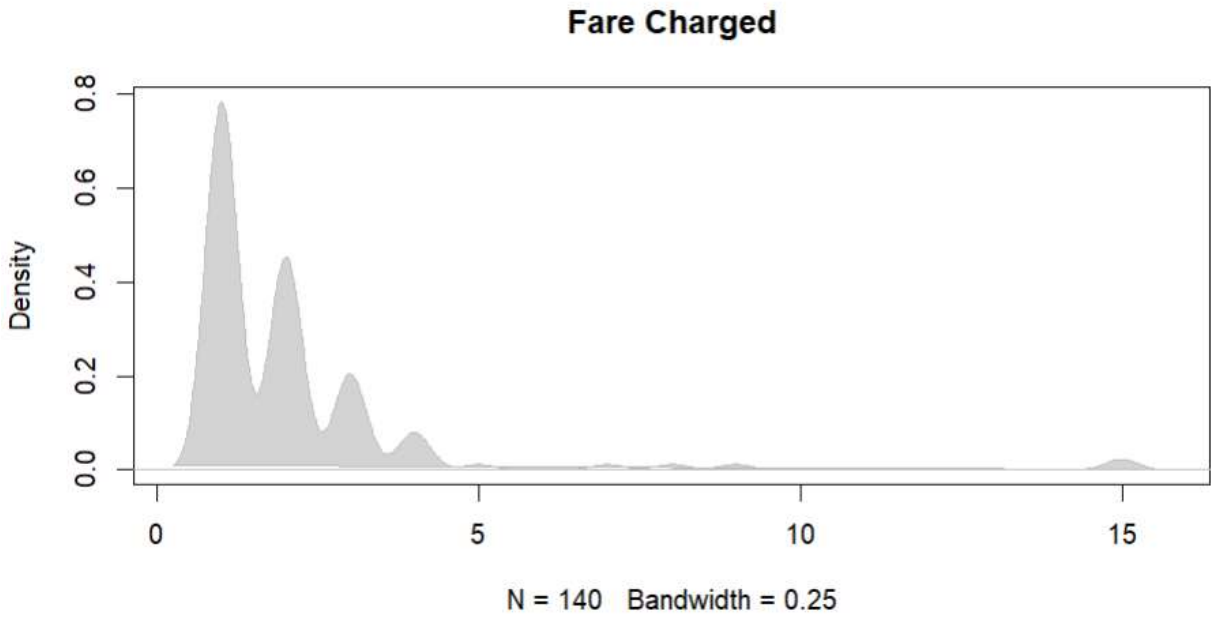
Output:

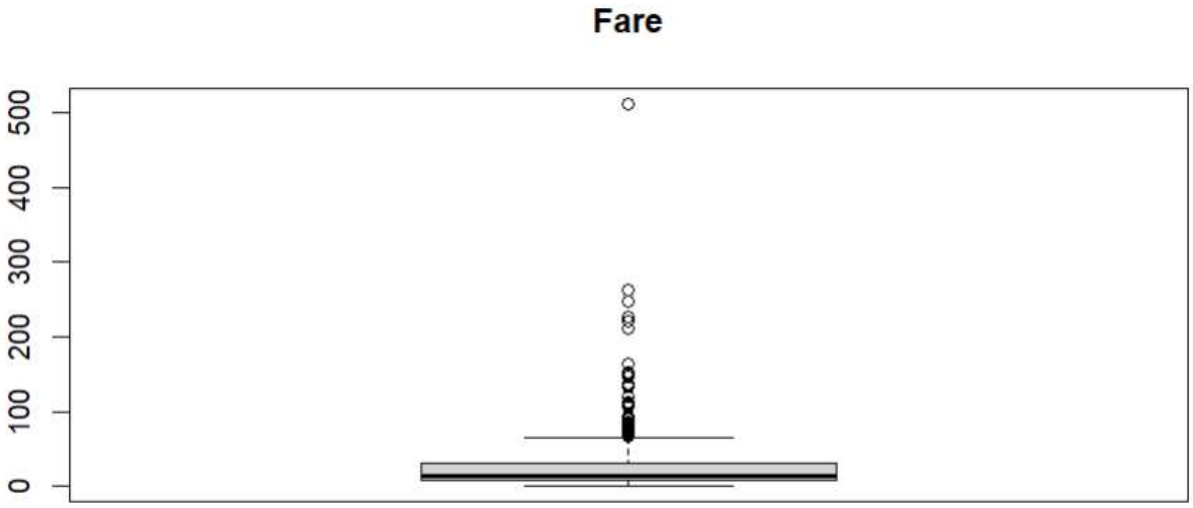
```
R 4.4.1 · ~/
> library(tidyverse)
> library(titanic)
>
> data("titanic_train")
> data("titanic_test")
> titanic_test$Survived <- NA
> titanic <- rbind(titanic_train, titanic_test)
> head(titanic)
  PassengerId Survived Pclass                                                Name
1           1        0      3                             Braund, Mr. Owen Harris
2           2        1      1 Cumings, Mrs. John Bradley (Florence Briggs Thayer)
3           3        1      3                              Heikkinen, Miss. Laina
4           4        1      1        Futrelle, Mrs. Jacques Heath (Lily May Peel)
5           5        0      3                            Allen, Mr. William Henry
6           6        0      3                                    Moran, Mr. James
     Sex Age SibSp Parch            Ticket    Fare Cabin Embarked
1   male  22     1     0         A/5 21171  7.2500              S
2 female  38     1     0          PC 17599 71.2833   C85        C
3 female  26     0     0 STON/O2. 3101282  7.9250              S
4 female  35     1     0           113803 53.1000  C123        S
5   male  35     0     0           373450  8.0500              S
6   male  NA     0     0           330877  8.4583              Q
> titanic$Sex = as.factor(titanic$Sex)
> titanic$Survived = as.factor(titanic$Survived)
>
> summary(titanic)
  PassengerId   Survived      Pclass          Name               Sex
 Min.   :   1   0   :549   Min.   :1.000   Length:1309        female:466
 1st Qu.: 328   1   :342   1st Qu.:2.000   Class :character   male  :843
 Median : 655   NA's:418   Median :3.000   Mode  :character
 Mean   : 655              Mean   :2.295
 3rd Qu.: 982              3rd Qu.:3.000
 Max.   :1309              Max.   :3.000

      Age            SibSp           Parch          Ticket             Fare
 Min.   : 0.17   Min.   :0.0000   Min.   :0.000   Length:1309        Min.   :  0.000
 1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000   Class :character   1st Qu.:  7.896
 Median :28.00   Median :0.0000   Median :0.000   Mode  :character   Median : 14.454
 Mean   :29.88   Mean   :0.4989   Mean   :0.385                      Mean   : 33.295
 3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000                      3rd Qu.: 31.275
 Max.   :80.00   Max.   :8.0000   Max.   :9.000                      Max.   :512.329
 NA's   :263                                                         NA's   :1
    Cabin             Embarked
 Length:1309        Length:1309
 Class :character   Class :character
 Mode  :character   Mode  :character
```


