



King Saud University
College of Computer and Information Sciences
Information Technology department

IT 326: Data Mining
Course Project

Water quality

Project Report: Data Summarization and Preprocessing

| Group#: | 4 | |
|----------------------|--------------------|----|
| Section#: | 52847 | |
| Group Members | Name | ID |
| | Jumanah aldawsari | |
| | Lama alshaya | |
| | Nouf alsadhan | |
| | Aljawharah alzamil | |

[Pick the date]

1 Problem

in our project we chose water quality dataset, it is an imaginary dataset in an urban environment, this dataset provide a set of elements (e.g. alumunium,ammonia,barium..etc) that can determine whether the water is safe or not.

Water is the most important natural resource on earth specially clean water which is the core to any community's success, we recognize the essential role of healthy water as the key to human growth and vitality. Globally, at least 2 billion people use a contaminated drinking water source, our goal is to bring knowledge to people and make this dataset a reference for people who wants to know if water is safe or not.

2 Data Mining Task

We will use classification and clustering as data mining task, for classification it is a binary classification , the class label is (is_safe) which has two possible values : 1 which indicates that water is safe , 0 which indicates that water is not safe ,The goal of classification in our case is to predict the water class label easier and faster and determining whether water is safe or not based in a set of elements(eg.Ammonia, Barium, Arsenic) provided in the dataset.

In clustering we will study and analyze water based on the elements which in turn represents the quality of the water, our goal is to find the similarity to identify distinct groups and clusters and what are the elements that distinguishes the water to be safe or not .

3 Data

Source: We chose the dataset from kaggle website (<https://www.kaggle.com/mssmartypants/water-quality>)

Number of objects is 7999(figure1), Number of attributes 21(figure2), with 3 missing values for all attribute(figure3) in row 7552 , 7569 and 7891 all in attribute ammonia .

```
> nrow(waterQuality)
[1] 7999
```

figure(1)

```
> ncol(waterQuality)
[1] 21
```

figure(2)

```
> sum(is.na(waterQuality))
[1] 3
figure(3)
```

Attributes information:

| Attribute name | Description | Data Type | Possible values |
|----------------|----------------------------------------------|-----------|-----------------|
| aluminum | <i>Level of aluminum per liter</i> | Numeric | 0~5.5 |
| ammonia | <i>Level of ammonia per liter</i> | Numeric | -2~30 |
| arsenic | <i>Level of arsenic per liter</i> | Numeric | 0~1.1 |
| barium | <i>Level of barium per liter</i> | Numeric | 0~5 |
| cadmium | <i>Level of cadmium per liter</i> | Numeric | 0~0.13 |
| chloramine | <i>Level of chloramine per liter</i> | Numeric | 0~9 |
| chromium | <i>Level of chromium per liter</i> | Numeric | 0~0.9 |
| copper | <i>Level of copper in water per liter.</i> | Numeric | 0~2 |
| fluoride | <i>Level of fluoride in water per liter.</i> | Numeric | 0~1.5 |
| bacteria | <i>Level of bacteria in water per liter.</i> | Numeric | 0~1 |
| viruses | <i>Level of viruses in water per liter.</i> | Numeric | 0~1 |
| lead | <i>Level of lead in water per liter.</i> | Numeric | 0~0.2 |

| | | | |
|--------------------|-------------------------------------------------|----------------|-------------------------------|
| <i>nitrates</i> | <i>Level of nitrates in water per liter.</i> | <i>Numeric</i> | <i>0~20</i> |
| <i>nitrites</i> | <i>Level of nitrites in water per liter.</i> | <i>Numeric</i> | <i>0~3</i> |
| <i>mercury</i> | <i>Level of mercury in water per liter.</i> | <i>Numeric</i> | <i>0~0.01</i> |
| <i>perchlorate</i> | <i>Level of perchlorate in water per liter.</i> | <i>Numeric</i> | <i>0~65</i> |
| <i>radium</i> | <i>Level of radium in water per liter.</i> | <i>Numeric</i> | <i>0~8</i> |
| <i>selenium</i> | <i>Level of selenium in water per liter.</i> | <i>Numeric</i> | <i>0~0.1</i> |
| <i>silver</i> | <i>Level of silver in water per liter.</i> | <i>Numeric</i> | <i>0~0.5</i> |
| <i>uranium</i> | <i>Level of uranium in water per liter.</i> | <i>Numeric</i> | <i>0~0.09</i> |
| <i>is_safe</i> | <i>Level of is_safe in water per liter.</i> | <i>Binary</i> | <i>0: not safe 1:safe</i> |

Five number summary:

```
> summary(waterQuality)
  aluminium      ammonia      arsenic
Min. :0.0000  Length:7999    Min. :0.0000
1st Qu.:0.0400 Class :character 1st Qu.:0.0300
Median :0.0700 Mode  :character Median :0.0500
Mean   :0.6662                           Mean  :0.1614
3rd Qu.:0.2800                           3rd Qu.:0.1000
Max.  :5.0500                           Max. :1.0500
  barium       cadmium      chloramine
Min. :0.0000  Min. :0.00000  Min. :0.000
1st Qu.:0.560  1st Qu.:0.00800 1st Qu.:0.100
Median :1.190  Median :0.04000  Median :0.530
Mean   :1.568  Mean   :0.04281  Mean   :2.177
3rd Qu.:2.480 3rd Qu.:0.07000 3rd Qu.:4.240
Max.  :4.940  Max.  :0.13000  Max.  :8.680
  chromium     copper      flouride
Min. :0.0000  Min. :0.0000  Min. :0.0000
1st Qu.:0.0500 1st Qu.:0.0900 1st Qu.:0.4050
Median :0.0900  Median :0.7500  Median :0.7700
Mean   :0.2472  Mean   :0.8059  Mean   :0.7716
3rd Qu.:0.4400 3rd Qu.:1.3900 3rd Qu.:1.1600
Max.  :0.9000  Max.  :2.0000  Max.  :1.5000
  silver       uranium      is_safe
Min. :0.0000  Min. :0.00000  Length:7999
1st Qu.:0.0400 1st Qu.:0.02000  Class :character
Median :0.0800  Median :0.05000  Mode  :character
Mean   :0.1478  Mean   :0.04467
3rd Qu.:0.2400 3rd Qu.:0.07000
Max.  :0.5000  Max.  :0.09000
```

↑

| | bacteria | viruses | lead |
|---------|-------------|----------------|------------------|
| Min. | :0.0000 | Min. :0.0000 | Min. :0.00000 |
| 1st Qu. | :0.0000 | 1st Qu.:0.0020 | 1st Qu.:0.04800 |
| Median | :0.2200 | Median :0.0080 | Median :0.10200 |
| Mean | :0.3197 | Mean :0.3286 | Mean :0.09945 |
| 3rd Qu. | :0.6100 | 3rd Qu.:0.7000 | 3rd Qu.:0.15100 |
| Max. | :1.0000 | Max. :1.0000 | Max. :0.20000 |
| | nitrates | nitrites | mercury |
| Min. | : 0.000 | Min. :0.00 | Min. :0.000000 |
| 1st Qu. | : 5.000 | 1st Qu.:1.00 | 1st Qu.:0.003000 |
| Median | : 9.930 | Median :1.42 | Median :0.005000 |
| Mean | : 9.819 | Mean :1.33 | Mean :0.005194 |
| 3rd Qu. | :14.610 | 3rd Qu.:1.76 | 3rd Qu.:0.008000 |
| Max. | :19.830 | Max. :2.93 | Max. :0.010000 |
| | perchlorate | radium | selenium |
| Min. | : 0.00 | Min. :0.000 | Min. :0.00000 |
| 1st Qu. | : 2.17 | 1st Qu.:0.820 | 1st Qu.:0.02000 |
| Median | : 7.74 | Median :2.410 | Median :0.05000 |
| Mean | :16.46 | Mean :2.921 | Mean :0.04968 |
| 3rd Qu. | :29.48 | 3rd Qu.:4.670 | 3rd Qu.:0.07000 |
| Max. | :60.01 | Max. :7.990 | Max. :0.10000 |

after we implement the code to show the five number summary (above) we noticed that Ammonia data type is character which is not helpful in our study

| | aluminium | ammonia | arsenic | barium | cadmium | chloramine |
|---|-----------|---------|------------------------|--------|---------|------------|
| 1 | 1.65 | 9.08 | column 2: character 40 | 2.85 | 0.007 | |
| 2 | 2.32 | 21.16 | | 3.31 | 0.002 | |
| 3 | 1.01 | 14.02 | | 0.58 | 0.008 | |
| 4 | 1.36 | 11.33 | | 2.96 | 0.001 | |
| 5 | 0.92 | 24.33 | | 0.20 | 0.006 | |
| 6 | 0.94 | 14.47 | | 2.88 | 0.003 | |
| 7 | 2.26 | 5.6 | | 1.25 | 0.004 | |

we think it is a mistake that happened while data entry process, so we decided to convert Ammonia data type from character to numeric so that we can analyse the five number summary .

```
> waterQuality$ammonia=as.numeric(waterQuality$ammonia)
```

Our dataset after the conversion

```
> summary(waterQuality)
aluminium      ammonia
Min. :0.0000  Min. :-0.080
1st Qu.:0.0400 1st Qu.: 6.577
Median :0.0700  Median :14.130
Mean   :0.6662  Mean   :14.278
3rd Qu.:0.2800 3rd Qu.:22.133
Max.   :5.0500  Max.   :29.840
NA's    :3

arsenic        barium
Min. :0.0000  Min. :0.000
1st Qu.:0.0300 1st Qu.:0.560
Median :0.0500  Median :1.190
Mean   :0.1614  Mean   :1.568
3rd Qu.:0.1000 3rd Qu.:2.480
Max.   :1.0500  Max.   :4.940

viruses        lead
Min. :0.0000  Min. :0.00000
1st Qu.:0.0020 1st Qu.:0.04800
Median :0.0080  Median :0.10200
Mean   :0.3286  Mean   :0.09945
3rd Qu.:0.7000 3rd Qu.:0.15100
Max.   :1.0000  Max.   :0.20000

nitrates       nitrites
Min. : 0.000  Min. :0.00
1st Qu.: 5.000 1st Qu.:1.00
Median : 9.930  Median :1.42
Mean   : 9.819  Mean   :1.33
3rd Qu.:14.610 3rd Qu.:1.76
Max.   :19.830  Max.   :2.93

mercury         perchlorate
Min. :0.000000  Min. : 0.00
1st Qu.:0.003000 1st Qu.: 2.17
Median :0.005000  Median : 7.74
Mean   :0.005194  Mean   :16.46
3rd Qu.:0.008000 3rd Qu.:29.48
Max.   :0.010000  Max.   :60.01
```

| | cadmium | chloramine |
|---------|-----------|-----------------|
| Min. | :0.00000 | Min. :0.000 |
| 1st Qu. | :0.00800 | 1st Qu.:0.100 |
| Median | :0.04000 | Median :0.530 |
| Mean | :0.04281 | Mean :2.177 |
| 3rd Qu. | :0.07000 | 3rd Qu.:4.240 |
| Max. | :0.13000 | Max. :8.680 |
| | | |
| | chromium | copper |
| Min. | :0.0000 | Min. :0.0000 |
| 1st Qu. | :0.0500 | 1st Qu.:0.0900 |
| Median | :0.0900 | Median :0.7500 |
| Mean | :0.2472 | Mean :0.8059 |
| 3rd Qu. | :0.4400 | 3rd Qu.:1.3900 |
| Max. | :0.9000 | Max. :2.0000 |
| | | |
| | flouride | bacteria |
| Min. | :0.0000 | Min. :0.0000 |
| 1st Qu. | :0.4050 | 1st Qu.:0.0000 |
| Median | :0.7700 | Median :0.2200 |
| Mean | :0.7716 | Mean :0.3197 |
| 3rd Qu. | :1.1600 | 3rd Qu.:0.6100 |
| Max. | :1.5000 | Max. :1.0000 |
| | | |
| | radium | selenium |
| Min. | :0.000 | Min. :0.00000 |
| 1st Qu. | :0.820 | 1st Qu.:0.02000 |
| Median | :2.410 | Median :0.05000 |
| Mean | :2.921 | Mean :0.04968 |
| 3rd Qu. | :4.670 | 3rd Qu.:0.07000 |
| Max. | :7.990 | Max. :0.10000 |
| | | |
| | silver | uranium |
| Min. | :0.0000 | Min. :0.00000 |
| 1st Qu. | :0.0400 | 1st Qu.:0.02000 |
| Median | :0.0800 | Median :0.05000 |
| Mean | :0.1478 | Mean :0.04467 |
| 3rd Qu. | :0.2400 | 3rd Qu.:0.07000 |
| Max. | :0.5000 | Max. :0.09000 |
| | | |
| | is_safe | |
| Length: | 7999 | |
| Class : | character | |
| Mode : | character | |

Outliers:

```
> OutlierAl = outlier(waterQuality$aluminium, logical=TRUE)
> sum(OutlierAl)
[1] 1
> OutlierAmm = outlier(waterQuality$ammonia, logical=TRUE)
> sum(OutlierAmm)
[1] NA
> OutlierArs = outlier(waterQuality$arsenic, logical=TRUE)
> sum(OutlierArs)
[1] 2
> OutlierBar = outlier(waterQuality$barium, logical=TRUE)
> sum(OutlierBar)
[1] 1
> OutlierCad = outlier(waterQuality$cadmium, logical=TRUE)
> sum(OutlierCad)
[1] 62
> OutlierChlo = outlier(waterQuality$chloramine, logical=TRUE)
> sum(OutlierChlo)
[1] 1
> OutlierChrom = outlier(waterQuality$chromium, logical=TRUE)
> sum(OutlierChrom)
[1] 12
> OutlierCo <- outlier(waterQuality$copper, logical=TRUE)
> sum(OutlierCo)
[1] 6
> OutlierFl <- outlier(waterQuality$flouride, logical=TRUE)
> sum(OutlierFl)
[1] 25
> OutlierBa <- outlier(waterQuality$bacteria, logical=TRUE)
> sum(OutlierBa)
[1] 45
> OutlierVi <- outlier(waterQuality$viruses, logical=TRUE)
> sum(OutlierVi)
[1] 62
> OutlierLe <- outlier(waterQuality$lead, logical=TRUE)
> sum(OutlierLe)
[1] 16
> OutlierNi1 <- outlier(waterQuality$nitrates, logical=TRUE)
> sum(OutlierNi1)
[1] 1
> OutlierNi2 <- outlier(waterQuality$nitrites, logical=TRUE)
> sum(OutlierNi2)
[1] 1
```

```
> OutlierMe <- outlier(waterQuality$mercury, logical=TRUE)
> sum(OutlierMe)
[1] 371
> OutlierPe <- outlier(waterQuality$perchlorate, logical=TRUE)
> sum(OutlierPe)
[1] 1
> OutlierRa <- outlier(waterQuality$radium, logical=TRUE)
> sum(OutlierRa)
[1] 4
> OutlierSe <- outlier(waterQuality$selenium, logical=TRUE)
> sum(OutlierSe)
[1] 400
> Outliersi <- outlier(waterQuality$silver, logical=TRUE)
> sum(Outliersi)
[1] 36
> OutlierUr <- outlier(waterQuality$uranium, logical=TRUE)
> sum(OutlierUr)
[1] 447
>
```

