Tugas Besar IF2220 - Probabilitas dan Statistika

Part 1: Statistics Description

Anggota:
13521116 - Juan Christopher Santoso
13521162 - Antonio Natthan Krishna

Data Preparation and Data Description (Run this Code First)

In [1]:
```
# Import Dataset
df <- read.csv("..\\test\\anggur.csv")

# Data Statistics
properties <- c("Rows", "Columns")
value <- c(nrow(df), ncol(df))
cbind(properties, value)

# List of Columns
columns_index <- c(1:ncol(df))
columns_name <- colnames(df)

# Display List
cbind(columns_index, columns_name)
```

A matrix: 2 × 2 of type chr

properties	value
Rows	1000
Columns	12

A matrix: 12 × 2 of type chr

columns_index	columns_name
1	fixed.acidity
2	volatile.acidity
3	citric.acid
4	residual.sugar
5	chlorides
6	free.sulfur.dioxide
7	total.sulfur.dioxide
8	density
9	pH
10	sulphates
11	alcohol
12	quality

1. Mean

In [2]:
```
cat("Column's Mean:\n")

columns_mean <- colMeans(df)
cbind(columns_mean)
```

```
Column's Mean:
```

A matrix: 12 × 1 of type dbl

	columns_mean
fixed.acidity	7.15253000
volatile.acidity	0.52083850
citric.acid	0.27051700
residual.sugar	2.56710368
chlorides	0.08119515
free.sulfur.dioxide	14.90767925
total.sulfur.dioxide	40.29015000
density	0.99592530
pH	3.30361000
sulphates	0.59839000
alcohol	10.59228000
quality	7.95800000

2. Median

In [3]:
```
cat("Column's Median: \n")

columns_median = c(1:ncol(df))
for (i in columns_index){
    columns_median[i] <- median(df[,columns_name[i]])
}
cbind(columns_name, columns_median)
```

```
Column's Median:
```

A matrix: 12 × 2 of type chr

columns_name	columns_median
fixed.acidity	7.15
volatile.acidity	0.52485
citric.acid	0.2722
residual.sugar	2.51943027286579
chlorides	0.0821669021645236
free.sulfur.dioxide	14.8603462365689
total.sulfur.dioxide	40.19
density	0.996
pH	3.3
sulphates	0.595
alcohol	10.61
quality	8

3. Mode

In [4]:
```
getmode <- function(v) {
    uniquev <- unique(v)
    maxCount <- 0
    maxElmt <- 0
    for (j in c(1:nrow(df))){
        temp <- v[i]
        count <- 0
        for (k in c(i:nrow(df))){
            if (temp == v[k]){
                count <- count + 1
            }
        }
        if (count > maxCount){
            maxCount <- count
            maxElmt <- temp
        }
    }
    return(maxElmt)
}

cat("Column's Mode: \n")

columns_mode = c(1:ncol(df))
for (i in columns_index){
    columns_mode[i] <- getmode(df[,columns_name[i]])
}

cbind(columns_name, columns_mode)
```

```
Column's Mode:
```

A matrix: 12 × 2 of type chr

columns_name	columns_mode
fixed.acidity	5.9
volatile.acidity	0.5768
citric.acid	0.3248
residual.sugar	3.37181458927355
chlorides	0.0663785866479429
free.sulfur.dioxide	12.2321700848591
total.sulfur.dioxide	44.26
density	0.9999
pH	3.27
sulphates	0.51
alcohol	10.52
quality	9

4. Standard Deviation

In [5]:
```
cat("Column's Standard Deviation: \n")

columns_std = c(1:ncol(df))

for (i in columns_index) {
    columns_std[i] <- sd(df[,columns_name[i]])
}

cbind(columns_name, columns_std)
```

```
Column's Standard Deviation:
```

A matrix: 12 × 2 of type chr

columns_name	columns_std
fixed.acidity	1.20159757649383
volatile.acidity	0.0958482740553495
citric.acid	0.0490983714707635
residual.sugar	0.987915436504693
chlorides	0.0201106472439967
free.sulfur.dioxide	4.88809970575656
total.sulfur.dioxide	9.9657673762183
density	0.00202018094264871
pH	0.104875482200402
sulphates	0.100819007991412
alcohol	1.51070600522876
quality	0.902801778382747

5. Variance

In [6]:
```
cat("Column's Variance: \n")

columns_var = c(1:ncol(df))

for (i in columns_index) {
    columns_var[i] <- sd(df[,columns_name[i]]) ^ 2
}

cbind(columns_name, columns_var)
```

```
Column's Variance:
```

A matrix: 12 × 2 of type chr

columns_name	columns_var
fixed.acidity	1.44383673583584
volatile.acidity	0.00918689163938939
citric.acid	0.00241065008108108
residual.sugar	0.975976909684258
chlorides	0.000404438132572474
free.sulfur.dioxide	23.8935187334174
total.sulfur.dioxide	99.3165193968969
density	4.08113104104104e-06
pH	0.0109988667667668
sulphates	0.0101644723723724
alcohol	2.28223263423423
quality	0.815051051051051

6. Range

In [7]:
```
cat("Column's Range: \n")

columns_range = c(1:ncol(df))

for (i in columns_index) {
    columns_range[i] <- max(df[,columns_name[i]]) - min(df[,columns_name[i]])
}

cbind(columns_name, columns_range)
```

```
Column's Range:
```

A matrix: 12 × 2 of type chr

columns_name	columns_range
fixed.acidity	8.17
volatile.acidity	0.6652
citric.acid	0.2929
residual.sugar	5.51820040970786
chlorides	0.125635130265349
free.sulfur.dioxide	27.2678469010989
total.sulfur.dioxide	66.81
density	0.0137999999999999
pH	0.74
sulphates	0.67
alcohol	8.99
quality	5

7. Minimum Value

In [8]:
```
cat("Column's Minimum Value: \n")

columns_min = c(1:ncol(df))

for (i in columns_index) {
    columns_min[i] <- min(df[,columns_name[i]])
}

cbind(columns_name, columns_min)
```

```
Column's Minimum Value:
```

A matrix: 12 × 2 of type chr

columns_name	columns_min
fixed.acidity	3.32
volatile.acidity	0.1399
citric.acid	0.1167
residual.sugar	0.032554525015195
chlorides	0.0151224391657095
free.sulfur.dioxide	0.194678523326937
total.sulfur.dioxide	3.15
density	0.9888
pH	2.97
sulphates	0.29
alcohol	6.03
quality	5

8. Maximum Value

In [9]:
```
cat("Column's Maximum Value: \n")

columns_max = c(1:ncol(df))

for (i in columns_index) {
    columns_max[i] <- max(df[,columns_name[i]])
}

cbind(columns_name, columns_max)
```

```
Column's Maximum Value:
```

A matrix: 12 × 2 of type chr

columns_name	columns_max
fixed.acidity	11.49
volatile.acidity	0.8051
citric.acid	0.4096
residual.sugar	5.55075493472306
chlorides	0.140757569431058
free.sulfur.dioxide	27.4625254244258
total.sulfur.dioxide	69.96
density	1.0026
pH	3.71
sulphates	0.96
alcohol	15.02
quality	10

9. Quartile

In [10]:
```
cat("Column's Quartiles: \n")

columns_q25 = c(1:ncol(df))
columns_q50 = c(1:ncol(df))
columns_q75 = c(1:ncol(df))

for (i in columns_index) {
    columns_q25[i] <- quantile(df[,columns_name[i]], probs= 0.25)
    columns_q50[i] <- quantile(df[,columns_name[i]], probs= 0.5)
    columns_q75[i] <- quantile(df[,columns_name[i]], probs= 0.75)
}

cbind(columns_name, columns_q25, columns_q50, columns_q75)
```

```
Column's Quartiles:
```

A matrix: 12 × 4 of type chr

columns_name	columns_q25	columns_q50	columns_q75
fixed.acidity	6.3775	7.15	8
volatile.acidity	0.4561	0.52485	0.585375
citric.acid	0.2378	0.2722	0.302325
residual.sugar	1.89632994348868	2.51943027286579	3.22087348282979
chlorides	0.0665736319097736	0.0821669021645236	0.0953115014855626
free.sulfur.dioxide	11.4267169494576	14.8603462365689	18.313097915395
total.sulfur.dioxide	33.785	40.19	47.0225
density	0.9946	0.996	0.9972
pH	3.23	3.3	3.37
sulphates	0.53	0.595	0.67
alcohol	9.56	10.61	11.6225
quality	7	8	9

10. Interquartile Range

In [11]:
```
cat("Column's Interquartiles Range: \n")

columns_qrange = c(1:ncol(df))

for (i in columns_index) {
    tempQuantile <- quantile(df[,columns_name[i]], probs= c(0.25, 0.75))
    columns_qrange[i] <- tempQuantile[2] - tempQuantile[1]
}

cbind(columns_name, columns_qrange)
```

```
Column's Interquartiles Range:
```

A matrix: 12 × 2 of type chr

columns_name	columns_qrange
fixed.acidity	1.6225
volatile.acidity	0.129275
citric.acid	0.064525
residual.sugar	1.3245435393411
chlorides	0.028737869575789
free.sulfur.dioxide	6.88638096593739
total.sulfur.dioxide	13.2375
density	0.00259999999999994
pH	0.14
sulphates	0.14
alcohol	2.0625
quality	2

11. Skewness

In [12]:
```
cat("Column's Skewness: \n")

columns_skewness = c(1:ncol(df))

for (i in columns_index){
    tempMean <- mean(df[,columns_name[i]])
    tempStddev <- sd(df[,columns_name[i]])
    count <- 0
    for (j in c(1:nrow(df))){
        count <- count + (df[j, columns_name[i]] - tempMean) ^3
    }
    columns_skewness[i] <- count / (nrow(df) * (tempStddev ^3))
}

cbind(columns_name, columns_skewness)
```

```
Column's Skewness:
```

A matrix: 12 × 2 of type chr

columns_name	columns_skewness
fixed.acidity	-0.0287919975632131
volatile.acidity	-0.197105997910775
citric.acid	-0.045439421661083
residual.sugar	0.132240437207526
chlorides	-0.0511654421670584
free.sulfur.dioxide	0.0071090390040008
total.sulfur.dioxide	-0.0239878948518848
density	-0.0766522945530875
pH	0.147229872668135
sulphates	0.148751602565093
alcohol	-0.0189344680909653
quality	-0.0887871070594255

12. Kurtosis

In [13]:
```
cat("Column's Kurtosis: \n")

columns_kurtosis = c(1:ncol(df))

for (i in columns_index){
    tempMean <- mean(df[,columns_name[i]])
    tempSd <- sd(df[,columns_name[i]])
    numerator <- 0
    for (j in c(1:nrow(df))){
        numerator <- numerator + (df[j, columns_name[i]] - tempMean) ^ 4
    }

    columns_kurtosis[i] <- numerator / ( nrow(df) * tempSd ^ 4)
}

cbind(columns_name, columns_kurtosis)
```

```
Column's Kurtosis:
```

A matrix: 12 × 2 of type chr

columns_name	columns_kurtosis
fixed.acidity	2.96886355314195
volatile.acidity	3.14874381854168
citric.acid	2.88407259945928
residual.sugar	2.94534102928964
chlorides	2.74323396581525
free.sulfur.dioxide	2.62560551750591
total.sulfur.dioxide	3.05152426622451
density	3.0042723321488
pH	3.0683656154671
sulphates	3.05238768805287
alcohol	2.85720916960675
quality	3.09555588797803

Tugas Besar IF2220 - Probabilitas dan Statistika

Part 2: Data Visualization

Anggota:
13521116 - Juan Christopher Santoso
13521162 - Antonio Natthan Krishna

Data Preparation and Data Description

In [1]:
```
# Import Dataset
df <- read.csv("..\\test\\anggur.csv")

# Data Statistics
properties <- c("Rows", "Columns")
value <- c(nrow(df), ncol(df))
cbind(properties, value)

# List of Columns
columns_index <- c(1:ncol(df))
columns_name <- colnames(df)

# Display List
cbind(columns_index, columns_name)
```

A matrix: 2 × 2 of type chr

properties	value
Rows	1000
Columns	12

A matrix: 12 × 2 of type chr

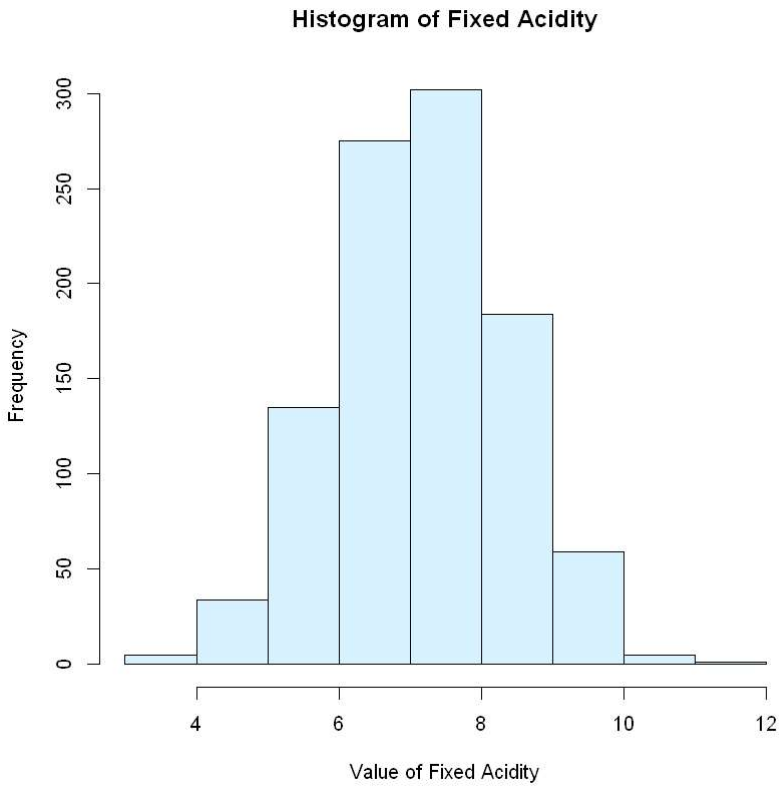
columns_index	columns_name
1	fixed.acidity
2	volatile.acidity
3	citric.acid
4	residual.sugar
5	chlorides
6	free.sulfur.dioxide
7	total.sulfur.dioxide
8	density
9	pH
10	sulphates
11	alcohol
12	quality

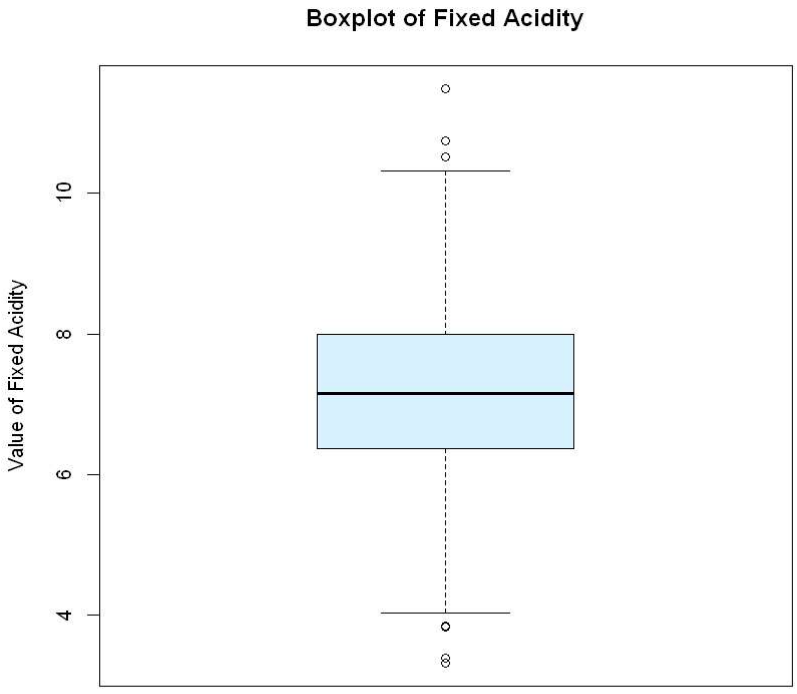
Global Functions used for Data Visualization

In [2]:
```
getHist <- function(v, name, color) {
    hist(v,
    main = paste("Histogram of", name),
    xlab = paste("Value of", name),
    ylab = "Frequency",
    col = color)
}

getBoxPlot <- function(v, name, color){
    boxplot(v,
    main = paste("Boxplot of", name),
    ylab = paste("Value of", name),
    col = "#D8F2FF"
    )
}
```

1. Kolom *fixed.acidity*

In [3]:
```
fixed_acidity <- df$fixed.acidity

getHist(fixed_acidity,"Fixed Acidity", "#D8F2FF")
getBoxPlot(fixed_acidity, "Fixed Acidity", "#D8F2FF")
```



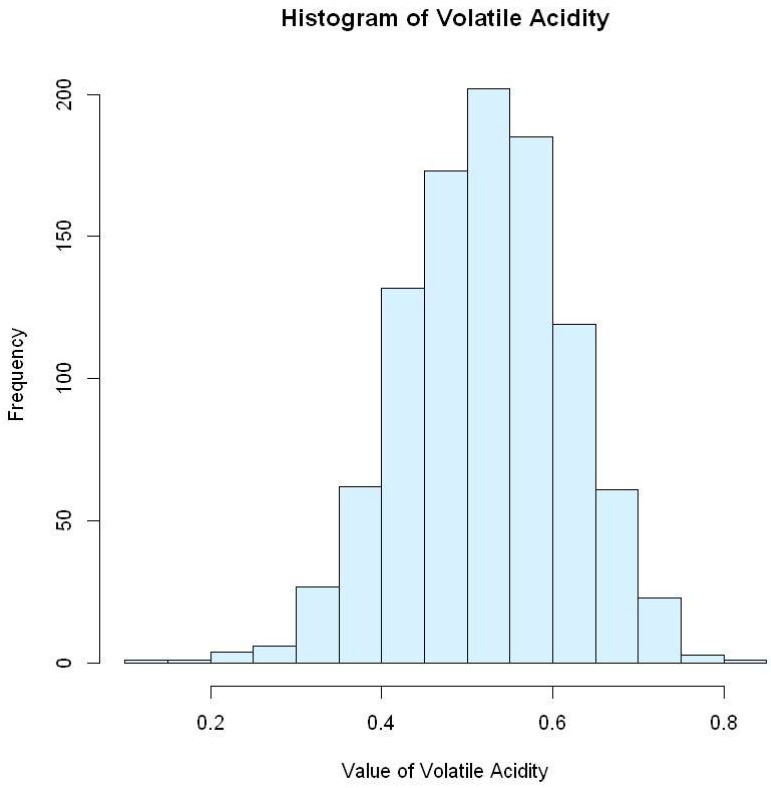


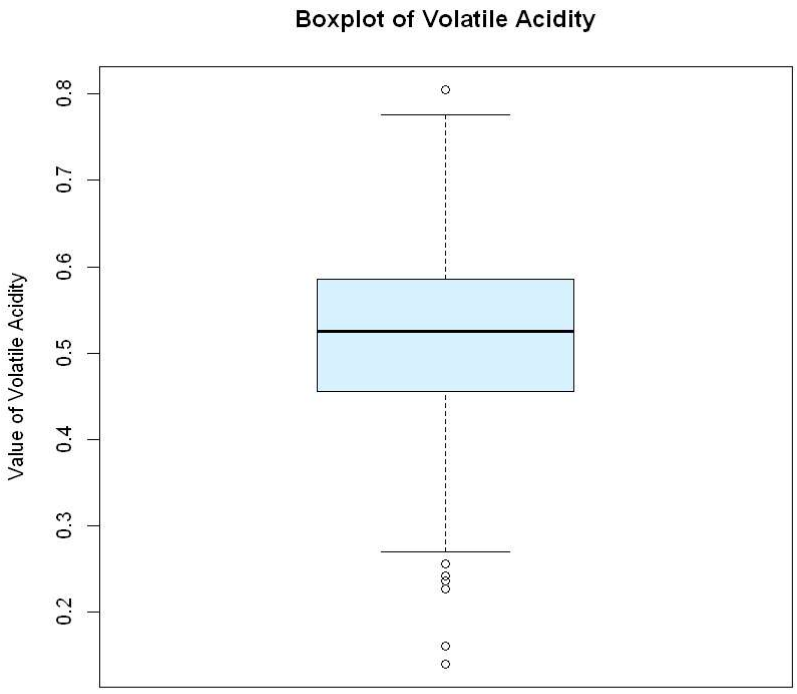
Deskripsi

Kolom Fixed Acidity secara kualitatif hampir terdistribusi normal, terdapat beberapa outlier (nilai tidak wajar) berdasarkan boxplot yang terlalu besar dan terlalu kecil

2. Kolom *volatile.acidity*

In [4]:
```
volatile_acidity <- df$volatile.acidity

getHist(volatile_acidity,"Volatile Acidity", "#D8F2FF")
getBoxPlot(volatile_acidity, "Volatile Acidity", "#D8F2FF")
```



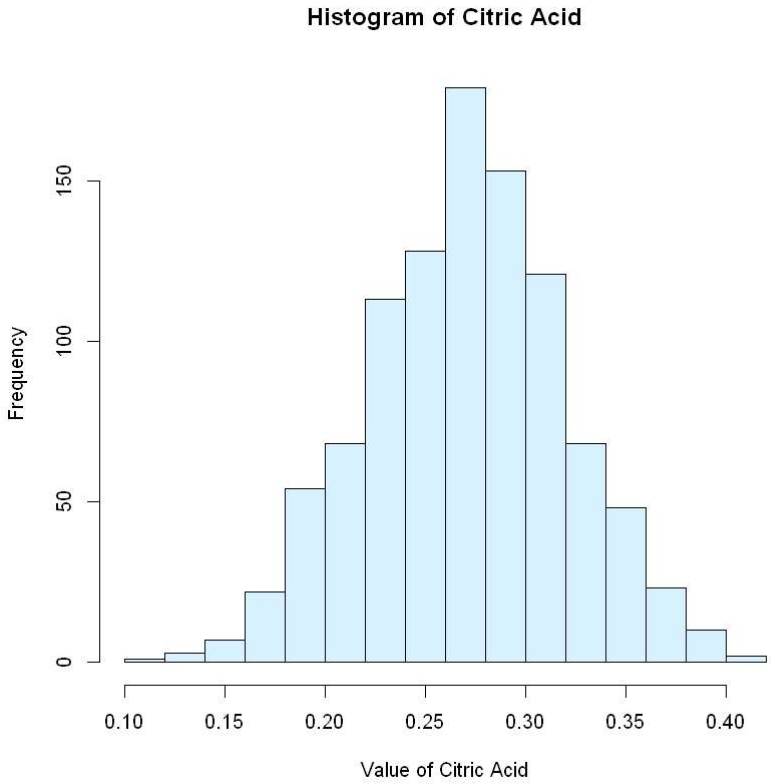


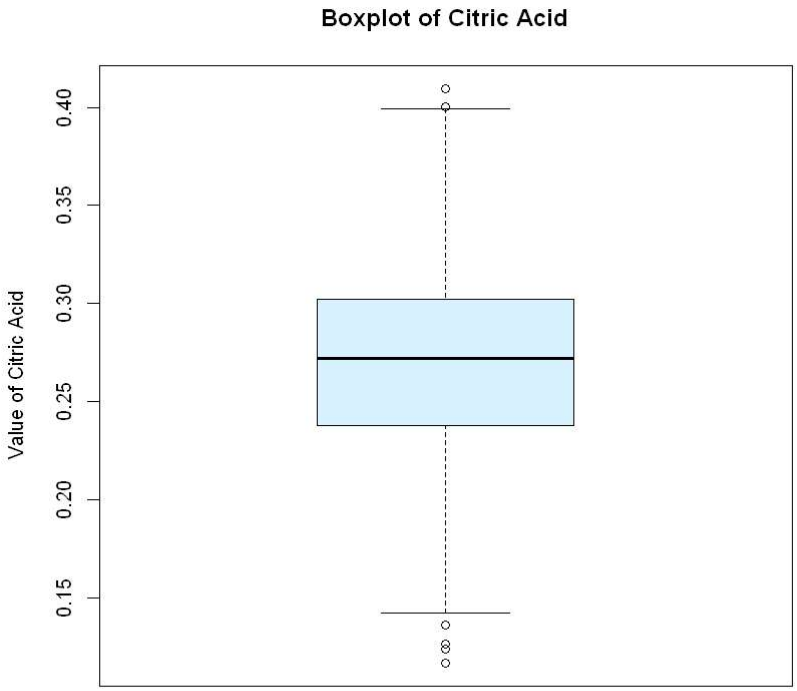
Deskripsi

Kolom Volatile Acidity secara kualitatif hampir terdistribusi normal. Terdapat satu nilai outlier atas dan cukup banyak data yang berada pada outlier yang bawah.

