*******************import the datasets*******************

```r
air.quality <- read.csv('C:/Users/admin/Desktop/QUT 2019S1/dm/dataset/southbrisbane-aq-2018.csv')
#air quality observation
#8760 observiations of 14 variables
weather.obs <- read.csv('C:/Users/admin/Desktop/QUT 2019S1/dm/dataset/weatherAUS.csv') # weather observ
# 157344 observations with 24 variables


#create a function that checks NA in percentage
percentmiss = function(x)
              {sum(is.na(x))/length(x)*100}
```

*********clean up southbrisbane-aq-2018.csv ***************

Step 1. Remove error/irrelevant data

```r
#Check Summary
cat("\nDataset Summary\n")
```

```
##
## Dataset Summary
```

```r
summary(air.quality)
```

```
##          Date           Time      Wind.Direction..degTN. Wind.Speed..m.s.
##  01/01/2018:  24   00:00  : 365   Min.   :  0.0          Min.   :0.100
##  01/02/2018:  24   01:00  : 365   1st Qu.: 88.0          1st Qu.:1.000
##  01/03/2018:  24   02:00  : 365   Median :177.0          Median :1.400
##  01/04/2018:  24   03:00  : 365   Mean   :166.3          Mean   :1.602
##  01/05/2018:  24   04:00  : 365   3rd Qu.:234.0          3rd Qu.:2.100
##  01/06/2018:  24   05:00  : 365   Max.   :360.0          Max.   :5.900
##  (Other)   :8616   (Other):6570   NA's   :21             NA's   :21
##  Wind.Sigma.Theta..deg. Wind.Speed.Std.Dev..m.s. Air.Temperature..degC.
##  Min.   : 10.80         Min.   :0.1000           Min.   : 4.30
##  1st Qu.: 25.20         1st Qu.:0.5000           1st Qu.:19.70
##  Median : 30.10         Median :0.7000           Median :23.90
##  Mean   : 38.14         Mean   :0.8045           Mean   :23.57
##  3rd Qu.: 38.40         3rd Qu.:1.0000           3rd Qu.:27.90
##  Max.   :193.20         Max.   :3.0000           Max.   :41.80
##  NA's   :21             NA's   :21               NA's   :21
##  Relative.Humidity.... Nitrogen.Oxide..ppm. Nitrogen.Dioxide..ppm.
##  Min.   :13.00         Min.   :0.0000       Min.   :0.0000
##  1st Qu.:54.20         1st Qu.:0.0040       1st Qu.:0.0080
##  Median :69.20         Median :0.0090       Median :0.0130
##  Mean   :66.32         Mean   :0.0156       Mean   :0.0148
##  3rd Qu.:80.40         3rd Qu.:0.0200       3rd Qu.:0.0190
##  Max.   :95.80         Max.   :0.1940       Max.   :0.0550
```

```
##  NA's   :21           NA's   :425          NA's   :425
##  Nitrogen.Oxides..ppm. Carbon.Monoxide..ppm. PM10..ug.m.3.
##  Min.   :0.0000        Min.   :0.0000        Min.   : -1.10
##  1st Qu.:0.0130        1st Qu.:0.1000        1st Qu.: 10.50
##  Median :0.0230        Median :0.1000        Median : 14.70
##  Mean   :0.0302        Mean   :0.1569        Mean   : 17.38
##  3rd Qu.:0.0390        3rd Qu.:0.2000        3rd Qu.: 19.90
##  Max.   :0.2430        Max.   :1.5000        Max.   :403.20
##  NA's   :425           NA's   :411           NA's   :102
##  PM2.5..ug.m.3.
##  Min.   :-4.400
##  1st Qu.: 3.800
##  Median : 6.200
##  Mean   : 7.264
##  3rd Qu.: 9.100
##  Max.   :61.100
##  NA's   :102
```

```
#As from provided air quality dataset description, the negative value in PM2.5 and PM10 are resulting f

air.quality$PM2.5..ug.m.3.[air.quality$PM2.5..ug.m.3. < 0] <- NA
air.quality$PM10..ug.m.3.[air.quality$PM10..ug.m.3. < 0] <- NA
```

Step 2: Deal with NA

```
# Check the Percentage of NA with our defined function. use function to check NA percentage
apply(air.quality,2,percentmiss)
```

```
##                    Date                    Time   Wind.Direction..degTN.
##                0.000000                0.000000                0.239726
##         Wind.Speed..m.s.   Wind.Sigma.Theta..deg. Wind.Speed.Std.Dev..m.s.
##                0.239726                0.239726                0.239726
##   Air.Temperature..degC.    Relative.Humidity....     Nitrogen.Oxide..ppm.
##                0.239726                0.239726                4.851598
##   Nitrogen.Dioxide..ppm.    Nitrogen.Oxides..ppm.    Carbon.Monoxide..ppm.
##                4.851598                4.851598                4.691781
##           PM10..ug.m.3.           PM2.5..ug.m.3.
##                1.198630                2.990868
```

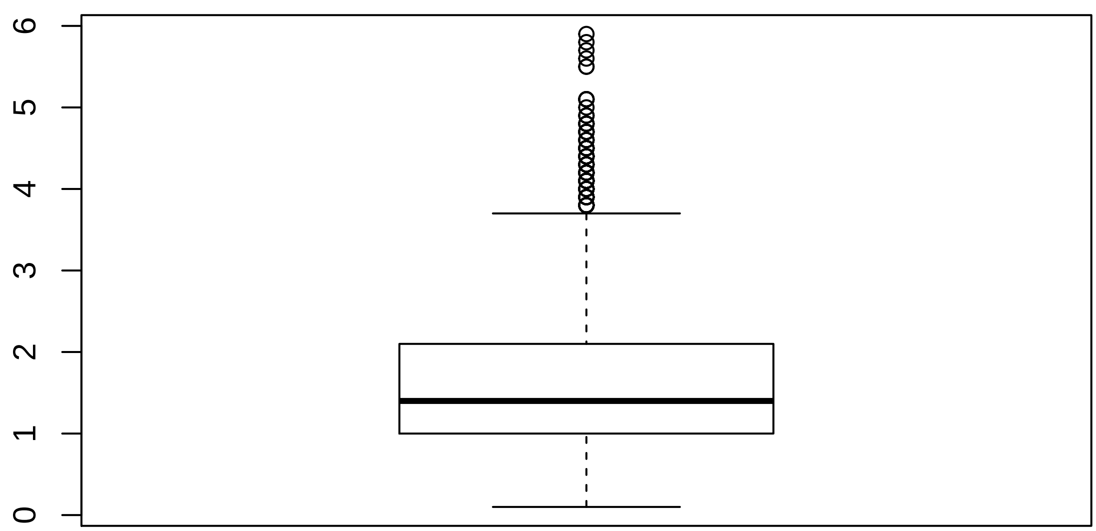
```
# It is not too high so we can replace or remove all NA (In our case as our data is hourley basis, we d



#Imputation of multiple columns
library(imputeTS)
air.quality <- na.mean(air.quality)

#keep the decimal places as original
names(air.quality)
```

```
##  [1] "Date"                    "Time"
##  [3] "Wind.Direction..degTN."   "Wind.Speed..m.s."
##  [5] "Wind.Sigma.Theta..deg."   "Wind.Speed.Std.Dev..m.s."
##  [7] "Air.Temperature..degC."   "Relative.Humidity...."
##  [9] "Nitrogen.Oxide..ppm."     "Nitrogen.Dioxide..ppm."
## [11] "Nitrogen.Oxides..ppm."    "Carbon.Monoxide..ppm."
## [13] "PM10..ug.m.3."            "PM2.5..ug.m.3."
```

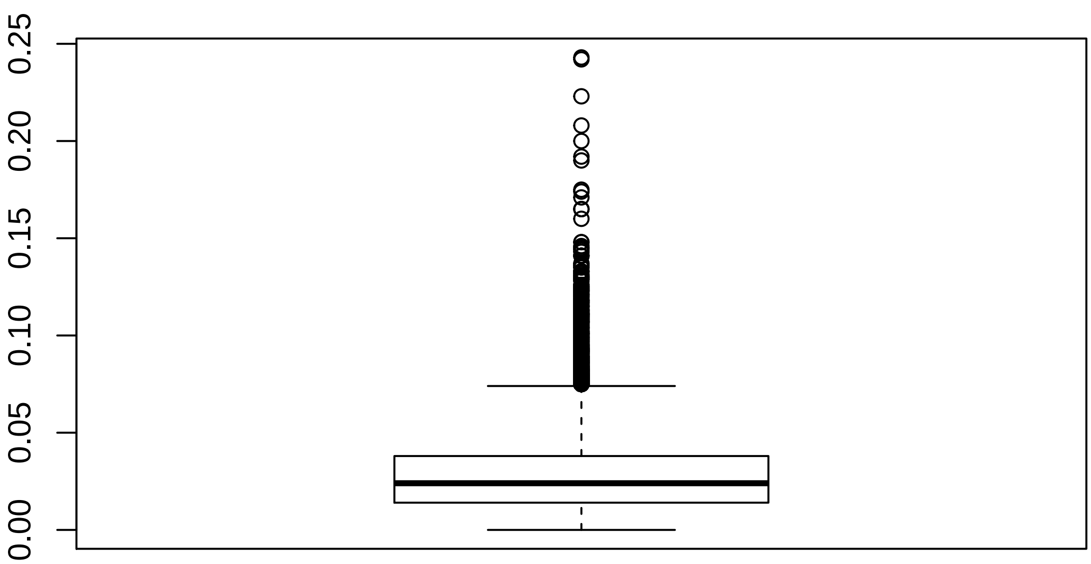
```r
air.quality[12:14] <- round(air.quality[12:14],1)
air.quality[9:11] <- round(air.quality[9:11],3)
air.quality[3:8] <- round(air.quality[3:8],1)

#air.quality <- na.omit(air.quality)
```

Step 3. Check outliers.

```r
library(outliers)

# A. Wind.Speed..m.s has some outliers that can be removed.
boxplot(air.quality$Wind.Speed..m.s.)
```



```
#calculate z-score
outlier_scores<- scores(air.quality$Wind.Speed..m.s.)

# every value more than three standard deviation from the mean we treat it as an outlier
is_outlier <- outlier_scores > 3 | outlier_scores < -3

# try to check the value of these outliers, and decide if we can remove the outliers
air.quality_outliers <- air.quality[outlier_scores > 3| outlier_scores < -3, ]
# head(air.quality_outliers)

# The outliers for wind.speed..ms. we see that outliers values are 4.4, 4.3, 5.1, compared to avg other
# However we assume on this particular time, there was extreme climatic condition that made weather win


# B. Nitrogen.Oxides..ppm. has some outliers that can be removed.
boxplot(air.quality$Nitrogen.Oxides..ppm.)
```




```
#calculate z-score
outlier_scores<- scores(air.quality$Nitrogen.Oxides..ppm.)

# every value more than three standard deviation from the mean we treat it as an outlier
is_outlier <- outlier_scores > 3 | outlier_scores < -3

# try to check the value of these outliers, and decide if we can remove the outliers
air.quality_outliers <- air.quality[outlier_scores > 3| outlier_scores < -3, ]
```

```r
# head(air.quality_outliers)


#add a column is_outlier with result from is_outliers
air.quality$is_outlier <- is_outlier

# replace outliers with NA, which will be imputed later
air.quality$Nitrogen.Oxides..ppm.[air.quality$is_outlier== T] <- NA

# again use imputation and put rounding to original decimal places
library(imputeTS)
air.quality$Nitrogen.Oxides..ppm. <- na.mean(air.quality$Nitrogen.Oxides..ppm.)
air.quality$Nitrogen.Oxides..ppm. <- round(air.quality$Nitrogen.Oxides..ppm.,3)

# Now as we have removed necessary outliers, we can detelet is_outlier column
#names(air.quality)
air.quality <- air.quality[,-(15)]
```

