Data wrangling assignment 2

Harold Choi 3866530

10/8/2020

Required packages

```
library(readr)
library(xlsx)
library(dplyr)
library(openxlsx)
library(readxl)
library(tidyr)
library(stringr)
library(outliers)
library(MVN)
library(forecast)
library(moments)
```

Executive Summary

- Are criminal offences in an area influenced by the alcohol purchased and sold? This investigation examines the alcohol wholesale data in Victoria and the number of offences committed by individuals in the year 2018. The data used is provided by Victorian government agency and the two datasets will be joined to examine how much alcohol is purchased in each Local Government Area and how many offence is committed in that area .
- In addition, data will be tidied and manipulated so that the dataset is understandable and clean. Outliers and missing values will be explained and investigated and subsequently removed.

Data

- In 2014, the Liquor Control Reform Act commissioned the collection of wholesale liquor transactions in Victoria. The dataset named 'alcohol' contains 7 variables and 1360 observations from the year 2017 to 2018. The data collected can be broken down into:

1. Local Government Area (LGA) of retailer for the sales of liquor
2. The type of liquor sold
3. The volume of liquor that was sold

- The dataset used in this task was generated from: https://discover.data.vic.gov.au/dataset/victorian-wholesale-liquor-sales

The data description includes:

- Year: Year of data collected, 2017-2018.
- Local Government Area (LGA): Local government area is the naming convention for the area of which the alcohol was sold. A classification for each LGA is through either one of the symbols (C), (S), (RC) or (B) indicating City, Shrine, Rural city or Boroughs respectively. For the purpose of this investigation, this symbol is unimportant and will be removed.

- Liquor Type: The type of liquor sold according to the quantity in liters and content of alcohol.
- Liquor Group: Group of liquor sold consisting of beer, wine, spirits and cider.
- Liquor Volume (L): Volume in liters of liquor sold.
- Ratio of alcohol: The ratio of alcohol in the liquor. 1 representing 100% alcohol content and 0 representing 0% alcohol.
- Alcohol volume (L): The volume of alcohol sold in liters, calculated the multiplication of Liquor Volume (L) and Ratio of alcohol.
- The Crime Statistics Agency (CSA) collects and produces crime data from the Victorian Police. CSA then conducts a series of quality checks and processes the data to be released to the public. In this investigation, an offence is defined as any criminal act of omission by a person or organisation for which a penalty could by imposed by the Victorian legal system. The dataset named ‘offences’ contains 6 variables and 870 observation from 2011 to 2020. The data collected can be broken down into:

1. Local Government Area (LGA) of the offence that has taken place
2. Police region
3. Offence count

The dataset used in this task was generated from: https://www.crimestatistics.vic.gov.au/crime-statistics/latest-victorian-crime-data/download-data

The data description includes:

- Year: Year of the data collected.
- Year ending: The month in which the year was ended in, in this case June for all observation.
- Police region: The region of police operations which includes North West Metro, Eastern, Southern Metro and Western. Furthermore the number in front of each Police region will be removed.
- Local Government Area: As mentioned above, LGA is the used to classify the location. In this case the classification of (C),(S),(RC),(B) is not present.
- Offence count: The number of offence committed and reported.