3. Kolom *citric.acid*

In [5]:
```
citric_acid <- df$citric.acid

getHist(citric_acid,"Citric Acid", "#D8F2FF")
getBoxPlot(citric_acid, "Citric Acid", "#D8F2FF")
```



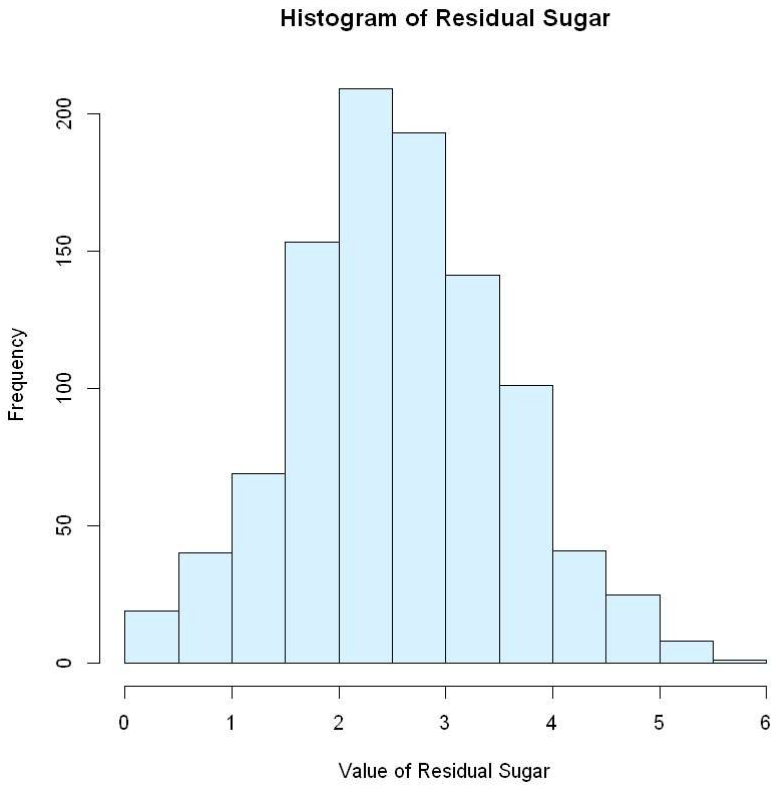


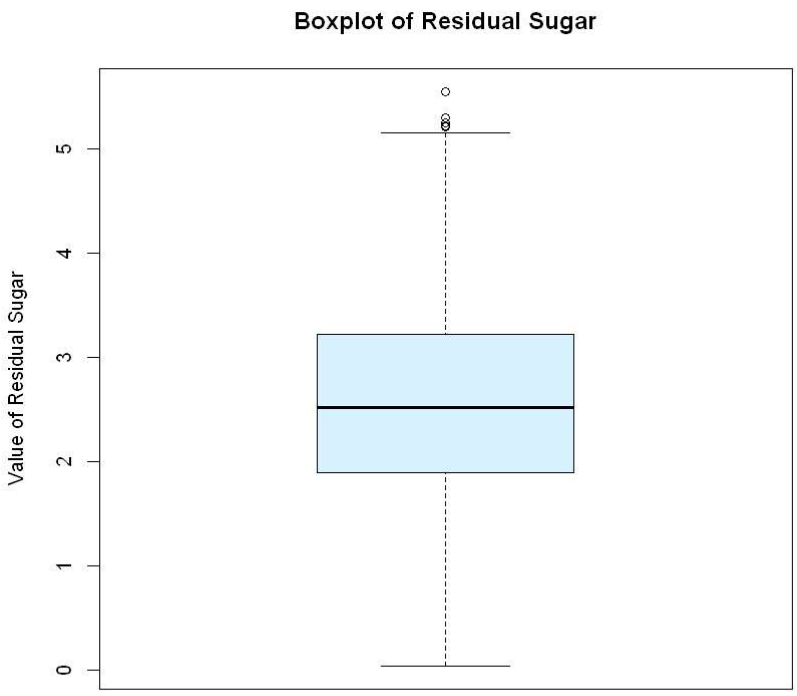
Deskripsi

Kolom Citric Acid secara kualitatif hampir terdistribusi normal, terdapat beberapa outlier (nilai tidak wajar), terdapat 2 buah data yang berada pada outlier atas dan empat data yang terdapat pada outlier bawah

4. Kolom *residual.sugar*

In [6]:
```
residual_sugar <- df$residual.sugar

getHist(residual_sugar,"Residual Sugar", "#D8F2FF")
getBoxPlot(residual_sugar, "Residual Sugar", "#D8F2FF")
```





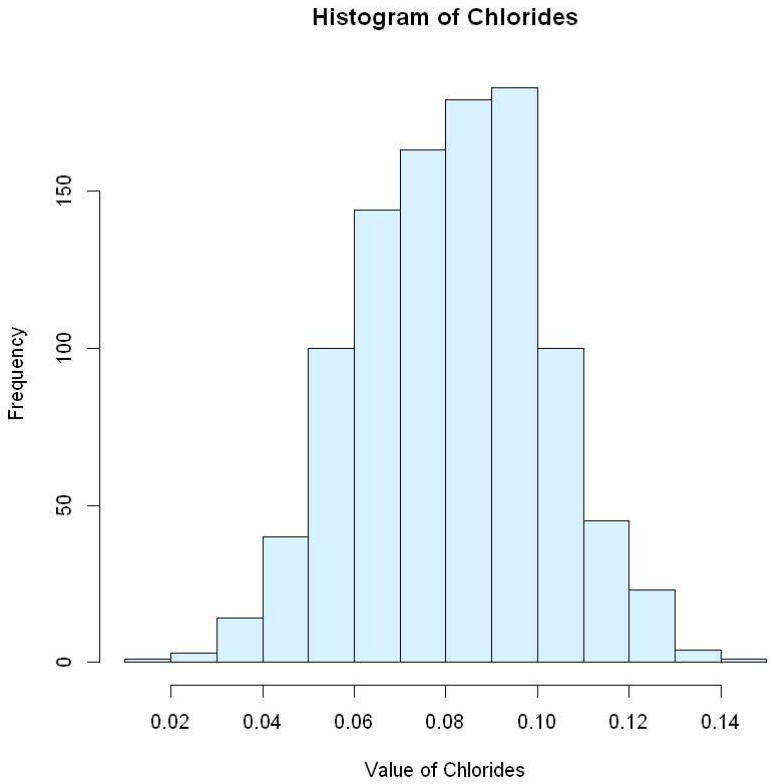
Deskripsi

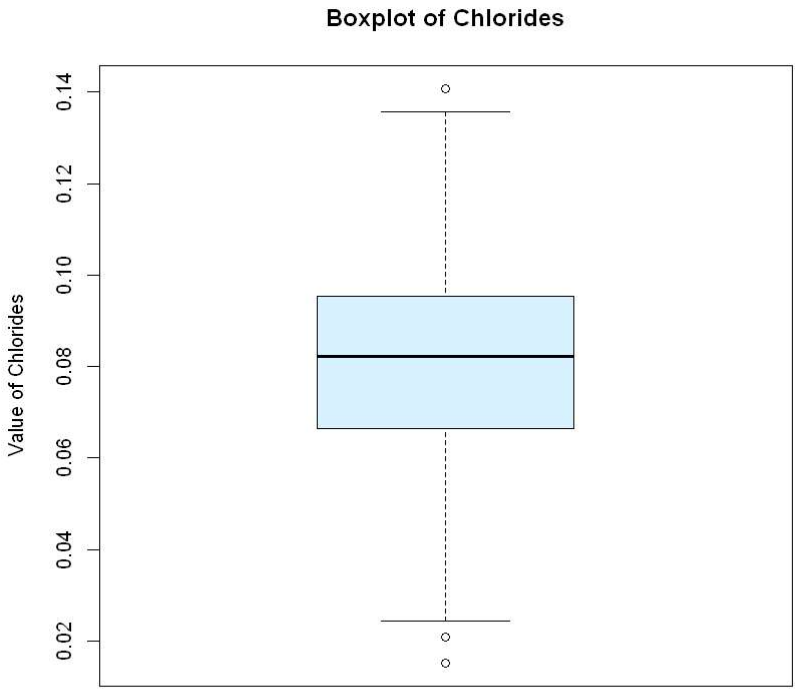
Deskripsi

Kolom Residual Sugar secara kualitatif hampir terdistribusi normal, terdapat cukup banyak data yang merupakan outlier atas berdasarkan boxplot. Tidak ada data yang menjadi outlier bawah

5. Kolom *chlorides*

In [7]:
```
chlorides <- df$chlorides

getHist(chlorides,"Chlorides", "#D8F2FF")
getBoxPlot(chlorides, "Chlorides", "#D8F2FF")
```



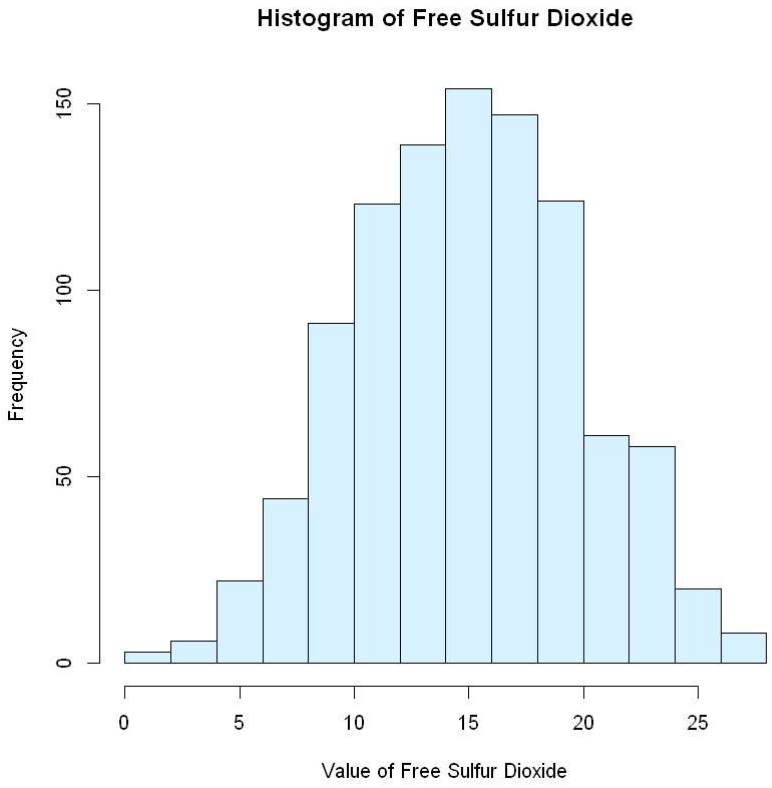


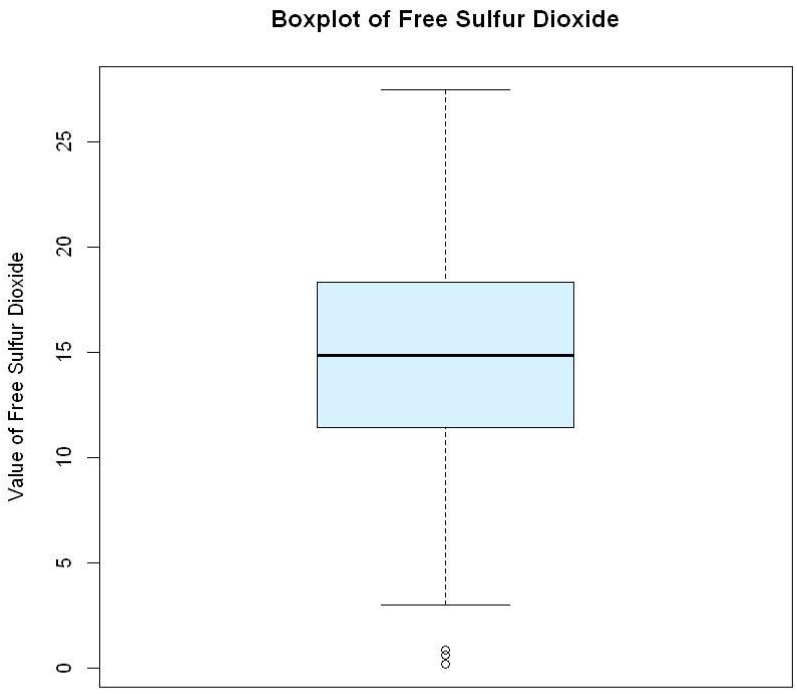
Deskripsi

Kolom Chlorides secara kualitatif hampir terdistribusi normal, terdapat 1 outlier atas dan 2 outlier bawah.

6. Kolom *free.sulfur.dioxide*

In [8]:
```
free_sulfur_dioxide <- df$free.sulfur.dioxide

getHist(free_sulfur_dioxide,"Free Sulfur Dioxide", "#D8F2FF")
getBoxPlot(free_sulfur_dioxide, "Free Sulfur Dioxide", "#D8F2FF")
```



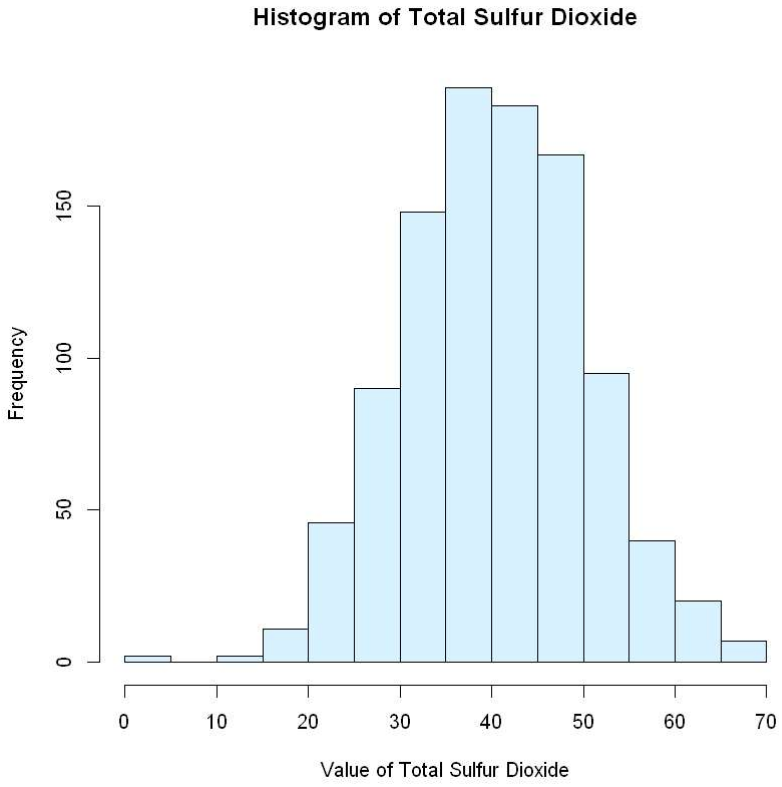


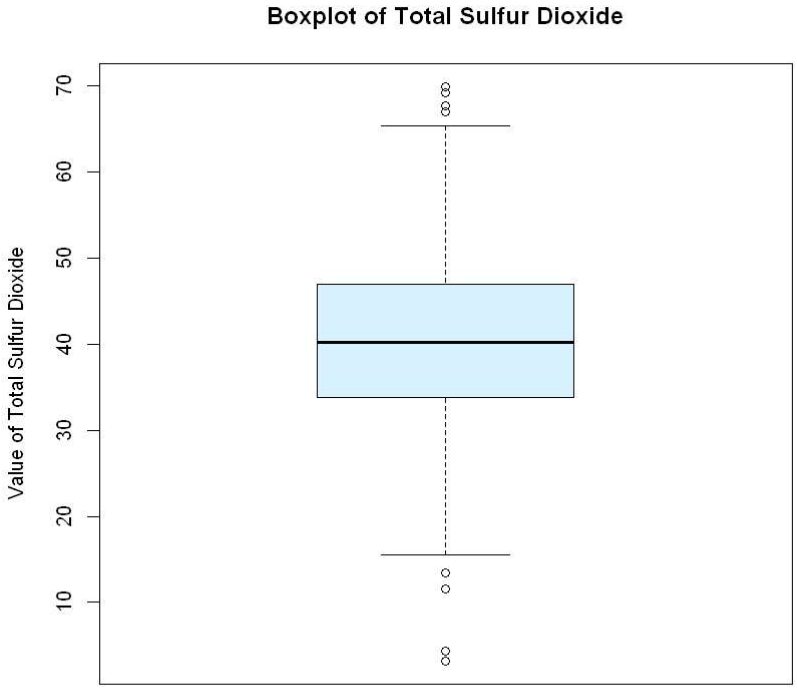
Deskripsi

Kolom Free Sulfur Dioxide secara kualitatif hampir terdistribusi normal, tidak terdapat outlier atas namun terdapat beberapa data yang merupakan outlier bawah.