Step 4: Check Data types

```r
# Check structure of air.quality:
#str(air.quality)

#removes scientific notication for numerical value of combined date and time:
options(scipen=999)


# Date and Time are factor, but we would like these two variables to be convertied to numerical as we w
# Also combining Date and time will be better idea as than using date and time separately

air.quality$Date     <- as.Date(air.quality$Date, format = "%d/%m/%Y")
air.quality$DateTime <- paste(air.quality$Date, air.quality$Time)
air.quality$DateTime <- gsub("[: -]", "" , air.quality$DateTime, perl=TRUE)
air.quality$DateTime <- as.numeric(air.quality$DateTime)

# Now as we have desired format of DateTime, we can delete existing Date and Time variable.
#names(air.quality)

air.quality <- air.quality[,-(1:2)]
```

Step 5: Remove Redundant variable

```r
names(air.quality)
```

```
##  [1] "Wind.Direction..degTN."   "Wind.Speed..m.s."
##  [3] "Wind.Sigma.Theta..deg."   "Wind.Speed.Std.Dev..m.s."
##  [5] "Air.Temperature..degC."   "Relative.Humidity...."
```

```
##  [7] "Nitrogen.Oxide..ppm."     "Nitrogen.Dioxide..ppm."
##  [9] "Nitrogen.Oxides..ppm."    "Carbon.Monoxide..ppm."
## [11] "PM10..ug.m.3."            "PM2.5..ug.m.3."
## [13] "DateTime"
```

```
# Here Nitrogen.Oxides..ppm. = Nitrogen.Oxide..ppm. + Nitrogen.Dioxide..ppm., So we can onily use Nitro

air.quality <- air.quality[,-(7:8)]

# Here Wind.Speed.Std.Dev..m.s. is standard deviation of Wind.Speed..m.s.
# And Wind.Sigma.Theta..deg. is standard deviation of Wind.Direction..degTN.
# so we can remove Wind.Speed.Std.Dev..m.s. and Wind.Sigma.Theta..deg.

air.quality <- air.quality[,-(3:4)]

# str(air.quality)
```

***************** clean up weatherAUS.csv ***************************

Step 1. Remove error/irrelevant data

```
# A. Filter Date
# As we will be combining this data with previouseventually, for clean up we will only consider data of
# Merging data of different timestamp can give false output!!


#str(weather.obs)

# Convert Date to Date data type first
weather.obs$Date     <- as.Date(weather.obs$Date, format = "%Y-%m-%d")

# create new dataframe with only data for 2018
weather.obs2018 <- weather.obs[format(weather.obs$Date,'%Y') == "2018", ]

# B. Filter City
# Now as the other dataset only contains data of city Brisbane, in order to provide consistant data and

weather.obsBris2018 <- weather.obs2018[weather.obs2018$Location == "Brisbane", ]

# C. Now as another dataset is hourly basis lets try to make this data as hourly basis

#install.packages("splitstackshape")
library(splitstackshape)
weather.obsBris2018 <- expandRows(weather.obsBris2018, 24, count.is.col=FALSE)

# Now we have 365X24=8760 observations for this dataset.

#str(weather.obsBris2018)
#summary(weather.obsBris2018)
```

Step 2: Check Data types

```
#str(weather.obsBris2018)

# A. Map YES to 1 and No to 0
weather.obsBris2018$RainToday <- as.integer(as.character(weather.obsBris2018$RainToday)=="Yes")
weather.obsBris2018$RainTomorrow <- as.integer(as.character(weather.obsBris2018$RainTomorrow)=="Yes")

#crosscheck output
#table(weather.obsBris2018$RainToday)
```

Step 3: Remove Redundant variable

```
#names(weather.obsBris2018)
#str(weather.obsBris2018)

# we can remove some features in this data set as we already have these information in abother dataset
# Infomration related to wind direction and wind speed

weather.obsBris2018 <- weather.obsBris2018 [,-(8:13)]


#This Dataset is based on Brisbane ie Location="Brisbane" so we can remove Location variable

weather.obsBris2018 <- weather.obsBris2018 [,-2]

#str(weather.obsBris2018)
```

Step 4 : Deal with NA

```
# Check the Percentage of NA with our defined function. use function to check NA percentage
apply(weather.obsBris2018,2,percentmiss)
```

```
##         Date      MinTemp      MaxTemp     Rainfall  Evaporation
##    0.0000000    1.3698630    3.0136986    4.6575342    0.2739726
##     Sunshine  Humidity9am  Humidity3pm  Pressure9am  Pressure3pm
##    0.2739726    0.2739726    0.0000000    0.0000000    0.0000000
##     Cloud9am     Cloud3pm      Temp9am      Temp3pm    RainToday
##    0.0000000    0.0000000    0.2739726    0.0000000    4.6575342
##      RISK_MM RainTomorrow
##    4.6575342    4.6575342
```

```
# It is not too high so we can replace or remove all NA (In our case as our data is hourley basis, we d

#str(weather.obsBris2018)
#summary(weather.obsBris2018)
```

```r
#Imputation of multiple columns (i.e. the whole data frame except first two column, which are catagoric
library(imputeTS)
weather.obsBris2018 <- na.mean(weather.obsBris2018)

#keep the decimal places as original
names(weather.obsBris2018)
```

```
##  [1] "Date"         "MinTemp"      "MaxTemp"      "Rainfall"
##  [5] "Evaporation"  "Sunshine"     "Humidity9am"  "Humidity3pm"
##  [9] "Pressure9am"  "Pressure3pm"  "Cloud9am"     "Cloud3pm"
## [13] "Temp9am"      "Temp3pm"      "RainToday"    "RISK_MM"
## [17] "RainTomorrow"
```

```r
weather.obsBris2018[2:6] <- round(weather.obsBris2018[2:6],1)
weather.obsBris2018[16] <- round(weather.obsBris2018[16],1)
weather.obsBris2018$RainToday <- round(weather.obsBris2018$RainToday,0)
weather.obsBris2018$RainTomorrow <- round(weather.obsBris2018$RainTomorrow,0)
```

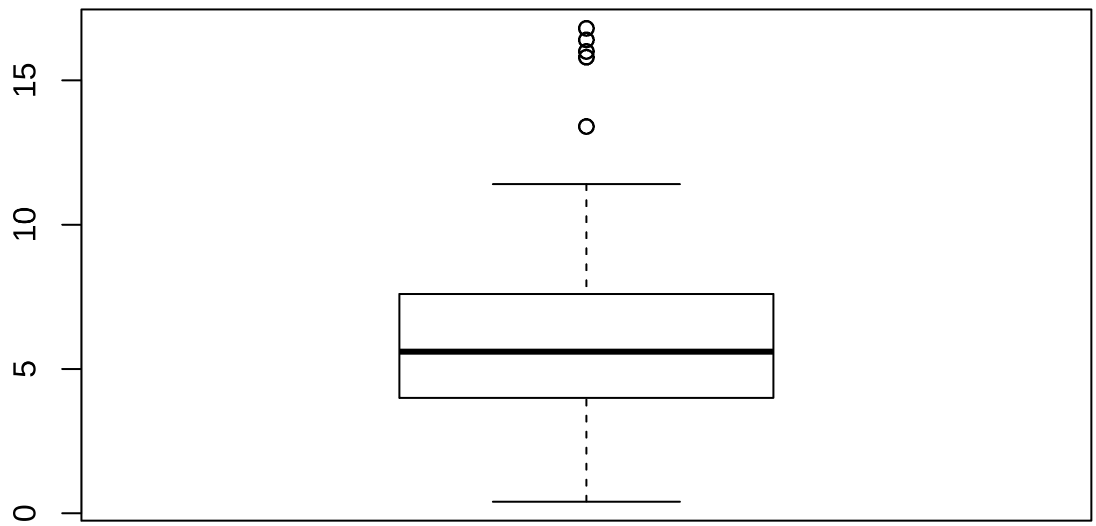
Step 5: Merge features having data in 2 differnt timestamp.

As we are mearging the two datasets on hourly basis, the features in 2 different times does not give significance value. So we can get mean from the two features and create an new variable.

```r
# Create a new variable taking mean from two similar variables.
weather.obsBris2018$Pressure <- rowMeans(weather.obsBris2018[c('Pressure9am', 'Pressure3pm')], na.rm=TRU
weather.obsBris2018$Humidity <- rowMeans(weather.obsBris2018[c('Humidity9am', 'Humidity3pm')], na.rm=TRU
weather.obsBris2018$Cloud    <- rowMeans(weather.obsBris2018[c('Cloud9am', 'Cloud3pm')], na.rm=TRUE)
weather.obsBris2018$Temp    <- rowMeans(weather.obsBris2018[c('Temp9am', 'Temp3pm')], na.rm=TRUE)

# crosscheck the output
#head(weather.obsBris2018)


# As we have created new variable, we can remove existing one.
#names(weather.obsBris2018)
weather.obsBris2018 <- weather.obsBris2018 [,-(7:14)]

#summary(weather.obsBris2018)
```

Step 6. Check outliers.

```r
# A. Wind.Speed..m.s has some outliers that can be removed.
boxplot(weather.obsBris2018$Evaporation)
```




```
#calculate z-score
outlier_scores<- scores(weather.obsBris2018$Evaporation)

# every value more than three standard deviation from the mean we treat it as an outlier
is_outlier <- outlier_scores > 3 | outlier_scores < -3

# try to check the value of these outliers, and decide if we can remove the outliers
weather.obsBris2018_outliers <- weather.obsBris2018[outlier_scores > 3| outlier_scores < -3, ]
#head(weather.obsBris2018_outliers)

nrow(weather.obsBris2018_outliers)
```

```
## [1] 96
```

```
#add a column is_outlier with result from is_outliers
weather.obsBris2018$is_outlier <- is_outlier

# replace outliers with NA, which will be imputed later
weather.obsBris2018$Evaporation[weather.obsBris2018$is_outlier== T] <- NA

#summary(weather.obsBris2018)

#Imputation of multiple columns
weather.obsBris2018 <- na.mean(weather.obsBris2018)
```

```r
#keep the decimal places as original
weather.obsBris2018$Evaporation <- round(weather.obsBris2018$Evaporation,1)

# Now as we have removed necessary outliers, we can detelet is_outlier column
#names(weather.obsBris2018)
weather.obsBris2018 <- weather.obsBris2018[,-(14)]
```

