```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────────────────────────────────────────────────────
──────────── tidyverse 1.2.1 ──
```

```
## ✔ ggplot2 3.2.1     ✔ purrr   0.3.2
## ✔ tibble  2.1.3     ✔ dplyr   0.8.3
## ✔ tidyr   1.0.0     ✔ stringr 1.4.0
## ✔ readr   1.3.1     ✔ forcats 0.4.0
```

```
## ── Conflicts ────────────────────────────────────────────────────────────────────────────────────────────
────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(readr)
library(ggplot2)
library(imputeTS)
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(outliers)
```

```
##
## Attaching package: 'outliers'
```

```
## The following object is masked from 'package:psych':
##
##     outlier
```

```
library(OutlierDetection)
```

```
## Registered S3 method overwritten by 'spatstat':
##   method     from
##   print.boxx cli
```

```
library(fastDummies)
library(ggvis)
```

```
##
## Attaching package: 'ggvis'
```

```
## The following object is masked from 'package:ggplot2':
##
##     resolution
```

```
library(ggpubr)
```

```
## Loading required package: magrittr
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(recipes)
```

```
##
## Attaching package: 'recipes'
```

```
## The following object is masked from 'package:stringr':
##
##     fixed
```

```
## The following object is masked from 'package:stats':
##
##     step
```

Loading Data Set

```
myData <- read.csv("Weather Forecast Training.csv")
View(myData)
```

Step1: Data Cleaning

Removing Duplicates if any

```
myData1 <- myData %>% distinct(Location,MinTemp,MaxTemp,Rainfall,Evaporation,Sunshine,WindGustDir,WindGustSpeed,Wi
ndDir,WindSpeed,Humidity,Pressure,Cloud,Temp,RainToday,RainTomorrow)
str(myData1)
```

```
## 'data.frame':    51959 obs. of  16 variables:
##  $ Location     : Factor w/ 49 levels "Adelaide","Albany",..: 27 29 15 7 44 20 21 46 45 26 ...
##  $ MinTemp      : num  18.9 11.1 15.9 0 9.1 6.6 9.7 17.1 13.5 12.5 ...
##  $ MaxTemp      : num  23.7 20.8 19.5 14.9 22.7 16.3 20.6 21.9 23.5 21.3 ...
##  $ Rainfall     : num  0 0 17.6 0 0 ...
##  $ Evaporation  : num  NA 4.8 NA NA NA 3.4 1 NA 6.8 NA ...
##  $ Sunshine     : num  NA 8.3 NA NA NA 6.9 4.2 0 6.5 NA ...
##  $ WindGustDir  : Factor w/ 17 levels "","E","ENE","ESE",..: 12 15 11 17 7 10 16 12 9 15 ...
##  $ WindGustSpeed: int  41 39 44 35 41 50 61 54 46 65 ...
##  $ WindDir      : Factor w/ 17 levels "","E","ENE","ESE",..: 12 17 14 16 4 10 9 12 6 15 ...
##  $ WindSpeed    : int  28 26 9 19 7 30 20 28 19 22 ...
##  $ Humidity     : int  55 48 99 55 40 55 35 99 53 82 ...
##  $ Pressure     : num  1026 1014 1028 1023 1027 ...
##  $ Cloud        : int  NA 7 NA 4 NA 7 7 8 6 NA ...
##  $ Temp         : num  22.4 19.5 17.8 14.2 22.5 14.6 19.5 17.9 22.2 13.1 ...
##  $ RainToday    : Factor w/ 3 levels "","No","Yes": 2 2 3 2 2 3 2 3 2 2 ...
##  $ RainTomorrow : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 1 1 2 2 2 ...
```

Removing NA's

Discarding these columns as they consist dirty data i.e missing values >50%

```
myData1$Evaporation <- NULL
myData1$Sunshine <- NULL
myData1$Cloud <- NULL
```

mode function

```
  getmode <- function(m) {
  um <- na.omit(unique(m) )
 tab <- tabulate(match(m, um)); um[tab == max(tab) ]
}
```

```
      MinTemp_location <- myData1 %>%
      group_by(Location)%>%
      summarise(median(MinTemp, na.rm=TRUE))

    placeNA <- which(is.na(myData1$MinTemp))
    for (i in placeNA) {
      myData1$MinTemp[i] <- as.numeric(MinTemp_location[MinTemp_location$Location==myData1[i,"Location"],2])
    }
```

```
MaxTemp_location <- myData1 %>%
      group_by(Location)%>%
      summarise(median(MaxTemp, na.rm=TRUE))

    placeNA  <- which(is.na(myData1$MaxTemp))
    for (i in placeNA ) {
      myData1$MaxTemp[i] <- as.numeric(MaxTemp_location[MaxTemp_location$Location==myData1[i,"Location"],2])
    }
```

```
Rainfall_location <- myData1 %>%
      group_by(Location)%>%
      summarise(median(Rainfall, na.rm=TRUE))

    placeNA  <- which(is.na(myData1$Rainfall))
    for (i in placeNA ) {
      myData1$Rainfall[i] <- as.numeric(Rainfall_location[Rainfall_location$Location==myData1[i,"Location"],2])
    }
```

```
    placeNA  <- which((myData1$WindGustDir == ''))
    for (i in placeNA ) {
      myData1$WindGustDir[i] <- getmode(myData1$WindGustDir)
    }
```

```
myData1$WindGustSpeed[which(is.na(myData1$WindGustSpeed))] <- median(myData1$WindGustSpeed, na.rm = TRUE)
```

```
    placeNA  <- which(myData1$WindDir == "")
   for (i in placeNA ) {
      myData1$WindDir[i] <- getmode(myData1$WindDir)
    }
```

```
windspeed_location <- myData1 %>%
      group_by(Location)%>%
      summarise(median(WindSpeed, na.rm=TRUE))

    placeNA  <- which(is.na(myData1$WindSpeed))
    for (i in placeNA ) {
      myData1$WindSpeed[i] <- as.numeric(windspeed_location[windspeed_location$Location==myData1[i,"Location"],2]
)
    }
```

```
humidity_location <- myData1 %>%
      group_by(Location)%>%
      summarise(median(Humidity, na.rm=TRUE))

    placeNA  <- which(is.na(myData1$Humidity))
    for (i in placeNA ) {
      myData1$Humidity[i] <- as.numeric(humidity_location[humidity_location$Location==myData1[i,"Location"],2])
    }
```

```
placeNA  <- which(is.na(myData1$Pressure))
    for (i in placeNA ) {
      myData1$Pressure[i] <- median(myData1$Pressure, na.rm = TRUE)
    }
```

```
temp_location <- myData1 %>%
      group_by(Location)%>%
      summarise(median(Temp, na.rm=TRUE))

    placeNA  <- which(is.na(myData1$Temp))
    for (i in placeNA ) {
      myData1$Temp[i] <- as.numeric(temp_location[temp_location$Location==myData1[i,"Location"],2])
    }
```

Replace Na's/blanks in Rain Today with No

```
places<- which(myData1$RainToday == "")
```

```
myData1$RainToday[places] <- as.factor("No")
```

View structure and summary of the final treated table **There are no missing values in any of the columns after treating all the columns**

```
str(myData1)
```

```
## 'data.frame':    51959 obs. of  13 variables:
##  $ Location     : Factor w/ 49 levels "Adelaide","Albany",..: 27 29 15 7 44 20 21 46 45 26 ...
##  $ MinTemp      : num  18.9 11.1 15.9 0 9.1 6.6 9.7 17.1 13.5 12.5 ...
##  $ MaxTemp      : num  23.7 20.8 19.5 14.9 22.7 16.3 20.6 21.9 23.5 21.3 ...
##  $ Rainfall     : num  0 0 17.6 0 0 ...
##  $ WindGustDir  : Factor w/ 17 levels "","E","ENE","ESE",..: 12 15 11 17 7 10 16 12 9 15 ...
##  $ WindGustSpeed: int  41 39 44 35 41 50 61 54 46 65 ...
##  $ WindDir      : Factor w/ 17 levels "","E","ENE","ESE",..: 12 17 14 16 4 10 9 12 6 15 ...
##  $ WindSpeed    : num  28 26 9 19 7 30 20 28 19 22 ...
##  $ Humidity     : num  55 48 99 55 40 55 35 99 53 82 ...
##  $ Pressure     : num  1026 1014 1028 1023 1027 ...
##  $ Temp         : num  22.4 19.5 17.8 14.2 22.5 14.6 19.5 17.9 22.2 13.1 ...
##  $ RainToday    : Factor w/ 3 levels "","No","Yes": 2 2 3 2 2 3 2 3 2 2 ...
##  $ RainTomorrow : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 1 1 2 2 2 ...
```

```
colSums(is.na(myData1))
```

```
##      Location       MinTemp       MaxTemp      Rainfall   WindGustDir
##             0             0             0             0             0
## WindGustSpeed       WindDir     WindSpeed      Humidity      Pressure
##             0             0             0             0             0
##          Temp     RainToday  RainTomorrow
##             0             0             0
```

##Creating Dummies for categorical variables

```
myData_dummies <- fastDummies::dummy_cols(myData1,  select_columns = c('Location','WindGustDir','WindDir','RainTod
ay','RainTomorrow'))
```

```
myData_dummies$WindGustDir_ <- NULL
myData_dummies$WindDir_ <- NULL
myData_dummies$RainToday_ <- NULL
```

```
head(myData_dummies)
```

```
##           Location MinTemp MaxTemp Rainfall WindGustDir WindGustSpeed
## 1        NorahHead    18.9    23.7      0.0         SSE            41
## 2        Nuriootpa    11.1    20.8      0.0           W            39
## 3        GoldCoast    15.9    19.5     17.6          SE            44
## 4          Bendigo     0.0    14.9      0.0         WSW            35
## 5          Walpole     9.1    22.7      0.0         NNE            41
## 6 MelbourneAirport     6.6    16.3      8.4           S            50
##   WindDir WindSpeed Humidity Pressure Temp RainToday RainTomorrow
## 1     SSE        28       55   1026.0 22.4        No          Yes
## 2     WSW        26       48   1014.4 19.5        No           No
## 3      SW         9       99   1028.5 17.8       Yes          Yes
## 4     WNW        19       55   1023.0 14.2        No          Yes
## 5     ESE         7       40   1027.1 22.5        No           No
## 6       S        30       55   1021.4 14.6       Yes           No
##   Location_Adelaide Location_Albany Location_Albury Location_AliceSprings
## 1                 0                0               0                      0
## 2                 0                0               0                      0
## 3                 0                0               0                      0
## 4                 0                0               0                      0
## 5                 0                0               0                      0
## 6                 0                0               0                      0
##   Location_BadgerysCreek Location_Ballarat Location_Bendigo
## 1                      0                 0                0
## 2                      0                 0                0
## 3                      0                 0                0
## 4                      0                 0                1
## 5                      0                 0                0
## 6                      0                 0                0
##   Location_Brisbane Location_Cairns Location_Canberra Location_Cobar
## 1                 0                0                 0              0
## 2                 0                0                 0              0
## 3                 0                0                 0              0
## 4                 0                0                 0              0
## 5                 0                0                 0              0
## 6                 0                0                 0              0
##   Location_CoffsHarbour Location_Dartmoor Location_Darwin
## 1                     0                 0               0
## 2                     0                 0               0
## 3                     0                 0               0
## 4                     0                 0               0
## 5                     0                 0               0
## 6                     0                 0               0
##   Location_GoldCoast Location_Hobart Location_Katherine
## 1                  0                0                  0
## 2                  0                0                  0
## 3                  1                0                  0
## 4                  0                0                  0
## 5                  0                0                  0
## 6                  0                0                  0
##   Location_Launceston Location_Melbourne Location_MelbourneAirport
## 1                   0                  0                         0
## 2                   0                  0                         0
## 3                   0                  0                         0
## 4                   0                  0                         0
## 5                   0                  0                         0
```

```
## 6                   0                 0                         1
##   Location_Mildura Location_Moree Location_MountGambier
## 1                0              0                     0
## 2                0              0                     0
## 3                0              0                     0
## 4                0              0                     0
## 5                0              0                     0
## 6                0              0                     0
##   Location_MountGinini Location_Newcastle Location_Nhil Location_NorahHead
## 1                    0                  0             0                  1
## 2                    0                  0             0                  0
## 3                    0                  0             0                  0
## 4                    0                  0             0                  0
## 5                    0                  0             0                  0
## 6                    0                  0             0                  0
##   Location_NorfolkIsland Location_Nuriootpa Location_PearceRAAF
## 1                      0                  0                   0
## 2                      0                  1                   0
## 3                      0                  0                   0
## 4                      0                  0                   0
## 5                      0                  0                   0
## 6                      0                  0                   0
##   Location_Penrith Location_Perth Location_PerthAirport Location_Portland
## 1                0              0                     0                 0
## 2                0              0                     0                 0
## 3                0              0                     0                 0
## 4                0              0                     0                 0
## 5                0              0                     0                 0
## 6                0              0                     0                 0
##   Location_Richmond Location_Sale Location_SalmonGums Location_Sydney
## 1                 0             0                   0               0
## 2                 0             0                   0               0
## 3                 0             0                   0               0
## 4                 0             0                   0               0
## 5                 0             0                   0               0
## 6                 0             0                   0               0
##   Location_SydneyAirport Location_Townsville Location_Tuggeranong
## 1                      0                   0                    0
## 2                      0                   0                    0
## 3                      0                   0                    0
## 4                      0                   0                    0
## 5                      0                   0                    0
## 6                      0                   0                    0
##   Location_Uluru Location_WaggaWagga Location_Walpole Location_Watsonia
## 1              0                   0                0                 0
## 2              0                   0                0                 0
## 3              0                   0                0                 0
## 4              0                   0                0                 0
## 5              0                   0                1                 0
## 6              0                   0                0                 0
##   Location_Williamtown Location_Witchcliffe Location_Wollongong
## 1                    0                    0                   0
## 2                    0                    0                   0
## 3                    0                    0                   0
## 4                    0                    0                   0
## 5                    0                    0                   0
## 6                    0                    0                   0
##   Location_Woomera WindGustDir_E WindGustDir_ENE WindGustDir_ESE
## 1                0             0                0               0
## 2                0             0                0               0
## 3                0             0                0               0
## 4                0             0                0               0
## 5                0             0                0               0
## 6                0             0                0               0
##   WindGustDir_N WindGustDir_NE WindGustDir_NNE WindGustDir_NNW
## 1             0              0                0               0
## 2             0              0                0               0
## 3             0              0                0               0
## 4             0              0                0               0
## 5             0              0                1               0
## 6             0              0                0               0
##   WindGustDir_NW WindGustDir_S WindGustDir_SE WindGustDir_SSE
## 1              0             0              0              1
## 2              0             0              0              0
## 3              0             0              1              0
## 4              0             0              0              0
## 5              0             0              0              0
## 6              0             1              0              0
##   WindGustDir_SSW WindGustDir_SW WindGustDir_W WindGustDir_WNW
## 1               0              0             0               0
## 2               0              0             1               0
## 3               0              0             0               0
## 4               0              0             0               0
## 5               0              0             0               0
## 6               0              0             0               0
##   WindGustDir_WSW WindDir_E WindDir_ENE WindDir_ESE WindDir_N WindDir_NE
## 1               0         0           0           0         0          0
## 2               0         0           0           0         0          0
## 3               0         0           0           0         0          0
## 4               1         0           0           0         0          0
## 5               0         0           0           1         0          0
## 6               0         0           0           0         0          0
##   WindDir_NNE WindDir_NNW WindDir_NW WindDir_S WindDir_SE WindDir_SSE
## 1           0           0          0         0          0           1
```

```
## 2           0           0          0         0          0           0
## 3           0           0          0         0          0           0
## 4           0           0          0         0          0           0
## 5           0           0          0         0          0           0
## 6           0           0          0         1          0           0
##   WindDir_SSW WindDir_SW WindDir_W WindDir_WNW WindDir_WSW RainToday_No
## 1           0          0         0           0           0            1
## 2           0          0         0           0           1            1
## 3           0          1         0           0           0            0
## 4           0          0         0           1           0            1
## 5           0          0         0           0           0            1
## 6           0          0         0           0           0            0
##   RainToday_Yes RainTomorrow_No RainTomorrow_Yes
## 1             0               0                1
## 2             0               1                0
## 3             1               0                1
## 4             0               0                1
## 5             0               1                0
## 6             1               1                0
```