7. Kolom *total.sulfur.dioxide*

In [9]:
```
total_sulfur_dioxide <- df$total.sulfur.dioxide

getHist(total_sulfur_dioxide,"Total Sulfur Dioxide", "#D8F2FF")
getBoxPlot(total_sulfur_dioxide, "Total Sulfur Dioxide", "#D8F2FF")
```



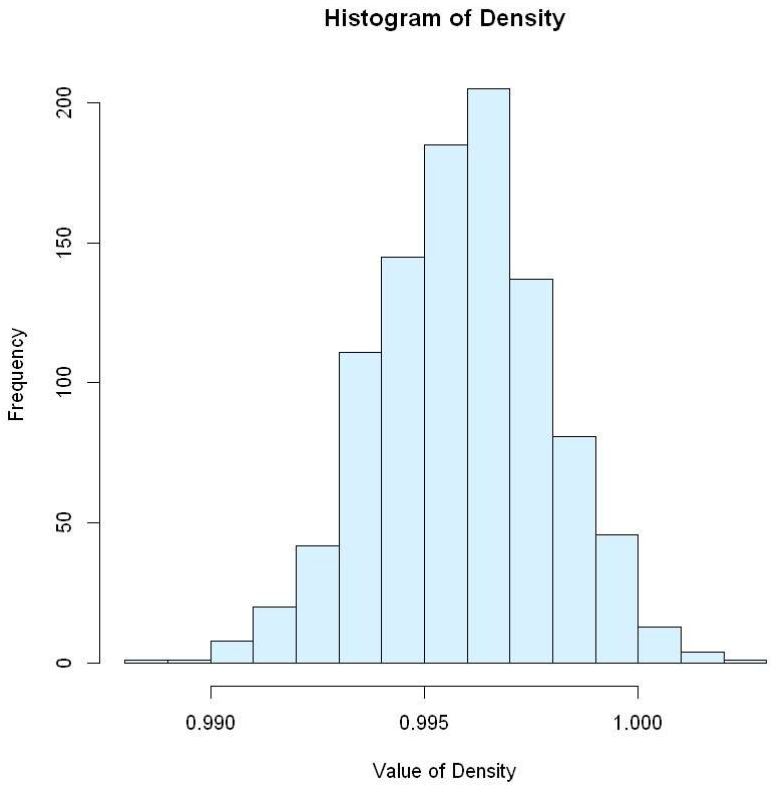


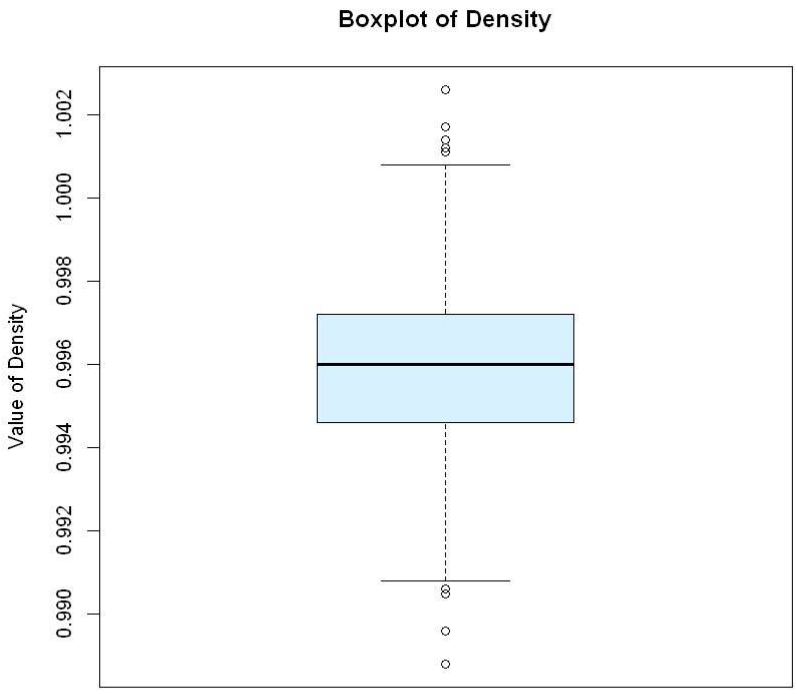
Deskripsi

Kolom Total Sulfur Dioxide secara kualitatif hampir terdistribusi normal, terdapat beberapa outlier atas dan bawah yang jumlahnya relatif lebih banyak dibandingkan kolom lain

8. Kolom *density*

In [10]:
```
density <- df$density

getHist(density,"Density", "#D8F2FF")
getBoxPlot(density, "Density", "#D8F2FF")
```



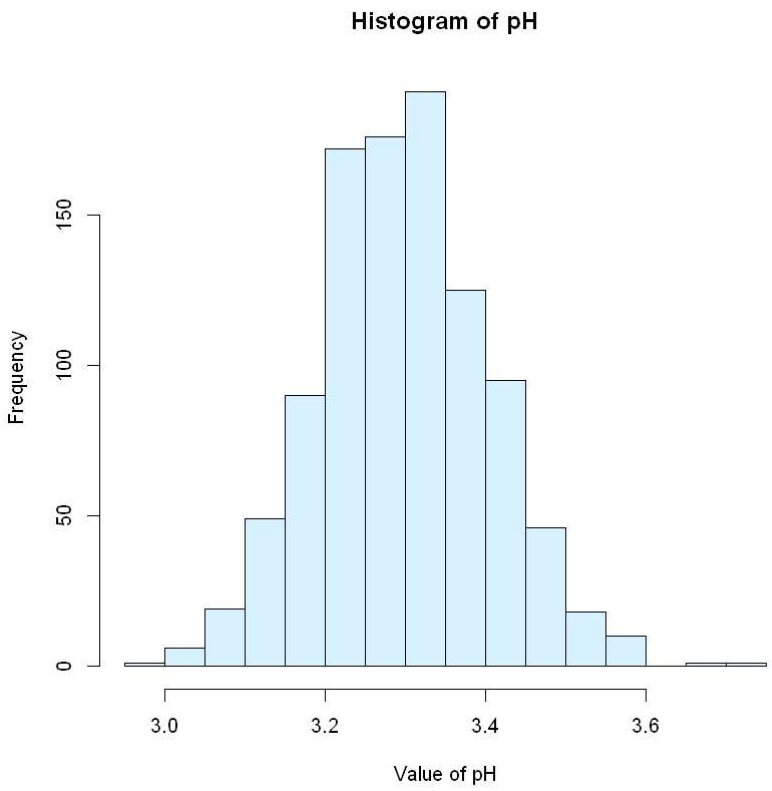


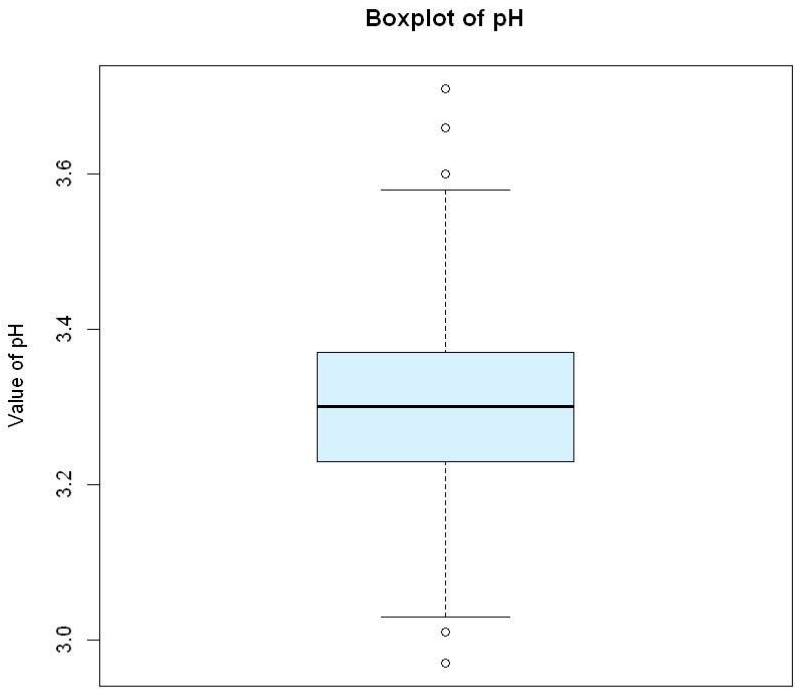
Deskripsi

Kolom Density secara kualitatif hampir terdistribusi normal, terdapat beberapa outlier atas dan bawah yang jumlahnya relatif lebih banyak dibandingkan kolom lain

9. Kolom *pH*

In [11]:
```
pH <- df$pH

getHist(pH,"pH", "#D8F2FF")
getBoxPlot(pH, "pH", "#D8F2FF")
```



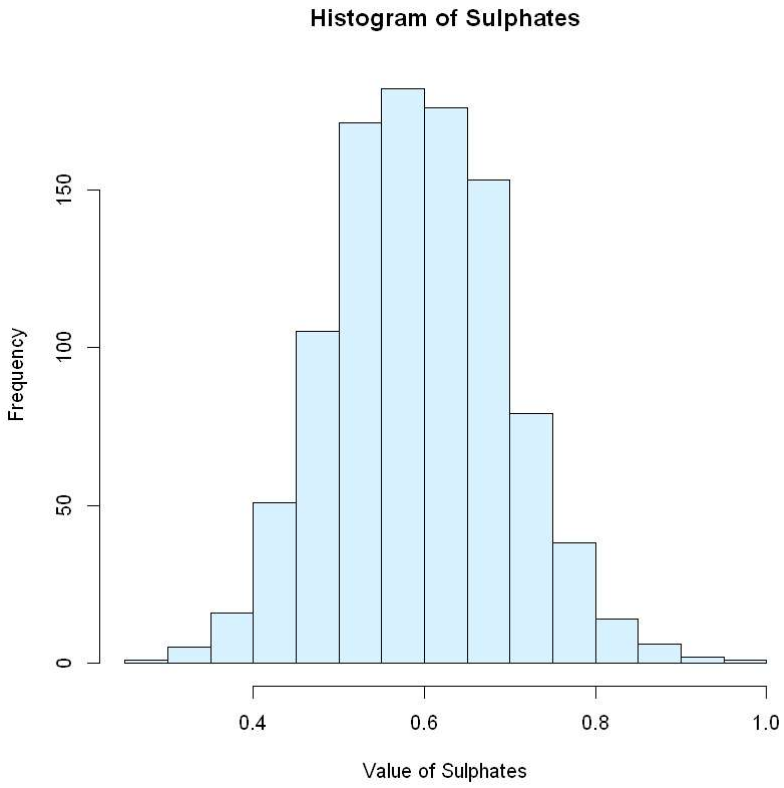


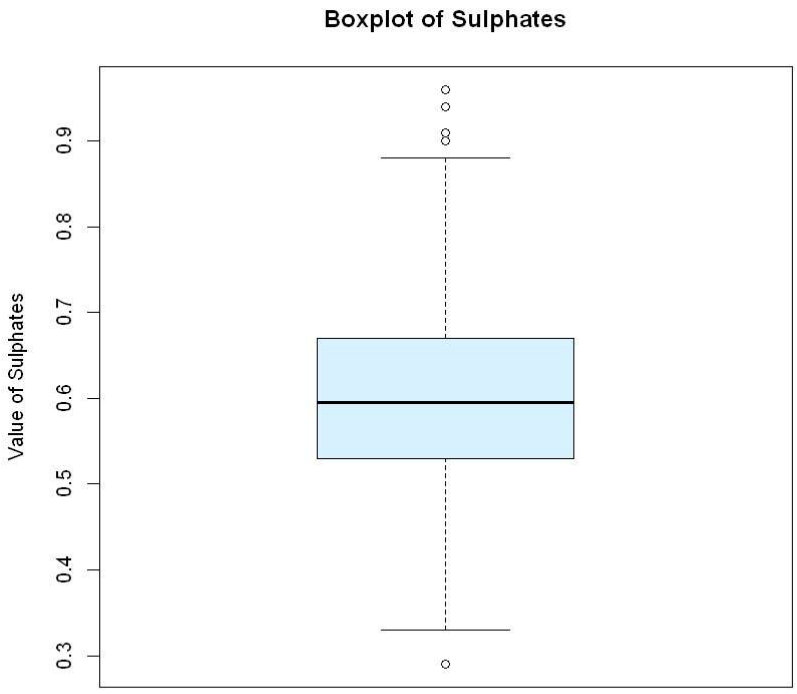
Deskripsi

Kolom Total Sulfur Dioxide secara kualitatif hampir terdistribusi normal, outlier atas dan bawah yang masing masing cukup memiliki jarak yang jauh dari range "kenormalan" data dan satu sama lain

10. Kolom *sulphates*

In [12]:
```
sulphates <- df$sulphates

getHist(sulphates,"Sulphates", "#D8F2FF")
getBoxPlot(sulphates, "Sulphates", "#D8F2FF")
```



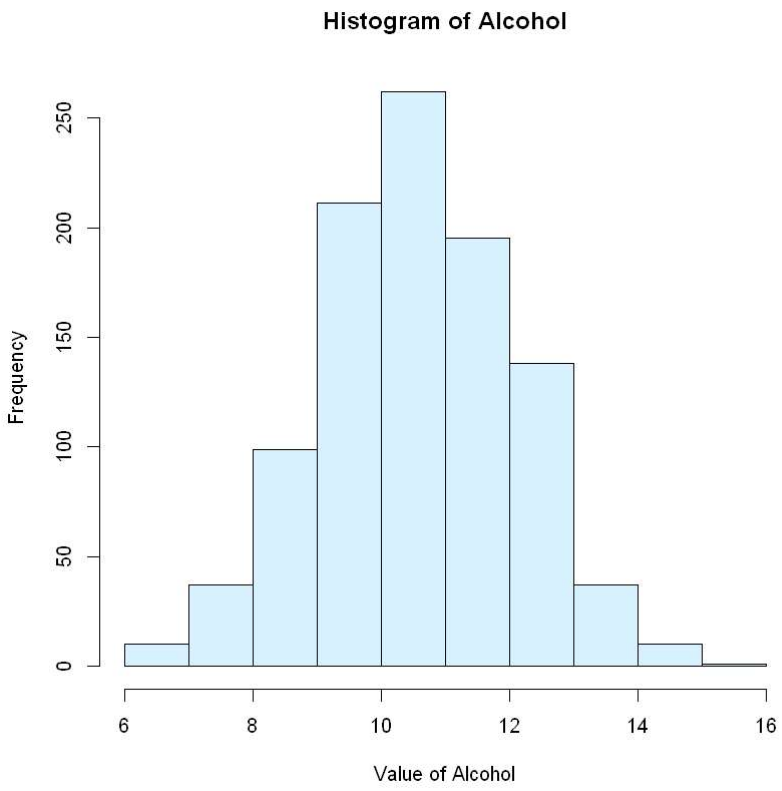


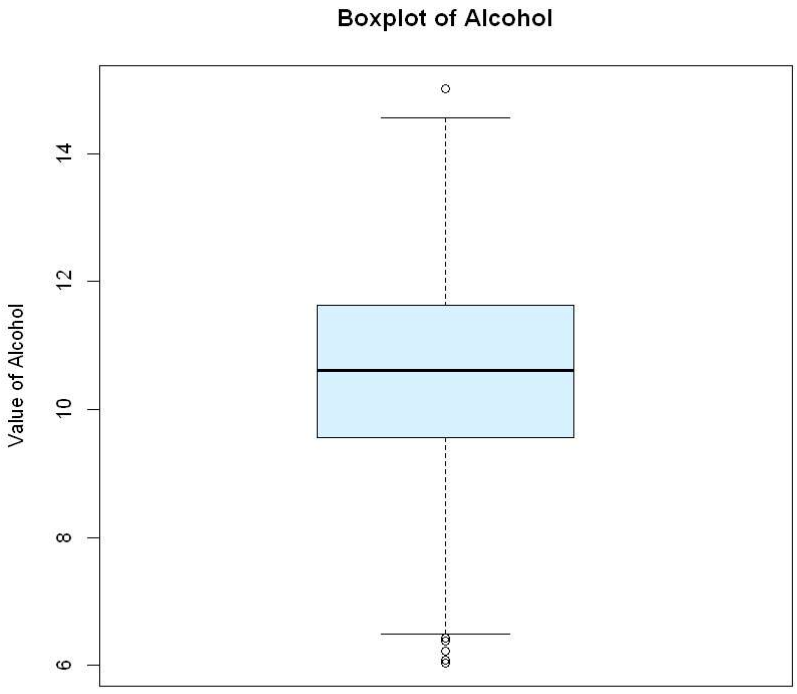
Deskripsi

Kolom Total Sulphates secara kualitatif hampir terdistribusi normal, terdapat beberapa outlier atas dan 1 outlier bawah

11. Kolom *alcohol*

In [13]:
```
alcohol <- df$alcohol

getHist(alcohol,"Alcohol", "#D8F2FF")
getBoxPlot(alcohol, "Alcohol", "#D8F2FF")
```



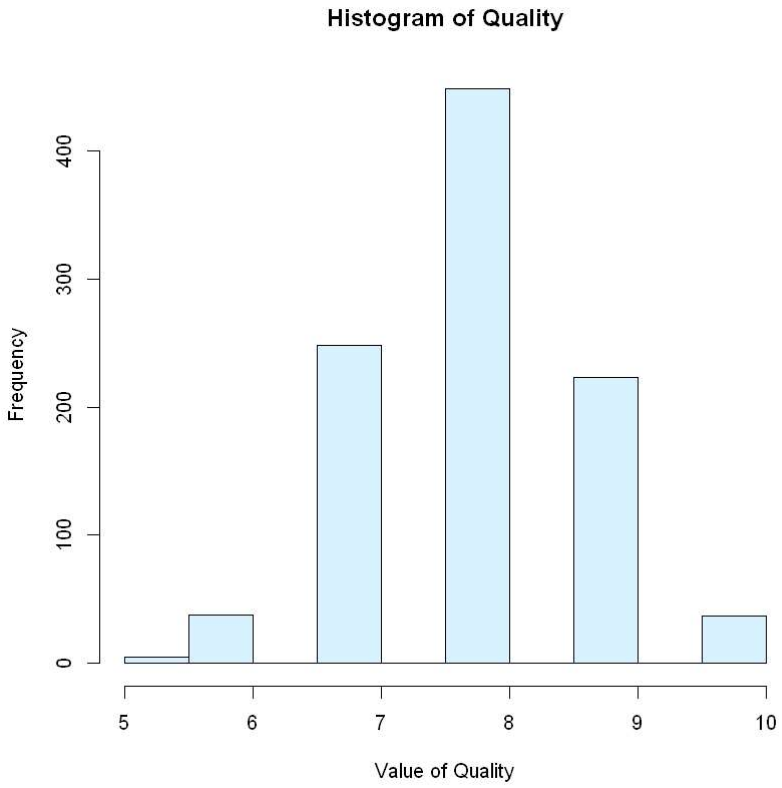


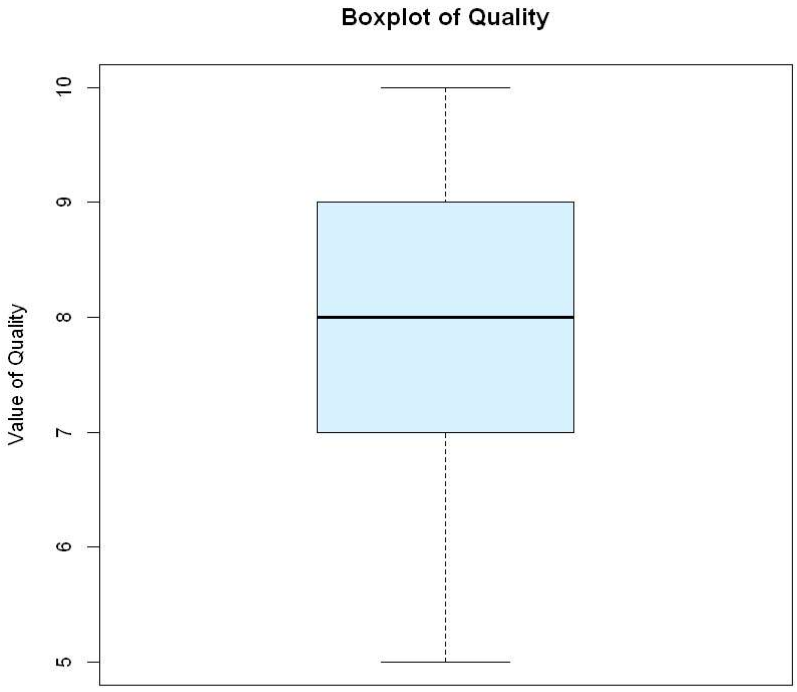
Deskripsi

Kolom Alcohol secara kualitatif hampir terdistribusi normal, terdapat cukup banyak outlier bawah, hanya terdapat 1 buah outlier atas

12. Kolom *quality*

In [14]:
```
quality <- df$quality

getHist(quality,"Quality", "#D8F2FF")
getBoxPlot(quality, "Quality", "#D8F2FF")
```





Deskripsi

Kolom Total Quality secara kualitatif hampir terdistribusi normal, tidak terdapat outlier yang menandakan bahwa data kolom quality merupakan instance yang bagus

Tugas Besar IF2220 - Probabilitas dan Statistika

Part 3: Normality Test

Anggota:
13521116 - Juan Christopher Santoso
13521162 - Antonio Natthan Krishna

Data Preparation and Data Description

In [1]:
```
df <- read.csv("..\\test\\anggur.csv")

# get rows and columns of data
row <- nrow(df)
col <- ncol(df)

# enumeration of columns and rows
enums_col <- c(1:col)
enums_row <- c(1:row)

# Data Statistics
properties <- c("Rows", "Columns")
value <- c(nrow(df), ncol(df))
cbind(properties, value)

# List of Columns
columns_index <- c(1:ncol(df))
columns_name <- colnames(df)

# Display List
cbind(columns_index, columns_name)
```

A matrix: 2 × 2 of type chr

properties	value
Rows	1000
Columns	12

A matrix: 12 × 2 of type chr

columns_index	columns_name
1	fixed.acidity
2	volatile.acidity
3	citric.acid
4	residual.sugar
5	chlorides
6	free.sulfur.dioxide
7	total.sulfur.dioxide
8	density
9	pH
10	sulphates
11	alcohol
12	quality

Global Function used for Data Visualization

In [2]:
```
getHist <- function(v, name, color) {
    hist(v,
    main = paste("Histogram of", name),
    xlab = paste("Value of", name),
    ylab = "Frequency",
    col = color)
}
getQQPlot <- function(v, name, color){
    qqnorm(v,
    main = paste("Histogram of", name)
    )
    qqline(v,
    col = color)
}
```

