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

```
> which(is.na(mtcars))
 [1]   1  10  33  37  42  48  66  69  73  76  77  85 101 105 112 115 116 136 149 16
2 170 171
[23] 174 175 193 194 196 203 206 213 239 245 261 290 298 305

> sum(is.na(mtcars))
[1] 36
```

```
> 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
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
Fiat 128          32.4   4  78.7  66 4.08 2.200 19.47  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, 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)
```

<u>Output:</u>

```
> 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)
 [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

```
view(Titanic)

sum(is.na(Titanic))

Titanic = Titanic %>%

 na.omit()

dim(Titanic)

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

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

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

summary(fwd)

summary(bwd)

summary(full)

coef(fwd, 3)

coef(bwd, 3)

coef(full, 3)
```

Output:

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

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

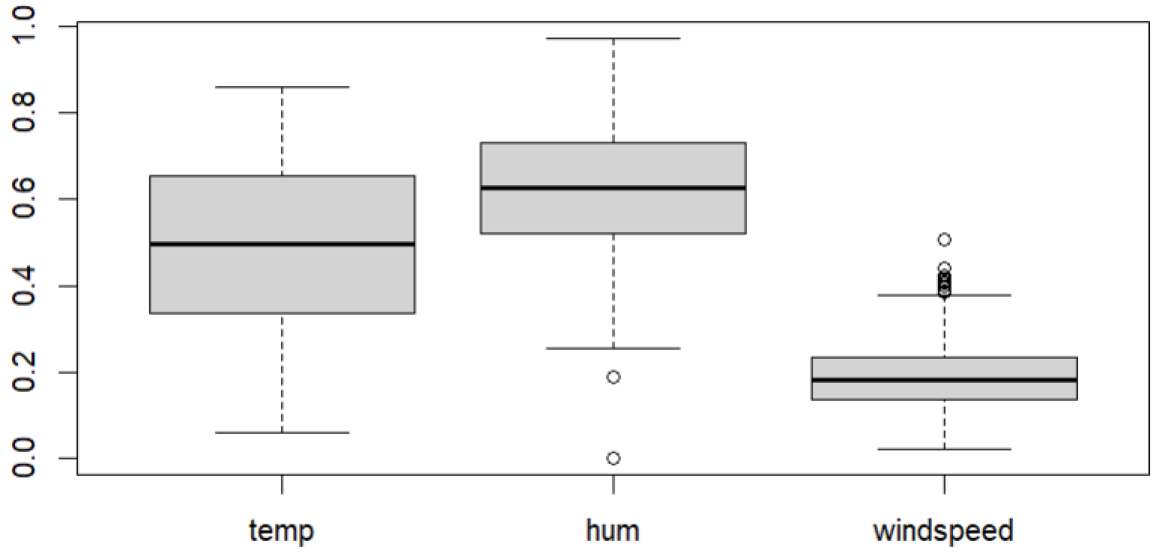
```
> summary(full)
Subset selection object
Call: regsubsets.formula(Freq ~ ., data = Titanic, nvmax = 19)
6 Variables  (and intercept)
          Forced in Forced out
Class2nd       FALSE      FALSE
Class3rd       FALSE      FALSE
ClassCrew      FALSE      FALSE
SexFemale      FALSE      FALSE
AgeAdult       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(fwd, 3)
(Intercept)   SexFemale    AgeAdult SurvivedYes
    70.5625    -78.8125    123.9375    -48.6875
> coef(bwd, 3)
(Intercept)   SexFemale    AgeAdult SurvivedYes
    70.5625    -78.8125    123.9375    -48.6875
> coef(full, 3)
(Intercept)   SexFemale    AgeAdult SurvivedYes
    70.5625    -78.8125    123.9375    -48.6875