Pie Chart
PassengerId
549
Survived
342


Histogram of titanic$Age
Frequency
0
50
100
150
200
0
20
40
60
80
Age


Gender of Non-Survivors
Frequency
0
50
150
250
350
female
male
Gender


Fare Charged
Density
0.0
0.2
0.4
0.6
0.8
0
5
10
15
N = 140 Bandwidth = 0.25

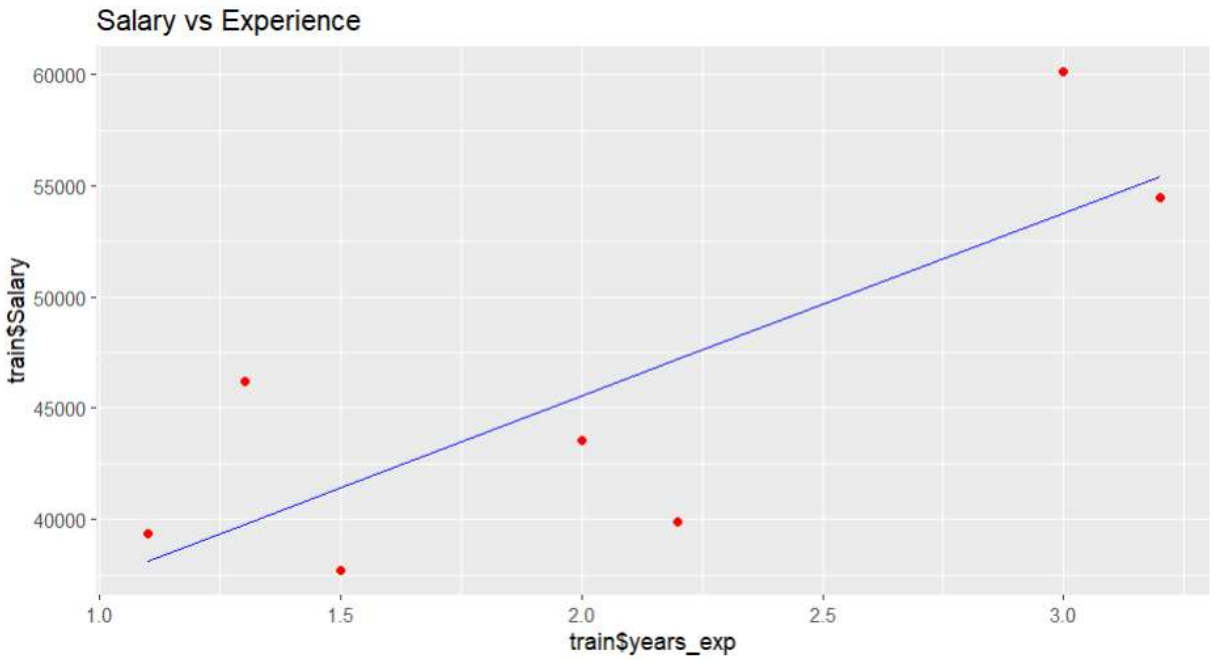

Fare
0
100
200
300
400
500

7. Implement Linear Regression

```
data <- data.frame(
  years_exp = c(1.1, 1.3, 1.5, 2.0, 2.2, 2.9, 3.0, 3.2, 3.2, 3.7),
  Salary = c(39343.00, 46205.00, 37731.00, 43525.00, 39891.00, 56642.00, 60150.00, 54445.00,
64445.00, 57189.00)
)

library(caTools)
library(ggplot2)

split <- sample.split(data$Salary, SplitRatio = 0.7)
train <- subset(data, split == TRUE)
test <- subset(data, split == FALSE)
lm.r <- lm(formula = Salary ~ years_exp, data = train)
coef(lm.r)

# Corrected ggplot code
ggplot() +
  geom_point(aes(x = train$years_exp, y = train$Salary), col = 'red') +
  geom_line(aes(x = train$years_exp, y = predict(lm.r, newdata = train)), col = 'blue') +
  ggtitle('Salary vs Experience')
```

