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

KUKATPALLY, HYDERABAD – 500085

CERTIFICATE

This is to certify that **PIDUGU PAVANI** of B.Tech III year II Semester bearing the Hall-Ticket number **22015A0502** has fulfilled his/her DATA ANALYTICS LAB record for the academic year 2024-2025.

Signature of the Head of the Department | Signature of the staff member

Date of Examination:

Internal Examiner | External Examiner

TABLE OF CONTENTS

1. Demonstrate data cleaning - missing values

library(tidyverse)

View(airquality)

Untitled1* | newaq | airquality

Filter

	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
11	7	NA	6.9	74	5	11
12	16	256	9.7	69	5	12
13	11	290	9.2	66	5	13
14	14	274	10.9	68	5	14
15	18	65	13.2	58	5	15
16	14	334	11.5	64	5	16
17	34	307	12.0	66	5	17

Showing 1 to 17 of 153 entries, 6 total columns

which(is.na(airquality))

sum(is.na(airquality))

```
> which(is.na(airquality))
 [1]   5  10  25  26  27  32  33  34  35  36  37  39  42  43  45  46  52  53  54  55  56  57
[23]  58  59  60  61  65  72  75  83  84 102 103 107 115 119 150 158 159 164 180 249 250 251
> sum(is.na(airquality))
[1] 44
```

Remove rows with NA values (without modifying the original

data) cleaned_aq <- na.exclude(airquality)

Calculate the mean of 'Ozone' excluding NA values

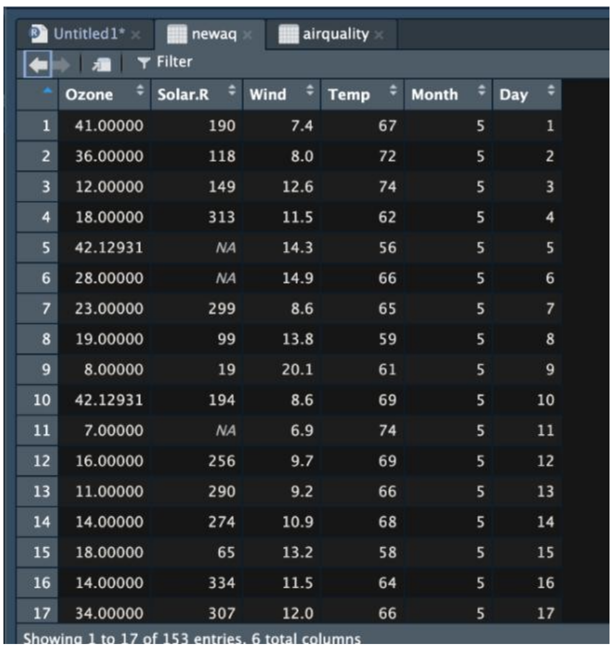
ozone_mean <- mean(airquality$Ozone, na.rm = TRUE)

Fill NA values in 'Ozone' with the calculated mean

newaq <- airquality %>% mutate(Ozone = ifelse(is.na(Ozone), ozone_mean, Ozone))

View the modified dataset

View(newaq)



	Ozone	Solar.R	Wind	Temp	Month	Day
1	41.00000	190	7.4	67	5	1
2	36.00000	118	8.0	72	5	2
3	12.00000	149	12.6	74	5	3
4	18.00000	313	11.5	62	5	4
5	42.12931	NA	14.3	56	5	5
6	28.00000	NA	14.9	66	5	6
7	23.00000	299	8.6	65	5	7
8	19.00000	99	13.8	59	5	8
9	8.00000	19	20.1	61	5	9
10	42.12931	194	8.6	69	5	10
11	7.00000	NA	6.9	74	5	11
12	16.00000	256	9.7	69	5	12
13	11.00000	290	9.2	66	5	13
14	14.00000	274	10.9	68	5	14
15	18.00000	65	13.2	58	5	15
16	14.00000	334	11.5	64	5	16
17	34.00000	307	12.0	66	5	17

Showing 1 to 17 of 153 entries, 6 total columns

2. Implement data normalization (min-max, 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)
```

```
> print(arr2)
 [1] 0 0 0 1 1 0 0 0 0 1 0 1
```

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

```
> print(arr2)
 [1] -0.18 -0.68 -0.27  0.69  1.42 -0.09 -0.80 -1.13 -1.47  1.79 -0.10  0.81
```

3. Implement attribute subset selection for data reduction

library(dplyr)
library(leaps)

View(Titanic)
Titanic = Titanic %>% na.omit()

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

```
> 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 ) "*"      "*"      "*"       "*"       "*"      "*"
```

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

```
> 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 ) "*"      "*"      "*"       "*"       "*"      "*"
```

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

```
> 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 ) "*"      "*"      "*"       "*"       "*"      "*"
```

4. Demonstrate outlier detection