Global Functions used for Normality Test

In [3]:
```
# loading library package
library("dplyr")

# Mean data
cat("Column's Mean:\n")
columns_mean <- colMeans(df)
cbind(columns_mean)

cat("Column's Median: \n")
```

```
columns_median = c(1:ncol(df))
for (i in columns_index){
    columns_median[i] <- median(df[,columns_name[i]])
}
cbind(columns_name, columns_median)

getmode <- function(v) {
    uniquev <- unique(v)
    maxCount <- 0
    maxElmt <- 0
    for (j in c(1:nrow(df))){
        temp <- v[i]
        count <- 0
        for (k in c(i:nrow(df))){
            if (temp == v[k]){
                count <- count + 1
            }
        }
        if (count > maxCount){
            maxCount <- count
            maxElmt <- temp
        }
    }
    return(maxElmt)
}

cat("Column's Mode: \n")

columns_mode = c(1:ncol(df))
for (i in columns_index){
    columns_mode[i] <- getmode(df[,columns_name[i]])
}

cbind(columns_name, columns_mode)

cat("Column's Skewness: \n")

columns_skewness = c(1:ncol(df))

for (i in columns_index){
    tempMean <- mean(df[,columns_name[i]])
    tempStddev <- sd(df[,columns_name[i]])
    count <- 0
    for (j in c(1:nrow(df))){
        count <- count + (df[j, columns_name[i]] - tempMean) ^3
    }
    columns_skewness[i] <- count / (nrow(df) * (tempStddev ^3))
}

cbind(columns_name, columns_skewness)

cat("Column's Kurtosis: \n")

columns_kurtosis = c(1:ncol(df))
```

```
for (i in columns_index){
    tempMean <- mean(df[,columns_name[i]])
    tempSd <- sd(df[,columns_name[i]])
    numerator <- 0
    for (j in c(1:nrow(df))){
        numerator <- numerator + (df[j, columns_name[i]] - tempMean) ^ 4
    }

    columns_kurtosis[i] <- numerator / ( nrow(df) * tempSd ^ 4)
}

cbind(columns_name, columns_kurtosis)
```

```
Attaching package: 'dplyr'


The following objects are masked from 'package:stats':

    filter, lag


The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

Column's Mean:

A matrix: 12 × 1 of type dbl

	columns_mean
fixed.acidity	7.15253000
volatile.acidity	0.52083850
citric.acid	0.27051700
residual.sugar	2.56710368
chlorides	0.08119515
free.sulfur.dioxide	14.90767925
total.sulfur.dioxide	40.29015000
density	0.99592530
pH	3.30361000
sulphates	0.59839000
alcohol	10.59228000
quality	7.95800000

Column's Median:

A matrix: 12 × 2 of type chr

columns_name	columns_median
fixed.acidity	7.15
volatile.acidity	0.52485
citric.acid	0.2722
residual.sugar	2.51943027286579
chlorides	0.0821669021645236
free.sulfur.dioxide	14.8603462365689
total.sulfur.dioxide	40.19
density	0.996
pH	3.3
sulphates	0.595
alcohol	10.61
quality	8

```
Column's Mode:
```

A matrix: 12 × 2 of type chr

columns_name	columns_mode
fixed.acidity	5.9
volatile.acidity	0.5768
citric.acid	0.3248
residual.sugar	3.37181458927355
chlorides	0.0663785866479429
free.sulfur.dioxide	12.2321700848591
total.sulfur.dioxide	44.26
density	0.9999
pH	3.27
sulphates	0.51
alcohol	10.52
quality	9

```
Column's Skewness:
```

A matrix: 12 × 2 of type chr

columns_name	columns_skewness
fixed.acidity	-0.0287919975632131
volatile.acidity	-0.197105997910775
citric.acid	-0.045439421661083
residual.sugar	0.132240437207526
chlorides	-0.0511654421670584
free.sulfur.dioxide	0.0071090390040008
total.sulfur.dioxide	-0.0239878948518848
density	-0.0766522945530875
pH	0.147229872668135
sulphates	0.148751602565093
alcohol	-0.0189344680909653
quality	-0.0887871070594255

```
Column's Kurtosis:
```

A matrix: 12 × 2 of type chr

columns_name	columns_kurtosis
fixed.acidity	2.96886355314195
volatile.acidity	3.14874381854168
citric.acid	2.88407259945928
residual.sugar	2.94534102928964
chlorides	2.74323396581525
free.sulfur.dioxide	2.62560551750591
total.sulfur.dioxide	3.05152426622451
density	3.0042723321488
pH	3.0683656154671
sulphates	3.05238768805287
alcohol	2.85720916960675
quality	3.09555588797803

Methods used for Normality Test

Uji kualitatif dilakukan dengan menggunakan 2 grafik, yaitu:

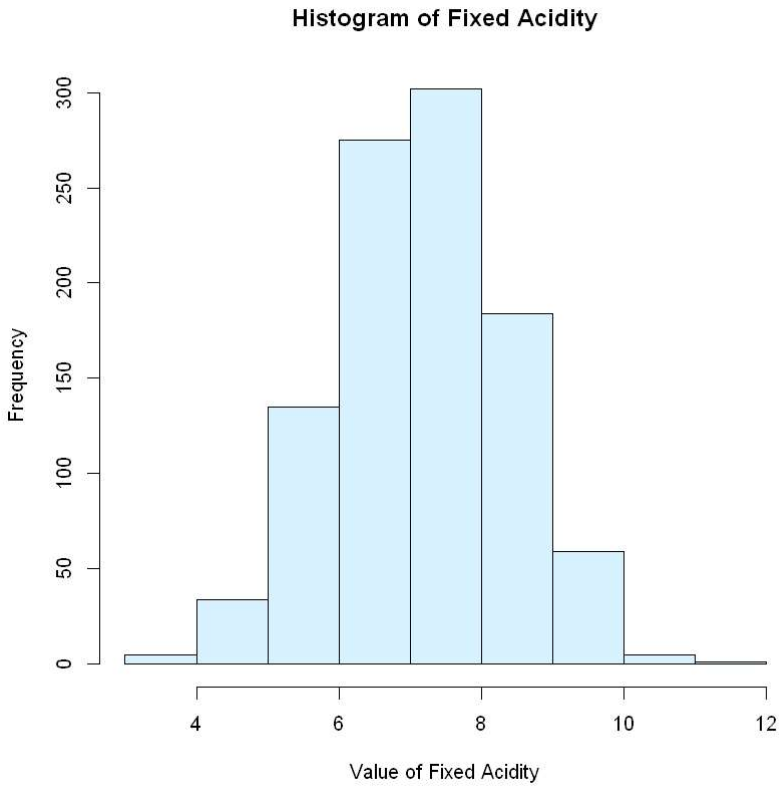
1. Grafik Histogram
2. Grafik QQPlot

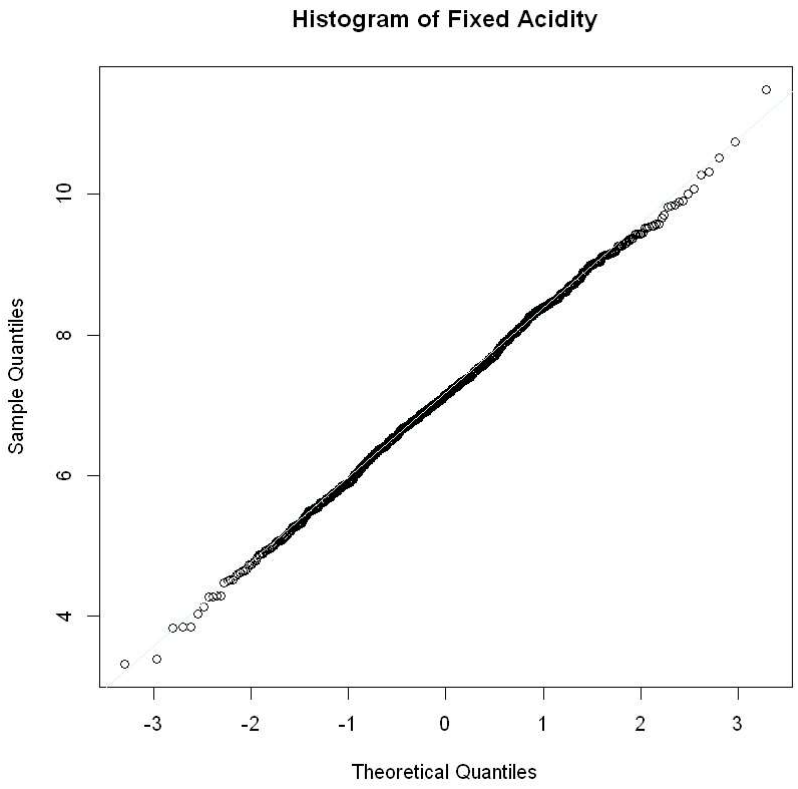
Uji kuantitatif dapat dilakukan dengan berbagai cara, yaitu:

1. Shapiro-Wilk Test
2. Analisis Modus, Median, dan Mean
3. Analisis Skewness dan Kurtosis

1. Kolom *fixed.acidity*

In [4]:
```
# Uji Kualitatif
fixed_acidity <- df$fixed.acidity

getHist(fixed_acidity, "Fixed Acidity", "#D8F2FF")
getQQPlot(fixed_acidity, "Fixed Acidity", "#D8F2FF")
```





Berdasarkan grafik histogram, dapat dilihat bahwa data memiliki bentuk yang terdistribusi normal. Modus dari data berada pada data yang memiliki nilai tengah. Tak hanya itu, didukung dengan grafik QQPlot, dapat dilihat bahwa data masih memenuhi garis lurus yang tertera. Dengan begitu, berdasarkan uji kualitatif data pada kolom `fixed.acidity` terdistribusi normal.

In [5]:
```
# Uji Kuantitatif

shapiro.test(df$fixed.acidity)
```

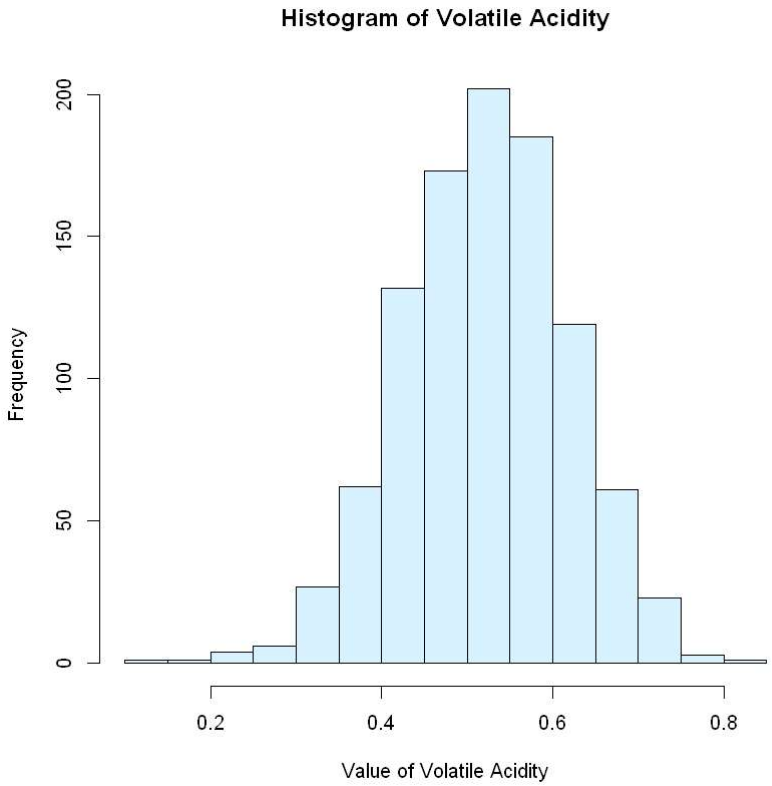
```
	Shapiro-Wilk normality test

data:  df$fixed.acidity
W = 0.99904, p-value = 0.8937
```

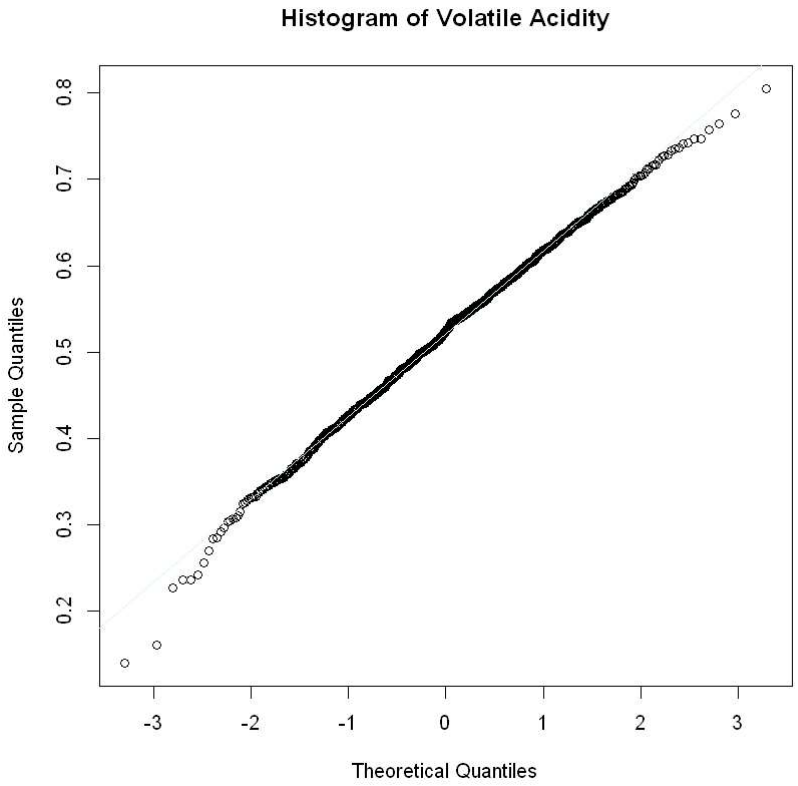
Berdasarkan Saphiro Wilk test, didapatkan nilai p-value yang lebih besar dibandingkan 0.05. Maka dari itu, dapat diasumikan bahwa kolom `fixed.acidity` terdistribusi normal. Hal ini juga didukung dengan nilai mean dan median yang hampir serupa, serta nilai modus yang berada pada suatu rentang yang sama pada histogram. Terakhir, nilai skewness juga mendekati 0 yaitu -0.02879 dan nilai kurtosis yang mendekati 3 yaitu 2.9688.

2. Kolom *volatile.acidity*

In [6]:
```
# Uji Kualitatif
volatile_acidity <- df$volatile.acidity
```

```
getHist(volatile_acidity, "Volatile Acidity", "#D8F2FF")
getQQPlot(volatile_acidity, "Volatile Acidity", "#D8F2FF")
```





Berdasarkan grafik histogram, dapat dilihat bahwa data memiliki bentuk yang terdistribusi normal. Modus pada data berada pada rentang yang masih tergolong berada di tengah-

tengah data. Tak hanya itu, didukung dengan grafik QQPlot, dapat dilihat bahwa data masih memenuhi garis lurus yang tertera. Dengan begitu, berdasarkan uji kualitatif data pada kolom `volatile.acidity` terdistribusi normal.

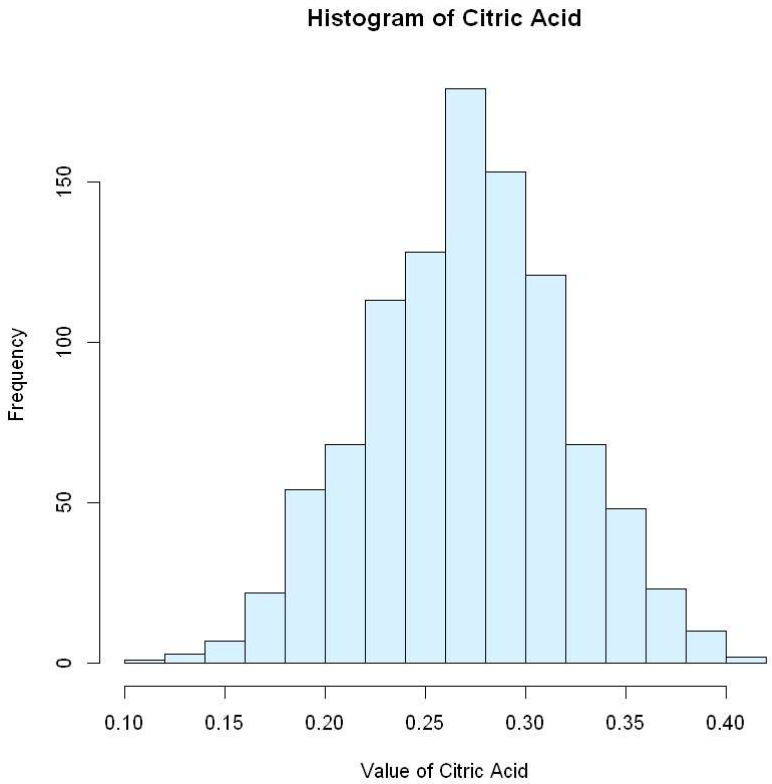
In [7]:
```
# Uji Kuantitatif

shapiro.test(df$volatile.acidity)
```

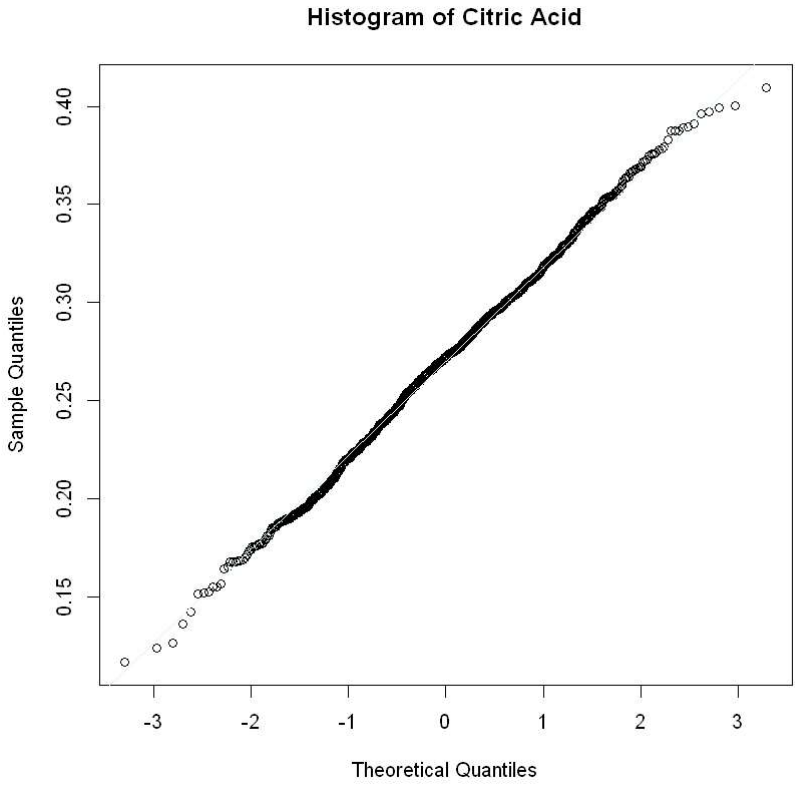
```
	Shapiro-Wilk normality test

data:  df$volatile.acidity
W = 0.99703, p-value = 0.05991
```

Berdasarkan Saphiro Wilk test, didapatkan nilai p-value yang lebih besar dibandingkan 0.05. Maka dari itu, dapat diasumikan bahwa kolom `volatile.acidity` terdistribusi normal. Hal ini juga didukung dengan nilai mean dan median yang hampir serupa, serta nilai modus yang berada pada suatu rentang yang sama pada histogram. Terakhir, nilai skewness juga mendekati 0 yaitu -0.1971 dan nilai kurtosis yang mendekati 3 yaitu 3.1487.

3. Kolom *citric.acid*

In [8]:
```
# Uji Kualitatif
citric_acid <- df$citric.acid

getHist(citric_acid, "Citric Acid", "#D8F2FF")
getQQPlot(citric_acid, "Citric Acid", "#D8F2FF")
```





Berdasarkan grafik histogram, dapat dilihat bahwa data memiliki bentuk yang terdistribusi normal. Modus pada data berada pada rentang yang masih tergolong berada di tengah-tengah data. Tak hanya itu, didukung dengan grafik QQPlot, dapat dilihat bahwa data masih memenuhi garis lurus yang tertera. Dengan begitu, berdasarkan uji kualitatif data pada kolom `citric.acid` terdistribusi normal.

In [9]:
```
# Uji Kuantitatif

shapiro.test(df$citric.acid)
```

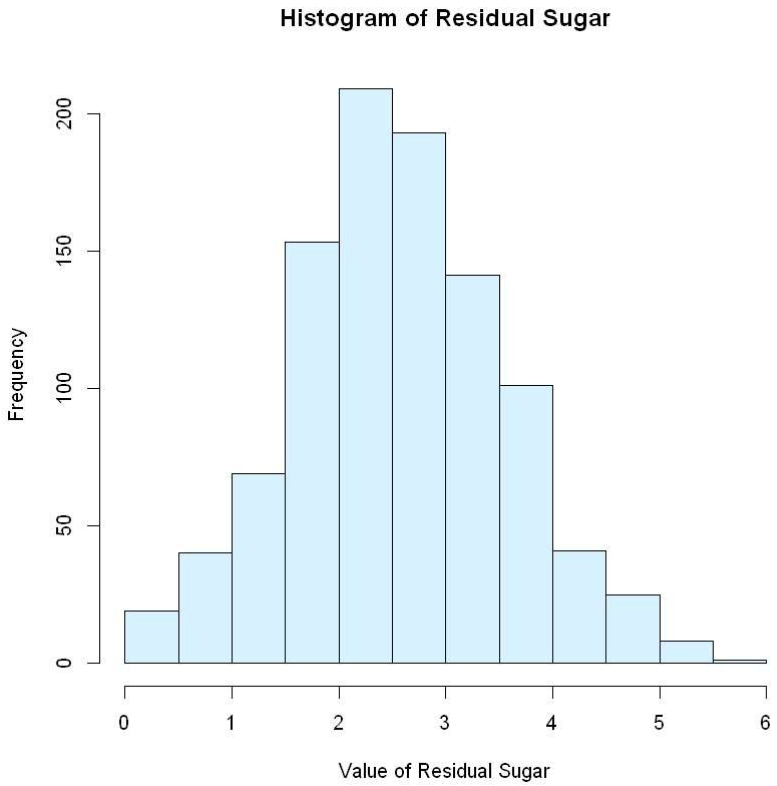
```
	Shapiro-Wilk normality test

data:  df$citric.acid
W = 0.99796, p-value = 0.2649
```

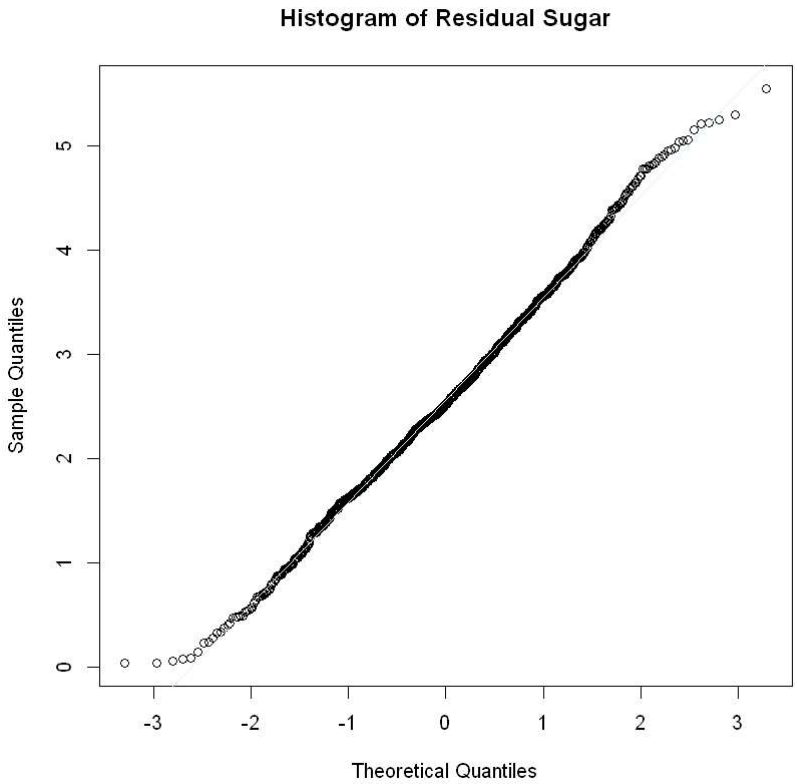
Berdasarkan Saphiro Wilk test, didapatkan nilai p-value yang lebih besar dibandingkan 0.05. Maka dari itu, dapat diasumikan bahwa kolom `citric.acid` terdistribusi normal. Hal ini juga didukung dengan nilai mean dan median yang hampir serupa, serta nilai modus yang berada pada suatu rentang yang sama pada histogram. Terakhir, nilai skewness juga mendekati 0 yaitu -0.0454 dan nilai kurtosis yang mendekati 3 yaitu 2.88407.

4. Kolom *residual.sugar*

In [10]:
```
# Uji Kualitatif
residual_sugar <- df$residual.sugar
```