Output:

```
R 4.4.1 · ~/
> data <- data.frame(
+   years_exp = c(1.1, 1.3, 1.5, 2.0, 2.2, 2.9, 3.0, 3.2, 3.2, 3.7),
+   Salary = c(39343.00, 46205.00, 37731.00, 43525.00, 39891.00, 56642.00, 60150.00, 54445.0
0, 64445.00, 57189.00)
+ )
>
> library(caTools)
> library(ggplot2)
>
> split <- sample.split(data$Salary, SplitRatio = 0.7)
> train <- subset(data, split == TRUE)
> test <- subset(data, split == FALSE)
> lm.r <- lm(formula = Salary ~ years_exp, data = train)
> coef(lm.r)
(Intercept)   years_exp
  29078.818    8233.446
>
> # Corrected ggplot code
> ggplot() +
+   geom_point(aes(x = train$years_exp, y = train$Salary), col = 'red') +
+   geom_line(aes(x = train$years_exp, y = predict(lm.r, newdata = train)), col = 'blue') +
+   ggtitle('Salary vs Experience')
>
```


Salary vs Experience
60000
55000
50000
45000
40000
train$Salary
1.0
1.5
2.0
2.5
3.0
train$years_exp

8. Implement Logistic Regression

```
library(tidyverse)
library(ROCR)
library(caTools)

View(mtcars)

split <- sample.split(mtcars, SplitRatio = 0.8)
train <- subset(mtcars, split == TRUE)
test <- subset(mtcars, split == FALSE)

logistic_model <- glm(vs~wt + disp, data = train, family = binomial)
summary(logistic_model)

predict_reg <- predict(logistic_model, test, type = "response")
predict_reg

predict_reg <- ifelse(predict_reg > 0.5, 1, 0)
table(test$vs, predict_reg)

missing_classerr <- mean(predict_reg != test$vs)
missing_classerr

print(paste("Accuracy = ", 100 * (1 - missing_classerr)))

library(ggplot2)
ggplot(mtcars, aes(x = wt + disp, y = vs))+
  geom_point(alpha = 0.5)+
  stat_smooth(method = "glm", se = FALSE, method.args = list(family = binomial), col = 'red')

library(ROCR)
ROCPred = prediction(predict_reg, test$vs)
ROCPer = performance(ROCPred, measure = 'tpr', x.measure = 'fpr')
auc <- performance(ROCPred, measure = 'auc')
auc <- auc@y.values[[1]]
auc

plot(ROCPer, colorize = TRUE, print.cutoffs.at = seq(0.1, by = 0.1), main = "ROC Curve")
abline(a = 0, b = 1)
auc <- round(auc, 4)
legend(.6, .4, auc, title = "AUC", cex = 1)
```

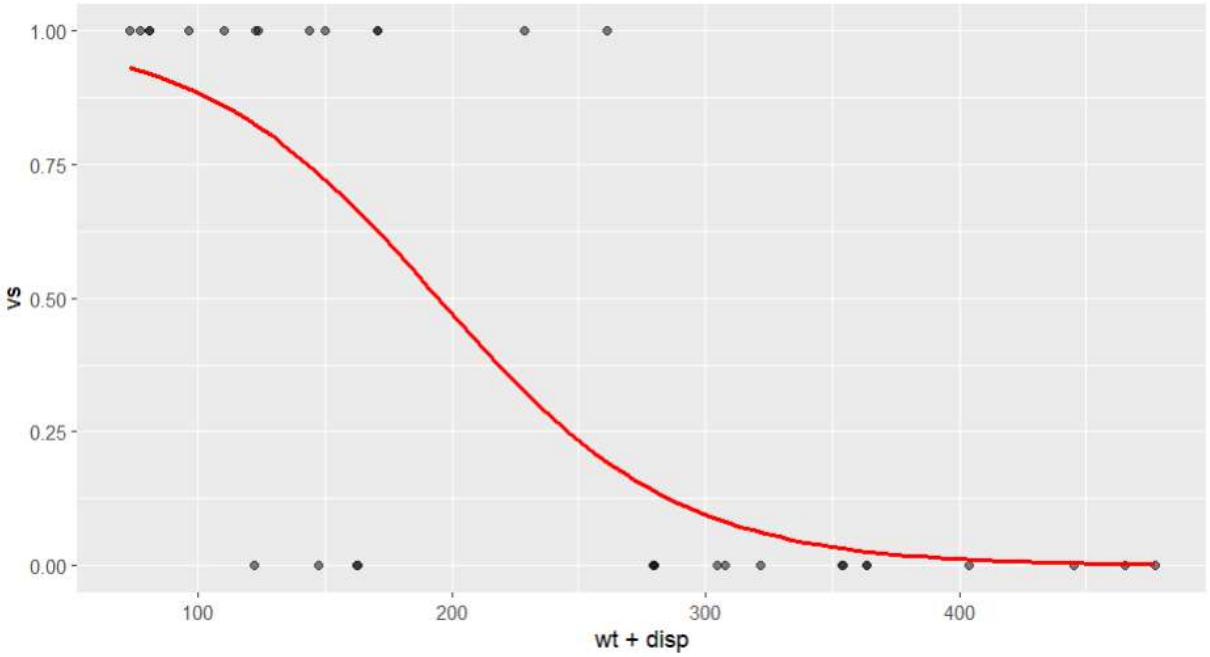
Output:

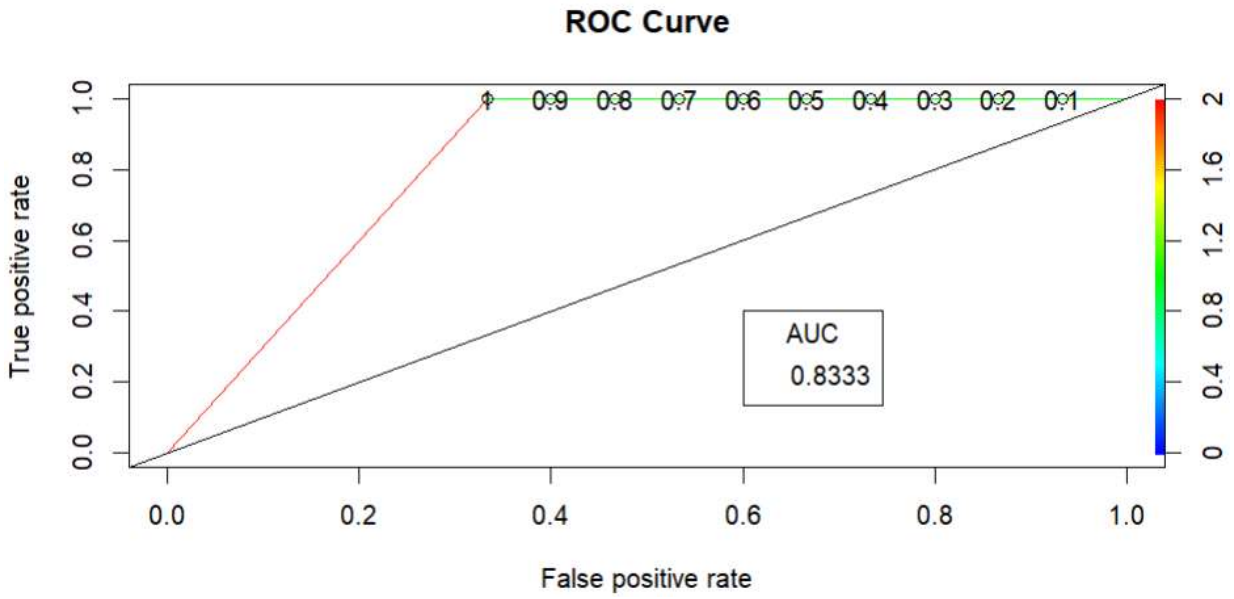
	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2

R 4.4.1 · ~/

```
> library(tidyverse)
> library(ROCR)
> library(caTools)
>
> View(mtcars)
>
> split <- sample.split(mtcars, SplitRatio = 0.8)
> train <- subset(mtcars, split == TRUE)
> test <- subset(mtcars, split == FALSE)
>
> logistic_model <- glm(vs~wt + disp, data = train, family = binomial)
> summary(logistic_model)

Call:
glm(formula = vs ~ wt + disp, family = binomial, data = train)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.10933    3.25655   0.955   0.3397
wt           1.12527    1.66397   0.676   0.4989
disp        -0.03073    0.01513  -2.032   0.0422 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 31.492  on 22  degrees of freedom
Residual deviance: 14.387  on 20  degrees of freedom
AIC: 20.387

Number of Fisher Scoring iterations: 6
```

```
> predict_reg <- predict(logistic_model, test, type = "response")
> predict_reg
      Mazda RX4 Wag          Datsun 710   Hornet Sportabout          Merc 450SL
        0.806431360         0.916876588         0.016559439         0.236865438
        Merc 450SLC Lincoln Continental          Camaro Z28    Pontiac Firebird
        0.247185712         0.007210381         0.034668186         0.007708147
      Porsche 914-2
        0.860573996
>
> predict_reg <- ifelse(predict_reg > 0.5, 1, 0)
> table(test$vs, predict_reg)
   predict_reg
    0 1
  0 6 2
  1 0 1
>
> missing_classerr <- mean(predict_reg != test$vs)
> missing_classerr
[1] 0.2222222
>
> print(paste("Accuracy = ", 100 * (1 - missing_classerr)))
[1] "Accuracy =  77.7777777777778"
>
> library(ggplot2)
> ggplot(mtcars, aes(x = wt + disp, y = vs))+
+   geom_point(alpha = 0.5)+
+   stat_smooth(method = "glm", se = FALSE, method.args = list(family = binomial), col = 'r
ed')
`geom_smooth()` using formula = 'y ~ x'
>
> library(ROCR)
> ROCPred = prediction(predict_reg, test$vs)
> ROCPer = performance(ROCPred, measure = 'tpr', x.measure = 'fpr')
> auc <- performance(ROCPred, measure = 'auc')
> auc <- auc@y.values[[1]]
> auc
[1] 0.875
```

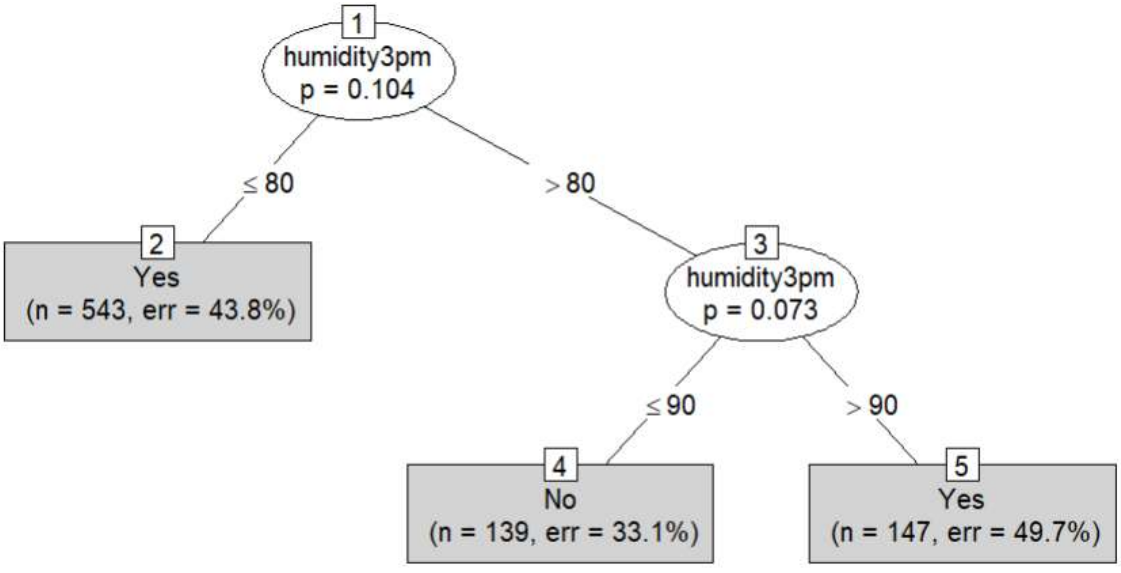



ROC Curve
True positive rate
False positive rate
AUC
0.8333
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.2
0.4
0.6
0.8
1.0
0
0.4
0.8
1.2
1.6
2