```
#download dataset:
#https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset

file_path<-"/Users/pavani/Downloads/bike+sharing+dataset/day.cs
v"
day<-read.csv(file_path)
View(day)

sum(is.na(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
}

sum(is.na(data))
day<-na.exclude(day)
boxplot(day[,c("temp","hum","windspeed")])
```

OUTPUT:

Untitled1* day

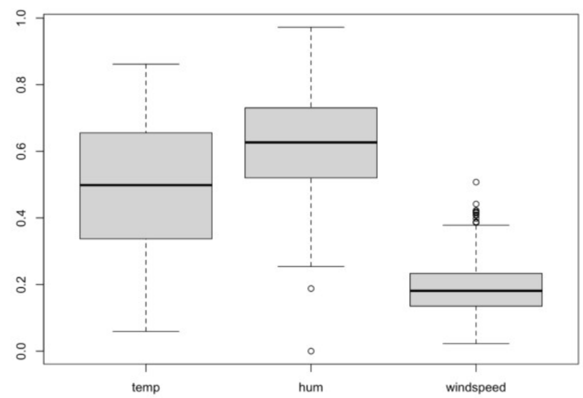
Filter

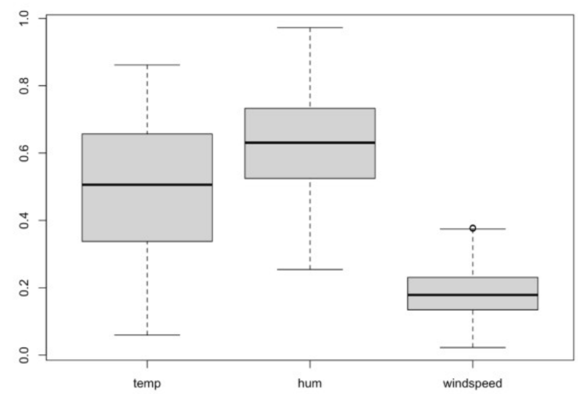
	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
1	1	2011-01-01	1	0	1	0	6	0	2	0.3441670	0.3636250	0.805833	0.1604460	331	654	985
2	2	2011-01-02	1	0	1	0	0	0	2	0.3634780	0.3537390	0.696087	0.2485390	131	670	801
3	3	2011-01-03	1	0	1	0	1	1	1	0.1963640	0.1894050	0.437273	0.2483090	120	1229	1349
4	4	2011-01-04	1	0	1	0	2	1	1	0.2000000	0.2121220	0.590435	0.1602960	108	1454	1562
5	5	2011-01-05	1	0	1	0	3	1	1	0.2269570	0.2292700	0.436957	0.1869000	82	1518	1600
6	6	2011-01-06	1	0	1	0	4	1	1	0.2043480	0.2332090	0.518261	0.0895652	88	1518	1606
7	7	2011-01-07	1	0	1	0	5	1	2	0.1965220	0.2088390	0.498696	0.1687260	148	1362	1510
8	8	2011-01-08	1	0	1	0	6	0	2	0.1650000	0.1622540	0.535833	0.2668040	68	891	959
9	9	2011-01-09	1	0	1	0	0	0	1	0.1383330	0.1161750	0.434167	0.3619500	54	768	822
10	10	2011-01-10	1	0	1	0	1	1	1	0.1508330	0.1508880	0.482917	0.2232670	41	1280	1321
11	11	2011-01-11	1	0	1	0	2	1	2	0.1690910	0.1914640	0.686364	0.1221320	43	1220	1263
12	12	2011-01-12	1	0	1	0	3	1	1	0.1727270	0.1604730	0.599545	0.3046270	25	1137	1162
13	13	2011-01-13	1	0	1	0	4	1	1	0.1650000	0.1508830	0.470417	0.3010000	38	1368	1406
14	14	2011-01-14	1	0	1	0	5	1	1	0.1608700	0.1884130	0.537826	0.1265480	54	1367	1421
15	15	2011-01-15	1	0	1	0	6	0	2	0.2333330	0.2481120	0.498750	0.1579630	222	1026	1248

Showing 1 to 15 of 717 entries, 16 total columns

```
> sum(is.na(day))
[1] 0
```

```
> sum(is.na(data))
[1] 13
```



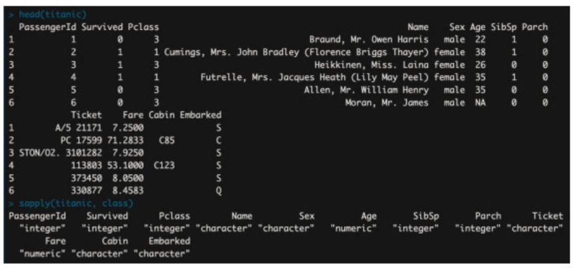


5. Perform analytics on any standard data set

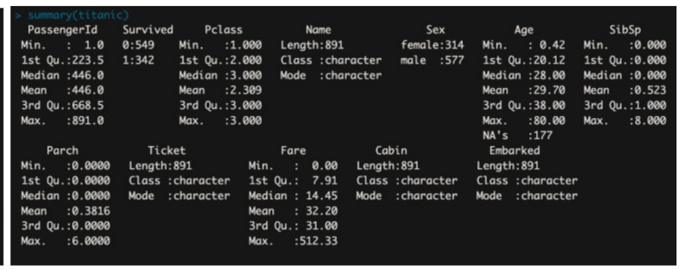
```
#download dataset:
#https://github.com/datasciencedojo/datasets/blob/master/titanic.csv

titanic <- read.csv("/Users/pavani/Downloads/titanic.csv")
library(tidyverse)

head(titanic)
sapply(titanic, class)

#Convert Sex & Survived into factor
titanic$Sex = as.factor(titanic$Sex)
titanic$Survived = as.factor(titanic$Survived)

summary(titanic)