************ Merge 2 datasets **

```r
# First lets copy the DateTime variable from air.quality dataframe to weather.obsBris2018 dataframe
# We will use this variable to merge the two dataframes
weather.obsBris2018$DateTime <- air.quality$DateTime

# merge two data frames by Date
brisbane.climateHour <- merge(air.quality,weather.obsBris2018,by="DateTime")


# new cleaned and mearged dataframe brisbane.climate is created with 22 variables and 8760 observations

#Cross-Check
#head(brisbane.climate)
#summary(brisbane.climate)


#Lets also create Day wise Dataset "brisbane.climateDay"

brisbane.climateDay <- brisbane.climateHour
names(brisbane.climateDay)
```

```
##  [1] "DateTime"                "Wind.Direction..degTN."
##  [3] "Wind.Speed..m.s."        "Air.Temperature..degC."
##  [5] "Relative.Humidity...."   "Nitrogen.Oxides..ppm."
##  [7] "Carbon.Monoxide..ppm."   "PM10..ug.m.3."
##  [9] "PM2.5..ug.m.3."          "Date"
## [11] "MinTemp"                 "MaxTemp"
## [13] "Rainfall"                "Evaporation"
## [15] "Sunshine"                "RainToday"
## [17] "RISK_MM"                 "RainTomorrow"
## [19] "Pressure"                "Humidity"
## [21] "Cloud"                   "Temp"
```

```r
brisbane.climateDay <- brisbane.climateDay[,-1] #remove DateTime

brisbane.climateDay <- aggregate(brisbane.climateDay, by=list(brisbane.climateDay$Date), FUN=mean, na.r

head(brisbane.climateDay)
```

```
##      Group.1 Wind.Direction..degTN. Wind.Speed..m.s.
## 1 2018-01-01               150.8750         1.204167
## 2 2018-01-02               149.9167         1.387500
```

```
## 3 2018-01-03                166.2083         1.666667
## 4 2018-01-04                151.3750         1.470833
## 5 2018-01-05                120.0000         1.495833
## 6 2018-01-06                104.5833         1.333333
##   Air.Temperature..degC. Relative.Humidity.... Nitrogen.Oxides..ppm.
## 1               28.28750              71.06250            0.01400000
## 2               26.43333              73.94167            0.01820833
## 3               26.62500              68.07083            0.01745833
## 4               26.15833              62.46667            0.01762500
## 5               26.64583              61.90833            0.01512500
## 6               27.32500              57.35417            0.01125000
##   Carbon.Monoxide..ppm. PM10..ug.m.3. PM2.5..ug.m.3.       Date MinTemp
## 1            0.13750000     12.495833       7.245833 2018-01-01    24.2
## 2            0.12500000     13.512500       9.425000 2018-01-02    23.4
## 3            0.11250000      8.025000       4.091667 2018-01-03    21.7
## 4            0.08333333     13.287500       4.183333 2018-01-04    21.0
## 5            0.06666667     11.916667       3.679167 2018-01-05    22.2
## 6            0.06666667      9.445833       3.162500 2018-01-06    20.7
##   MaxTemp Rainfall Evaporation Sunshine RainToday RISK_MM RainTomorrow
## 1    32.1      0.4         4.4      6.5         0     0.0            0
## 2    30.8      0.0         5.8      1.3         0     4.8            1
## 3    31.2      4.8         4.6      9.9         1     0.0            0
## 4    29.5      0.0         8.8     11.7         0     0.0            0
## 5    29.5      0.0         9.6      9.7         0     0.0            0
## 6    31.1      0.0         9.2      8.5         0     2.5            0
##   Pressure Humidity Cloud  Temp
## 1  1003.95     66.5   7.0 30.10
## 2  1003.25     70.0   7.5 28.45
## 3  1006.20     66.0   5.0 27.10
## 4  1013.75     56.0   3.0 27.20
## 5  1015.90     58.0   5.5 27.85
## 6  1018.00     51.5   1.5 28.45
```

```r
names(brisbane.climateDay)
```

```
##  [1] "Group.1"                "Wind.Direction..degTN."
##  [3] "Wind.Speed..m.s."       "Air.Temperature..degC."
##  [5] "Relative.Humidity...."  "Nitrogen.Oxides..ppm."
##  [7] "Carbon.Monoxide..ppm."  "PM10..ug.m.3."
##  [9] "PM2.5..ug.m.3."         "Date"
## [11] "MinTemp"                "MaxTemp"
## [13] "Rainfall"               "Evaporation"
## [15] "Sunshine"               "RainToday"
## [17] "RISK_MM"                "RainTomorrow"
## [19] "Pressure"               "Humidity"
## [21] "Cloud"                  "Temp"
```

```r
brisbane.climateDay <- brisbane.climateDay[,-10] #Remove Date

# Convert Group.1 (date) to numeric
brisbane.climateDay$Group.1 <- gsub("[: -]", "" , brisbane.climateDay$Group.1, perl=TRUE)
str(brisbane.climateDay)
```

```
## 'data.frame':    365 obs. of  21 variables:
##  $ Group.1                : chr  "20180101" "20180102" "20180103" "20180104" ...
##  $ Wind.Direction..degTN.: num  151 150 166 151 120 ...
##  $ Wind.Speed..m.s.      : num  1.2 1.39 1.67 1.47 1.5 ...
##  $ Air.Temperature..degC.: num  28.3 26.4 26.6 26.2 26.6 ...
##  $ Relative.Humidity.... : num  71.1 73.9 68.1 62.5 61.9 ...
##  $ Nitrogen.Oxides..ppm. : num  0.014 0.0182 0.0175 0.0176 0.0151 ...
##  $ Carbon.Monoxide..ppm. : num  0.1375 0.125 0.1125 0.0833 0.0667 ...
##  $ PM10..ug.m.3.         : num  12.5 13.51 8.03 13.29 11.92 ...
##  $ PM2.5..ug.m.3.        : num  7.25 9.42 4.09 4.18 3.68 ...
##  $ MinTemp               : num  24.2 23.4 21.7 21 22.2 20.7 16.4 16.4 22.3 22 ...
##  $ MaxTemp               : num  32.1 30.8 31.2 29.5 29.5 31.1 31.5 32.4 32.1 32.3 ...
##  $ Rainfall              : num  0.4 0 4.8 0 0 0 2.5 2.5 0 0 ...
##  $ Evaporation           : num  4.4 5.8 4.6 8.8 9.6 9.2 8.4 9.4 11 10 ...
##  $ Sunshine              : num  6.5 1.3 9.9 11.7 9.7 8.5 13 13.1 11.7 11.5 ...
##  $ RainToday             : num  0 0 1 0 0 0 0 0 0 0 ...
##  $ RISK_MM               : num  0 4.8 0 0 0 2.5 2.5 0 0 0 ...
##  $ RainTomorrow          : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ Pressure              : num  1004 1003 1006 1014 1016 ...
##  $ Humidity              : num  66.5 70 66 56 58 51.5 55.5 51 48.5 49.5 ...
##  $ Cloud                 : num  7 7.5 5 3 5.5 1.5 1.5 1 1.5 3 ...
##  $ Temp                  : num  30.1 28.5 27.1 27.2 27.9 ...
```

```r
brisbane.climateDay$Group.1 <- as.numeric(brisbane.climateDay$Group.1)

#round to original decimal place
names(brisbane.climateDay)
```

```
##  [1] "Group.1"                "Wind.Direction..degTN."
##  [3] "Wind.Speed..m.s."       "Air.Temperature..degC."
##  [5] "Relative.Humidity...."  "Nitrogen.Oxides..ppm."
##  [7] "Carbon.Monoxide..ppm."  "PM10..ug.m.3."
##  [9] "PM2.5..ug.m.3."         "MinTemp"
## [11] "MaxTemp"                "Rainfall"
## [13] "Evaporation"            "Sunshine"
## [15] "RainToday"              "RISK_MM"
## [17] "RainTomorrow"           "Pressure"
## [19] "Humidity"               "Cloud"
## [21] "Temp"
```

```r
brisbane.climateDay[2]    <- round(brisbane.climateDay[2],0)
brisbane.climateDay[3:5]  <- round(brisbane.climateDay[3:5],1)
brisbane.climateDay[6]   <- round(brisbane.climateDay[6],3)
brisbane.climateDay[7:9] <- round(brisbane.climateDay[7:9],1)
brisbane.climateDay[18:21] <- round(brisbane.climateDay[18:21],2)



str(brisbane.climateDay)
```

```
## 'data.frame':    365 obs. of  21 variables:
##  $ Group.1                : num  20180101 20180102 20180103 20180104 20180105 ...
##  $ Wind.Direction..degTN.: num  151 150 166 151 120 105 117 177 125 121 ...
```

```
##  $ Wind.Speed..m.s.       : num  1.2 1.4 1.7 1.5 1.5 1.3 1.5 1.8 1.7 1.5 ...
##  $ Air.Temperature..degC.: num  28.3 26.4 26.6 26.2 26.6 27.3 27.4 28.1 28.4 28.6 ...
##  $ Relative.Humidity.... : num  71.1 73.9 68.1 62.5 61.9 57.4 60.1 58.9 56.8 58.5 ...
##  $ Nitrogen.Oxides..ppm. : num  0.014 0.018 0.017 0.018 0.015 0.011 0.011 0.016 0.018 0.021 ...
##  $ Carbon.Monoxide..ppm. : num  0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ...
##  $ PM10..ug.m.3.         : num  12.5 13.5 8 13.3 11.9 9.4 10 11.2 9.9 12 ...
##  $ PM2.5..ug.m.3.        : num  7.2 9.4 4.1 4.2 3.7 3.2 4 4.1 4.2 4.5 ...
##  $ MinTemp               : num  24.2 23.4 21.7 21 22.2 20.7 16.4 16.4 22.3 22 ...
##  $ MaxTemp               : num  32.1 30.8 31.2 29.5 29.5 31.1 31.5 32.4 32.1 32.3 ...
##  $ Rainfall              : num  0.4 0 4.8 0 0 0 2.5 2.5 0 0 ...
##  $ Evaporation           : num  4.4 5.8 4.6 8.8 9.6 9.2 8.4 9.4 11 10 ...
##  $ Sunshine              : num  6.5 1.3 9.9 11.7 9.7 8.5 13 13.1 11.7 11.5 ...
##  $ RainToday             : num  0 0 1 0 0 0 0 0 0 0 ...
##  $ RISK_MM               : num  0 4.8 0 0 0 2.5 2.5 0 0 0 ...
##  $ RainTomorrow          : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ Pressure              : num  1004 1003 1006 1014 1016 ...
##  $ Humidity              : num  66.5 70 66 56 58 51.5 55.5 51 48.5 49.5 ...
##  $ Cloud                 : num  7 7.5 5 3 5.5 1.5 1.5 1 1.5 3 ...
##  $ Temp                  : num  30.1 28.4 27.1 27.2 27.9 ...
```