9. Construct Decision Tree for Weather data set

```
set.seed(42)
weatherdata <- data.frame(
  cloud3pm = sample(1:10, 1000, replace = TRUE),
  pressure3pm = sample(1000:1025, 1000, replace = TRUE),
  humidity3pm = sample(40:100, 1000, replace = TRUE),
  temp3pm = sample(10:35, 1000, replace = TRUE),
  wind3pm = sample(0:50, 1000, replace = TRUE),
  RainTomorrow = factor(sample(c('No', 'Yes'), 1000, replace = TRUE))
)

table(weatherdata$RainTomorrow)
repeat {
  sample <- sample(c(TRUE, FALSE), nrow(weatherdata), replace = TRUE, prob = c(0.8, 0.2))
  train <- weatherdata[sample, ]
  test <- weatherdata[!sample, ]
  if (length(unique(test$RainTomorrow)) == 2) break
}

library(partykit)
model <- ctree(RainTomorrow ~ ., data = train, control = ctree_control(minsplit = 5, minbucket =
5, maxdepth = 10, mincriterion = 0.7))
plot(model, type = 'simple')

predict_model <- predict(model, test)
mat <- table(test$RainTomorrow, predict_model)
accuracy <- sum(diag(mat)) / sum(mat)
sprintf("Accuracy: %f ", accuracy * 100)
```

Output:



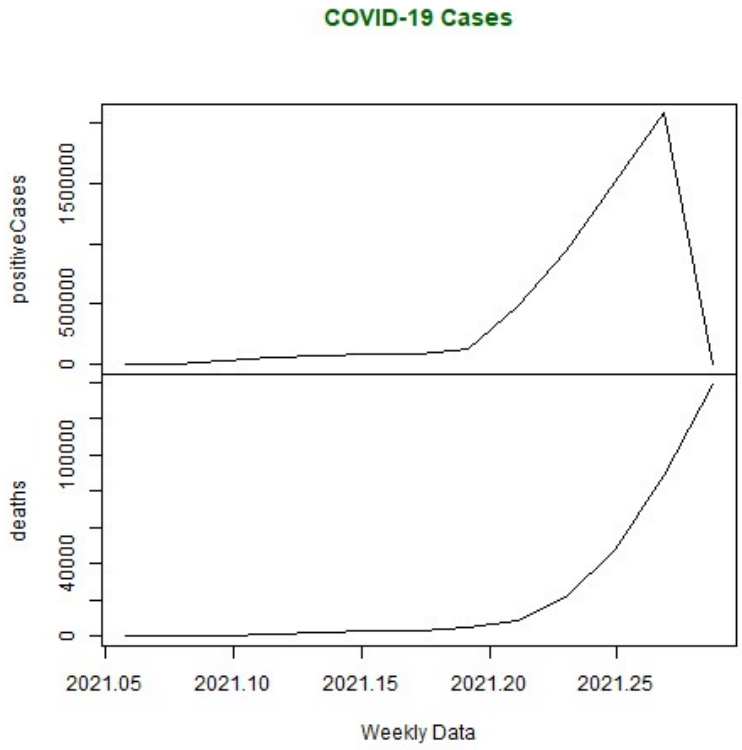
```
R 4.4.1 · ~/
> set.seed(42)
> weatherdata <- data.frame(
+   cloud3pm = sample(1:10, 1000, replace = TRUE),
+   pressure3pm = sample(1000:1025, 1000, replace = TRUE),
+   humidity3pm = sample(40:100, 1000, replace = TRUE),
+   temp3pm = sample(10:35, 1000, replace = TRUE),
+   wind3pm = sample(0:50, 1000, replace = TRUE),
+   RainTomorrow = factor(sample(c('No', 'Yes'), 1000, replace = TRUE))
+ )
>
> table(weatherdata$RainTomorrow)

 No Yes
503 497
> repeat {
+   sample <- sample(c(TRUE, FALSE), nrow(weatherdata), replace = TRUE, prob = c(0.8, 0.2))
+   train <- weatherdata[sample, ]
+   test <- weatherdata[!sample, ]
+   if (length(unique(test$RainTomorrow)) == 2) break
+ }
>
> library(partykit)
> model <- ctree(RainTomorrow ~ ., data = train, control = ctree_control(minsplit = 5, minbuc
ket = 5, maxdepth = 10, mincriterion = 0.7))
> plot(model, type = 'simple')
>
> predict_model <- predict(model, test)
> mat <- table(test$RainTomorrow, predict_model)
> accuracy <- sum(diag(mat)) / sum(mat)
> sprintf("Accuracy: %f ", accuracy * 100)
[1] "Accuracy: 45.614035 "
```