#Filter rows with missing values
dropnull_titanic = titanic[rowSums(is.na(titanic)) <= 0, ]

#Splitting based on survival
survivedList = dropnull_titanic[dropnull_titanic$Survived == 1 , ]
notSurvivedList = dropnull_titanic[dropnull_titanic$Survived == 0, ]

#Pie chart of Survived & Not
Survived mytable <-
table(titanic$Survived) lbls <- c("Not
Survived","Survived") pie(
  mytable,
  labels = lbls,
  main = "pie chart"
)

#Histogram of Ages
hist(titanic$Age, xlab = "gender", ylab = "frequency")

#Bar plot of Gender Distribution among Non-Survivors
barplot(table(notSurvivedList$Sex), xlab = "gender", ylab = "frequency")
```

```
#Density plot of fare of Survivors
temp <- density(table(survivedList$Fare))
plot(temp, type = "n", main = "fare charged")
polygon(temp, col = "lightgray", border = "gray")

#Box plot of Fare
boxplot(titanic$Fare, main = "fare")
```

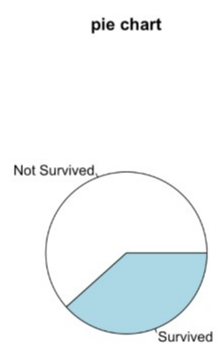
OUTPUT:

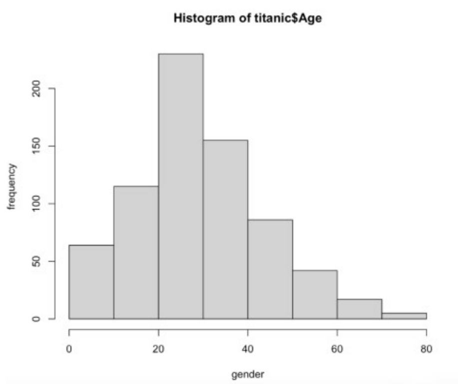


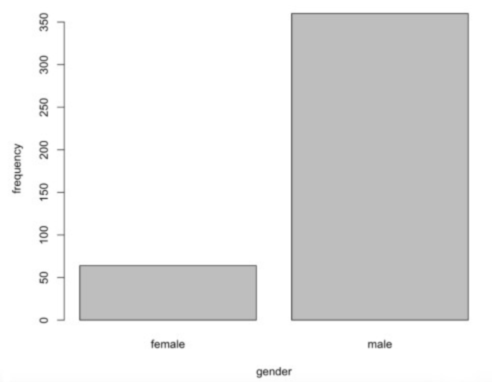
```
> head(titanic)
  PassengerId Survived Pclass                                                Name    Sex Age SibSp Parch
1           1        0      3                             Braund, Mr. Owen Harris   male  22     1     0
2           2        1      1 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
3           3        1      3                              Heikkinen, Miss. Laina female  26     0     0
4           4        1      1        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0
5           5        0      3                            Allen, Mr. William Henry   male  35     0     0
6           6        0      3                                    Moran, Mr. James   male  NA     0     0
            Ticket    Fare Cabin Embarked
1        A/5 21171  7.2500              S
2         PC 17599 71.2833   C85        C
3 STON/O2. 3101282  7.9250              S
4           113803 53.1000  C123        S