************ Corelation ***

```r
#install.packages("ggplot2")
library(ggplot2)

library(corrr)

names(brisbane.climateDay)
```

```
##  [1] "Group.1"                "Wind.Direction..degTN."
##  [3] "Wind.Speed..m.s."       "Air.Temperature..degC."
##  [5] "Relative.Humidity...."  "Nitrogen.Oxides..ppm."
##  [7] "Carbon.Monoxide..ppm."  "PM10..ug.m.3."
##  [9] "PM2.5..ug.m.3."         "MinTemp"
## [11] "MaxTemp"                "Rainfall"
## [13] "Evaporation"            "Sunshine"
## [15] "RainToday"              "RISK_MM"
## [17] "RainTomorrow"           "Pressure"
## [19] "Humidity"               "Cloud"
## [21] "Temp"
```

```r
str(brisbane.climateDay)
```

```
## 'data.frame':    365 obs. of  21 variables:
##  $ Group.1               : num  20180101 20180102 20180103 20180104 20180105 ...
##  $ Wind.Direction..degTN.: num  151 150 166 151 120 105 117 177 125 121 ...
##  $ Wind.Speed..m.s.       : num  1.2 1.4 1.7 1.5 1.5 1.3 1.5 1.8 1.7 1.5 ...
##  $ Air.Temperature..degC.: num  28.3 26.4 26.6 26.2 26.6 27.3 27.4 28.1 28.4 28.6 ...
##  $ Relative.Humidity.... : num  71.1 73.9 68.1 62.5 61.9 57.4 60.1 58.9 56.8 58.5 ...
##  $ Nitrogen.Oxides..ppm. : num  0.014 0.018 0.017 0.018 0.015 0.011 0.011 0.016 0.018 0.021 ...
```

```
##  $ Carbon.Monoxide..ppm. : num  0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ...
##  $ PM10..ug.m.3.         : num  12.5 13.5 8 13.3 11.9 9.4 10 11.2 9.9 12 ...
##  $ PM2.5..ug.m.3.        : num  7.2 9.4 4.1 4.2 3.7 3.2 4 4.1 4.2 4.5 ...
##  $ MinTemp               : num  24.2 23.4 21.7 21 22.2 20.7 16.4 16.4 22.3 22 ...
##  $ MaxTemp               : num  32.1 30.8 31.2 29.5 29.5 31.1 31.5 32.4 32.1 32.3 ...
##  $ Rainfall              : num  0.4 0 4.8 0 0 0 2.5 2.5 0 0 ...
##  $ Evaporation           : num  4.4 5.8 4.6 8.8 9.6 9.2 8.4 9.4 11 10 ...
##  $ Sunshine              : num  6.5 1.3 9.9 11.7 9.7 8.5 13 13.1 11.7 11.5 ...
##  $ RainToday             : num  0 0 1 0 0 0 0 0 0 0 ...
##  $ RISK_MM               : num  0 4.8 0 0 0 2.5 2.5 0 0 0 ...
##  $ RainTomorrow          : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ Pressure              : num  1004 1003 1006 1014 1016 ...
##  $ Humidity              : num  66.5 70 66 56 58 51.5 55.5 51 48.5 49.5 ...
##  $ Cloud                 : num  7 7.5 5 3 5.5 1.5 1.5 1 1.5 3 ...
##  $ Temp                  : num  30.1 28.4 27.1 27.2 27.9 ...
```

```
d <- correlate(brisbane.climateDay)
```

```
##
## Correlation method: 'pearson'
## Missing treated using: 'pairwise.complete.obs'
```

```
a <- rearrange(d)
a
```

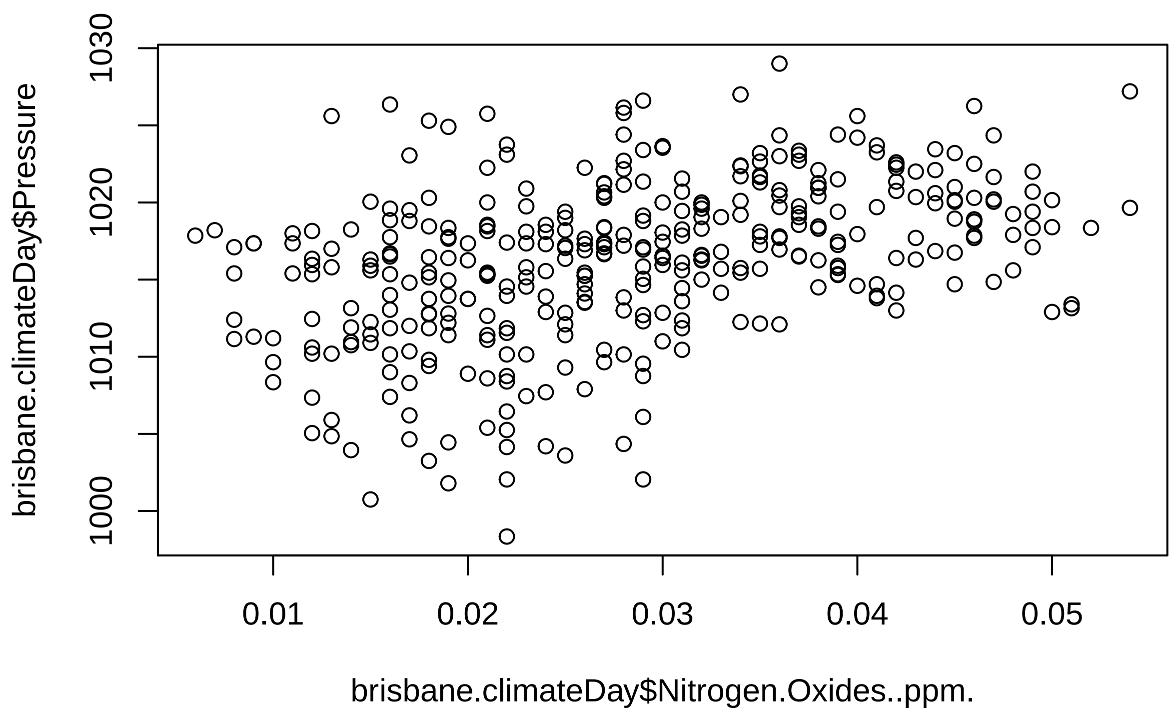
```
## # A tibble: 21 x 22
##    rowname   Temp Air.Temperature~ MaxTemp Nitrogen.Oxides~ Pressure
##    <chr>    <dbl>            <dbl>   <dbl>            <dbl>    <dbl>
##  1 Temp    NA                0.870   0.937           -0.652   -0.714
##  2 Air.Te~  0.870           NA       0.815           -0.667   -0.638
##  3 MaxTemp  0.937            0.815  NA               -0.553   -0.726
##  4 Nitrog~ -0.652           -0.667  -0.553           NA        0.421
##  5 Pressu~ -0.714           -0.638  -0.726            0.421   NA
##  6 MinTemp  0.849            0.766   0.732           -0.628   -0.595
##  7 Carbon~ -0.507           -0.621  -0.405            0.676    0.287
##  8 Evapor~  0.550            0.573   0.501           -0.494   -0.378
##  9 Wind.D~ -0.467           -0.576  -0.424            0.488    0.221
## 10 Wind.S~  0.213            0.257   0.137           -0.332   -0.180
## # ... with 11 more rows, and 16 more variables: MinTemp <dbl>,
## #   Carbon.Monoxide..ppm. <dbl>, Evaporation <dbl>,
## #   Wind.Direction..degTN. <dbl>, Wind.Speed..m.s. <dbl>, Group.1 <dbl>,
## #   PM2.5..ug.m.3. <dbl>, PM10..ug.m.3. <dbl>, Sunshine <dbl>,
## #   Rainfall <dbl>, RISK_MM <dbl>, RainToday <dbl>, Cloud <dbl>,
## #   Relative.Humidity.... <dbl>, RainTomorrow <dbl>, Humidity <dbl>
```

```
# From this dataframe we can see there is some +ve corelation between Carbon.Monoxide..ppm. (AirQuality

# Lets examine the corelation between those two features.
cor.test(brisbane.climateDay$Nitrogen.Oxides..ppm., brisbane.climateDay$Wind.Direction..degTN., method=
```

```
##
##  Pearson's product-moment correlation
```

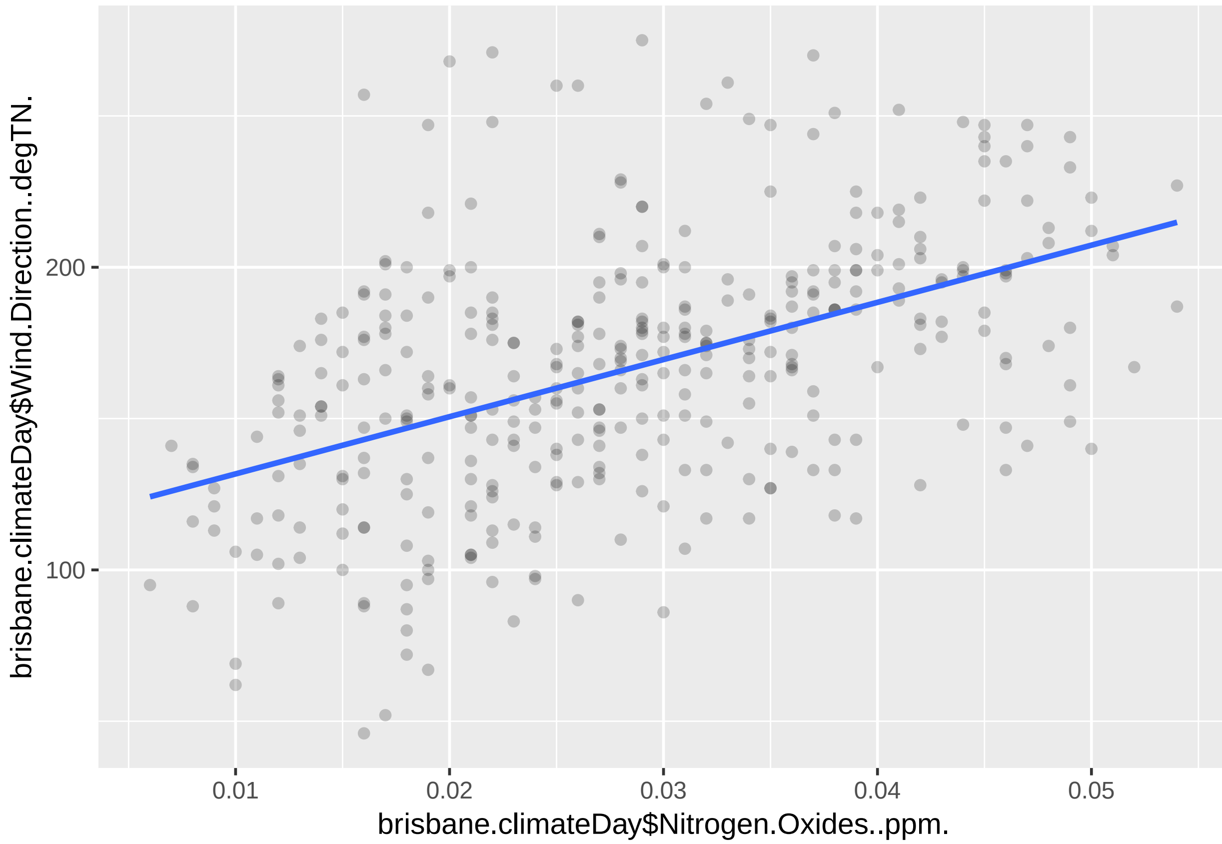
```
##
## data:  brisbane.climateDay$Nitrogen.Oxides..ppm. and brisbane.climateDay$Wind.Direction..degTN.
## t = 10.658, df = 363, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4059088 0.5626689
## sample estimates:
##       cor
## 0.4882169
```

```
plot(brisbane.climateDay$Nitrogen.Oxides..ppm., brisbane.climateDay$Pressure)
```




```
qplot(x = brisbane.climateDay$Nitrogen.Oxides..ppm.,
      y = brisbane.climateDay$Wind.Direction..degTN.,
      geom = c("point", "smooth"),
      method = "lm",
      alpha = I(1 / 5),
      se = FALSE)
```

```
## Warning: Ignoring unknown parameters: method, se
```