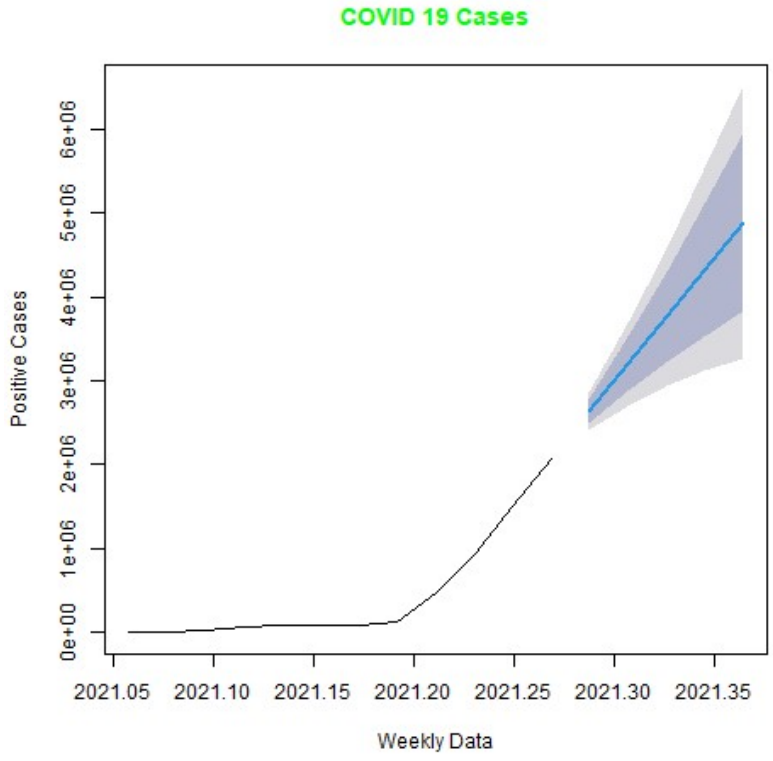
10. Analyse Time-Series Data

```
positiveCases <-
c(580,7813,28266,59287,75700,87820,95314,126214,471497,936851,1508725,2072113)
deaths <- c(17,270,565,1261,2126,2800,3285,4628,8951,21283,47210,88480,138475)

library(lubridate)
png(file='MultiTimeSeries.png')
mts <- ts(cbind(positiveCases, deaths), start = decimal_date(ymd("2021-01-22")), frequency =
365.25/7)
plot(mts, xlab = "Weekly Data", main = "COVID-19 Cases", col.main = "darkgreen")
dev.off()

library(forecast)
png(file="TimeSeries.png")
mts1 <- ts(positiveCases, decimal_date(ymd("2021-01-22")), frequency = 365.25/7)
fit <- auto.arima(mts1)
fit <- forecast(fit, 5)
plot(forecast(fit, 5), xlab = "Weekly Data", ylab = "Positive Cases", main = "COVID 19 Cases",
col.main = "green")
dev.off()
```

Output:

```
R 4.4.1 · ~/
> positiveCases <- c(580,7813,28266,59287,75700,87820,95314,126214,471497,936851,1508725,2072
113)
> deaths <- c(17,270,565,1261,2126,2800,3285,4628,8951,21283,47210,88480,138475)
> library(lubridate)
> png(file='MultiTimeSeries.png')
> mts <- ts(cbind(positiveCases, deaths), start = decimal_date(ymd("2021-01-22")), frequency
= 365.25/7)
Warning message:
In cbind(positiveCases, deaths) :
  number of rows of result is not a multiple of vector length (arg 1)
> plot(mts, xlab = "Weekly Data", main = "COVID-19 Cases", col.main = "darkgreen")
> dev.off()
null device
          1
> library(forecast)
> png(file="TimeSeries.png")
> mts1 <- ts(positiveCases, decimal_date(ymd("2021-01-22")), frequency = 365.25/7)
> fit <- auto.arima(mts1)
> fit <- forecast(fit, 5)
> plot(forecast(fit, 5), xlab = "Weekly Data", ylab = "Positive Cases", main = "COVID 19 Case
s", col.main = "green")
> dev.off()
null device
          1
```


COVID-19 Cases
positiveCases
0
500000
1500000
deaths
0
40000
100000
2021.05
2021.10
2021.15
2021.20
2021.25
Weekly Data


COVID 19 Cases
Positive Cases
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
2021.05
2021.10
2021.15
2021.20
2021.25
2021.30
2021.35
Weekly Data

11. Work on any Data Visualization tools

View(airquality)

barplot(airquality$Ozone, main = 'Ozone Concentration in Air', xlab = 'Ozone Levels', horiz = TRUE)

hist(airquality$Temp, main = "La Guardia Airport's Maximum Temperature(Daily)", xlab = "Temperature(Fahrenheit)", xlim = c(50,125), col = "yellow", freq = TRUE)

boxplot(airquality[,0:4], main = "Box Plots for Air Quality Parameters")

plot(airquality$Ozone, airquality$Month, main = "Scatterplot Example", xlab = "Ozone Concentration in ppb", ylab = "Month of Observation", pch = 19)

