Real-time environmental monitoring and weather prediction using Multiple Linear Regression, SVM and Naïve Bayes classification

Real time environmental monitoring refers to the real-time environmental data management system that measures, records, and analyzes data. The application is widely used to observe environmental change in order to make the necessary actions for making the world a better place. Weather prediction gives an insight on what the weather might be on the upcoming days by analyzing the past data. For the weather forecasting model, time series analysis using simple moving average, Arima and exponential smoothing is performed. Also, the weather prediction is performed by the Support Vector Machine and Naïve Bayes classification algorithm.

Procedure

1. Load and read the dataset
2. Pre-process the data and view its summary
3. Use the Simple Moving Average forecasting model and visualize the output
4. Use the Arima forecasting model and view the output
5. Use the Exponential smoothing forecasting model and visualize the output
6. Perform Exploratory Data Analysis using box plot and histograms
7. Extract the features of the dataset
8. Get the correlation between the numeric variables
9. Get the numeric and categorical variables
10. Remove the outliers
11. Cleanse and normalize the data
12. Convert the class to predict into numeric data
13. Split the dataset into training and testing in the ratio of 75:25.
14. Perform Multiple linear regression and view the results in confusion matrix
15. Construct the ROCR curve
16. Perform SVM and view the results
17. Perform Naïve Bayes and view the results

Program Execution

1. Install the required packages

In [68]:

```
install.packages('TTR')
install.packages('caret')
install.packages('outliers')
install.packages('caTools')
install.packages('ROCR')
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

2. Create function to get mode of categorical data and for removing outliers

In [10]:

```
#Data preprocessing
#Function to get mode of categorical data
#usage of functions
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}#Function for removing outliers
out.rem<-function(x) {
  x[which(x==outlier(x))]=NA
  x
}
```

3. Load the dataset and pre-process the data

In [11]:

```
set.seed(1023)
weather_data <- read.csv("/content/weatherAUS.csv", header = TRUE, sep = ",",
is.numeric(weather_data$WindGustDir)
weather_data2 <- subset(weather_data, select = -c(Date, Location, Rainfall, R
colnames(weather_data2)
weather_data3 <- weather_data2[complete.cases(weather_data2),]
summary(weather_data2)
weather_data2
```

FALSE

'MinTemp' · 'MaxTemp' · 'Evaporation' · 'Sunshine' · 'WindGustDir' · 'WindGustSpeed' · 'WindDir9am' · 'WindDir3pm' · 'WindSpeed9am' · 'WindSpeed3pm' · 'Humidity9am' · 'Humidity3pm' · 'Pressure9am' · 'Pressure3pm' · 'Cloud9am' · 'Cloud3pm' · 'Temp9am' · 'Temp3pm' · 'RainTomorrow'

```
    MinTemp          MaxTemp       Evaporation        Sunshine     
 Min.   :-8.50   Min.   :-4.80   Min.   :  0.00   Min.   : 0.00  
 1st Qu.: 7.60   1st Qu.:17.90   1st Qu.:  2.60   1st Qu.: 4.80  
 Median :12.00   Median :22.60   Median :  4.80   Median : 8.40  
 Mean   :12.19   Mean   :23.22   Mean   :  5.47   Mean   : 7.61  
 3rd Qu.:16.90   3rd Qu.:28.20   3rd Qu.:  7.40   3rd Qu.:10.60  
 Max.   :33.90   Max.   :48.10   Max.   :145.00   Max.   :14.50  
 NA's   :1485    NA's   :1261    NA's   :62790    NA's   :69835  
  WindGustDir    WindGustSpeed      WindDir9am      WindDir3pm    
 W      : 9915   Min.   :  6.00   N      :11758   SE     :10838  
 SE     : 9418   1st Qu.: 31.00   SE     : 9287   W      :10110  
 N      : 9313   Median : 39.00   E      : 9176   S      : 9926  
 SSE    : 9216   Mean   : 40.03   SSE    : 9112   WSW    : 9518  
 E      : 9181   3rd Qu.: 48.00   NW     : 8749   SSE    : 9399  
 (Other):88091   Max.   :135.00   (Other):86812   (Other):91441  
 NA's   :10326   NA's   :10263    NA's   :10566   NA's   : 4228  
  WindSpeed9am     WindSpeed3pm    Humidity9am      Humidity3pm    
 Min.   :  0.00   Min.   : 0.00   Min.   :  0.00   Min.   :  0.00  
 1st Qu.:  7.00   1st Qu.:13.00   1st Qu.: 57.00   1st Qu.: 37.00  
 Median : 13.00   Median :19.00   Median : 70.00   Median : 52.00  
```

```
Mean   : 14.04   Mean   :18.66   Mean   : 68.88   Mean   : 51.54
3rd Qu.: 19.00   3rd Qu.:24.00   3rd Qu.: 83.00   3rd Qu.: 66.00
Max.   :130.00   Max.   :87.00   Max.   :100.00   Max.   :100.00
NA's   :1767     NA's   :3062    NA's   :2654     NA's   :4507
 Pressure9am      Pressure3pm        Cloud9am        Cloud3pm
Min.   : 980.5   Min.   : 977.1   Min.   :0.00    Min.   :0.00
1st Qu.:1012.9   1st Qu.:1010.4   1st Qu.:1.00    1st Qu.:2.00
Median :1017.6   Median :1015.2   Median :5.00    Median :5.00
Mean   :1017.6   Mean   :1015.3   Mean   :4.45    Mean   :4.51
3rd Qu.:1022.4   3rd Qu.:1020.0   3rd Qu.:7.00    3rd Qu.:7.00
Max.   :1041.0   Max.   :1039.6   Max.   :9.00    Max.   :9.00
NA's   :15065    NA's   :15028    NA's   :55888   NA's   :59358
   Temp9am         Temp3pm      RainTomorrow
Min.   :-7.20   Min.   :-5.40   No  :110316
1st Qu.:12.30   1st Qu.:16.60   Yes : 31877
Median :16.70   Median :21.10   NA's:  3267
Mean   :16.99   Mean   :21.68
3rd Qu.:21.60   3rd Qu.:26.40
Max.   :40.20   Max.   :46.70
NA's   :1767    NA's   :3609
```

	MinTemp	MaxTemp	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am
	<dbl>	<dbl>	<dbl>	<dbl>	<fct>	<int>	<fct>
1	13.4	22.9	NA	NA	W	44	W
2	7.4	25.1	NA	NA	WNW	44	NNW
3	12.9	25.7	NA	NA	WSW	46	W
4	9.2	28.0	NA	NA	NE	24	SE
5	17.5	32.3	NA	NA	W	41	ENE
6	14.6	29.7	NA	NA	WNW	56	W
7	14.3	25.0	NA	NA	W	50	SW
8	7.7	26.7	NA	NA	W	35	SSE
9	9.7	31.9	NA	NA	NNW	80	SE
10	13.1	30.1	NA	NA	W	28	S
11	13.4	30.4	NA	NA	N	30	SSE
12	15.9	21.7	NA	NA	NNE	31	NE
13	15.9	18.6	NA	NA	W	61	NNW
14	12.6	21.0	NA	NA	SW	44	W
15	8.4	24.6	NA	NA	NA	NA	S
16	9.8	27.7	NA	NA	WNW	50	NA
17	14.1	20.9	NA	NA	ENE	22	SSW
18	13.5	22.9	NA	NA	W	63	N
19	11.2	22.5	NA	NA	SSE	43	WSW
20	9.8	25.6	NA	NA	SSE	26	SE
21	11.5	29.3	NA	NA	S	24	SE
22	17.1	33.0	NA	NA	NE	43	NE
23	20.5	31.8	NA	NA	WNW	41	W
24	15.3	30.9	NA	NA	N	33	ESE

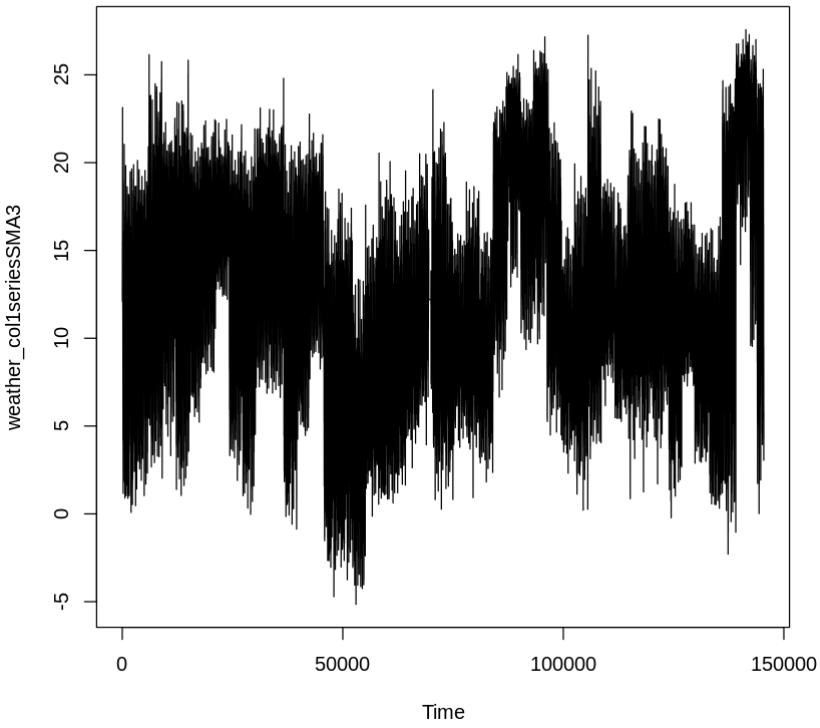
	MinTemp	MaxTemp	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am
	<dbl>	<dbl>	<dbl>	<dbl>	<fct>	<int>	<fct>
25	12.6	32.4	NA	NA	W	43	E
26	16.2	33.9	NA	NA	WSW	35	SE
27	16.9	33.0	NA	NA	WSW	57	NA
28	20.1	32.7	NA	NA	WNW	48	N
29	19.7	27.2	NA	NA	WNW	46	NW
30	12.5	24.2	NA	NA	WNW	50	WSW
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
145431	9.3	28.0	NA	NA	W	28	SSW
145432	8.0	24.6	NA	NA	E	33	SE
145433	12.7	22.2	NA	NA	E	37	E
145434	9.4	22.7	NA	NA	E	35	SE
145435	5.4	20.5	NA	NA	E	46	E
145436	5.6	19.4	NA	NA	E	43	E
145437	1.5	19.2	NA	NA	E	37	ESE
145438	3.8	19.8	NA	NA	ENE	39	ESE
145439	4.5	18.8	NA	NA	E	31	ESE
145440	4.9	20.7	NA	NA	E	37	ESE
145441	1.2	20.8	NA	NA	E	43	ESE
145442	0.5	21.7	NA	NA	E	43	ESE
145443	4.0	20.0	NA	NA	SE	39	ESE
145444	3.4	19.8	NA	NA	E	43	ESE
145445	3.2	18.6	NA	NA	E	46	ESE
145446	2.4	19.1	NA	NA	E	33	SE
145447	5.1	19.7	NA	NA	E	46	ESE
145448	2.5	21.2	NA	NA	ENE	35	ESE
145449	2.3	21.4	NA	NA	SE	22	SE
145450	2.6	22.5	NA	NA	S	19	S
145451	5.2	24.3	NA	NA	E	24	SE
145452	6.4	23.4	NA	NA	ESE	31	S
145453	8.0	20.7	NA	NA	ESE	41	SE
145454	7.4	20.6	NA	NA	E	35	ESE
145455	3.5	21.8	NA	NA	E	31	ESE
145456	2.8	23.4	NA	NA	E	31	SE
145457	3.6	25.3	NA	NA	NNW	22	SE
145458	5.4	26.9	NA	NA	N	37	SE
145459	7.8	27.0	NA	NA	SE	28	SSE
145460	14.9	NA	NA	NA	NA	NA	ESE

4. Use the Simple Moving Average forecasting model and visualize the output

In [69]:
```
#SMA - Simple Moving Average
library("TTR")
library("data.table")

weather_col1 <- fread("/content/weatherAUS.csv",
                      select = c("MinTemp"))
weather_col1series <- ts(weather_col1,frequency=12, start=c(2015,1))
```

In [48]:
```
weather_col1series[is.na(weather_col1series)]<-mean(weather_col1series,na.rm=
```

In [49]:
```
weather_col1seriesSMA3 <- SMA(weather_col1series,n=12)
plot.ts(weather_col1seriesSMA3)
```



5. Use the Exponential smoothing forecasting model and visualize the output

In [58]:
```
#Forecast using Exponential smoothing
weather_col1 <- fread("/content/weatherAUS.csv",
                      select = c("MinTemp"))
weather_col1series <- ts(weather_col1,frequency=12, start=c(2015,1))
weather_col1series[is.na(weather_col1series)]<-mean(weather_col1series,na.rm=
```

In [63]:
```
weather_col1seriesforecasts <- HoltWinters(weather_col1series, beta=FALSE, ga
weather_col1seriesforecasts
```

```
Holt-Winters exponential smoothing without trend and without seasonal compone
nt.

Call:
HoltWinters(x = weather_col1series, beta = FALSE, gamma = FALSE)
```

```
Smoothing parameters:
 alpha: 0.5329318
 beta : FALSE
 gamma: FALSE

Coefficients:
      [,1]
a 10.89158
```

In [65]:
```
weather_col1seriesforecasts$SSE
```

1269896.20993687

In [66]:
```
HoltWinters(weather_col1series, beta=FALSE, gamma=FALSE, l.start=23.56)
```

```
Holt-Winters exponential smoothing without trend and without seasonal compone
nt.

Call:
HoltWinters(x = weather_col1series, beta = FALSE, gamma = FALSE,     l.start
= 23.56)

Smoothing parameters:
 alpha: 0.533059
 beta : FALSE
 gamma: FALSE

Coefficients:
      [,1]
a 10.89288
```

6. Perform Exploratory Data Analysis using box plot and histograms

In [52]:
```
# Arima Model

library("TTR")
v1 <- weather_data[[4]]
weather_datats <- ts(v1)



## partition into train and test
train_series=weather_datats[1:40]
test_series=weather_datats[41:50]

## make arima models
arimaModel_1=arima(train_series, order=c(0,1,2))
arimaModel_2=arima(train_series, order=c(1,1,0))
arimaModel_3=arima(train_series, order=c(1,1,2))

## look at the parameters
print(arimaModel_1);print(arimaModel_2);print(arimaModel_3)

forecast1=predict(arimaModel_1, 10)
forecast2=predict(arimaModel_2, 10)
forecast3=predict(arimaModel_3, 10)

forecast1
forecast2
forecast3
```

```
Call:
arima(x = train_series, order = c(0, 1, 2))

Coefficients:
         ma1      ma2
      0.0073  -0.2757
s.e.  0.2119   0.2592

sigma^2 estimated as 12.86:  log likelihood = -105.23,  aic = 216.45

Call:
arima(x = train_series, order = c(1, 1, 0))

Coefficients:
         ar1
      0.1014
s.e.  0.1579

sigma^2 estimated as 13.37:  log likelihood = -105.91,  aic = 215.82

Call:
arima(x = train_series, order = c(1, 1, 2))

Coefficients:
         ar1      ma1      ma2
      0.5774  -0.6371  -0.3031
s.e.  0.2116   0.2446   0.1817