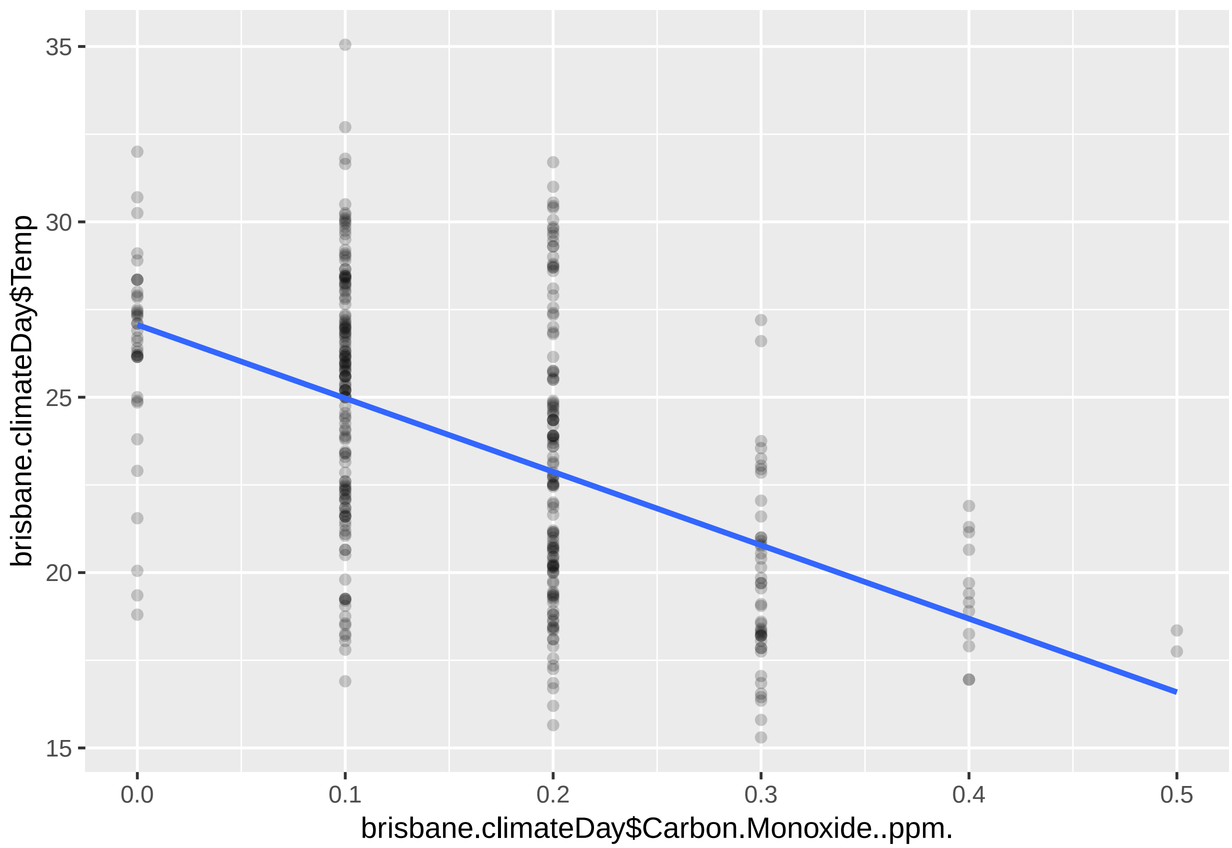



```
cor.test(brisbane.climateDay$Carbon.Monoxide..ppm., brisbane.climateDay$Temp, method="pearson") #-0.506
```

```
##
##  Pearson's product-moment correlation
##
## data:  brisbane.climateDay$Carbon.Monoxide..ppm. and brisbane.climateDay$Temp
## t = -11.197, df = 363, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5791825 -0.4261679
## sample estimates:
##        cor
## -0.5066542
```

```
qplot(x = brisbane.climateDay$Carbon.Monoxide..ppm.,
      y = brisbane.climateDay$Temp,
      geom = c("point", "smooth"),
      method = "lm",
      alpha = I(1 / 5),
      se = FALSE)
```

```
## Warning: Ignoring unknown parameters: method, se
```



************************* Decision Tree *****************************

```
#install.packages("party")
#install.packages("caret")
#install.packages("e1071")
#install.packages("lattice")
#install.packages("sandwich")
#str(brisbane.climateDT)
library(caret)
```

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

```
library(party)
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

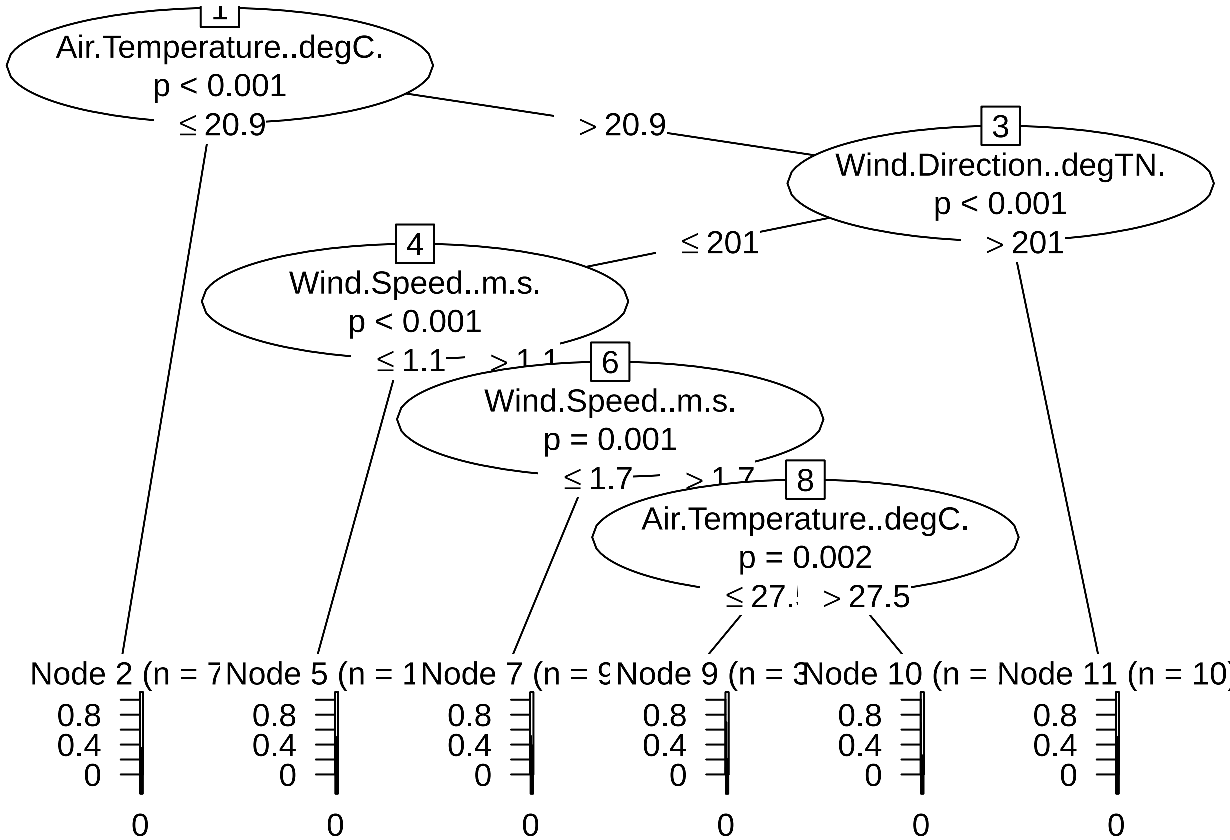
```
## Loading required package: strucchange

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following object is masked from 'package:imputeTS':
##
##     na.locf

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich
```

```
# We select brisbane.climateDay dataset instead of hourly dataset because the questions ask us to predi

# lets prepare our dataset from "brisbane.climateDay", target variable as Carbon.Monoxide..ppm. (air qu
brisbane.climateDT <- brisbane.climateDay[,c("Carbon.Monoxide..ppm.",
                                        "Wind.Direction..degTN.",
                                        "Wind.Speed..m.s.",
                                        "Pressure",
                                        "Evaporation",
                                        "Air.Temperature..degC.",
                                        "Temp"
                                        )]



# convert "Carbon.Monoxide..ppm." to factor
brisbane.climateDT$Carbon.Monoxide..ppm. <- as.factor(brisbane.climateDT$Carbon.Monoxide..ppm.)
str(brisbane.climateDT)
```

```
## 'data.frame':    365 obs. of  7 variables:
##  $ Carbon.Monoxide..ppm. : Factor w/ 6 levels "0","0.1","0.2",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Wind.Direction..degTN.: num  151 150 166 151 120 105 117 177 125 121 ...
##  $ Wind.Speed..m.s.      : num  1.2 1.4 1.7 1.5 1.5 1.3 1.5 1.8 1.7 1.5 ...
##  $ Pressure              : num  1004 1003 1006 1014 1016 ...
##  $ Evaporation           : num  4.4 5.8 4.6 8.8 9.6 9.2 8.4 9.4 11 10 ...
##  $ Air.Temperature..degC.: num  28.3 26.4 26.6 26.2 26.6 27.3 27.4 28.1 28.4 28.6 ...
##  $ Temp                  : num  30.1 28.4 27.1 27.2 27.9 ...
```

```
# set seed as a random number
set.seed(19)

#data set allocation, 70% training and 30% test
data_split <- sample(2, nrow(brisbane.climateDT), replace=TRUE, prob=c(0.7, 0.3))
train_data <- brisbane.climateDT [data_split==1,]
test_data <- brisbane.climateDT [data_split==2,]
```

```r
# formula for decision tree
formula <- Carbon.Monoxide..ppm. ~  Wind.Direction..degTN. +  Wind.Speed..m.s. + Pressure + Evaporation

# train our model
dt <- ctree(formula, data = train_data)

# plot of decision tree
plot(dt)
```




```r
# make predictions for the test set
predictions <- predict(dt, newdata = test_data)

# make a confusion matrix for the test set
confusionMatrix(predict(dt, newdata = test_data), test_data$Carbon.Monoxide..ppm.)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0 0.1 0.2 0.3 0.4 0.5
##        0    4   5   0   0   0   0
##        0.1  8  44  28   2   0   0
##        0.2  0   3   4   2   0   0
##        0.3  0   2  14  11   1   1
##        0.4  0   0   0   0   0   0
```