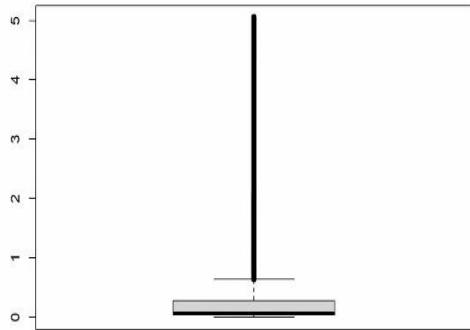
Correlation analysis:

We used correlation analysis to determine the relationship between two different attributes in our dataset and how strongly one attribute implies the other, and if they are dependant or independent of each other. Since our dataset attributes is all of numeric type, then the right method to be applied is correlation coefficient. Each two attributes will be postively correlated if the correlation coefficient value is larger than 0, and they will be negatively correlated if the value is negative which is smaller than 0, also the two values will be totally independent if the value is equal to 0.

```
-- 
> cor(waterQuality$lead, waterQuality$arsenic)
[1] -0.08775631 ← Negatively correlated
> cor(waterQuality$aluminium, waterQuality$silver)
[1] 0.3349928 ← Positively correlated
> cor(waterQuality$bacteria, waterQuality$viruses)
[1] 0.6184804 ← Positively correlated
> cor(waterQuality$selenium, waterQuality$arsenic)
[1] -0.007009078 ← Negatively correlated
```

Boxplots :

Aluminium



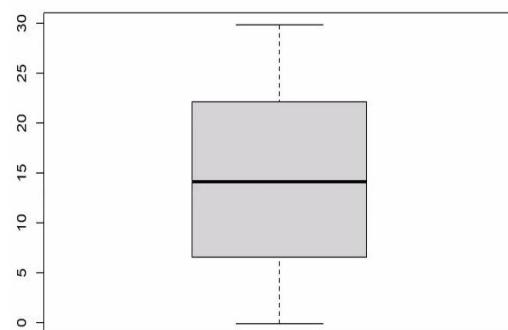
```
> boxplot(waterQuality$aluminium, data = waterQuality)
```

the boxplot show the values of the attribute

Aluminium, with min 0 and max 5.05 it show

that there are outliers Above The value 0.8.

Ammonia



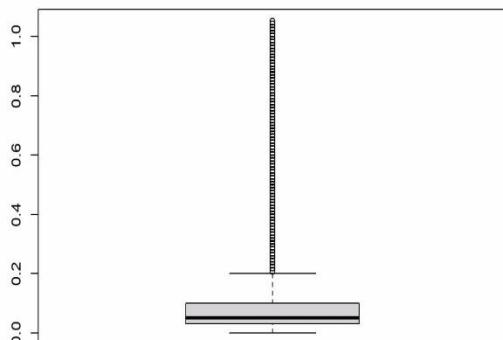
```
> boxplot(waterQuality$ammonia, data = waterQuality)
```

the boxplot show the values of the attribute

ammonia, with min -2 and max 30, there is no

outliers.

Arsenic



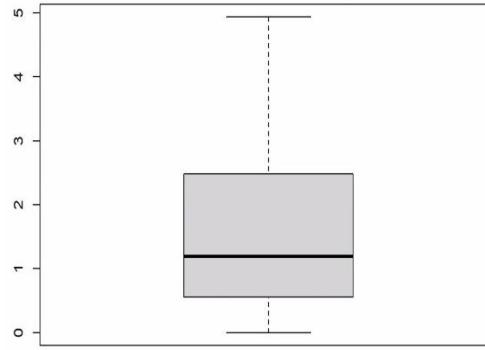
```
> boxplot(waterQuality$arсеніc, data = waterQuality)
```

the boxplot show the values of the attribute

arsenic, with min 0 and max 1.1, it show that

there are outliers Above The value 0.2

Barium

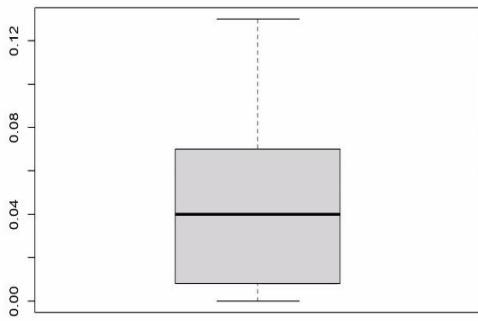


```
> boxplot(waterQuality$barium, data = waterQuality)
```

the boxplot show the values of the attribute

Barium, with min 0 and max 5 there is no outliers.

Cadmium

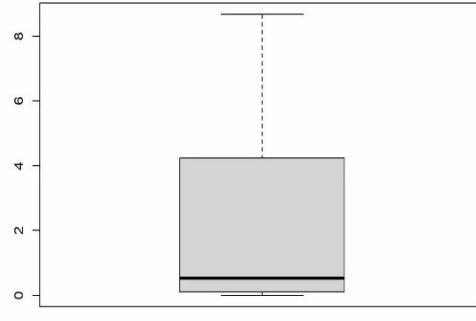


```
> boxplot(waterQuality$cadmium, data = waterQuality)
```

the boxplot show the values of the attribute

Cadmium, with min 0 and max 0.13 there is no outliers.

Chloramine

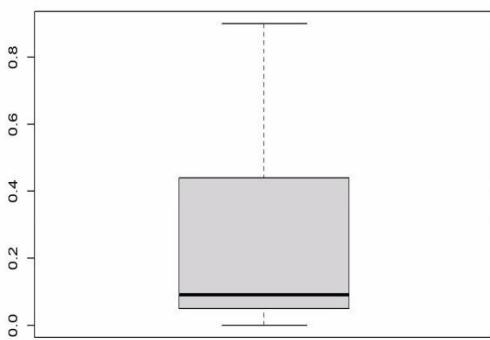


```
> boxplot(waterQuality$chloramine, data = waterQuality)
```

the boxplot show the values of the attribute

Chloramine, with min 0 and max 9 there is no outliers.

Chromium

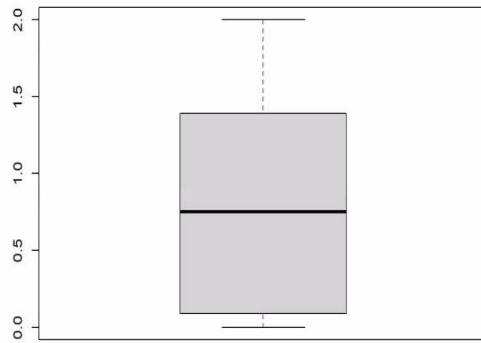


```
> boxplot(waterQuality$chromium, data = waterQuality)
```

the boxplot show the values of the attribute

Chromium, with min 0 and max 0.9 there is no outliers.

Copper

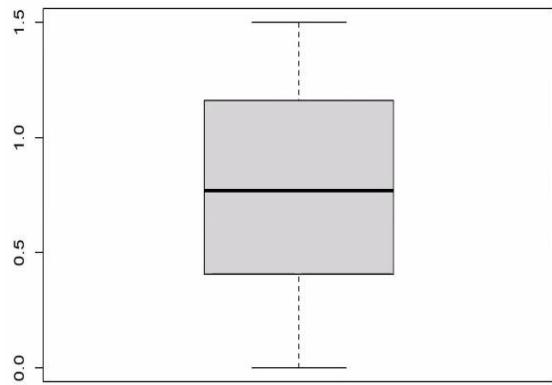


```
> boxplot(waterQuality$copper, data = waterQuality)
```

the boxplot show the values of the attribute

Copper, with min 0 and max 2 there is no outliers.

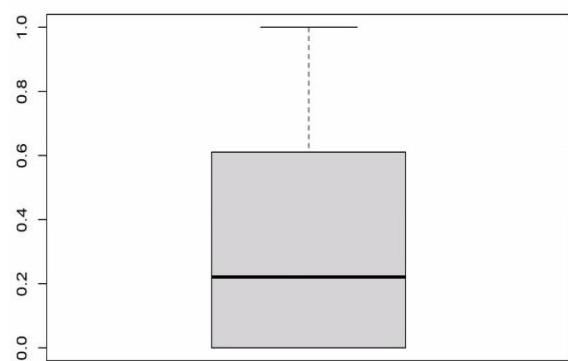
Fluoride



```
> boxplot(waterQuality$fluoride, data = waterQuality)
```

the boxplot show the values of the attribute
fluoride, with min 0 and max 1.5 there is
no outliers.

Bacteria



```
> boxplot(waterQuality$bacteria, data = waterQuality)
```

the boxplot show the values of the attribute
bacteria, with min 0 and max 1 there is no outliers.

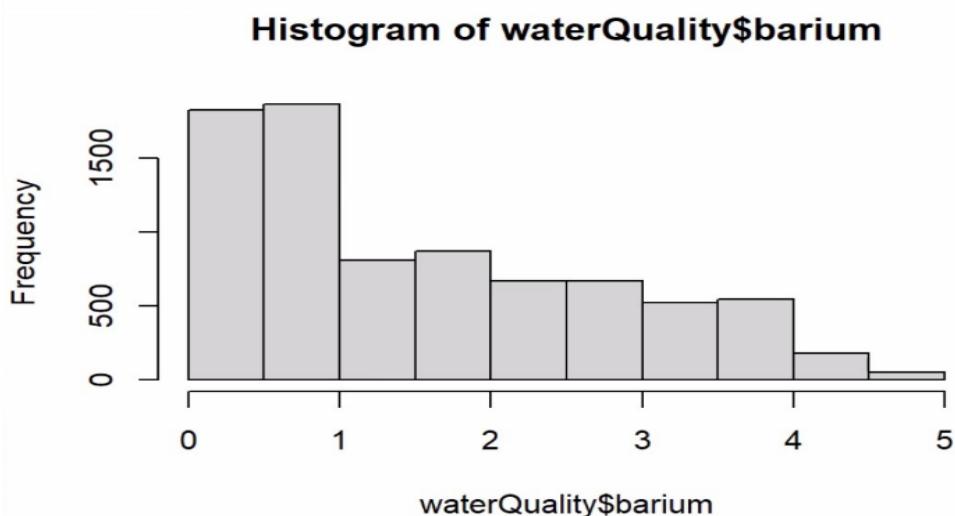
Plotting methods :

Histogram

The histogram shows the frequency of the barium, it shows that most water have barium concentration between 0-1.1 which is the normal concentration in a liter of water, also there are a few water which have between 4-5.1.

As you can see there is water that contain 0 barium, which is not normal because when barium is combined with other chemicals it effects water well.

```
> hist(waterQuality$barium)
```

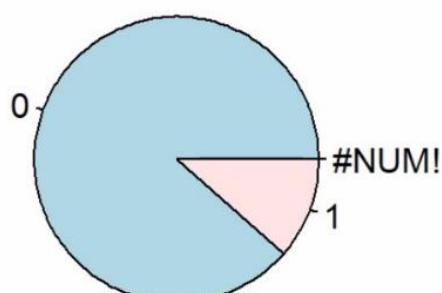


Pie chart

The pie chart shows that the data are not equally distributed between (0 not safe, 1 safe)

Based on class label (is_safe), cleaning needed since there are missing value. here we noticed that ("#Num!") is not one of the possible values (this problem is handled in preprocessing).