Removing Outliers using z score approach

```
myData_dummies$MinTempZscore <- scale(myData_dummies$MinTemp)
nrow(myData_dummies[abs(myData_dummies$MinTempZscore)>3,])
```

```
## [1] 9
```

```
myData_dummies <- myData_dummies[abs(myData_dummies$MinTempZscore)<3,]

myData_dummies$MaxTempZscore <- scale(myData_dummies$MaxTemp)
nrow(myData_dummies[abs(myData_dummies$MaxTempZscore)>3,])
```

```
## [1] 139
```

```
myData_dummies <- myData_dummies[abs(myData_dummies$MaxTempZscore)<3,]

myData_dummies$RainfallZscore <- scale(myData_dummies$Rainfall)
nrow(myData_dummies[abs(myData_dummies$RainfallZscore)>4,])
```

```
## [1] 572
```

```
myData_dummies <- myData_dummies[abs(myData_dummies$RainfallZscore)<4,]

myData_dummies$WindSpeedZscore <- scale(myData_dummies$WindSpeed)
nrow(myData_dummies[abs(myData_dummies$WindSpeedZscore)>3,])
```

```
## [1] 311
```

```
myData_dummies <- myData_dummies[abs(myData_dummies$WindSpeedZscore)<3,]

myData_dummies$WindGustSpeedZscore <- scale(myData_dummies$WindGustSpeed)
nrow(myData_dummies[abs(myData_dummies$WindGustSpeedZscore)>3,])
```

```
## [1] 453
```

```
myData_dummies <- myData_dummies[abs(myData_dummies$WindGustSpeedZscore)<3,]

myData_dummies$HumidityZscore <- scale(myData_dummies$Humidity)
nrow(myData_dummies[abs(myData_dummies$HumidityZscore)>3,])
```

```
## [1] 0
```

```
myData_dummies$PressureZscore <- scale(myData_dummies$Pressure)
nrow(myData_dummies[abs(myData_dummies$PressureZscore)>3,])
```

```
## [1] 240
```

```
myData_dummies <- myData_dummies[abs(myData_dummies$PressureZscore)<3,]

myData_dummies$TempZscore <- scale(myData_dummies$Temp)
nrow(myData_dummies[abs(myData_dummies$TempZscore)>3,])
```

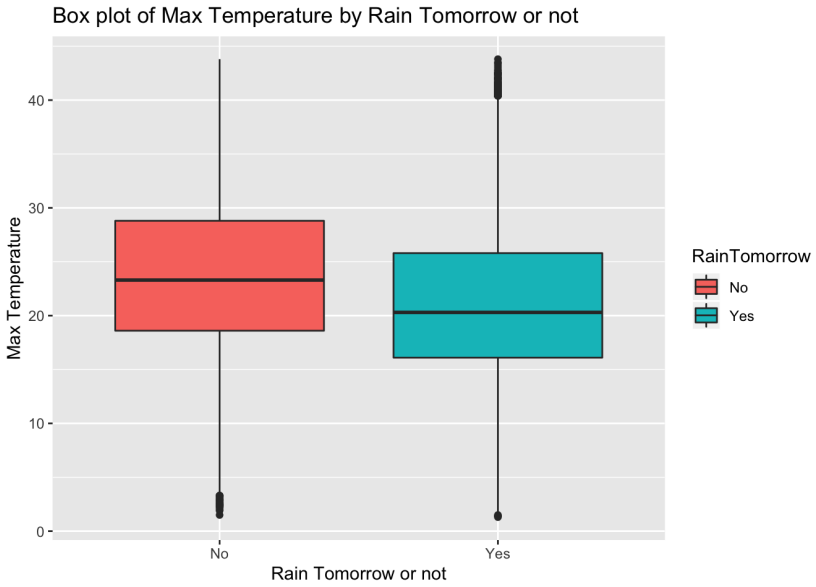
```
## [1] 68
```

```
myData_dummies <- myData_dummies[abs(myData_dummies$TempZscore)<3,]

myData_dummies$MinTempZscore <- NULL

myData_dummies$MaxTempZscore <- NULL

myData_dummies$RainfallZscore <- NULL

myData_dummies$WindSpeedZscore <- NULL

myData_dummies$WindGustSpeedZscore <- NULL

myData_dummies$HumidityZscore <- NULL

myData_dummies$PressureZscore <- NULL

myData_dummies$TempZscore <- NULL
```

OUR DATA IS CLEANED

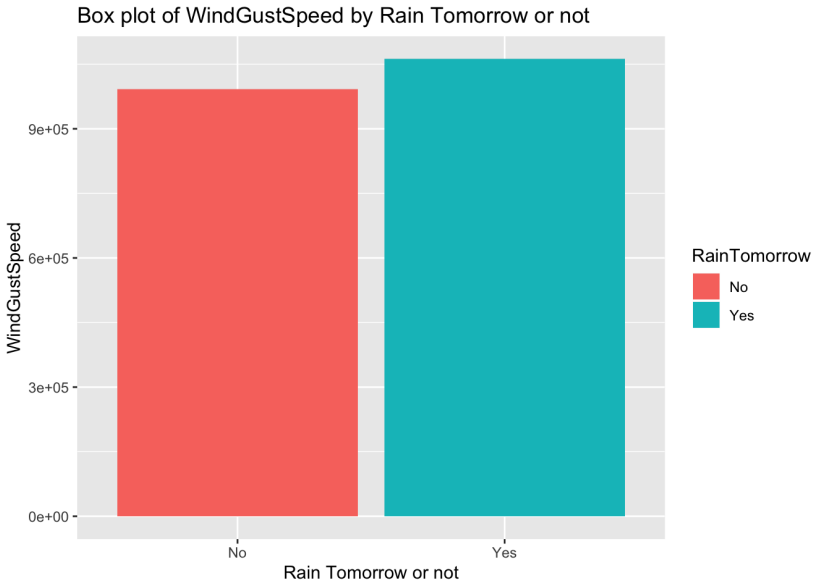
Step 2: EDA

```
Plot1 <- ggplot(myData_dummies,aes(RainTomorrow,MaxTemp))+geom_boxplot(aes(fill=RainTomorrow))+
          xlab("Rain Tomorrow or not")+ ylab("Max Temperature")+ ggtitle("Box plot of Max Temperature by Rain
Tomorrow or not")
Plot1
```



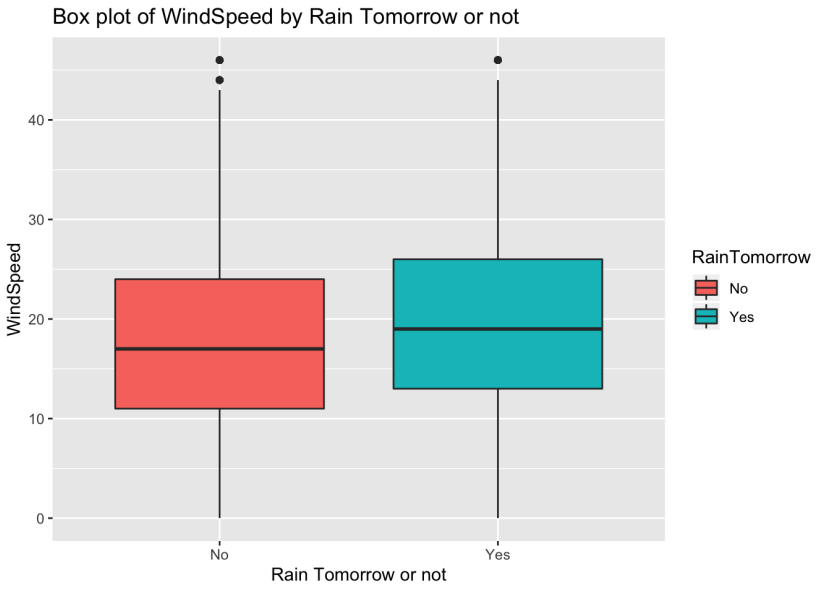
Analysis: We can observe that when it does not rain tomorrow the median Max Temperature is higher

```
Plot2 <- ggplot(myData_dummies,aes(RainTomorrow,WindGustSpeed))+geom_col(aes(fill=RainTomorrow))+
          xlab("Rain Tomorrow or not")+ ylab("WindGustSpeed")+ ggtitle("Box plot of WindGustSpeed by Rain Tomo
rrow or not")
Plot2
```



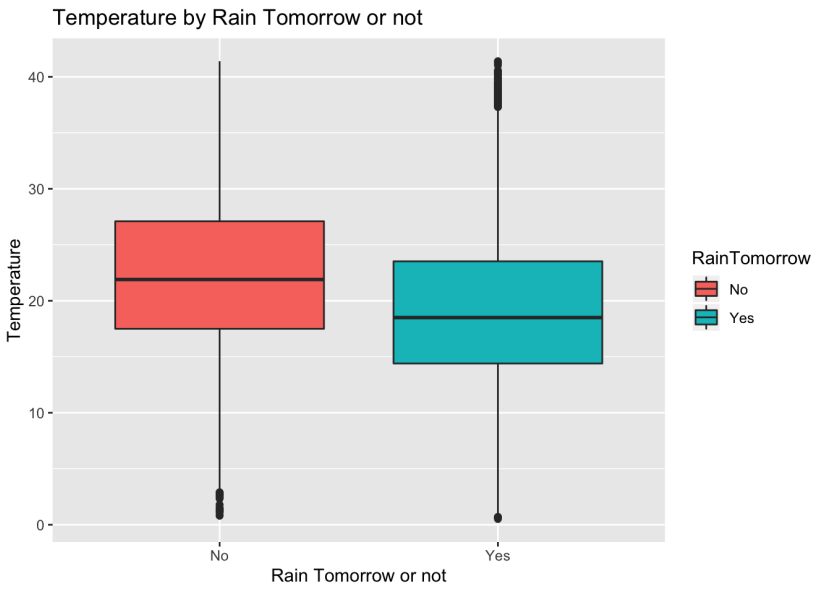
Analysis: We can observe that when it rains tomorrow the median Wind Gust Speed is higher

```
Plot3 <- ggplot(myData_dummies,aes(RainTomorrow,WindSpeed))+geom_boxplot(aes(fill=RainTomorrow))+
            xlab("Rain Tomorrow or not")+ ylab("WindSpeed")+ ggtitle("Box plot of WindSpeed by Rain Tomorrow or
not")
Plot3
```



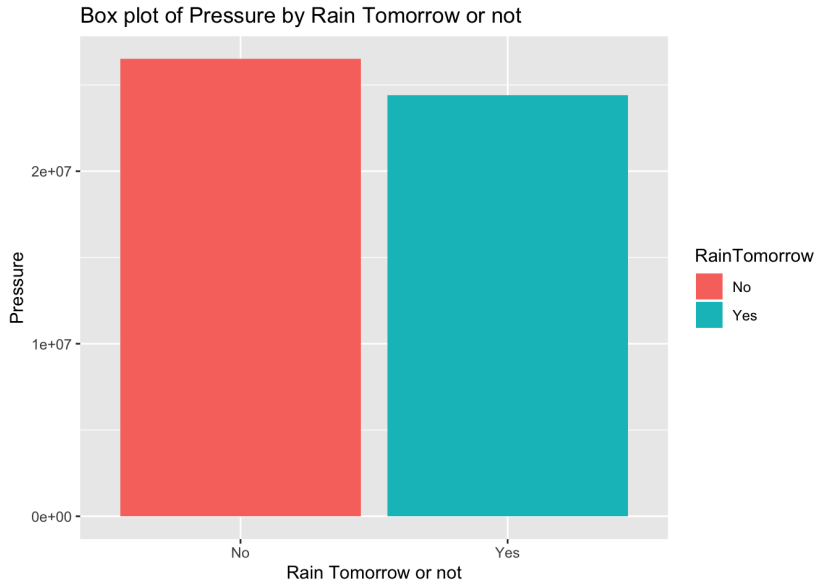
Analysis: We can observe that when it rains tomorrow the median Wind Speed is higher

```
plot4 <- ggplot(myData_dummies,aes(RainTomorrow,Temp))+geom_boxplot(aes(fill=RainTomorrow))+
            xlab("Rain Tomorrow or not")+ ylab("Temperature")+ ggtitle("Temperature by Rain Tomorrow or not")
plot4
```



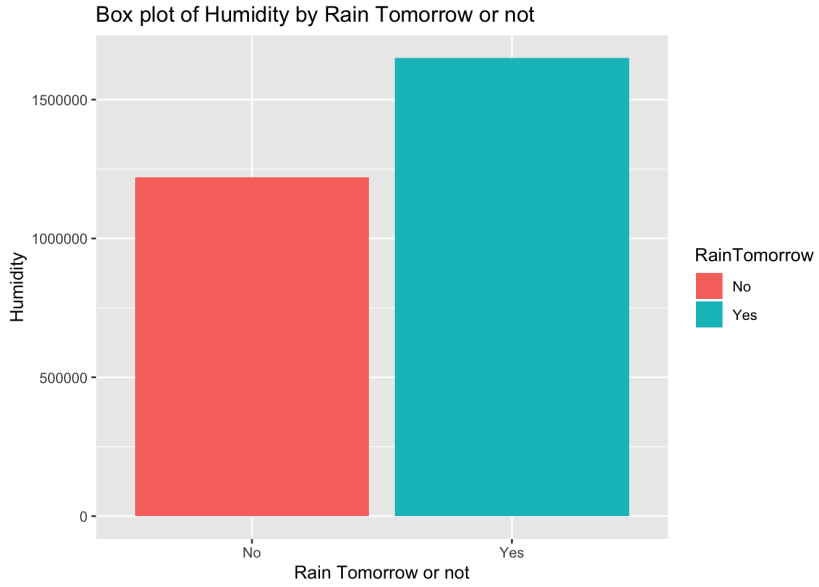
Analysis: We can observe that when it does not rain tomorrow the median temperature is higher

```
Plot5 <- ggplot(myData_dummies,aes(RainTomorrow,Pressure))+geom_col(aes(fill=RainTomorrow))+
            xlab("Rain Tomorrow or not")+ ylab("Pressure")+ ggtitle("Box plot of Pressure by Rain Tomorrow or not
")
Plot5
```



Analysis: We can observe that when it does not rain tomorrow the median pressure is higher

```
Plot6 <- ggplot(myData_dummies,aes(RainTomorrow,Humidity))+geom_col(aes(fill=RainTomorrow))+
            xlab("Rain Tomorrow or not")+ ylab("Humidity")+ ggtitle("Box plot of Humidity by Rain Tomorrow or not
")
Plot6
```



Analysis: We can observe that when it rains tomorrow the median Humidity is higher

Step3: CLUSTERING

```
myData_cluster <- myData_dummies
```

Remove the Categorical non dummy columns to perform clustering and also remove the target variable Rain Tomorrow Yes

```
myData_cluster$Location <- NULL
myData_cluster$WindGustDir <- NULL
myData_cluster$WindDir <- NULL
myData_cluster$RainToday <- NULL
myData_cluster$RainTomorrow <- NULL
myData_cluster$RainToday_No <- NULL
myData_cluster$RainTomorrow_No <- NULL
myData_cluster$RainTomorrow_Yes <- NULL
```