sigma^2 estimated as 11.24:  log likelihood = -103.07,  aic = 214.15
```

$pred A Time Series:

31.0688118658979 · 30.892753617835 · 30.892753617835 · 30.892753617835 · 30.892753617835 · 30.892753617835 · 30.892753617835 · 30.892753617835 · 30.892753617835 · 30.892753617835

$se A Time Series:

3.5862728062742 · 5.09033904380461 · 5.7267560181389 · 6.2991997429593 · 6.82378929276886 · 7.3108332782601 · 7.76739764225598 · 8.19857603754044 · 8.60818401021613 · 8.99916745394534

$pred A Time Series:

28.4101372254683 · 28.4111648588702 · 28.4112690323852 · 28.4112795926893 · 28.4112806632111 · 28.4112807717323 · 28.4112807827334 · 28.4112807838486 · 28.4112807839616 · 28.4112807839731

$se A Time Series:

3.65655592547888 · 5.43957504039026 · 6.79055360121409 · 7.91613803073529 · 8.90067450920703 · 9.7866792464736 · 10.5988776255277 · 11.3531196788409 · 12.0602838397408 · 12.7282192855789

$pred A Time Series:

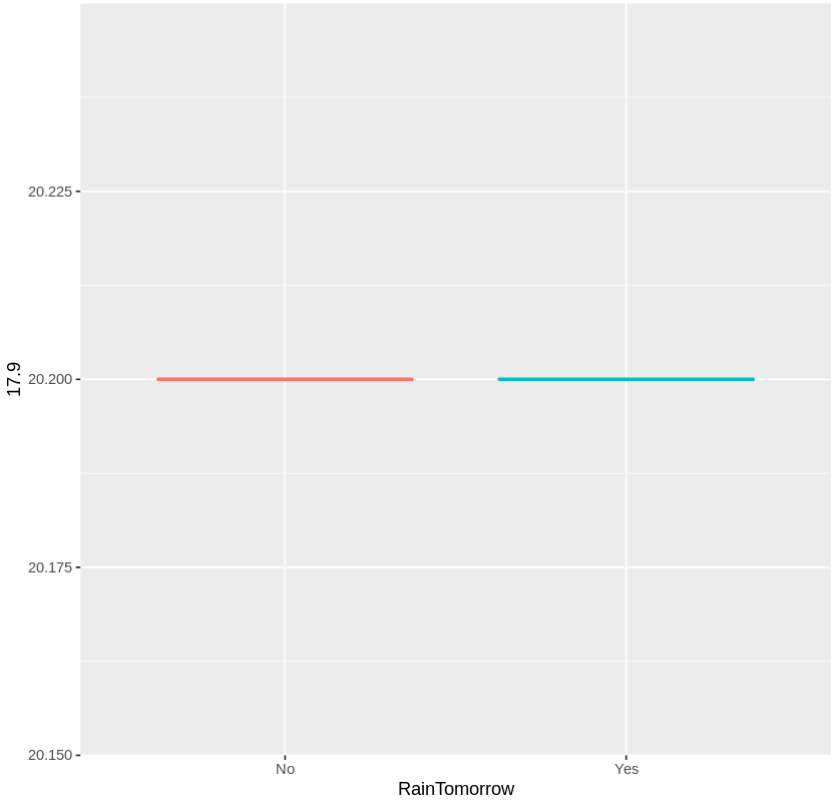
29.3394759164271 · 29.185946542968 · 29.0973051564728 · 29.046127357575 · 29.0165794544613 · 28.9995197411243 · 28.9896701819857 · 28.9839834618569 · 28.9807001894415 · 28.9788045663917

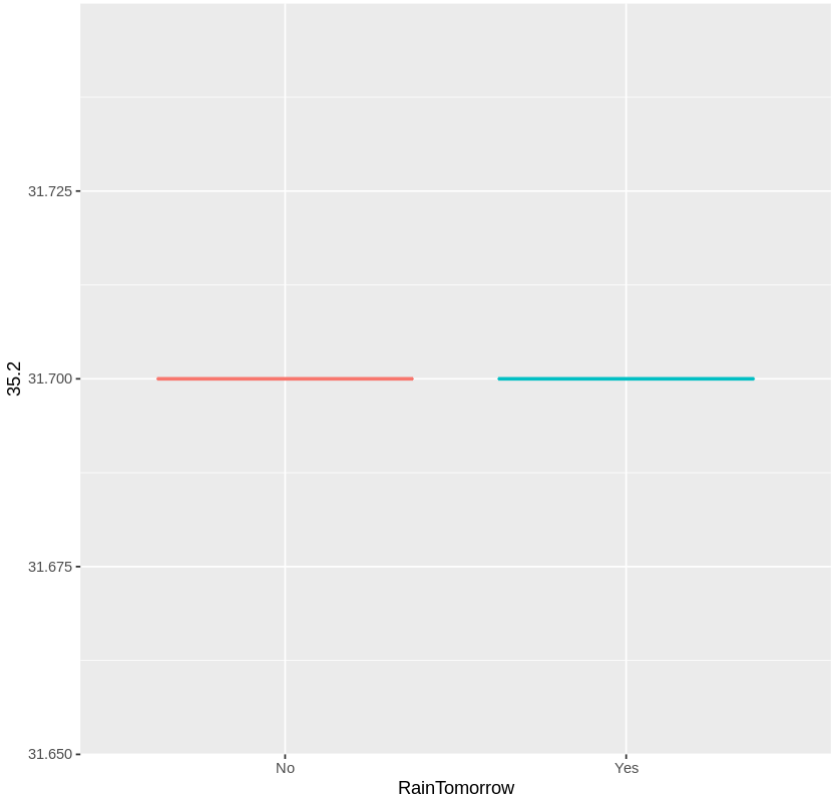
$se A Time Series:
3.35500838810451 · 4.60862932041385 · 5.03611449022357 · 5.22104565919146 · 5.31570700089267 · 5.37247357244169 · 5.41179322008997 · 5.44245791758936 · 5.4685742425329 · 5.49219452878841

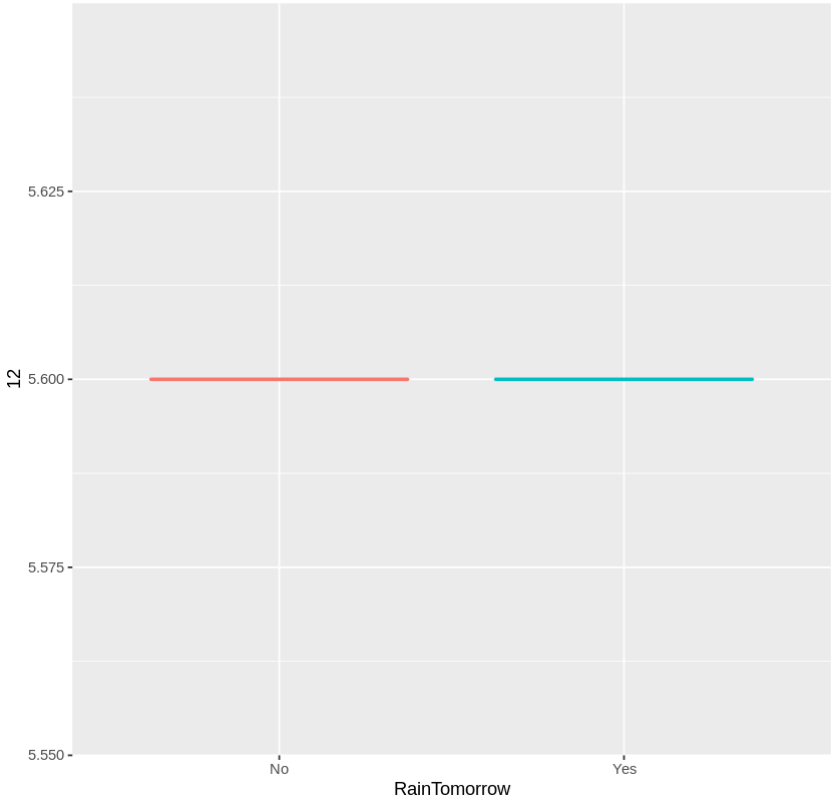
7. Exploratory Data Analysis(EDA) using box plot and histogram

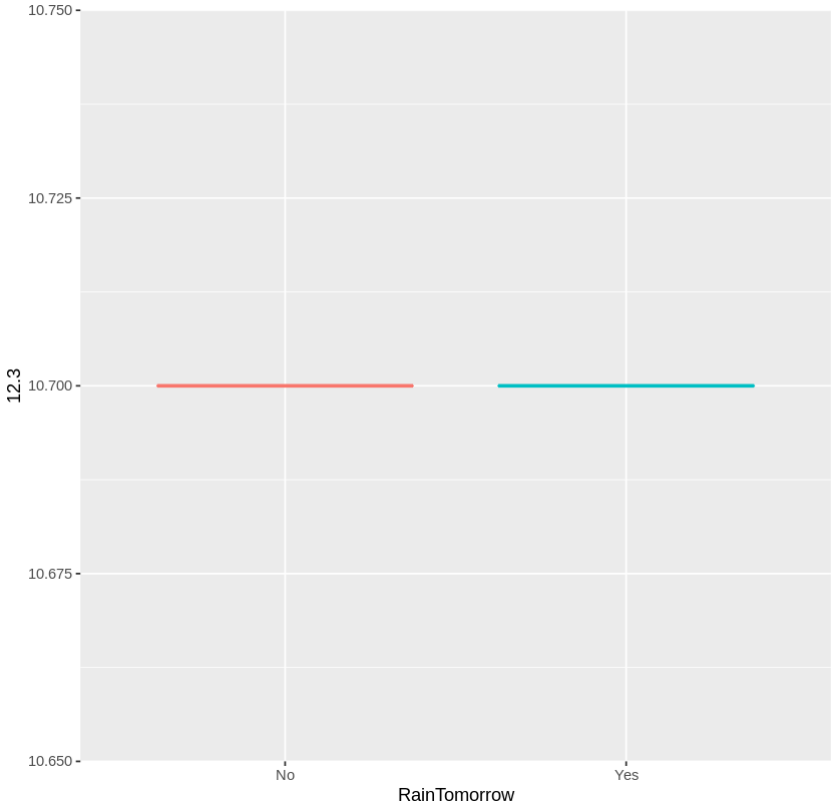
In [70]:

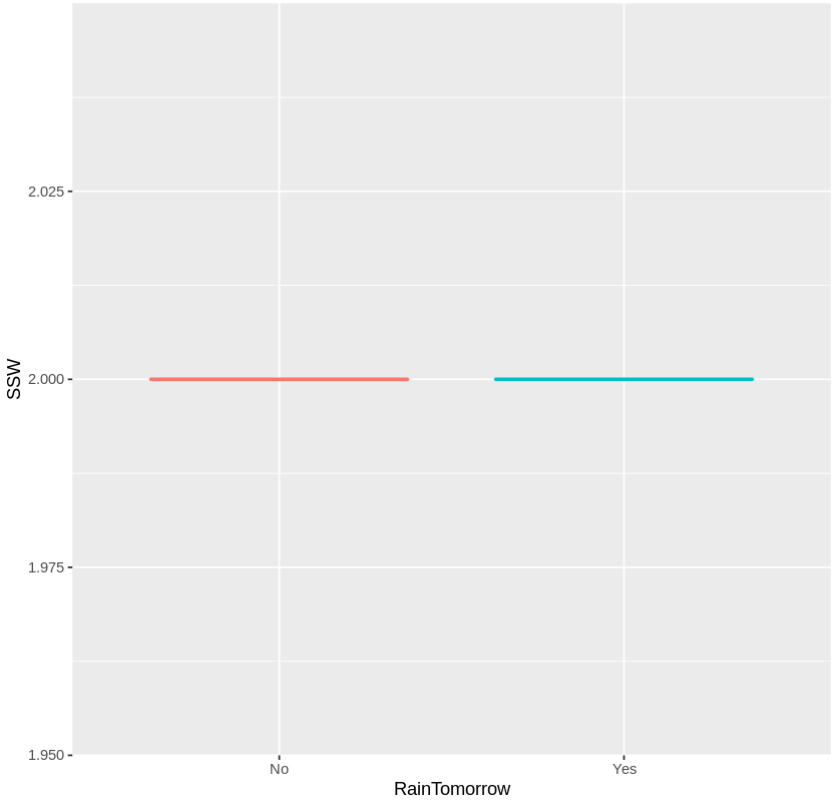
```
##1. Exploratory Data Analysis(EDA)
#get the boxplots
library(ggplot2)
gp <- invisible(lapply(weather_data3, function(x) {
  ggplot(data=weather_data3, aes(x= RainTomorrow, y=eval(parse(text=x)), col
gp[[1]]
gp[[2]]
gp[[3]]
gp[[4]]
gp[[5]]
gp[[6]]
gp[[7]]
