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