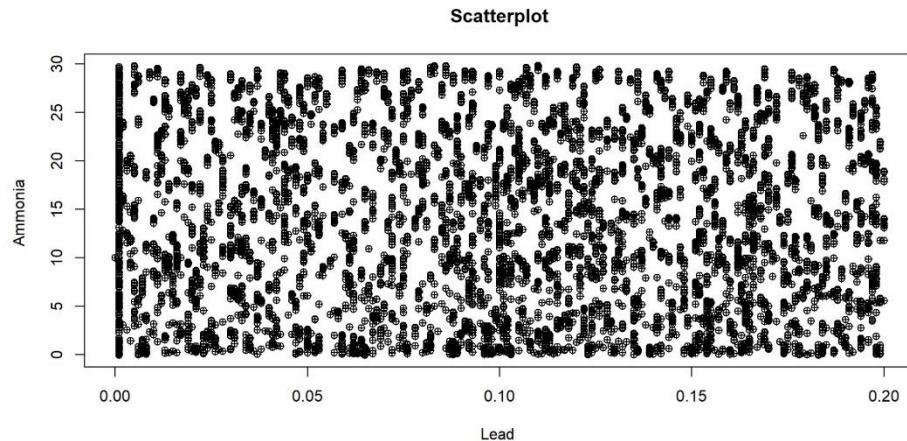
```
> pie(table(waterQuality$is_safe))
```



Scatterplot

This scatter plot shows that there is no correlation between ammonia and lead because there is huge dispersion between the points and there is no specific pattern .

```
> plot(waterQuality$lead,waterQuality$ammonia, main="Scatterplot",xlab="Lead",ylab="Ammonia")
```



4 Data preprocessing

1- Cleaning data for missing values :

Since dealing with missing values is an important step in data cleaning , in our case we have 3 missing values to deal with .

```
> sum(is.na(waterQuality))
[1] 3
```

By this command we locate the missing values .

```
> is.na(waterQuality$ammonia)
```

Output :

We decide to deal with missing values by replacing them with the average of Ammonia with this code.

```
> waterQuality$ammonia=ifelse(is.na(waterQuality$ammonia),ave(waterQuality$ammonia,FUN=function(x) mean(x,na.rm=TRUE)),waterQuality$ammonia)
>
```

Output : (rows with missing values after cleaning)

First row that was containing missing value

| | aluminium | ammonia | arsenic | barium | cadmium | chloramine |
|------|-----------|----------|---------|--------|---------|------------|
| 7889 | 0.04 | 17.00000 | 0.05 | 0.10 | 0.02 | 0 |
| 7890 | 0.10 | 4.79000 | 0.04 | 0.70 | 0.03 | 0 |
| 7891 | 0.01 | 14.27821 | 0.08 | 0.49 | 0.00 | 0 |
| 7892 | 0.05 | 16.61000 | 0.09 | 0.20 | 0.00 | 0 |
| 7893 | 0.08 | 3.25000 | 0.10 | 0.32 | 0.05 | 0 |
| 7894 | 0.05 | 0.05000 | 0.01 | 0.29 | 0.06 | 0 |

Second row that was containing missing value

| | aluminium | ammonia | arsenic | barium | cadmium | chloramine |
|------|-----------|----------|---------|--------|---------|------------|
| 7551 | 0.01 | 0.13000 | 0.05 | 0.28 | 0.04 | 0 |
| 7552 | 0.03 | 14.27821 | 0.08 | 0.79 | 0.07 | 0 |
| 7553 | 0.10 | 8.76000 | 0.10 | 0.66 | 0.07 | 0 |
| 7554 | 0.04 | 0.26000 | 0.08 | 1.58 | 0.08 | 0 |
| 7555 | 0.07 | 0.52000 | 0.01 | 0.31 | 0.04 | 0 |

Third row that was containing missing value

| | aluminium | ammonia | arsenic | barium | cadmium | chloramine |
|------|-----------|----------|---------|--------|---------|------------|
| 7568 | 0.06 | 8.84000 | 0.04 | 3.47 | 0.02 | 0 |
| 7569 | 0.06 | 14.27821 | 0.07 | 1.72 | 0.08 | 0 |
| 7570 | 0.09 | 0.26000 | 0.09 | 0.45 | 0.01 | 0 |
| 7571 | 0.00 | 8.57000 | 0.05 | 0.82 | 0.03 | 0 |
| 7572 | 0.08 | 14.23000 | 0.02 | 0.98 | 0.05 | 0 |

2- Detecting and removing outliers :

Outliers are values that are not usual in the dataset, excluding outliers can make the dataset statistically significant . We decide to deal with outliers to improve the dataset quality and make it more accurate.

We used this code to remove the outliers (the rest of the code are in appendix page below)

```
> library(outliers)
> OutlierUran=outlier(waterQuality$uranium,logical=TRUE)
> Find_outlier=which(OutlierUran==TRUE,arr.ind=TRUE)
> waterQuality= waterQuality[-Find_outlier,]
```

Dataset before removing outliers :

| Data | |
|--------------|---------------------------|
| waterQuality | 7999 obs. of 21 variables |

Dataset after removing outliers :

| Data | |
|--------------|----------------------------|
| waterQual... | 6627 obs. of 21 variabl... |

3- Encoding :

In our dataset we have one nominal attribute(binary) which is (is_safe) containing two possible values which is 0 indicating that water is not safe , 1 indicating water is safe.

We used this code on the class label (is_safe) to do the encoding process from (0,1) to(No,Yes) to make our data set more understandable .

```
> waterQuality$is_safe = factor(waterQuality$is_safe, levels = c("0","1"), labels = c("No","Yes"))
```

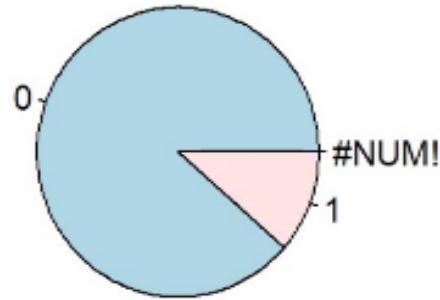
attribute(is_safe) before encoding :

| nitrites | mercury | perchlorate | radium | selenium | silver | uranium | is_safe |
|----------|---------|-------------|--------|----------|--------|---------|---------|
| 1.13 | 0.007 | 37.75 | 6.78 | 0.08 | 0.34 | 0.02 | 1 |
| 1.93 | 0.003 | 32.26 | 3.21 | 0.08 | 0.27 | 0.05 | 1 |
| 1.11 | 0.006 | 50.28 | 7.07 | 0.07 | 0.44 | 0.01 | 0 |
| 1.29 | 0.004 | 9.12 | 1.72 | 0.02 | 0.45 | 0.05 | 1 |
| 1.11 | 0.003 | 16.90 | 2.41 | 0.02 | 0.06 | 0.02 | 1 |
| 1.89 | 0.006 | 27.17 | 5.42 | 0.08 | 0.19 | 0.02 | 1 |
| 1.81 | 0.001 | 53.35 | 7.24 | 0.08 | 0.08 | 0.07 | 0 |
| 1.84 | 0.004 | 23.43 | 4.99 | 0.08 | 0.25 | 0.08 | 1 |
| 1.46 | 0.010 | 30.42 | 0.08 | 0.03 | 0.31 | 0.01 | 1 |
| 1.25 | 0.006 | 55.40 | 7.80 | 0.05 | 0.33 | 0.06 | 0 |

attribute(is_safe) after encoding :

| nitrites | mercury | perchlorate | radium | selenium | silver | uranium | is_safe |
|----------|---------|-------------|--------|----------|--------|---------|---------|
| 1.13 | 0.007 | 37.75 | 6.78 | 0.08 | 0.34 | 0.02 | Yes |
| 1.93 | 0.003 | 32.26 | 3.21 | 0.08 | 0.27 | 0.05 | Yes |
| 1.11 | 0.006 | 50.28 | 7.07 | 0.07 | 0.44 | 0.01 | No |
| 1.29 | 0.004 | 9.12 | 1.72 | 0.02 | 0.45 | 0.05 | Yes |
| 1.11 | 0.003 | 16.90 | 2.41 | 0.02 | 0.06 | 0.02 | Yes |
| 1.89 | 0.006 | 27.17 | 5.42 | 0.08 | 0.19 | 0.02 | Yes |
| 1.81 | 0.001 | 53.35 | 7.24 | 0.08 | 0.08 | 0.07 | No |
| 1.84 | 0.004 | 23.43 | 4.99 | 0.08 | 0.25 | 0.08 | Yes |
| 1.46 | 0.010 | 30.42 | 0.08 | 0.03 | 0.31 | 0.01 | Yes |

We noticed that after we created the pie chart it shows that the class label is_safe have three values which are (0 , 1, "#Num!") as shown .



"#Num!" is unknown value and ambiguous and not acceptable in our dataset.

when we encoded the class label (is_safe) we realized the "#Num!" becomes a null value so we used this code to delete the null rows and clean our data set from unaccepted values.

```
waterQuality <- na.omit(waterQuality)
```

Here is our dataset before deleting the null values :

```
waterQua... | 6627 obs. of 21 varia...
```

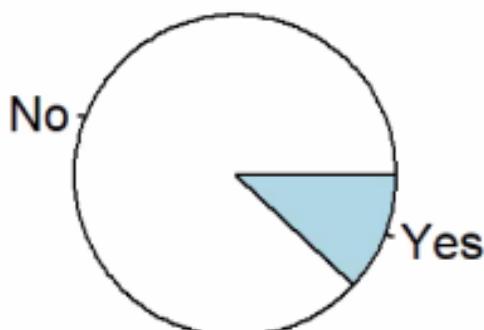
Here is our dataset after deleting the null values :

```
waterQuality | 6624 obs. of 21 variables
```

Then we used this code to show the pie chart without null values :

```
pie(table(waterQuality$is_safe))
```

then The pie chart appears without the unknown value ("#Num!"):



4- Normalization :

Normalization is where the attribute data are scaled so as to fall in a smaller range , we apply min-max normalization for the highest 3 attributes in range which are (Ammonia , Nitrates, Perchlorate).

We used this code to do normalization process :

```
94 ##Normalize ammonia
95 normlize<- function(x){
96   return((x-min(x)) / (max(x)-min(x)))
97 }
98 waterQuality$ammonia<-normlize(waterQuality$ammonia)
99 ##Normalize perchlorate
L00 normlize<- function(x){
L01   return((x-min(x)) / (max(x)-min(x)))
L02 }
L03 waterQuality$perchlorate<-normlize(waterQuality$perchlorate)
L04 ##Normalize nitrates
L05 normlize<- function(x){
L06   return((x-min(x)) / (max(x)-min(x)))
L07 }
L08 waterQuality$nitrates<-normlize(waterQuality$nitrates)
```

Nitrates before normalization :

| bacteria | viruses | lead | nitrates | nitrites | mercury | perchlorate | radium | selenium |
|----------|---------|-------|----------|----------|---------|-------------|--------|----------|
| 0.20 | 0.000 | 0.054 | 16.08 | 1.13 | 0.007 | 0.63032226 | 6.78 | |
| 0.65 | 0.650 | 0.100 | 2.01 | 1.93 | 0.003 | 0.53865420 | 3.21 | |
| 0.05 | 0.003 | 0.078 | 14.16 | 1.11 | 0.006 | 0.83953916 | 7.07 | |
| 0.71 | 0.710 | 0.016 | 1.41 | 1.29 | 0.004 | 0.15227918 | 1.72 | |
| 0.13 | 0.001 | 0.117 | 6.74 | 1.11 | 0.003 | 0.28218400 | 2.41 | |
| 0.67 | 0.670 | 0.135 | 9.75 | 1.89 | 0.006 | 0.45366505 | 5.42 | |
| 0.16 | 0.005 | 0.197 | 13.65 | 1.81 | 0.001 | 0.89079980 | 7.24 | |

Nitrates after normalization :

| opper | flouride | bacteria | viruses | lead | nitrates | nitrites | mercury | perchlorate |
|-------|----------|----------|---------|-------|------------|----------|---------|-------------|
| 0.17 | 0.05 | 0.20 | 0.000 | 0.054 | 0.81120646 | 1.13 | 0.007 | 0.630 |
| 0.66 | 0.90 | 0.65 | 0.650 | 0.100 | 0.10095911 | 1.93 | 0.003 | 0.538 |
| 0.02 | 0.99 | 0.05 | 0.003 | 0.078 | 0.71428571 | 1.11 | 0.006 | 0.839 |
| 1.66 | 1.08 | 0.71 | 0.710 | 0.016 | 0.07067138 | 1.29 | 0.004 | 0.152 |
| 0.57 | 0.61 | 0.13 | 0.001 | 0.117 | 0.33972741 | 1.11 | 0.003 | 0.282 |
| 1.38 | 0.11 | 0.67 | 0.670 | 0.135 | 0.49167087 | 1.89 | 0.006 | 0.453 |
| 1.86 | 0.86 | 0.16 | 0.005 | 0.197 | 0.68854114 | 1.81 | 0.001 | 0.890 |

Ammonia before normalization :

| | aluminium | ammonia | arsenic | barium | cadmium | chloramine |
|----|-----------|---------|---------|--------|---------|------------|
| 1 | 1.65 | 9.08 | 0.040 | 2.85 | 0.007 | |
| 2 | 2.32 | 21.16 | 0.010 | 3.31 | 0.002 | |
| 3 | 1.01 | 14.02 | 0.040 | 0.58 | 0.008 | |
| 4 | 1.36 | 11.33 | 0.040 | 2.96 | 0.001 | |
| 5 | 0.92 | 24.33 | 0.030 | 0.20 | 0.006 | |
| 6 | 0.94 | 14.47 | 0.030 | 2.88 | 0.003 | |
| 8 | 3.93 | 19.87 | 0.040 | 0.66 | 0.001 | |
| 9 | 0.60 | 24.58 | 0.010 | 0.71 | 0.005 | |
| 10 | 0.22 | 16.76 | 0.020 | 1.37 | 0.007 | |
| 11 | 3.27 | 3.60 | 0.001 | 2.69 | 0.005 | |
| 12 | 1.35 | 21.96 | 0.040 | 0.84 | 0.002 | |
| 13 | 1.88 | 19.26 | 0.020 | 2.78 | 0.008 | |

Ammonia after normalization :

| | aluminium | ammonia | arsenic | barium | cadmium | chloramine | chromium | cop |
|----|-----------|-------------|---------|--------|---------|------------|----------|-----|
| 1 | 1.65 | 0.305917753 | 0.040 | 2.85 | 0.007 | 0.35 | 0.83 | |
| 2 | 2.32 | 0.709796055 | 0.010 | 3.31 | 0.002 | 5.28 | 0.68 | |
| 3 | 1.01 | 0.471079906 | 0.040 | 0.58 | 0.008 | 4.24 | 0.53 | |
| 4 | 1.36 | 0.381143430 | 0.040 | 2.96 | 0.001 | 7.23 | 0.03 | |
| 5 | 0.92 | 0.815780675 | 0.030 | 0.20 | 0.006 | 2.67 | 0.69 | |
| 6 | 0.94 | 0.486125042 | 0.030 | 2.88 | 0.003 | 0.80 | 0.43 | |
| 8 | 3.93 | 0.666666667 | 0.040 | 0.66 | 0.001 | 6.22 | 0.10 | |
| 9 | 0.60 | 0.824139084 | 0.010 | 0.71 | 0.005 | 3.14 | 0.77 | |
| 10 | 0.22 | 0.562688064 | 0.020 | 1.37 | 0.007 | 6.40 | 0.49 | |
| 11 | 3.27 | 0.122721122 | 0.001 | 3.60 | 0.005 | 5.75 | 0.15 | |