5           373450  8.0500              S
6           330877  8.4583              Q
> sapply(titanic, class)
PassengerId    Survived      Pclass        Name         Sex         Age       SibSp       Parch      Ticket
  "integer"   "integer"   "integer" "character" "character"   "numeric"   "integer"   "integer" "character"
       Fare       Cabin    Embarked
  "numeric" "character" "character"
```

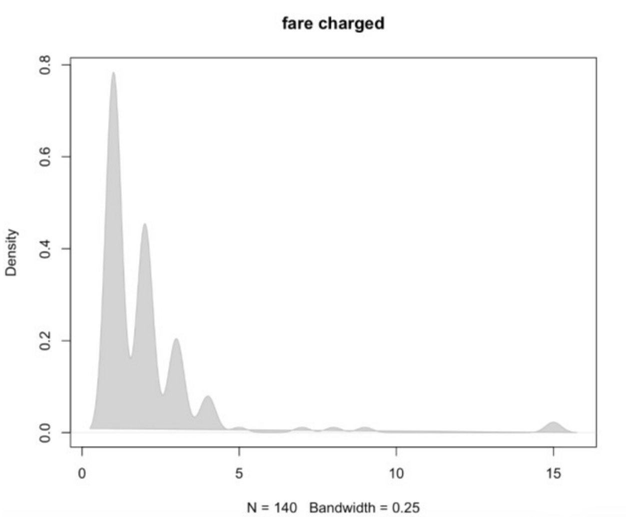


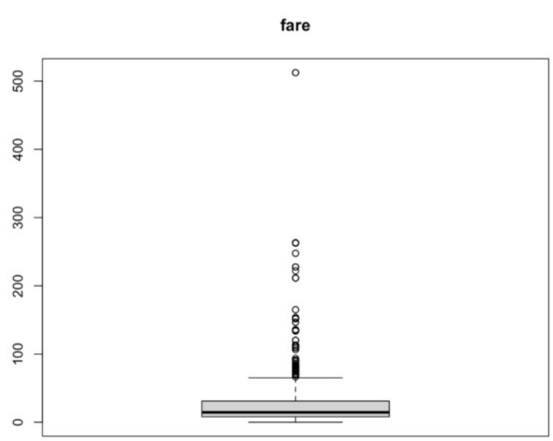
```
> summary(titanic)
  PassengerId    Survived     Pclass          Name               Sex           Age            SibSp
 Min.   :  1.0   0:549    Min.   :1.000   Length:891         female:314   Min.   : 0.42   Min.   :0.000
 1st Qu.:223.5   1:342    1st Qu.:2.000   Class :character   male  :577   1st Qu.:20.12   1st Qu.:0.000
 Median :446.0            Median :3.000   Mode  :character                Median :28.00   Median :0.000
 Mean   :446.0            Mean   :2.309                                   Mean   :29.70   Mean   :0.523
 3rd Qu.:668.5            3rd Qu.:3.000                                   3rd Qu.:38.00   3rd Qu.:1.000
 Max.   :891.0            Max.   :3.000                                   Max.   :80.00   Max.   :8.000
                                                                          NA's   :177
     Parch           Ticket               Fare            Cabin              Embarked
 Min.   :0.0000   Length:891         Min.   :  0.00   Length:891         Length:891
 1st Qu.:0.0000   Class :character   1st Qu.:  7.91   Class :character   Class :character
 Median :0.0000   Mode  :character   Median : 14.45   Mode  :character   Mode  :character
 Mean   :0.3816                      Mean   : 32.20
 3rd Qu.:0.0000                      3rd Qu.: 31.00
 Max.   :6.0000                      Max.   :512.33
```







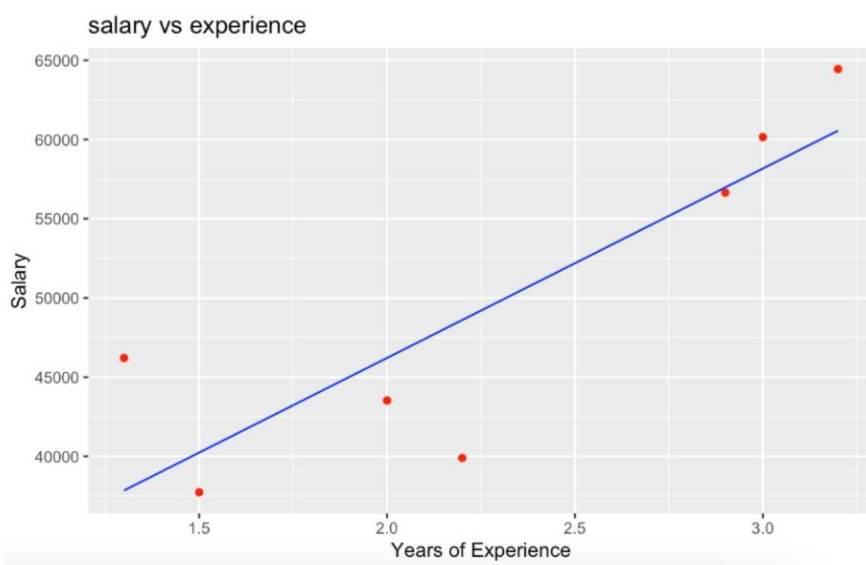




6. Implement linear regression