gp[[8]]
gp[[9]]
gp[[10]]
gp[[11]]
gp[[12]]
gp[[13]]
gp[[14]]
#histograms
#Check the skewness of data
hist(weather_data$MinTemp)
hist(weather_data$MaxTemp)
hist(weather_data$Evaporation)
hist(weather_data$Sunshine)
hist(weather_data$WindGustSpeed)
hist(weather_data$WindSpeed9am)
hist(weather_data$WindSpeed3pm)
hist(weather_data$Humidity9am)
hist(weather_data$Humidity3pm)
hist(weather_data$Temp9am)
hist(weather_data$Temp3pm)
hist(weather_data$Pressure9am)
hist(weather_data$Pressure3pm)
```

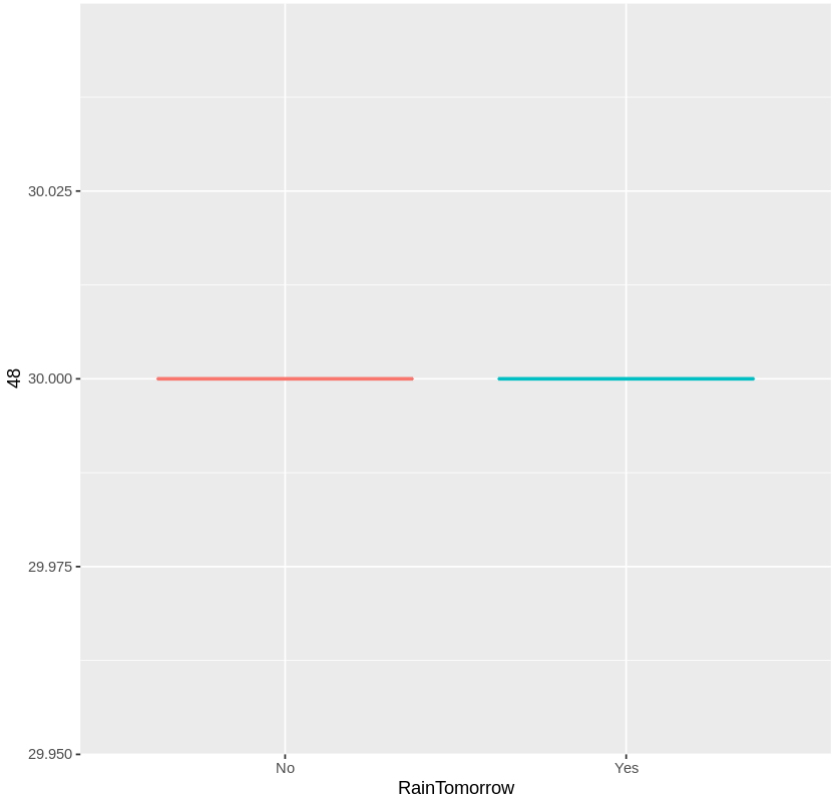

17.9
20.225
20.200
20.175
20.150
No
Yes
RainTomorrow

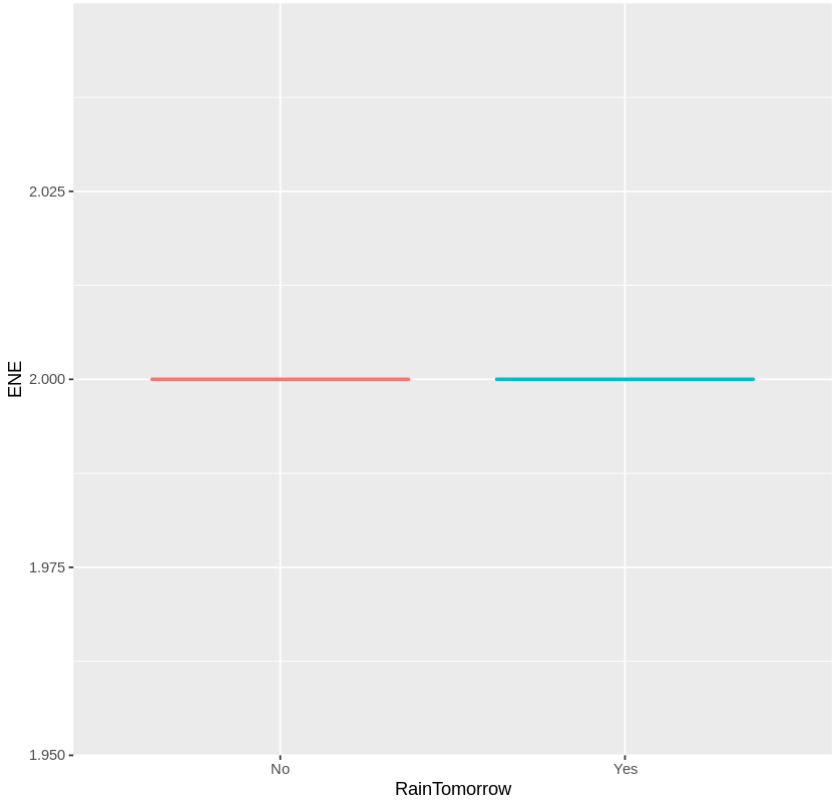

35.2
31.725
31.700
31.675
31.650
No
Yes
RainTomorrow

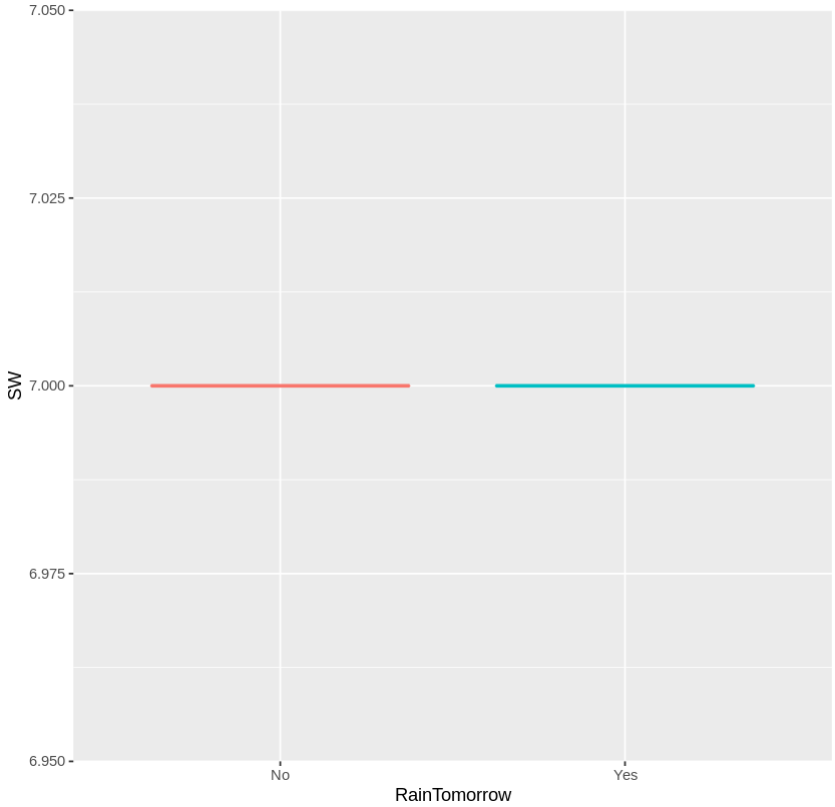

12
5.625
5.600
5.575
5.550
No
Yes
RainTomorrow

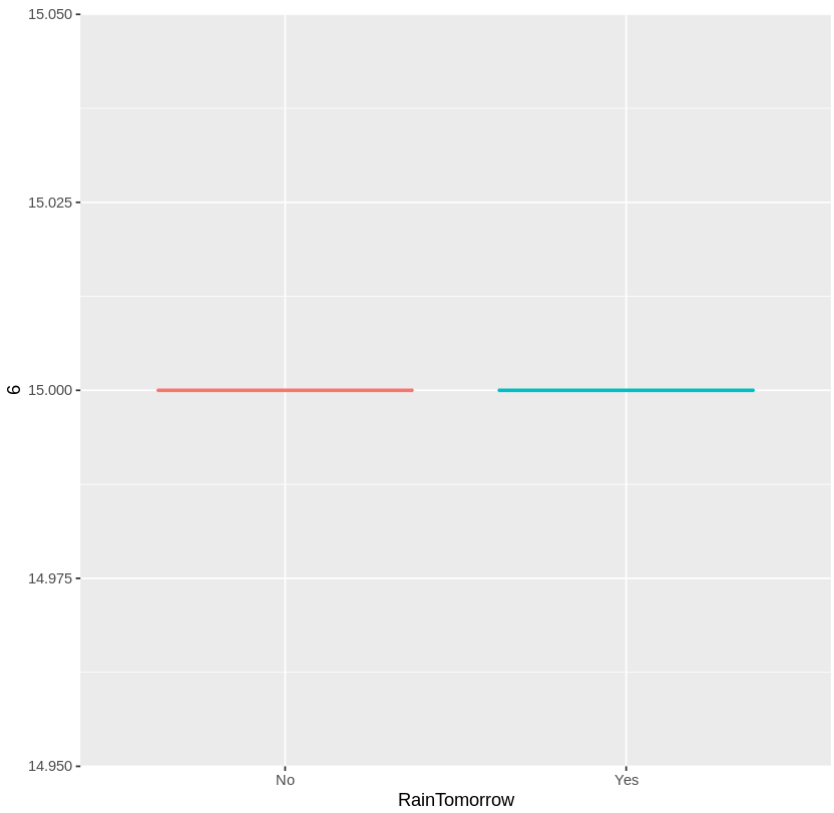

12.3
10.750
10.725
10.700
10.675
10.650
No
Yes
RainTomorrow

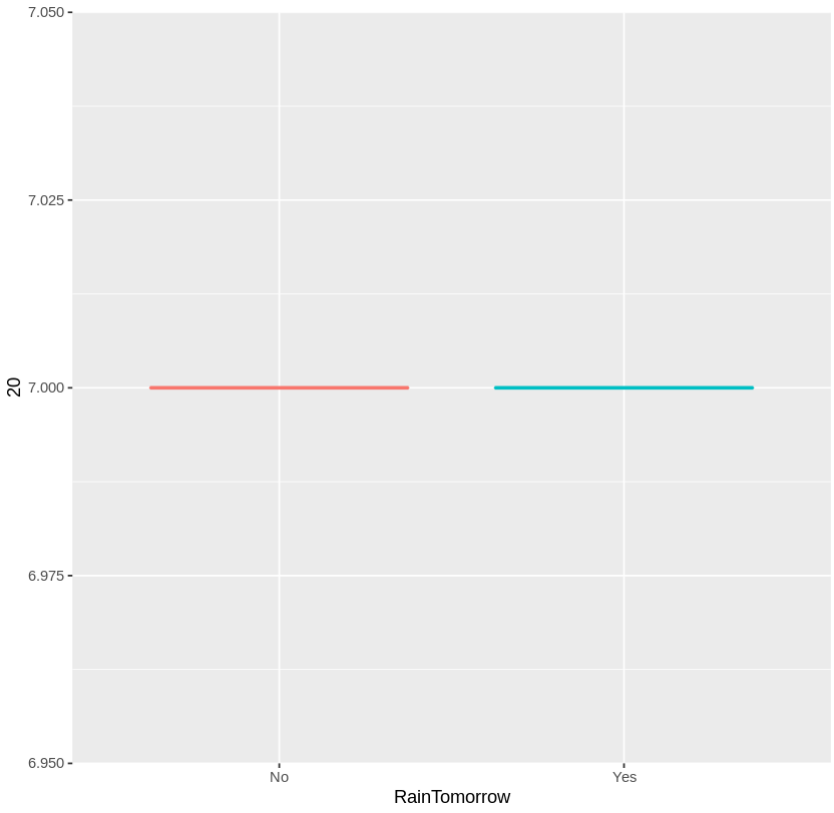

SSW
2.025
2.000
1.975
1.950
No
Yes
RainTomorrow

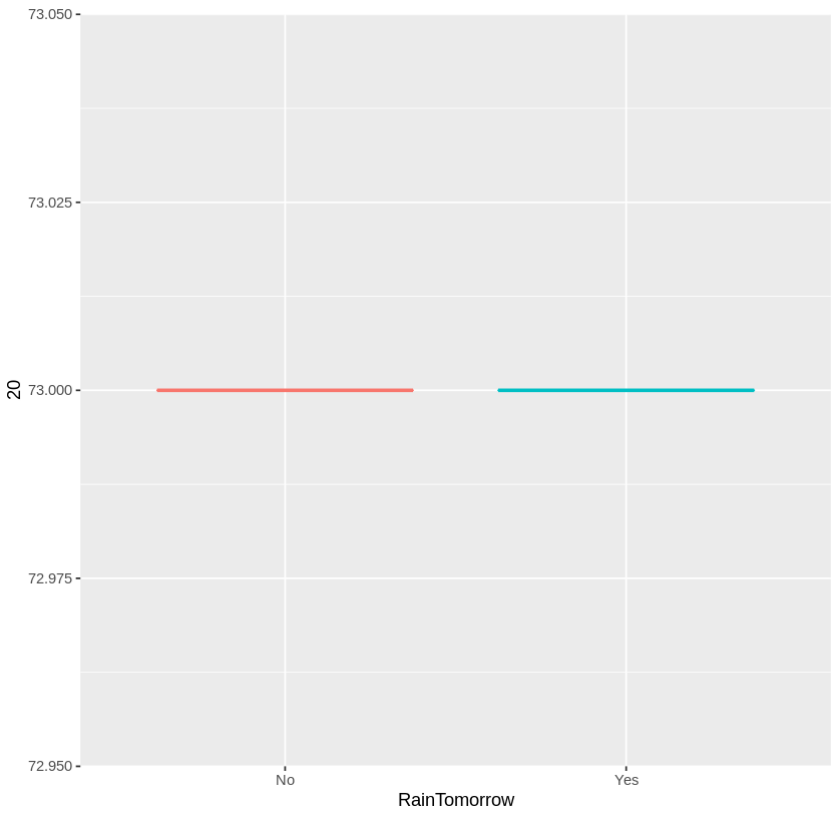

48
30.025
30.000
29.975
29.950
No
Yes
RainTomorrow

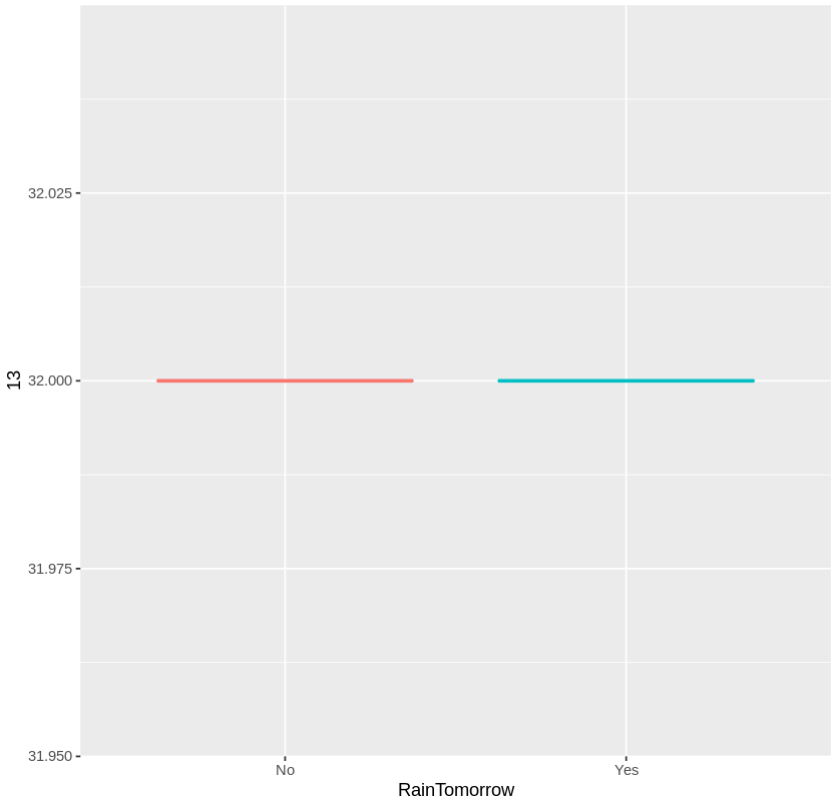

2.025
2.000
1.975
1.950
ENE
No
Yes
RainTomorrow

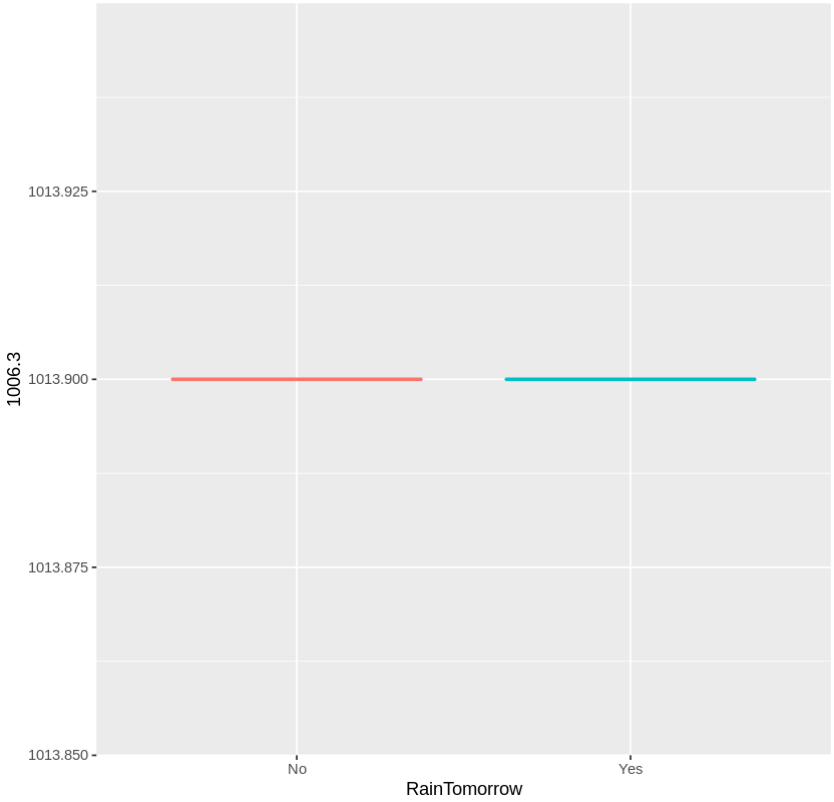

7.050
7.025
7.000
6.975
6.950
SW
No
Yes
RainTomorrow

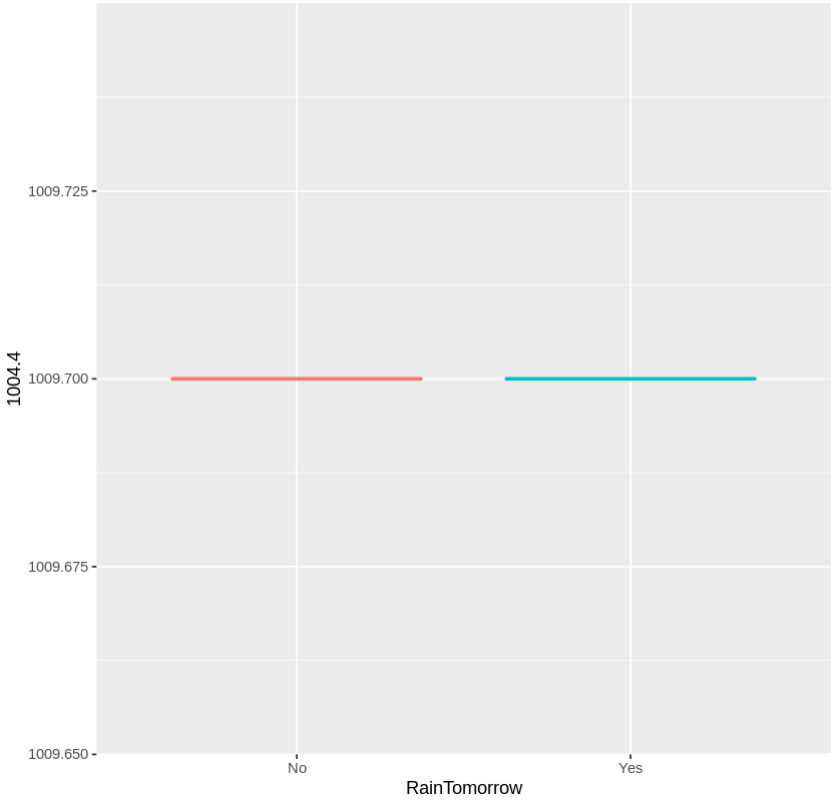

15.050
15.025
15.000
14.975
14.950
6
No
Yes
RainTomorrow

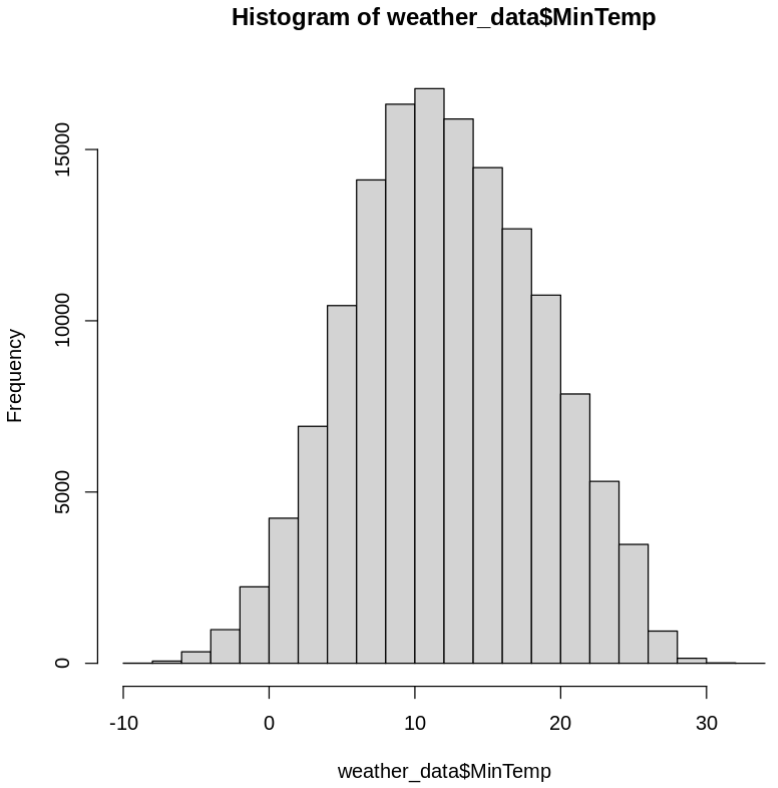

7.050
7.025
7.000
6.975
6.950
20
No
Yes
RainTomorrow

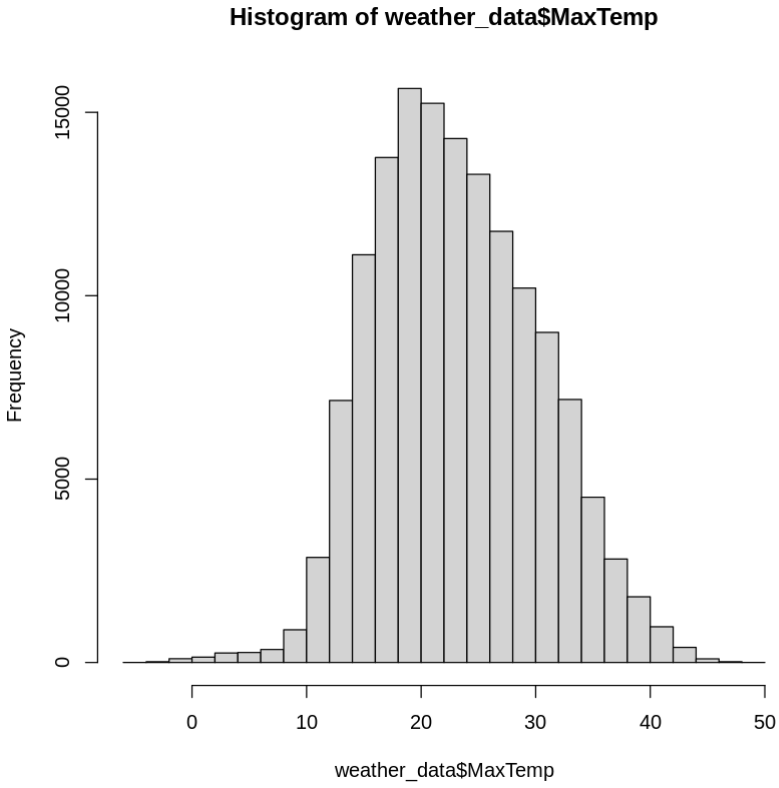

73.050
73.025
20
73.000
72.975
72.950
No
Yes
RainTomorrow

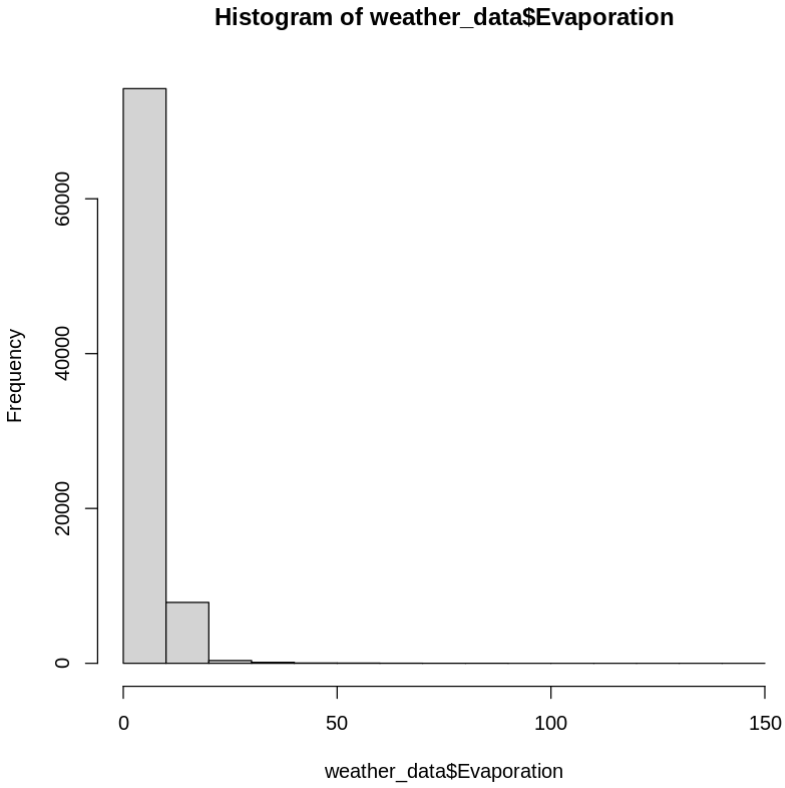

32.025
13
32.000
31.975
31.950
No
Yes
RainTomorrow

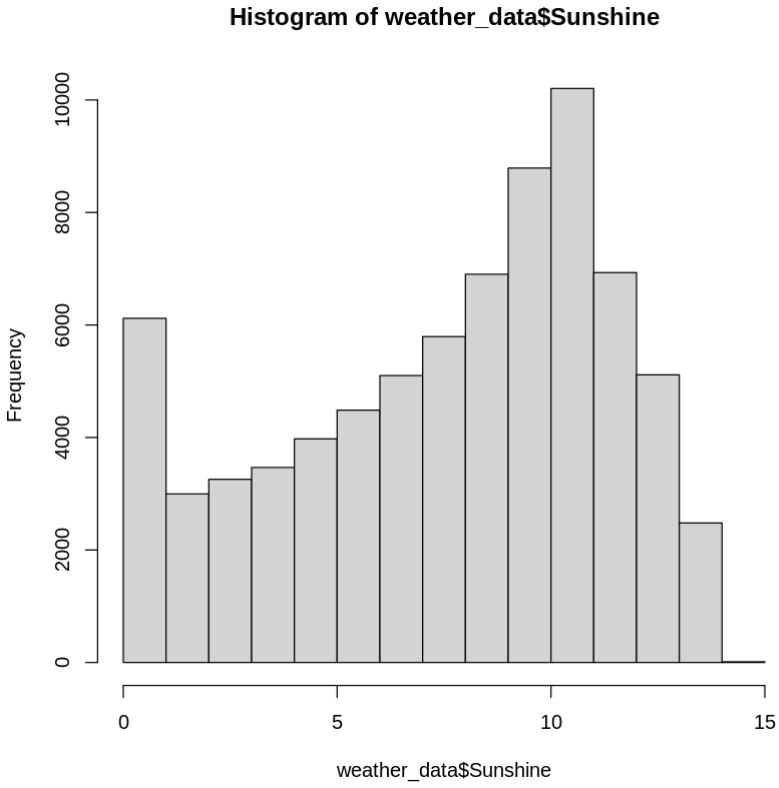

1006.3
1013.925
1013.900
1013.875
1013.850
No
Yes
RainTomorrow

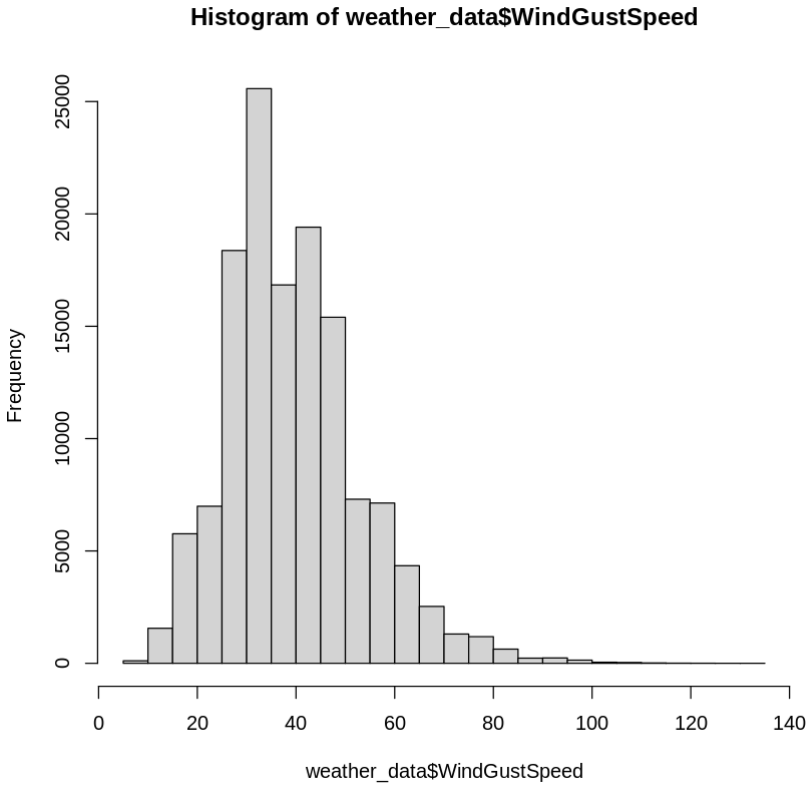

1004.4
1009.725
1009.700
1009.675
1009.650
No
Yes
RainTomorrow