Perchlorate before normalization :

| | nitrates | nitrites | mercury | perchlorate | radium | selenium | si |
|----|-----------------|-----------------|----------------|--------------------|---------------|-----------------|-----------|
| 16 | 1.41 | 1.29 | 0.004 | 9.12 | 1.72 | 0.02 | |
| 17 | 6.74 | 1.11 | 0.003 | 16.90 | 2.41 | 0.02 | |
| 35 | 9.75 | 1.89 | 0.006 | 27.17 | 5.42 | 0.08 | |
| 97 | 13.65 | 1.81 | 0.001 | 53.35 | 7.24 | 0.08 | |
| 57 | 14.66 | 1.84 | 0.004 | 23.43 | 4.99 | 0.08 | |
| 09 | 4.79 | 1.46 | 0.010 | 30.42 | 0.08 | 0.03 | |
| 45 | 8.47 | 1.25 | 0.006 | 55.40 | 7.80 | 0.05 | |
| 11 | 18.40 | 1.49 | 0.009 | 21.52 | 1.30 | 0.08 | |
| 03 | 4.37 | 1.95 | 0.006 | 22.12 | 1.97 | 0.03 | |
| 01 | 1.16 | 1.11 | 0.008 | 26.80 | 5.58 | 0.09 | |
| 17 | 1.99 | 1.08 | 0.007 | 11.16 | 0.98 | 0.01 | |
| 30 | 1.91 | 1.20 | 0.008 | 0.18 | 6.89 | 0.06 | |

Perchlorate after normalization :

| | nitrites | mercury | perchlorate | radium | selenium | silver | uranium | is_safe |
|----|-----------------|----------------|--------------------|---------------|-----------------|---------------|----------------|----------------|
| 08 | 1.13 | 0.007 | 0.63032226 | 6.78 | 0.08 | 0.34 | 0.02 | Yes |
| 01 | 1.93 | 0.003 | 0.53865420 | 3.21 | 0.08 | 0.27 | 0.05 | Yes |
| 6 | 1.11 | 0.006 | 0.83953916 | 7.07 | 0.07 | 0.44 | 0.01 | No |
| 11 | 1.29 | 0.004 | 0.15227918 | 1.72 | 0.02 | 0.45 | 0.05 | Yes |
| 4 | 1.11 | 0.003 | 0.28218400 | 2.41 | 0.02 | 0.06 | 0.02 | Yes |
| 5 | 1.89 | 0.006 | 0.45366505 | 5.42 | 0.08 | 0.19 | 0.02 | Yes |
| 55 | 1.81 | 0.001 | 0.89079980 | 7.24 | 0.08 | 0.08 | 0.07 | No |

Dataset before :

| | waterQuality | | | | | | | | | | | | | | | | | | | | | | | |
|----|--------------|---------|---------|--------|---------|------------|----------|--------|----------|----------|---------|-------|------------|----------|---------|-------------|--------|----------|--------|---------|---------|--|--|--|
| | aluminum | ammonia | arsenic | barium | cadmium | chloramine | chromium | copper | fluoride | bacteria | viruses | lead | nitrates | nitrates | mercury | perchlorate | radium | selenium | silver | uranium | is_safe | | | |
| 1 | 1.65 | 9.08 | 0.040 | 2.85 | 0.007 | 0.35 | 0.83 | 0.17 | 0.05 | 0.20 | 0.000 | 0.054 | 0.81120461 | 1.13 | 0.007 | 0.63032257 | 6.78 | 0.08 | 0.34 | 0.02 | 1 | | | |
| 2 | 2.32 | 21.16 | 0.010 | 3.31 | 0.002 | 5.28 | 0.68 | 0.66 | 0.90 | 0.65 | 0.650 | 0.100 | 2.01 | 1.93 | 0.003 | 32.26 | 3.21 | 0.08 | 0.27 | 0.05 | 1 | | | |
| 3 | 1.01 | 14.02 | 0.040 | 0.58 | 0.008 | 4.24 | 0.53 | 0.02 | 0.99 | 0.05 | 0.003 | 0.078 | 14.16 | 1.11 | 0.006 | 50.28 | 7.07 | 0.07 | 0.44 | 0.01 | 0 | | | |
| 4 | 1.36 | 11.33 | 0.040 | 2.96 | 0.001 | 7.23 | 0.03 | 1.66 | 1.08 | 0.71 | 0.710 | 0.016 | 1.41 | 1.29 | 0.004 | 9.12 | 1.72 | 0.02 | 0.45 | 0.05 | 1 | | | |
| 5 | 0.92 | 24.33 | 0.030 | 0.20 | 0.006 | 2.67 | 0.69 | 0.57 | 0.61 | 0.13 | 0.001 | 0.117 | 6.74 | 1.11 | 0.003 | 16.90 | 2.41 | 0.02 | 0.06 | 0.02 | 1 | | | |
| 6 | 0.94 | 14.47 | 0.030 | 2.88 | 0.003 | 0.80 | 0.43 | 1.38 | 0.11 | 0.67 | 0.670 | 0.135 | 9.75 | 1.89 | 0.006 | 27.17 | 5.42 | 0.08 | 0.19 | 0.02 | 1 | | | |
| 7 | 2.36 | 5.6 | 0.010 | 1.35 | 0.004 | 1.28 | 0.62 | 1.88 | 0.33 | 0.13 | 0.007 | 0.021 | 18.60 | 1.78 | 0.007 | 45.34 | 2.84 | 0.10 | 0.24 | 0.08 | 0 | | | |
| 8 | 3.93 | 19.87 | 0.040 | 0.66 | 0.001 | 6.22 | 0.10 | 1.86 | 0.86 | 0.16 | 0.005 | 0.197 | 13.65 | 1.81 | 0.001 | 53.35 | 7.24 | 0.08 | 0.08 | 0.07 | 0 | | | |
| 9 | 0.60 | 24.58 | 0.010 | 0.71 | 0.005 | 3.14 | 0.77 | 1.45 | 0.98 | 0.35 | 0.002 | 0.167 | 14.66 | 1.84 | 0.004 | 23.43 | 4.99 | 0.08 | 0.25 | 0.08 | 1 | | | |
| 10 | 0.22 | 16.76 | 0.020 | 1.37 | 0.007 | 6.40 | 0.49 | 0.02 | 1.24 | 0.03 | 0.830 | 0.109 | 4.79 | 1.46 | 0.010 | 30.42 | 0.08 | 0.03 | 0.31 | 0.01 | 1 | | | |
| 11 | 3.27 | 3.6 | 0.001 | 2.69 | 0.005 | 5.75 | 0.15 | 0.60 | 1.29 | 0.04 | 0.008 | 0.145 | 6.47 | 1.25 | 0.006 | 55.40 | 7.80 | 0.05 | 0.33 | 0.06 | 0 | | | |
| 12 | 1.35 | 21.96 | 0.040 | 0.84 | 0.002 | 0.10 | 0.76 | 0.17 | 0.58 | 0.52 | 0.520 | 0.011 | 18.40 | 1.49 | 0.009 | 21.52 | 1.30 | 0.08 | 0.48 | 0.08 | 1 | | | |
| 13 | 1.88 | 19.26 | 0.020 | 2.78 | 0.008 | 0.05 | 0.42 | 1.00 | 0.09 | 0.91 | 0.910 | 0.103 | 4.37 | 1.95 | 0.006 | 22.12 | 1.97 | 0.03 | 0.06 | 0.05 | 1 | | | |
| 14 | 4.93 | 23.98 | 0.040 | 3.05 | 0.008 | 0.70 | 0.51 | 1.35 | 1.07 | 0.70 | 0.700 | 0.101 | 1.16 | 1.11 | 0.008 | 26.80 | 5.58 | 0.09 | 0.38 | 0.03 | 1 | | | |
| 15 | 2.89 | 18.82 | 0.050 | 3.77 | 0.008 | 5.99 | 0.54 | 0.79 | 0.54 | 0.20 | 0.009 | 0.126 | 17.56 | 1.82 | 0.009 | 17.54 | 4.33 | 0.10 | 0.05 | 0.02 | 1 | | | |
| 16 | 0.61 | 2.41 | 0.030 | 0.59 | 0.002 | 1.94 | 0.77 | 1.54 | 0.62 | 0.23 | 0.001 | 0.017 | 1.99 | 1.06 | 0.007 | 11.16 | 0.56 | 0.01 | 0.47 | 0.03 | 1 | | | |
| 17 | 3.47 | 15.84 | 0.020 | 0.06 | 0.001 | 5.29 | 0.47 | 1.08 | 1.43 | 0.89 | 0.890 | 0.080 | 1.91 | 1.20 | 0.008 | 0.18 | 6.89 | 0.06 | 0.12 | 0.08 | 1 | | | |
| 18 | 2.11 | 17.03 | 0.020 | 0.88 | 0.009 | 7.78 | 0.88 | 1.15 | 0.34 | 0.85 | 0.850 | 0.065 | 17.86 | 1.53 | 0.003 | 19.40 | 1.14 | 0.10 | 0.40 | 0.01 | 1 | | | |
| 19 | 4.88 | 26.94 | 0.020 | 0.36 | 0.001 | 1.21 | 0.68 | 0.71 | 0.99 | 0.75 | 0.750 | 0.071 | 0.31 | 1.22 | 0.002 | 56.70 | 1.00 | 0.00 | 0.41 | 0.05 | 0 | | | |
| 20 | 4.12 | 17.99 | 0.020 | 3.43 | 0.006 | 0.01 | 0.41 | 1.62 | 0.22 | 0.99 | 0.990 | 0.108 | 8.06 | 1.76 | 0.005 | 24.29 | 0.68 | 0.10 | 0.10 | 0.07 | 1 | | | |
| 21 | 0.68 | 18.99 | 0.001 | 0.04 | 0.006 | 4.57 | 0.20 | 1.10 | 1.00 | 0.92 | 0.920 | 0.06 | 9.46 | 1.41 | 0.007 | 21.79 | 3.05 | 0.03 | 0.13 | 0.08 | 1 | | | |
| 22 | 1.15 | 8.12 | 0.020 | 0.97 | 0.007 | 3.47 | 0.65 | 1.51 | 1.46 | 0.58 | 0.590 | 0.061 | 8.96 | 1.50 | 0.004 | 14.60 | 1.74 | 0.03 | 0.01 | 0.06 | 1 | | | |
| 23 | 0.77 | 10.67 | 0.020 | 0.55 | 0.001 | 3.74 | 0.12 | 1.77 | 0.43 | 0.80 | 0.800 | 0.114 | 12.69 | 1.18 | 0.008 | 34.64 | 0.90 | 0.02 | 0.16 | 0.06 | 1 | | | |
| 24 | 4.32 | 20.64 | 0.030 | 2.60 | 0.008 | 7.24 | 0.61 | 1.23 | 1.44 | 0.56 | 0.560 | 0.012 | 9.42 | 1.74 | 0.004 | 36.23 | 3.22 | 0.07 | 0.18 | 0.08 | 0 | | | |
| 25 | 2.36 | 27.05 | 0.010 | 0.68 | 0.003 | 4.07 | 0.13 | 1.34 | 0.29 | 0.96 | 0.960 | 0.167 | 15.05 | 1.98 | 0.002 | 56.32 | 7.99 | 0.06 | 0.50 | 0.06 | 0 | | | |
| 26 | 3.31 | 22.07 | 0.030 | 0.46 | 0.001 | 7.22 | 0.73 | 1.05 | 1.00 | 0.25 | 0.007 | 0.109 | 1.92 | 1.07 | 0.001 | 39.40 | 0.49 | 0.04 | 0.47 | 0.05 | 1 | | | |
| 27 | 1.62 | 6.81 | 0.010 | 0.85 | 0.006 | 2.55 | 0.25 | 1.09 | 1.35 | 0.16 | 0.002 | 0.031 | 16.99 | 1.70 | 0.007 | 44.76 | 1.15 | 0.08 | 0.26 | 0.08 | 1 | | | |
| 28 | 3.42 | 2.4 | 0.001 | 2.80 | 0.003 | 2.87 | 0.73 | 0.27 | 0.53 | 0.44 | 0.002 | 0.110 | 13.73 | 1.69 | 0.009 | 55.27 | 3.29 | 0.04 | 0.17 | 0.00 | 0 | | | |
| 29 | 4.41 | 15.14 | 0.030 | 1.76 | 0.007 | 6.63 | 0.62 | 1.57 | 0.26 | 0.69 | 0.690 | 0.182 | 1.49 | 1.81 | 0.008 | 24.91 | 2.39 | 0.01 | 0.20 | 0.05 | 1 | | | |
| 30 | 4.57 | 25.84 | 0.010 | 3.04 | 0.002 | 2.78 | 0.72 | 0.46 | 1.41 | 0.08 | 0.006 | 0.049 | 9.64 | 1.43 | 0.006 | 36.57 | 4.55 | 0.03 | 0.19 | 0.00 | 0 | | | |
| 31 | 1.69 | 22.8 | 0.001 | 2.01 | 0.002 | 5.87 | 0.78 | 1.53 | 0.87 | 0.13 | 0.008 | 0.079 | 8.64 | 1.19 | 0.007 | 24.61 | 7.77 | 0.06 | 0.11 | 0.05 | 1 | | | |
| 32 | 1.07 | 6.1 | 0.050 | 3.74 | 0.006 | 2.86 | 0.52 | 1.03 | 0.51 | 0.77 | 0.770 | 0.012 | 1.11 | 1.55 | 0.008 | 20.32 | 2.21 | 0.02 | 0.48 | 0.04 | 1 | | | |
| 33 | 2.71 | 26.19 | 0.040 | 3.02 | 0.006 | 4.00 | 0.17 | 1.56 | 0.91 | 0.38 | 0.005 | 0.013 | 11.00 | 1.00 | 0.006 | 14.16 | 6.16 | 0.05 | 0.05 | 0.08 | 1 | | | |
| 34 | 1.63 | 15.75 | 0.030 | 2.54 | 0.008 | 4.25 | 0.74 | 1.35 | 1.24 | 0.12 | 0.006 | 0.193 | 12.79 | 1.27 | 0.001 | 58.15 | 1.71 | 0.09 | 0.44 | 0.05 | 0 | | | |
| 35 | 0.01 | 29.29 | 0.001 | 2.93 | 0.007 | 7.75 | 0.68 | 1.09 | 0.00 | 0.80 | 0.800 | 0.091 | 17.33 | 1.86 | 0.001 | 43.09 | 3.76 | 0.09 | 0.30 | 0.00 | 0 | | | |
| 36 | 4.49 | 4.07 | 0.050 | 1.87 | 0.005 | 4.55 | 0.32 | 0.90 | 0.55 | 0.30 | 0.002 | 0.135 | 5.23 | 1.43 | 0.004 | 36.52 | 6.15 | 0.04 | 0.16 | 0.04 | 1 | | | |
| 37 | 3.52 | 23.19 | 0.030 | 2.69 | 0.006 | 4.94 | 0.76 | 1.05 | 1.24 | 0.75 | 0.750 | 0.124 | 1.91 | 1.00 | 0.008 | 24.84 | 3.19 | 0.08 | 0.37 | 0.06 | 1 | | | |
| 38 | 4.35 | 26.85 | 0.020 | 0.88 | 0.004 | 0.64 | 0.61 | 0.91 | 0.42 | 0.44 | 0.006 | 0.071 | 0.094 | 1.75 | 0.002 | 94.673562 | 1.00 | 0.00 | 0.41 | 0.05 | 0 | | | |
| 39 | 2.51 | 13.08 | 0.040 | 1.58 | 0.002 | 2.86 | 0.87 | 1.88 | 0.40 | 0.13 | 0.004 | 0.085 | 11.96 | 1.35 | 0.008 | 46.30 | 4.70 | 0.10 | 0.04 | 0.05 | 0 | | | |
| 40 | 2.34 | 19.9 | 0.030 | 2.67 | 0.006 | 4.30 | 0.26 | 0.08 | 0.34 | 0.26 | 0.003 | 0.146 | 7.96 | 1.56 | 0.006 | 51.39 | 2.41 | 0.09 | 0.12 | 0.05 | 0 | | | |
| 41 | 1.71 | 9.49 | 0.050 | 3.63 | 0.006 | 2.24 | 0.90 | 0.22 | 0.68 | 0.63 | 0.162 | 11.40 | 1.83 | 0.005 | 42.64 | 6.14 | 0.09 | 0.44 | 0.06 | 0 | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---------|------|--------|-------|--------|--------|-----|------|------------|------|-------|-------|------------|-----------|--------------|-------------|-------------|------------|-----------|----------|------------|-----------|--------------|-------------|--------------|-------------|-----------------|----------------|
| Windows | blue | purple | green | yellow | orange | red | pink | light blue | grey | black | white | light grey | dark grey | light orange | dark orange | light green | dark green | light red | dark red | light pink | dark pink | light yellow | dark yellow | light purple | dark purple | light blue-grey | dark blue-grey |
|---------|------|--------|-------|--------|--------|-----|------|------------|------|-------|-------|------------|-----------|--------------|-------------|-------------|------------|-----------|----------|------------|-----------|--------------|-------------|--------------|-------------|-----------------|----------------|