- Rate per 100,000 population: The number of offence committed and reported based on a 100,000 population. Calculated by the taking the offence count mutiply by 100,000 and dividing by the total population of that LGA.

```r
alcohol <- read_excel("2017-18-wholesale-liquor-sales-data-by-LGA.XLSX", sheet = "Victorian Wholesale Li
offences <- read_excel("Data_Tables_LGA_Recorded_Offences_Year_Ending_June_2020.xlsx", sheet = "Table 0

head(alcohol)
```

```
## # A tibble: 6 x 7
##   Year  ‘Local Governme~ ‘Liquor Type‘ ‘Liquor Group‘ ‘Liquor Volume ~
##   <chr> <chr>            <chr>         <chr>                    <dbl>
## 1 2017~ Alpine (S)       Beer Low <=4~ Beer                    32796.
## 2 2017~ Alpine (S)       Beer Low >48~ Beer                      2722.
## 3 2017~ Alpine (S)       Beer Medium ~ Beer                    276899.
## 4 2017~ Alpine (S)       Beer Medium ~ Beer                     24226.
## 5 2017~ Alpine (S)       Beer Heavy <~ Beer                   1000682.
## 6 2017~ Alpine (S)       Beer Heavy >~ Beer                    198786.
## # ... with 2 more variables: ‘Ratio of alcohol‘ <dbl>, ‘Alcohol Volume
## #   (L)‘ <dbl>
```

```
head(offences)
```

```
## # A tibble: 6 x 6
##    Year ‘Year ending‘ ‘Police Region‘ ‘Local Governme~ ‘Offence Count‘
##   <dbl> <chr>         <chr>           <chr>                      <dbl>
## 1  2020 June          1 North West M~ Banyule                     9700
## 2  2020 June          1 North West M~ Brimbank                   20242
## 3  2020 June          1 North West M~ Darebin                    15161
## 4  2020 June          1 North West M~ Hobsons Bay                 6100
## 5  2020 June          1 North West M~ Hume                       21529
## 6  2020 June          1 North West M~ Maribyrnong                 9388
## # ... with 1 more variable: ‘Rate per 100,000 population‘ <dbl>
```

```
str(alcohol)
```

```
## tibble [1,360 x 7] (S3: tbl_df/tbl/data.frame)
##  $ Year                       : chr [1:1360] "2017-18" "2017-18" "2017-18" "2017-18" ...
##  $ Local Government Area (LGA): chr [1:1360] "Alpine (S)" "Alpine (S)" "Alpine (S)" "Alpine (S)" ...
##  $ Liquor Type                : chr [1:1360] "Beer Low <=48 Ltrs" "Beer Low >48 Ltrs" "Beer Medium <=
##  $ Liquor Group               : chr [1:1360] "Beer" "Beer" "Beer" "Beer" ...
##  $ Liquor Volume (L)          : num [1:1360] 32796 2722 276899 24226 1000682 ...
##  $ Ratio of alcohol           : num [1:1360] 0.0269 0.0269 0.0348 0.0348 0.0476 0.0476 0.123 0.123 0
##  $ Alcohol Volume (L)         : num [1:1360] 882.2 73.2 9636.1 843 47632.5 ...
```

```
str(offences)
```

```
## tibble [870 x 6] (S3: tbl_df/tbl/data.frame)
##  $ Year                       : num [1:870] 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...
##  $ Year ending                : chr [1:870] "June" "June" "June" "June" ...
##  $ Police Region              : chr [1:870] "1 North West Metro" "1 North West Metro" "1 North West
##  $ Local Government Area      : chr [1:870] "Banyule" "Brimbank" "Darebin" "Hobsons Bay" ...
##  $ Offence Count              : num [1:870] 9700 20242 15161 6100 21529 ...
##  $ Rate per 100,000 population: num [1:870] 7328 9630 9144 6187 8918 ...
```

Tidy and manipulate

- As mentioned in the data description, for the alcohol dataset the Local Government Area classification of (C), (S), (RC) or (B) will be removed using the gsub function.
- The Local Government Area has been renamed to match with the offence variable name as this variable will be the unique identifier for joining the alcohol and offence dataset together and thus have to be same.

```r
alcohol$`Local Government Area (LGA)`<- gsub("\\s*\\([^\\)]+\\)","", alcohol$`Local Government Area (LG

alcohol <- alcohol %>% rename( `Local Government Area`=`Local Government Area (LGA)`)
```

- The Total alcohol sold in liters column is added using the group by and mutate function to sum the alcohol volume sold in liters based on the LGA.

```r
alcohol <- alcohol %>% group_by(`Local Government Area`) %>%
            mutate(`Total alcohol sold (L)` = sum(`Alcohol Volume (L)`))
```