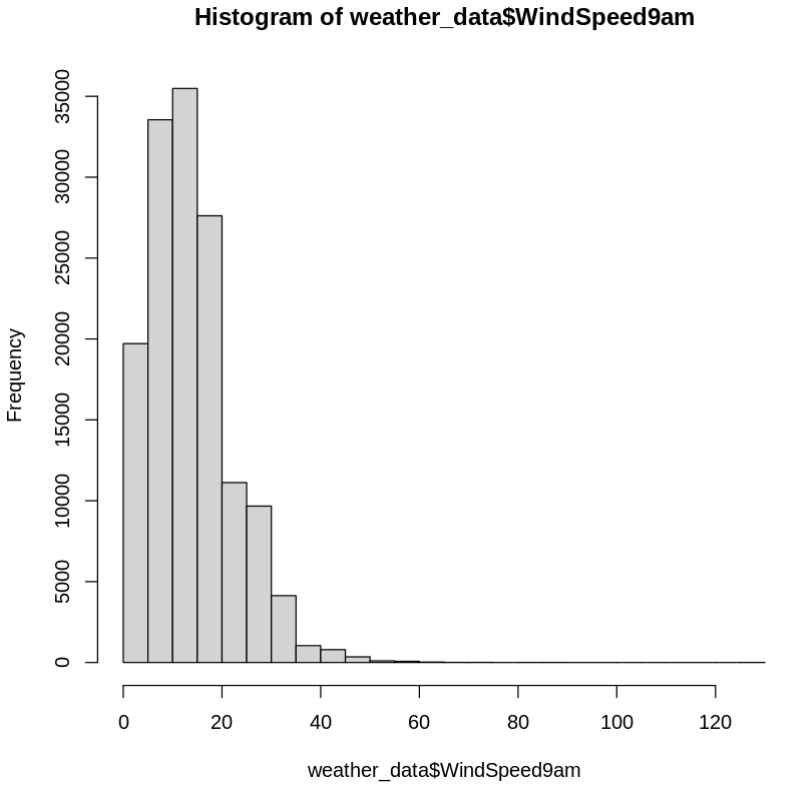

Histogram of weather_data$MinTemp
Frequency
0
5000
10000
15000
-10
0
10
20
30
weather_data$MinTemp

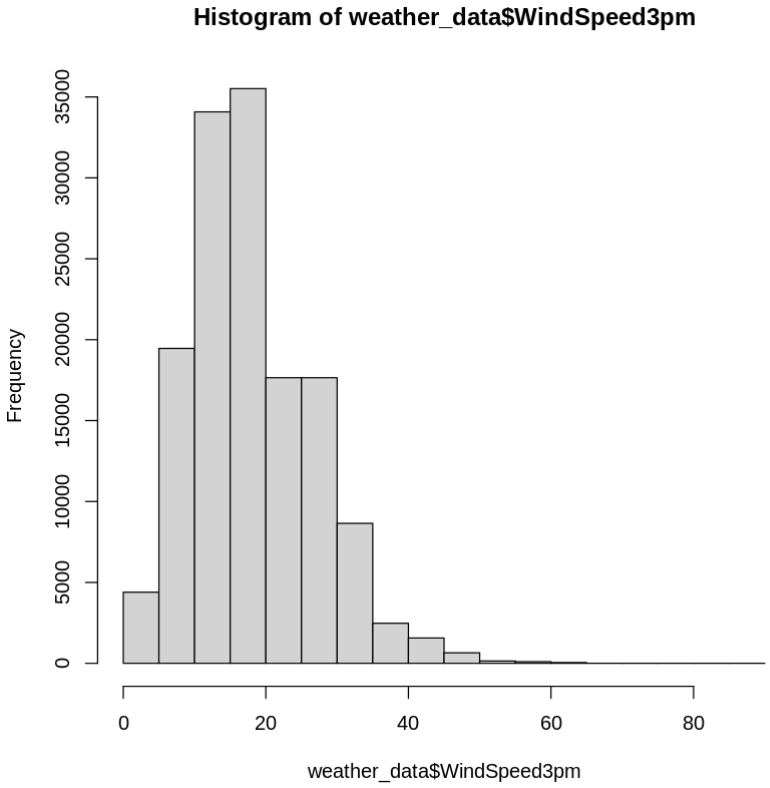

Histogram of weather_data$MaxTemp
Frequency
0
5000
10000
15000
0
10
20
30
40
50
weather_data$MaxTemp

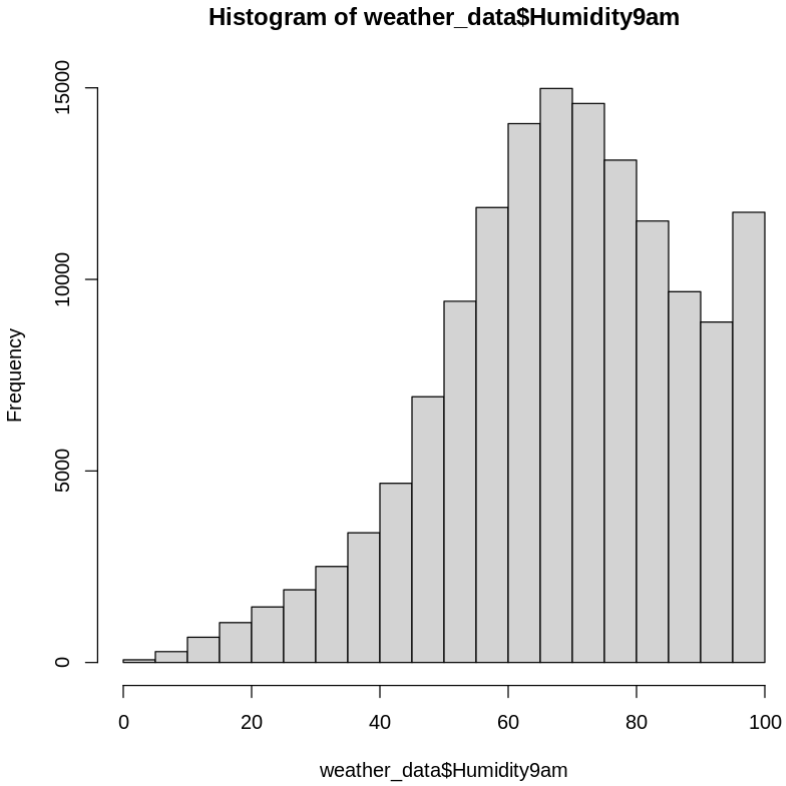

Histogram of weather_data$Evaporation
Frequency
0
20000
40000
60000
0
50
100
150
weather_data$Evaporation

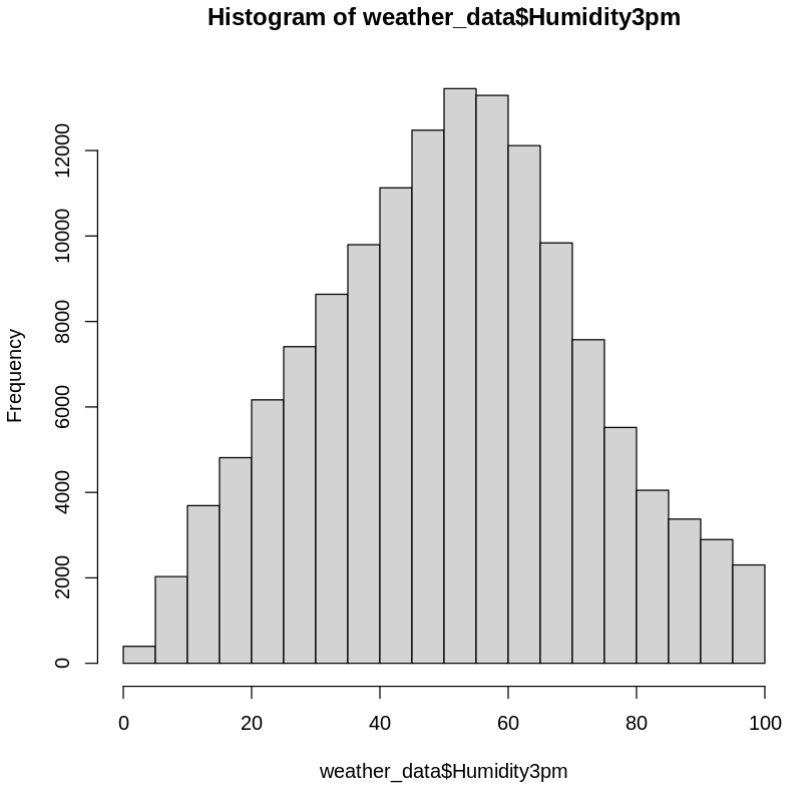

Histogram of weather_data$Sunshine
Frequency
0
2000
4000
6000
8000
10000
0
5
10
15
weather_data$Sunshine

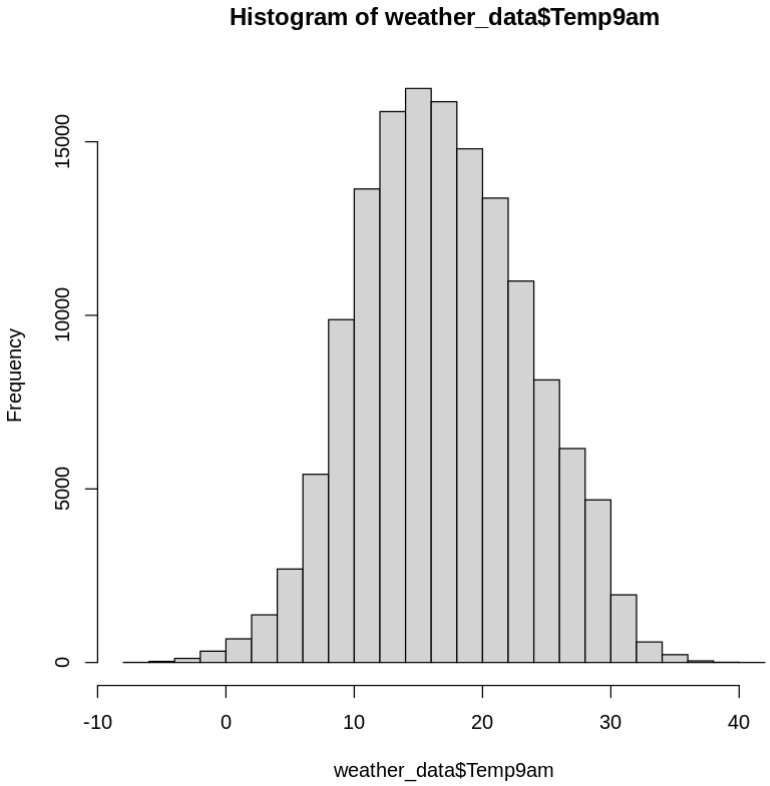

Histogram of weather_data$WindGustSpeed
Frequency
0
5000
10000
15000
20000
25000
0
20
40
60
80
100
120
140
weather_data$WindGustSpeed

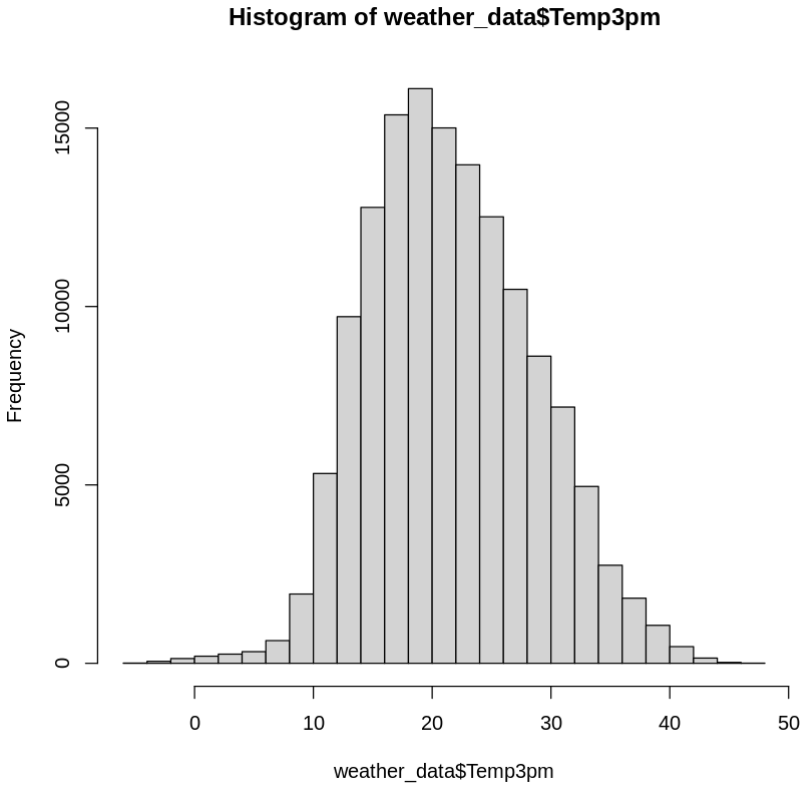

Histogram of weather_data$WindSpeed9am
Frequency
0
5000
10000
15000
20000
25000
30000
35000
0
20
40
60
80
100
120
weather_data$WindSpeed9am

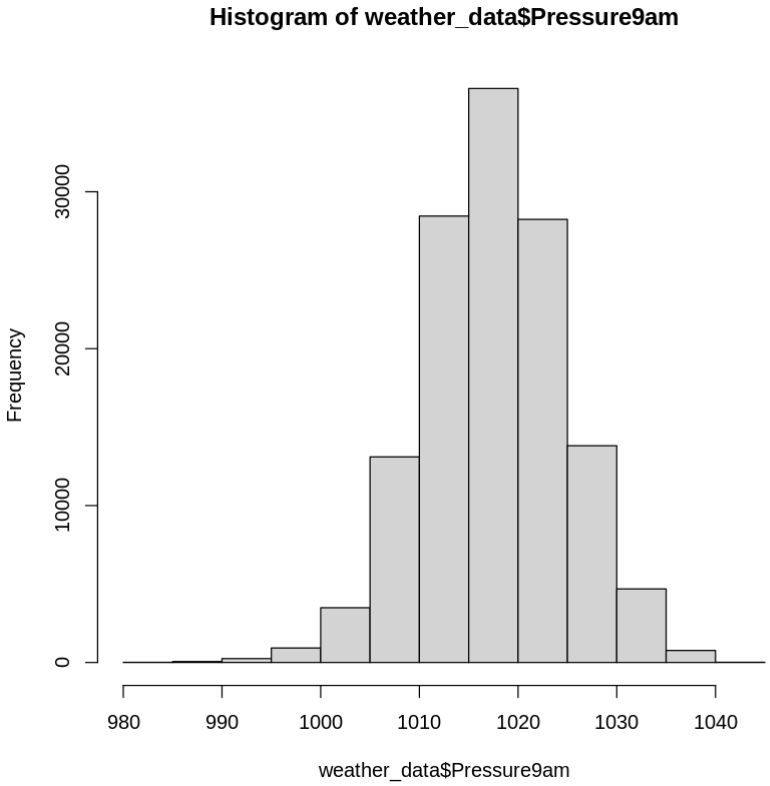

Histogram of weather_data$WindSpeed3pm
Frequency
0
5000
10000
15000
20000
25000
30000
35000
0
20
40
60
80
weather_data$WindSpeed3pm

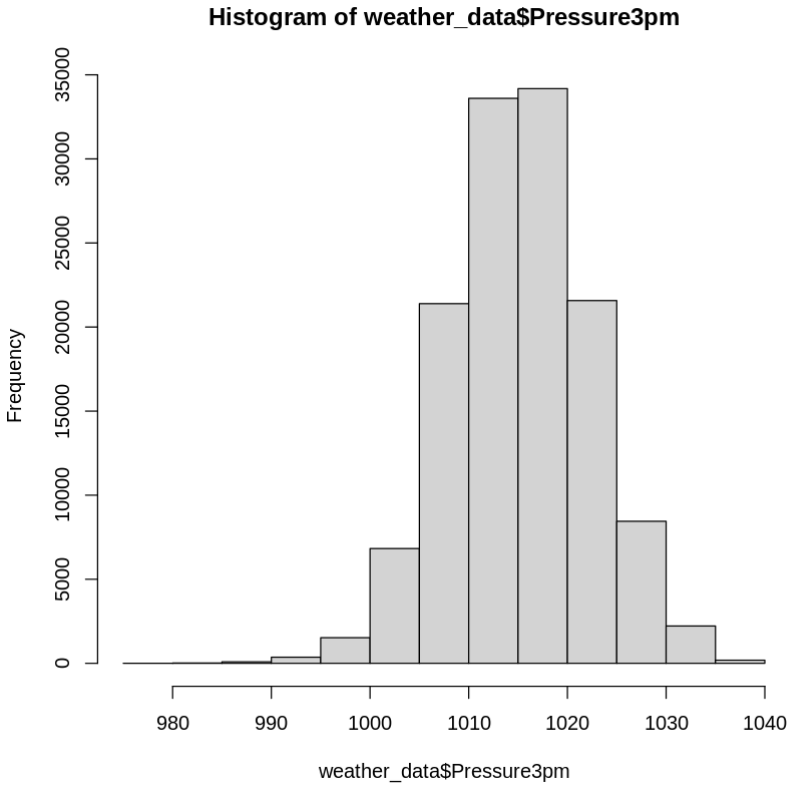

Histogram of weather_data$Humidity9am
Frequency
0
5000
10000
15000
0
20
40
60
80
100
weather_data$Humidity9am


Histogram of weather_data$Humidity3pm
Frequency
0
2000
4000
6000
8000
10000
12000
0
20
40
60
80
100
weather_data$Humidity3pm


Histogram of weather_data$Temp9am
Frequency
0
5000
10000
15000
-10
0
10
20
30
40
weather_data$Temp9am


Histogram of weather_data$Temp3pm
Frequency
0
5000
10000
15000
0
10
20
30
40
50
weather_data$Temp3pm


Histogram of weather_data$Pressure9am
Frequency
0
10000
20000
30000
980
990
1000
1010
1020
1030
1040
weather_data$Pressure9am



8. Feature extraction - use Chi-Square to check whether the variables are dependent on RainTomorrow

In [71]:

```
#2.Feature Extraction
#Get the categorical variables
#2.1. Chi-Square to check whether the variables are dependent on RainTomorrow
factor_vars1 <- names(which(sapply(weather_data3, class) == "factor"))
factor_vars1
factor_vars1 <- setdiff(factor_vars1, "RainTomorrow")
factor_vars1
chisq_test_res <- lapply(factor_vars1, function(x) {
  chisq.test(weather_data3[,x], weather_data3[, "RainTomorrow"], simulate.p.v
})
names(chisq_test_res) <- factor_vars1
chisq_test_res
#Baed on the chisquare values including categorical variables WindGustDir,Wir
```

'WindGustDir' · 'WindDir9am' · 'WindDir3pm' · 'RainTomorrow'

'WindGustDir' · 'WindDir9am' · 'WindDir3pm'

```
$WindGustDir

        Pearson's Chi-squared test with simulated p-value (based on 2000
        replicates)

data:  weather_data3[, x] and weather_data3[, "RainTomorrow"]
X-squared = 744.49, df = NA, p-value = 0.0004998


$WindDir9am

        Pearson's Chi-squared test with simulated p-value (based on 2000
        replicates)

data:  weather_data3[, x] and weather_data3[, "RainTomorrow"]
```

```
X-squared = 988.96, df = NA, p-value = 0.0004998


$WindDir3pm

        Pearson's Chi-squared test with simulated p-value (based on 2000
        replicates)

data:  weather_data3[, x] and weather_data3[, "RainTomorrow"]
X-squared = 685.77, df = NA, p-value = 0.0004998
```

9. Remove Categorical variables from dataset

In [72]:

```
#Feature Extraction for numberic variables
#2.2Method:Correlation
#Remove Categorical variables from dataset
weather_data4 <- subset(weather_data2, select = -c(WindDir9am, WindDir3pm))
colnames(weather_data4)
weather_data5 <- weather_data4[complete.cases(weather_data4),]
numeric_vars <- setdiff(colnames(weather_data5), factor_vars1)
numeric_vars <- setdiff(numeric_vars, "RainTomorrow")
numeric_vars_mat <- as.matrix(weather_data5[, numeric_vars, drop=FALSE])
numeric_vars_cor <- cor(numeric_vars_mat)
```

'MinTemp' · 'MaxTemp' · 'Evaporation' · 'Sunshine' · 'WindGustDir' · 'WindGustSpeed' · 'WindSpeed9am' · 'WindSpeed3pm' · 'Humidity9am' · 'Humidity3pm' · 'Pressure9am' · 'Pressure3pm' · 'Cloud9am' · 'Cloud3pm' · 'Temp9am' · 'Temp3pm' · 'RainTomorrow'

10. Find correlation

In [77]:

```
library(caret)
fndCorrelation = findCorrelation(numeric_vars_cor, cutoff=0.6) # putt any val
fndCorrelation = sort(fndCorrelation)
reduced_Data = numeric_vars_mat[,c(fndCorrelation)]
cols=colnames(reduced_Data)
cols
summary (reduced_Data)
```

'MaxTemp' · 'Sunshine' · 'WindGustSpeed' · 'Humidity9am' · 'Pressure3pm' · 'Cloud3pm' · 'Temp9am' · 'Temp3pm'

```
    MaxTemp         Sunshine      WindGustSpeed     Humidity9am
 Min.   : 4.10   Min.   : 0.000   Min.   :  9.00   Min.   :  0.00
 1st Qu.:18.60   1st Qu.: 5.000   1st Qu.: 31.00   1st Qu.: 55.00
 Median :23.80   Median : 8.600   Median : 39.00   Median : 67.00
 Mean   :24.13   Mean   : 7.699   Mean   : 40.55   Mean   : 66.22
 3rd Qu.:29.60   3rd Qu.:10.700   3rd Qu.: 48.00   3rd Qu.: 80.00
 Max.   :48.10   Max.   :14.500   Max.   :124.00   Max.   :100.00
  Pressure3pm        Cloud3pm        Temp9am         Temp3pm
 Min.   : 977.1   Min.   :0.000   Min.   :-0.90   Min.   : 3.70
 1st Qu.:1010.1   1st Qu.:2.000   1st Qu.:12.90   1st Qu.:17.30
 Median :1014.8   Median :5.000   Median :17.70   Median :22.30
 Mean   :1014.9   Mean   :4.328   Mean   :18.09   Mean   :22.63
 3rd Qu.:1019.5   3rd Qu.:7.000   3rd Qu.:23.20   3rd Qu.:27.80
 Max.   :1038.9   Max.   :9.000   Max.   :39.40   Max.   :46.10
```

11. Get the numeric and categorical variables

In [76]:

```
# Get the numeric and categorical variables
library(dplyr)
```

```
weather_data7= weather_data2[c("WindGustDir","WindDir9am","WindDir3pm","RainT
weather_data9= weather_data2[c(cols)]
```

12. Remove outliers

In [75]:

```
#remove outliers
library(outliers)
apply(weather_data9,2,out.rem)
colnames(weather_data9)
```

A matrix: 145460 × 8 of type dbl

	MaxTemp	Sunshine	WindGustSpeed	Humidity9am	Pressure3pm	Cloud3pm	Temp9am
1	22.9	NA	44	71	1007.1	NA	16.9
2	25.1	NA	44	44	1007.8	NA	17.2
3	25.7	NA	46	38	1008.7	2	21.0
4	28.0	NA	24	45	1012.8	NA	18.1
5	32.3	NA	41	82	1006.0	8	17.8
6	29.7	NA	56	55	1005.4	NA	20.6
7	25.0	NA	50	49	1008.2	NA	18.1
8	26.7	NA	35	48	1010.1	NA	16.3
9	31.9	NA	80	42	1003.6	NA	18.3
10	30.1	NA	28	58	1005.7	NA	20.1
11	30.4	NA	30	48	1008.7	NA	20.4
12	21.7	NA	31	89	1004.2	8	15.9
13	18.6	NA	61	76	993.0	8	17.4
14	21.0	NA	44	65	1001.8	7	15.8
15	24.6	NA	NA	57	1008.7	NA	15.9
16	27.7	NA	50	50	1010.3	NA	17.3
17	20.9	NA	22	69	1010.4	1	17.2
18	22.9	NA	63	80	1002.2	1	18.0
19	22.5	NA	43	47	1009.7	2	15.5
20	25.6	NA	26	45	1017.1	NA	15.8
21	29.3	NA	24	56	1014.8	NA	19.1
22	33.0	NA	43	38	1008.1	1	24.5
23	31.8	NA	41	54	1005.7	NA	23.8
24	30.9	NA	33	55	1008.2	NA	20.9
25	32.4	NA	43	49	1010.1	NA	21.5
26	33.9	NA	35	45	1007.6	1	23.2
27	33.0	NA	57	41	1003.6	1	26.6
28	32.7	NA	48	56	1001.7	NA	24.6
29	27.2	NA	46	49	1004.2	NA	21.6
30	24.2	NA	50	78	1003.4	8	12.5
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

	MaxTemp	Sunshine	WindGustSpeed	Humidity9am	Pressure3pm	Cloud3pm	Temp9am
145431	28.0	NA	28	36	1016.9	NA	16.6
145432	24.6	NA	33	46	1018.8	NA	13.8
145433	22.2	NA	37	59	1021.7	8	13.9
145434	22.7	NA	35	62	1023.6	4	11.8
145435	20.5	NA	46	56	1025.3	NA	11.1
145436	19.4	NA	43	61	1027.1	NA	9.7
145437	19.2	NA	37	45	1026.7	NA	6.8
145438	19.8	NA	39	42	1024.1	NA	8.6
145439	18.8	NA	31	42	1022.4	NA	9.5
145440	20.7	NA	37	38	1023.9	NA	10.3
145441	20.8	NA	43	29	1025.9	NA	7.9
145442	21.7	NA	43	27	1025.0	NA	9.5
145443	20.0	NA	39	58	1026.7	NA	9.1
145444	19.8	NA	43	54	1027.7	NA	8.7
145445	18.6	NA	46	56	1027.2	NA	8.2
145446	19.1	NA	33	59	1026.2	NA	8.0
145447	19.7	NA	46	56	1025.8	NA	8.5
145448	21.2	NA	35	57	1023.0	NA	10.3
145449	21.4	NA	22	60	1022.8	NA	8.9
145450	22.5	NA	19	59	1021.4	NA	8.8
145451	24.3	NA	24	53	1020.0	NA	12.3
145452	23.4	NA	31	53	1023.0	NA	11.2
145453	20.7	NA	41	56	1024.3	7	11.6
145454	20.6	NA	35	63	1023.3	NA	11.0
145455	21.8	NA	31	59	1021.2	NA	9.4
145456	23.4	NA	31	51	1020.3	NA	10.1
145457	25.3	NA	22	56	1019.1	NA	10.9
145458	26.9	NA	37	53	1016.8	NA	12.5
145459	27.0	NA	28	51	1016.5	2	15.1
145460	NA	NA	NA	62	1017.9	8	15.0

'MaxTemp' · 'Sunshine' · 'WindGustSpeed' · 'Humidity9am' · 'Pressure3pm' · 'Cloud3pm' · 'Temp9am' · 'Temp3pm'

13. merge numeric and factor columns

In [78]:

```
#merge numeric and factor columns
weather_data10=cbind(weather_data9,weather_data7)
summary(weather_data10)
dim(weather_data10$Pressure3pm)
dim(weather_data10$Cloud3pm)
```

```
   MaxTemp         Sunshine     WindGustSpeed     Humidity9am
Min.   :-4.80   Min.   : 0.00   Min.   :  6.00   Min.   :  0.00
1st Qu.:17.90   1st Qu.: 4.80   1st Qu.: 31.00   1st Qu.: 57.00
Median :22.60   Median : 8.40   Median : 39.00   Median : 70.00
Mean   :23.22   Mean   : 7.61   Mean   : 40.03   Mean   : 68.88
3rd Qu.:28.20   3rd Qu.:10.60   3rd Qu.: 48.00   3rd Qu.: 83.00
Max.   :48.10   Max.   :14.50   Max.   :135.00   Max.   :100.00
NA's   :1261    NA's   :69835   NA's   :10263    NA's   :2654
 Pressure3pm        Cloud3pm        Temp9am         Temp3pm
Min.   : 977.1   Min.   :0.00    Min.   :-7.20   Min.   :-5.40
1st Qu.:1010.4   1st Qu.:2.00    1st Qu.:12.30   1st Qu.:16.60
Median :1015.2   Median :5.00    Median :16.70   Median :21.10
Mean   :1015.3   Mean   :4.51    Mean   :16.99   Mean   :21.68
3rd Qu.:1020.0   3rd Qu.:7.00    3rd Qu.:21.60   3rd Qu.:26.40
Max.   :1039.6   Max.   :9.00    Max.   :40.20   Max.   :46.70
NA's   :15028    NA's   :59358   NA's   :1767    NA's   :3609
 WindGustDir      WindDir9am      WindDir3pm    RainTomorrow
W      : 9915   N      :11758   SE     :10838   No  :110316
SE     : 9418   SE     : 9287   W      :10110   Yes : 31877
N      : 9313   E      : 9176   S      : 9926   NA's:  3267
SSE    : 9216   SSE    : 9112   WSW    : 9518
E      : 9181   NW     : 8749   SSE    : 9399
(Other):88091   (Other):86812   (Other):91441
NA's   :10326   NA's   :10566   NA's   : 4228
NULL
NULL
```

14. Data normalisation/Cleaning - Replace NA values with mean,mode

In [123…