Appendix: this code show the outlier remove

```
33 library(outliers)
34 OutlierUran=outlier(waterQuality$uranium,logical=TRUE)
35 Find_outlier=which(OutlierUran==TRUE,arr.ind=TRUE)
36 waterQuality= waterQuality[-Find_outlier,]
37 OutlierAl=outlier(waterQuality$aluminium,logical=TRUE)
38 Find_outlier=which(OutlierAl==TRUE,arr.ind=TRUE)
39 waterQuality= waterQuality[-Find_outlier,]
40 OutlierArs=outlier(waterQuality$arsenic,logical=TRUE)
41 Find_outlier=which(OutlierArs==TRUE,arr.ind=TRUE)
42 waterQuality= waterQuality[-Find_outlier,]
43 OutlierBar=outlier(waterQuality$barium,logical=TRUE)
44 Find_outlier=which(OutlierBar==TRUE,arr.ind=TRUE)
45 waterQuality= waterQuality[-Find_outlier,]
46 OutlierCad=outlier(waterQuality$cadmium,logical=TRUE)
47 Find_outlier=which(OutlierCad==TRUE,arr.ind=TRUE)
48 waterQuality= waterQuality[-Find_outlier,]
49 OutlierChlo=outlier(waterQuality$chloramine,logical=TRUE)
50 Find_outlier=which(OutlierChlo==TRUE,arr.ind=TRUE)
51 waterQuality= waterQuality[-Find_outlier,]
52 OutlierChrom=outlier(waterQuality$chromium,logical=TRUE)

52 OutlierChrom=outlier(waterQuality$chromium,logical=TRUE)
53 Find_outlier=which(OutlierChrom==TRUE,arr.ind=TRUE)
54 waterQuality= waterQuality[-Find_outlier,]
55 OutlierCo=outlier(waterQuality$copper,logical=TRUE)
56 Find_outlier=which(OutlierCo==TRUE,arr.ind=TRUE)
57 waterQuality= waterQuality[-Find_outlier,]
58 OutlierFl=outlier(waterQuality$flouride,logical=TRUE)
59 Find_outlier=which(OutlierFl==TRUE,arr.ind=TRUE)
60 waterQuality= waterQuality[-Find_outlier,]
61 OutlierBa=outlier(waterQuality$bacteria,logical=TRUE)
62 Find_outlier=which(OutlierBa==TRUE,arr.ind=TRUE)
63 waterQuality= waterQuality[-Find_outlier,]
64 OutlierVi=outlier(waterQuality$viruses,logical=TRUE)
65 Find_outlier=which(OutlierVi==TRUE,arr.ind=TRUE)
66 waterQuality= waterQuality[-Find_outlier,]
67 OutlierLe=outlier(waterQuality$lead,logical=TRUE)
68 Find_outlier=which(OutlierLe==TRUE,arr.ind=TRUE)
69 waterQuality= waterQuality[-Find_outlier,]
70 OutlierNi1=outlier(waterQuality$nitrates,logical=TRUE)
71 Find_outlier=which(OutlierNi1==TRUE,arr.ind=TRUE)|

71 Find_outlier=which(OutlierNi1==TRUE,arr.ind=TRUE)
72 waterQuality= waterQuality[-Find_outlier,]
73 OutlierNi2=outlier(waterQuality$nitrites,logical=TRUE)
74 Find_outlier=which(OutlierNi2==TRUE,arr.ind=TRUE)
75 waterQuality= waterQuality[-Find_outlier,]
76 OutlierMe=outlier(waterQuality$mercury,logical=TRUE)
77 Find_outlier=which(OutlierMe==TRUE,arr.ind=TRUE)
78 waterQuality= waterQuality[-Find_outlier,]
79 OutlierPe=outlier(waterQuality$perchlorate,logical=TRUE)
80 Find_outlier=which(OutlierPe==TRUE,arr.ind=TRUE)
81 waterQuality= waterQuality[-Find_outlier,]
82 OutlierRa=outlier(waterQuality$radium,logical=TRUE)
83 Find_outlier=which(OutlierRa==TRUE,arr.ind=TRUE)
84 waterQuality= waterQuality[-Find_outlier,]
85 OutlierSe=outlier(waterQuality$selenium,logical=TRUE)
86 Find_outlier=which(OutlierSe==TRUE,arr.ind=TRUE)
87 waterQuality= waterQuality[-Find_outlier,]
88 OutlierSi=outlier(waterQuality$silver,logical=TRUE)
89 Find_outlier=which(OutlierSi==TRUE,arr.ind=TRUE)
90 waterQuality= waterQuality[-Find_outlier,]
```

5 Data Mining Technique

For our dataset, we will use both classification and clustering.

We applied classification technique(supervised) because the class label (is_safe) is provided in our dataset which indicates whether water is safe or not by predicting the amount of elements(barium,ammonia..etc) for each liter of water.

We will divide our dataset to training and testing data by applying decision tree. We will use the training data set to construct the classification model, and the test data set to determine the accuracy of the classification model so we can predict the new data class labels accurately.

In order to apply the clustering technique(unsupervised), the class label (is_safe) must be removed from our dataset.

We used the K-means technique, which that each cluster is represented by the center of the cluster.

K-means assign each object to the cluster with the nearest center point based on euclidean distance.

The used packages for classification: party – caret

The used packages for clustering: factoextra – cluster – NbClust

The used methods for classification: set.seed() – nrow() – sample() – ctree() – table() – predict() – print()
– plot()

confusionMatrix()

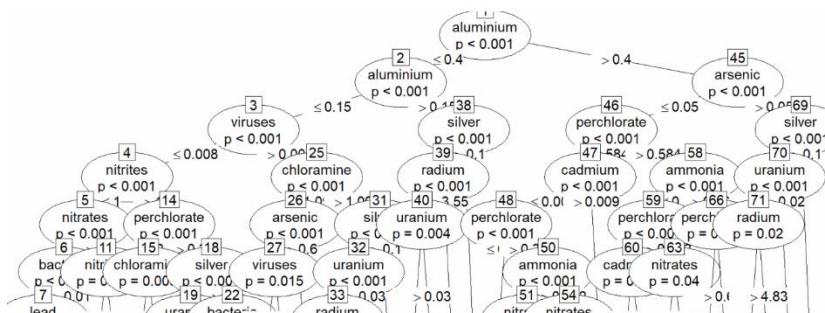
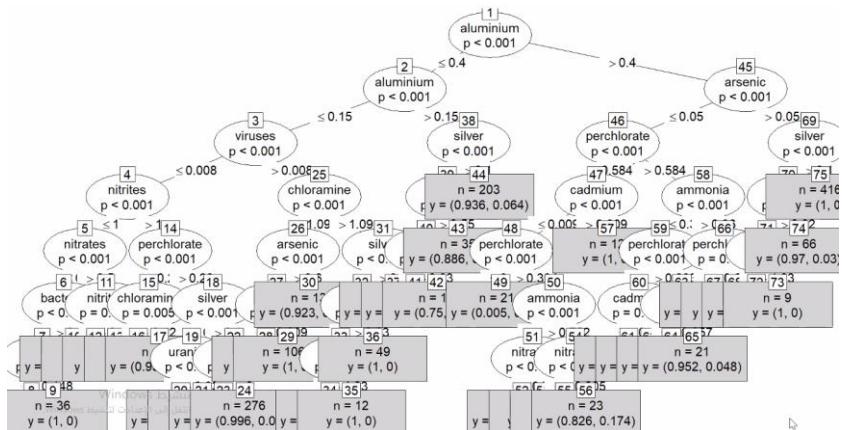
The used methods for clustering: set.seed() – subset() – scale() – kmeans() – fviz_cluster() – silhouette()
– fviz_sillhouette() – fviz_NbClust() – lab() - dist() .

6 Evaluation and Comparison

Classification:

| Mining task | Comparison Criteria | | |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|------------------------------|
| Classification | <ul style="list-style-type: none"> Decision Tree #1: 70% training 30% testing Decision Tree #2: 50% training 50% testing Decision Tree #3: 80% training 20% testing | | |
| | #1: 70% training 30% testing | #2: 50% training 50% testing | #3: 80% training 20% testing |
| Accuracy | 95.50791% | 96.25799% | 96% |
| precision | 96.02% | 96.77% | 96.35% |
| sensitivity | 99.02% | 99.07% | 99.23% |
| specificity | 69.26% | 74.74% | 71.61% |
| Preferred partition? | ✗ | ✓ | ✗ |

Decision Tree #1: 70%-30%



Confusion Matrix and Statistics

| Reference | | |
|------------|------|-----|
| Prediction | No | Yes |
| No | 1711 | 71 |
| Yes | 17 | 160 |

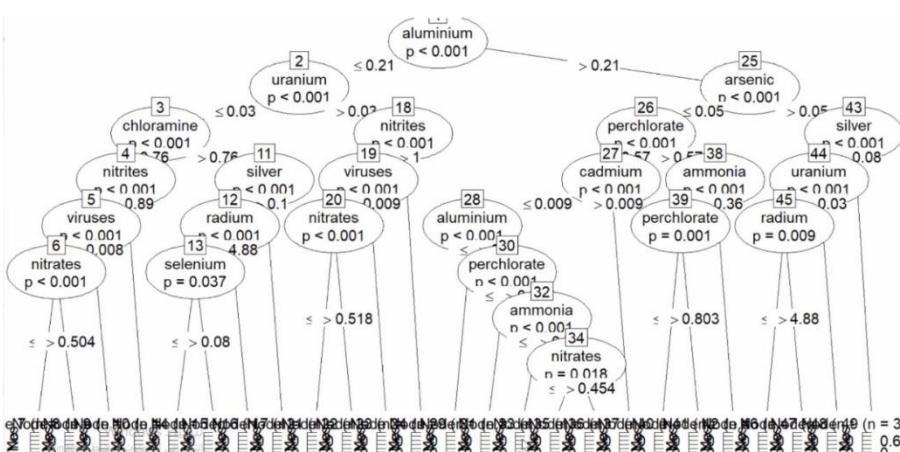
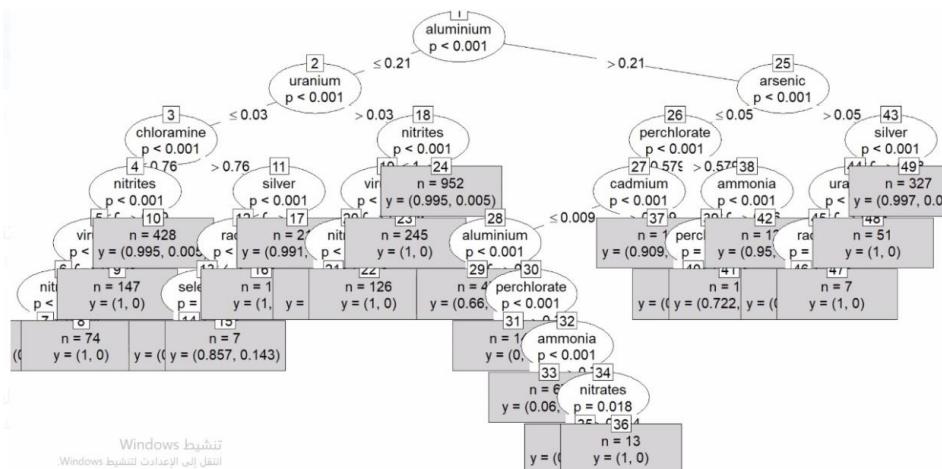
Accuracy : 0.9551
 95% CI : (0.9449, 0.9638)
 No Information Rate : 0.8821
 P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.7597

McNemar's Test P-Value : 1.606e-08

Sensitivity : 0.9902
 Specificity : 0.6926
 Pos Pred Value : 0.9602
 Neg Pred Value : 0.9040
 Prevalence : 0.8821
 Detection Rate : 0.8734
 Detection Prevalence : 0.9096
 Balanced Accuracy : 0.8414
 انتقل إلى الأعداد لتنشيط Windows
 'Positive' Class : No

Decision Tree #2: 50%-50%



Confusion Matrix and Statistics

| | Reference | | I |
|------------|-----------|-----|---|
| Prediction | No | Yes | |
| No | 2880 | 96 | |
| Yes | 27 | 284 | |