```
library(caTools)
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))

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)

library(ggplot2)
ggplot() +
  geom_point(aes(x = train$Years_Exp, y = train$Salary), col = 'red') +
  geom_line(aes(x = train$Years_Exp, y = predict(lm.r, data = train)), col =
"blue") +
  ggtitle("salary vs experience") +
  xlab("Years of Experience") +
  ylab("Salary")
```

OUTPUT: 



7. Implement logistic regression

```
# Load necessary libraries
install.packages("pROC")
library(ggplot2)
library(pROC)

# Load the iris dataset
data(iris)

# Convert the Species column to a binary outcome (Setosa vs.
Non-Setosa)
iris$SpeciesBinary <- ifelse(iris$Species == "setosa", 1, 0)

# Logistic regression model: Predict if the flower is Setosa based
on Sepal.Length
logistic_model <- glm(SpeciesBinary ~ Sepal.Length, data = iris, family =
"binomial")

# View the summary of the logistic regression model
summary(logistic_model)
```

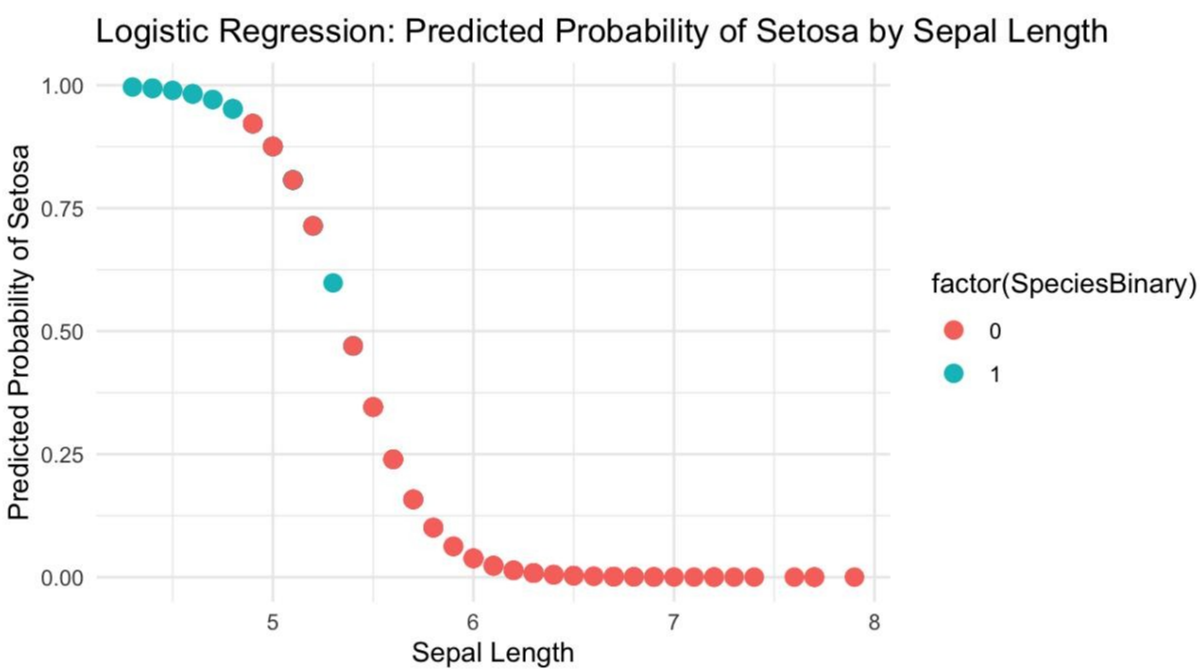
```
> summary(logistic_model)

Call:
glm(formula = SpeciesBinary ~ Sepal.Length, family = "binomial",
    data = iris)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   27.8285     4.8276   5.765 8.19e-09 ***
Sepal.Length  -5.1757     0.8934  -5.793 6.90e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 190.954  on 149  degrees of freedom
Residual deviance:  71.836  on 148  degrees of freedom
AIC: 75.836

Number of Fisher Scoring iterations: 7
```

```
# Predicted probabilities
iris$predicted_probabilities <- predict(logistic_model, type = "response")

# Add a column for predicted class (0 or 1) based on threshold of 0.5
iris$predicted_class <- ifelse(iris$predicted_probabilities > 0.5, 1, 0)
```

# Create a confusion matrix
confusion_matrix <- table(Actual = iris$SpeciesBinary, Predicted = iris$predicted_class)
print(confusion_matrix)

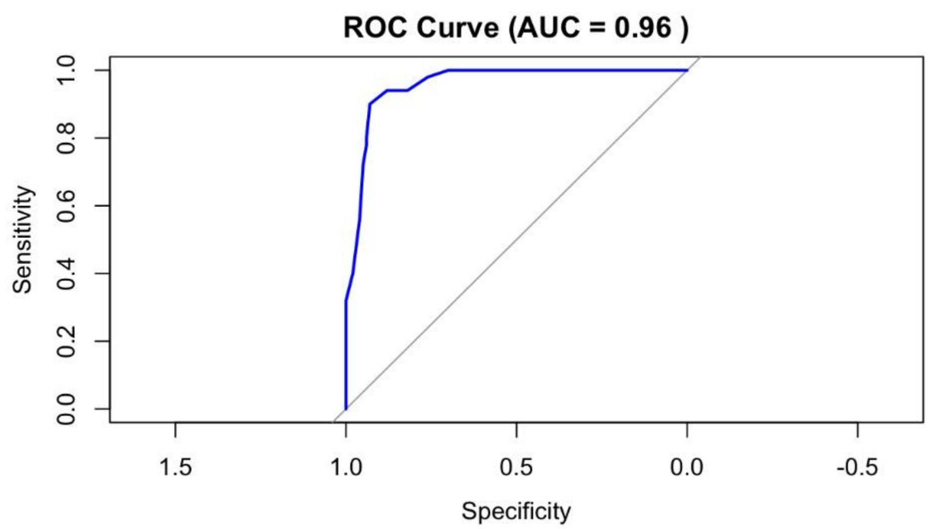
```
> print(confusion_matrix)
      Predicted
Actual  0  1
     0 94  6
     1 10 40
```

Plot the predicted probabilities
ggplot(iris, aes(x = Sepal.Length, y = predicted_probabilities)) +
  geom_point(aes(color = factor(SpeciesBinary)), size = 3) + labs(x = "Sepal Length", y = "Predicted Probability of Setosa") +
  ggtitle("Logistic Regression: Predicted Probability of Setosa by Sepal Length") +
  theme_minimal()



# Create ROC curve and calculate AUC
roc_curve <- roc(iris$SpeciesBinary, iris$predicted_probabilities)

```
# Print the AUC value
auc_value <- auc(roc_curve)
print(paste("AUC:", auc_value))