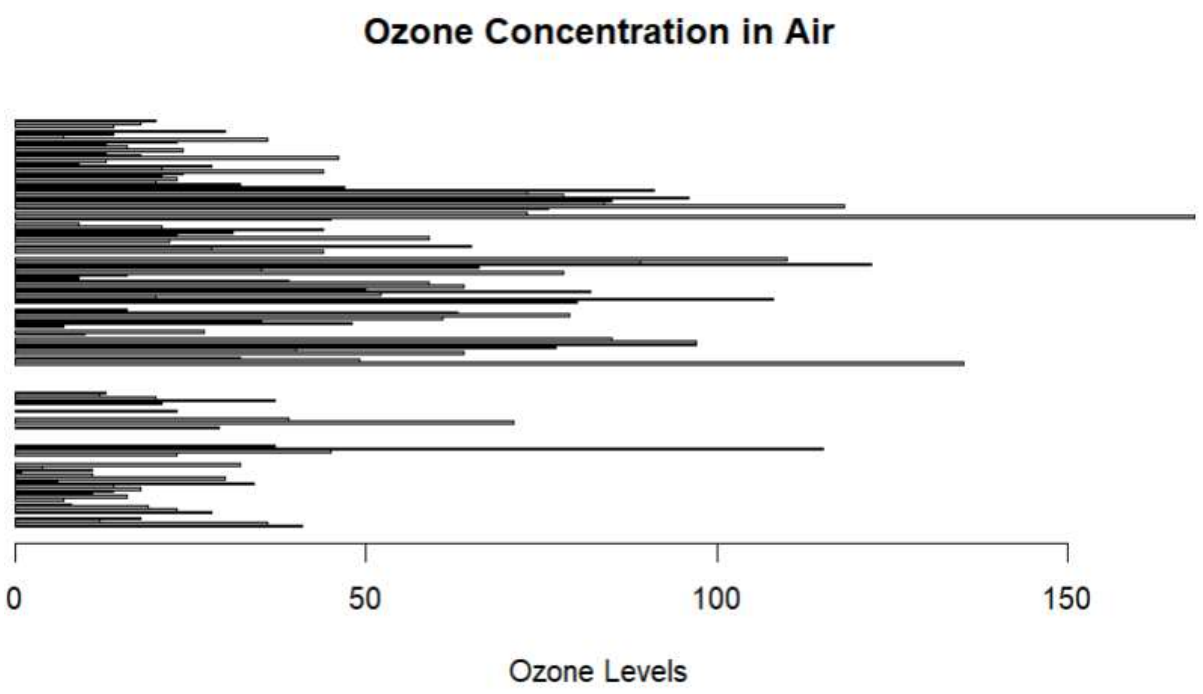
data <- matrix(rnorm(50, 0, 5), nrow = 5, ncol = 5)

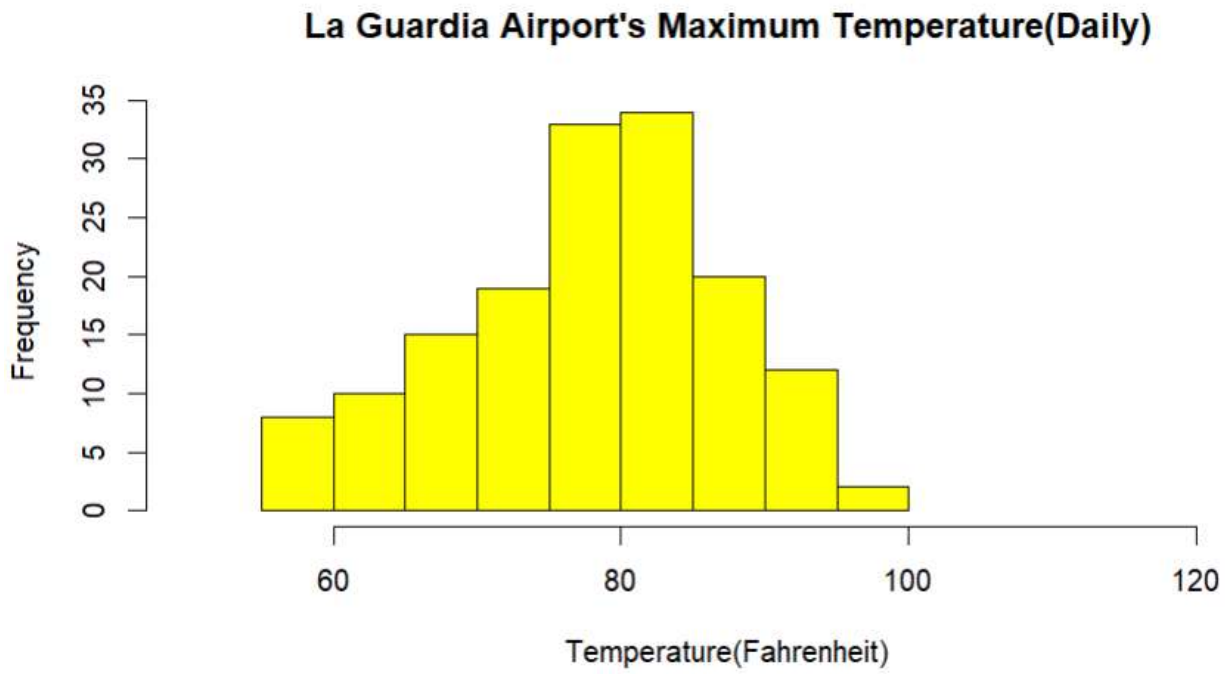
colnames(data) <- paste0("col", 1:5)