Normalising Data

```
normalize <- function(x) {
return ((x - min(x)) / (max(x) - min(x)))
}
```

```
myData_cluster$MinTemp <- normalize(myData_cluster$MinTemp)
myData_cluster$MaxTemp <- normalize(myData_cluster$MaxTemp)
myData_cluster$Rainfall <- normalize(myData_cluster$Rainfall)
myData_cluster$WindGustSpeed <- normalize(myData_cluster$WindGustSpeed)
myData_cluster$WindSpeed <- normalize(myData_cluster$WindSpeed)
myData_cluster$Humidity <- normalize(myData_cluster$Humidity)
myData_cluster$Pressure <- normalize(myData_cluster$Pressure)
myData_cluster$Temp <- normalize(myData_cluster$Temp)
str(myData_cluster)
```

```
## 'data.frame':    50167 obs. of  90 variables:
##  $ MinTemp                 : num  0.67 0.464 0.591 0.172 0.412 ...
##  $ MaxTemp                 : num  0.527 0.459 0.428 0.32 0.504 ...
##  $ Rainfall                : num  0 0 0.373 0 0 ...
##  $ WindGustSpeed           : num  0.459 0.432 0.5 0.378 0.459 ...
##  $ WindSpeed               : num  0.609 0.565 0.196 0.413 0.152 ...
##  $ Humidity                : num  0.545 0.475 0.99 0.545 0.394 ...
##  $ Pressure                : num  0.784 0.5 0.846 0.711 0.811 ...
##  $ Temp                    : num  0.535 0.465 0.423 0.335 0.538 ...
##  $ Location_Adelaide       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Albany         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Albury         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_AliceSprings   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_BadgerysCreek  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Ballarat       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Bendigo        : int  0 0 0 1 0 0 0 0 0 1 ...
##  $ Location_Brisbane       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Cairns         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Canberra       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Cobar          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_CoffsHarbour   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Dartmoor       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Darwin         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_GoldCoast      : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ Location_Hobart         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Katherine      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Launceston     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Melbourne      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_MelbourneAirport: int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Location_Mildura        : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Location_Moree          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_MountGambier   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_MountGinini    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Newcastle      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Nhil           : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ Location_NorahHead      : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Location_NorfolkIsland  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Nuriootpa      : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Location_PearceRAAF     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Penrith        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Perth          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_PerthAirport   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Portland       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Richmond       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Sale           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_SalmonGums     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Sydney         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_SydneyAirport  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Townsville     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Tuggeranong    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Uluru          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_WaggaWagga     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Walpole        : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Location_Watsonia       : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Location_Williamtown    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Witchcliffe    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Wollongong     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Woomera        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_E           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_ENE         : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ WindGustDir_ESE         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_N           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_NE          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_NNE         : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ WindGustDir_NNW         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_NW          : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ WindGustDir_S           : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ WindGustDir_SE          : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ WindGustDir_SSE         : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_SSW         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_SW          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_W           : int  0 1 0 0 0 0 0 0 1 0 ...
##  $ WindGustDir_WNW         : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ WindGustDir_WSW         : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ WindDir_E               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_ENE             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_ESE             : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ WindDir_N               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_NE              : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ WindDir_NNE             : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ WindDir_NNW             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_NW              : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ WindDir_S               : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ WindDir_SE              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_SSE             : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_SSW             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_SW              : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ WindDir_W               : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ WindDir_WNW             : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ WindDir_WSW             : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ RainToday_Yes           : int  0 0 1 0 0 1 0 0 0 0 ...
```

Kmeans

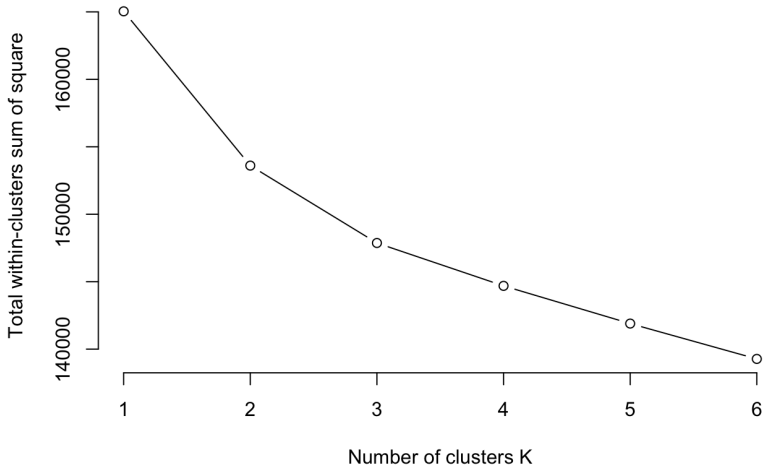
Using Elbow Curve to identify the optimum number of clusters

```
set.seed(10)
wss <- function(k){
  return(kmeans(myData_cluster, k, nstart = 25)$tot.withinss)
}
```

```
k_values <- 1:6
wss_values <-c()
for (i in k_values)
{

  wss_values<- c(wss_values,wss(i))
}
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 2508350)
```

```
plot(x = k_values, y = wss_values,
     type = "b", frame = F,
     xlab = "Number of clusters K",
     ylab = "Total within-clusters sum of square")
```



Analysis: We can observe that the steep drop in SSE reduces after K = 2 and the drop becomes more gradual. Therefore optimum number of cluster would be 2

The Hyperparameters for K Means Clustering are Number of Clusters K, initial choice of centroids & number of repeats, i.e. centers, nstart, iter.max are the hyperparameters that can be tuned to optimize the objective function for K Means which is total sum of squared errors (tot.withinss)

The best model will have the lowest total SSE

```
km1 <- kmeans(myData_cluster, centers = 2, nstart = 50, iter.max = 100, algorithm = "Hartigan-Wong")
str(km1)
```

```
## List of 9
##  $ cluster     : Named int [1:50167] 2 2 1 2 2 1 2 2 2 2 ...
##   ..- attr(*, "names")= chr [1:50167] "1" "2" "3" "4" ...
##  $ centers     : num [1:2, 1:90] 0.519 0.495 0.44 0.525 0.192 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:90] "MinTemp" "MaxTemp" "Rainfall" "WindGustSpeed" ...
##  $ totss       : num 165041
##  $ withinss    : num [1:2] 44535 109069
##  $ tot.withinss: num 153603
##  $ betweenss   : num 11438
##  $ size        : int [1:2] 14550 35617
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
km2 <- kmeans(myData_cluster, centers = 2, nstart = 150, iter.max = 350)
str(km2)
```

```
## List of 9
##  $ cluster     : Named int [1:50167] 2 2 1 2 2 1 2 2 2 2 ...
##   ..- attr(*, "names")= chr [1:50167] "1" "2" "3" "4" ...
##  $ centers     : num [1:2, 1:90] 0.519 0.495 0.44 0.525 0.192 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:90] "MinTemp" "MaxTemp" "Rainfall" "WindGustSpeed" ...
##  $ totss       : num 165041
##  $ withinss    : num [1:2] 44535 109069
##  $ tot.withinss: num 153603
##  $ betweenss   : num 11438
##  $ size        : int [1:2] 14550 35617
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
km3 <- kmeans(myData_cluster, centers = 3, nstart = 50, iter.max = 100)
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 2508350)

## Warning: Quick-TRANSfer stage steps exceeded maximum (= 2508350)
```

```
str(km3)
```

```
## List of 9
##  $ cluster     : Named int [1:50167] 1 2 3 1 1 3 1 1 2 1 ...
##   ..- attr(*, "names")= chr [1:50167] "1" "2" "3" "4" ...
##  $ centers     : num [1:3, 1:90] 0.497 0.483 0.526 0.53 0.463 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "1" "2" "3"
##   .. ..$ : chr [1:90] "MinTemp" "MaxTemp" "Rainfall" "WindGustSpeed" ...
##  $ totss       : num 165041
##  $ withinss    : num [1:3] 95029 15706 37132
##  $ tot.withinss: num 147868
##  $ betweenss   : num 17174
##  $ size        : int [1:3] 30999 7089 12079
##  $ iter        : int 3
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
km4 <- kmeans(myData_cluster, centers = 3, nstart = 150, iter.max = 350)
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 2508350)
```

```
str(km4)
```

```
## List of 9
##  $ cluster     : Named int [1:50167] 1 2 3 1 1 3 1 1 2 1 ...
##   ..- attr(*, "names")= chr [1:50167] "1" "2" "3" "4" ...
##  $ centers     : num [1:3, 1:90] 0.497 0.483 0.526 0.53 0.463 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "1" "2" "3"
##   .. ..$ : chr [1:90] "MinTemp" "MaxTemp" "Rainfall" "WindGustSpeed" ...
##  $ totss       : num 165041
##  $ withinss    : num [1:3] 95029 15706 37132
##  $ tot.withinss: num 147868
##  $ betweenss   : num 17174
##  $ size        : int [1:3] 30999 7089 12079
##  $ iter        : int 3
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

Total sum of squared errors:

```
km1$tot.withinss
```

```
## [1] 153603.5
```

```
km2$tot.withinss
```

```
## [1] 153603.5
```

```
km3$tot.withinss
```

```
## [1] 147867.8
```

```
km4$tot.withinss
```

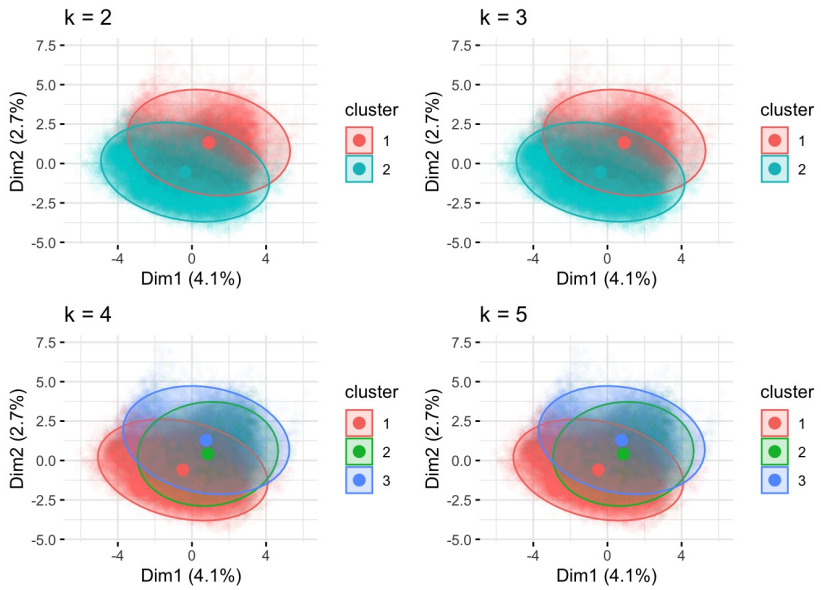
```
## [1] 147867.8
```

```
# plots to compare
p1 <- fviz_cluster(km1, data = myData_cluster,geom = "point",repel = FALSE,ggtheme = theme_minimal(),alpha=0.02,sh
ape = 19,ellipse.type = "norm") + ggtitle("k = 2")
p2 <- fviz_cluster(km2, data = myData_cluster,geom = "point",repel = FALSE,ggtheme = theme_minimal(),alpha=0.02,sh
ape = 19,ellipse.type = "norm") + ggtitle("k = 3")
p3 <- fviz_cluster(km3, data = myData_cluster,geom = "point",repel = FALSE,ggtheme = theme_minimal(),alpha=0.02,sh
ape = 19,ellipse.type = "norm") + ggtitle("k = 4")
p4 <- fviz_cluster(km4, data = myData_cluster,geom = "point",repel = FALSE,ggtheme = theme_minimal(),alpha=0.02,sh
ape = 19,ellipse.type = "norm") + ggtitle("k = 5")

library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
grid.arrange(p1, p2, p3, p4, nrow = 2)
```



Performance Evaluation

The measures of classification such as Entropy, Precision, Recall and F Score can be used to evaluate the extent to which a cluster contains objects of a single class.

```
length(km1$cluster[km1$cluster==1])
```

```
## [1] 14550
```

```
length(km1$cluster[km1$cluster==2])
```

```
## [1] 35617
```

```
myData_eval <- myData_cluster
```

```
myData_eval$RainTomorrowYes <- myData_dummies$RainTomorrow_Yes
myData_eval$clusterassigned <- km1$cluster
```

Count of Rain Tomorrow binary labels, present in each cluster

```
cluster <- c(1,2)
yes <- c(length(myData_eval[myData_eval$clusterassigned ==1 & myData_eval$RainTomorrowYes==1,1]),length(myData_eva
l[myData_eval$clusterassigned ==2 & myData_eval$RainTomorrowYes==1,1]))
no <- c(length(myData_eval[myData_eval$clusterassigned ==1 & myData_eval$RainTomorrowYes==0,1]),length(myData_eval
[myData_eval$clusterassigned ==2 & myData_eval$RainTomorrowYes==0,1]))
cluster_performance_eval_df <- data.frame(cluster,yes,no)
```

```
ent_clust1 <- 0
ent_clust2 <- 0
for (i in as.numeric(as.vector(cluster_performance_eval_df[cluster_performance_eval_df$cluster==1,c("yes","no")]))
)
  {
ent_clust1 <- ent_clust1 + (i/sum(as.numeric(as.vector(cluster_performance_eval_df[cluster_performance_eval_df$clu
ster==1,c("yes","no")])))*(log2(i/sum(as.numeric(as.vector(cluster_performance_eval_df[cluster_performance_eval_d
f$cluster==1,c("yes","no")]))))
}
ent_clust1 <- -1*(ent_clust1)
paste("Entropy of cluster 1 is: ",ent_clust1)
```

```
## [1] "Entropy of cluster 1 is:  0.834425597807652"
```

```
for (i in as.numeric(as.vector(cluster_performance_eval_df[cluster_performance_eval_df$cluster==2,c("yes","no")]))
)
  {
ent_clust2 <- ent_clust2 + (i/sum(as.numeric(as.vector(cluster_performance_eval_df[cluster_performance_eval_df$clu
ster==2,c("yes","no")])))*(log2(i/sum(as.numeric(as.vector(cluster_performance_eval_df[cluster_performance_eval_d
f$cluster==2,c("yes","no")]))))
}
ent_clust2 <- -1*(ent_clust2)
paste("Entropy of cluster 2 is: ",ent_clust2)
```

```
## [1] "Entropy of cluster 2 is:  0.955168056010434"
```

From the below results we can observe that cluster 1 is more pure than cluster 2

Measuring Precision, Recall and F Measure of Rain Tomorrow 'Yes' in cluster 1 & cluster 2

```
precision_yes_clust1 = (cluster_performance_eval_df$yes[cluster_performance_eval_df$cluster==1])/sum(as.numeric(as
.vector(cluster_performance_eval_df[cluster_performance_eval_df$cluster==1,c("yes","no")])))

recall_yes_clust1 = (cluster_performance_eval_df$yes[cluster_performance_eval_df$cluster==1])/sum(cluster_performa
nce_eval_df$yes)

F_yes_clust1 = (2*precision_yes_clust1*recall_yes_clust1)/(precision_yes_clust1+recall_yes_clust1)

paste("Precision of class Yes for Rain Tomorrow in cluster 1 is:",precision_yes_clust1)
```

```
## [1] "Precision of class Yes for Rain Tomorrow in cluster 1 is: 0.734845360824742"
```

```
paste("Recall of class Yes for Rain Tomorrow in cluster 1 is:",recall_yes_clust1)
```