# Plot the ROC curve
plot(roc_curve, main = paste("ROC Curve (AUC =", round(auc_value,
2), ")"), col = "blue", lwd = 2)
```



8. Construct decision tree for weather data set

install.packages("partykit")

# Load necessary libraries
library(tidyverse)
library(partykit)

download & load dataset:
#https://www.kaggle.com/datasets/petalme/seattle-weather-prediction-data set

weatherdata <-
read.csv("/Users/pavani/Downloads/seattle-weather.csv")

# Inspect the dataset
head(weatherdata)
str(weatherdata)

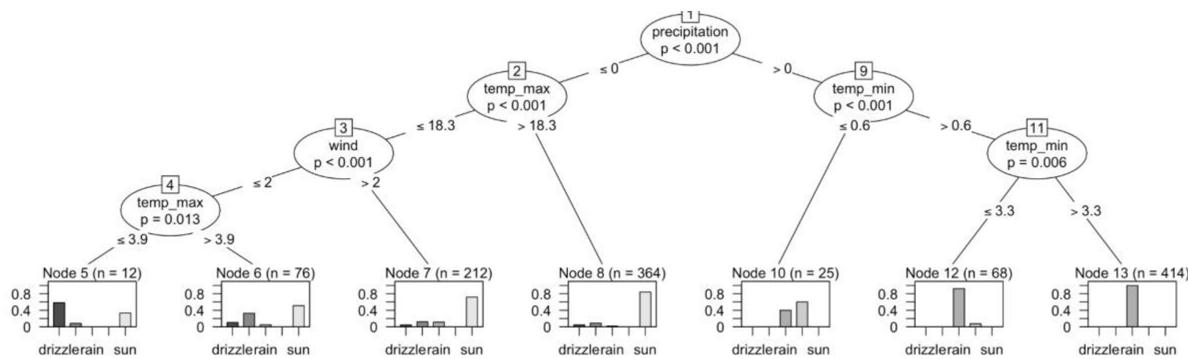
```
> head(weatherdata)
        date precipitation temp_max temp_min wind weather
1 2012-01-01           0.0     12.8      5.0  4.7 drizzle
2 2012-01-02          10.9     10.6      2.8  4.5    rain
3 2012-01-03           0.8     11.7      7.2  2.3    rain
4 2012-01-04          20.3     12.2      5.6  4.7    rain
5 2012-01-05           1.3      8.9      2.8  6.1    rain
6 2012-01-06           2.5      4.4      2.2  2.2    rain
> str(weatherdata)
'data.frame':   1461 obs. of  6 variables:
 $ date         : chr  "2012-01-01" "2012-01-02" "2012-01-03" "2012-01-04" ...
 $ precipitation: num  0 10.9 0.8 20.3 1.3 2.5 0 0 4.3 1 ...
 $ temp_max     : num  12.8 10.6 11.7 12.2 8.9 4.4 7.2 10 9.4 6.1 ...
 $ temp_min     : num  5 2.8 7.2 5.6 2.8 2.2 2.8 2.8 5 0.6 ...
 $ wind         : num  4.7 4.5 2.3 4.7 6.1 2.2 2.3 2 3.4 3.4 ...
 $ weather      : chr  "drizzle" "rain" "rain" "rain" ...
```

Convert 'weather' to a factor for classification
weatherdata$weather <- as.factor(weatherdata$weather)

# Split the dataset into training and testing sets (80-20 split)
sample <- sample(c(TRUE, FALSE), nrow(weatherdata), replace =
TRUE, prob = c(0.8, 0.2))
train <- weatherdata[sample,]
test <- weatherdata[!sample,]

# Train a decision tree model to predict 'weather'
model <- ctree(weather ~ precipitation + temp_max + temp_min + wind, data = train)

# Plot the decision tree
plot(model)



# Make predictions on the test set
predict_model <- predict(model, test)

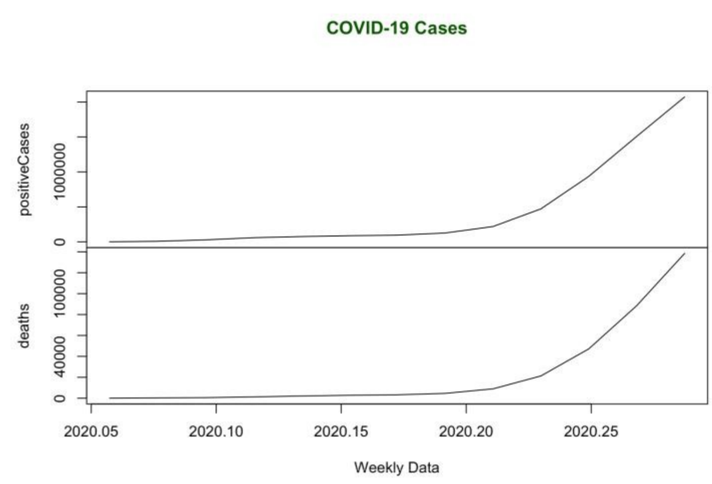
# Generate a confusion matrix to evaluate model performance mat <- table(test$weather, predict_model)
print(mat)

# Calculate the accuracy of the model
accuracy <- sum(diag(mat)) / sum(mat)
print(paste("Accuracy:", accuracy))

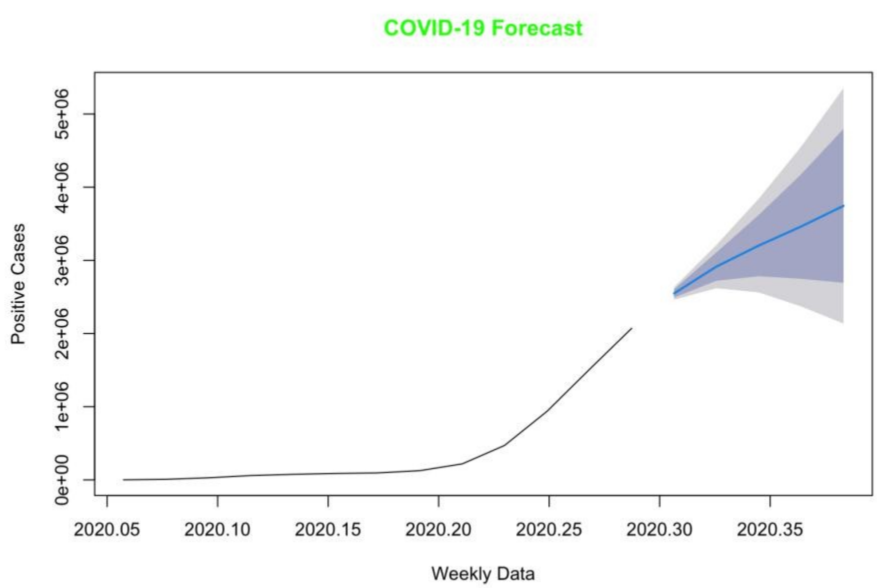
```
> print(mat)
         predict_model
          drizzle fog rain snow sun
  drizzle       0   0    0    0  10
  fog           0   0    0    0  16
  rain          0   0  108    3   9
  snow          0   0    1    4   0
  sun           0   0    0    0 139
> accuracy <- sum(diag(mat)) / sum(mat)
> print(paste("Accuracy:", accuracy))
[1] "Accuracy: 0.86551724137931"
```

9. Analyze time-series data