- However the alcohol dataset is untidy as there are multiple observation but the same Total alcohol due to the different Liquor Type, Liquor Group, Liquor Volume, Ratio of alcohol, Alcohol Volume (L).
- Using the summarise function the alcohol dataset is grouped according to Local Government Area and the Total alcohol sold (L) is added using the summarise function which would allow us to compare the Total alcohol consumed (L) accordingly.

```r
head(alcohol)
```

```
## # A tibble: 6 x 8
## # Groups:   Local Government Area [1]
##   Year  `Local Governme~ `Liquor Type` `Liquor Group` `Liquor Volume ~
##   <chr> <chr>            <chr>         <chr>                     <dbl>
## 1 2017~ Alpine           Beer Low <=4~ Beer                      32796.
## 2 2017~ Alpine           Beer Low >48~ Beer                       2722.
## 3 2017~ Alpine           Beer Medium ~ Beer                     276899.
## 4 2017~ Alpine           Beer Medium ~ Beer                      24226.
## 5 2017~ Alpine           Beer Heavy <~ Beer                    1000682.
## 6 2017~ Alpine           Beer Heavy >~ Beer                     198786.
## # ... with 3 more variables: `Ratio of alcohol` <dbl>, `Alcohol Volume
## #   (L)` <dbl>, `Total alcohol sold (L)` <dbl>
```

```r
alcohol_a <- alcohol %>% group_by(`Local Government Area`) %>%
            summarise(`Total alcohol sold (L)` = sum(`Alcohol Volume (L)`))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
head(alcohol_a)
```

```
## # A tibble: 6 x 2
##   `Local Government Area` `Total alcohol sold (L)`
##   <chr>                                      <dbl>
## 1 Alpine                                   163637.
## 2 Ararat                                    76142.
## 3 Ballarat                                 768553.
## 4 Banyule                                  373338.
## 5 Bass Coast                               405641.
## 6 Baw Baw                                  267551.
```

- Since the offence dataset contains data from 2011 to 2020, the filter function is used to subset the offence dataset for only observations in the year 2018 so that this dataset can be used subsequently to compare with the alcohol sold in the year 2018.

```
offences2018 <- offences %>% filter(Year == 2018)
head(offences2018)
```

```
## # A tibble: 6 x 6
##    Year ‘Year ending‘ ‘Police Region‘ ‘Local Governme~ ‘Offence Count‘
##   <dbl> <chr>         <chr>           <chr>                       <dbl>
## 1  2018 June          1 North West M~ Banyule                      9688
## 2  2018 June          1 North West M~ Brimbank                    18425
## 3  2018 June          1 North West M~ Darebin                     14677
## 4  2018 June          1 North West M~ Hobsons Bay                  6028
## 5  2018 June          1 North West M~ Hume                        20217
## 6  2018 June          1 North West M~ Maribyrnong                  8297
## # ... with 1 more variable: ‘Rate per 100,000 population‘ <dbl>
```

Merge Data

- The two datasets are now prepared and can be merged with a left join that retains all the data in the offence_2018 dataset and matches the alcohol_a dataset using the Local Government Area variable.

```
alcohol_offence <- left_join(offences2018, alcohol_a, by="Local Government Area")

alcohol_offence$‘Total alcohol sold (L)‘ <- round(alcohol_offence$‘Total alcohol sold (L)‘, 0)

head(alcohol_offence)
```

```
## # A tibble: 6 x 7
##    Year ‘Year ending‘ ‘Police Region‘ ‘Local Governme~ ‘Offence Count‘
##   <dbl> <chr>         <chr>           <chr>                       <dbl>
## 1  2018 June          1 North West M~ Banyule                      9688
## 2  2018 June          1 North West M~ Brimbank                    18425
## 3  2018 June          1 North West M~ Darebin                     14677
## 4  2018 June          1 North West M~ Hobsons Bay                  6028
## 5  2018 June          1 North West M~ Hume                        20217
## 6  2018 June          1 North West M~ Maribyrnong                  8297
## # ... with 2 more variables: ‘Rate per 100,000 population‘ <dbl>, ‘Total
## #   alcohol sold (L)‘ <dbl>
```

```
str(alcohol_offence)
```

```
## tibble [87 x 7] (S3: tbl_df/tbl/data.frame)
##  $ Year                       : num [1:87] 2018 2018 2018 2018 2018 ...
##  $ Year ending                : chr [1:87] "June" "June" "June" "June" ...
##  $ Police Region              : chr [1:87] "1 North West Metro" "1 North West Metro" "1 North West Me
##  $ Local Government Area      : chr [1:87] "Banyule" "Brimbank" "Darebin" "Hobsons Bay" ...
##  $ Offence Count              : num [1:87] 9688 18425 14677 6028 20217 ...
##  $ Rate per 100,000 population: num [1:87] 7438 8827 9079 6248 9008 ...
##  $ Total alcohol sold (L)     : num [1:87] 373338 652607 781075 895441 1036497 ...
```

Tidy and Manipulate and Understand

- In the new dataset, alcohol_offence, Rate per 100,000 population have been renamed to Offence per 100,000 population.