>
```

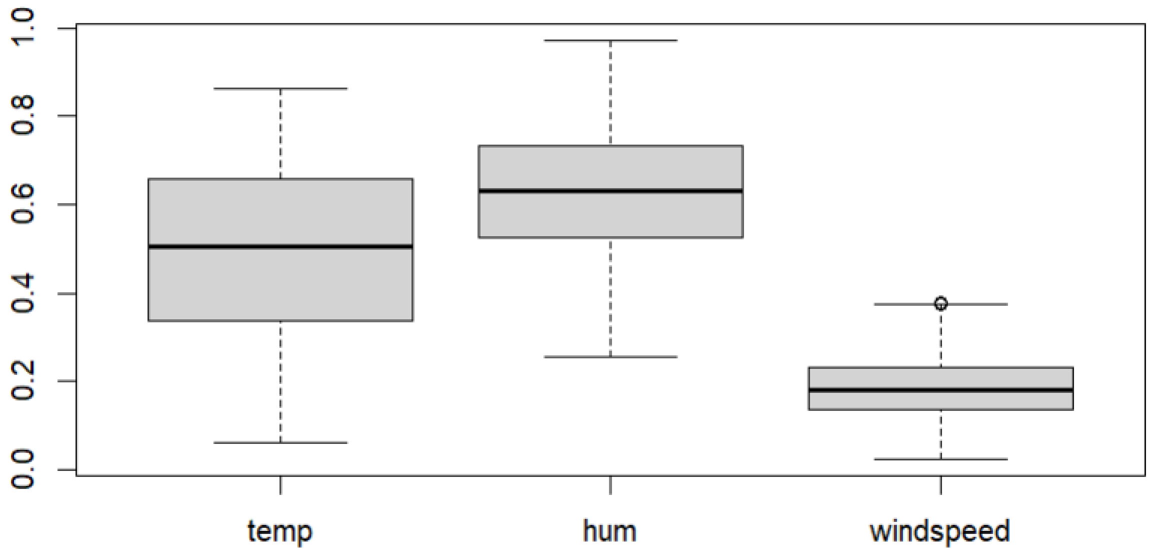
4. Demonstrate outlier detection

```
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 = drop_na(day)
boxplot(day[, c("temp", "hum", "windspeed")])
```

Output:



0.0
0.2
0.4
0.6
0.8
1.0
temp
hum
windspeed



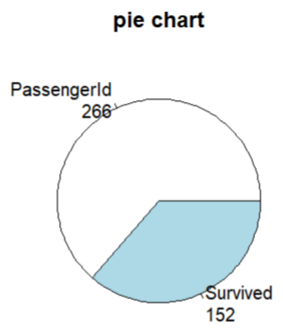
0.0
0.2
0.4
0.6
0.8
1.0
temp
hum
windspeed

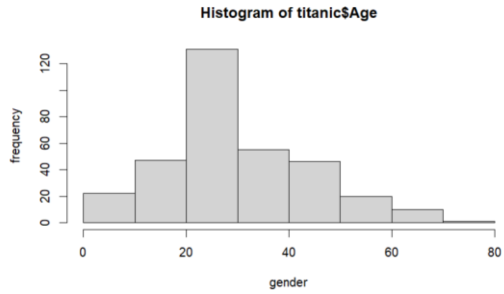
5. Perform analytics on any standard data set

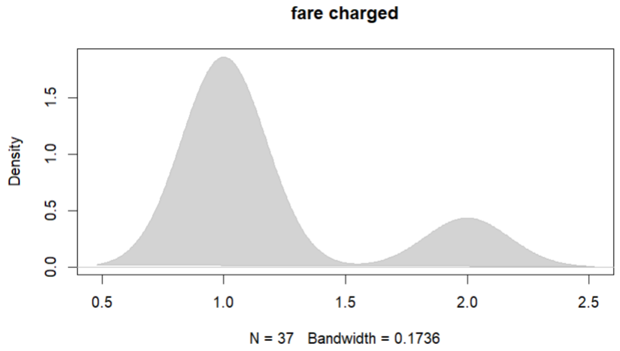
```
library(tidyverse)
head(titanic)
sapply(titanic, class)
titanic$Sex  = as.factor(titanic$Sex)
titanic$Survived = as.factor(titanic$Survived)
summary(titanic)
dropnull_titanic = titanic[rowSums(is.na(titanic)) <= 0, ]
survivedList = dropnull_titanic[dropnull_titanic$Survived == 1 , ]
notSurvivedList = dropnull_titanic[dropnull_titanic$Survived == 0, ]
mytable <- table(titanic$Survived)
lbls <- paste(names(titanic), "\n", mytable, sep = "")
pie(
  mytable,
  labels = lbls,
  main = "pie chart"
)
hist(titanic$Age, xlab = "gender", ylab = "frequency")
barplot(table(notSurvivedList$Sex), xlab = "gender", ylab = "frequency")
temp <- density(table(survivedList$Fare))
plot(temp, type = "n", main = "fare charged")
polygon(temp, col = "lightgray", border = "gray")
boxplot(titanic$Fare, main = "fare")
```