```
## [1] "Recall of class Yes for Rain Tomorrow in cluster 1 is: 0.443946188340807"
```

```
paste("F Measure of class Yes for Rain Tomorrow in cluster 1 is:",F_yes_clust1)
```

```
## [1] "F Measure of class Yes for Rain Tomorrow in cluster 1 is: 0.553502096598851"
```

```
precision_yes_clust2 = (cluster_performance_eval_df$yes[cluster_performance_eval_df$cluster==2])/sum(as.numeric(as
.vector(cluster_performance_eval_df[cluster_performance_eval_df$cluster==2,c("yes","no")])))

recall_yes_clust2 = (cluster_performance_eval_df$yes[cluster_performance_eval_df$cluster==2])/sum(cluster_performa
nce_eval_df$yes)

F_yes_clust2 = (2*precision_yes_clust2*recall_yes_clust2)/(precision_yes_clust2+recall_yes_clust2)

paste("Precision of class Yes for Rain Tomorrow in cluster 2 is:",precision_yes_clust2)
```

```
## [1] "Precision of class Yes for Rain Tomorrow in cluster 2 is: 0.376000224611843"
```

```
paste("Recall of class Yes for Rain Tomorrow in cluster 2 is:",recall_yes_clust2)
```

```
## [1] "Recall of class Yes for Rain Tomorrow in cluster 2 is: 0.556053811659193"
```

```
paste("F Measure of class Yes for Rain Tomorrow in cluster 2 is:",F_yes_clust2)
```

```
## [1] "F Measure of class Yes for Rain Tomorrow in cluster 2 is: 0.448635701244535"
```

Precision of 'Yes' in cluster 1 is the number of 'Yes' class in cluster 1 out of total number of datapoints in cluster 1
Recall of 'Yes' in cluster 1 is the number of 'Yes' in clutser 1 out of the total number of 'Yes' present across the 2 clusters
F Measure is the harmonic mean of Precision & Recall. In this case, clutser 1 has higher F measire for class 'Yes'.

Since cluster 1 has higher F Measure as seen above, Cluster 1 is being used to classify Rain Tomorrow as 'Yes' and cluster 2 is used to classify Rain Tomorrow as 'No'

```
paste("The average Rainfall for cluster 1 is: ",mean(myData_eval$Rainfall[myData_eval$clusterassigned ==1]))
```

```
## [1] "The average Rainfall for cluster 1 is:  0.191915370726309"
```

```
paste("The average MinTemp for cluster 1 is: ",mean(myData_eval$MinTemp[myData_eval$clusterassigned ==1]))
```

```
## [1] "The average MinTemp for cluster 1 is:  0.518878491962027"
```

```
paste("The average MaxTemp for cluster 1 is: ",mean(myData_eval$MaxTemp[myData_eval$clusterassigned ==1]))
```

```
## [1] "The average MaxTemp for cluster 1 is:  0.439741257327673"
```

```
paste("The average Humidity for cluster 1 is: ",mean(myData_eval$Humidity[myData_eval$clusterassigned ==1]))
```

```
## [1] "The average Humidity for cluster 1 is:  0.700965670450207"
```

```
paste("The average Pressure for cluster 1 is: ",mean(myData_eval$Pressure[myData_eval$clusterassigned ==1]))
```

```
## [1] "The average Pressure for cluster 1 is:  0.477020340610472"
```

```
paste("The average Temp for cluster 1 is: ",mean(myData_eval$Temp[myData_eval$clusterassigned ==1]))
```

```
## [1] "The average Temp for cluster 1 is:  0.437353531789042"
```

```
paste("The average WindGustSpeed for cluster 1 is: ",mean(myData_eval$WindGustSpeed[myData_eval$clusterassigned ==
1]))
```

```
## [1] "The average WindGustSpeed for cluster 1 is:  0.500109594130213"
```

```
paste("The average Rainfall for cluster 2 is: ",mean(myData_eval$Rainfall[myData_eval$clusterassigned ==2]))
```

```
## [1] "The average Rainfall for cluster 2 is:  0.00191889656577058"
```

```
paste("The average MinTemp for cluster 2 is: ",mean(myData_eval$MinTemp[myData_eval$clusterassigned ==2]))
```

```
## [1] "The average MinTemp for cluster 2 is:  0.495203033326634"
```

```
paste("The average MaxTemp for cluster 2 is: ",mean(myData_eval$MaxTemp[myData_eval$clusterassigned ==2]))
```

```
## [1] "The average MaxTemp for cluster 2 is:  0.525408785295852"
```

```
paste("The average Humidity for cluster 2 is: ",mean(myData_eval$Humidity[myData_eval$clusterassigned ==2]))
```

```
## [1] "The average Humidity for cluster 2 is:  0.512984663151718"
```

```
paste("The average Pressure for cluster 2 is: ",mean(myData_eval$Pressure[myData_eval$clusterassigned ==2]))
```

```
## [1] "The average Pressure for cluster 2 is:  0.512907113781864"
```

```
paste("The average Temp for cluster 2 is: ",mean(myData_eval$Temp[myData_eval$clusterassigned ==2]))
```

```
## [1] "The average Temp for cluster 2 is:  0.524922784530587"
```

```
paste("The average WindGustSpeed for cluster 2 is: ",mean(myData_eval$WindGustSpeed[myData_eval$clusterassigned ==
2]))
```

```
## [1] "The average WindGustSpeed for cluster 2 is:  0.441573982663912"
```

Classifying cluster 1 as 'Yes' & cluster 2 as 'No' for Rain Tomorrow

```
myData_eval$RainTomorrowYesPred <- 1
myData_eval$RainTomorrowYesPred[myData_eval$clusterassigned==2] <- 0
```

Calculating Accuracy using Confusion matrix

```
confusion_matrix_kmeans <- data.frame(table(myData_eval$RainTomorrowYes,myData_eval$RainTomorrowYesPred))
colnames(confusion_matrix_kmeans) <- c("Actual class","Predicted Class","Count")

Accuracy_kmeans <- sum(confusion_matrix_kmeans$Count[confusion_matrix_kmeans$`Actual class`==confusion_matrix_kmea
ns$`Predicted Class`])/sum(confusion_matrix_kmeans$Count)
Precision_kmeans <- confusion_matrix_kmeans$Count[confusion_matrix_kmeans$`Actual class`==1 & confusion_matrix_kme
ans$`Predicted Class`==1]/sum(confusion_matrix_kmeans$Count[confusion_matrix_kmeans$`Predicted Class`==1])
Recall_kmeans <- confusion_matrix_kmeans$Count[confusion_matrix_kmeans$`Actual class`==1 & confusion_matrix_kmeans
$`Predicted Class`==1]/sum(confusion_matrix_kmeans$Count[confusion_matrix_kmeans$`Actual class`==1])
F1_score_kmeans <- (2*Precision_kmeans*Recall_kmeans)/(Precision_kmeans+Recall_kmeans)

paste("Accuracy of K Means Algorithm in classifying Rain Tomorrow is :",Accuracy_kmeans)
```

```
## [1] "Accuracy of K Means Algorithm in classifying Rain Tomorrow is : 0.656148464129806"
```

```
paste("Precision of K Means Algorithm in classifying Rain Tomorrow is :",Precision_kmeans)
```

```
## [1] "Precision of K Means Algorithm in classifying Rain Tomorrow is : 0.734845360824742"
```

```
paste("Recall of K Means Algorithm in classifying Rain Tomorrow is :",Recall_kmeans)
```

```
## [1] "Recall of K Means Algorithm in classifying Rain Tomorrow is : 0.443946188340807"
```

```
paste("F1 Score of K Means Algorithm in classifying Rain Tomorrow is :",F1_score_kmeans)
```

```
## [1] "F1 Score of K Means Algorithm in classifying Rain Tomorrow is : 0.553502096598851"
```

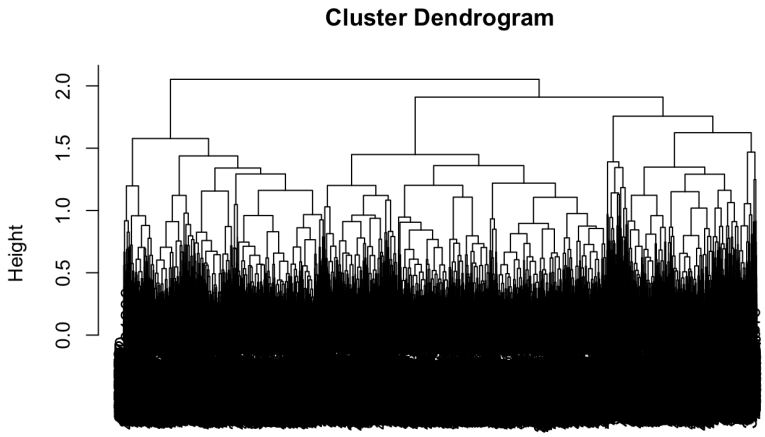
The clustering model has good Accuracy and Precision in classifying the dataset into 'Yes' & 'No' classes for the Rain Tomorrow attribute.

Step4: Hierarchical Agglomerative Clustering (HAC)

Considering half rows and reoving all dummy columns

```
myData_hac <- myData_cluster[seq(1,nrow(myData_cluster),2),]
myData_hac <- myData_hac[,-9:-90]
```

```
hac <- hclust(dist(myData_hac, method = "euclidean"), method = "complete")
plot(hac)
```



Since HAC has no objective function that can be optimized, there is no need to fine tune the hyperparameters

Cutting dendogram to get 2 clusters

```
hac_cut <- cutree(hac, 2)
```

```
hac_eval <- myData_hac
```

HAC Performance Evaluation

```
hac_eval$RainTomorrowYes <- myData_dummies$RainTomorrow_Yes[seq(1,nrow(myData_cluster),2)]
hac_eval$clusterassigned <- hac_cut
```

```
cluster <- c(1,2)
yes <- c(length(hac_eval[hac_eval$clusterassigned ==1 & hac_eval$RainTomorrowYes==1,1]),length(hac_eval[hac_eval$c
lusterassigned ==2 & hac_eval$RainTomorrowYes==1,1]))
no <- c(length(hac_eval[hac_eval$clusterassigned ==1 & hac_eval$RainTomorrowYes==0,1]),length(hac_eval[hac_eval$cl
usterassigned ==2 & hac_eval$RainTomorrowYes==0,1]))
cluster_performance_eval_df <- data.frame(cluster,yes,no)
```

```
ent_clust1 <- 0
ent_clust2 <- 0
for (i in as.numeric(as.vector(cluster_performance_eval_df[cluster_performance_eval_df$cluster==1,c("yes","no")]))
)
  {
ent_clust1 <- ent_clust1 + (i/sum(as.numeric(as.vector(cluster_performance_eval_df[cluster_performance_eval_df$clu
ster==1,c("yes","no")])))*(log2(i/sum(as.numeric(as.vector(cluster_performance_eval_df[cluster_performance_eval_d
f$cluster==1,c("yes","no")]))))
}
ent_clust1 <- -1*(ent_clust1)
paste("Entropy of cluster 1 is: ",ent_clust1)
```

```
## [1] "Entropy of cluster 1 is:  0.994024630045345"
```

```
for (i in as.numeric(as.vector(cluster_performance_eval_df[cluster_performance_eval_df$cluster==2,c("yes","no")]))
)
  {
ent_clust2 <- ent_clust2 + (i/sum(as.numeric(as.vector(cluster_performance_eval_df[cluster_performance_eval_df$clu
ster==2,c("yes","no")])))*(log2(i/sum(as.numeric(as.vector(cluster_performance_eval_df[cluster_performance_eval_d
f$cluster==2,c("yes","no")])))))
}
ent_clust2 <- -1*(ent_clust2)
paste("Entropy of cluster 2 is: ",ent_clust2)
```

```
## [1] "Entropy of cluster 2 is:  0.92255777961442"
```

From the below results we can observe that cluster 2 is more pure than cluster 1

Measure of Precision, Recall & F Measure of Rain Tomorrow 'Yes' in cluster 1 & cluster 2

```
precision_yes_clust1 = (cluster_performance_eval_df$yes[cluster_performance_eval_df$cluster==1])/sum(as.numeric(as
.vector(cluster_performance_eval_df[cluster_performance_eval_df$cluster==1,c("yes","no")])))

recall_yes_clust1 = (cluster_performance_eval_df$yes[cluster_performance_eval_df$cluster==1])/sum(cluster_performa
nce_eval_df$yes)

F_yes_clust1 = (2*precision_yes_clust1*recall_yes_clust1)/(precision_yes_clust1+recall_yes_clust1)

paste("Precision of class Yes for Rain Tomorrow in cluster 1 is:",precision_yes_clust1)
```

```
## [1] "Precision of class Yes for Rain Tomorrow in cluster 1 is: 0.545475759873986"
```

```
paste("Recall of class Yes for Rain Tomorrow in cluster 1 is:",recall_yes_clust1)
```

```
## [1] "Recall of class Yes for Rain Tomorrow in cluster 1 is: 0.777094414893617"
```

```
paste("F Measure of class Yes for Rain Tomorrow in cluster 1 is:",F_yes_clust1)
```

```
## [1] "F Measure of class Yes for Rain Tomorrow in cluster 1 is: 0.641003667774997"
```

```
precision_yes_clust2 = (cluster_performance_eval_df$yes[cluster_performance_eval_df$cluster==2])/sum(as.numeric(as
.vector(cluster_performance_eval_df[cluster_performance_eval_df$cluster==2,c("yes","no")])))

recall_yes_clust2 = (cluster_performance_eval_df$yes[cluster_performance_eval_df$cluster==2])/sum(cluster_performa
nce_eval_df$yes)

F_yes_clust2 = (2*precision_yes_clust2*recall_yes_clust2)/(precision_yes_clust2+recall_yes_clust2)

paste("Precision of class Yes for Rain Tomorrow in cluster 2 is:",precision_yes_clust2)
```

```
## [1] "Precision of class Yes for Rain Tomorrow in cluster 2 is: 0.337655797557598"
```

```
paste("Recall of class Yes for Rain Tomorrow in cluster 2 is:",recall_yes_clust2)
```

```
## [1] "Recall of class Yes for Rain Tomorrow in cluster 2 is: 0.222905585106383"
```

```
paste("F Measure of class Yes for Rain Tomorrow in cluster 2 is:",F_yes_clust2)
```

```
## [1] "F Measure of class Yes for Rain Tomorrow in cluster 2 is: 0.268535669586984"
```

clutser 1 has higher F measure for class 'Yes'.

Since cluster 1 has higher F Measure, Cluster 1 is being used to classify Rain Tomorrow as 'Yes' and cluster 2 is used to classify the Rain Tomorrow as 'No'.