- Using the mutate function the Total population variable is created using mutate function by dividing the Offence count by the Offence per 100,000 population and multiplying by 100,000.
- In addition, the Average alcohol sold per person in liters is added into the dataset using mutate, calculated by the Total alcohol sold divided by the Total population.
- The Average alcohol sold per person in Liters is then rounded off to 2 decimal points.

```
alcohol_offence <- alcohol_offence %>% rename(‘Offence per 100,000 population‘ = ‘Rate per 100,000 popul
alcohol_offence <- alcohol_offence %>% mutate(‘Total Population‘ = (‘Offence Count‘/‘Offence per 100,00
alcohol_offence <- alcohol_offence %>% mutate(‘Average sold per person in Liters‘ = ‘Total alcohol sold
alcohol_offence$‘Average sold per person in Liters‘ <- round(alcohol_offence$‘Average sold per person in
```

```
head(alcohol_offence$‘Offence per 100,000 population‘)
```

```
## [1] 7438.004 8826.601 9079.324 6247.862 9008.435 9076.390
```

```
head(alcohol_offence$‘Total Population‘)
```

```
## [1] 130250 208744 161653  96481 224423  91413
```

```
head(alcohol_offence$‘Average sold per person in Liters‘)
```

```
## [1] 2.87 3.13 4.83 9.28 4.62 5.60
```

- The alcohol_offence dataset contains different data type conversions mainly characters and numerics. The Year ending and Local Government Area are a character type while the rest of the variables are numeric type.
- The Police region of North West Metro, Eastern, Southern Metro, Western is converted from a character to a factor.

```
alcohol_offence$‘Police Region‘ <- alcohol_offence$‘Police Region‘ %>% factor(levels=c("1 North West Me
                                                                              labels=c("North West Metro",

str(alcohol_offence)
```

```
## tibble [87 x 9] (S3: tbl_df/tbl/data.frame)
##  $ Year                             : num [1:87] 2018 2018 2018 2018 2018 ...
##  $ Year ending                      : chr [1:87] "June" "June" "June" "June" ...
##  $ Police Region                    : Factor w/ 4 levels "North West Metro",..: 1 1 1 1 1 1 1 1 1 1
##  $ Local Government Area            : chr [1:87] "Banyule" "Brimbank" "Darebin" "Hobsons Bay" ...
##  $ Offence Count                    : num [1:87] 9688 18425 14677 6028 20217 ...
##  $ Offence per 100,000 population   : num [1:87] 7438 8827 9079 6248 9008 ...
##  $ Total alcohol sold (L)           : num [1:87] 373338 652607 781075 895441 1036497 ...
##  $ Total Population                 : num [1:87] 130250 208744 161653 96481 224423 ...
##  $ Average sold per person in Liters: num [1:87] 2.87 3.13 4.83 9.28 4.62 ...
```

```
head(alcohol_offence)
```

```
## # A tibble: 6 x 9
##    Year ‘Year ending‘ ‘Police Region‘ ‘Local Governme~ ‘Offence Count‘
##   <dbl> <chr>         <fct>           <chr>                          <dbl>
## 1  2018 June          North West Met~ Banyule                         9688
## 2  2018 June          North West Met~ Brimbank                       18425
## 3  2018 June          North West Met~ Darebin                        14677
## 4  2018 June          North West Met~ Hobsons Bay                     6028
## 5  2018 June          North West Met~ Hume                           20217
## 6  2018 June          North West Met~ Maribyrnong                     8297
## # ... with 4 more variables: ‘Offence per 100,000 population‘ <dbl>, ‘Total
## #   alcohol sold (L)‘ <dbl>, ‘Total Population‘ <dbl>, ‘Average sold per person
## #   in Liters‘ <dbl>
```