```
getHist(residual_sugar, "Residual Sugar", "#D8F2FF")
getQQPlot(residual_sugar, "Residual Sugar", "#D8F2FF")
```





Berdasarkan grafik histogram, dapat dilihat bahwa data memiliki bentuk yang terdistribusi normal. Modus pada data berada pada rentang yang masih tergolong berada di tengah-

tengah data. Tak hanya itu, didukung dengan grafik QQPlot, dapat dilihat bahwa data masih memenuhi garis lurus yang tertera. Dengan begitu, berdasarkan uji kualitatif data pada kolom `residual.sugar` terdistribusi normal.

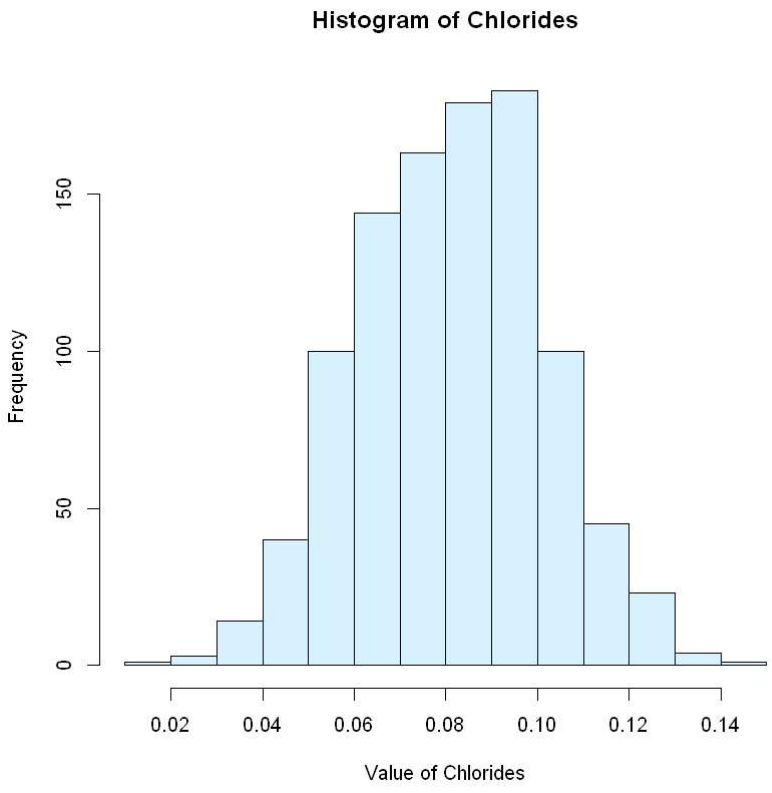
In [11]:
```
# Uji Kuantiatif

shapiro.test(df$residual.sugar)
```

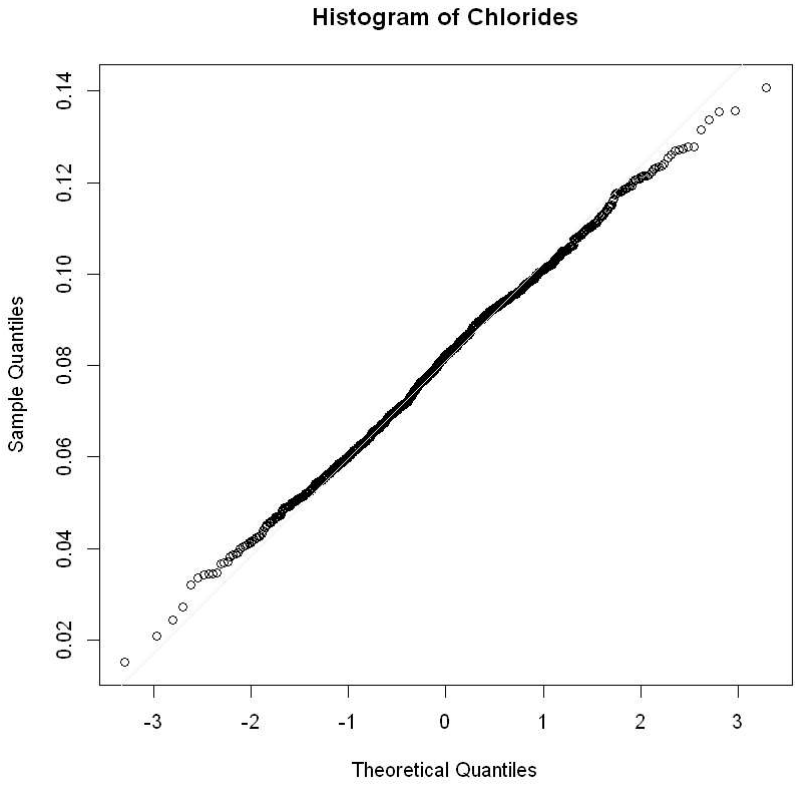
```
	Shapiro-Wilk normality test

data:  df$residual.sugar
W = 0.99686, p-value = 0.045
```

Berdasarkan Saphiro Wilk test, didapatkan nilai p-value yang lebih kecil dibandingkan 0.05. Maka dari itu, dapat diasumikan bahwa berdasarkan tes tersebut, kolom `citric.acid` tidak terdistribusi normal. Namun, hal ini ini bertentangan dengan nilai mean dan median yang hampir serupa, serta nilai modus yang berada pada suatu rentang yang sama pada histogram. Terakhir, nilai skewness juga masih berkisar nilai 0 yaitu 0.1322 dan nilai kurtosis yang berkisar nilai 3 yaitu 2.94534.

5. Kolom *chlorides*

In [12]:
```
# Uji Kualitatif
chlorides <- df$chlorides

getHist(chlorides, "Chlorides", "#D8F2FF")
getQQPlot(chlorides, "Chlorides", "#D8F2FF")
```





Berdasarkan grafik histogram, dapat dilihat bahwa data memiliki bentuk yang terdistribusi normal. Modus pada data berada pada rentang yang masih tergolong berada di tengah-tengah data. Tak hanya itu, didukung dengan grafik QQPlot, dapat dilihat bahwa data masih memenuhi garis lurus yang tertera. Dengan begitu, berdasarkan uji kualitatif data pada kolom `chlorides` terdistribusi normal.

In [13]:
```
# Uji Kuantitatif

shapiro.test(df$chlorides)
```

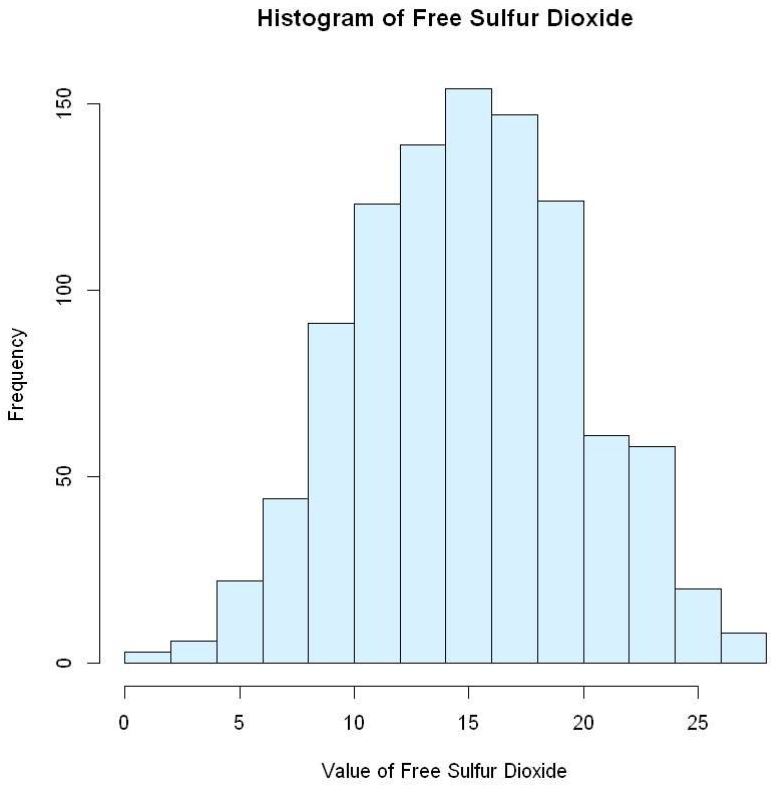
```
	Shapiro-Wilk normality test

data:  df$chlorides
W = 0.99769, p-value = 0.1745
```

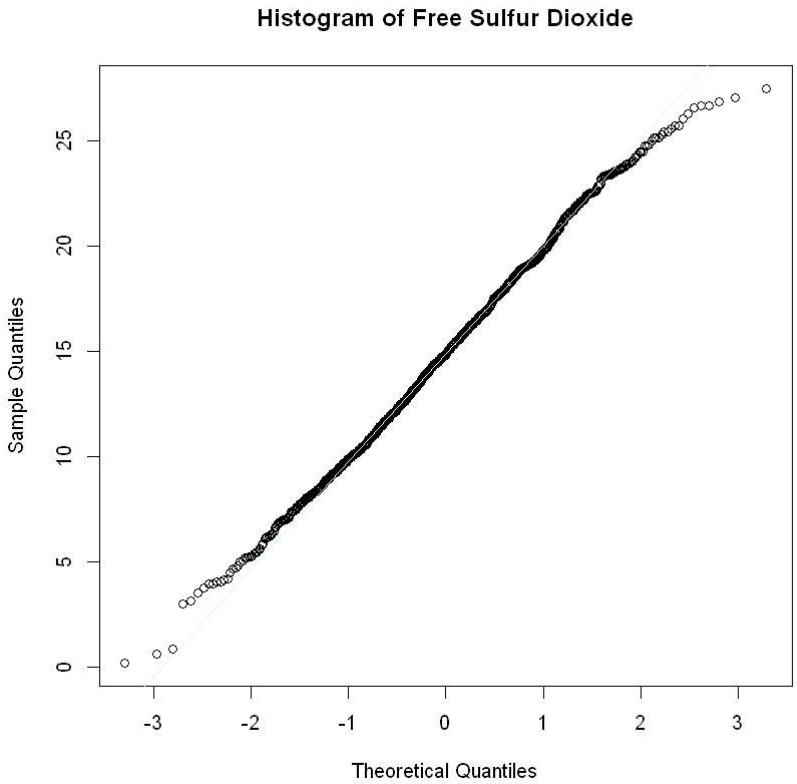
Berdasarkan Saphiro Wilk test, didapatkan nilai p-value yang lebih besar dibandingkan 0.05. Maka dari itu, dapat diasumikan bahwa kolom `chlorides` terdistribusi normal. Hal ini juga didukung dengan nilai mean dan median yang hampir serupa, serta nilai modus yang berada pada suatu rentang yang sama pada histogram. Terakhir, nilai skewness juga mendekati 0 yaitu -0.05116 dan nilai kurtosis yang mendekati 3 yaitu 2.74323.

6. Kolom *free.sulfur.dioxide*

In [14]:
```
# Uji Kualitatif
free_sulfur_dioxide <- df$free.sulfur.dioxide
```

```
getHist(free_sulfur_dioxide, "Free Sulfur Dioxide", "#D8F2FF")
getQQPlot(free_sulfur_dioxide, "Free Sulfur Dioxide", "#D8F2FF")
```





Berdasarkan grafik histogram, dapat dilihat bahwa data memiliki bentuk yang terdistribusi normal. Modus pada data berada pada rentang yang masih tergolong berada di tengah-

tengah data. Tak hanya itu, didukung dengan grafik QQPlot, dapat dilihat bahwa data masih memenuhi garis lurus yang tertera. Dengan begitu, berdasarkan uji kualitatif data pada kolom `free.sulfur.dioxide` terdistribusi normal.

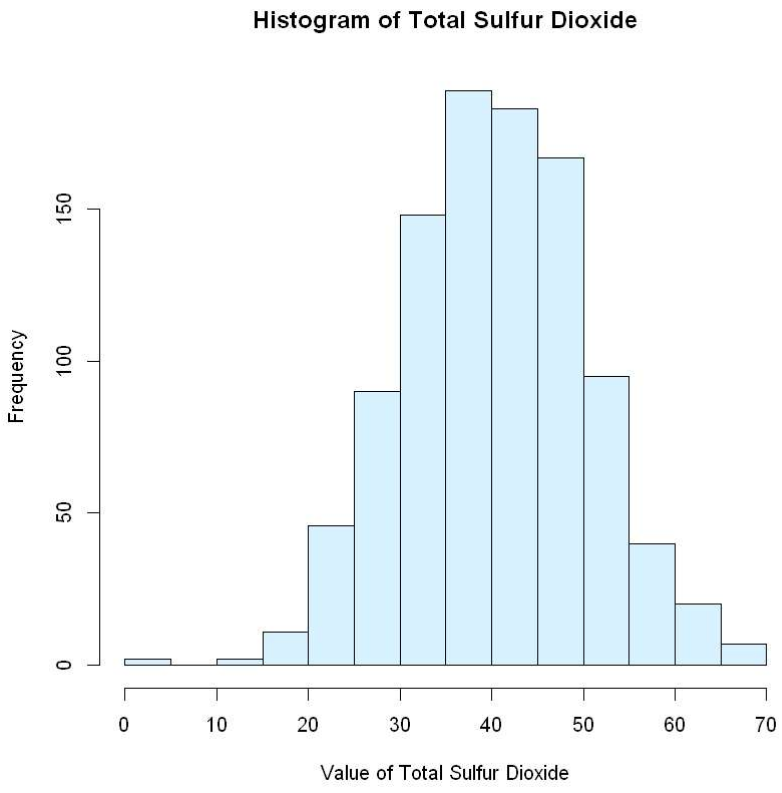
In [15]:
```
# Uji Kuantitatif

shapiro.test(df$free.sulfur.dioxide)
```

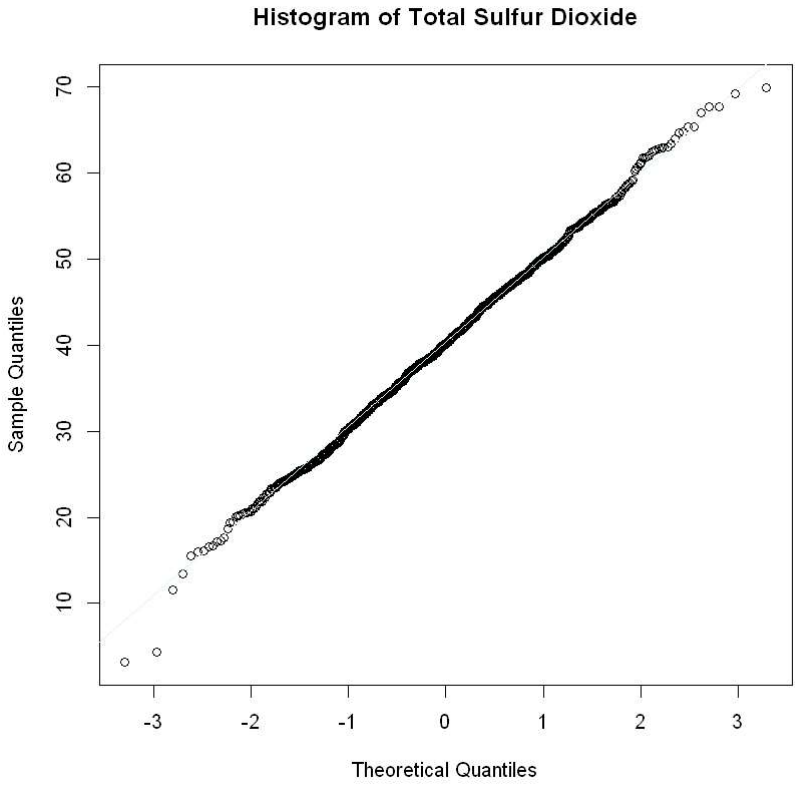
```
	Shapiro-Wilk normality test

data:  df$free.sulfur.dioxide
W = 0.99682, p-value = 0.04247
```

Berdasarkan Saphiro Wilk test, didapatkan nilai p-value yang lebih kecil dibandingkan 0.05. Maka dari itu, dapat diasumikan bahwa berdasarkan tes tersebut, kolom `free.sulfur.dioxide` tidak terdistribusi normal. Namun, hal ini ini bertentangan dengan nilai mean dan median yang hampir serupa, serta nilai modus yang berada pada suatu rentang yang sama pada histogram. Terakhir, nilai skewness juga masih berkisar nilai 0 yaitu 0.00710 dan nilai kurtosis yang berkisar nilai 3 yaitu 2.6256.

7. Kolom *total.sulfur.dioxide*

In [16]:
```
# Uji Kualitatif
total_sulfur_dioxide <- df$total.sulfur.dioxide

getHist(total_sulfur_dioxide, "Total Sulfur Dioxide", "#D8F2FF")
getQQPlot(total_sulfur_dioxide, "Total Sulfur Dioxide", "#D8F2FF")
```





Berdasarkan grafik histogram, dapat dilihat bahwa data memiliki bentuk yang terdistribusi normal. Modus pada data berada pada rentang yang masih tergolong berada di tengah-tengah data. Tak hanya itu, didukung dengan grafik QQPlot, dapat dilihat bahwa data masih memenuhi garis lurus yang tertera. Dengan begitu, berdasarkan uji kualitatif data pada kolom `total.sulfur.dioxide` terdistribusi normal.

In [17]:
```
# Uji Kuantitatif

shapiro.test(df$total.sulfur.dioxide)
```

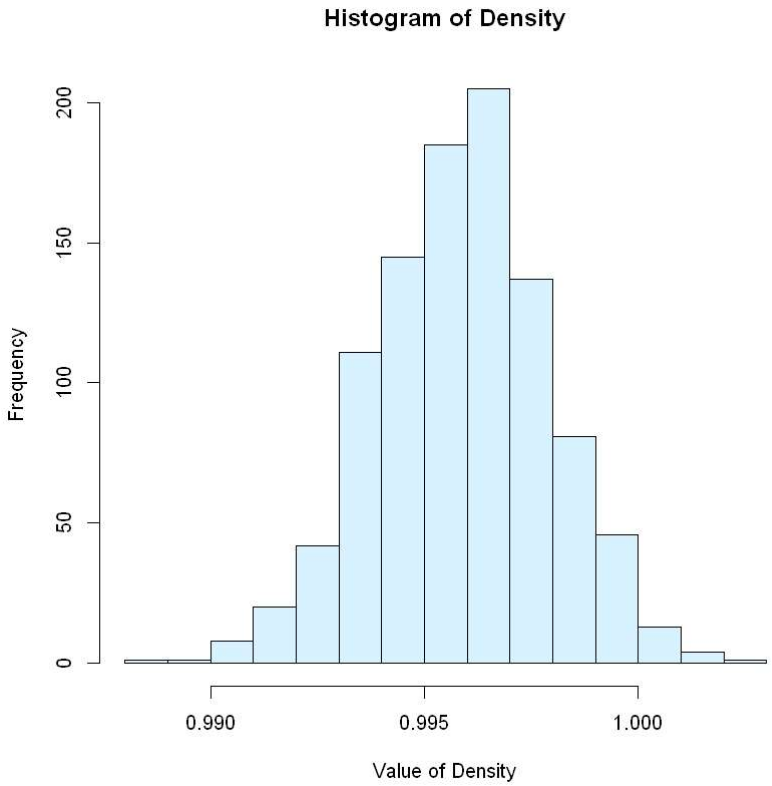
```
	Shapiro-Wilk normality test

data:  df$total.sulfur.dioxide
W = 0.99847, p-value = 0.5367
```

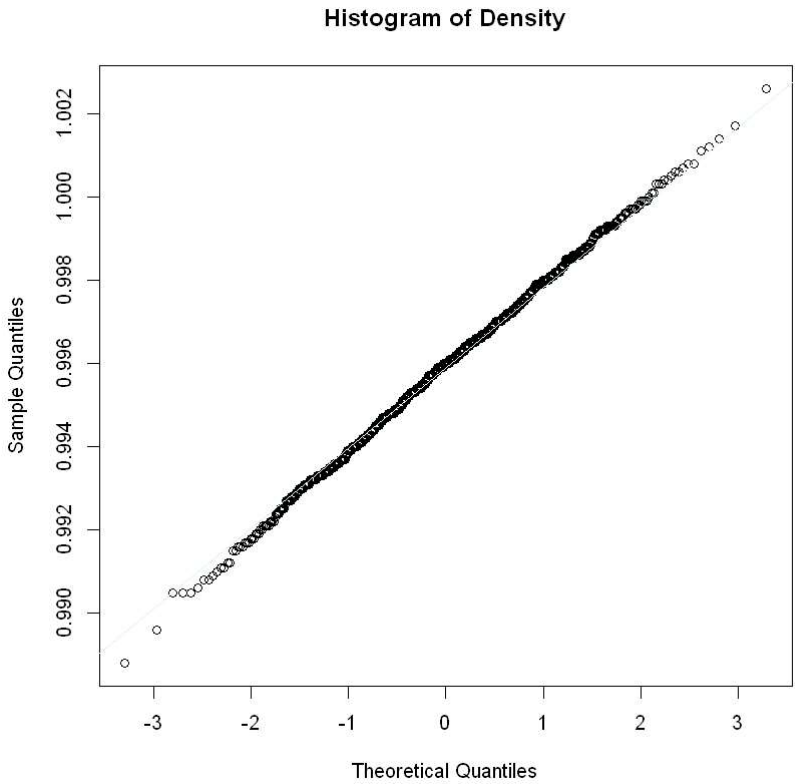
Berdasarkan Saphiro Wilk test, didapatkan nilai p-value yang lebih besar dibandingkan 0.05. Maka dari itu, dapat diasumikan bahwa kolom `total.sulfur.dioxide` terdistribusi normal. Hal ini juga didukung dengan nilai mean dan median yang hampir serupa, serta nilai modus yang berada pada suatu rentang yang sama pada histogram. Terakhir, nilai skewness juga mendekati 0 yaitu -0.02398 dan nilai kurtosis yang mendekati 3 yaitu 3.05152.

8. Kolom *density*

In [18]:
```
# Uji Kualitatif
density <- df$density
```