```
#Data normalisation/Cleaning
#Replace NA values with mean,mode
library(dplyr)
weather_data10=weather_data10 %>% mutate_if(is.numeric, funs(replace(.,is.na(
  mutate_if(is.factor, funs(replace(.,is.na(.), getmode(na.omit(.))))))
summary(weather_data10)
```

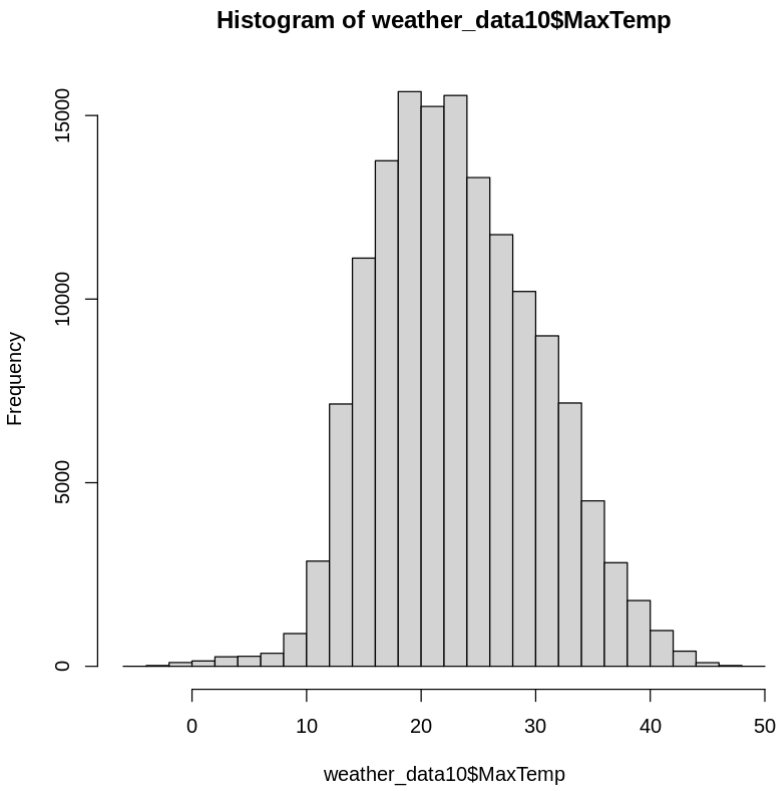
```
   MaxTemp         Sunshine      WindGustSpeed     Humidity9am
Min.   :-4.80   Min.   : 0.000   Min.   :  6.00   Min.   :  0.00
1st Qu.:18.00   1st Qu.: 7.611   1st Qu.: 31.00   1st Qu.: 57.00
Median :22.70   Median : 7.611   Median : 39.00   Median : 69.00
Mean   :23.22   Mean   : 7.611   Mean   : 40.04   Mean   : 68.88
3rd Qu.:28.20   3rd Qu.: 8.700   3rd Qu.: 46.00   3rd Qu.: 83.00
Max.   :48.10   Max.   :14.500   Max.   :135.00   Max.   :100.00
 Pressure3pm        Cloud3pm       Temp9am         Temp3pm
Min.   : 977.1   Min.   :0.00   Min.   :-7.20   Min.   :-5.40
1st Qu.:1011.1   1st Qu.:4.00   1st Qu.:12.30   1st Qu.:16.70
Median :1015.3   Median :4.51   Median :16.80   Median :21.40
Mean   :1015.3   Mean   :4.51   Mean   :16.99   Mean   :21.68
3rd Qu.:1019.4   3rd Qu.:6.00   3rd Qu.:21.50   3rd Qu.:26.20
Max.   :1039.6   Max.   :9.00   Max.   :40.20   Max.   :46.70
 WindGustDir       WindDir9am   WindDir3pm     RainTomorrow
Min.   : 1.000   Min.   : 1   Min.   : 1.000   0:113583
1st Qu.: 5.000   1st Qu.: 4   1st Qu.: 5.000   1: 31877
Median :10.000   Median : 8   Median : 9.000
Mean   : 9.094   Mean   : 8   Mean   : 8.799
3rd Qu.:14.000   3rd Qu.:12   3rd Qu.:13.000
Max.   :16.000   Max.   :16   Max.   :16.000
```

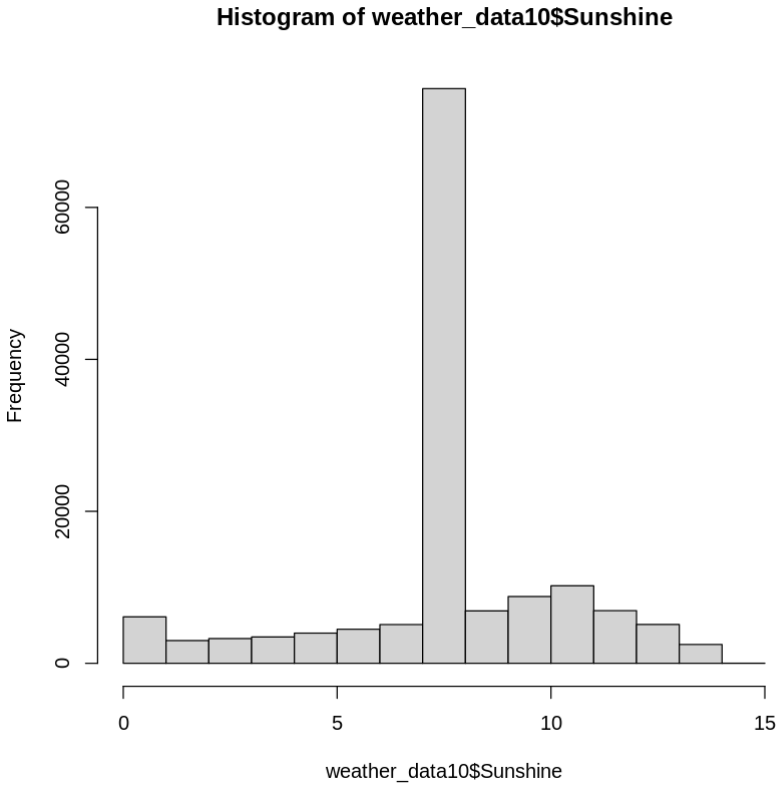
15. Plot to verify results of Data Preprocessing

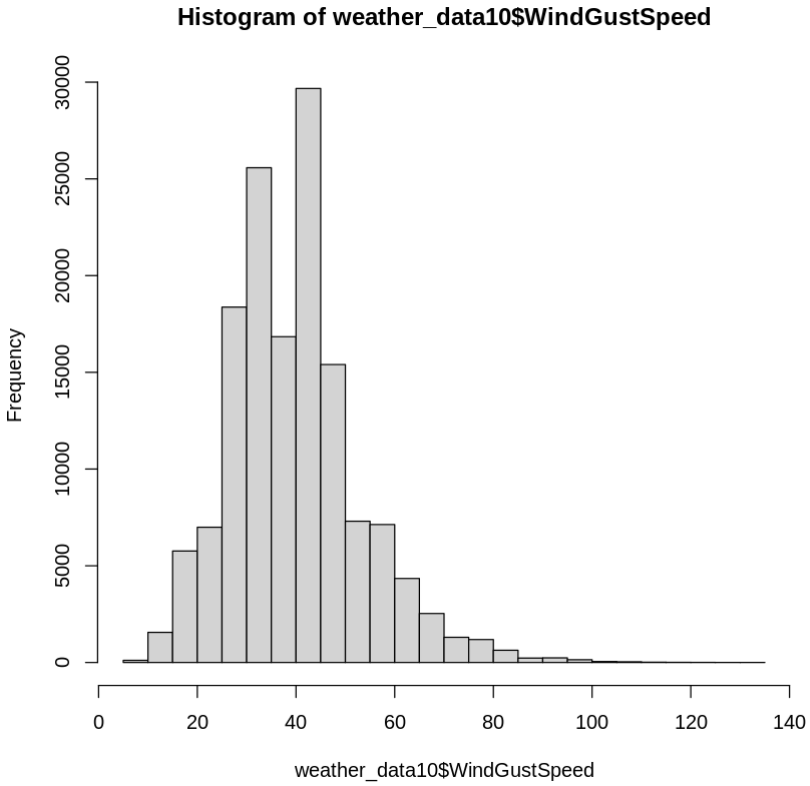
In [80]:

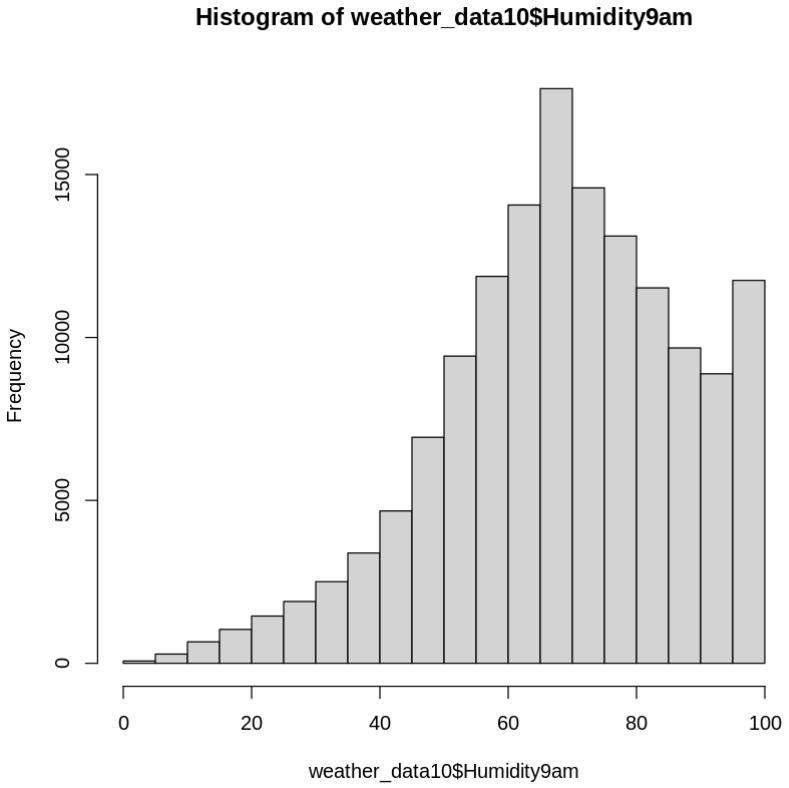
```
#plot to verify results of Data Preprocessing
hist(weather_data10$MaxTemp)
hist(weather_data10$Sunshine)
```

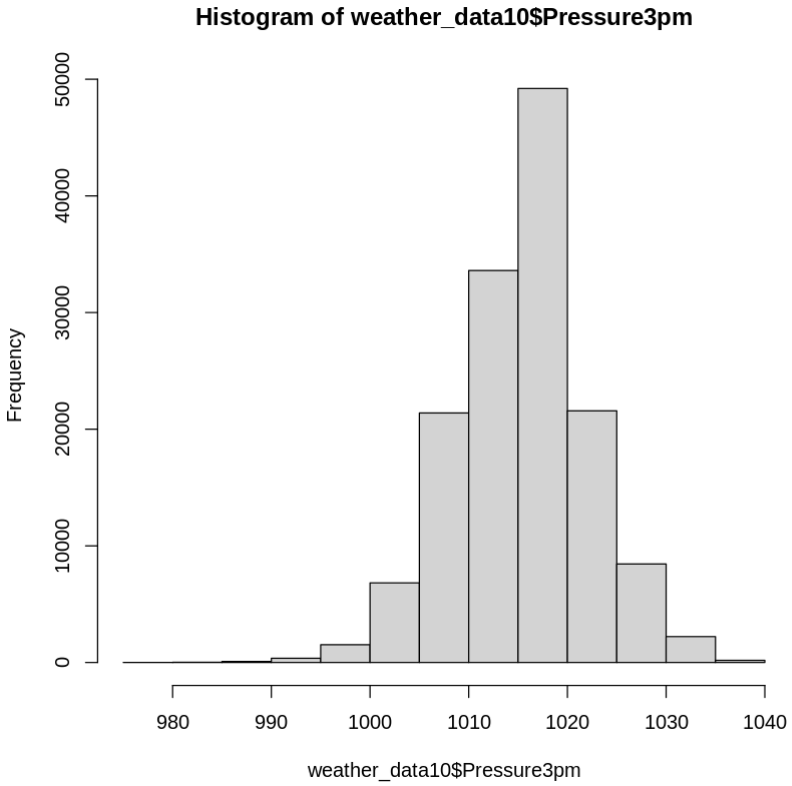
```
hist(weather_data10$WindGustSpeed)
hist(weather_data10$Humidity9am)
hist(weather_data10$Pressure3pm)
hist(weather_data10$Cloud3pm)
hist(weather_data10$Temp9am)
hist(weather_data10$Temp3pm)
WindGustDirnum=as.numeric(weather_data10$WindGustDir)
WindGustDirnum
WindDir9amnum=as.numeric(weather_data10$WindDir9am)
WindDir9amnum
WindDir3pmnum=as.numeric(weather_data10$WindDir3pm)
WindDir3pmnum
hist(WindGustDirnum)
hist(WindDir9amnum)
hist(WindDir3pmnum)
```

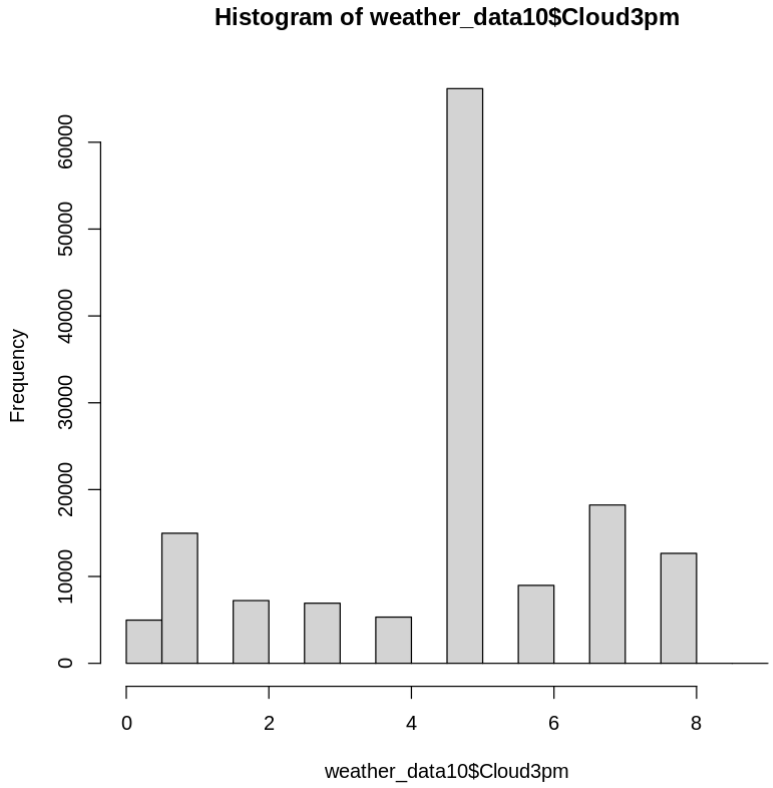


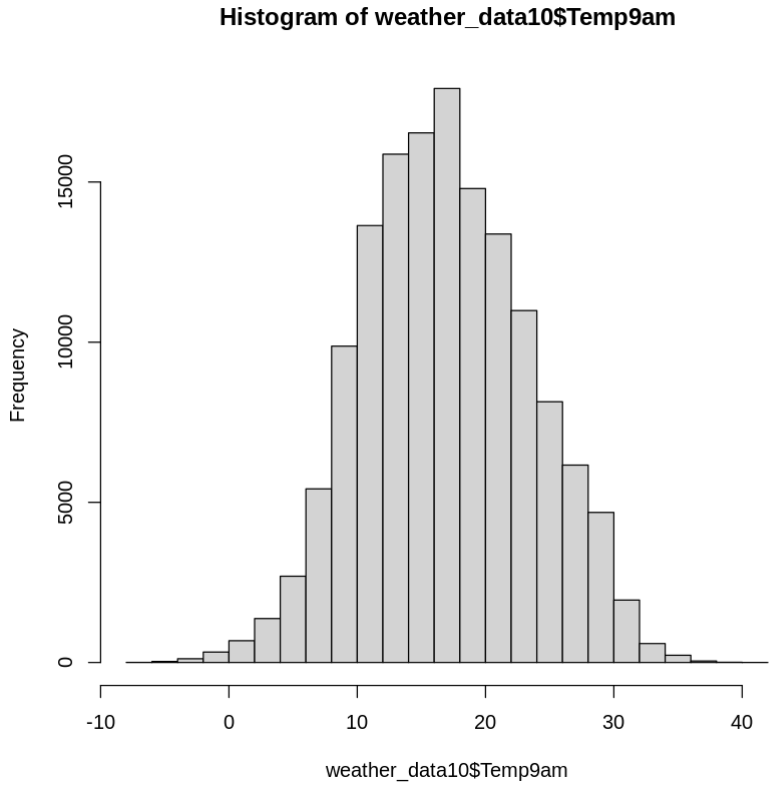

Histogram of weather_data10$Sunshine
Frequency
0
20000
40000
60000
0
5
10
15
weather_data10$Sunshine


Histogram of weather_data10$WindGustSpeed
Frequency
0
5000
10000
15000
20000
25000
30000
0
20
40
60
80
100
120
140
weather_data10$WindGustSpeed


Histogram of weather_data10$Humidity9am
Frequency
0
5000
10000
15000
0
20
40
60
80
100
weather_data10$Humidity9am


Histogram of weather_data10$Pressure3pm
Frequency
0
10000
20000
30000
40000
50000
980
990
1000
1010
1020
1030
1040
weather_data10$Pressure3pm



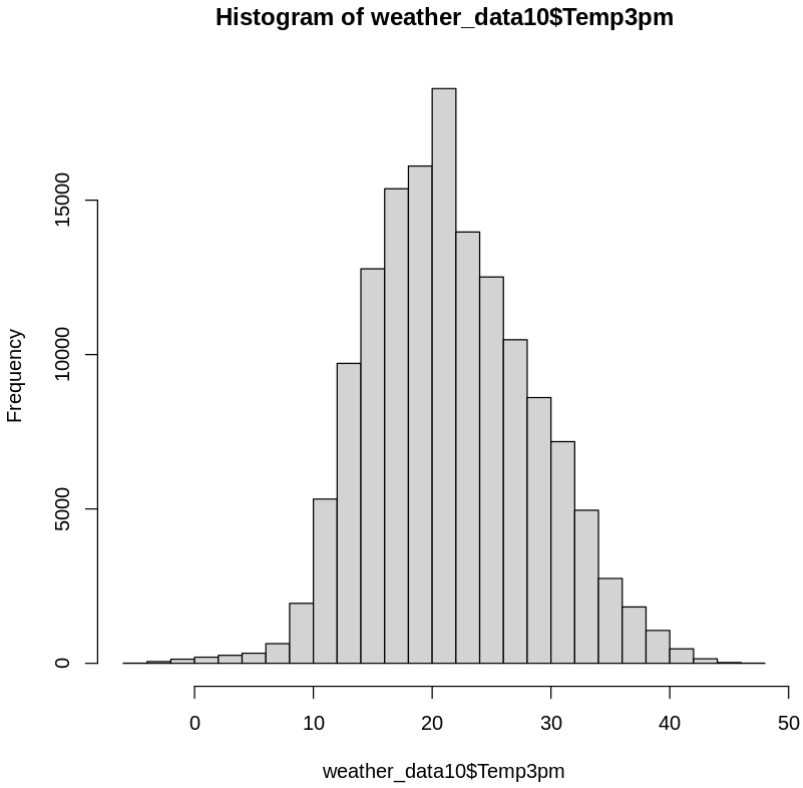


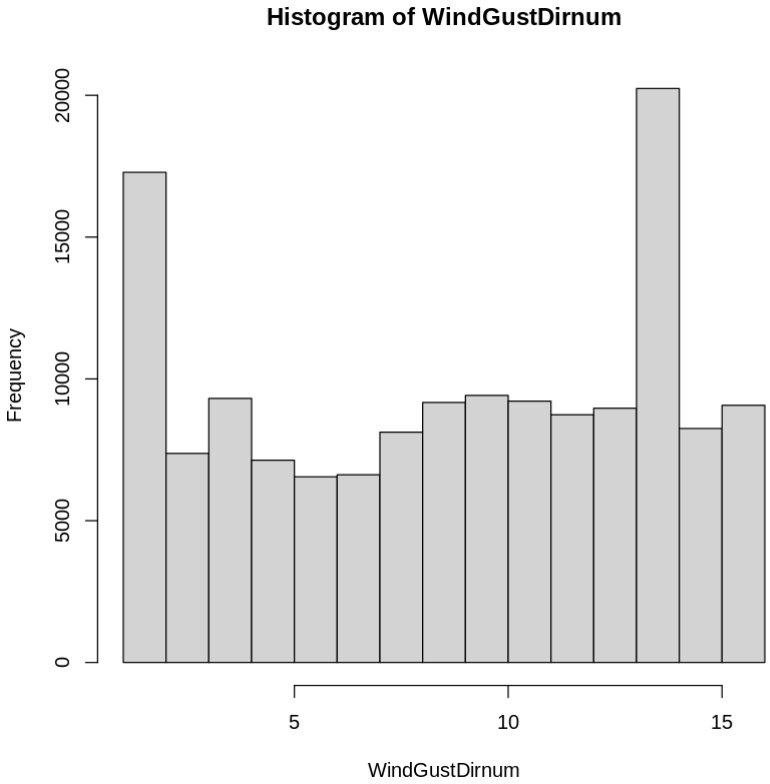
14 · 15 · 16 · 5 · 14 · 15 · 14 · 14 · 7 · 14 · 4 · 6 · 14 · 13 · 14 · 15 · 2 · 14 · 11 · 11 · 9 · 5 · 15 · 4 · 14 · 16 · 16 · 15 · 15 · 15 · 14 · 15 · 14 · 11 · 15 · 15 · 14 · 14 · 14 · 5 · 5 · 9 · 13 · 7 · 8 · 15 · 13 · 10 · 15 · 15 · 7 · 7 · 8 · 15 · 14 · 16 · 9 · 6 · 4 · 13 · 6 · 3 · 13 · 13 · 14 · 14 · 16 · 14 · 7 · 8 · 10 · 11 · 11 · 10 · 11 · 5 · 6 · 1 · 4 · 8 · 16 · 12 · 11 · 8 · 8 · 15 · 15 · 3 · 14 · 16 · 14 · 7 · 15 · 15 · 13 · 16 · 14 · 11 · 6 · 5 · 5 · 14 · 2 · 13 · 15 · 14 · 15 · 8 · 16 · 11 · 7 · 5 · 8 · 9 · 2 · 16 · 15 · 7 · 9 · 2 · 16 · 10 · 5 · 14 · 14 · 16 · 14 · 10 · 11 · 2 · 14 · 14 · 16 · 6 · 15 · 14 · 14 · 15 · 14 · 10 · 11 · 3 · 3 · 5 · 15 · 14 · 7 · 8 · 15 · 11 · 13 · 13 · 13 · 9 · 14 · 3 · 13 · 13 · 14 · 14 · 10 · 3 · 14 · 15 · 15 · 15 · 14 · 14 · 1 · 11 · 1 · 10 · 10 · 1 · 10 · 5 · 1 · 14 · 11 · 3 · 10 · 8 · 10 · 10 · 10 · 2 ·

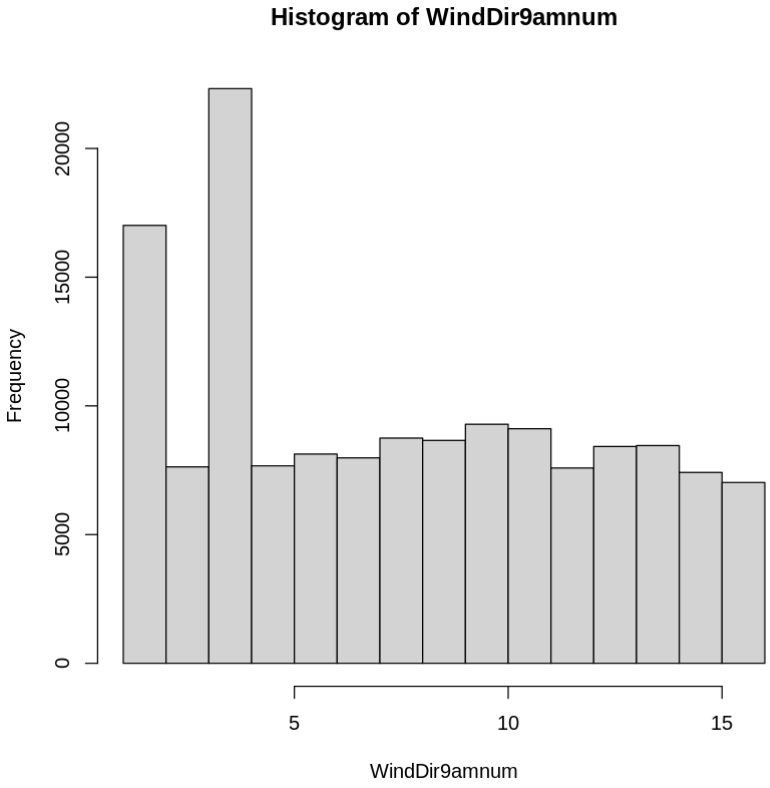
14 · 3 · 7 · 15 · 14 · 15 · 10 · 3 · 2 · 6 · 3 · 10 · 16 · 4 · ⋯ · 9 · 3 · 1 · 1 · 2 · 13 · 8 · 9 · 11 · 1 · 14 · 5 · 9 · 1 · 5 · 7 · 4 · 7 · 2 · 15 · 8 · 12 · 1 · 5 · 1 · 3 · 1 · 3 · 1 · 15 · 2 · 2 · 10 · 1 · 2 · 6 · 16 · 3 · 1 · 2 · 5 · 2 · 7 · 14 · 5 · 6 · 12 · 11 · 1 · 5 · 5 · 1 · 2 · 14 · 1 · 2 · 14 · 2 · 6 · 9 · 2 · 5 · 5 · 2 · 15 · 9 · 10 · 3 · 2 · 1 · 13 · 11 · 11 · 3 · 2 · 14 · 12 · 1 · 10 · 1 · 3 · 11 · 11 · 3 · 10 · 1 · 3 · 1 · 10 · 1 · 7 · 5 · 6 · 12 · 12 · 1 · 11 · 9 · 9 · 1 · 2 · 2 · 12 · 9 · 2 · 4 · 5 · 1 · 7 · 11 · 1 · 11 · 11 · 1 · 11 · 1 · 1 · 10 · 3 · 2 · 15 · 11 · 11 · 1 · 1 · 2 · 1 · 1 · 2 · 2 · 14 · 2 · 9 · 1 · 11 · 2 · 1 · 9 · 11 · 11 · 3 · 3 · 3 · 2 · 12 · 11 · 1 · 1 · 2 · 1 · 2 · 1 · 1 · 2 · 1 · 7 · 8 · 15 · 15 · 8 · 16 · 9 · 2 · 8 · 7 · 7 · 10 · 1 · 9 · 15 · 14 · 1 · 1 · 1 · 1 · 1 · 1 · 2 · 1 · 1 · 1 · 1 · 10 · 1 · 1 · 1 · 1 · 2 · 10 · 9 · 1 · 3 · 3 · 1 · 1 · 1 · 7 · 4 · 10 · 14

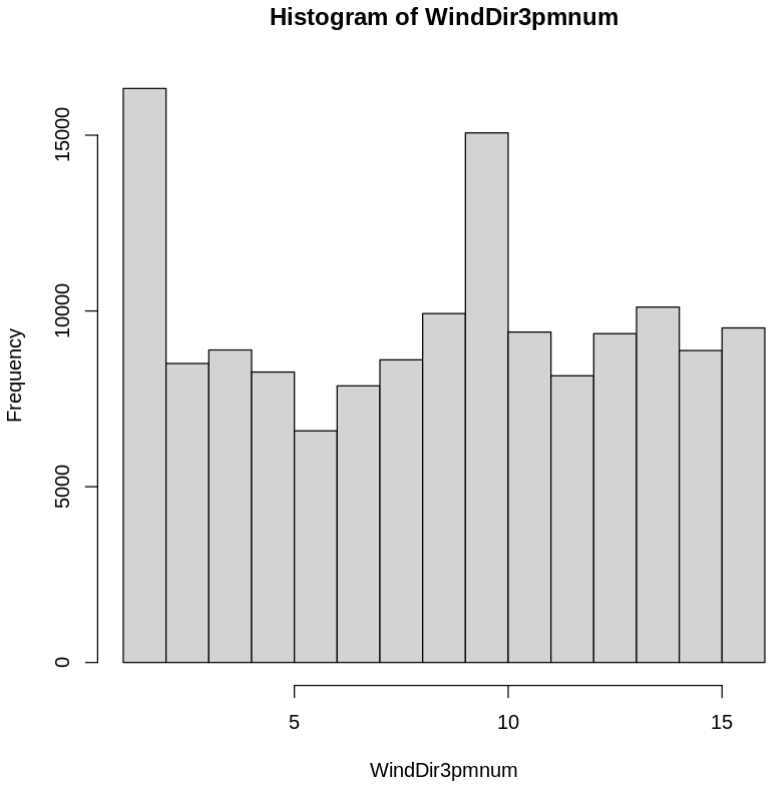
14 · 7 · 14 · 10 · 2 · 14 · 13 · 11 · 10 · 9 · 11 · 5 · 7 · 14 · 9 · 4 · 12 · 4 · 16 · 10 · 10 · 5 · 14 · 3 · 1 · 10 · 4 · 4 · 8 · 16 · 15 · 14 · 16 · 11 · 11 · 2 · 10 · 1 · 14 · 11 · 6 · 11 · 10 · 10 · 11 · 14 · 16 · 10 · 2 · 11 · 11 · 5 · 4 · 9 · 14 · 3 · 11 · 10 · 1 · 3 · 3 · 3 · 10 · 5 · 4 · 4 · 11 · 10 · 10 · 4 · 13 · 11 · 11 · 11 · 10 · 9 · 11 · 10 · 11 · 2 · 10 · 7 · 11 · 9 · 10 · 13 · 4 · 3 · 4 · 1 · 14 · 9 · 5 · 14 · 16 · 4 · 9 · 1 · 11 · 3 · 5 · 2 · 4 · 1 · 14 · 9 · 11 · 11 · 9 · 5 · 4 · 1 · 4 · 11 · 11 · 4 · 4 · 4 · 4 · 4 · 9 · 4 · 4 · 1 · 4 · 1 · 11 · 9 · 11 · 3 · 10 · 9 · 4 · 2 · 4 · 8 · 4 · 2 · 11 · 10 · 10 · 2 · 4 · 4 · 3 · 15 · 8 · 4 · 14 · 10 · 4 · 4 · 2 · 4 · 2 · 4 · 2 · 10 · 4 · 4 · 4 · 7 · 2 · 10 · 15 · 8 · 14 · 1 · 4 · 4 · 11 · 4 · 9 · 5 · 4 · 10 · 1 · 4 · 4 · 4 · 4 · 4 · 4 · 4 · 2 · 8 · 6 · 1 · 7 · 15 · 7 · 14 · 4 · 1 · 1 · 4 · 6 · 4 · 4 · 4 · ⋯ · 9 · 3 · 1 · 1 · 2 · 6 · 6 · 10 · 1 · 3 · 1 · 5 · 5 · 3 · 1 · 6 · 5 · 5 · 4 · 15 · 8 · 9 · 3 · 5 · 11 · 3 · 1 · 1 · 3 · 4 · 4 · 10 · 1 · 3 · 3 · 1 · 8 · 10 · 1 · 2 · 3 · 6 · 7 · 8 · 2 · 2 · 2 · 11 · 1 · 1 · 1 · 2 · 1 · 1 · 3 · 3 · 3 · 2 · 5 · 7 · 3 · 1 · 4 · 4 · 1 · 9 · 11 · 3 · 1 · 2 · 1 · 10 · 10 · 3 · 1 · 2 · 12 · 3 · 1 · 3 · 1 · 1 · 1 · 1 · 3 · 11 · 1 · 3 · 10 · 3 · 2 · 11 · 5 · 3 · 11 · 1 · 3 · 10 · 11 · 1 · 1 · 1 · 9 · 10 · 1 · 1 · 3 · 3 · 3 · 11 · 3 · 3 · 10 · 1 · 1 · 3 · 1 · 3 · 10 · 1 · 1 · 11 · 11 · 3 · 1 · 1 · 1 · 3 · 11 · 10 · 3 · 5 · 1 · 1 · 11 · 3 · 3 · 10 · 10 · 10 · 3 · 1 · 1 · 1 · 10 · 11 · 1 · 1 · 1 · 3 · 3 · 2 · 1 · 3 · 3 · 3 · 10 · 11 · 10 · 4 · 9 · 9 · 4 · 10 · 3 · 5 · 9 · 3 · 12 · 4 · 12 · 10 · 1 · 10 · 1 · 1 · 3 · 3 · 3 · 3 · 3 · 3 · 3 · 3 · 3 · 10 · 3 · 3 · 10 · 9 · 10 · 9 · 10 · 3 · 3 · 10 · 10 · 10 · 11 · 3

15 · 16 · 16 · 1 · 8 · 14 · 14 · 14 · 8 · 11 · 3 · 2 · 7 · 12 · 15 · 15 · 1 · 15 · 13 · 7 · 10 · 4 · 14 · 8 · 14 · 16 · 14 · 15 · 16 · 13 · 15 · 15 · 12 · 1 · 8 · 8 · 15 · 14 · 16 · 9 · 1 · 4 · 16 · 8 · 12 · 14 · 13 · 13 · 12 · 6 · 15 · 16 · 7 · 8 · 14 · 11 · 16 · 7 · 6 · 9 · 10 · 13 · 1 · 8 · 10 · 10 · 9 · 14 · 8 · 14 · 16 · 10 · 1 · 5 · 11 · 10 · 12 · 3 · 2 · 13 · 16 · 11 · 9 · 15 · 8 · 15 · 16 · 9 · 9 · 10 · 16 · 2 · 6 · 15 · 14 · 15 · 14 · 11 · 5 · 1 · 4 · 9 · 9 · 4 · 8 · 15 · 16 · 8 · 13 · 4 · 6 · 6 · 15 · 15 · 11 · 15 · 14 · 4 · 10 · 10 · 12 · 2 · 2 · 6 · 16 · 8 · 16 · 9 · 11 · 9 · 5 · 1 · 10 · 12 · 14 · 14 · 15 · 14 · 14 · 10 · 10 · 11 · 13 · 5 · 6 · 7 · 15 · 7 · 14 · 11 · 13 · 16 · 14 · 11 · 16 · 10 · 15 · 15 · 14 · 16 · 10 · 9 · 15 · 14 · 15 · 8 · 16 · 1 · 2 · 11 · 2 · 10 · 10 · 10 · 10 · 5 · 11 · 16 · 9 · 10 · 10 · 10 · 10 · 3 · 10 · 1 · 14 · 10 · 7 · 15 · 15 · 16 · 3 · 11 · 5 · 5 · 10 · 3 · 14 · 10 · ⋯ · 11 · 10 · 2 · 3 · 7 · 16 · 7 · 11 · 4 · 3 · 6 · 7 · 3 · 3 · 5 · 8 · 5 · 5 · 7 · 14 · 14 · 16 · 14 · 2 · 1 · 3 · 3 · 2 · 2 · 5 · 2 · 5 · 2 · 1 · 1 · 4 · 3 · 1 · 1 · 5 · 3 · 2 · 8 · 8 · 11 · 16 · 14 · 10 · 1 · 2 · 2 · 5 · 10 · 11 · 2 · 5 · 1 · 1 · 7 · 4 · 1 · 5 · 6 · 4 · 2 · 14 · 10 · 3 · 2 · 10 · 4 · 11 · 9 · 11 · 1 · 13 · 14 · 3 · 10 · 2 · 2 · 12 · 11 · 10 · 3 · 3 · 11 · 3 · 10 · 1 · 8 · 1 · 11 · 15 · 3 · 16 · 12 · 15 · 3 · 2 · 10 · 11 · 6 · 6 · 15 · 10 · 2 · 11 · 4 · 10 · 3 · 15 · 10 · 3 · 10 · 3 · 10 · 11 · 11 · 4 · 7 · 8 · 3 · 3 · 1 · 10 · 1 · 5 · 6 · 2 · 11 · 2 · 12 · 1 · 10 · 1 · 5 · 13 · 11 · 10 · 1 · 3 · 2 · 5 · 11 · 3 · 3 · 3 · 15 · 10 · 1 · 1 · 10 · 1 · 1 · 15 · 15 · 8 · 8 · 8 · 13 · 11 · 4 · 15 · 4 · 15 · 5 · 1 · 9 · 8 · 15 · 3 · 3 · 1 · 1 · 3 · 3 · 1 · 1 · 3 · 1 · 2 · 3 · 1 · 1 · 1 · 1 · 1 · 11 · 1 · 1 · 3 · 1 · 1 · 1 · 2 · 4 · 15 · 4 · 3


Histogram of weather_data10$Temp3pm
Frequency
0
5000
10000
15000
0
10
20
30
40
50
weather_data10$Temp3pm


Histogram of WindGustDirnum
Frequency
0
5000
10000
15000
20000
5
10
15
WindGustDirnum





16. Data modeling

In [81]:
```
# 4.Data Modeling
weather_data10$WindGustDir=as.numeric(weather_data10$WindGustDir)
weather_data10$WindDir9am=as.numeric(weather_data10$WindDir9am)
weather_data10$WindDir3pm=as.numeric(weather_data10$WindDir3pm)
```

17. Convert Raintomrrow data to numeric

In [122…