```
paste("The average Rainfall for cluster 1 is: ",mean(hac_eval$Rainfall[hac_eval$clusterassigned ==1]))
```

```
## [1] "The average Rainfall for cluster 1 is:  0.0746342153168288"
```

```
paste("The average MinTemp for cluster 1 is: ",mean(hac_eval$MinTemp[hac_eval$clusterassigned ==1]))
```

```
## [1] "The average MinTemp for cluster 1 is:  0.446901525589635"
```

```
paste("The average MaxTemp for cluster 1 is: ",mean(hac_eval$MaxTemp[hac_eval$clusterassigned ==1]))
```

```
## [1] "The average MaxTemp for cluster 1 is:  0.419520722588084"
```

```
paste("The average Humidity for cluster 1 is: ",mean(hac_eval$Humidity[hac_eval$clusterassigned ==1]))
```

```
## [1] "The average Humidity for cluster 1 is:  0.63855608768391"
```

```
paste("The average Pressure for cluster 1 is: ",mean(hac_eval$Pressure[hac_eval$clusterassigned ==1]))
```

```
## [1] "The average Pressure for cluster 1 is:  0.538563414631357"
```

```
paste("The average Temp for cluster 1 is: ",mean(hac_eval$Temp[hac_eval$clusterassigned ==1]))
```

```
## [1] "The average Temp for cluster 1 is:  0.416243014753656"
```

```
paste("The average WindGustSpeed for cluster 1 is: ",mean(hac_eval$WindGustSpeed[hac_eval$clusterassigned ==1]))
```

```
## [1] "The average WindGustSpeed for cluster 1 is:  0.454970459637632"
```

```
paste("The average Rainfall for cluster 2 is: ",mean(hac_eval$Rainfall[hac_eval$clusterassigned ==2]))
```

```
## [1] "The average Rainfall for cluster 2 is:  0.0214144956544191"
```

```
paste("The average MinTemp for cluster 2 is: ",mean(hac_eval$MinTemp[hac_eval$clusterassigned ==2]))
```

```
## [1] "The average MinTemp for cluster 2 is:  0.621261747204771"
```

```
paste("The average MaxTemp for cluster 2 is: ",mean(hac_eval$MaxTemp[hac_eval$clusterassigned ==2]))
```

```
## [1] "The average MaxTemp for cluster 2 is:  0.67618102509794"
```

```
paste("The average Humidity for cluster 2 is: ",mean(hac_eval$Humidity[hac_eval$clusterassigned ==2]))
```

```
## [1] "The average Humidity for cluster 2 is:  0.416176113393789"
```

```
paste("The average Pressure for cluster 2 is: ",mean(hac_eval$Pressure[hac_eval$clusterassigned ==2]))
```

```
## [1] "The average Pressure for cluster 2 is:  0.426660976615247"
```

```
paste("The average Temp for cluster 2 is: ",mean(hac_eval$Temp[hac_eval$clusterassigned ==2]))
```

```
## [1] "The average Temp for cluster 2 is:  0.678938291069592"
```

```
paste("The average WindGustSpeed for cluster 2 is: ",mean(hac_eval$WindGustSpeed[hac_eval$clusterassigned ==2]))
```

```
## [1] "The average WindGustSpeed for cluster 2 is:  0.462426545896268"
```

Classifying cluster 1 as 'Yes' & cluster 2 as 'No' for Rain Tomorrow

```
hac_eval$RainTomorrowYesPred <- 1
hac_eval$RainTomorrowYesPred[hac_eval$clusterassigned==2] <- 0
```

Accuracy measurement using confusion Matrix

```
confusion_matrix_kmeans <- data.frame(table(hac_eval$RainTomorrowYes,hac_eval$RainTomorrowYesPred))
colnames(confusion_matrix_kmeans) <- c("Actual class","Predicted Class","Count")

Accuracy_kmeans <- sum(confusion_matrix_kmeans$Count[confusion_matrix_kmeans$`Actual class`==confusion_matrix_kmeans$`Predicted Class`])/sum(confusion_matrix_kmeans$Count)
Precision_kmeans <- confusion_matrix_kmeans$Count[confusion_matrix_kmeans$`Actual class`==1 & confusion_matrix_kmeans$`Predicted Class`==1]/sum(confusion_matrix_kmeans$Count[confusion_matrix_kmeans$`Predicted Class`==1])
Recall_kmeans <- confusion_matrix_kmeans$Count[confusion_matrix_kmeans$`Actual class`==1 & confusion_matrix_kmeans$`Predicted Class`==1]/sum(confusion_matrix_kmeans$Count[confusion_matrix_kmeans$`Actual class`==1])
F1_score_kmeans <- (2*Precision_kmeans*Recall_kmeans)/(Precision_kmeans+Recall_kmeans)

paste("Accuracy of HAC Algorithm in classifying Rain Tomorrow is :",Accuracy_kmeans)
```

```
## [1] "Accuracy of HAC Algorithm in classifying Rain Tomorrow is : 0.582482857598469"
```

```
paste("Precision of HAC Algorithm in classifying Rain Tomorrow is :",Precision_kmeans)
```

```
## [1] "Precision of HAC Algorithm in classifying Rain Tomorrow is : 0.545475759873986"
```

```
paste("Recall of HAC Algorithm in classifying Rain Tomorrow is :",Recall_kmeans)
```

```
## [1] "Recall of HAC Algorithm in classifying Rain Tomorrow is : 0.777094414893617"
```

```
paste("F1 Score of HAC Algorithm in classifying Rain Tomorrow is :",F1_score_kmeans)
```

```
## [1] "F1 Score of HAC Algorithm in classifying Rain Tomorrow is : 0.641003667774997"
```

```
Eval_df <- c("Accuracy","Precision","Recall","F1 Score")
KMeans_including_dummies <- c(0.656148464129806,0.734845360824742,0.443946188340807,0.553502096598851)
HAC_without_dummies <- c(0.582482857598469,0.545475759873986,0.777094414893617,0.641003667774997)
data.frame(Eval_df,KMeans_including_dummies,HAC_without_dummies)
```

```
##     Eval_df KMeans_including_dummies HAC_without_dummies
## 1  Accuracy                0.6561485           0.5824829
## 2 Precision                0.7348454           0.5454758
## 3    Recall                0.4439462           0.7770944
## 4  F1 Score                0.5535021           0.6410037
```

We can observe that KMeans including dummy variables has the highest accuracy in classifying but HAC has the highest Recall & F1 score.

DECISION TREE INDUCTION

Decision Tree is robust to outliers

```
decision_tree <- myData1
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(rpart)
```

Generating Training and Test datasets for decision tree with 60% & 40% splits respectively

```
train_indices <- sample(seq_len(nrow(decision_tree)),size = 0.7*(nrow(decision_tree)))
 decision_tree_trainData <- decision_tree[train_indices,]      ### TRAINING DATA
  rownames(decision_tree_trainData) <- NULL
 decision_tree_testData <- decision_tree[-train_indices,]      ### TESTING DATA
 rownames(decision_tree_testData) <- NULL
```

Building a Decision Tree model

```
decision_tree_model <- train(RainTomorrow ~ ., data = decision_tree_trainData, method = "rpart",
                      metric = "Accuracy",
                      tuneLength = 6)
```

6 models were tuned and the model with lowest cp produced the highest accuracy, which was used as the final resulting model

```
print(decision_tree_model)
```

```
## CART
##
## 36371 samples
##    12 predictor
##     2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 36371, 36371, 36371, 36371, 36371, 36371, ...
## Resampling results across tuning parameters:
##
##   cp           Accuracy   Kappa
##   0.002743868  0.7569438  0.5131637
##   0.003359839  0.7556296  0.5104862
##   0.004423788  0.7496915  0.4986694
##   0.006131706  0.7448500  0.4889925
##   0.021166984  0.7270682  0.4530808
##   0.434651137  0.6450698  0.2831468
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.002743868.
```

Making Classifications on Test Data using the Decision Tree Model built above using the Training data

```
decision_tree_predict <- predict(decision_tree_model, newdata = decision_tree_testData, na.action = na.omit, type
= "raw")
```

Accuracy using Confusion Matrix

```
decision_tree_testData$predicted_rain_tomorrow <- decision_tree_predict
conf_matrix <- data.frame(table(decision_tree_testData$RainTomorrow,decision_tree_testData$predicted_rain_tomorrow
))
colnames(conf_matrix)<- c('Actual class','Predicted Class','Count')

Accuracy_DT <- sum(conf_matrix$Count[conf_matrix$`Actual class`==conf_matrix$`Predicted Class`])/sum(conf_matrix$C
ount)
Precision_DT <- conf_matrix$Count[conf_matrix$`Actual class`=='Yes' & conf_matrix$`Predicted Class`=='Yes']/sum(co
nf_matrix$Count[conf_matrix$`Predicted Class`=='Yes'])
Recall_DT <- conf_matrix$Count[conf_matrix$`Actual class`=='Yes' & conf_matrix$`Predicted Class`=='Yes']/sum(conf_
matrix$Count[conf_matrix$`Actual class`=='Yes'])
F1_score_DT <- (2*Precision_DT*Recall_DT)/(Precision_DT+Recall_DT)

paste("Accuracy of DT Algorithm in classifying Rain Tomorrow is :",Accuracy_DT)
```

```
## [1] "Accuracy of DT Algorithm in classifying Rain Tomorrow is : 0.756030279702335"
```

```
paste("Precision of DT Algorithm in classifying Rain Tomorrow is :",Precision_DT)
```

```
## [1] "Precision of DT Algorithm in classifying Rain Tomorrow is : 0.760508176780646"
```

```
paste("Recall of DT Algorithm in classifying Rain Tomorrow is :",Recall_DT)
```

```
## [1] "Recall of DT Algorithm in classifying Rain Tomorrow is : 0.734787150692087"
```

```
paste("F1 Score of DT Algorithm in classifying Rain Tomorrow is :",F1_score_DT)
```

```
## [1] "F1 Score of DT Algorithm in classifying Rain Tomorrow is : 0.747426446171216"
```

Hyperparameters are minbucket, minsplit and maxdepth alongwith CP for tuning the model

minsplit, maxdepth and minbucket should be adjusted so as to pre prune the model to produce a least complex model(to reduce overfitting and variance) with maximum accuracy (to minimize bias)

Not running while knitting to HTML

`{r} #gridsearch <- list(minsplit = c(20,15,10,5), #  maxdepth = c(30,25,20,15), #  minbucket = c(6,5,4,3)) %>%  #`

Performing Grid Search with all combinations of different values of minsplit, maxdepth and minbucket created above and measuring the accuracy for each combination #```{r} #accuracy_values <- c() #for (i in seq(1,nrow(gridsearch))) {

dt_model_weather2 <- train(RainTomorrow ~ ., data = decision_tree_trainData,method = “rpart”,metric = “Accuracy”,

tuneLength = 8,control = rpart.control(minsplit =as.numeric(gridsearch[i,“minsplit”]), minbucket = #as.numeric(gridsearch[i,“minbucket”]), maxdepth = as.numeric(gridsearch[i,“maxdepth”])))

accuracy_values<- c(accuracy_values,max(dt_model_weather2\(results\)Accuracy))

#} #```

Displaying the Accuracy of 64 Decision Tree Models for different combinations of the hyperparameters

`{r} #gridsearch$accuracy <- accuracy_values #gridsearch[order(-gridsearch$accuracy),] #` *Since Accuracy is a good measure of model performance We can see that the first row below with minsplit = 15, maxdepth = 30, minbucket = 5 has the best accuracy and hence can be considered the best model.*

Based on Occam's Razor principle, we need to chose a simpler model, amongst the models that produce the same accuracy

Creating a decision tree model with maxdepth 15 for minsplit 15 & minbucket 5 so as avoid overfitting and reduce estimate variance

```
 decision_tree_model2 <- train(RainTomorrow ~ ., data = decision_tree_trainData,method = "rpart",metric = "Accura
cy",
                      tuneLength = 8,control = rpart.control(minsplit = 15, minbucket = 5, maxdepth = 15))
print(decision_tree_model2)
```

```
## CART
##
## 36371 samples
##    12 predictor
##     2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 36371, 36371, 36371, 36371, 36371, 36371, ...
## Resampling results across tuning parameters:
##
##   cp           Accuracy   Kappa
##   0.001455930  0.7625520  0.5243974
##   0.002715870  0.7572815  0.5137688
##   0.002743868  0.7572338  0.5136537
##   0.003359839  0.7550583  0.5092369
##   0.004423788  0.7503758  0.5000436
##   0.006131706  0.7452512  0.4897598
##   0.021166984  0.7288841  0.4566084
##   0.434651137  0.6715408  0.3378044
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.00145593.
```

The Decision Tree Model is producing a Training accuracy of 76.08%

Using the best performing model

```
decision_tree_predict <- predict(decision_tree_model2, newdata = decision_tree_testData, na.action = na.omit, type
= "raw")
```

calculating Accuracy Precision and Recall on the classification done on the Test Data by using the Confusion Matrix

```
decision_tree_testData$predicted_rain_tomorrow <- decision_tree_predict
conf_matrix <- data.frame(table(decision_tree_testData$RainTomorrow,decision_tree_testData$predicted_rain_tomorrow
))
colnames(conf_matrix)<- c('Actual class','Predicted Class','Count')

Accuracy_DT <- sum(conf_matrix$Count[conf_matrix$`Actual class`==conf_matrix$`Predicted Class`])/sum(conf_matrix$C
ount)
Precision_DT <- conf_matrix$Count[conf_matrix$`Actual class`=='Yes' & conf_matrix$`Predicted Class`=='Yes']/sum(co
nf_matrix$Count[conf_matrix$`Predicted Class`=='Yes'])
Recall_DT <- conf_matrix$Count[conf_matrix$`Actual class`=='Yes' & conf_matrix$`Predicted Class`=='Yes']/sum(conf_
matrix$Count[conf_matrix$`Actual class`=='Yes'])
F1_score_DT <- (2*Precision_DT*Recall_DT)/(Precision_DT+Recall_DT)

paste("Accuracy of DT Algorithm in classifying Rain Tomorrow is :",Accuracy_DT)
```

```
## [1] "Accuracy of DT Algorithm in classifying Rain Tomorrow is : 0.76058506543495"
```

```
paste("Precision of DT Algorithm in classifying Rain Tomorrow is :",Precision_DT)
```

```
## [1] "Precision of DT Algorithm in classifying Rain Tomorrow is : 0.761245674740484"
```

```
paste("Recall of DT Algorithm in classifying Rain Tomorrow is :",Recall_DT)
```

```
## [1] "Recall of DT Algorithm in classifying Rain Tomorrow is : 0.746931313658919"
```

```
paste("F1 Score of DT Algorithm in classifying Rain Tomorrow is :",F1_score_DT)
```

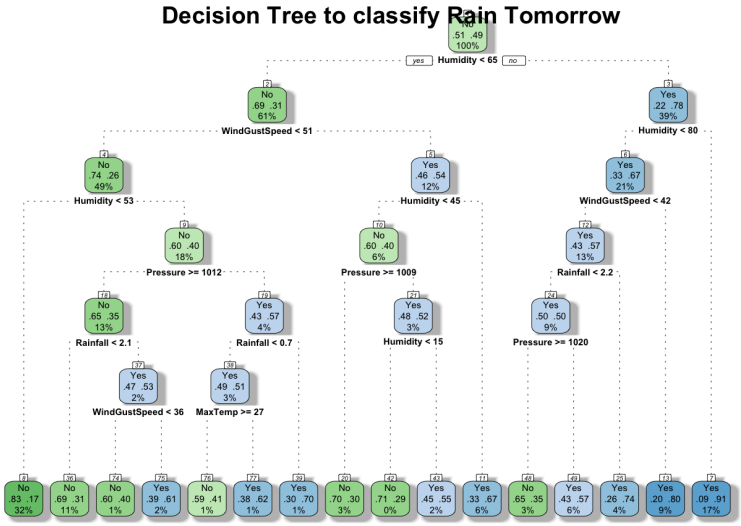
```
## [1] "F1 Score of DT Algorithm in classifying Rain Tomorrow is : 0.754020564197205"
```

```
library(rattle)
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.3.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

Decision Tree Plot

```
fancyRpartPlot(decision_tree_model2$finalModel, main = "Decision Tree to classify Rain Tomorrow")
```