Scan Missing values

- Using sapply and is.na function the location of missing values, NAs in each column is checked. There are missing values in the Police Region, Offence per 100,000 population, Total alcohol sold, Total population, Average sold per person in liters.
- There are missing values in the data due to the Totals that are for each of the 4 Police region that was introduced when the data was merged. In addition, there are 2 observations for both Justice Institution and Immigration Facilities police region and Unincorporate Victoria police region and their respectively totals which has values that are very small and will not be used in this investigation.

```r
sapply(alcohol_offence, function (x) which(is.na(x)))
```

```
## $Year
## integer(0)
##
## $‘Year ending‘
## integer(0)
##
## $‘Police Region‘
## [1] 84 85 86 87
##
## $‘Local Government Area‘
## integer(0)
##
## $‘Offence Count‘
## integer(0)
##
## $‘Offence per 100,000 population‘
## [1] 84 85 86 87
##
## $‘Total alcohol sold (L)‘
## [1] 15 41 52 83 84 85 87
##
## $‘Total Population‘
## [1] 84 85 86 87
##
## $‘Average sold per person in Liters‘
## [1] 15 41 52 83 84 85 86 87
```

- The sum of missing values in each column is displayed below using the colSums function.

```r
colSums(is.na(alcohol_offence))
```

```
##                              Year                       Year ending 
##                                 0                                 0 
##                     Police Region             Local Government Area 
##                                 4                                 0 
##                     Offence Count    Offence per 100,000 population 
##                                 0                                 4 
##            Total alcohol sold (L)                  Total Population 
##                                 7                                 4 
## Average sold per person in Liters 
##                                 8
```

- To scan the dataset for infinite values and "not a number" values is.infinite and is.nan is used respectively.
- The output below suggests there are no infinite values of Nan values in each column of the dataset.

```r
head(sapply(alcohol_offence, function (x) is.infinite(x)))
```

```
##       Year Year ending Police Region Local Government Area Offence Count
## [1,] FALSE       FALSE         FALSE                 FALSE         FALSE
## [2,] FALSE       FALSE         FALSE                 FALSE         FALSE
## [3,] FALSE       FALSE         FALSE                 FALSE         FALSE
## [4,] FALSE       FALSE         FALSE                 FALSE         FALSE
## [5,] FALSE       FALSE         FALSE                 FALSE         FALSE
## [6,] FALSE       FALSE         FALSE                 FALSE         FALSE
##      Offence per 100,000 population Total alcohol sold (L) Total Population
## [1,]                          FALSE                  FALSE            FALSE
## [2,]                          FALSE                  FALSE            FALSE
## [3,]                          FALSE                  FALSE            FALSE
## [4,]                          FALSE                  FALSE            FALSE
## [5,]                          FALSE                  FALSE            FALSE
## [6,]                          FALSE                  FALSE            FALSE
##      Average sold per person in Liters
## [1,]                             FALSE
## [2,]                             FALSE
## [3,]                             FALSE
## [4,]                             FALSE
## [5,]                             FALSE
## [6,]                             FALSE
```

```r
head(sapply(alcohol_offence, function (x) is.nan(x)))
```

```
##       Year Year ending Police Region Local Government Area Offence Count
## [1,] FALSE       FALSE         FALSE                 FALSE         FALSE
## [2,] FALSE       FALSE         FALSE                 FALSE         FALSE
## [3,] FALSE       FALSE         FALSE                 FALSE         FALSE
## [4,] FALSE       FALSE         FALSE                 FALSE         FALSE
## [5,] FALSE       FALSE         FALSE                 FALSE         FALSE
## [6,] FALSE       FALSE         FALSE                 FALSE         FALSE
##      Offence per 100,000 population Total alcohol sold (L) Total Population
## [1,]                          FALSE                  FALSE            FALSE
## [2,]                          FALSE                  FALSE            FALSE
## [3,]                          FALSE                  FALSE            FALSE
## [4,]                          FALSE                  FALSE            FALSE
## [5,]                          FALSE                  FALSE            FALSE
## [6,]                          FALSE                  FALSE            FALSE
##      Average sold per person in Liters
## [1,]                             FALSE
## [2,]                             FALSE
## [3,]                             FALSE
## [4,]                             FALSE
## [5,]                             FALSE
## [6,]                             FALSE
```