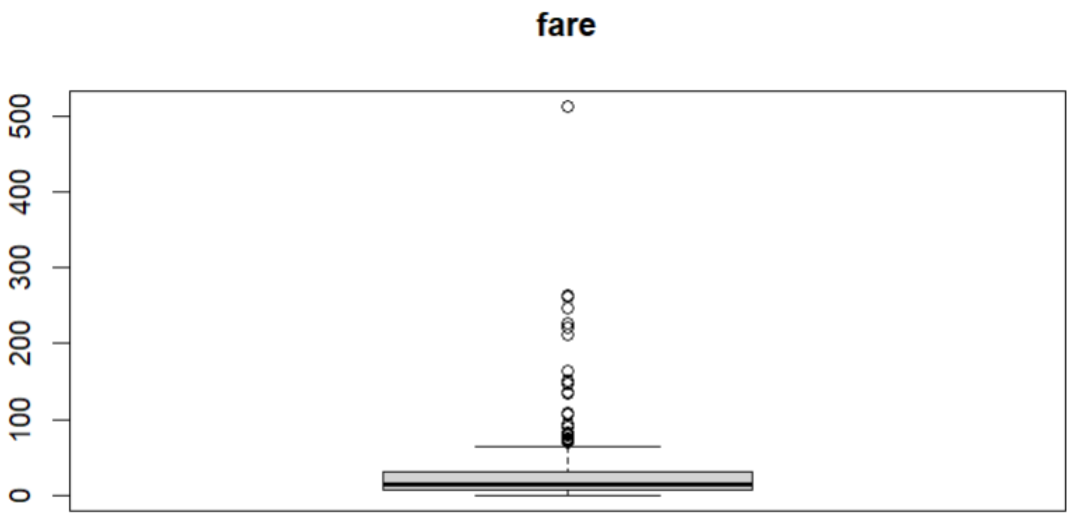
```
# A tibble: 6 × 12
  PassengerId Survived Pclass Name        Sex    Age SibSp Parch Ticket  Fare Ca
bin Embarked
       <dbl> <fct>     <dbl> <chr>       <fct> <dbl> <dbl> <dbl> <chr>  <dbl> <chr> <
chr>
1        892 0            3 Kelly, Mr. J… male   34.5     0     0 330911  7.83 NA    Q
2        893 1            3 Wilkes, Mrs.… fema…  47       1     0 363272  7    NA    S
3        894 0            2 Myles, Mr. T… male   62       0     0 240276  9.69 NA    Q
4        895 0            3 Wirz, Mr. Al… male   27       0     0 315154  8.66 NA    S
5        896 1            3 Hirvonen, Mr… fema…  22       1     1 31012… 12.3  NA    S
6        897 0            3 Svensson, Mr… male   14       0     0 7538    9.22 NA    S
> sapply(titanic, class)
PassengerId    Survived      Pclass        Name         Sex         Age       SibSp
  "numeric"    "factor"   "numeric" "character"    "factor"   "numeric"   "numeric"
      Parch      Ticket        Fare       Cabin    Embarked
  "numeric" "character"   "numeric" "character" "character"
> summary(titanic)
  PassengerId     Survived     Pclass          Name               Sex           Age
 Min.   : 892.0   0:266    Min.   :1.000   Length:418         female:152   Min.   : 0.17