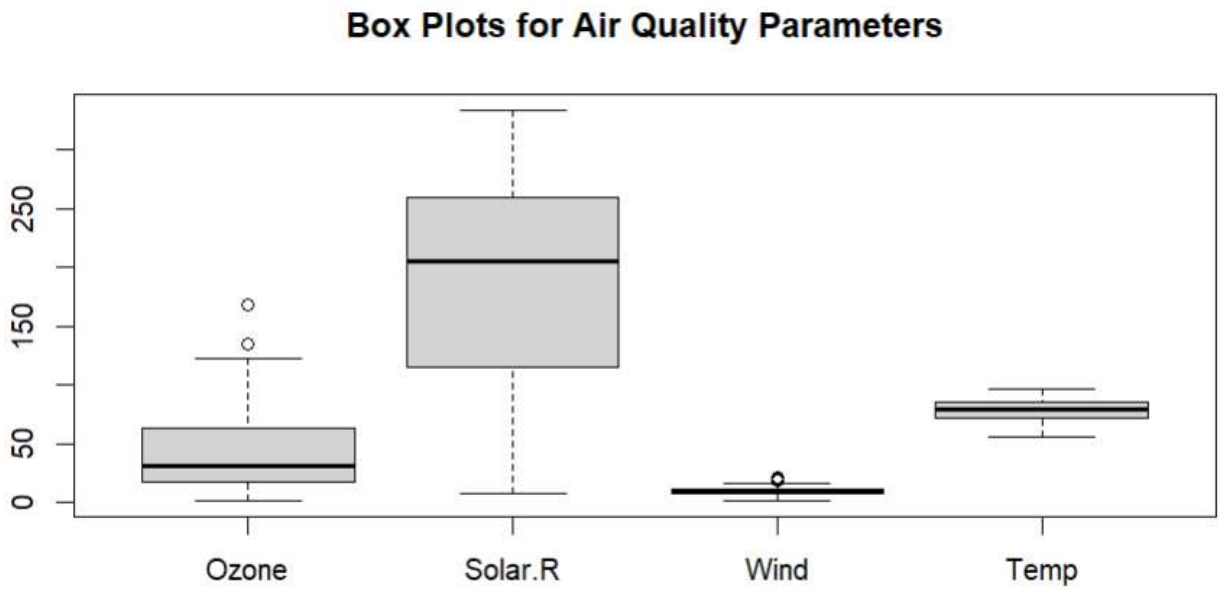
rownames(data) <- paste0("row", 1:5)
heatmap(data)

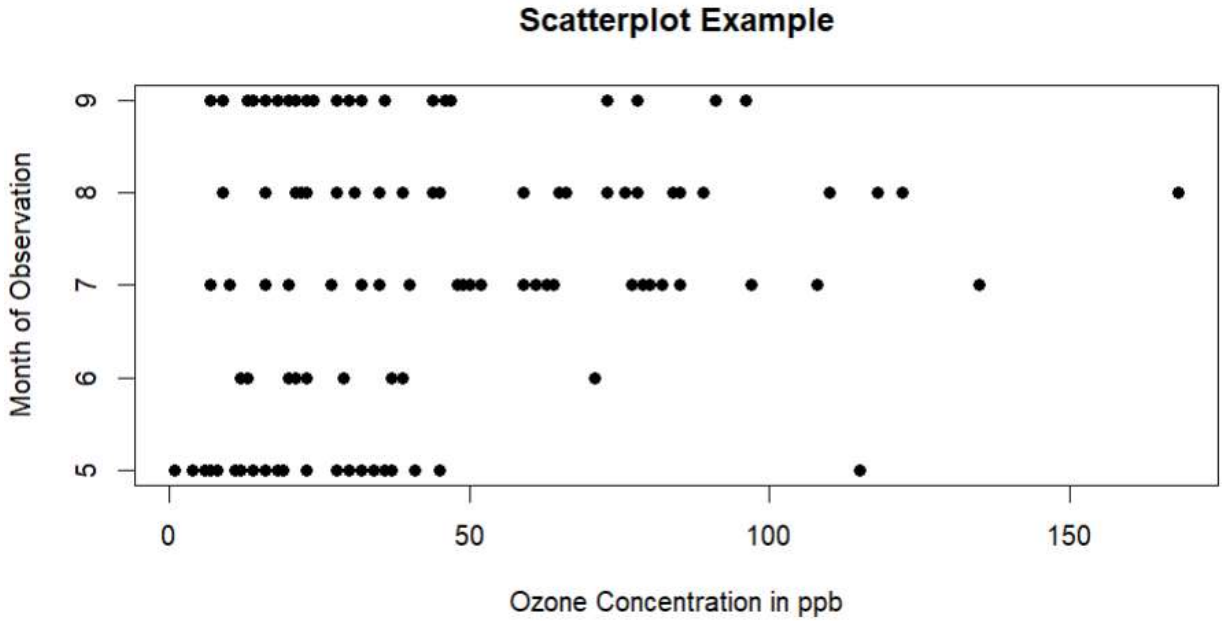
Output:

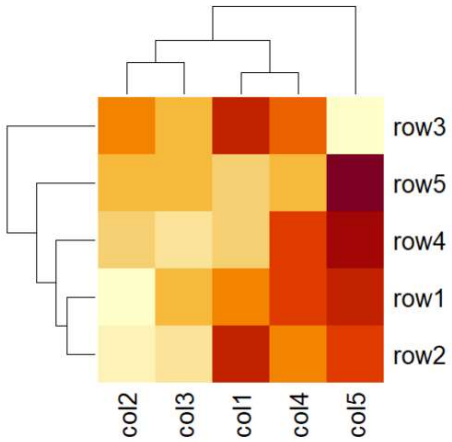
	Ozone	Solar.R	Wind	Temp	Month	Day
1	41	190	7.4	67	5	1
2	36	118	8.0	72	5	2
3	12	149	12.6	74	5	3
4	18	313	11.5	62	5	4
5	NA	NA	14.3	56	5	5
6	28	NA	14.9	66	5	6
7	23	299	8.6	65	5	7
8	19	99	13.8	59	5	8
9	8	19	20.1	61	5	9
10	NA	194	8.6	69	5	10


Ozone Concentration in Air
0
50
100
150
Ozone Levels


La Guardia Airport's Maximum Temperature(Daily)
Frequency
0
5
10
15
20
25
30
35
60
80
100
120
Temperature(Fahrenheit)


Box Plots for Air Quality Parameters
0
50
150
250
Ozone
Solar.R
Wind
Temp


Scatterplot Example
Month of Observation
5
6
7
8
9
0
50
100
150
Ozone Concentration in ppb


row3
row5
row4
row1
row2
col2
col3
col1
col4
col5