```
getHist(density, "Density", "#D8F2FF")
getQQPlot(density, "Density", "#D8F2FF")
```





Berdasarkan grafik histogram, dapat dilihat bahwa data memiliki bentuk yang terdistribusi normal. Modus pada data berada pada rentang yang masih tergolong berada di tengah-

tengah data. Tak hanya itu, didukung dengan grafik QQPlot, dapat dilihat bahwa data masih memenuhi garis lurus yang tertera. Dengan begitu, berdasarkan uji kualitatif data pada kolom `density` terdistribusi normal.

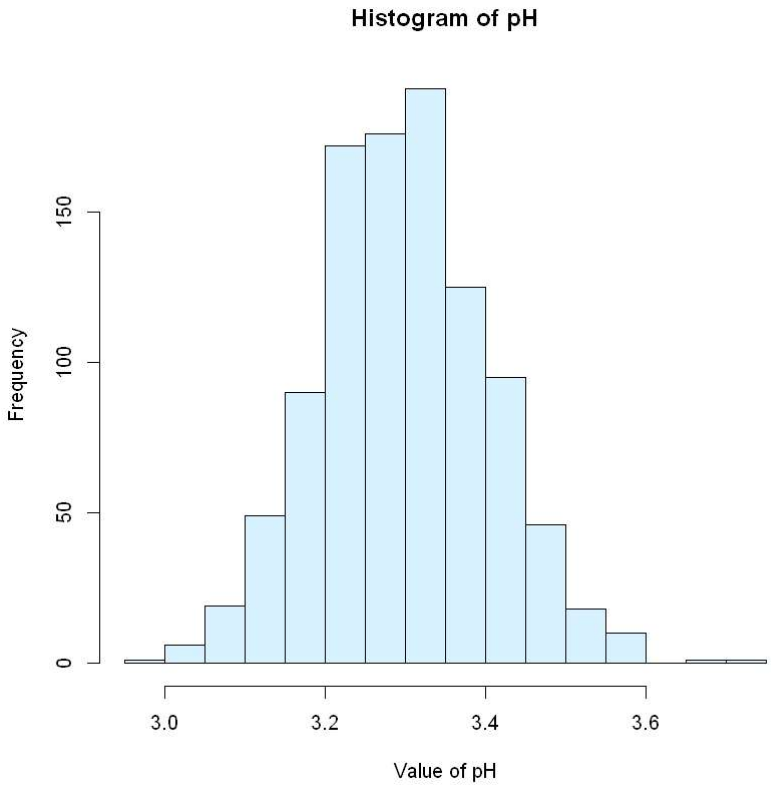
In [19]:
```
# Uji Kuantitatif

shapiro.test(df$density)
```

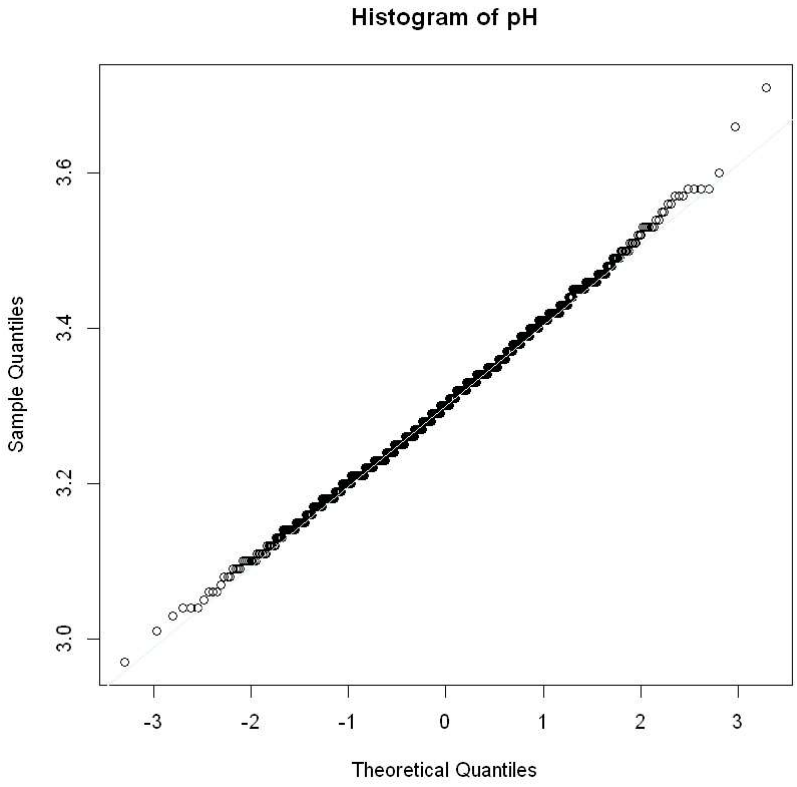
```
	Shapiro-Wilk normality test

data:  df$density
W = 0.99896, p-value = 0.8533
```

Berdasarkan Saphiro Wilk test, didapatkan nilai p-value yang lebih besar dibandingkan 0.05. Maka dari itu, dapat diasumikan bahwa kolom `density` terdistribusi normal. Hal ini juga didukung dengan nilai mean dan median yang hampir serupa, serta nilai modus yang berada pada suatu rentang yang sama pada histogram. Terakhir, nilai skewness juga mendekati 0 yaitu -0.07665 dan nilai kurtosis yang mendekati 3 yaitu 3.00427.

9. Kolom *pH*

In [20]:
```
# Uji Kualitatif
pH <- df$pH

getHist(pH, "pH", "#D8F2FF")
getQQPlot(pH, "pH", "#D8F2FF")
```





Berdasarkan grafik histogram, dapat dilihat bahwa data memiliki bentuk yang terdistribusi normal. Modus pada data berada pada rentang yang masih tergolong berada di tengah-tengah data. Tak hanya itu, didukung dengan grafik QQPlot, dapat dilihat bahwa data masih memenuhi garis lurus yang tertera. Dengan begitu, berdasarkan uji kualitatif data pada kolom `pH` terdistribusi normal.

In [21]:
```
# Uji Kuantitatif

shapiro.test(df$pH)
```

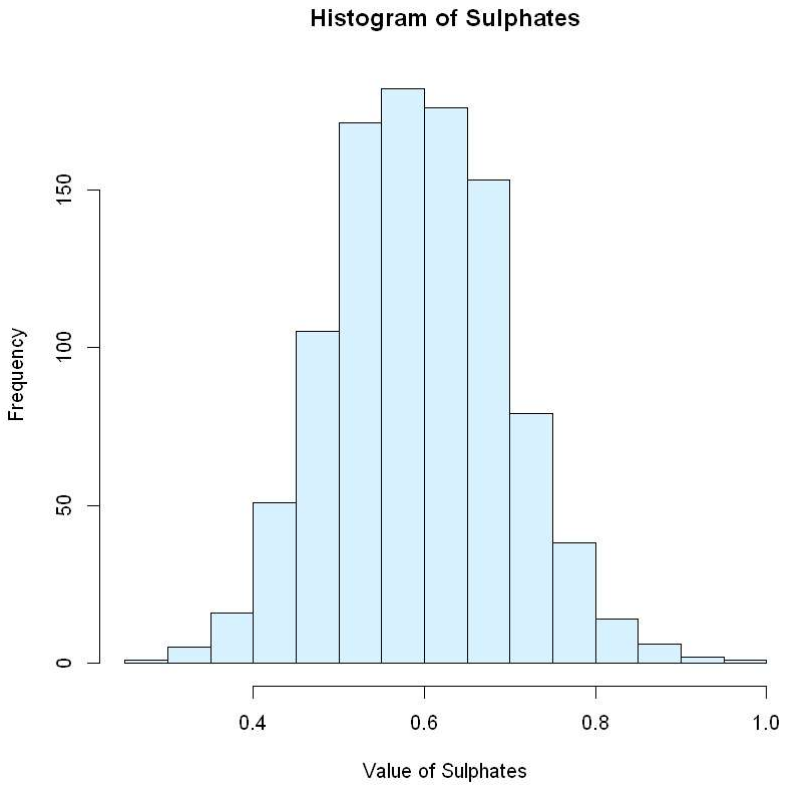
```
	Shapiro-Wilk normality test

data:  df$pH
W = 0.99754, p-value = 0.1373
```

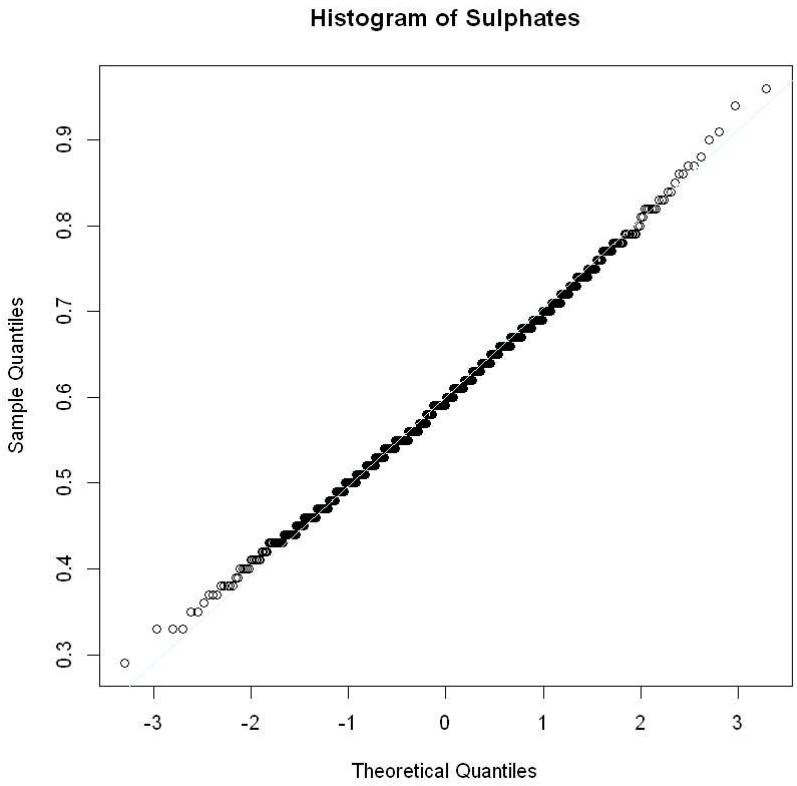
Berdasarkan Saphiro Wilk test, didapatkan nilai p-value yang lebih besar dibandingkan 0.05. Maka dari itu, dapat diasumikan bahwa kolom `pH` terdistribusi normal. Hal ini juga didukung dengan nilai mean dan median yang hampir serupa, serta nilai modus yang berada pada suatu rentang yang sama pada histogram. Terakhir, nilai skewness juga mendekati 0 yaitu 0.14722 dan nilai kurtosis yang mendekati 3 yaitu 3.0683.

10. Kolom *sulphates*

In [22]:
```
# Uji Kualitatif
sulphates <- df$sulphates
```

```
getHist(sulphates, "Sulphates", "#D8F2FF")
getQQPlot(sulphates, "Sulphates", "#D8F2FF")
```





Berdasarkan grafik histogram, dapat dilihat bahwa data memiliki bentuk yang terdistribusi normal. Modus pada data berada pada rentang yang masih tergolong berada di tengah-

tengah data. Tak hanya itu, didukung dengan grafik QQPlot, dapat dilihat bahwa data masih memenuhi garis lurus yang tertera. Dengan begitu, berdasarkan uji kualitatif data pada kolom `sulphates` terdistribusi normal.

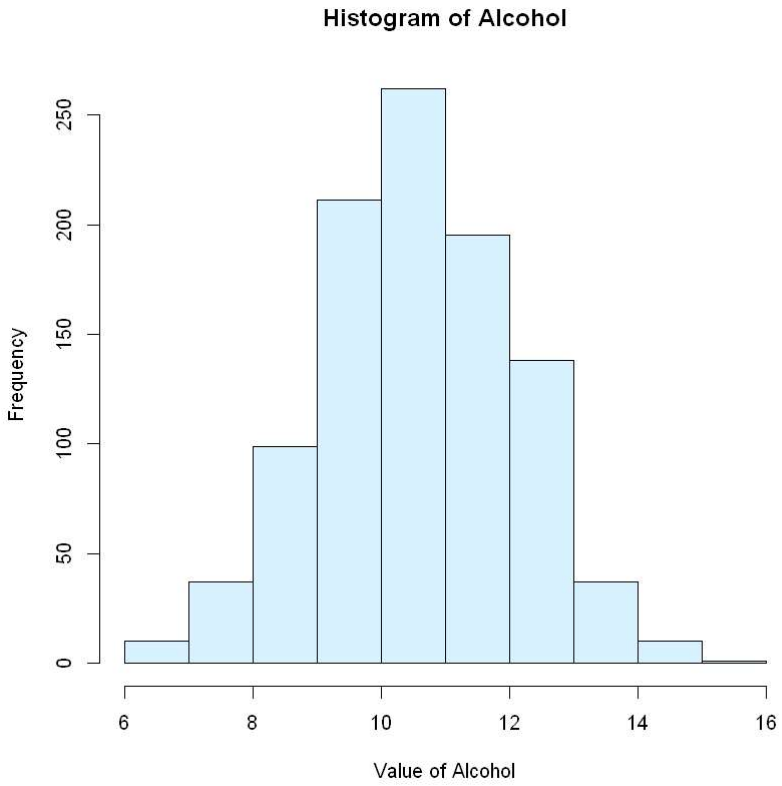
In [23]:
```
# Uji Kuantitatif
shapiro.test(df$sulphates)
```

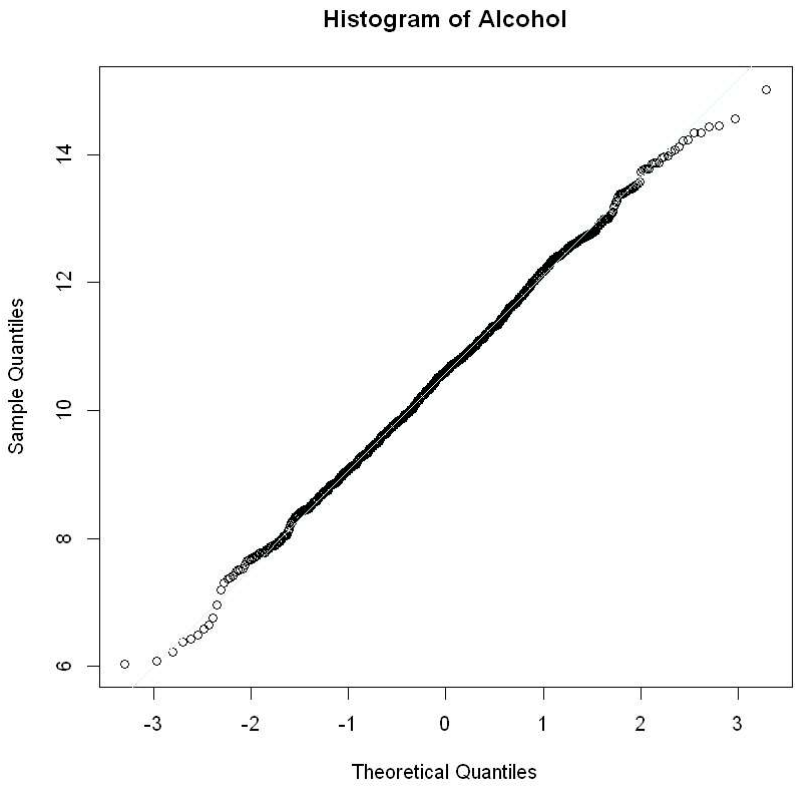
```
	Shapiro-Wilk normality test

data:  df$sulphates
W = 0.99741, p-value = 0.1123
```

Berdasarkan Saphiro Wilk test, didapatkan nilai p-value yang lebih besar dibandingkan 0.05. Maka dari itu, dapat diasumikan bahwa kolom `sulphates` terdistribusi normal. Hal ini juga didukung dengan nilai mean dan median yang hampir serupa, serta nilai modus yang berada pada suatu rentang yang sama pada histogram. Terakhir, nilai skewness juga mendekati 0 yaitu 0.14875 dan nilai kurtosis yang mendekati 3 yaitu 3.052387.

11. Kolom *alcohol*

In [24]:
```
# Uji Kualitatif
alcohol <- df$alcohol

getHist(alcohol, "Alcohol", "#D8F2FF")
getQQPlot(alcohol, "Alcohol", "#D8F2FF")
```





Berdasarkan grafik histogram, dapat dilihat bahwa data memiliki bentuk yang terdistribusi normal. Modus pada data berada pada rentang yang masih tergolong berada di tengah-tengah data. Tak hanya itu, didukung dengan grafik QQPlot, dapat dilihat bahwa data masih memenuhi garis lurus yang tertera. Dengan begitu, berdasarkan uji kualitatif data pada kolom `alcohol` terdistribusi normal.

In [25]:
```
# Uji Kuantitatif
shapiro.test(df$alcohol)
```

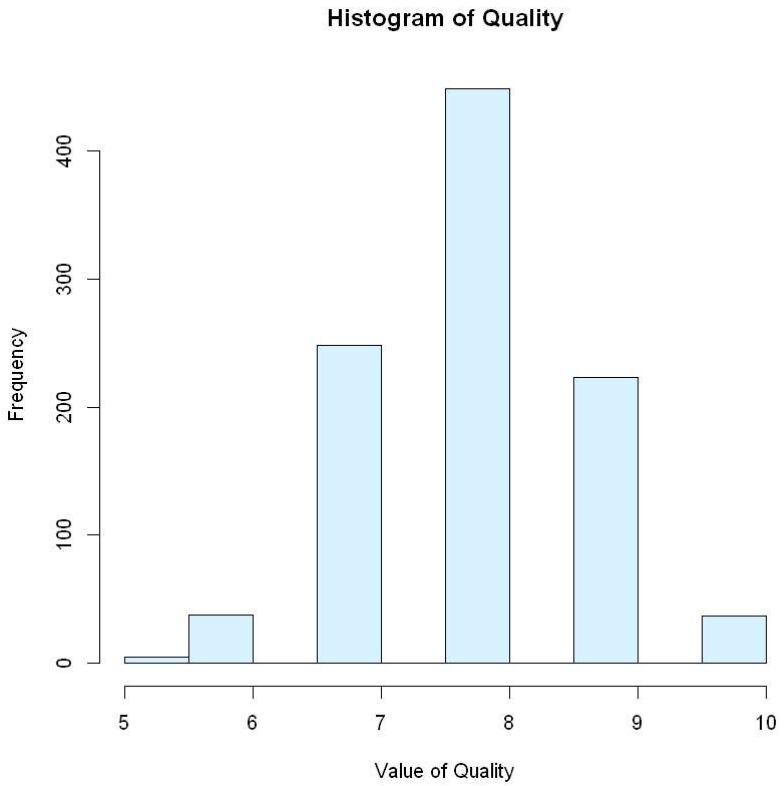
```
	Shapiro-Wilk normality test

data:  df$alcohol
W = 0.99844, p-value = 0.5191
```

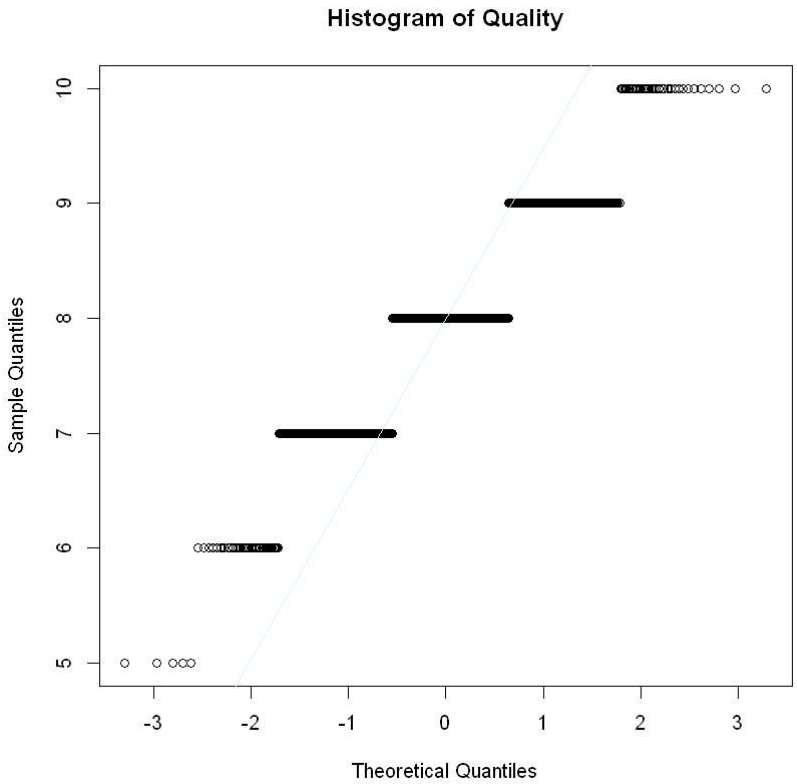
Berdasarkan Saphiro Wilk test, didapatkan nilai p-value yang lebih besar dibandingkan 0.05. Maka dari itu, dapat diasumikan bahwa kolom `sulphates` terdistribusi normal. Hal ini juga didukung dengan nilai mean dan median yang hampir serupa, serta nilai modus yang berada pada suatu rentang yang sama pada histogram. Terakhir, nilai skewness juga mendekati 0 yaitu -0.01893 dan nilai kurtosis yang mendekati 3 yaitu 2.8572.

12. Kolom *quality*

In [26]:
```
# Uji Kualitatif
quality <- df$quality
```