 1st Qu.: 996.2   1:152    1st Qu.:1.000   Class :character   male  :266   1st Qu.:21
.00
 Median :1100.5            Median :3.000   Mode  :character                Median :27.0
0
 Mean   :1100.5            Mean   :2.266                                   Mean   :30.27
 3rd Qu.:1204.8            3rd Qu.:3.000                                   3rd Qu.:39.00
 Max.   :1309.0            Max.   :3.000                                   Max.   :76.00
                                                                           NA's   :86
     SibSp            Parch           Ticket               Fare            Cabin
 Min.   :0.0000   Min.   :0.0000   Length:418         Min.   :  0.000   Length:418
 1st Qu.:0.0000   1st Qu.:0.0000   Class :character   1st Qu.:  7.896   Class :chara
cter
 Median :0.0000   Median :0.0000   Mode  :character   Median : 14.454   Mode
:character
 Mean   :0.4474   Mean   :0.3923                      Mean   : 35.627
 3rd Qu.:1.0000   3rd Qu.:0.0000                      3rd Qu.: 31.500
 Max.   :8.0000   Max.   :9.0000                      Max.   :512.329
                                                      NA's   :1
   Embarked
 Length:418
 Class :character
```

Mode :character







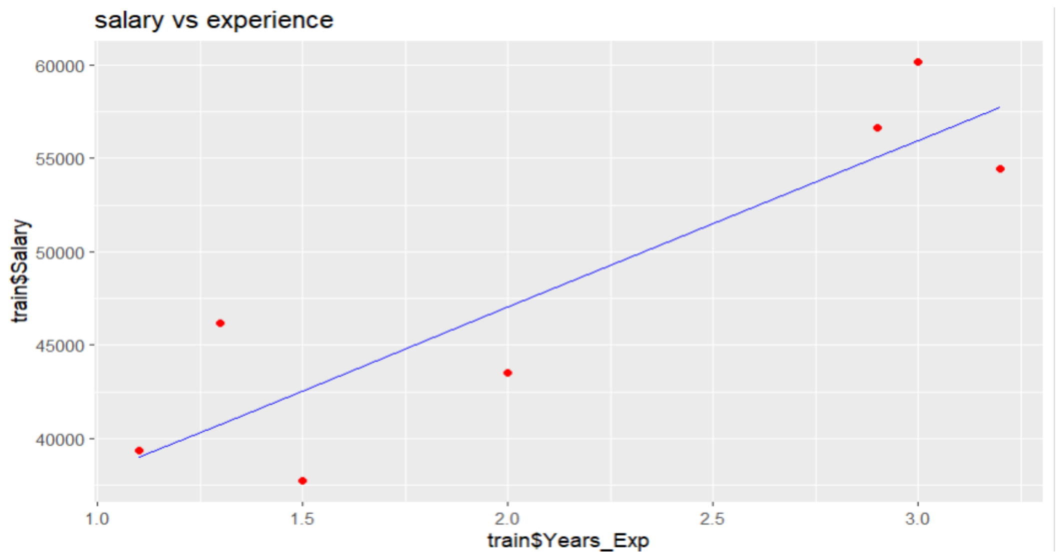


6. Implement linear regression