```
##        0.5  0   0   0   0   0   0
##
## Overall Statistics
##
##                Accuracy : 0.4884
##                  95% CI : (0.3994, 0.5779)
##     No Information Rate : 0.4186
##     P-Value [Acc > NIR] : 0.06526
##
##                   Kappa : 0.2436
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 0 Class: 0.1 Class: 0.2 Class: 0.3 Class: 0.4
## Sensitivity           0.33333     0.8148    0.08696    0.73333   0.000000
## Specificity           0.95726     0.4933    0.93976    0.84211   1.000000
## Pos Pred Value        0.44444     0.5366    0.44444    0.37931        NaN
## Neg Pred Value        0.93333     0.7872    0.65000    0.96000   0.992248
## Prevalence            0.09302     0.4186    0.35659    0.11628   0.007752
## Detection Rate        0.03101     0.3411    0.03101    0.08527   0.000000
## Detection Prevalence  0.06977     0.6357    0.06977    0.22481   0.000000
## Balanced Accuracy     0.64530     0.6541    0.51336    0.78772   0.500000
##                      Class: 0.5
## Sensitivity            0.000000
## Specificity            1.000000
## Pos Pred Value              NaN
## Neg Pred Value         0.992248
## Prevalence             0.007752
## Detection Rate         0.000000
## Detection Prevalence   0.000000
## Balanced Accuracy      0.500000
```

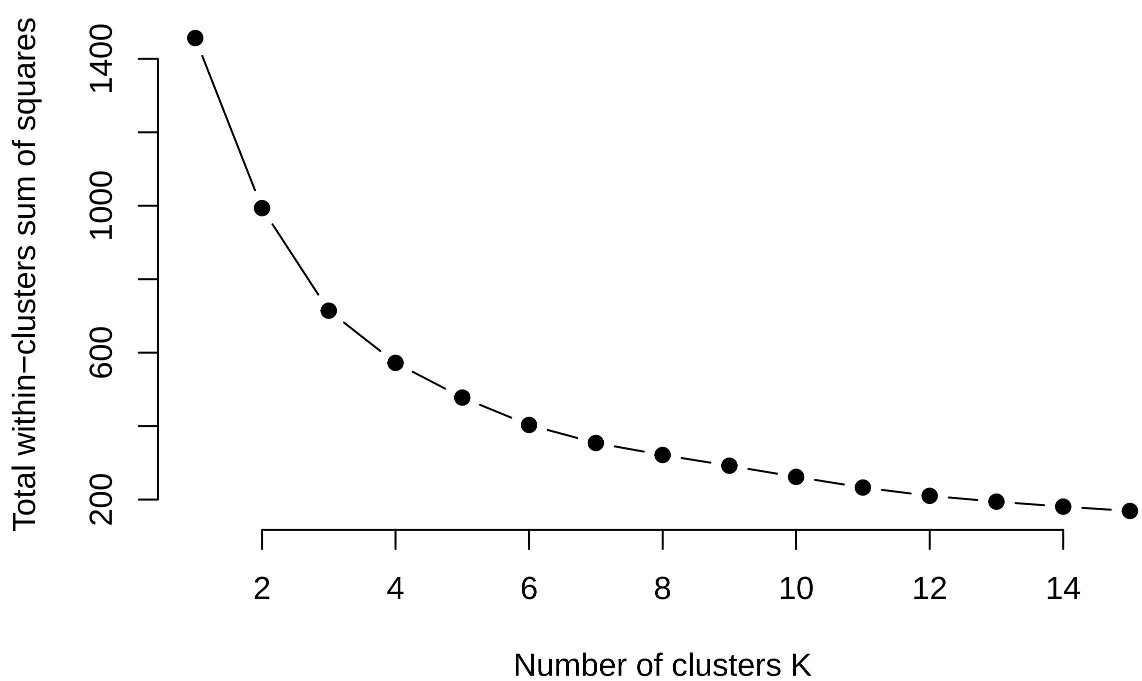
************************* K-means Clustering ************************

```r
#install.packages("NbClust")
#install.packages("factoextra")
library(ggplot2)
library(gridExtra)
library(NbClust)
library(factoextra)
```

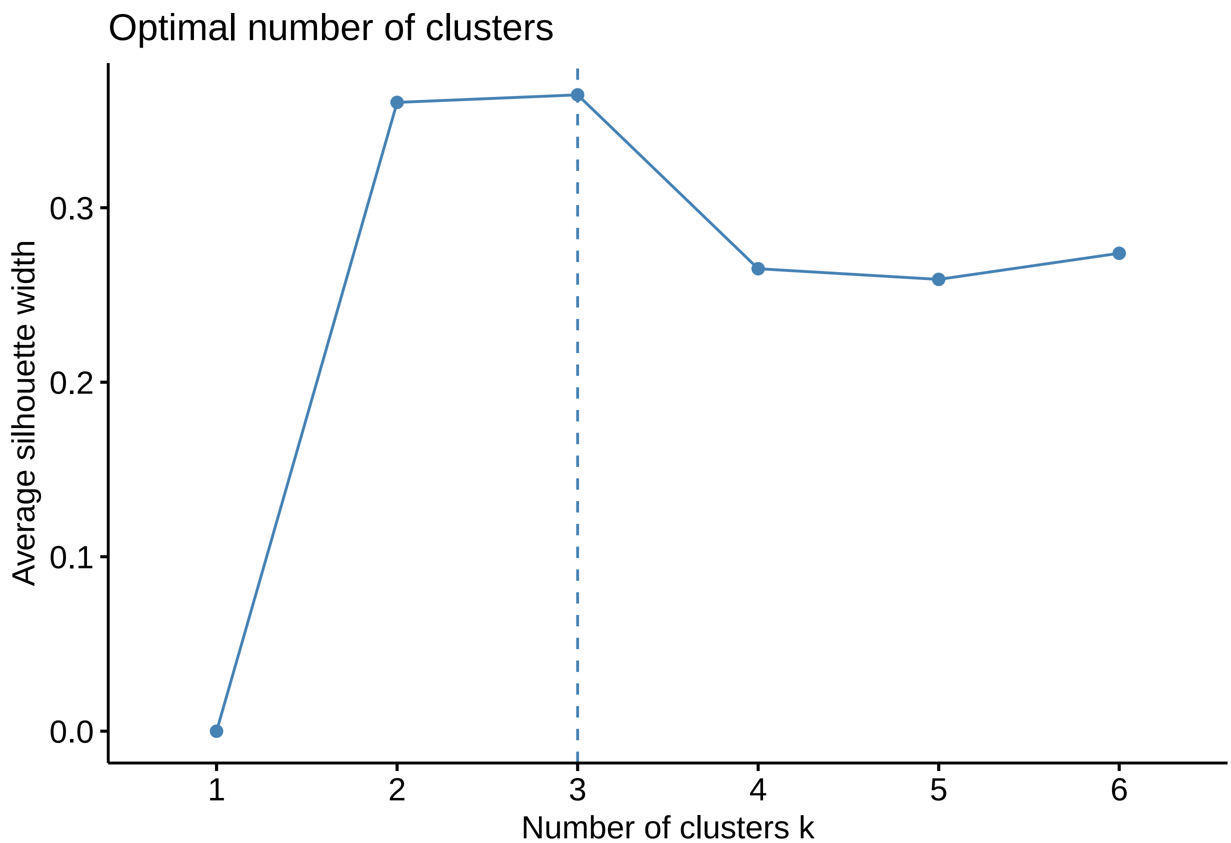
```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```r
# Function for finding the optimal number of clusters using Elbow Method

OptimalK = function(data.clu) {
           set.seed(123)
           # Compute and plot wss for k = 2 to k = 15.
           k.max <- 15
```

```
        data <- data.clu
        wss <- sapply(1:k.max,
                      function(k){kmeans(data, k, nstart=50,iter.max = 15 )$tot.withinss})
        wss
        plot(1:k.max, wss,
             type="b", pch = 19, frame = FALSE,
             xlab="Number of clusters K",
             ylab="Total within-clusters sum of squares")
       }

# Prepare our dataset for clustering
brisbane.climateCLU <- brisbane.climateDay[,c("Nitrogen.Oxides..ppm.",
                                              "Carbon.Monoxide..ppm.",
                                              "PM10..ug.m.3.",
                                              "PM2.5..ug.m.3."
                                               )]


brisbane.climateCLU_scaled <- scale(brisbane.climateCLU)
brisbane.climateCLU_scaled <- as.data.frame(brisbane.climateCLU_scaled)
brisbane.climateCLU_scaled <- round(brisbane.climateCLU_scaled,2)
#str(brisbane.climateCLU)

# Call our function for finding Optimal K using Elbow method
OptimalK(brisbane.climateCLU_scaled)
```



```r
# From Elbow method we see Optimal number for cluster is between 2,3,4,5 and 6
# Now we can choose K between 2,3,4,5 and 6 using Silhouette method

fviz_nbclust(brisbane.climateCLU_scaled, kmeans, method = "silhouette", k.max = 6)
```

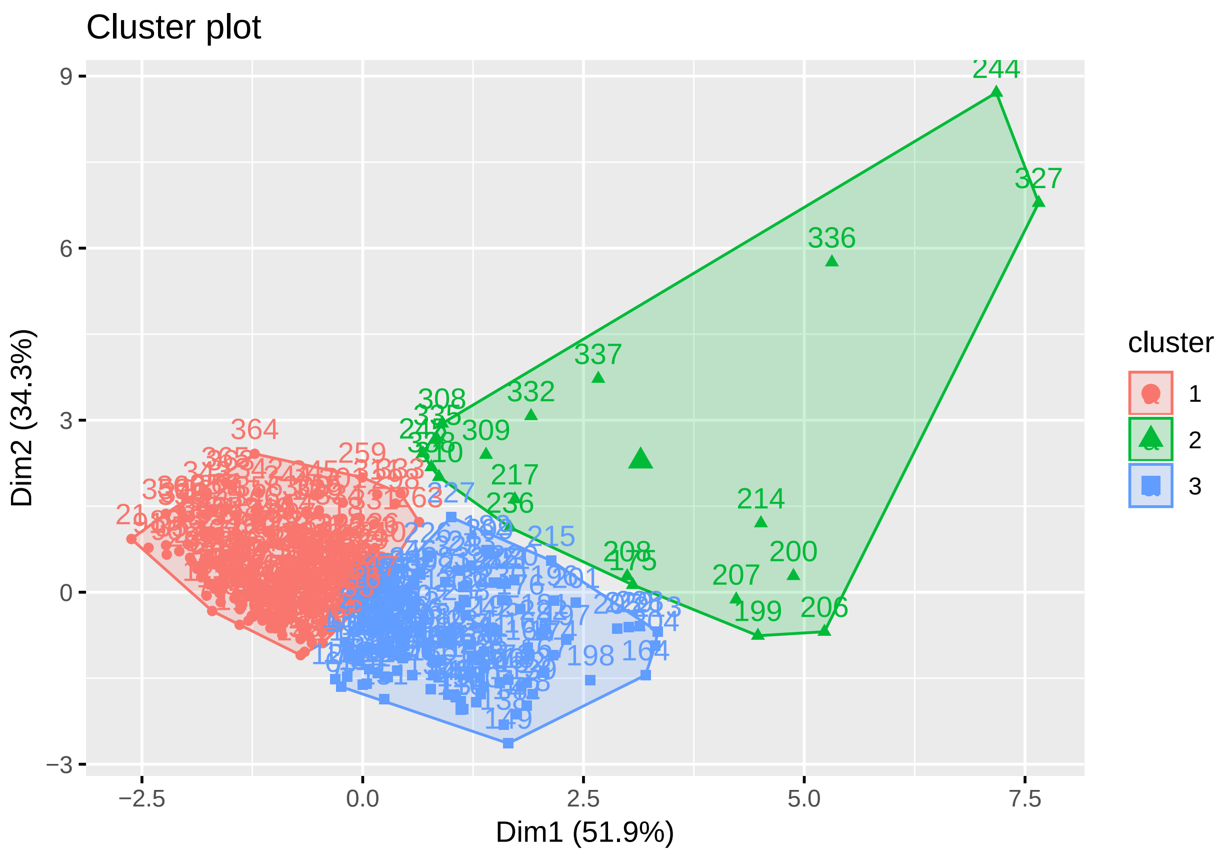