```
# Load necessary libraries
library(lubridate) #converts the starting date into a decimal date
library(forecast) #use it to fit ARIMA models and make forecasts

# Data for positive cases and deaths (as weekly counts)
positiveCases <- c(580, 7813, 28266, 59287, 75700, 87820,
95314, 126214, 218843, 471497, 936851, 1508725, 2072113)

deaths <- c(17, 270, 565, 1261, 2126, 2800, 3285, 4628, 8951,
21283, 47210, 88480, 138475)

# Create a multivariate time series object
# Starting from January 22, 2020, with weekly frequency
mts <- ts(cbind(positiveCases, deaths),
        start = decimal_date(ymd("2020-01-22")), # Date conversion
to decimal format
        frequency = 365.25 / 7)        # Approximate weekly frequency

# Plot the multivariate time series data (positive cases and deaths)
plot(mts,
    xlab = "Weekly Data", main
    = "COVID-19 Cases",
    col.main = "darkgreen")
```



```
# Create a time series object for positive cases alone
mts1 <- ts(positiveCases,
          start = decimal_date(ymd("2020-01-22")),
          frequency = 365.25 / 7) # Weekly frequency

# Fit an ARIMA model to the positive cases time series
fit <- auto.arima(mts1)

# Generate forecasts for the next 5 periods (weeks)
fit_forecast <- forecast(fit, h = 5)

# Plot the forecast of positive cases for the next 5 weeks
plot(fit_forecast,