```
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)
split = sample.split(data$Salary, SplitRatio = 0.7)
train = subset(data, split == TRUE)
test = subset(data, split == FALSE)
lm.r = lm(formula = Salary ~ Years_Exp, data = train)
coef(lm.r)
ggplot() + geom_point(aes(x = train$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")
```

Output:

```
(Intercept)   Years_Exp
  29172.310    8922.322
```



7. Implement logistic regression

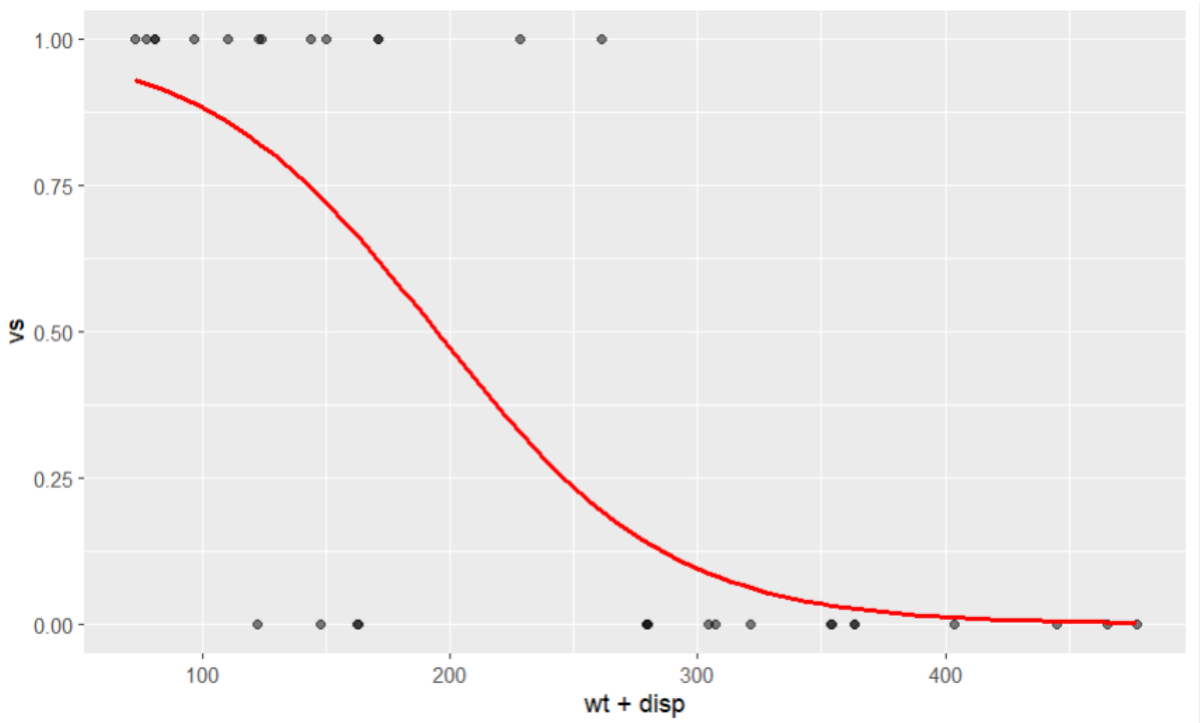
```
library(tidyverse)
library(ROCR)
library(caTools)
view(mtcars)
split <- sample.split(mtcars, SplitRatio = 0.8)
train <- subset(mtcars, split == "TRUE")
test <- subset(mtcars, split == "FALSE")

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 = ", (1 - missing_classerr)))

library(ggplot2)

#plot logistic regression curve
ggplot(mtcars, aes(x=wt + disp, y=vs)) +
```

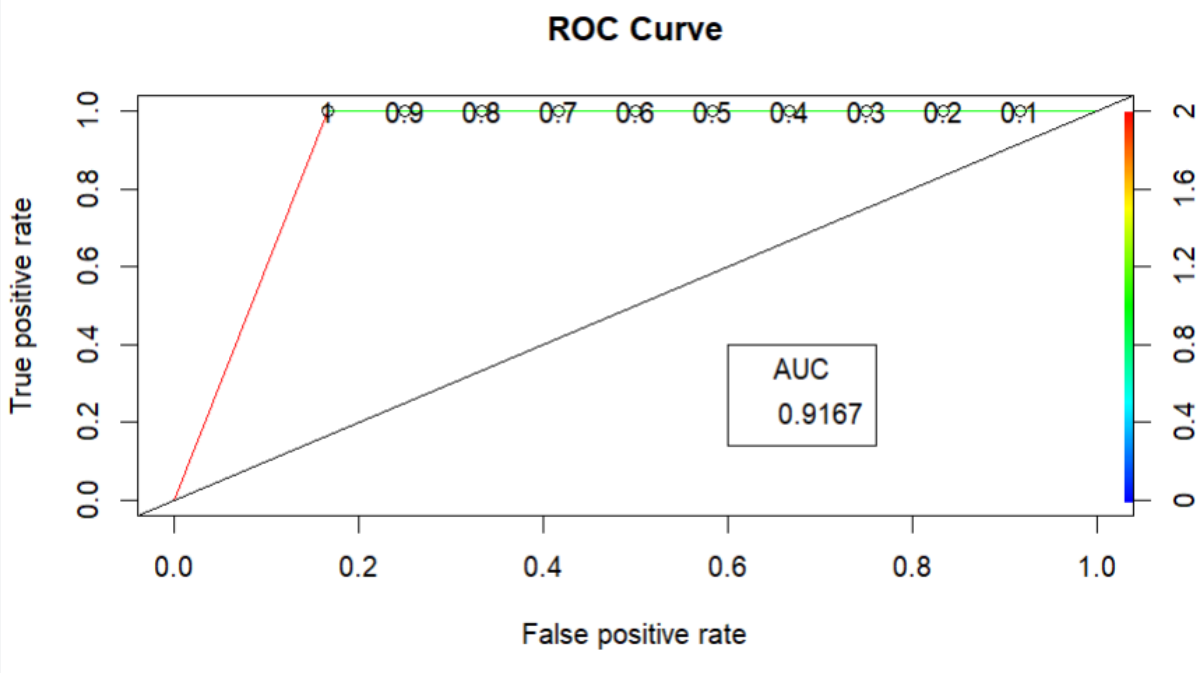
```
  geom_point(alpha=.5) +
  stat_smooth(method="glm", se=FALSE, method.args = list(family=binomial),
              col="red")
library(ROCR)
ROCPred = prediction(predict_reg, test$vs)
ROCPer = performance(ROCPred, measure = "tpr", x.measure = "fpr")
auc <- performance(ROCPred, measure = "auc")
auc <- 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