Accuracy : 0.9626
 95% CI : (0.9555, 0.9688)

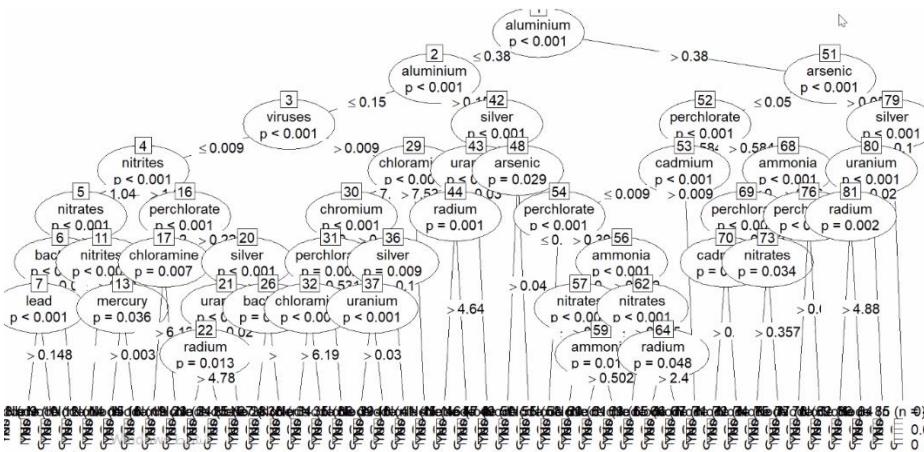
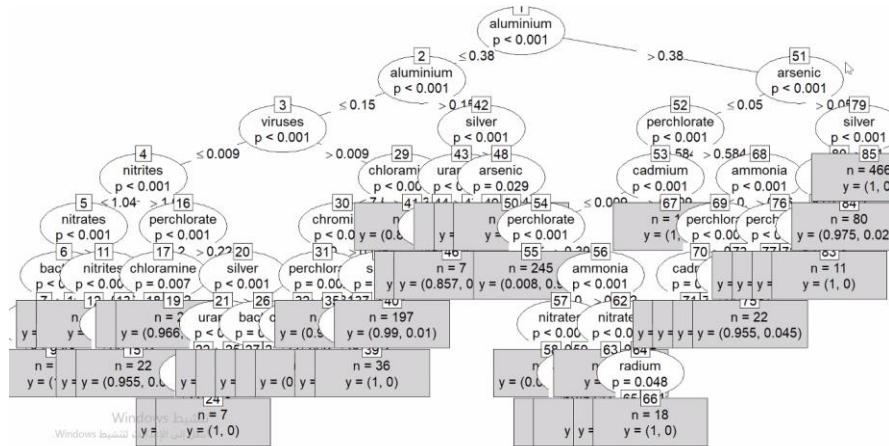
No Information Rate : 0.8844
 P-value [Acc > NIR] : < 2.2e-16

Kappa : 0.8013

Mcnemar's Test P-Value : 8.713e-10

Sensitivity : 0.9907
 Specificity : 0.7474
 Pos Pred Value : 0.9677
 Neg Pred Value : 0.9132
 Prevalence : 0.8844
 Detection Rate : 0.8762
 Detection Prevalence : 0.9054
 Balanced Windows占比: 0.8690
 Windows انتقل إلى الإعدادات لتنشيط
 'Positive' Class : No

Decision Tree #3: 80%-20%



Confusion Matrix and statistics

| Reference | | |
|------------|------|-----|
| Prediction | No | Yes |
| No | 1161 | 44 |
| Yes | 9 | 111 |



Accuracy : 0.96
95% CI : (0.948, 0.9699)

No Information Rate : 0.883
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.7854

McNemar's Test P-Value : 3.008e-06

Sensitivity : 0.9923
Specificity : 0.7161
Pos Pred Value : 0.9635
Neg Pred Value : 0.9250
Prevalence : 0.8830
Detection Rate : 0.8762
Detection Prevalence : 0.9094
Balanced Accuracy : 0.8542
Windows انتقل إلى الإعدادات لتنشيط
'Positive' Class : No

Windows انتقل إلى الإعدادات لتنشيط

Clustering:

Now we will apply clustering to our dataset after removing the class label attribute

```
##Removing the class label for clustering technique
waterQuality<- subset(waterQuality, select = -is_safe )
```

Dataset after removing class label(is_safe)

Class label was here

| | aluminum | ammonia | arsenic | barium | cadmium | chloramine | chromium | copper | fluoride | bacteria | viruses | lead | nitrates | nitrates | mercury | perchlorate | radium | selenium | silver | uranium |
|----|----------|-------------|---------|--------|---------|------------|----------|--------|----------|----------|---------|-------|--------------|----------|---------|-------------|--------|----------|--------|---------|
| 1 | 1.65 | 0.305917753 | 0.040 | 2.85 | 0.007 | 0.35 | 0.83 | 0.17 | 0.05 | 0.20 | 0.000 | 0.054 | 0.811206461 | 1.13 | 0.007 | 0.63032226 | 6.78 | 0.08 | 0.34 | 0.02 |
| 2 | 2.32 | 0.709796055 | 0.010 | 3.31 | 0.002 | 5.28 | 0.68 | 0.66 | 0.90 | 0.69 | 0.650 | 0.100 | 0.100959112 | 1.93 | 0.003 | 0.536865420 | 3.21 | 0.06 | 0.27 | 0.05 |
| 3 | 1.01 | 0.471079904 | 0.040 | 0.58 | 0.008 | 4.24 | 0.53 | 0.02 | 0.99 | 0.05 | 0.003 | 0.076 | 0.714265714 | 1.11 | 0.006 | 0.83953916 | 7.07 | 0.07 | 0.44 | 0.01 |
| 4 | 1.36 | 0.381143430 | 0.040 | 2.96 | 0.001 | 7.23 | 0.03 | 1.66 | 1.08 | 0.71 | 0.710 | 0.016 | 0.070671378 | 1.29 | 0.004 | 0.15227918 | 1.72 | 0.02 | 0.45 | 0.05 |
| 5 | 0.92 | 0.815790675 | 0.030 | 0.20 | 0.006 | 2.67 | 0.69 | 0.57 | 0.61 | 0.13 | 0.001 | 0.117 | 0.339727410 | 1.11 | 0.003 | 0.26218400 | 2.41 | 0.02 | 0.06 | 0.02 |
| 6 | 0.94 | 0.486123042 | 0.030 | 2.68 | 0.003 | 0.80 | 0.43 | 1.38 | 0.11 | 0.67 | 0.670 | 0.135 | 0.491670873 | 1.89 | 0.006 | 0.45366505 | 5.42 | 0.08 | 0.19 | 0.02 |
| 8 | 3.93 | 0.666666667 | 0.040 | 0.66 | 0.001 | 6.22 | 0.10 | 1.86 | 0.86 | 0.16 | 0.005 | 0.197 | 0.688541141 | 1.81 | 0.001 | 0.89079980 | 7.24 | 0.08 | 0.08 | 0.07 |
| 9 | 0.60 | 0.824193064 | 0.010 | 0.71 | 0.005 | 3.14 | 0.77 | 1.45 | 0.98 | 0.35 | 0.002 | 0.167 | 0.739525492 | 1.84 | 0.004 | 0.39121723 | 4.99 | 0.08 | 0.25 | 0.08 |
| 10 | 0.22 | 0.562680864 | 0.020 | 1.37 | 0.007 | 6.40 | 0.49 | 0.82 | 1.24 | 0.83 | 0.830 | 0.109 | 0.241292277 | 1.46 | 0.010 | 0.50795121 | 0.08 | 0.03 | 0.31 | 0.01 |
| 11 | 3.27 | 0.122701438 | 0.001 | 2.69 | 0.005 | 5.75 | 0.15 | 0.60 | 1.29 | 0.04 | 0.008 | 0.145 | 0.427057042 | 1.25 | 0.006 | 0.92502922 | 7.80 | 0.05 | 0.33 | 0.06 |
| 12 | 1.35 | 0.756542962 | 0.040 | 0.64 | 0.002 | 0.10 | 0.76 | 0.17 | 0.58 | 0.52 | 0.520 | 0.011 | 0.923319031 | 1.49 | 0.009 | 0.55952543 | 1.30 | 0.08 | 0.48 | 0.05 |
| 13 | 1.88 | 0.646272150 | 0.020 | 2.78 | 0.008 | 0.05 | 0.42 | 1.00 | 0.09 | 0.91 | 0.910 | 0.103 | 0.220090863 | 1.95 | 0.006 | 0.35934380 | 1.97 | 0.03 | 0.06 | 0.05 |
| 14 | 4.93 | 0.804078903 | 0.040 | 3.05 | 0.008 | 0.70 | 0.51 | 1.35 | 1.07 | 0.70 | 0.700 | 0.101 | 0.058051489 | 1.11 | 0.008 | 0.44748706 | 5.58 | 0.09 | 0.38 | 0.05 |
| 16 | 0.61 | 0.082915413 | 0.030 | 0.59 | 0.002 | 1.94 | 0.77 | 1.54 | 0.62 | 0.23 | 0.001 | 0.017 | 0.099949520 | 1.08 | 0.007 | 0.18634163 | 0.98 | 0.01 | 0.47 | 0.05 |
| 17 | 3.47 | 0.531929121 | 0.020 | 0.06 | 0.001 | 5.29 | 0.47 | 1.08 | 1.43 | 0.89 | 0.890 | 0.080 | 0.095911156 | 1.20 | 0.008 | 0.03030551 | 6.89 | 0.06 | 0.12 | 0.08 |
| 19 | 4.88 | 0.930342461 | 0.020 | 0.36 | 0.001 | 1.21 | 0.68 | 0.71 | 0.99 | 0.75 | 0.750 | 0.071 | 0.015143867 | 1.22 | 0.002 | 0.94673568 | 1.00 | 0.00 | 0.41 | 0.05 |
| 21 | 0.68 | 0.637245069 | 0.001 | 0.04 | 0.006 | 4.57 | 0.20 | 1.18 | 1.00 | 0.92 | 0.920 | 0.068 | 0.0477031802 | 1.41 | 0.007 | 0.35383370 | 3.05 | 0.03 | 0.13 | 0.08 |
| 22 | 1.15 | 0.273821446 | 0.020 | 0.97 | 0.007 | 3.47 | 0.65 | 1.51 | 1.46 | 0.58 | 0.580 | 0.061 | 0.451792024 | 1.50 | 0.004 | 0.24378026 | 1.74 | 0.03 | 0.01 | 0.06 |
| 23 | 0.27 | 0.359077232 | 0.020 | 0.55 | 0.001 | 3.74 | 0.12 | 1.77 | 0.43 | 0.80 | 0.800 | 0.114 | 0.640080767 | 1.18 | 0.008 | 0.578039372 | 0.90 | 0.02 | 0.16 | 0.06 |
| 24 | 4.32 | 0.692410565 | 0.030 | 2.60 | 0.008 | 7.24 | 0.61 | 1.23 | 1.44 | 0.56 | 0.560 | 0.012 | 0.475012620 | 1.74 | 0.004 | 0.060494239 | 3.22 | 0.07 | 0.18 | 0.08 |
| 26 | 3.31 | 0.740220662 | 0.030 | 0.46 | 0.001 | 7.22 | 0.73 | 1.05 | 1.00 | 0.25 | 0.007 | 0.109 | 0.096415952 | 1.07 | 0.001 | 0.65787277 | 0.49 | 0.04 | 0.47 | 0.05 |

| | | | |
|------------|-----------------------------------|----------------------------|----------------------------|
| Clustering | - K=2 | | |
| | Silhouette width for each cluster | K=2 | K=3 |
| | | cluster size ave.sil.width | cluster size ave.sil.width |
| | 1 | 1 3168 0.07 | 1 1072 0.11 |
| | 2 | 2 3456 0.26 | 2 2951 0.05 |
| | Silhouette width for all clusters | 0.17 | 0.09 |
| | Visualization | Figure 1 | Figure 2 |
| | Preferred partition? | ✓ | ✗ |
| | | | ✗ |