ROC & AUROC for Decision Tree Induction

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

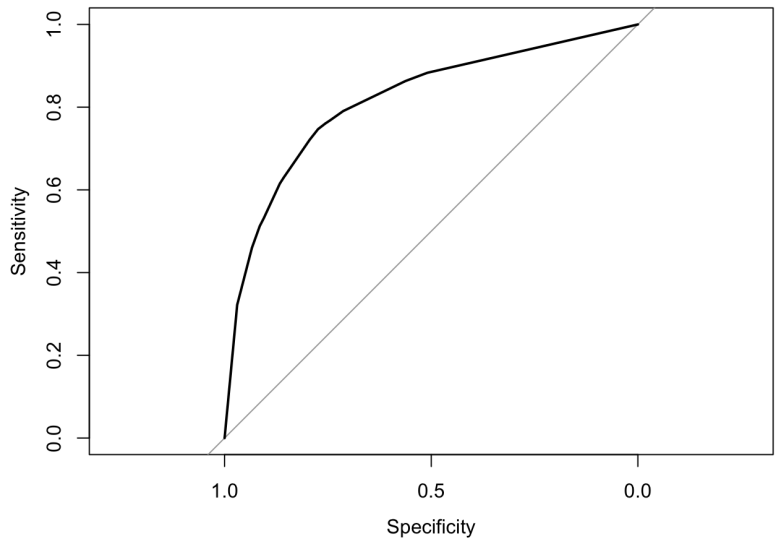
Generated ROC curve and calculated Area Under Curve metric for the identified best performing decision tree model

```
decision_tree_prob <- predict(decision_tree_model2, newdata = decision_tree_testData, na.action = na.omit, type = "prob")
roc_curve <- roc(decision_tree_testData$RainTomorrow,decision_tree_prob$Yes)
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
plot(roc_curve)
```




```
paste("Area Under the ROC curve is :",auc(roc_curve))
```

```
## [1] "Area Under the ROC curve is : 0.815780471064884"
```

Obtained an area of 0.8154 under the ROC curve

TEST DATA

Reading the Testing data csv and storing it into a dataframe

```
myData_test <- read.csv("Weather Forecast Testing.csv")
View(myData_test)
```

Step1: Data Cleaning

Removing NA's

Discarding these columns as they consist dirty data i.e missing values >50%

```
myData_test$Evaporation <- NULL
myData_test$Sunshine <- NULL
myData_test$Cloud <- NULL
```

mode function

```
  getmode <- function(m) {
  um <- na.omit(unique(m) )
 tab <- tabulate(match(m, um)); um[tab == max(tab) ]
}
```

```
      MinTemp_location <- myData_test %>%
      group_by(Location)%>%
      summarise(median(MinTemp, na.rm=TRUE))

    placeNA <- which(is.na(myData_test$MinTemp))
    for (i in placeNA) {
      myData_test$MinTemp[i] <- as.numeric(MinTemp_location[MinTemp_location$Location==myData_test[i,"Location"],2
])
    }
```

```
MaxTemp_location <- myData_test %>%
      group_by(Location)%>%
      summarise(median(MaxTemp, na.rm=TRUE))

    placeNA  <- which(is.na(myData_test$MaxTemp))
    for (i in placeNA ) {
      myData_test$MaxTemp[i] <- as.numeric(MaxTemp_location[MaxTemp_location$Location==myData_test[i,"Location"],2
])
    }
```

```
Rainfall_location <- myData_test %>%
      group_by(Location)%>%
      summarise(median(Rainfall, na.rm=TRUE))

    placeNA  <- which(is.na(myData_test$Rainfall))
    for (i in placeNA ) {
      myData_test$Rainfall[i] <- as.numeric(Rainfall_location[Rainfall_location$Location==myData_test[i,"Location"
],2])
    }
```

```
    placeNA  <- which((myData_test$WindGustDir == ''))
    for (i in placeNA ) {
      myData_test$WindGustDir[i] <- getmode(myData_test$WindGustDir)
    }
```

```
myData_test$WindGustSpeed[which(is.na(myData_test$WindGustSpeed))] <- median(myData_test$WindGustSpeed, na.rm = TR
UE)
```

```
    placeNA  <- which(myData_test$WindDir == "")
   for (i in placeNA ) {
      myData_test$WindDir[i] <- getmode(myData_test$WindDir)
    }
```

```
windspeed_location <- myData_test %>%
      group_by(Location)%>%
      summarise(median(WindSpeed, na.rm=TRUE))

    placeNA  <- which(is.na(myData_test$WindSpeed))
    for (i in placeNA ) {
      myData_test$WindSpeed[i] <- as.numeric(windspeed_location[windspeed_location$Location==myData_test[i,"Locati
on"],2])
    }
```

```
humidity_location <- myData_test %>%
      group_by(Location)%>%
      summarise(median(Humidity, na.rm=TRUE))

    placeNA  <- which(is.na(myData_test$Humidity))
    for (i in placeNA ) {
      myData_test$Humidity[i] <- as.numeric(humidity_location[humidity_location$Location==myData_test[i,"Location"
],2])
    }
```

```
placeNA  <- which(is.na(myData_test$Pressure))
    for (i in placeNA ) {
      myData_test$Pressure[i] <- median(myData_test$Pressure, na.rm = TRUE)
    }
```

```
temp_location <- myData_test %>%
      group_by(Location)%>%
      summarise(median(Temp, na.rm=TRUE))

    placeNA  <- which(is.na(myData_test$Temp))
    for (i in placeNA ) {
      myData_test$Temp[i] <- as.numeric(temp_location[temp_location$Location==myData_test[i,"Location"],2])
    }
```

Replace Na's/blanks in Rain Today with No

```
places<- which(myData_test$RainToday == "")
```

```
myData_test$RainToday[places] <- as.factor("No")
```

View structure and summary of the final treated table **There are no missing values in any of the columns after treating all the columns**

```
str(myData_test)
```

```
## 'data.frame':    12994 obs. of  13 variables:
##  $ ID           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Location     : Factor w/ 49 levels "Adelaide","Albany",..: 32 34 16 49 40 29 45 45 22 3 ...
##  $ MinTemp      : num  15.3 7.1 3.6 17.6 11.4 4.4 12.4 8.3 22.8 18 ...
##  $ MaxTemp      : num  21.5 11 16.6 37.4 25.6 13 22.4 14.1 38.6 26.9 ...
##  $ Rainfall     : num  4.4 38.8 0.2 0 0 0.4 0.6 1 0 0.2 ...
##  $ WindGustDir  : Factor w/ 17 levels "","E","ENE","ESE",..: 9 12 9 8 4 9 14 14 2 9 ...
##  $ WindGustSpeed: num  70 48 37 37 31 39 20 56 54 46 ...
##  $ WindDir      : Factor w/ 17 levels "","E","ENE","ESE",..: 15 14 5 9 12 9 12 14 8 15 ...
##  $ WindSpeed    : num  22 19 15 19 13 17 4 22 11 22 ...
##  $ Humidity     : num  69 46 48 11 52 69 57 58 31 31 ...
##  $ Pressure     : num  998 1014 1017 1010 1016 ...
##  $ Temp         : num  19.8 10.7 15.4 34.6 24.3 11.3 21.6 13.6 37.1 25.9 ...
##  $ RainToday    : Factor w/ 3 levels "","No","Yes": 3 3 2 2 2 2 2 2 2 2 ...
```

```
colSums(is.na(myData_test))
```

```
##            ID      Location       MinTemp       MaxTemp      Rainfall
##             0             0             0             0             0
##   WindGustDir WindGustSpeed       WindDir     WindSpeed      Humidity
##             0             0             0             0             0
##      Pressure          Temp     RainToday
##             0             0             0
```

##Creating Dummies for categorical variables

```
myData_test_dummies <- fastDummies::dummy_cols(myData_test,  select_columns = c('Location','WindGustDir','WindDir'
,'RainToday'))
```

```
myData_test_dummies$WindGustDir_ <- NULL
myData_test_dummies$WindDir_ <- NULL
myData_test_dummies$RainToday_ <- NULL
```

```
myData_test_cluster <- myData_test_dummies
```

Remove the Categorical non dummy columns to perform clustering on Test Data and also remove the target variable Rain Tomorrow Yes

```
myData_test_cluster$Location <- NULL
myData_test_cluster$WindGustDir <- NULL
myData_test_cluster$WindDir <- NULL
myData_test_cluster$RainToday <- NULL
myData_test_cluster$RainTomorrow <- NULL
myData_test_cluster$RainToday_No <- NULL
myData_test_cluster$ID<-NULL
```

**

#Normalising

```
normalize <- function(x) {
return ((x - min(x)) / (max(x) - min(x)))
}
```

Normalizing all the numeric columns

```
myData_test_cluster$MinTemp <- normalize(myData_test_cluster$MinTemp)
myData_test_cluster$MaxTemp <- normalize(myData_test_cluster$MaxTemp)
myData_test_cluster$Rainfall <- normalize(myData_test_cluster$Rainfall)
myData_test_cluster$WindGustSpeed <- normalize(myData_test_cluster$WindGustSpeed)
myData_test_cluster$WindSpeed <- normalize(myData_test_cluster$WindSpeed)
myData_test_cluster$Humidity <- normalize(myData_test_cluster$Humidity)
myData_test_cluster$Pressure <- normalize(myData_test_cluster$Pressure)
myData_test_cluster$Temp <- normalize(myData_test_cluster$Temp)
```

```
str(myData_test_cluster)
```

```
## 'data.frame':    12994 obs. of  90 variables:
##  $ MinTemp                 : num  0.602 0.376 0.279 0.666 0.494 ...
##  $ MaxTemp                 : num  0.498 0.291 0.401 0.812 0.579 ...
##  $ Rainfall                : num  0.01197 0.10555 0.000544 0 0 ...
##  $ WindGustSpeed           : num  0.492 0.32 0.234 0.234 0.188 ...
##  $ WindSpeed               : num  0.253 0.218 0.172 0.218 0.149 ...
##  $ Humidity                : num  0.687 0.455 0.475 0.101 0.515 ...
##  $ Pressure                : num  0.305 0.593 0.639 0.53 0.635 ...
##  $ Temp                    : num  0.475 0.294 0.388 0.769 0.565 ...
##  $ Location_Adelaide       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Albany         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Albury         : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Location_AliceSprings   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_BadgerysCreek  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Ballarat       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Bendigo        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Brisbane       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Cairns         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Canberra       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Cobar          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_CoffsHarbour   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Dartmoor       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Darwin         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_GoldCoast      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Hobart         : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ Location_Katherine      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Launceston     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Melbourne      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_MelbourneAirport: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Mildura        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Moree          : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ Location_MountGambier   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_MountGinini    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Newcastle      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Nhil           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_NorahHead      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_NorfolkIsland  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Nuriootpa      : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Location_PearceRAAF     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Penrith        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Perth          : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Location_PerthAirport   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Portland       : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Location_Richmond       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Sale           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_SalmonGums     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Sydney         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_SydneyAirport  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Townsville     : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Location_Tuggeranong    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Uluru          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_WaggaWagga     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Walpole        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Watsonia       : int  0 0 0 0 0 0 1 1 0 0 ...
##  $ Location_Williamtown    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Witchcliffe    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Wollongong     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Location_Woomera        : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ WindGustDir_E           : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ WindGustDir_ENE         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_ESE         : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ WindGustDir_N           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_NE          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_NNE         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_NNW         : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ WindGustDir_NW          : int  1 0 1 0 0 1 0 0 0 1 ...
##  $ WindGustDir_S           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_SE          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_SSE         : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_SSW         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_SW          : int  0 0 0 0 0 0 1 1 0 0 ...
##  $ WindGustDir_W           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_WNW         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindGustDir_WSW         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_E               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_ENE             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_ESE             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_N               : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ WindDir_NE              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_NNE             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_NNW             : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ WindDir_NW              : int  0 0 0 1 0 1 0 0 0 0 ...
##  $ WindDir_S               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_SE              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_SSE             : int  0 0 0 0 1 0 1 0 0 0 ...
##  $ WindDir_SSW             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_SW              : int  0 1 0 0 0 0 0 1 0 0 ...
##  $ WindDir_W               : int  1 0 0 0 0 0 0 0 0 1 ...
##  $ WindDir_WNW             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WindDir_WSW             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ RainToday_Yes           : int  1 1 0 0 0 0 0 0 0 0 ...
```

```
summary(myData_test_cluster)
```

```
##     MinTemp          MaxTemp          Rainfall        WindGustSpeed
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000000   Min.   :0.0000
##  1st Qu.:0.4006   1st Qu.:0.4150   1st Qu.:0.000000   1st Qu.:0.1875
##  Median :0.5166   Median :0.5020   Median :0.000000   Median :0.2500
##  Mean   :0.5249   Mean   :0.5176   Mean   :0.009810   Mean   :0.2719
##  3rd Qu.:0.6519   3rd Qu.:0.6146   3rd Qu.:0.005985   3rd Qu.:0.3359
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.000000   Max.   :1.0000
##    WindSpeed         Humidity         Pressure           Temp
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.1494   1st Qu.:0.4167   1st Qu.:0.5211   1st Qu.:0.3976
##  Median :0.2184   Median :0.5758   Median :0.5965   Median :0.4831
##  Mean   :0.2180   Mean   :0.5687   Mean   :0.5965   Mean   :0.4961
##  3rd Qu.:0.2759   3rd Qu.:0.7273   3rd Qu.:0.6719   3rd Qu.:0.5865
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  Location_Adelaide Location_Albany   Location_Albury
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.00000   Median :0.00000   Median :0.0000
##  Mean   :0.02209   Mean   :0.02455   Mean   :0.0207
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
##  Location_AliceSprings Location_BadgerysCreek Location_Ballarat
##  Min.   :0.00000       Min.   :0.00000        Min.   :0.00000
##  1st Qu.:0.00000       1st Qu.:0.00000        1st Qu.:0.00000
##  Median :0.00000       Median :0.00000        Median :0.00000
##  Mean   :0.01547       Mean   :0.01747        Mean   :0.02186
##  3rd Qu.:0.00000       3rd Qu.:0.00000        3rd Qu.:0.00000
##  Max.   :1.00000       Max.   :1.00000        Max.   :1.00000
##  Location_Bendigo  Location_Brisbane Location_Cairns  Location_Canberra
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.0000   Median :0.00000
##  Mean   :0.02039   Mean   :0.02263   Mean   :0.0234   Mean   :0.02293
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.0000   Max.   :1.00000
##  Location_Cobar    Location_CoffsHarbour Location_Dartmoor
##  Min.   :0.00000   Min.   :0.00000       Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000       1st Qu.:0.00000
##  Median :0.00000   Median :0.00000       Median :0.00000
##  Mean   :0.01701   Mean   :0.02316       Mean   :0.02478
##  3rd Qu.:0.00000   3rd Qu.:0.00000       3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000       Max.   :1.00000
##  Location_Darwin   Location_GoldCoast Location_Hobart   Location_Katherine
##  Min.   :0.00000   Min.   :0.00000    Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.00000   1st Qu.:0.00000    1st Qu.:0.00000   1st Qu.:0.000000
##  Median :0.00000   Median :0.00000    Median :0.00000   Median :0.000000
##  Mean   :0.02386   Mean   :0.02147    Mean   :0.02586   Mean   :0.009389
##  3rd Qu.:0.00000   3rd Qu.:0.00000    3rd Qu.:0.00000   3rd Qu.:0.000000
##  Max.   :1.00000   Max.   :1.00000    Max.   :1.00000   Max.   :1.000000
##  Location_Launceston Location_Melbourne Location_MelbourneAirport
##  Min.   :0.00000     Min.   :0.00000    Min.   :0.00000
##  1st Qu.:0.00000     1st Qu.:0.00000    1st Qu.:0.00000
##  Median :0.00000     Median :0.00000    Median :0.00000
##  Mean   :0.02001     Mean   :0.01801    Mean   :0.01962
##  3rd Qu.:0.00000     3rd Qu.:0.00000    3rd Qu.:0.00000
##  Max.   :1.00000     Max.   :1.00000    Max.   :1.00000
##  Location_Mildura  Location_Moree    Location_MountGambier
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.01862   Mean   :0.01862   Mean   :0.02232
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##  Location_MountGinini Location_Newcastle Location_Nhil
##  Min.   :0.00000      Min.   :0.00000    Min.   :0.000000
##  1st Qu.:0.00000      1st Qu.:0.00000    1st Qu.:0.000000
##  Median :0.00000      Median :0.00000    Median :0.000000
##  Mean   :0.02347      Mean   :0.02224    Mean   :0.008158
##  3rd Qu.:0.00000      3rd Qu.:0.00000    3rd Qu.:0.000000
##  Max.   :1.00000      Max.   :1.00000    Max.   :1.000000
##  Location_NorahHead Location_NorfolkIsland Location_Nuriootpa
##  Min.   :0.00000    Min.   :0.00000        Min.   :0.00000
##  1st Qu.:0.00000    1st Qu.:0.00000        1st Qu.:0.00000
##  Median :0.00000    Median :0.00000        Median :0.00000
##  Mean   :0.02086    Mean   :0.02486        Mean   :0.02032
##  3rd Qu.:0.00000    3rd Qu.:0.00000        3rd Qu.:0.00000
##  Max.   :1.00000    Max.   :1.00000        Max.   :1.00000
##  Location_PearceRAAF Location_Penrith  Location_Perth
##  Min.   :0.00000     Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000     1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000     Median :0.00000   Median :0.00000
##  Mean   :0.01639     Mean   :0.02186   Mean   :0.02178
##  3rd Qu.:0.00000     3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000     Max.   :1.00000   Max.   :1.00000
##  Location_PerthAirport Location_Portland Location_Richmond
##  Min.   :0.00000       Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000       1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000       Median :0.00000   Median :0.00000
##  Mean   :0.02101       Mean   :0.02747   Mean   :0.02039
##  3rd Qu.:0.00000       3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000       Max.   :1.00000   Max.   :1.00000
```

```
##  Location_Sale     Location_SalmonGums Location_Sydney
##  Min.   :0.00000   Min.   :0.00000     Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.00000     1st Qu.:0.0000
##  Median :0.00000   Median :0.00000     Median :0.0000
##  Mean   :0.02001   Mean   :0.01732     Mean   :0.0247
##  3rd Qu.:0.00000   3rd Qu.:0.00000     3rd Qu.:0.0000
##  Max.   :1.00000   Max.   :1.00000     Max.   :1.0000
##  Location_SydneyAirport Location_Townsville Location_Tuggeranong
##  Min.   :0.00000        Min.   :0.00000     Min.   :0.00000
##  1st Qu.:0.00000        1st Qu.:0.00000     1st Qu.:0.00000
##  Median :0.00000        Median :0.00000     Median :0.00000
##  Mean   :0.02139        Mean   :0.01955     Mean   :0.02147
##  3rd Qu.:0.00000        3rd Qu.:0.00000     3rd Qu.:0.00000
##  Max.   :1.00000        Max.   :1.00000     Max.   :1.00000
##  Location_Uluru     Location_WaggaWagga Location_Walpole
##  Min.   :0.000000   Min.   :0.0000      Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.0000      1st Qu.:0.00000
##  Median :0.000000   Median :0.0000      Median :0.00000
##  Mean   :0.009004   Mean   :0.0197      Mean   :0.02116
##  3rd Qu.:0.000000   3rd Qu.:0.0000      3rd Qu.:0.00000
##  Max.   :1.000000   Max.   :1.0000      Max.   :1.00000
##  Location_Watsonia Location_Williamtown Location_Witchcliffe
##  Min.   :0.00000   Min.   :0.00000      Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000      1st Qu.:0.00000
##  Median :0.00000   Median :0.00000      Median :0.00000
##  Mean   :0.02163   Mean   :0.02024      Mean   :0.02316
##  3rd Qu.:0.00000   3rd Qu.:0.00000      3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000      Max.   :1.00000
##  Location_Wollongong Location_Woomera WindGustDir_E     WindGustDir_ENE
##  Min.   :0.00000     Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000     1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000     Median :0.0000   Median :0.00000   Median :0.00000
##  Mean   :0.02093     Mean   :0.0167   Mean   :0.05518   Mean   :0.04933
##  3rd Qu.:0.00000     3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000     Max.   :1.0000   Max.   :1.00000   Max.   :1.00000
##  WindGustDir_ESE  WindGustDir_N     WindGustDir_NE    WindGustDir_NNE
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.0000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.0434   Mean   :0.06965   Mean   :0.04879   Mean   :0.04387
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##  WindGustDir_NNW   WindGustDir_NW    WindGustDir_S     WindGustDir_SE
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.05264   Mean   :0.06341   Mean   :0.06403   Mean   :0.06465
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##  WindGustDir_SSE   WindGustDir_SSW   WindGustDir_SW    WindGustDir_W
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.05826   Mean   :0.05641   Mean   :0.06057   Mean   :0.07003
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##  WindGustDir_WNW   WindGustDir_WSW     WindDir_E         WindDir_ENE
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.06264   Mean   :0.06565   Mean   :0.05218   Mean   :0.04956
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##   WindDir_ESE         WindDir_N         WindDir_NE        WindDir_NNE
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.05402   Mean   :0.06772   Mean   :0.05533   Mean   :0.04587
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##   WindDir_NNW         WindDir_NW         WindDir_S          WindDir_SE
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.0000
##  Mean   :0.05833   Mean   :0.06834   Mean   :0.06264   Mean   :0.0725
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
##   WindDir_SSE        WindDir_SSW        WindDir_SW         WindDir_W
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.000
##  Mean   :0.06157   Mean   :0.05264   Mean   :0.06249   Mean   :0.103
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.000
##   WindDir_WNW        WindDir_WSW      RainToday_Yes
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.00000   Median :0.00000   Median :0.0000
##  Mean   :0.06611   Mean   :0.06772   Mean   :0.3024
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
```