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

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.79114    2.96489   0.941    0.347
wt           0.85989    1.55388   0.553    0.580
disp        -0.02718    0.01456  -1.866    0.062 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 31.841  on 22  degrees of freedom
Residual deviance: 17.188  on 20  degrees of freedom
```

```
AIC: 23.188

Number of Fisher Scoring iterations: 6

>
> predict_reg <- predict(logistic_model, test, type = "response")
> predict_reg
      Datsun 710   Hornet Sportabout        Merc 230        Merc 450SLC
     0.864210634       0.017371341       0.841966715       0.189302645
Lincoln Continental     Toyota Corolla    Pontiac Firebird      Porsche 914-2
     0.006385438       0.919574847       0.008373046       0.796023875
    Maserati Bora
     0.089476536
>
> predict_reg <- ifelse(predict_reg >0.5, 1, 0)
>
> table(test$vs, predict_reg)
   predict_reg
    0 1
  0 5 1
  1 0 3
>
> missing_classerr <- mean(predict_reg != test$vs)
> missing_classerr
[1] 0.1111111
> print(paste("accuracy = ", (1 - missing_classerr)))
[1] "accuracy =  0.888888888888889"
>
> library(ggplot2)
>
> #plot logistic regression curve
> ggplot(mtcars, aes(x=wt + disp, y=vs)) +
+   geom_point(alpha=.5) +
+   stat_smooth(method="glm", se=FALSE, method.args = list(family=binomial),
+               col="red")
`geom_smooth()` using formula = 'y ~ x'
> auc
[1] 0.9166667
```


1.00
0.75
0.50
0.25
0.00
vs
100
200
300
400
wt + disp

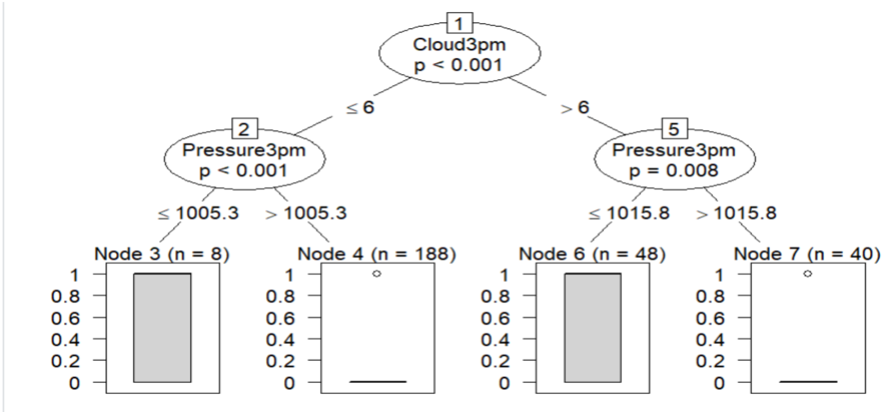

ROC Curve
True positive rate
False positive rate
AUC
0.9167
0.0
0.2
0.4
0.6
0.8
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
2
1.6
1.2
0.8
0.4
0

8. Construct decision tree for weather data set