     xlab = "Weekly Data", ylab =
     "Positive Cases", main =
     "COVID-19 Forecast",
     col.main = "green")
```



10. Work on any data visualization tool

view(airquality)

Bar plot for Ozone

concentration barplot(

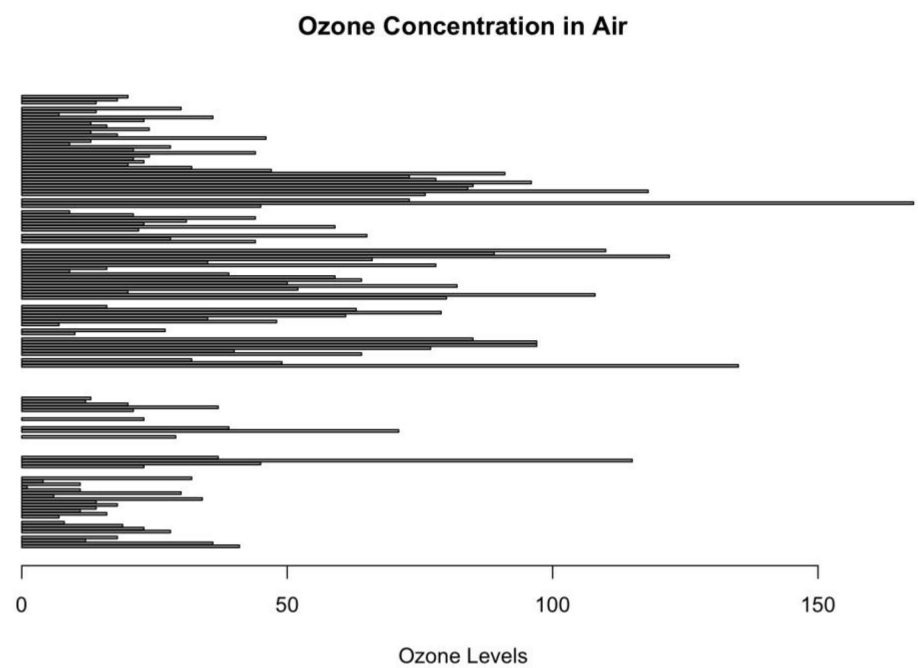
  airquality$Ozone,

  main = "Ozone Concentration in

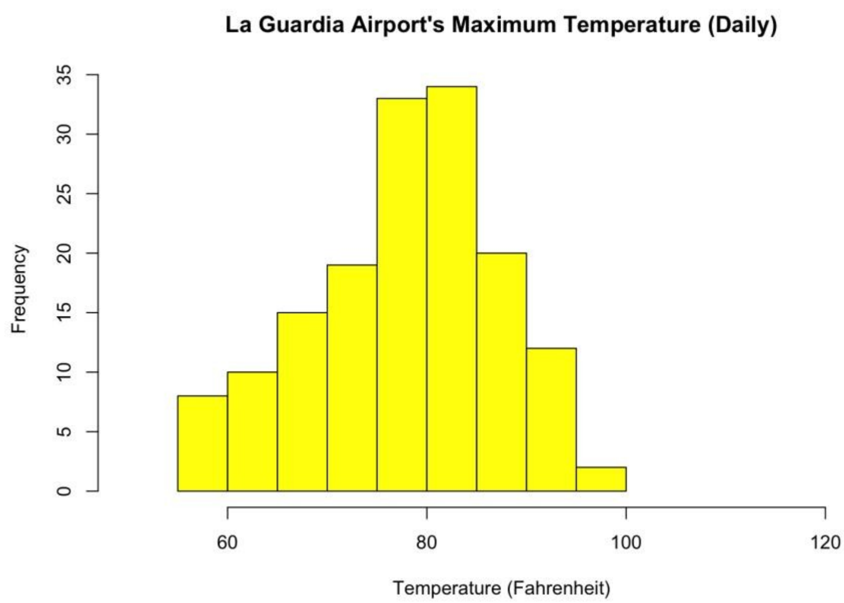
  Air", xlab = "Ozone Levels",

  horiz = TRUE

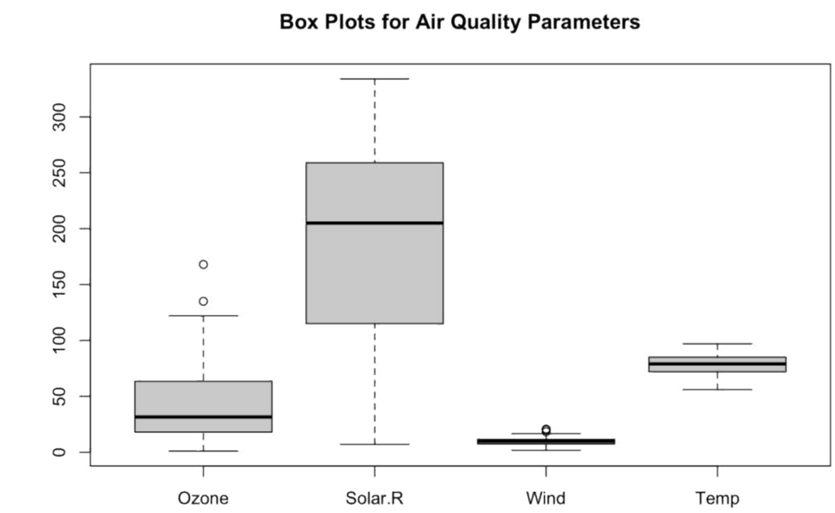
)



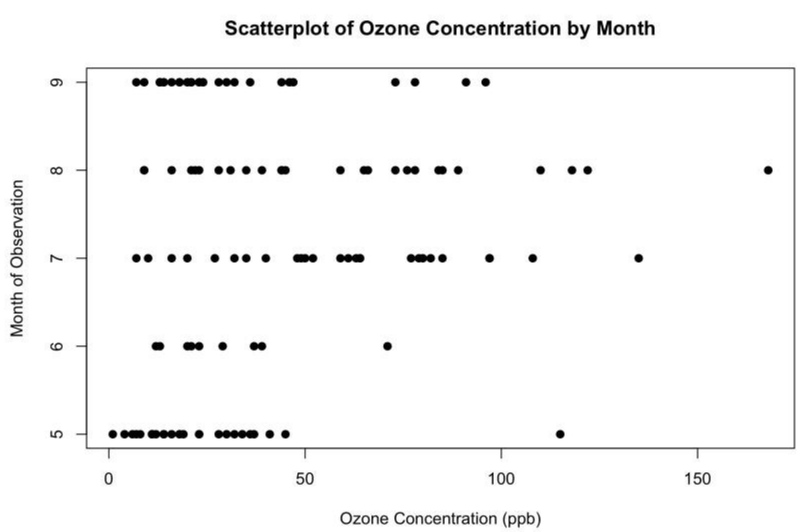
```
# Histogram of Temperature at La Guardia
Airport hist(
  airquality$Temp,
  main = "La Guardia Airport's Maximum Temperature
  (Daily)", xlab = "Temperature (Fahrenheit)",
  xlim = c(50, 125),
  col = "yellow",
  freq = TRUE
)
```




```
# Box plots for selected air quality parameters
boxplot(
  airquality[, 1:4],
  main = "Box Plots for Air Quality Parameters"
)
```




```
# Scatter plot for Ozone concentration by
Month plot(
  airquality$Ozone,
  airquality$Month,
  main = "Scatterplot of Ozone Concentration by
  Month", xlab = "Ozone Concentration (ppb)",
  ylab = "Month of
  Observation", pch = 19
)
```



Creating a sample matrix and drawing a

heatmap data <- matrix(rnorm(25, 0, 5), nrow = 5,

ncol = 5) colnames(data) <- paste("Col", 1:5)

rownames(data) <- paste("Row", 1:5)

heatmap(data)