```r
# So using Elbow method and silhouette scale, we can set our no of cluster to 3 (K=3)

# set the seed for the random initialisation
set.seed(1234)

# cluster the dataframe e2_df using kmeans, with the value of K from the function parameters
kmeans_result <- kmeans(brisbane.climateCLU_scaled, 3)
kmeans_result
```

```
## K-means clustering with 3 clusters of sizes 211, 20, 134
##
## Cluster means:
##   Nitrogen.Oxides..ppm. Carbon.Monoxide..ppm. PM10..ug.m.3. PM2.5..ug.m.3.
## 1            -0.6034597            -0.5743602   -0.23208531     -0.3990521
## 2            -0.0095000             0.7165000    2.59400000      2.8275000
## 3             0.9512687             0.7937313   -0.02164179      0.2064925
##
## Clustering vector:
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1
##  [36] 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 3 1 1 1 1 1
```

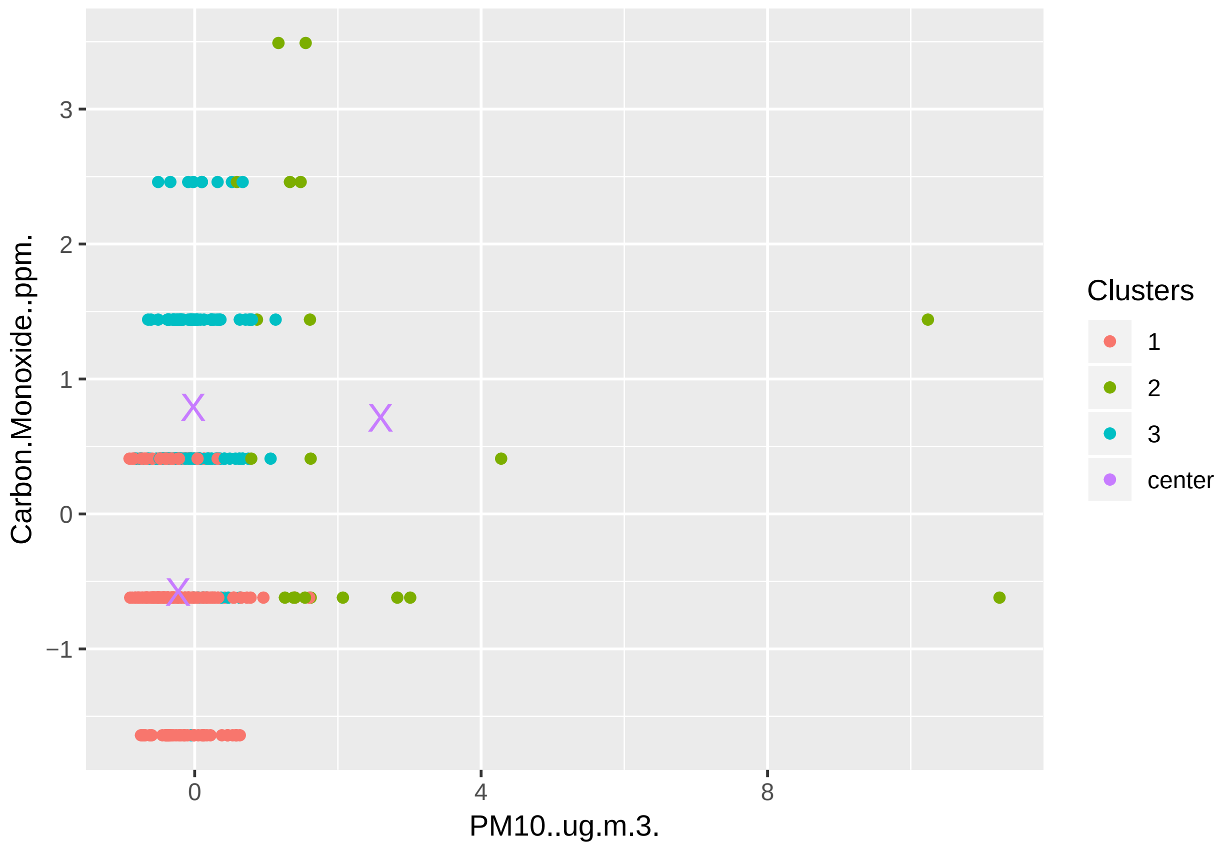
```
##  [71] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 3 1 1 1
## [106] 3 3 3 3 3 1 1 3 3 1 3 3 1 1 3 3 3 3 3 1 1 1 3 3 3 3 1 1 3 3 3 3 3 3 3
## [141] 3 3 3 3 3 3 1 1 3 3 3 3 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 3 3 3 3 3 2
## [176] 3 3 3 3 3 3 3 3 1 3 3 1 1 1 3 3 3 3 3 3 3 3 3 2 2 3 3 3 3 3 2 2 2 1 1
## [211] 3 3 3 2 3 3 2 3 3 3 3 3 1 1 3 3 3 3 3 3 1 3 3 3 3 2 1 3 1 3 3 3 3 2 2
## [246] 1 1 1 3 3 1 1 1 1 1 3 3 1 1 1 3 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 3 3 3 3 1
## [281] 3 3 3 1 3 1 1 1 1 3 1 1 1 1 1 1 3 1 3 1 1 3 3 3 3 1 1 2 2 2 1 3 3 1 1
## [316] 1 1 1 1 1 1 1 1 1 1 1 1 2 3 1 3 1 2 1 1 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1
## [351] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 261.9405 263.6868 223.3253
##  (between_SS / total_SS =  48.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```r
# This function will plot K means result
fviz_cluster(kmeans_result, data = brisbane.climateCLU_scaled)
```

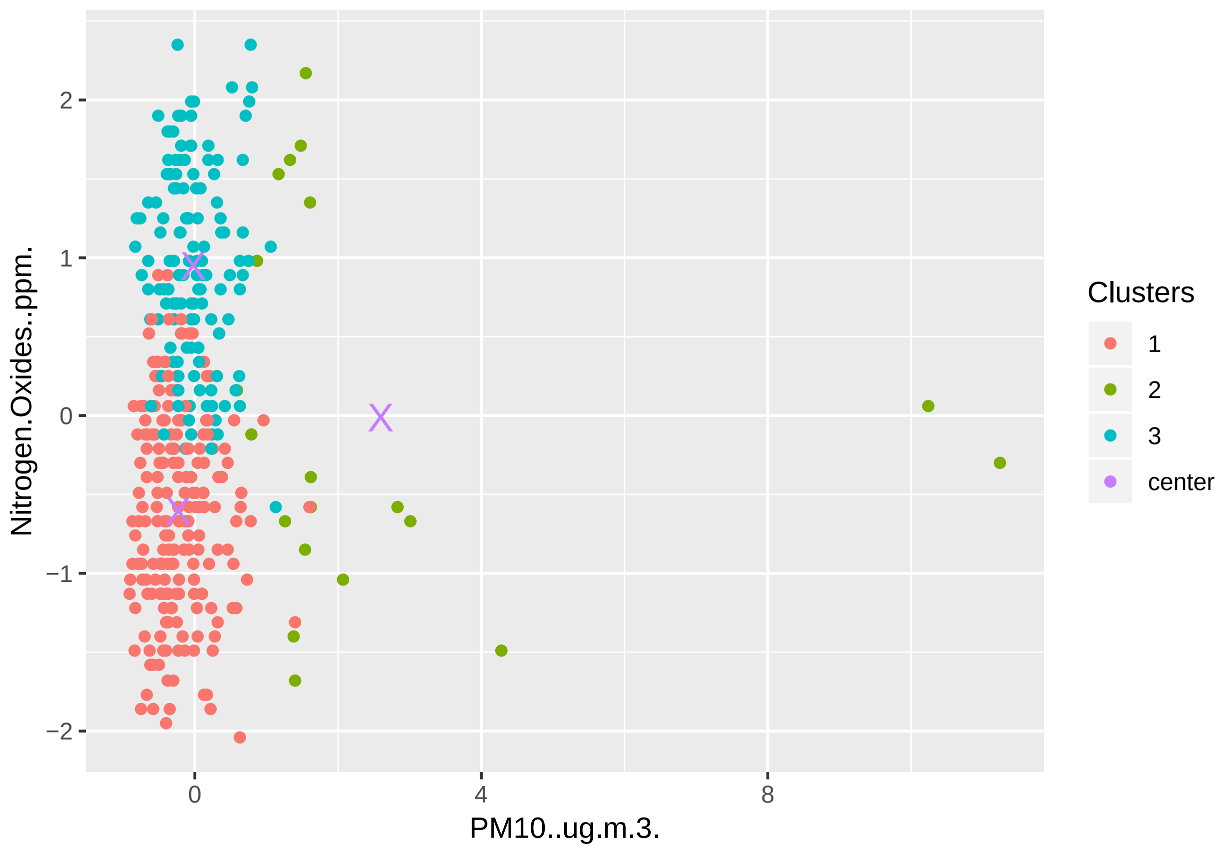



```r
# Load the iris data set into a new data frame
df <- brisbane.climateCLU_scaled
```

```r
# Add a variable to the data frame for the clustered data
df$Clusters <- factor(kmeans_result$cluster)