```
sample = sample(c(TRUE, FALSE), nrow(weatherdata), replace = TRUE, prob = c
(0.8, 0.2))

train <- weatherdata[sample, ]
test <- weatherdata[!sample, ]

library(partykit)
model <- ctree(RainTomorrow ~ ., train)
plot(model)

predict_model <- predict(model, test)
predict_model

mat <- table(test$RainTomorrow, predict_model)
mat

accuracy <- sum(diag(mat)) / sum(mat)
accuracy
```

Output:

```
  predict_model
   0.0478723404255319 0.175 0.625
 0                  52    10     5
 1                   7     4     4
>
> accuracy <- sum(diag(mat)) / sum(mat)
> accuracy
[1] 0.6829268
```

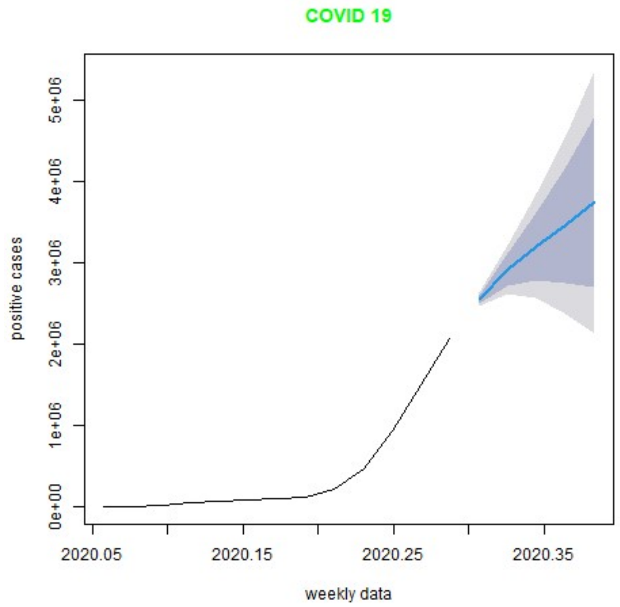


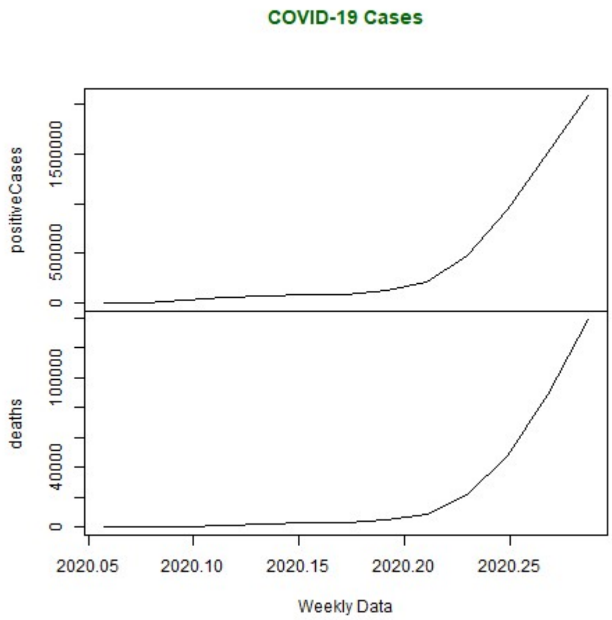
9. Analyse time-series data

```
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)
library(lubridate)
# output to be created as png file
png(file="multivariateTimeSeries.png")
# creating multivariate time series object
# from date 22 January, 2020
mts <- ts(cbind(positiveCases, deaths),
       start = decimal_date(ymd("2020-01-22")),
       frequency = 365.25 / 7)
# plotting the graph
plot(mts, xlab ="Weekly Data",
   main ="COVID-19 Cases",
   col.main ="darkgreen")
library(forecast)
library(lubridate)
png(file = "timeseries.png")
mts1 <- ts(positiveCases, decimal_date(ymd("2020-01-22")), 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", col.main = "green")