#Kmeans **Excluding Dummies**

```
myData_test_cluster2 <- myData_test_cluster[,-9:-90]
str(myData_test_cluster2)
```

```
## 'data.frame':    12994 obs. of  8 variables:
##  $ MinTemp      : num  0.602 0.376 0.279 0.666 0.494 ...
##  $ MaxTemp      : num  0.498 0.291 0.401 0.812 0.579 ...
##  $ Rainfall     : num  0.01197 0.10555 0.000544 0 0 ...
##  $ WindGustSpeed: num  0.492 0.32 0.234 0.234 0.188 ...
##  $ WindSpeed    : num  0.253 0.218 0.172 0.218 0.149 ...
##  $ Humidity     : num  0.687 0.455 0.475 0.101 0.515 ...
##  $ Pressure     : num  0.305 0.593 0.639 0.53 0.635 ...
##  $ Temp         : num  0.475 0.294 0.388 0.769 0.565 ...
```

```
summary(myData_test_cluster2)
```

```
##     MinTemp          MaxTemp          Rainfall        WindGustSpeed
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000000   Min.   :0.0000
##  1st Qu.:0.4006   1st Qu.:0.4150   1st Qu.:0.000000   1st Qu.:0.1875
##  Median :0.5166   Median :0.5020   Median :0.000000   Median :0.2500
##  Mean   :0.5249   Mean   :0.5176   Mean   :0.009810   Mean   :0.2719
##  3rd Qu.:0.6519   3rd Qu.:0.6146   3rd Qu.:0.005985   3rd Qu.:0.3359
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.000000   Max.   :1.0000
##    WindSpeed         Humidity         Pressure           Temp
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.1494   1st Qu.:0.4167   1st Qu.:0.5211   1st Qu.:0.3976
##  Median :0.2184   Median :0.5758   Median :0.5965   Median :0.4831
##  Mean   :0.2180   Mean   :0.5687   Mean   :0.5965   Mean   :0.4961
##  3rd Qu.:0.2759   3rd Qu.:0.7273   3rd Qu.:0.6719   3rd Qu.:0.5865
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
```

```
km_test <- kmeans(myData_test_cluster2, centers = 2, nstart = 25, iter.max = 100)
str(km_test)
```

```
## List of 9
##  $ cluster     : int [1:12994] 2 2 2 1 1 2 2 2 1 1 ...
##  $ centers     : num [1:2, 1:8] 0.6438 0.44463 0.65001 0.42833 0.00706 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:8] "MinTemp" "MaxTemp" "Rainfall" "WindGustSpeed" ...
##  $ totss       : num 2037
##  $ withinss    : num [1:2] 573 850
##  $ tot.withinss: num 1423
##  $ betweenss   : num 614
##  $ size        : int [1:2] 5234 7760
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

Total SSE is 1423

Cluster plot

```
fviz_cluster(km_test, data = myData_test_cluster2,geom = "point",repel = FALSE,ggtheme = theme_minimal(),alpha=0.
02,shape = 19,ellipse.type = "norm") + ggtitle("TEST DATA CLUSTER")
```


```
length(km_test$cluster[km_test$cluster==1])
```

```
## [1] 5234
```

```
length(km_test$cluster[km_test$cluster==2])
```

```
## [1] 7760
```

```
myData_test_cluster2$clusterassigned <- km_test$cluster
```

summary statistics to support why cluster 1 should be used to classify Rain Tomorrow as 'Yes'

```
paste("The average Rainfall for cluster 1 is: ",mean(myData_test_cluster2$Rainfall[myData_test_cluster2$clusterass
igned ==1]))
```

```
## [1] "The average Rainfall for cluster 1 is:  0.00706323806466716"
```

```
paste("The average MinTemp for cluster 1 is: ",mean(myData_test_cluster2$MinTemp[myData_test_cluster2$clusterassig
ned ==1]))
```

```
## [1] "The average MinTemp for cluster 1 is:  0.643804744583334"
```

```
paste("The average MaxTemp for cluster 1 is: ",mean(myData_test_cluster2$MaxTemp[myData_test_cluster2$clusterassig
ned ==1]))
```

```
## [1] "The average MaxTemp for cluster 1 is:  0.650013744126651"
```

```
paste("The average Humidity for cluster 1 is: ",mean(myData_test_cluster2$Humidity[myData_test_cluster2$clusterass
igned ==1]))
```

```
## [1] "The average Humidity for cluster 1 is:  0.42931705283635"
```

```
paste("The average Pressure for cluster 1 is: ",mean(myData_test_cluster2$Pressure[myData_test_cluster2$clusterass
igned ==1]))
```

```
## [1] "The average Pressure for cluster 1 is:  0.556408503107213"
```

```
paste("The average Temp for cluster 1 is: ",mean(myData_test_cluster2$Temp[myData_test_cluster2$clusterassigned ==
1]))
```

```
## [1] "The average Temp for cluster 1 is:  0.627233351894745"
```

```
paste("The average WindGustSpeed for cluster 1 is: ",mean(myData_test_cluster2$WindGustSpeed[myData_test_cluster2$
clusterassigned ==1]))
```

```
## [1] "The average WindGustSpeed for cluster 1 is:  0.277642278849828"
```

```
paste("The average Rainfall for cluster 2 is: ",mean(myData_test_cluster2$Rainfall[myData_test_cluster2$clusterass
igned ==2]))
```

```
## [1] "The average Rainfall for cluster 2 is:  0.011663387758994"
```

```
paste("The average MinTemp for cluster 2 is: ",mean(myData_test_cluster2$MinTemp[myData_test_cluster2$clusterassig
ned ==2]))
```

```
## [1] "The average MinTemp for cluster 2 is:  0.444628744945036"
```

```
paste("The average MaxTemp for cluster 2 is: ",mean(myData_test_cluster2$MaxTemp[myData_test_cluster2$clusterassig
ned ==2]))
```

```
## [1] "The average MaxTemp for cluster 2 is:  0.428330650747728"
```

```
paste("The average Humidity for cluster 2 is: ",mean(myData_test_cluster2$Humidity[myData_test_cluster2$clusterass
igned ==2]))
```

```
## [1] "The average Humidity for cluster 2 is:  0.662759684473602"
```

```
paste("The average Pressure for cluster 2 is: ",mean(myData_test_cluster2$Pressure[myData_test_cluster2$clusterass
igned ==2]))
```

```
## [1] "The average Pressure for cluster 2 is:  0.62354630132031"
```

```
paste("The average Temp for cluster 2 is: ",mean(myData_test_cluster2$Temp[myData_test_cluster2$clusterassigned ==
2]))
```

```
## [1] "The average Temp for cluster 2 is:  0.407574527064418"
```

```
paste("The average WindGustSpeed for cluster 2 is: ",mean(myData_test_cluster2$WindGustSpeed[myData_test_cluster2$
clusterassigned ==1]))
```

```
## [1] "The average WindGustSpeed for cluster 2 is:  0.277642278849828"
```

Classifying cluster 1 as 'Yes' & cluster 2 as 'No' for Rain Tomorrow on the testing dataset

```
myData_test_cluster2$RainTomorrowPred<- 'Yes'
myData_test_cluster2$RainTomorrowPred[myData_test_cluster2$clusterassigned==2] <- 'No'
myData_test_cluster2$ID <- myData_test_dummies$ID
```

```
str(myData_test_cluster2)
```

```
## 'data.frame':    12994 obs. of  11 variables:
##  $ MinTemp         : num  0.602 0.376 0.279 0.666 0.494 ...
##  $ MaxTemp         : num  0.498 0.291 0.401 0.812 0.579 ...
##  $ Rainfall        : num  0.01197 0.10555 0.000544 0 0 ...
##  $ WindGustSpeed   : num  0.492 0.32 0.234 0.234 0.188 ...
##  $ WindSpeed       : num  0.253 0.218 0.172 0.218 0.149 ...
##  $ Humidity        : num  0.687 0.455 0.475 0.101 0.515 ...
##  $ Pressure        : num  0.305 0.593 0.639 0.53 0.635 ...
##  $ Temp            : num  0.475 0.294 0.388 0.769 0.565 ...
##  $ clusterassigned : int  2 2 2 1 1 2 2 2 1 1 ...
##  $ RainTomorrowPred: chr  "No" "No" "No" "Yes" ...
##  $ ID              : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
summary(myData_test_cluster2)
```

```
##     MinTemp          MaxTemp          Rainfall        WindGustSpeed
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000000   Min.   :0.0000
##  1st Qu.:0.4006   1st Qu.:0.4150   1st Qu.:0.000000   1st Qu.:0.1875
##  Median :0.5166   Median :0.5020   Median :0.000000   Median :0.2500
##  Mean   :0.5249   Mean   :0.5176   Mean   :0.009810   Mean   :0.2719
##  3rd Qu.:0.6519   3rd Qu.:0.6146   3rd Qu.:0.005985   3rd Qu.:0.3359
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.000000   Max.   :1.0000
##    WindSpeed         Humidity         Pressure           Temp
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.1494   1st Qu.:0.4167   1st Qu.:0.5211   1st Qu.:0.3976
##  Median :0.2184   Median :0.5758   Median :0.5965   Median :0.4831
##  Mean   :0.2180   Mean   :0.5687   Mean   :0.5965   Mean   :0.4961
##  3rd Qu.:0.2759   3rd Qu.:0.7273   3rd Qu.:0.6719   3rd Qu.:0.5865
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  clusterassigned RainTomorrowPred         ID
##  Min.   :1.000   Length:12994       Min.   :    1
##  1st Qu.:1.000   Class :character   1st Qu.: 3249
##  Median :2.000   Mode  :character   Median : 6498
##  Mean   :1.597                      Mean   : 6498
##  3rd Qu.:2.000                      3rd Qu.: 9746
##  Max.   :2.000                      Max.   :12994
```

HAC