# Finally, extract the ceters for later use
centers <- as.data.frame(kmeans_result$centers)

# Let's now make some plots!
p1 <- ggplot(data=df, aes(x=PM10..ug.m.3., y=Carbon.Monoxide..ppm., color=Clusters)) + geom_point() + g
aes(x=PM10..ug.m.3., y=Carbon.Monoxide..ppm., color='center'), shape='X', size=5, show.legend=FALSE)
p2 <- ggplot(data=df, aes(x=PM10..ug.m.3., y=Nitrogen.Oxides..ppm., color=Clusters)) + geom_point() + g
p3 <- ggplot(data=df, aes(x=PM2.5..ug.m.3., y=Carbon.Monoxide..ppm., color=Clusters)) + geom_point() + g
p4 <- ggplot(data=df, aes(x=PM2.5..ug.m.3., y=Carbon.Monoxide..ppm., color=Clusters)) + geom_point() +
p1
```

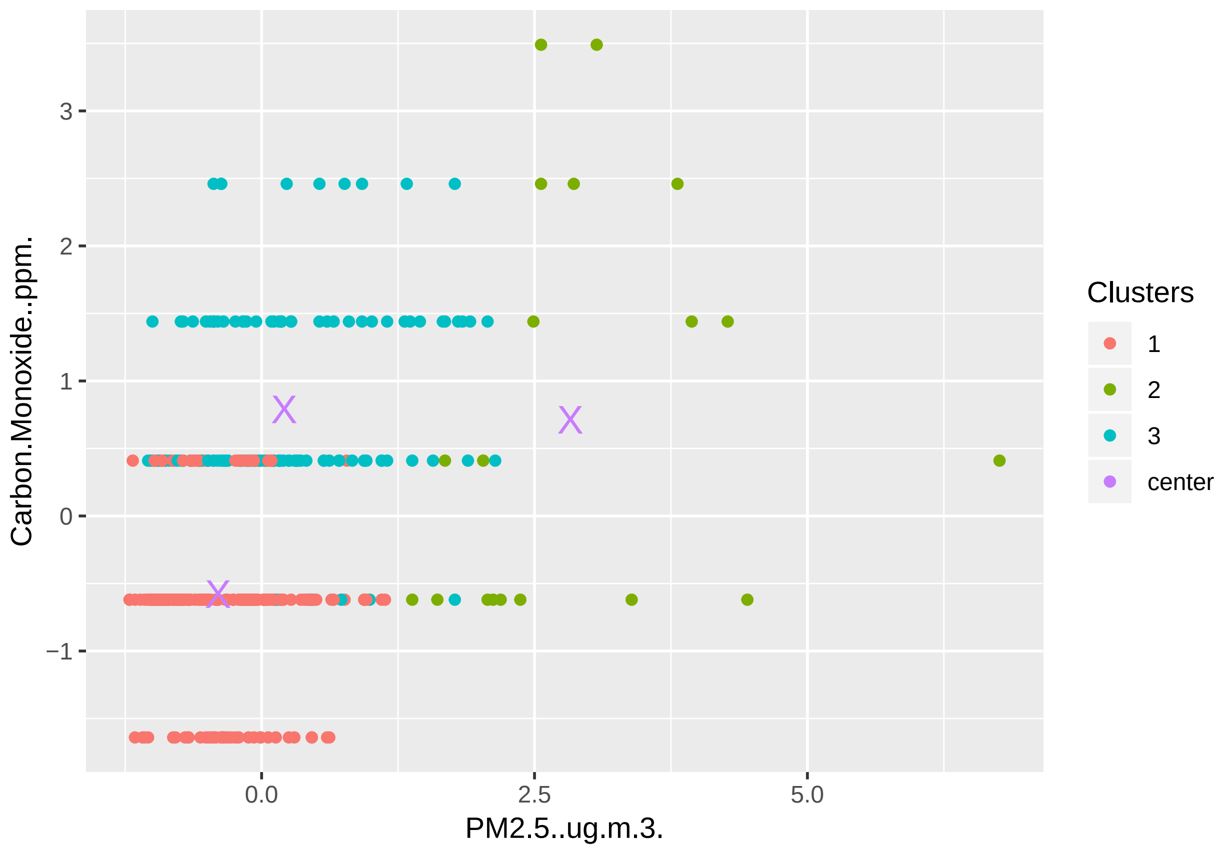



```r
p2
```



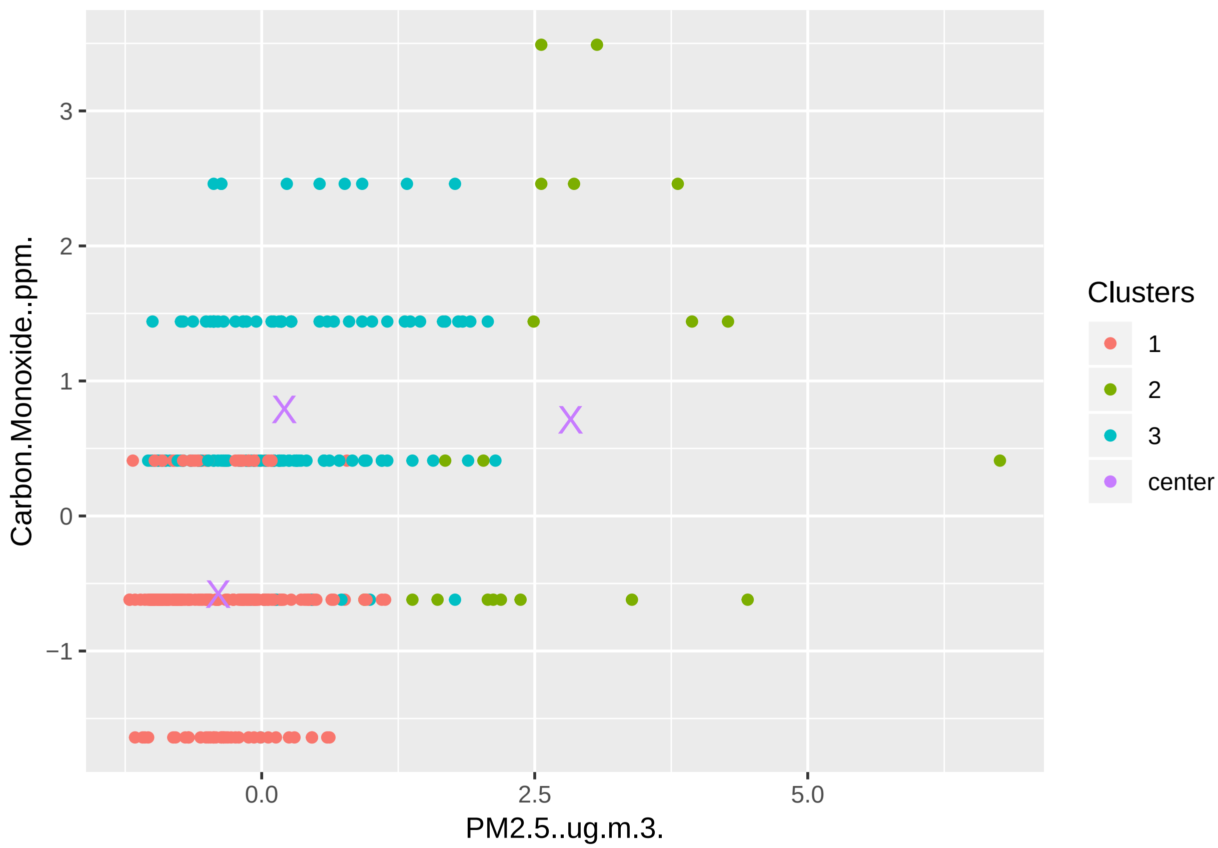
Nitrogen.Oxides..ppm.
2
1
0
−1
−2
0
4
8
PM10..ug.m.3.
Clusters
1
2
3
center


```
p3
```



Clusters
1
2
3
center
Carbon.Monoxide..ppm.
PM2.5..ug.m.3.
0.0
2.5
5.0
3
2
1
0
−1


```
p4
```




```r
library(DT)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
df$cluster <- as.factor(kmeans_result$cluster)

str(kmeans_result)
```

```
## List of 9
##  $ cluster     : int [1:365] 1 1 1 1 1 1 1 1 1 1 ...
```

```
## $ centers     : num [1:3, 1:4] -0.6035 -0.0095 0.9513 -0.5744 0.7165 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "1" "2" "3"
##   .. ..$ : chr [1:4] "Nitrogen.Oxides..ppm." "Carbon.Monoxide..ppm." "PM10..ug.m.3." "PM2.5..ug.m.3.
## $ totss       : num 1457
## $ withinss    : num [1:3] 262 264 223
## $ tot.withinss: num 749
## $ betweenss   : num 708
## $ size        : int [1:3] 211 20 134
## $ iter        : int 3
## $ ifault      : int 0
## - attr(*, "class")= chr "kmeans"
```

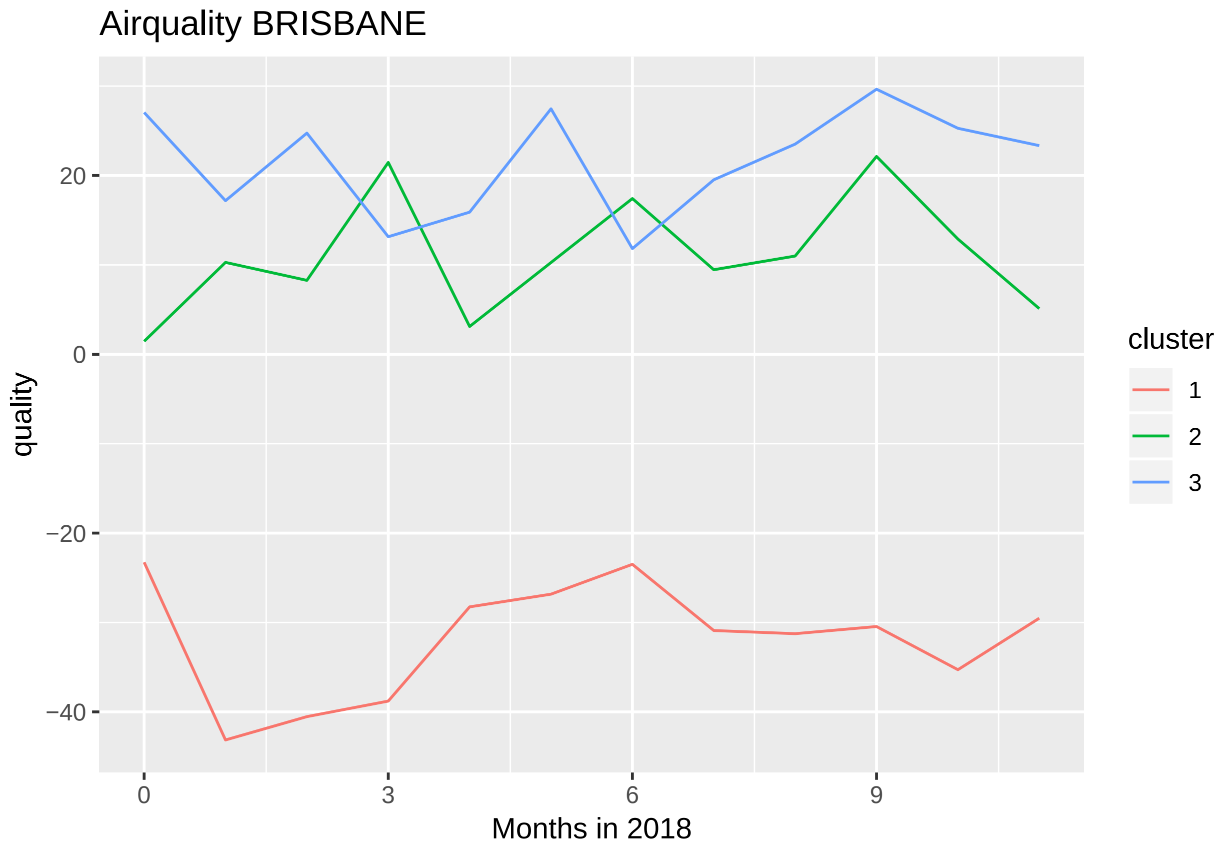
```r
df$Date <- round(brisbane.climateDay$Group.1%%12,0)


test1 <-df %>%
  group_by(Date,cluster) %>%
  summarise(quality= sum(c(Nitrogen.Oxides..ppm.,Carbon.Monoxide..ppm.,PM10..ug.m.3.,PM2.5..ug.m.3.)))

datatable(test1)
```

Show 10 entries Search:

	Date	cluster	quality
1	0	1	-23.26
2	0	2	1.45
3	0	3	27.05
4	1	1	-43.14
5	1	2	10.28
6	1	3	17.18
7	2	1	-40.53
8	2	2	8.27
9	2	3	24.74
10	3	1	-38.79

Showing 1 to 10 of 35 entries Previous 1 2 3 4 Next

```r
#Let's get a graphical view of the same...

monthly_growth <- test1 %>%
  ##mutate(Date = paste("04", Date)) %>%
  ggplot(aes(Date, quality, colour = cluster)) + geom_line() +
  ggtitle("Airquality BRISBANE") + xlab("Months in 2018")
monthly_growth
```


Airquality BRISBANE
20
0
−20
−40
quality
0
3
6
9
Months in 2018
cluster
1
2
3