dev.off()
```

Output:

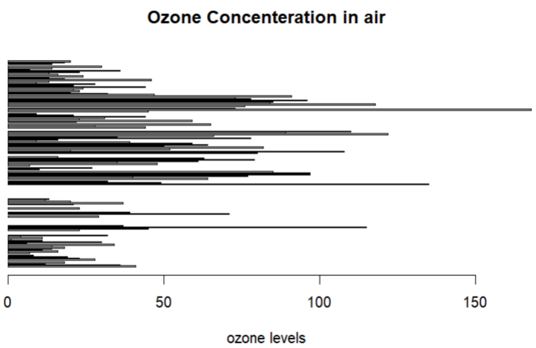


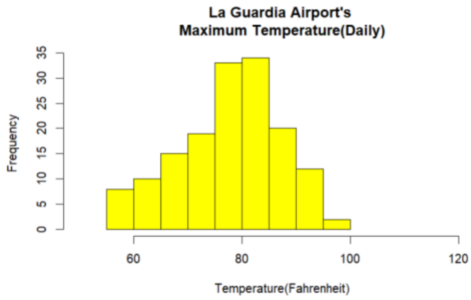


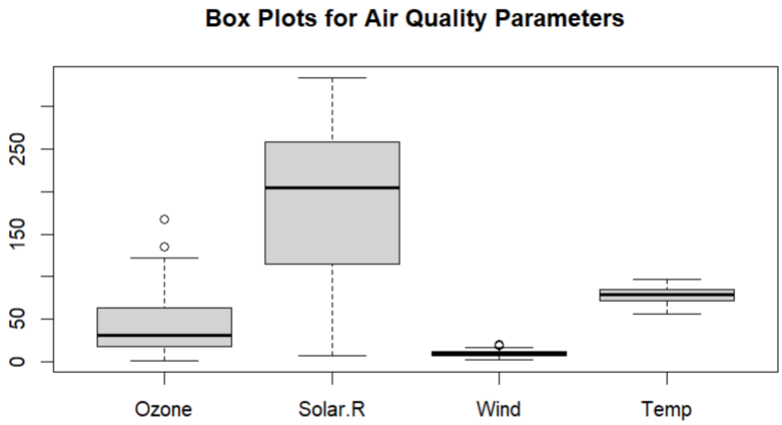
10. Work on any data visualization tool

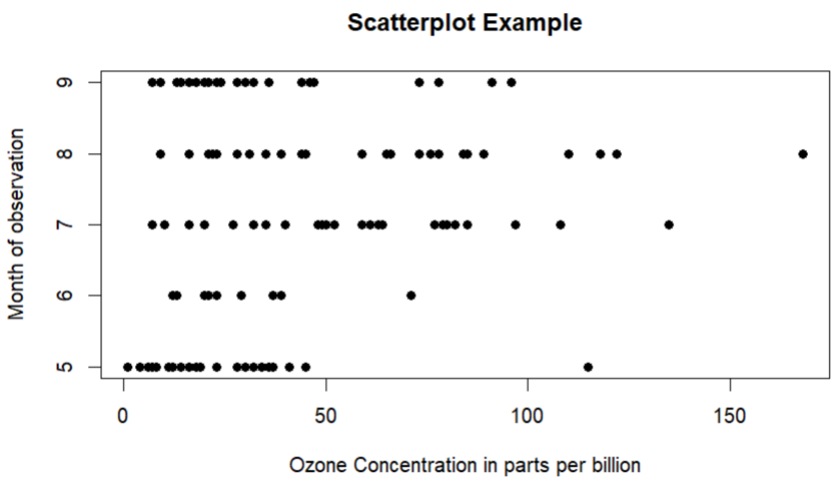
```
view(airquality)
barplot(airquality$Ozone,
        main = 'Ozone Concenteration in air',
        xlab = 'ozone levels', horiz = TRUE)
hist(airquality$Temp, main ="La Guardia Airport's\
Maximum Temperature(Daily)",
     xlab ="Temperature(Fahrenheit)",
     xlim = c(50, 125), col ="yellow",
     freq = TRUE)
boxplot(airquality[, 0:4],
        main ='Box Plots for Air Quality Parameters')
plot(airquality$Ozone, airquality$Month,
     main ="Scatterplot Example",
     xlab ="Ozone Concentration in parts per billion",
     ylab =" Month of observation ", pch = 19)
data <- matrix(rnorm(50, 0, 5), nrow = 5, ncol = 5)

# Column names
colnames(data) <- paste0("col", 1:5)
rownames(data) <- paste0("row", 1:5)

# Draw a heatmap
heatmap(data)
```

Output


Ozone Concenteration 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 parts per billion