```
#Convert Raintomrrow data to numeric
library(plyr)
weather_data10
weather_data10$RainTomorrow <- revalue(weather_data10$RainTomorrow, c("Yes"=1
weather_data10$RainTomorrow <- revalue(weather_data10$RainTomorrow, c("No"=0)
```

A data.frame: 145460 × 12

	MaxTemp	Sunshine	WindGustSpeed	Humidity9am	Pressure3pm	Cloud3pm	Temp9am
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	22.9	7.611178	44.00000	71	1007.1	4.50993	16.9
2	25.1	7.611178	44.00000	44	1007.8	4.50993	17.2
3	25.7	7.611178	46.00000	38	1008.7	2.00000	21.0
4	28.0	7.611178	24.00000	45	1012.8	4.50993	18.1
5	32.3	7.611178	41.00000	82	1006.0	8.00000	17.8
6	29.7	7.611178	56.00000	55	1005.4	4.50993	20.6
7	25.0	7.611178	50.00000	49	1008.2	4.50993	18.1
8	26.7	7.611178	35.00000	48	1010.1	4.50993	16.3
9	31.9	7.611178	80.00000	42	1003.6	4.50993	18.3
10	30.1	7.611178	28.00000	58	1005.7	4.50993	20.1
11	30.4	7.611178	30.00000	48	1008.7	4.50993	20.4
12	21.7	7.611178	31.00000	89	1004.2	8.00000	15.9
13	18.6	7.611178	61.00000	76	993.0	8.00000	17.4
14	21.0	7.611178	44.00000	65	1001.8	7.00000	15.8
15	24.6	7.611178	40.03523	57	1008.7	4.50993	15.9
16	27.7	7.611178	50.00000	50	1010.3	4.50993	17.3
17	20.9	7.611178	22.00000	69	1010.4	1.00000	17.2
18	22.9	7.611178	63.00000	80	1002.2	1.00000	18.0
19	22.5	7.611178	43.00000	47	1009.7	2.00000	15.5
20	25.6	7.611178	26.00000	45	1017.1	4.50993	15.8
21	29.3	7.611178	24.00000	56	1014.8	4.50993	19.1
22	33.0	7.611178	43.00000	38	1008.1	1.00000	24.5
23	31.8	7.611178	41.00000	54	1005.7	4.50993	23.8
24	30.9	7.611178	33.00000	55	1008.2	4.50993	20.9
25	32.4	7.611178	43.00000	49	1010.1	4.50993	21.5
26	33.9	7.611178	35.00000	45	1007.6	1.00000	23.2
27	33.0	7.611178	57.00000	41	1003.6	1.00000	26.6
28	32.7	7.611178	48.00000	56	1001.7	4.50993	24.6
29	27.2	7.611178	46.00000	49	1004.2	4.50993	21.6
30	24.2	7.611178	50.00000	78	1003.4	8.00000	12.5
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
145431	28.00000	7.611178	28.00000	36	1016.9	4.50993	16.6
145432	24.60000	7.611178	33.00000	46	1018.8	4.50993	13.8

	MaxTemp	Sunshine	WindGustSpeed	Humidity9am	Pressure3pm	Cloud3pm	Temp9am
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
145433	22.20000	7.611178	37.00000	59	1021.7	8.00000	13.9
145434	22.70000	7.611178	35.00000	62	1023.6	4.00000	11.8
145435	20.50000	7.611178	46.00000	56	1025.3	4.50993	11.1
145436	19.40000	7.611178	43.00000	61	1027.1	4.50993	9.7
145437	19.20000	7.611178	37.00000	45	1026.7	4.50993	6.8
145438	19.80000	7.611178	39.00000	42	1024.1	4.50993	8.6
145439	18.80000	7.611178	31.00000	42	1022.4	4.50993	9.5
145440	20.70000	7.611178	37.00000	38	1023.9	4.50993	10.3
145441	20.80000	7.611178	43.00000	29	1025.9	4.50993	7.9
145442	21.70000	7.611178	43.00000	27	1025.0	4.50993	9.5
145443	20.00000	7.611178	39.00000	58	1026.7	4.50993	9.1
145444	19.80000	7.611178	43.00000	54	1027.7	4.50993	8.7
145445	18.60000	7.611178	46.00000	56	1027.2	4.50993	8.2
145446	19.10000	7.611178	33.00000	59	1026.2	4.50993	8.0
145447	19.70000	7.611178	46.00000	56	1025.8	4.50993	8.5
145448	21.20000	7.611178	35.00000	57	1023.0	4.50993	10.3
145449	21.40000	7.611178	22.00000	60	1022.8	4.50993	8.9
145450	22.50000	7.611178	19.00000	59	1021.4	4.50993	8.8
145451	24.30000	7.611178	24.00000	53	1020.0	4.50993	12.3
145452	23.40000	7.611178	31.00000	53	1023.0	4.50993	11.2
145453	20.70000	7.611178	41.00000	56	1024.3	7.00000	11.6
145454	20.60000	7.611178	35.00000	63	1023.3	4.50993	11.0
145455	21.80000	7.611178	31.00000	59	1021.2	4.50993	9.4
145456	23.40000	7.611178	31.00000	51	1020.3	4.50993	10.1
145457	25.30000	7.611178	22.00000	56	1019.1	4.50993	10.9
145458	26.90000	7.611178	37.00000	53	1016.8	4.50993	12.5
145459	27.00000	7.611178	28.00000	51	1016.5	2.00000	15.1
145460	23.22135	7.611178	40.03523	62	1017.9	8.00000	15.0