Figure 1

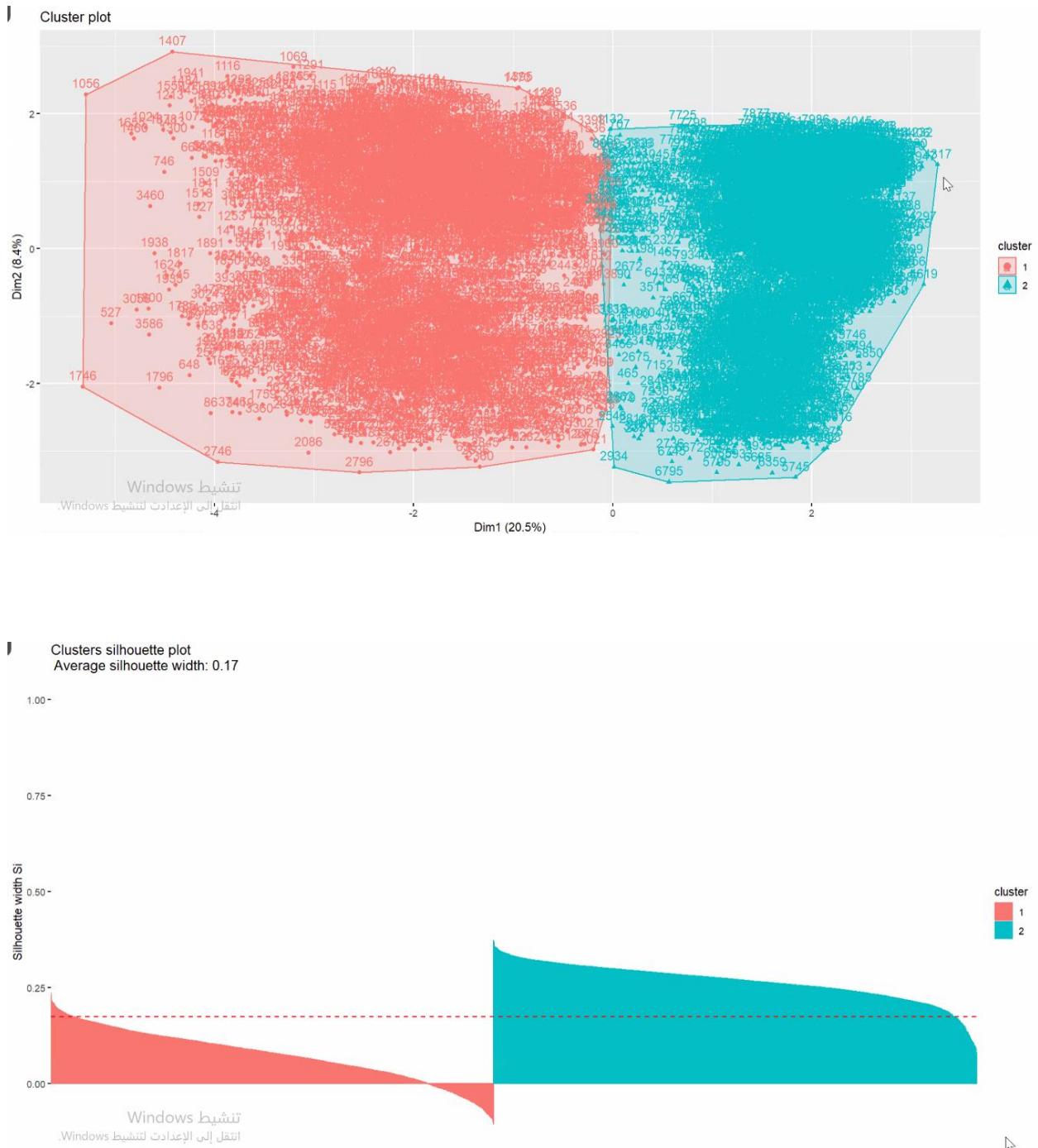


Figure 2

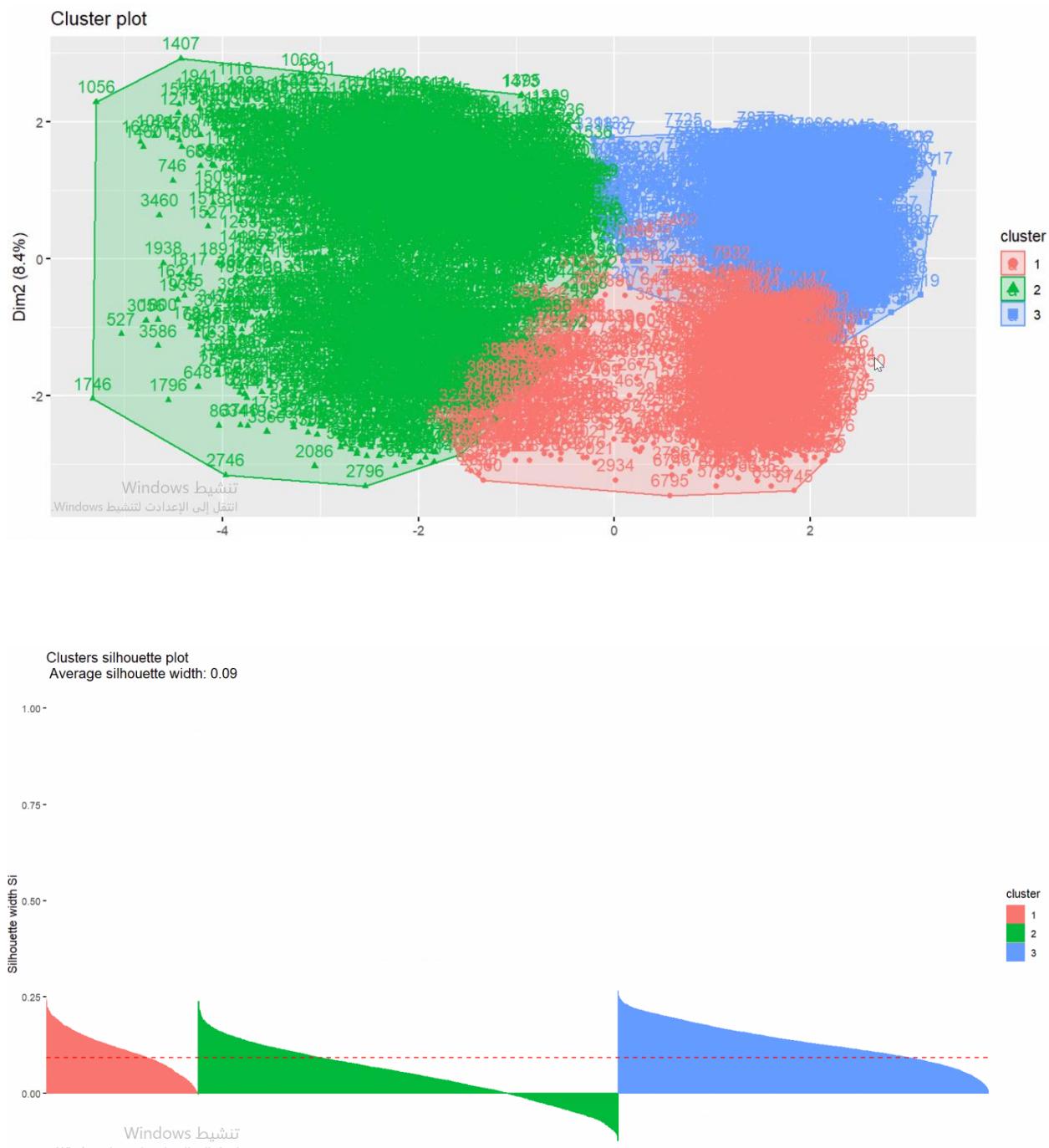
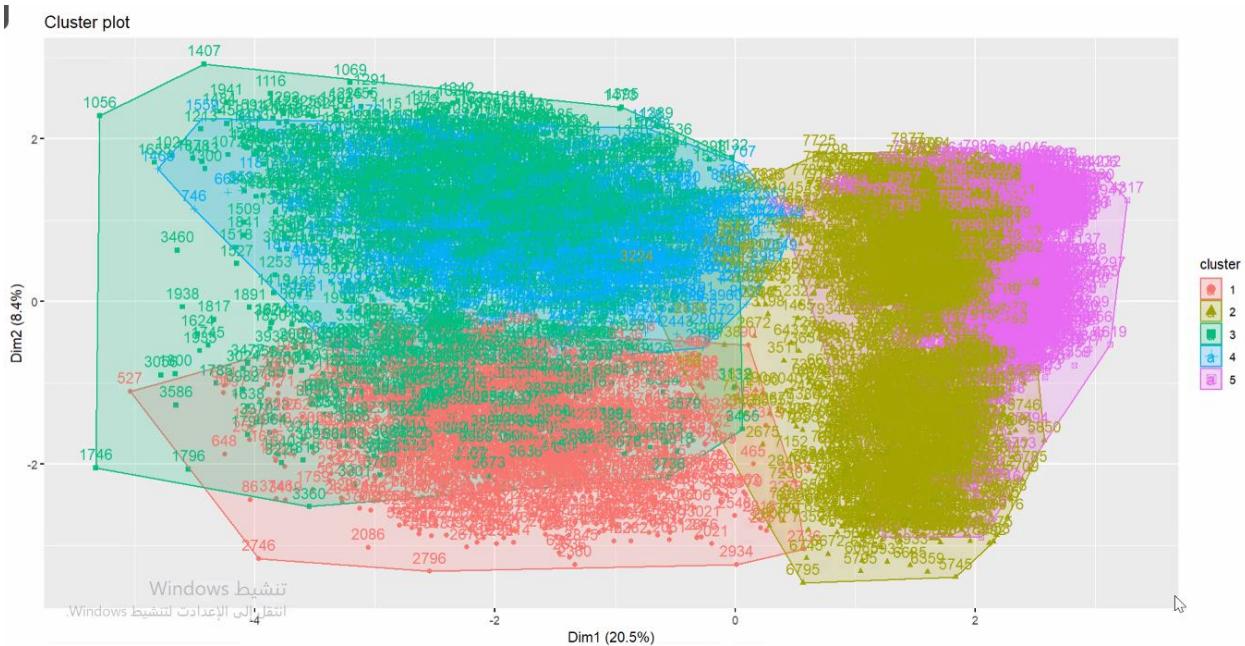
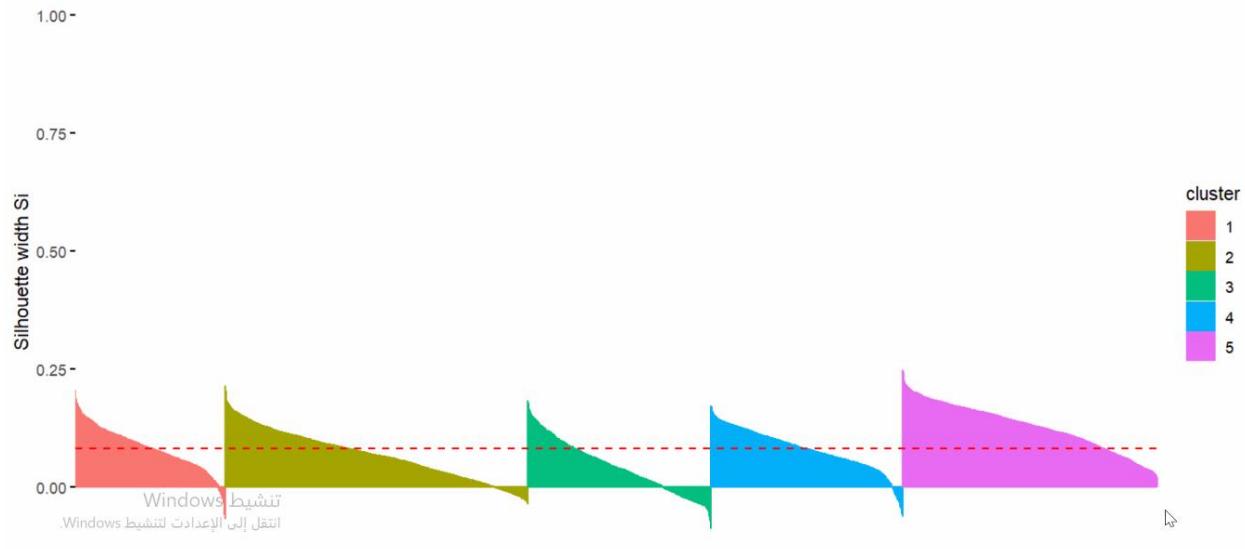
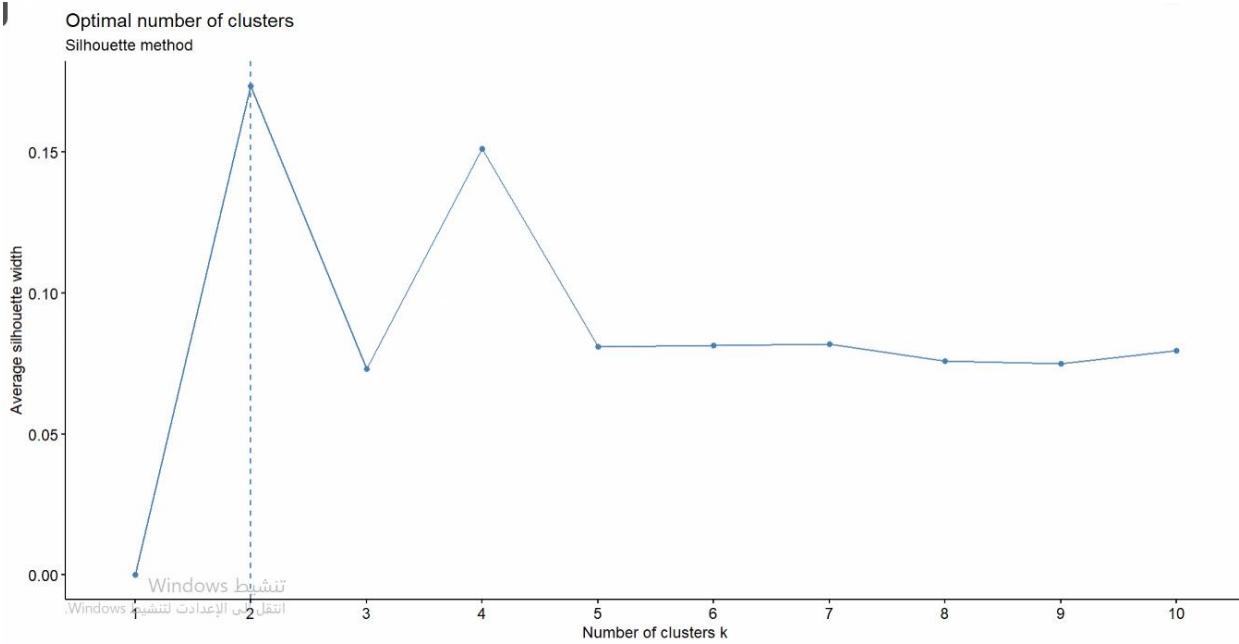


Figure 3



Clusters silhouette plot
Average silhouette width: 0.08





7 Findings

After studying the water quality dataset, from determining what each attribute do and how the attribute will effect each other, we apply some preprocessing methods such as cleaning, transformation to prepare our dataset for data mining process.

For classification we studied different cases by dividing our dataset by using Ctree method and then we came up with these results:

-70% training 30% testing , Accuracy=95.50791%

-50% training 50% testing , Accuracy=96.25799%

-80% training 20% testing , Accuracy=96%

We noticed that almost all accuracies are the same, but these results lead to determine the best model for classification technique which is (50%,50%) because it has the highest accuracy which means the class label (is_safe) is affected by all attributes.

(ammonia , barium , arsenic...etc) , that means the evaluation model we considered to be the best classify most of the tuples that are covered by the rule and it correctly classified by class label (is_safe), also this

model lead to determine the correct class_label for each object faster than the others . after analyzing the decision trees we noticed that (80%,20%) and (70%,30%) first splitting point was aluminum and the second level of the tree include aluminum and we think that what makes both two cases accuracies lower than (50%,50%), in (50%,50%) case the second level exchanged the aluminum with uranium.

We noticed that some of our dataset attributes has no affect in the decision tree and has not appeared as a decision node to determine the leaf node (decision) , and these attribute are barium , chromium , copper , fluoride , bacteria , lead and mercury .