```
getHist(quality, "Quality", "#D8F2FF")
getQQPlot(quality, "Quality", "#D8F2FF")
```





Berdasarkan grafik histogram, dapat dilihat bahwa data memiliki bentuk yang terdistribusi normal. Modus pada data berada pada rentang yang masih tergolong berada di tengah-

tengah data. Walau terlihat data yang tidak lengkap, mengingat data yang terdapat pada kolom ini bernilai bilangan bulat, histogram yang terbentuk masih menyerupai bentuk distribusi normal. Berbeda halnya dengan grafik QQPlot. Pada grafik ini, tidak begitu terlihat bahwa garis tetap mengikuti garis lurus pada QQPlot. Akan tetapi, secara garis besar data tetap memenuhi lintasan miring sesuai garis yang tertera. Dengan begitu, berdasarkan uji kualitatif data pada kolom `quality` terdistribusi normal.

In [27]:
```
# Uji Kuantitatif
shapiro.test(df$quality)
```

```
	Shapiro-Wilk normality test

data:  df$quality
W = 0.8955, p-value < 2.2e-16
```

Berdasarkan Saphiro Wilk test, didapatkan nilai p-value yang lebih besar dibandingkan 0.05. Maka dari itu, dapat diasumikan bahwa kolom `quality` terdistribusi normal. Hal ini juga didukung dengan nilai mean dan median yang hampir serupa, serta nilai modus yang berada pada suatu rentang yang sama pada histogram. Terakhir, nilai skewness juga mendekati 0 yaitu -0.08878 dan nilai kurtosis yang mendekati 3 yaitu 3.09555.

Tugas Besar IF2220 - Probabilitas dan Statistika

Part 4: One Sample Hypothesis

Anggota:
13521116 - Juan Christopher Santoso
13521162 - Antonio Natthan Krishna

All testing use significant of 5%

In [1]:
```
# Import Dataset
df <- read.csv("..\\test\\anggur.csv")

# Data Statistics
properties <- c("Rows", "Columns")
value <- c(nrow(df), ncol(df))
cbind(properties, value)

# List of Columns
columns_index <- c(1:ncol(df))
columns_name <- colnames(df)

# Display List
cbind(columns_index, columns_name)

# Significance
Significance <- 0.05
```

A matrix: 2 × 2 of type chr

properties	value
Rows	1000
Columns	12

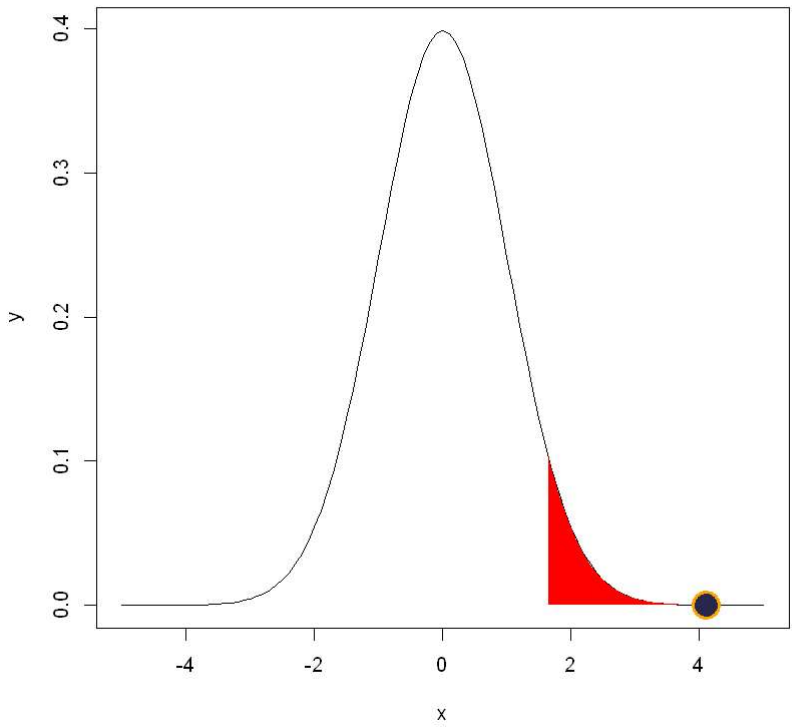
A matrix: 12 × 2 of type chr

columns_index	columns_name
1	fixed.acidity
2	volatile.acidity
3	citric.acid
4	residual.sugar
5	chlorides
6	free.sulfur.dioxide
7	total.sulfur.dioxide
8	density
9	pH
10	sulphates
11	alcohol
12	quality

1. Is the mean of `pH` greater than 3.29?

In [2]:
```
t <- (mean(df[,"pH"]) - 3.29) / (sd(df[,"pH"]) / sqrt(nrow(df)))
t0 <- qt(0.05, nrow(df)-1, lower.tail = FALSE)

cat("t  :", t, "\n")
cat("t0 :", t0, "\n")
cat("P-value:", 1- pt(t, nrow(df)-1))

# Plotting Critical Area
x <- seq(-5, 5, 0.1)
y <- dt(x, nrow(df)-1)
plot(x, y, type = "l")

x2 <- seq(t0,5,0.01)
y2 <- dt(x2, nrow(df)-1)
x2 = c(t0,x2,5)
y2 = c(0,y2,0)
polygon(x2,y2, col="red", border=NA)

lines(t, 0, type = "o", pch=21, bg="#28284d", cex=3, lwd=3, col="orange")
```

```
t  : 4.103781
t0 : 1.64638
P-value: 2.197958e-05
```



H0 = (mean pH == 3.29)
H1 = (mean pH > 3.29)

Use the significance 0.05
Using t distribution with degree 999 (1000-1),
Critical area : t > t(0.05),
Since t > t(0.05) (and p-Value < significance) which means t is located in critical area. Hence, we reject H0.

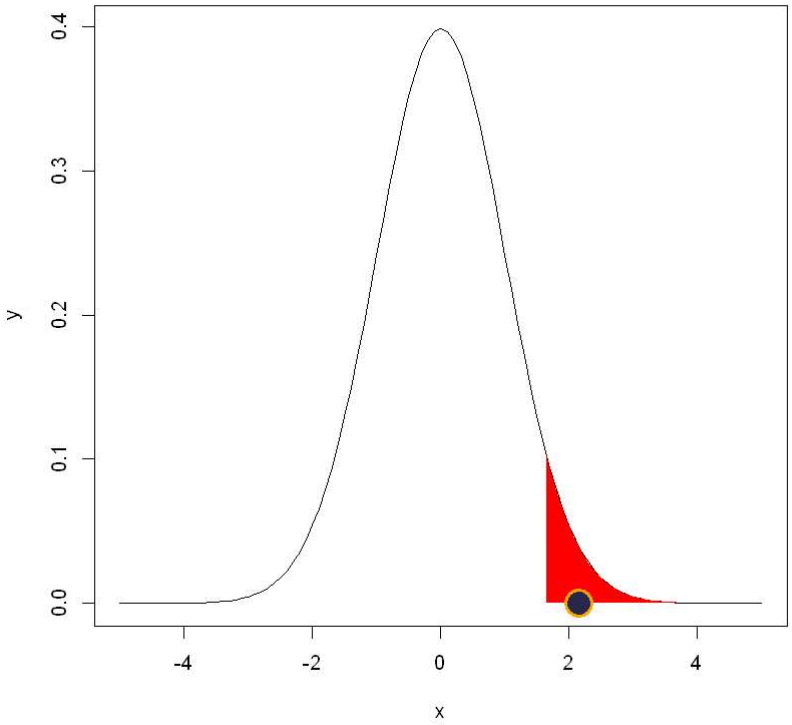
Conclusion: mean of population's pH greater than 3.29

2. Is the mean of `residual sugar` greater than 2.50?

In [3]:
```
t <- (mean(df[,"residual.sugar"])-2.5) / (sd(df[,"residual.sugar"]) / sqrt(nrow(df)
t0 <- qt(0.05, nrow(df)-1, lower.tail = FALSE)

cat("t  :", t, "\n")
cat("t0 :", t0, "\n")
cat("P-value:", 1 - pt(t, nrow(df)-1))

# Plotting Critical Area
x <- seq(-5, 5, 0.1)
y <- dt(x, nrow(df))
plot(x, y, type = "l")

x2 <- seq(t0,5,0.01)
y2 <- dt(x2, nrow(df)-1)
```

```
x2 = c(t0,x2,5)
y2 = c(0,y2,0)
polygon(x2,y2, col="red", border=NA)

lines(t, 0, type = "o", pch=21, bg="#28284d", cex=3, lwd=3, col="orange")
```

```
t  : 2.147962
t0 : 1.64638
P-value: 0.01597836
```



H0 = (mean residual sugar == 2.50)
H1 = (mean residual sugar > 2.50)

Use the significance 0.05
Using t distribution with degree 999 (1000-1),
Critical area : t > t(0.05),
Since t > t(0.05) (and p value < significance) which means t is located in critical area. Hence, we reject H0.

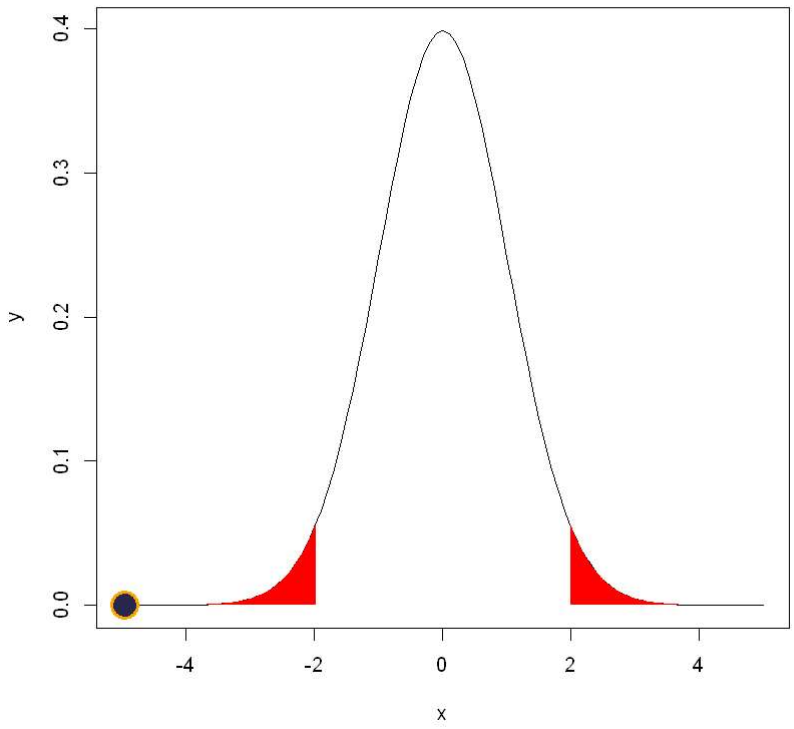
Conclusion: mean of population's residual sugar greater than 2.50

3. Is the mean of the first 150 row in column `sulphates` not 0.65?

In [4]:
```
t <- (mean(df[1:150,"sulphates"]) - 0.65) / (sd(df[1:150,"sulphates"]) / sqrt(150))
t0low <- qt(0.025, 150-1)
t0high <- qt(0.025, 150-1, lower.tail = FALSE)

cat("t  :", t, "\n")
```

```
cat("t0 low :", t0low, "\n")
cat("t0 high :", t0high, "\n")
cat("P-value:", pt(t, 150-1))

# Plotting Critical Area
x <- seq(-5, 5, 0.1)
y <- dt(x, nrow(df))
plot(x, y, type = "l")

x2 <- seq(-5,t0low,0.01)
y2 <- dt(x2, nrow(df)-1)
x2 = c(-5,x2,t0low)
y2 = c(0,y2,0)
polygon(x2,y2, col="red", border=NA)

x3 <- seq(t0high,5,0.01)
y3 <- dt(x3, nrow(df)-1)
x3 = c(x3,5,t0high)
y3 = c(0,y3,0)
polygon(x3,y3, col="red", border=NA)

lines(t, 0, type = "o", pch=21, bg="#28284d", cex=3, lwd=3, col="orange")
```

```
t  : -4.964843
t0 low : -1.976013
t0 high : 1.976013
P-value: 9.295076e-07
```



H0 = (mean first 150 sulphates == 2.50)

H1 = (mean first 150 sulphates != 2.50)

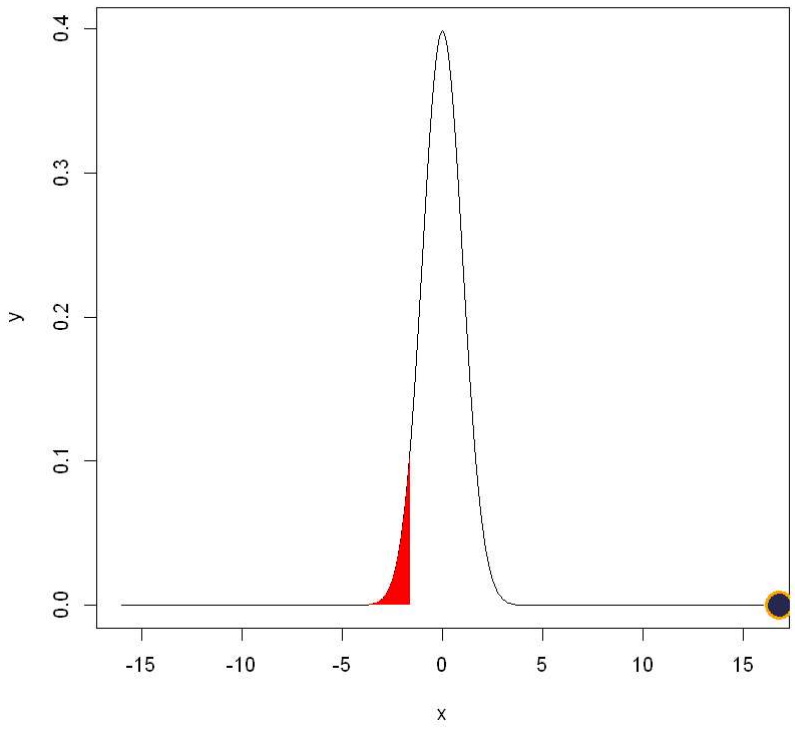
Use the significance 0.05
Using t distribution with degree 999 (1000-1),
Critical area : t < t(-0.025), t > t(0.025)
Since t < t0 (and p value < significance) which means t is located in critical area. Hence, we reject H0.

Conclusion: mean of population's sulphates is NOT 0.65

4. Is the mean of `total sulfur dioxide` lower than 35?

In [5]:
```
t <- (mean(df[,"total.sulfur.dioxide"]) - 35) / (sd(df[,"total.sulfur.dioxide"]) /
t0 <- qt(0.05, nrow(df)-1)

cat("t  :", t, "\n")
cat("t0 :", t0, "\n")
cat("P-value:", pt(t, nrow(df)-1))

# Plotting Critical Area
x <- seq(-16, 16, 0.1)
y <- dt(x, nrow(df))
plot(x, y, type = "l")

x2 <- seq(-16,t0,0.01)
y2 <- dt(x2, nrow(df)-1)
x2 = c(-16,x2,t0)
y2 = c(0,y2,0)
polygon(x2,y2, col="red", border=NA)

lines(t, 0, type = "o", pch=21, bg="#28284d", cex=3, lwd=3, col="orange")
```

```
t  : 16.78639
t0 : -1.64638
P-value: 1
```



H0 = (mean total sulfur dioxide == 35)
H1 = (mean total sulfur dioxide < 35)

Use the significance 0.05
Using t distribution with degree 999 (1000-1),
Critical area : t < t(0.05),
Since t > t(0.05) (and p value > significance) which means t is NOT located in critical area.
Hence, we accept H0.

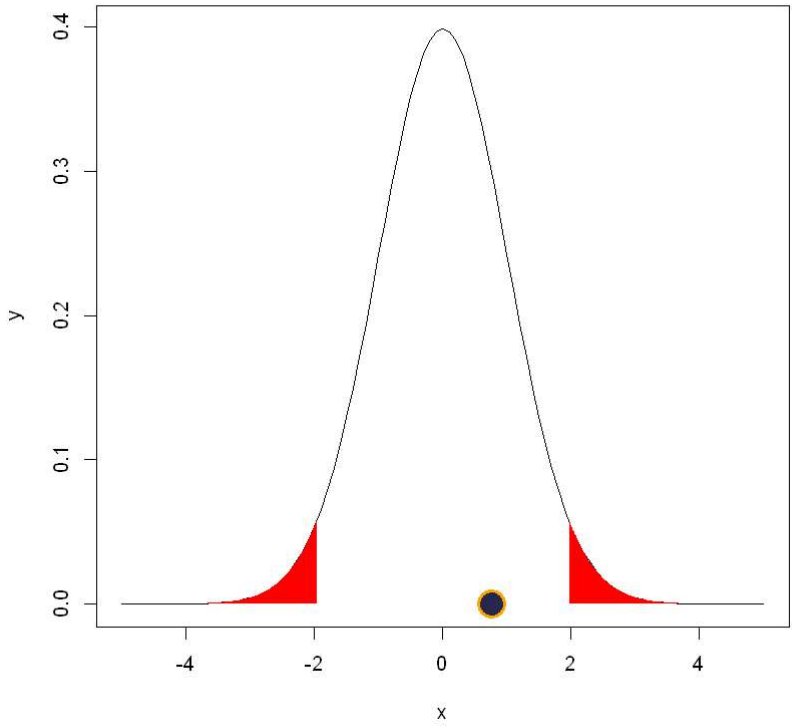
Conclusion: mean of population's total sulfur dioxide is NOT LOWER than 35

5. Is the proportion of the `total sulfur dioxide` which are more than 40 not 50%?

In [6]:
```
select <- df[df$"total.sulfur.dioxide" > 40,]
proportion <- nrow(select) / nrow(df)
z <- (proportion - 0.5) / sqrt(proportion*(1-proportion)/nrow(df))
z0 <- qnorm(0.025)

cat("z  :", z, "\n")
cat("z0 low :", z0, "\n")
cat("z0 high :", z0*-1, "\n")
cat("P-value:", 1-pnorm(z))