```
The following `from` values were not present in `x`: Yes

The following `from` values were not present in `x`: No
```

18. Splitting the data into train and test

In [83]:

```
#Data us split to test and train data in the ratio 75:25
weather_data10
library(caTools)
set.seed(123)
split = sample.split(weather_data10$RainTomorrow, SplitRatio = 0.75)
training_set = subset(weather_data10, split == TRUE)
test_set = subset(weather_data10, split == FALSE)
```

```
training_set$RainTomorrow
# Feature Scaling
training_set[-12] = scale(training_set[-12])
test_set[-12] = scale(test_set[-12])
```

A data.frame: 145460 × 12

	MaxTemp	Sunshine	WindGustSpeed	Humidity9am	Pressure3pm	Cloud3pm	Temp9am
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	22.9	7.611178	44.00000	71	1007.1	4.50993	16.9
2	25.1	7.611178	44.00000	44	1007.8	4.50993	17.2
3	25.7	7.611178	46.00000	38	1008.7	2.00000	21.0
4	28.0	7.611178	24.00000	45	1012.8	4.50993	18.1
5	32.3	7.611178	41.00000	82	1006.0	8.00000	17.8
6	29.7	7.611178	56.00000	55	1005.4	4.50993	20.6
7	25.0	7.611178	50.00000	49	1008.2	4.50993	18.1
8	26.7	7.611178	35.00000	48	1010.1	4.50993	16.3
9	31.9	7.611178	80.00000	42	1003.6	4.50993	18.3
10	30.1	7.611178	28.00000	58	1005.7	4.50993	20.1
11	30.4	7.611178	30.00000	48	1008.7	4.50993	20.4
12	21.7	7.611178	31.00000	89	1004.2	8.00000	15.9
13	18.6	7.611178	61.00000	76	993.0	8.00000	17.4
14	21.0	7.611178	44.00000	65	1001.8	7.00000	15.8
15	24.6	7.611178	40.03523	57	1008.7	4.50993	15.9
16	27.7	7.611178	50.00000	50	1010.3	4.50993	17.3
17	20.9	7.611178	22.00000	69	1010.4	1.00000	17.2
18	22.9	7.611178	63.00000	80	1002.2	1.00000	18.0
19	22.5	7.611178	43.00000	47	1009.7	2.00000	15.5
20	25.6	7.611178	26.00000	45	1017.1	4.50993	15.8
21	29.3	7.611178	24.00000	56	1014.8	4.50993	19.1
22	33.0	7.611178	43.00000	38	1008.1	1.00000	24.5
23	31.8	7.611178	41.00000	54	1005.7	4.50993	23.8
24	30.9	7.611178	33.00000	55	1008.2	4.50993	20.9
25	32.4	7.611178	43.00000	49	1010.1	4.50993	21.5
26	33.9	7.611178	35.00000	45	1007.6	1.00000	23.2
27	33.0	7.611178	57.00000	41	1003.6	1.00000	26.6
28	32.7	7.611178	48.00000	56	1001.7	4.50993	24.6
29	27.2	7.611178	46.00000	49	1004.2	4.50993	21.6
30	24.2	7.611178	50.00000	78	1003.4	8.00000	12.5
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
145431	28.00000	7.611178	28.00000	36	1016.9	4.50993	16.6
145432	24.60000	7.611178	33.00000	46	1018.8	4.50993	13.8
145433	22.20000	7.611178	37.00000	59	1021.7	8.00000	13.9

	MaxTemp	Sunshine	WindGustSpeed	Humidity9am	Pressure3pm	Cloud3pm	Temp9am
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
145434	22.70000	7.611178	35.00000	62	1023.6	4.00000	11.8
145435	20.50000	7.611178	46.00000	56	1025.3	4.50993	11.1
145436	19.40000	7.611178	43.00000	61	1027.1	4.50993	9.7
145437	19.20000	7.611178	37.00000	45	1026.7	4.50993	6.8
145438	19.80000	7.611178	39.00000	42	1024.1	4.50993	8.6
145439	18.80000	7.611178	31.00000	42	1022.4	4.50993	9.5
145440	20.70000	7.611178	37.00000	38	1023.9	4.50993	10.3
145441	20.80000	7.611178	43.00000	29	1025.9	4.50993	7.9
145442	21.70000	7.611178	43.00000	27	1025.0	4.50993	9.5
145443	20.00000	7.611178	39.00000	58	1026.7	4.50993	9.1
145444	19.80000	7.611178	43.00000	54	1027.7	4.50993	8.7
145445	18.60000	7.611178	46.00000	56	1027.2	4.50993	8.2
145446	19.10000	7.611178	33.00000	59	1026.2	4.50993	8.0
145447	19.70000	7.611178	46.00000	56	1025.8	4.50993	8.5
145448	21.20000	7.611178	35.00000	57	1023.0	4.50993	10.3
145449	21.40000	7.611178	22.00000	60	1022.8	4.50993	8.9
145450	22.50000	7.611178	19.00000	59	1021.4	4.50993	8.8
145451	24.30000	7.611178	24.00000	53	1020.0	4.50993	12.3
145452	23.40000	7.611178	31.00000	53	1023.0	4.50993	11.2
145453	20.70000	7.611178	41.00000	56	1024.3	7.00000	11.6
145454	20.60000	7.611178	35.00000	63	1023.3	4.50993	11.0
145455	21.80000	7.611178	31.00000	59	1021.2	4.50993	9.4
145456	23.40000	7.611178	31.00000	51	1020.3	4.50993	10.1
145457	25.30000	7.611178	22.00000	56	1019.1	4.50993	10.9
145458	26.90000	7.611178	37.00000	53	1016.8	4.50993	12.5
145459	27.00000	7.611178	28.00000	51	1016.5	2.00000	15.1
145460	23.22135	7.611178	40.03523	62	1017.9	8.00000	15.0

0 · 0 · 0 · 0 · 1 · 0 · 1 · 1 · 0 · 0 · 1 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 1 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 ·
0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 1 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 1 · 0 · 0 · 0 · 0 ·
0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 1 · 0 · 1 · 1 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 1 · 0 · 0 ·
0 · 0 · 1 · 1 · 0 · 0 · 0 · 0 · 1 · 1 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 1 · 1 · 1 · 1 · 0 · 0 · 0 · 0 · 0 ·
0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 1 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 1 · 0 · 0 · 0 · 0 · 0 · 1 · 1 · 1 · 1 · 0 · 0 · 1 ·
1 · 1 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 1 · 0 · 1 · 0 · 1 · 0 · 1 · 1 · 1 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 1 ·
0 · 0 · 0 · 0 · 0 · 0 · 1 · 0 · 0 · 1 · 0 · 1 · 0 · 0 · 0 · 0 · 0 · 1 · 0 · 0 · 1 · 0 · 1 · 0 · 1 · 1 · ⋯ · 0 · 0 ·
0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 1 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 ·
0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 1 · 1 · 0 · 0 · 0 ·
0 · 1 · 0 · 0 · 0 · 1 · 0 · 0 · 0 · 0 · 1 · 1 · 1 · 1 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 1 · 1 · 0 · 0 ·
1 · 1 · 1 · 1 · 1 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 1 · 0 · 0 · 1 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 ·
0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 ·

0 · 0 · 0 · 0 · 1 · 1 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 ·
0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0 · 0

▶ **Levels**:

19. Multiple Linear regression

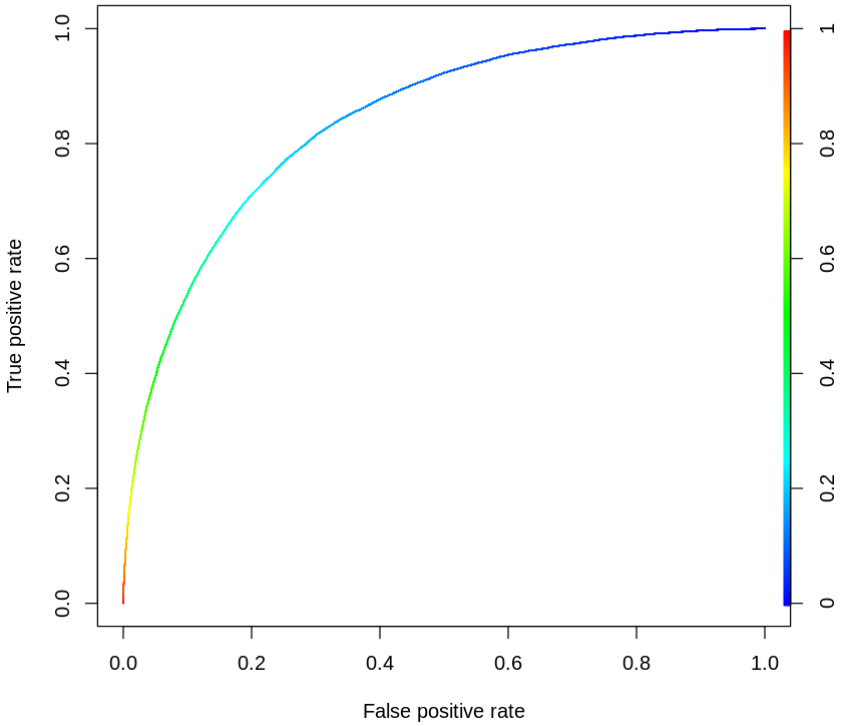
In [84]:

```
##Multiple Linear Regression
# Fitting Logistic Regression to the Training set
classifier = glm(formula = RainTomorrow ~ .,
                 family = binomial,
                 data = training_set)
summary(classifier)
#Predict using test set
prob_pred = predict(classifier, type = 'response')
prob_prd_glm=predict(classifier, type = 'response', newdata = test_set[-12])
y_pred = ifelse(prob_prd_glm > 0.5, 1, 0)
```

```
Call:
glm(formula = RainTomorrow ~ ., family = binomial, data = training_set)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.6960  -0.6070  -0.3547  -0.1358   3.2772  

Coefficients:
               Estimate Std. Error  z value Pr(>|z|)    
(Intercept)   -1.811110   0.010849 -166.945  < 2e-16 ***
MaxTemp       -0.410266   0.034912  -11.751  < 2e-16 ***
Sunshine      -0.356540   0.010266  -34.732  < 2e-16 ***
WindGustSpeed  0.449281   0.009438   47.605  < 2e-16 ***
Humidity9am    0.781716   0.011938   65.483  < 2e-16 ***
Pressure3pm   -0.484713   0.010196  -47.537  < 2e-16 ***
Cloud3pm       0.348153   0.011372   30.615  < 2e-16 ***
Temp9am        1.112824   0.021211   52.465  < 2e-16 ***
Temp3pm       -0.814044   0.030969  -26.285  < 2e-16 ***
WindGustDir   -0.002227   0.011257   -0.198    0.843    
WindDir9am    -0.066891   0.009789   -6.833 8.29e-12 ***
WindDir3pm    -0.071763   0.011250   -6.379 1.78e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 114730  on 109094  degrees of freedom
Residual deviance:  84024  on 109083  degrees of freedom
AIC: 84048

Number of Fisher Scoring iterations: 5
```

In [85]:

```
library(ROCR)
ROCRpred <- prediction(prob_pred, training_set$RainTomorrow)
ROCRperf <- performance(ROCRpred, 'tpr','fpr')
plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7))
```



In [86]:

```
cm3 = table(test_set[,12], y_pred)
confusionMatrix(cm3)
```

```
Confusion Matrix and Statistics

   y_pred
      0     1
 0 26835  1561
 1  4730  3239

               Accuracy : 0.827
                 95% CI : (0.8231, 0.8309)
    No Information Rate : 0.868
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4101

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8502
            Specificity : 0.6748
         Pos Pred Value : 0.9450
         Neg Pred Value : 0.4064
             Prevalence : 0.8680
         Detection Rate : 0.7379
   Detection Prevalence : 0.7809
      Balanced Accuracy : 0.7625

       'Positive' Class : 0
```

In [87]:

```
summary(training_set)
training_set$RainTomorrow=as.numeric(training_set$RainTomorrow)
is.numeric(training_set$RainTomorrow)
```

```
   MaxTemp            Sunshine          WindGustSpeed       Humidity9am
Min.   :-3.81702   Min.   :-2.7952127   Min.   :-2.51744   Min.   :-3.659270
1st Qu.:-0.73611   1st Qu.: 0.0006276   1st Qu.:-0.68728   1st Qu.:-0.631617
Median :-0.07188   Median : 0.0006276   Median :-0.07722   Median : 0.005784
Mean   : 0.00000   Mean   : 0.0000000   Mean   : 0.00000   Mean   : 0.000000
3rd Qu.: 0.70542   3rd Qu.: 0.3638552   3rd Qu.: 0.45657   3rd Qu.: 0.749418
Max.   : 3.51780   Max.   : 2.5311228   Max.   : 7.24344   Max.   : 1.652402
 Pressure3pm          Cloud3pm           Temp9am           Temp3pm
Min.   :-5.723076   Min.   :-2.160674   Min.   :-3.5929   Min.   :-3.91477
1st Qu.:-0.623766   1st Qu.:-0.245446   1st Qu.:-0.7255   1st Qu.:-0.72706
Median :-0.000468   Median :-0.001287   Median :-0.0280   Median :-0.05442
Mean   : 0.000000   Mean   : 0.000000   Mean   : 0.0000   Mean   : 0.00000
3rd Qu.: 0.621065   3rd Qu.: 0.712169   3rd Qu.: 0.7005   3rd Qu.: 0.66208
Max.   : 3.650655   Max.   : 2.148591   Max.   : 3.5989   Max.   : 3.65970
 WindGustDir        WindDir9am          WindDir3pm      RainTomorrow
Min.   :-1.7252   Min.   :-1.5489393   Min.   :-1.7188   0:85187
1st Qu.:-0.8727   1st Qu.:-0.8846985   1st Qu.:-0.8373   1:23908
Median : 0.1930   Median : 0.0009559   Median : 0.0443
Mean   : 0.0000   Mean   : 0.0000000   Mean   : 0.0000
3rd Qu.: 1.0455   3rd Qu.: 0.8866103   3rd Qu.: 0.9259
Max.   : 1.4717   Max.   : 1.7722647   Max.   : 1.5871
```

TRUE

20. Support Vector Machine

In [112…

```
#SVM
library(e1071)
svmfit = svm(formula = RainTomorrow ~ .,
             data = training_set,
             type = 'C-classification',
             kernel = 'linear')

# Predicting the Test set results
y_pred_svm = predict(svmfit, newdata = test_set[-12])
```

In [116…

```
y_pred_svm1 = ifelse(y_pred_svm == 1, 0, 1)
```

In [120…

```
# Making the Confusion Matrix
cm_svm = table(test_set[,12], y_pred_svm1)
confusionMatrix(cm_svm)
```

```
Confusion Matrix and Statistics

   y_pred_svm1
       0     1
 0 27226  1170
 1  5205  2764

               Accuracy : 0.8247
                 95% CI : (0.8207, 0.8286)
    No Information Rate : 0.8918
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3737

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8395
            Specificity : 0.7026
```

```
          Pos Pred Value : 0.9588
          Neg Pred Value : 0.3468
              Prevalence : 0.8918
          Detection Rate : 0.7487
    Detection Prevalence : 0.7809
       Balanced Accuracy : 0.7710

        'Positive' Class : 0
```

21. Naive Bayes

In [89]:

```
#Naive Bayes

# Loading package
library(e1071)
library(caTools)
library(caret)

set.seed(120)  # Setting Seed
classifier_cl <- naiveBayes(RainTomorrow ~ ., data = training_set)
classifier_cl

y_pred_nb = predict(classifier_cl, newdata = test_set[-12])
```

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
        1         2
0.7808516 0.2191484

Conditional probabilities:
   MaxTemp
Y         [,1]      [,2]
  1  0.0828858 0.9902072
  2 -0.2953318 0.9786641

   Sunshine
Y         [,1]      [,2]
  1  0.1705850 0.9116166
  2 -0.6078144 1.0625088

   WindGustSpeed
Y         [,1]      [,2]
  1 -0.1176828 0.9172153
  2  0.4193176 1.1577563

   Humidity9am
Y         [,1]      [,2]
  1 -0.1324883 0.9985110
  2  0.4720712 0.8516091

   Pressure3pm
Y         [,1]      [,2]
  1  0.1130348 0.9538024
  2 -0.4027561 1.0554217

   Cloud3pm
Y         [,1]      [,2]
```

```
  1 -0.1581406 0.9807340
  2  0.5634734 0.8540434

   Temp9am
Y          [,1]      [,2]
  1  0.01324901 1.0030146
  2 -0.04720776 0.9877612

   Temp3pm
Y          [,1]      [,2]
  1  0.09970539 0.9866396
  2 -0.35526195 0.9659080

   WindGustDir
Y          [,1]      [,2]
  1 -0.02509732 1.0072888
  2  0.08942468 0.9683325

   WindDir9am
Y          [,1]     [,2]
  1 -0.01917708 1.004012
  2  0.06833018 0.982555

   WindDir3pm
Y          [,1]      [,2]
  1 -0.01546647 1.0050289
  2  0.05510883 0.9799102
```

In [108…

```
y_pred_nb1 = ifelse(y_pred_nb == 1, 0, 1)
```

In [111…

```
# Making the Confusion Matrix
cm_nb = table(test_set[, 12], y_pred_nb1)

#Model Evaluation
confusionMatrix(cm_nb)
```

```
Confusion Matrix and Statistics

   y_pred_nb1
       0     1
  0 24654  3742
  1  4073  3896

               Accuracy : 0.7851
                 95% CI : (0.7808, 0.7893)
    No Information Rate : 0.79
    P-Value [Acc > NIR] : 0.9886935

                  Kappa : 0.3625

 Mcnemar's Test P-Value : 0.0001893

            Sensitivity : 0.8582
            Specificity : 0.5101
         Pos Pred Value : 0.8682
         Neg Pred Value : 0.4889
             Prevalence : 0.7900
         Detection Rate : 0.6780
   Detection Prevalence : 0.7809
      Balanced Accuracy : 0.6841
```

```
'Positive' Class : 0
```

Conclusion

Real-time environmental monitoring and weather prediction was successfully implemented and visuazlied using Multiple Linear Regression, SVM and Naïve Bayes classification