As a result, we think analyzing and studying the decision tree is interesting for individuals because they can use this tree to determine that the water they use and drink is safe or not, and for companies to end up selling water that is good for people and the environment. We can also extract some rules from this tree such as:

if the aluminum > 0.21 and the arsenic > 0.05 and the silver < 0.08 and the uranium > 0.03 then n=51 and y=(1,0)

For clustering we studied different cases by changing the number of clusters k and using k-means method and then we came up with these results:

-k=2 , Silhouette width for all clusters: 0.17

-k=3 , Silhouette width for all clusters: 0.09

-k=5 , Silhouette width for all clusters:0.08

These results lead to determine the best model for clustering technique is k=2 because it has the highest Silhouette width (0.17) for all clusters, (0.07) for the first cluster and (0.26) for the second cluster, since the value is approaching 1 ,that means the objects within a cluster are closer to each other than to the objects in the other cluster.

After analyzing the plots, we configure what supports the quality of k=2, that the clusters are not overlapping in figure 1 above, unlike k=3 and k=5, the clusters are overlapping pointedly, in k=3, we can

see that cluster 1,3 and cluster 1,2 are intertwined with each other in figure 2 above, as well as in k=5, we can see for example that cluster 2,5 and cluster 3,4 and other clusters are highly overlapped in figure 3 above, which can make k=5 the worst case in choosing number of clusters.

To capitalize, what makes k=3 and k=5 not in consideration is that because of the overlapping that result in inability when observing where each object belong to the right cluster, but k=2 the Silhouette width is 0.17 which is optimal because k-means method consider the number approach to 1 is better than the others, and what increased our confidence with k=2 is that it is not overlapping, to enhance our analyzing we validated which number of clusters is the best using fviz_nbclust() method , that came up with 2 clusters.

8 Code

```
1 #Data minning project
2 dataset=read.csv('/Users/admin/Desktop/WATER QUALITY/waterQuality.txt')
3
4 nrow(waterQuality)
5 ncol(waterQuality)
6 sum(is.na(waterQuality))
7 summary(waterQuality)
8 waterQuality$ammonia<-as.numeric(waterQuality$ammonia)
9
10 boxplot(waterQuality$saluminium , data=waterQuality)
11 boxplot(waterQuality$ammonia , data=waterQuality)
12 boxplot(waterQuality$arsenic , data=waterQuality)
13 boxplot(waterQuality$cadmium , data=waterQuality)
14 boxplot(waterQuality$chloramine,data=waterQuality)
15 boxplot(waterQuality$chromium,data=waterQuality)
16 boxplot(waterQuality$copper,data=waterQuality)
17 boxplot(waterQuality$flouride,data=waterQuality)
18 boxplot(waterQuality$bacteria,data=waterQuality)
19 hist(waterQuality$barium)
20 pie(table(waterQuality$is_safe))
21 plot(waterQuality$lead,waterQuality$ammonia , main="Scatterplot", xlab="Lead", ylab="Ammonia")
22
23 is.na(waterQuality$ammonia)
24 waterQuality$ammonia<-ifelse(is.na(waterQuality$ammonia),ave(waterQuality$ammonia,FUN=function(x) mean(x,na.rm=TRUE))
25
26 #install.packages("outliers")
27 library(outliers)
28 OutlierUran=outlier(waterQuality$uranium,logical=TRUE)
29 sum(OutlierUran)
30 Find_outlier=which(OutlierUran==TRUE,arr.ind=TRUE)
31 waterQuality= waterQuality[-Find_outlier,]
32
33 OutlierAl=outlier(waterQuality$saluminium,logical=TRUE)
34 sum(OutlierAl)
35 Find_outlier=which(OutlierAl==TRUE,arr.ind=TRUE)
36 waterQuality= waterQuality[-Find_outlier,]
37
38 OutlierArs=outlier(waterQuality$arsenic,logical=TRUE)
39 sum(OutlierArs)
40 Find_outlier=which(OutlierArs==TRUE,arr.ind=TRUE)
41 waterQuality= waterQuality[-Find_outlier,]
42
43 OutlierBar=outlier(waterQuality$barium,logical=TRUE)
44 sum(OutlierBar)
45 Find_outlier=which(OutlierBar==TRUE,arr.ind=TRUE)
46 waterQuality= waterQuality[-Find_outlier,]
47
48 OutlierCad=outlier(waterQuality$cadmium,logical=TRUE)
49 sum(OutlierCad)
50 Find_outlier=which(OutlierCad==TRUE,arr.ind=TRUE)
51 waterQuality= waterQuality[-Find_outlier,]
52
53 OutlierChlo=outlier(waterQuality$chloramine,logical=TRUE)
54 sum(OutlierChlo)
55 Find_outlier=which(OutlierChlo==TRUE,arr.ind=TRUE)
56 waterQuality= waterQuality[-Find_outlier,]
57
58 OutlierChrom=outlier(waterQuality$chromium,logical=TRUE)
59 sum(OutlierChrom)
60 Find_outlier=which(OutlierChrom==TRUE,arr.ind=TRUE)
61 waterQuality= waterQuality[-Find_outlier,]
62
63 OutlierCo=outlier(waterQuality$copper,logical=TRUE)
64 sum(OutlierCo)
65 Find_outlier=which(OutlierCo==TRUE,arr.ind=TRUE)
66 waterQuality= waterQuality[-Find_outlier,]
67
68 OutlierFl=outlier(waterQuality$flouride,logical=TRUE)
69 sum(OutlierFl)
70 Find_outlier=which(OutlierFl==TRUE,arr.ind=TRUE)
71 waterQuality= waterQuality[-Find_outlier,]
72
73 OutlierBa=outlier(waterQuality$bacteria,logical=TRUE)
74 sum(OutlierBa)
75 Find_outlier=which(OutlierBa==TRUE,arr.ind=TRUE)
76 waterQuality= waterQuality[-Find_outlier,]
77
78 OutlierVi=outlier(waterQuality$viruses,logical=TRUE)
79 sum(OutlierVi)
80 Find_outlier=which(OutlierVi==TRUE,arr.ind=TRUE)
81 waterQuality= waterQuality[-Find_outlier,]
82
83 OutlierLe=outlier(waterQuality$lead,logical=TRUE)
84 sum(OutlierLe)
85 Find_outlier=which(OutlierLe==TRUE,arr.ind=TRUE)
86 waterQuality= waterQuality[-Find_outlier,]
87
88 OutlierNi1=outlier(waterQuality$nitrates,logical=TRUE)
89 sum(OutlierNi1)
90 Find_outlier=which(OutlierNi1==TRUE,arr.ind=TRUE)
91 waterQuality= waterQuality[-Find_outlier,]
92
93 OutlierNi2=outlier(waterQuality$nitrites,logical=TRUE)
94 sum(OutlierNi2)
95 Find_outlier=which(OutlierNi2==TRUE,arr.ind=TRUE)
96 waterQuality= waterQuality[-Find_outlier,]
97
98 OutlierMe=outlier(waterQuality$mercury,logical=TRUE)
99 sum(OutlierMe)
100 Find_outlier=which(OutlierMe==TRUE,arr.ind=TRUE)
101 waterQuality= waterQuality[-Find_outlier,]
102
103 OutlierPe=outlier(waterQuality$perchlorate,logical=TRUE)
104 sum(OutlierPe)
```

```

103 OutlierPe=outlier(waterQuality$perchlorate,logical=TRUE)
104 sum(OutlierPe)
105 Find_outlier=which(OutlierPe==TRUE,arr.ind=TRUE)
106 waterQuality= waterQuality[-Find_outlier,]
107
108 OutlierRa=outlier(waterQuality$radium,logical=TRUE)
109 sum(OutlierRa)
110 Find_outlier=which(OutlierRa==TRUE,arr.ind=TRUE)
111 waterQuality= waterQuality[-Find_outlier,]
112
113 OutlierSe=outlier(waterQuality$selenium,logical=TRUE)
114 sum(OutlierSe)
115 Find_outlier=which(OutlierSe==TRUE,arr.ind=TRUE)
116 waterQuality= waterQuality[-Find_outlier,]
117
118 Outliersi=outlier(waterQuality$silver,logical=TRUE)
119 sum(Outliersi)
120 Find_outlier=which(Outliersi==TRUE,arr.ind=TRUE)
121 waterQuality= waterQuality[-Find_outlier,]
122
123 #Encoding:
124 waterQuality$is_safe = factor(waterQuality$is_safe,levels = c("0","1"), labels = c("No","Yes"))
125
126 ##Normalize ammonia
127 - normalize<- function(x){
128   return((x-min(x)) / (max(x)-min(x)))
129 }
130 waterQuality$ammonia<-normalize(waterQuality$ammonia)
131 ##Normalize perchlorate
132 - normalize<- function(x){
133   return((x-min(x)) / (max(x)-min(x)))
134 }
135 waterQuality$perchlorate<-normalize(waterQuality$perchlorate)
136 ##Normalize nitrates
137 - normalize<- function(x){
138   return((x-min(x)) / (max(x)-min(x)))
139 }
140 waterQuality$nitrates<-normalize(waterQuality$nitrates)
141
142 View(waterQuality)
143
144 waterQuality <- na.omit(waterQuality)
145 pie(table(waterQuality$is_safe))
146
147 ##Classification/30,70
148 set.seed(1234)
149 firstP <- sample(2, nrow(waterQuality), replace=TRUE, prob=c(0.7, 0.3))
150 trainData <- waterQuality[firstP==1,]
151 testData <- waterQuality[firstP==2,]
152 install.packages('party')
153 library(party)
154 myFormula <- is_safe ~ aluminium + ammonia + arsenic + barium + cadmium + chloramine+ chromium + copper + flouride +
155 waterQuality_ctree <- ctree(myFormula, data=trainData)
156 table(predict(waterQuality_ctree), trainData$is_safe)
157 print(waterQuality_ctree)
158 plot(waterQuality_ctree,type="simple")
159 plot(waterQuality_ctree)
160
161 testPred <- predict(waterQuality_ctree, newdata = testData)
162
163 #Evaluate the model
164 #Create the confusion matrix
165 table(testPred, testData$is_safe)
166
167 install.packages('caret')
168 library(caret)
169 results <- confusionMatrix(testPred, testData$is_safe)
170 acc <- results$overall["Accuracy"]*100
171 acc
172 results
173
174 ##Classification/50,50
175 set.seed(1234)
176 firstP <- sample(2, nrow(waterQuality), replace=TRUE, prob=c(0.5, 0.5))
177 trainData <- waterQuality[firstP==1,]
178 testData <- waterQuality[firstP==2,]
179
180 myFormula <- is_safe ~ aluminium + ammonia + arsenic + barium + cadmium + chloramine+ chromium + copper + flouride +
181 waterQuality_ctree <- ctree(myFormula, data=trainData)
182 table(predict(waterQuality_ctree), trainData$is_safe)
183 print(waterQuality_ctree)
184 plot(waterQuality_ctree,type="simple")
185 plot(waterQuality_ctree)
186
187 testPred <- predict(waterQuality_ctree, newdata = testData)
188
189 #Evaluate the model
190 #Create the confusion matrix
191 table(testPred, testData$is_safe)
192
193 results <- confusionMatrix(testPred, testData$is_safe)
194 acc <- results$overall["Accuracy"]*100
195 acc
196 results
197
198 ##Classification/80,20
199
200
201
202 ##Classification/80,20

```

```

202 ##Classification/80,20
203 set.seed(1234)
204 firstP <- sample(2, nrow(waterQuality), replace=TRUE, prob=c(0.8, 0.2))
205 trainData <- waterQuality[firstP==1,]
206 testData <- waterQuality[firstP==2,]
207
208 myFormula <- is_safe ~ aluminium + ammonia + arsenic + barium + cadmium + chloramine+ chromium + copper + flouride +
209
210 waterQuality_ctree <- ctree(myFormula, data=trainData)
211 table(predict(waterQuality_ctree), trainData$is_safe)
212 print(waterQuality_ctree)
213 plot(waterQuality_ctree,type="simple")
214 plot(waterQuality_ctree)
215
216 testPred <- predict(waterQuality_ctree, newdata = testData)
217
218 #Evaluate the model
219 #Create the confusion matrix
220 table(testPred, testData$is_safe)
221
222 results <- confusionMatrix(testPred, testData$is_safe)
223 acc <- results$overall["Accuracy"]*100
224 acc
225 results
226
227 ##Removing the class label for clustering technique
228 waterQuality<- subset( waterQuality, select = -is_safe )
229
230 # k-means clustering
231 set.seed(8953)
232
233 waterQuality <- scale(waterQuality)
234 #First clustering K=2:
235 kmeans.result1 <- kmeans(waterQuality, 2)
236 kmeans.result1
237
238 install.packages("factoextra")
239 library(factoextra)
240 fviz_cluster(kmeans.result1, data = waterQuality)
241
242 #####Cluster Validation
243 install.packages("cluster")
244 library(cluster)
245 #average for each cluster
246 avg_sil <- silhouette(kmeans.result1$cluster,dist(waterQuality))
247 fviz_silhouette(avg_sil)
248
249 ##########
250 #Second clustering K=3:
251 kmeans.result2 <- kmeans(waterQuality,3)
252 kmeans.result2
253
254 fviz_cluster(kmeans.result2, data = waterQuality)
255 #####Cluster validation
256
257 #average for each cluster
258 avg_sil <- silhouette(kmeans.result2$cluster,dist(waterQuality))
259 fviz_silhouette(avg_sil)
260
261 ##########
262 #Third clustering K=5:
263 kmeans.result3 <- kmeans(waterQuality,5)
264 kmeans.result3
265
266 fviz_cluster(kmeans.result3, data = waterQuality)
267
268 #####Cluster Validation
269 #average for each cluster
270 avg_sil <- silhouette(kmeans.result3$cluster,dist(waterQuality))
271 fviz_silhouette(avg_sil)
272
273 install.packages("NbClust")
274 library(NbClust)
275 fviz_nbclust(waterQuality, kmeans, method = "silhouette")+labs(subtitle = "Silhouette method")
276
277
278
279
280

```

9 References

<https://www.kaggle.com/mssmartypants/water-quality>

10 Tasks Distribution

| ID | Name | Responsibilities |
|--------------------|---------------------|------------------|
| Jumanah aldawsari | All project parts . | |
| Lama alshaya | All project parts . | |
| Nouf alsadhan | All project parts . | |
| Aljawharah alzamil | All project parts . | |