```
test_hac <- myData_test_cluster[,-9:-90]
```

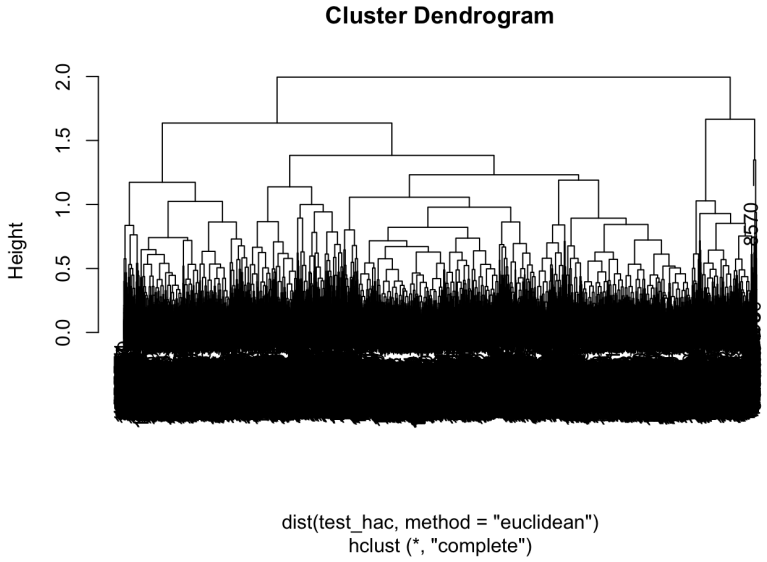
```
str(test_hac)
```

```
## 'data.frame':    12994 obs. of  8 variables:
##  $ MinTemp      : num  0.602 0.376 0.279 0.666 0.494 ...
##  $ MaxTemp      : num  0.498 0.291 0.401 0.812 0.579 ...
##  $ Rainfall     : num  0.01197 0.10555 0.000544 0 0 ...
##  $ WindGustSpeed: num  0.492 0.32 0.234 0.234 0.188 ...
##  $ WindSpeed    : num  0.253 0.218 0.172 0.218 0.149 ...
##  $ Humidity     : num  0.687 0.455 0.475 0.101 0.515 ...
##  $ Pressure     : num  0.305 0.593 0.639 0.53 0.635 ...
##  $ Temp         : num  0.475 0.294 0.388 0.769 0.565 ...
```

```
summary(test_hac)
```

```
##     MinTemp          MaxTemp          Rainfall        WindGustSpeed
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000000   Min.   :0.0000
##  1st Qu.:0.4006   1st Qu.:0.4150   1st Qu.:0.000000   1st Qu.:0.1875
##  Median :0.5166   Median :0.5020   Median :0.000000   Median :0.2500
##  Mean   :0.5249   Mean   :0.5176   Mean   :0.009810   Mean   :0.2719
##  3rd Qu.:0.6519   3rd Qu.:0.6146   3rd Qu.:0.005985   3rd Qu.:0.3359
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.000000   Max.   :1.0000
##    WindSpeed         Humidity         Pressure           Temp
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.1494   1st Qu.:0.4167   1st Qu.:0.5211   1st Qu.:0.3976
##  Median :0.2184   Median :0.5758   Median :0.5965   Median :0.4831
##  Mean   :0.2180   Mean   :0.5687   Mean   :0.5965   Mean   :0.4961
##  3rd Qu.:0.2759   3rd Qu.:0.7273   3rd Qu.:0.6719   3rd Qu.:0.5865
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
```

```
test_hac_output <- hclust(dist(test_hac, method = "euclidean"), method = "complete")
plot(test_hac_output)
```



cut the above dendogram to produce 2 clusters

```
hac_cut_test <- cutree(test_hac_output, 2)
```

Assigining the clusters to the dataset

```
test_hac$clusterassigned_hac <- hac_cut_test
```

```
table(test_hac$clusterassigned_hac)
```

```
##
##     1     2
## 11647  1347
```

summary statistics to support why cluster 1 should be used to classify Rain Tomorrow as 'Yes'

```
paste("The average Rainfall for cluster 1 is: ",mean(test_hac$Rainfall[test_hac$clusterassigned_hac ==1]))
```

```
## [1] "The average Rainfall for cluster 1 is:  0.0105211633140386"
```

```
paste("The average MinTemp for cluster 1 is: ",mean(test_hac$MinTemp[test_hac$clusterassigned_hac ==1]))
```

```
## [1] "The average MinTemp for cluster 1 is:  0.513509394921605"
```

```
paste("The average MaxTemp for cluster 1 is: ",mean(test_hac$MaxTemp[test_hac$clusterassigned_hac ==1]))
```

```
## [1] "The average MaxTemp for cluster 1 is:  0.494344503716202"
```

```
paste("The average Humidity for cluster 1 is: ",mean(test_hac$Humidity[test_hac$clusterassigned_hac ==1]))
```

```
## [1] "The average Humidity for cluster 1 is:  0.606404909401389"
```

```
paste("The average Pressure for cluster 1 is: ",mean(test_hac$Pressure[test_hac$clusterassigned_hac ==1]))
```

```
## [1] "The average Pressure for cluster 1 is:  0.604477321921615"
```

```
paste("The average Temp for cluster 1 is: ",mean(test_hac$Temp[test_hac$clusterassigned_hac ==1]))
```

```
## [1] "The average Temp for cluster 1 is:  0.472566336334189"
```

```
paste("The average WindGustSpeed for cluster 1 is: ",mean(test_hac$WindGustSpeed[test_hac$clusterassigned_hac ==1]
))
```

```
## [1] "The average WindGustSpeed for cluster 1 is:  0.265537128659741"
```

```
paste("The average Rainfall for cluster 2 is: ",mean(test_hac$Rainfall[test_hac$clusterassigned_hac ==2]))
```

```
## [1] "The average Rainfall for cluster 2 is:  0.00366509867977281"
```

```
paste("The average MinTemp for cluster 2 is: ",mean(test_hac$MinTemp[test_hac$clusterassigned_hac ==2]))
```

```
## [1] "The average MinTemp for cluster 2 is:  0.622976370653837"
```

```
paste("The average MaxTemp for cluster 2 is: ",mean(test_hac$MaxTemp[test_hac$clusterassigned_hac ==2]))
```

```
## [1] "The average MaxTemp for cluster 2 is:  0.718921567764407"
```

```
paste("The average Humidity for cluster 2 is: ",mean(test_hac$Humidity[test_hac$clusterassigned_hac ==2]))
```

```
## [1] "The average Humidity for cluster 2 is:  0.242956663892076"
```

```
paste("The average Pressure for cluster 2 is: ",mean(test_hac$Pressure[test_hac$clusterassigned_hac ==2]))
```

```
## [1] "The average Pressure for cluster 2 is:  0.527553106969353"
```

```
paste("The average Temp for cluster 2 is: ",mean(test_hac$Temp[test_hac$clusterassigned_hac ==2]))
```

```
## [1] "The average Temp for cluster 2 is:  0.699137026393975"
```

```
paste("The average WindGustSpeed for cluster 2 is: ",mean(test_hac$WindGustSpeed[test_hac$clusterassigned_hac ==1]
))
```

```
## [1] "The average WindGustSpeed for cluster 2 is:  0.265537128659741"
```

Classifying cluster 1 as 'Yes' & cluster 2 as 'No' for Rain Tomorrow on the testing dataset

```
test_hac$RainTomorrowPred_hac <- 'Yes'
test_hac$RainTomorrowPred_hac[test_hac$clusterassigned==2] <- 'No'
test_hac$ID <- myData_test_dummies$ID
```

```
str(test_hac)
```

```
## 'data.frame':    12994 obs. of  11 variables:
##  $ MinTemp             : num  0.602 0.376 0.279 0.666 0.494 ...
##  $ MaxTemp             : num  0.498 0.291 0.401 0.812 0.579 ...
##  $ Rainfall            : num  0.01197 0.10555 0.000544 0 0 ...
##  $ WindGustSpeed       : num  0.492 0.32 0.234 0.234 0.188 ...
##  $ WindSpeed           : num  0.253 0.218 0.172 0.218 0.149 ...
##  $ Humidity            : num  0.687 0.455 0.475 0.101 0.515 ...
##  $ Pressure            : num  0.305 0.593 0.639 0.53 0.635 ...
##  $ Temp                : num  0.475 0.294 0.388 0.769 0.565 ...
##  $ clusterassigned_hac : int  1 1 1 2 1 1 1 1 2 2 ...
##  $ RainTomorrowPred_hac: chr  "Yes" "Yes" "Yes" "No" ...
##  $ ID                  : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
summary(test_hac)
```

```
##     MinTemp          MaxTemp          Rainfall        WindGustSpeed
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000000   Min.   :0.0000
##  1st Qu.:0.4006   1st Qu.:0.4150   1st Qu.:0.000000   1st Qu.:0.1875
##  Median :0.5166   Median :0.5020   Median :0.000000   Median :0.2500
##  Mean   :0.5249   Mean   :0.5176   Mean   :0.009810   Mean   :0.2719
##  3rd Qu.:0.6519   3rd Qu.:0.6146   3rd Qu.:0.005985   3rd Qu.:0.3359
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.000000   Max.   :1.0000
##    WindSpeed         Humidity         Pressure           Temp
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.1494   1st Qu.:0.4167   1st Qu.:0.5211   1st Qu.:0.3976
##  Median :0.2184   Median :0.5758   Median :0.5965   Median :0.4831
##  Mean   :0.2180   Mean   :0.5687   Mean   :0.5965   Mean   :0.4961
##  3rd Qu.:0.2759   3rd Qu.:0.7273   3rd Qu.:0.6719   3rd Qu.:0.5865
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  clusterassigned_hac RainTomorrowPred_hac       ID
##  Min.   :1.000       Length:12994         Min.   :    1
##  1st Qu.:1.000       Class :character     1st Qu.: 3249
##  Median :1.000       Mode  :character     Median : 6498
##  Mean   :1.104                            Mean   : 6498
##  3rd Qu.:1.000                            3rd Qu.: 9746
##  Max.   :2.000                            Max.   :12994
```

DECISION TREE

Using the final best performing Decision Tree model that produced unbiased and low variance estimates to predict the classes (classify) on the cleaned test dataset

```
decision_tree_test <- predict(decision_tree_model2, newdata = myData_test, na.action = na.omit, type = "raw")
```

Evaluating Model Performance by calculating Accuracy Precision and Recall on the classification done on the Test Data by using the Confusion Matrix

```
myData_test$predicted_rain_tomorrow_dt <- decision_tree_test
```

```
table(myData_test$predicted_rain_tomorrow_dt)
```

```
##
##   No  Yes
## 6661 6333
```

```
str(myData_test)
```

```
## 'data.frame':    12994 obs. of  14 variables:
##  $ ID                        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Location                  : Factor w/ 49 levels "Adelaide","Albany",..: 32 34 16 49 40 29 45 45 22 3 ...
##  $ MinTemp                   : num  15.3 7.1 3.6 17.6 11.4 4.4 12.4 8.3 22.8 18 ...
##  $ MaxTemp                   : num  21.5 11 16.6 37.4 25.6 13 22.4 14.1 38.6 26.9 ...
##  $ Rainfall                  : num  4.4 38.8 0.2 0 0 0.4 0.6 1 0 0.2 ...
##  $ WindGustDir               : Factor w/ 17 levels "","E","ENE","ESE",..: 9 12 9 8 4 9 14 14 2 9 ...
##  $ WindGustSpeed             : num  70 48 37 37 31 39 20 56 54 46 ...
##  $ WindDir                   : Factor w/ 17 levels "","E","ENE","ESE",..: 15 14 5 9 12 9 12 14 8 15 ...
##  $ WindSpeed                 : num  22 19 15 19 13 17 4 22 11 22 ...
##  $ Humidity                  : num  69 46 48 11 52 69 57 58 31 31 ...
##  $ Pressure                  : num  998 1014 1017 1010 1016 ...
##  $ Temp                      : num  19.8 10.7 15.4 34.6 24.3 11.3 21.6 13.6 37.1 25.9 ...
##  $ RainToday                 : Factor w/ 3 levels "","No","Yes": 3 3 2 2 2 2 2 2 2 2 ...
##  $ predicted_rain_tomorrow_dt: Factor w/ 2 levels "No","Yes": 2 1 1 1 1 2 1 2 2 1 ...
```

```
summary(myData_test)
```

```
##        ID                Location        MinTemp        MaxTemp
##  Min.   :    1   Portland     :  357   Min.   :-6.5   Min.   :-3.70
##  1st Qu.: 3249   Hobart       :  336   1st Qu.: 8.0   1st Qu.:17.30
##  Median : 6498   NorfolkIsland:  323   Median :12.2   Median :21.70
##  Mean   : 6498   Dartmoor     :  322   Mean   :12.5   Mean   :22.49
##  3rd Qu.: 9746   Sydney       :  321   3rd Qu.:17.1   3rd Qu.:27.40
##  Max.   :12994   Albany       :  319   Max.   :29.7   Max.   :46.90
##                  (Other)      :11016
##     Rainfall        WindGustDir   WindGustSpeed      WindDir
##  Min.   :  0.000          : 929   Min.   :  7.0   W      :1338
##  1st Qu.:  0.000   W      : 910   1st Qu.: 31.0   SE     : 942
##  Median :  0.000   N      : 905   Median : 39.0   NW     : 888
##  Mean   :  3.606   WSW    : 853   Mean   : 41.8   N      : 880
##  3rd Qu.:  2.200   SE     : 840   3rd Qu.: 50.0   WSW    : 880
##  Max.   :367.600   S      : 832   Max.   :135.0   WNW    : 859
##                   (Other):7725                   (Other):7207
##    WindSpeed        Humidity         Pressure           Temp
##  Min.   : 0.00   Min.   :  1.00   Min.   : 980.2   Min.   :-4.10
##  1st Qu.:13.00   1st Qu.: 42.25   1st Qu.:1009.9   1st Qu.:15.90
##  Median :19.00   Median : 58.00   Median :1014.2   Median :20.20
##  Mean   :18.97   Mean   : 57.30   Mean   :1014.2   Mean   :20.85
##  3rd Qu.:24.00   3rd Qu.: 73.00   3rd Qu.:1018.5   3rd Qu.:25.40
##  Max.   :87.00   Max.   :100.00   Max.   :1037.2   Max.   :46.20
##
##  RainToday  predicted_rain_tomorrow_dt
##     :   0   No :6661
##  No :9064   Yes:6333
##  Yes:3930
##
##
##
##
```

Creating Final Test Data set by merging KMeans, HAC & DT with the 3 predictions for Rain Tomorrow

```
myData_testing_predictions <- merge(merge(myData_test, myData_test_cluster2, by="ID"),test_hac,by = 'ID')

myData_testing_predictions_csv <- myData_testing_predictions[,c('ID','RainTomorrowPred','RainTomorrowPred_hac','predicted_rain_tomorrow_dt')]
colnames(myData_testing_predictions_csv) <- c('ID','kmeans','HAC','DT')
rownames(myData_testing_predictions_csv)<-NULL
```

SUMMARIZING THE Predictions made for Rain Tomorrow on the Test Data by the 3 algorithms

```
predictions_summary <- data.frame(table(myData_testing_predictions_csv$kmeans),
           table(myData_testing_predictions_csv$HAC),
           table(myData_testing_predictions_csv$DT))
predictions_summary <- predictions_summary[,c(1,2,4,6)]
colnames(predictions_summary) <- c('Predicted Rain Tomorrow','kmeans','HAC','DT')
predictions_summary
```

```
##   Predicted Rain Tomorrow kmeans   HAC   DT
## 1                      No   7760  1347 6661
## 2                     Yes   5234 11647 6333
```

Writing to CSV

```
write.csv(myData_testing_predictions_csv,"HW02_AKSHAY_Predictions.csv")
```