# Plotting Critical Area
x <- seq(-5, 5, 0.1)
y <- dnorm(x)
plot(x, y, type = "l")
```

```
x2 <- seq(-5,z0,0.01)
y2 <- dnorm(x2)
x2 = c(-5,x2,z0)
y2 = c(0,y2,0)
polygon(x2,y2, col="red", border=NA)

x3 <- seq(z0*-1,5,0.01)
y3 <- dnorm(x3)
x3 = c(x3,5,z0)
y3 = c(0,y3,0)
polygon(x3,y3, col="red", border=NA)

lines(z, 0, type = "o", pch=21, bg="#28284d", cex=3, lwd=3, col="orange")
```

```
z  : 0.7591653
z0 low : -1.959964
z0 high : 1.959964
P-value: 0.2238768
```



p = proportion of the `total sulfur dioxide` which are more than 40
H0 = (p == 0.5)
H1 = (p != 0.5)

Use the significance 0.05
Using normal distribution,
Critical area : z < z(-0.025), z > z(0.025)
Since z(-0.025) < z < z(0.025) (and p value > significance/2) which means z is NOT located in critical area. Hence, we accept H0.

Conclusion: mean of proportion of the total sulfur dioxide which are more than 40 is 50%

Tugas Besar IF2220 - Probabilitas dan Statistika

Part 5: Two Samples Hypothesis

Anggota:
13521116 - Juan Christopher Santoso
13521162 - Antonio Natthan Krishna

All testing use significant of 5%

In [1]:
```
# Import Dataset
df <- read.csv("..\\test\\anggur.csv")

# Data Statistics
properties <- c("Rows", "Columns")
value <- c(nrow(df), ncol(df))
cbind(properties, value)

# List of Columns
columns_index <- c(1:ncol(df))
columns_name <- colnames(df)

# Display List
cbind(columns_index, columns_name)

# Significance
Significance <- 0.05
```

A matrix: 2 × 2 of type chr

properties	value
Rows	1000
Columns	12

A matrix: 12 × 2 of type chr

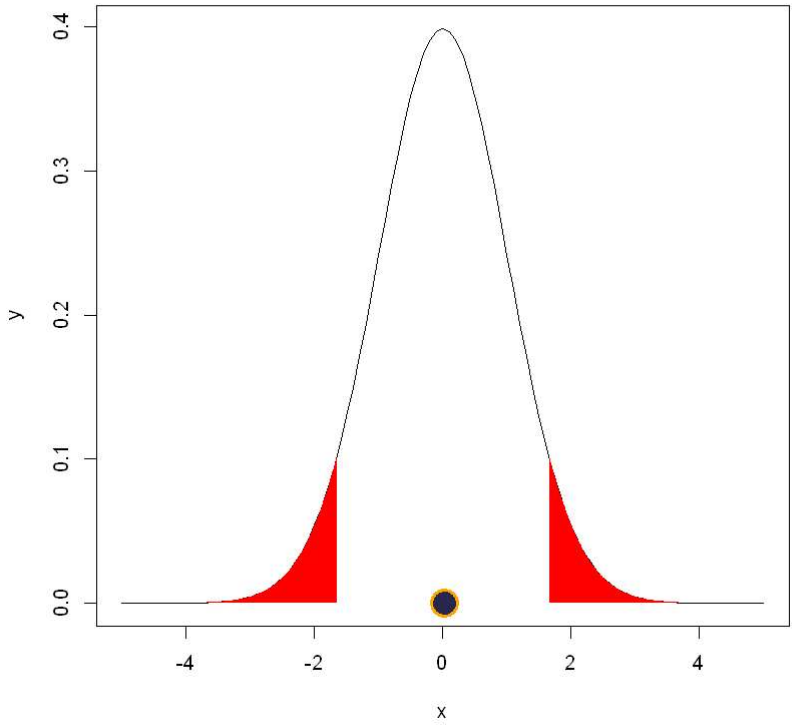
columns_index	columns_name
1	fixed.acidity
2	volatile.acidity
3	citric.acid
4	residual.sugar
5	chlorides
6	free.sulfur.dioxide
7	total.sulfur.dioxide
8	density
9	pH
10	sulphates
11	alcohol
12	quality

1. Data kolom fixed acidity dibagi 2 sama rata: bagian awal dan bagian akhir kolom. Benarkah rata-rata kedua bagian tersebut sama?

In [2]:
```
# Divide columns fixed acidity into 2 parts
numrow <- as.numeric(nrow(df)/2)
first_half <- df[1:numrow,]
second_half <- df[numrow+1:numrow,]

# Mean and Standard Deviation each part
first_half_mean <- mean(first_half,"fixed.acidity"])
second_half_mean <- mean(second_half,"fixed.acidity"])
first_half_sd <- sd(first_half,"fixed.acidity"])
second_half_sd <- sd(second_half,"fixed.acidity"])

T <- (first_half_mean - second_half_mean)/sqrt((first_half_sd^2)/numrow + (second_h
v <- round(((first_half_sd^2)/numrow + (second_half_sd^2)/numrow) ^ 2 / ((((first_h

t0low <- qt(0.05, v)
t0high <- qt(0.05, v, lower.tail = FALSE)

cat("T  :", T, "\n")
cat("v  :", v, "\n")
cat("t0 low :", t0low, "\n")
cat("t0 high :", t0high, "\n")
cat("P-value:", pt(T, v, lower.tail = FALSE))

# Plotting Critical Area
x <- seq(-5, 5, 0.1)
y <- dt(x, nrow(df))
```

```
plot(x, y, type = "l")

x2 <- seq(-5,t0low,0.01)
y2 <- dt(x2, nrow(df)-1)
x2 = c(-5,x2,t0low)
y2 = c(0,y2,0)
polygon(x2,y2, col="red", border=NA)

x3 <- seq(t0high,5,0.01)
y3 <- dt(x3, nrow(df)-1)
x3 = c(x3,5,t0high)
y3 = c(0,y3,0)
polygon(x3,y3, col="red", border=NA)

lines(T, 0, type = "o", pch=21, bg="#28284d", cex=3, lwd=3, col="orange")
```

```
T  : 0.02604107
v  : 998
t0 low : -1.646382
t0 high : 1.646382
P-value: 0.4896149
```



a = mean first half
b = mean second half

H0 = (a == b)
H1 = (a != b)

Use the significance 0.05
Using t distribution with degree 998,

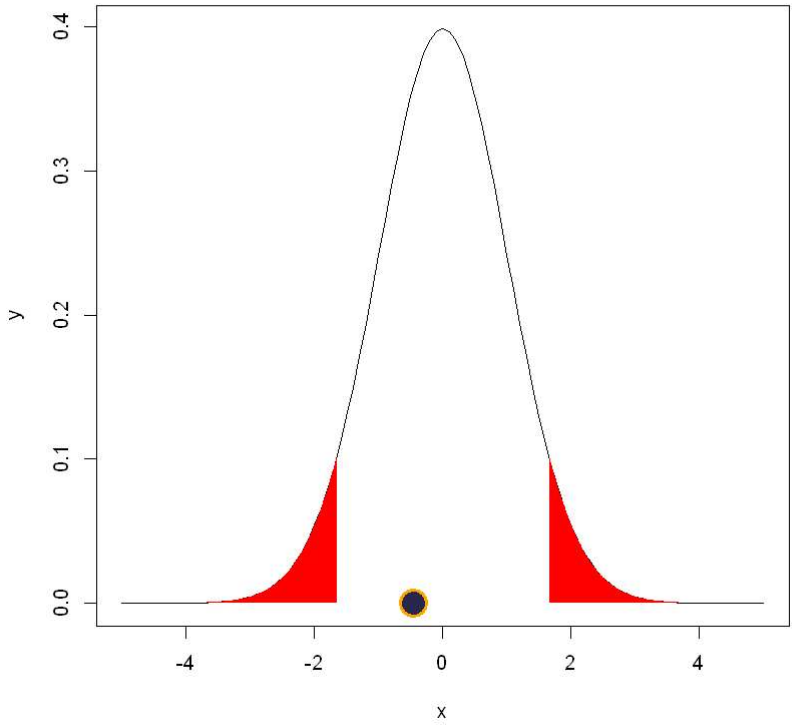
Critical area : t < t(-0.025), t > t(0.025)
Since t(-0.025) < t < t(0.025) (and p value > significance) which means t is NOT located in critical area. Hence, we accept H0.

Conclusion: mean first half is SAME as mean second half

2. Data kolom chlorides dibagi 2 sama rata: bagian awal dan bagian akhir kolom. Benarkah rata-rata bagian awal lebih besar daripada bagian akhir sebesar 0.001?

In [3]:
```
# Divide columns chlorides into 2 parts
numrow <- as.numeric(nrow(df)/2)
first_half <- df[1:numrow,]
second_half <- df[numrow+1:numrow,]

# Mean and Standard Deviation each part
first_half_mean <- mean(first_half[,"chlorides"])
second_half_mean <- mean(second_half[,"chlorides"])
first_half_sd <- sd(first_half[,"chlorides"])
second_half_sd <- sd(second_half[,"chlorides"])

T <- ((first_half_mean - second_half_mean)-0.001) / sqrt((first_half_sd^2)/numrow +
v <- round(((first_half_sd^2)/numrow + (second_half_sd^2)/numrow) ^ 2 / ((((first_h

t0low <- qt(0.05, v)
t0high <- qt(0.05, v, lower.tail = FALSE)

cat("T  :", T, "\n")
cat("v  :", v, "\n")
cat("t0 low :", t0low, "\n")
cat("t0 high :", t0high, "\n")
cat("P-value:", pt(T, v, lower.tail = FALSE))

# Plotting Critical Area
x <- seq(-5, 5, 0.1)
y <- dt(x, nrow(df))
plot(x, y, type = "l")

x2 <- seq(-5,t0low,0.01)
y2 <- dt(x2, nrow(df)-1)
x2 = c(-5,x2,t0low)
y2 = c(0,y2,0)
polygon(x2,y2, col="red", border=NA)

x3 <- seq(t0high,5,0.01)
y3 <- dt(x3, nrow(df)-1)
x3 = c(x3,5,t0high)
y3 = c(0,y3,0)
polygon(x3,y3, col="red", border=NA)

lines(T, 0, type = "o", pch=21, bg="#28284d", cex=3, lwd=3, col="orange")
```

```
T  : -0.4673171
v  : 998
t0 low : -1.646382
t0 high : 1.646382
P-value: 0.6798125
```



a = mean first half
b = mean second half

H0 = (a - b == 0.1)
H1 = (a - b != 0.1)

Use the significance 0.05
Using t distribution with degree 998,
Critical area : t < t(-0.025), t > t(0.025)
Since t(-0.025) < t < t(0.025) (and p value > significance) which means t is NOT located in critical area. Hence, we accept H0.

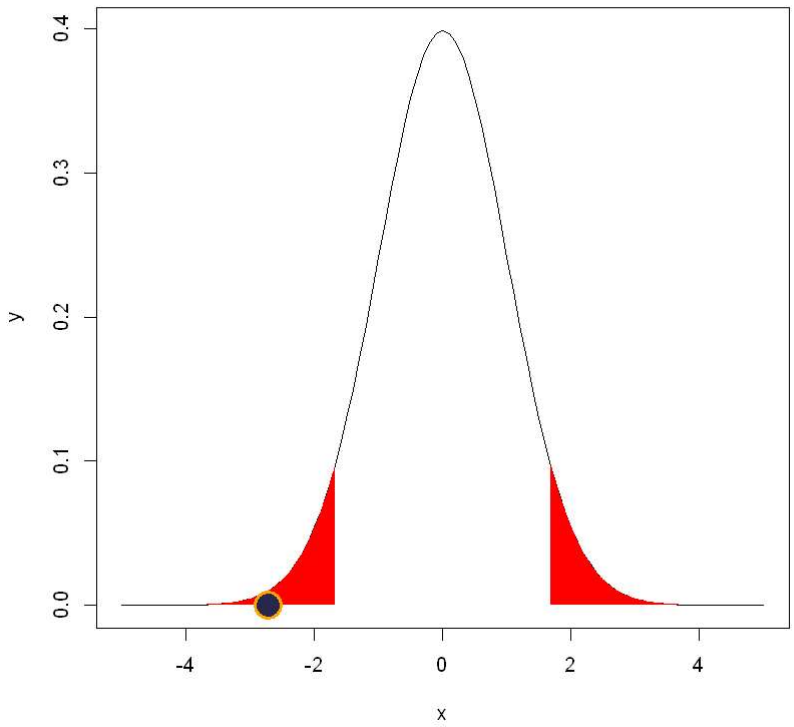
Conclusion: mean first half is GREATER than mean second half BY 0.001

3. Benarkah rata-rata sampel 25 baris pertama kolom Volatile Acidity sama dengan rata-rata 25 baris pertama kolom Sulphates ?

In [4]:
```
# Divide columns chlorides into 2 parts
numrow <- 25
first_25 <- df[1:25,]
```

```r
# Mean and Standard Deviation each part
volatile_acidity_mean <- mean(first_half[,"volatile.acidity"])
sulphates_mean <- mean(second_half[,"sulphates"])
volatile_acidity_sd <- sd(first_half[,"volatile.acidity"])
sulphates_sd <- sd(second_half[,"sulphates"])

T <- ((volatile_acidity_mean - sulphates_mean)-0.001) / sqrt((volatile_acidity_sd^2
v <- round(((volatile_acidity_sd^2)/numrow + (sulphates_sd^2)/numrow) ^ 2 / ((((vol

t0low <- qt(0.05, v)
t0high <- qt(0.05, v, lower.tail = FALSE)

cat("T  :", T, "\n")
cat("v  :", v, "\n")
cat("t0 low :", t0low, "\n")
cat("t0 high :", t0high, "\n")
cat("P-value:", pt(T, v, lower.tail = FALSE))

# Plotting Critical Area
x <- seq(-5, 5, 0.1)
y <- dt(x, nrow(df))
plot(x, y, type = "l")

x2 <- seq(-5,t0low,0.01)
y2 <- dt(x2, nrow(df)-1)
x2 = c(-5,x2,t0low)
y2 = c(0,y2,0)
polygon(x2,y2, col="red", border=NA)

x3 <- seq(t0high,5,0.01)
y3 <- dt(x3, nrow(df)-1)
x3 = c(x3,5,t0high)
y3 = c(0,y3,0)
polygon(x3,y3, col="red", border=NA)

lines(T, 0, type = "o", pch=21, bg="#28284d", cex=3, lwd=3, col="orange")
```

```
T  : -2.721924
v  : 48
t0 low : -1.677224
t0 high : 1.677224
P-value: 0.9954912
```



a = mean first 25 volatile acidity
b = mean first 25 sulphates

H0 = (a == b)
H1 = (a != b)

Use the significance 0.05
Using t distribution with degree 998,
Critical area : t < t(-0.025), t > t(0.025)
Since t < t(-0.025) (and p value < significance/2) which means t is located in critical area.
Hence, we reject H0.

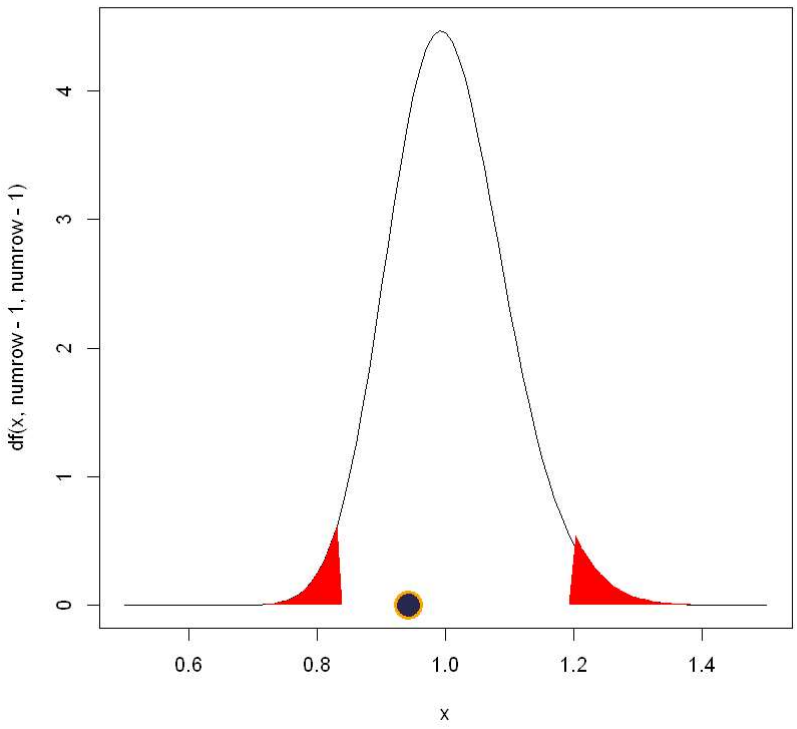
Conclusion: mean first 25 volatile acidity is NOT SAME as mean first 25 sulphates

4. Bagian awal kolom residual sugar memiliki variansi yang sama dengan bagian akhirnya?

In [5]:
```
numrow <- as.numeric(nrow(df)/2)
first_half <- df[1:numrow,]
second_half <- df[numrow+1:numrow,]

f <- sd(first_half[,"residual.sugar"])^2 / sd(second_half[,"residual.sugar"])^2
f0low <- qf(0.025, numrow-1, numrow-1)
f0high <- qf(0.025, numrow-1, numrow-1, lower.tail = FALSE)

cat("T  :", f, "\n")
cat("f0 low :", f0low, "\n")
```

```r
cat("f0 high :", f0high, "\n")
cat("P-value:", pf(f, numrow-1, numrow-1))

# Plotting Critical Area
x <- seq(0.5, 2, 0.01)
y <- df(x, numrow-1, numrow-1)
curve(df(x, numrow-1, numrow-1), 0.5, 1.5)

x2 <- seq(0.5,f0low,0.01)
y2 <- df(x2, numrow-1, numrow-1)
x2 = c(0.5,x2,f0low)
y2 = c(0,y2,0)
polygon(x2,y2, col="red", border=NA)

x3 <- seq(f0high,1.5,0.01)
y3 <- df(x3, numrow-1, numrow-1)
x3 = c(x3,1.5,f0high)
y3 = c(0,y3,0)
polygon(x3,y3, col="red", border=NA)

lines(f, 0, type = "o", pch=21, bg="#28284d", cex=3, lwd=3, col="orange")
```

```
T  : 0.9420041
f0 low : 0.8388858
f0 high : 1.192057
P-value: 0.2524102
```



a = variances first half

b = variances second half

H0 = (a == b)
H1 = (a != b)

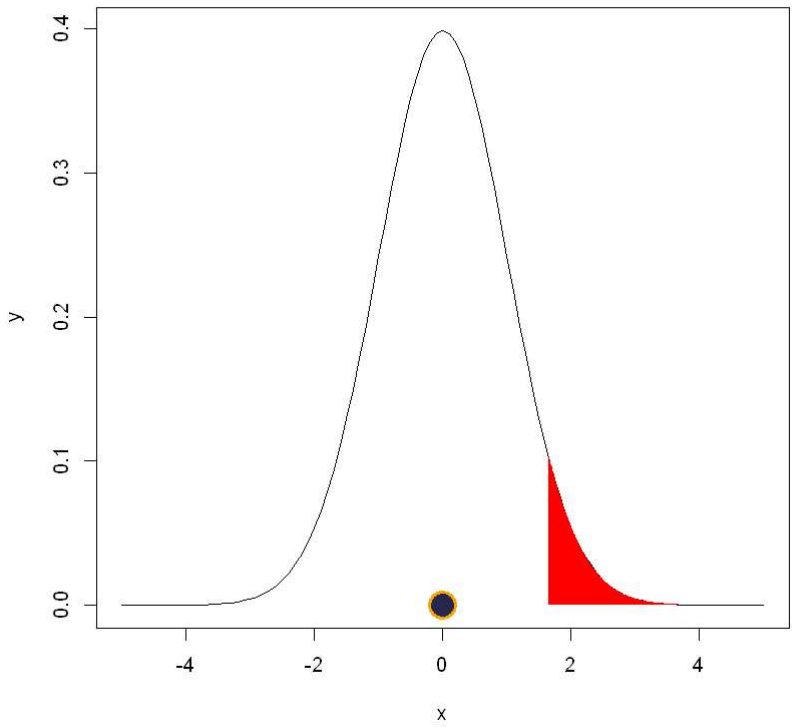
Use the significance 0.05
Using t distribution with degree 499 (first_half) and 499 (second_half)
Critical area : f < f(-0.025), f > f(0.025)
Since f(-0.025) < f < f(-0.025) (and p value > significance/2) which means f is NOT located in critical area. Hence, we accept H0.

Conclusion: variances first half is EQUAL to variances second half

5. Proporsi nilai setengah bagian awal alcohol yang lebih dari 7, adalah lebih besar daripada, proporsi nilai yang sama di setengah bagian akhir alcohol?

In [6]:
```
numrow <- as.numeric(nrow(df)/2)
first_half <- df[1:numrow,]
second_half <- df[numrow+1:numrow,]

proportion_first_half <- nrow(first_half[first_half$"alcohol" > 7,]) / numrow
proportion_second_half <- nrow(second_half[second_half$"alcohol" > 7,]) / numrow

z <- (proportion_first_half - proportion_second_half) / sqrt(proportion_first_half*
z0 <- qnorm(0.05, lower.tail = FALSE)

cat("z  :", z, "\n")
cat("z0 low :", z0, "\n")
cat("P-value:", 1-pnorm(z))

# Plotting Critical Area
x <- seq(-5, 5, 0.1)
y <- dnorm(x)
plot(x, y, type = "l")

x3 <- seq(z0,5,0.01)
y3 <- dnorm(x3)
x3 = c(x3,5,z0)
y3 = c(0,y3,0)
polygon(x3,y3, col="red", border=NA)

lines(z, 0, type = "o", pch=21, bg="#28284d", cex=3, lwd=3, col="orange")
```

```
z  : 0
z0 low : 1.644854
P-value: 0.5
```



a = proportion of first half alchohol which greater than 7
b = proportion of second half alcohol which greater than 7

H0 = (a == b)
H1 = (a > b)

Use the significance 0.05
Critical area : z > z(0.005)
Since z < z(0.005) (and p value > significance) which means z is NOT located in critical area.
Hence, we accept H0.

Conclusion: proportion first half alcohol which greater than 7 is NOT GREATER than proportion second half alcohol which greater than 7