- To omit the rows that have an NA, the complete.cases function is used. A total of 8 rows are removed from the dataset of which 6 are total values and 2 for Justice Institution and Immigration Facilities and Unincorporate Victoria.

```
alcohol_offence <- alcohol_offence[complete.cases(alcohol_offence),]
```

Scanning numeric variables for outliers

- The capping method is used for the numeric variables to replace the outlier with the nearest count that is not an outlier. This method is used as the outliers in the alcohol_offence dataset are not due to data entry errors and capping allows outliers that are above the upper limit to be replace with the nearest value that is at the 95th percentile.
- The cap function is used to cap the values that are above the quantiles of the 95% confidence interval and the values that are below the 0.05% confidence interval. The values outsides the limits would be considered outliers and removed.
- The 5 numeric variables are subsetted below and then using sapply and the cap function, the outliers are removed from these variables and the summary of the alcohol_sub dataset is given which shows the descriptive statistics of the 5 numeric variables after removing outliers.

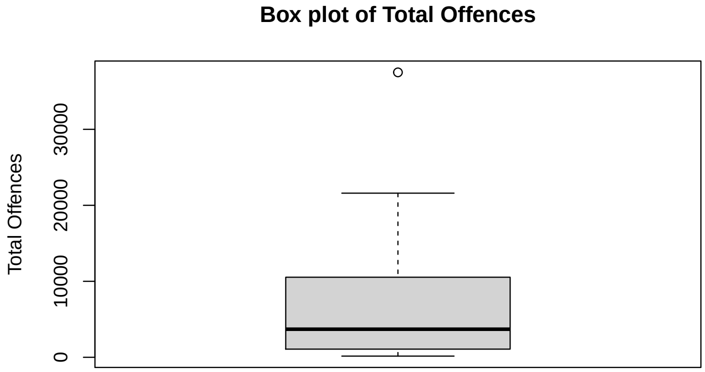
```
cap <- function(x){
    quantiles <- quantile( x, c(.05, 0.25, 0.75, .95 ) )
    x[ x < quantiles[2] - 1.5*IQR(x) ] <- quantiles[1]
    x[ x > quantiles[3] + 1.5*IQR(x) ] <- quantiles[4]
    x
}

alcohol_offence_df <- alcohol_offence %>% select(‘Offence Count‘, ‘Offence per 100,000 population‘, ‘To
alcohol_sub <- as.data.frame(sapply(alcohol_offence_df, FUN =cap))
summary(alcohol_sub)
```

```
##  Offence Count   Offence per 100,000 population Total alcohol sold (L)
##  Min.   :  159   Min.   : 2493                  Min.   :  20970
##  1st Qu.: 1076   1st Qu.: 5293                  1st Qu.: 114540
##  Median : 3698   Median : 6898                  Median : 360043
##  Mean   : 6130   Mean   : 7338                  Mean   : 504264
##  3rd Qu.:10539   3rd Qu.: 9203                  3rd Qu.: 843549
##  Max.   :21602   Max.   :13988                  Max.   :1916536
##  Total Population Average sold per person in Liters
##  Min.   :  2982   Min.   : 2.350
##  1st Qu.: 16314   1st Qu.: 4.940
##  Median : 46816   Median : 6.500
##  Mean   : 80306   Mean   : 7.188
##  3rd Qu.:136049   3rd Qu.: 8.925
##  Max.   :255367   Max.   :14.130
```

- The boxplot function have been used to detect univariate outliers for the following 5 numeric variables: Offence count, Offence per 100,000 population, Total alcohol consumed, Total population and Average consumption per person in liters.
- The z score approach is used to scan and extract all outliers and find the location of the outliers. In addition the outliers are located with the “which” function as shown below.
- Observing the boxplot below, there is an outlier for the total offences as indicated by the dot that out of the range of the maximum whisker of the boxplot.
- The outliers presented in this data is from Melbourne which has a total offence count of 37,497 which is much higher than the average of 6,130 total offences.

```
alcohol_offence$`Offence Count` %>% boxplot(main="Box plot of Total Offences",ylab="Total Offences")
```



```
z.scores <- alcohol_offence$`Offence Count` %>% scores(type="z")
z.scores %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.9128 -0.7781 -0.3927  0.0000  0.6128  4.5748
```

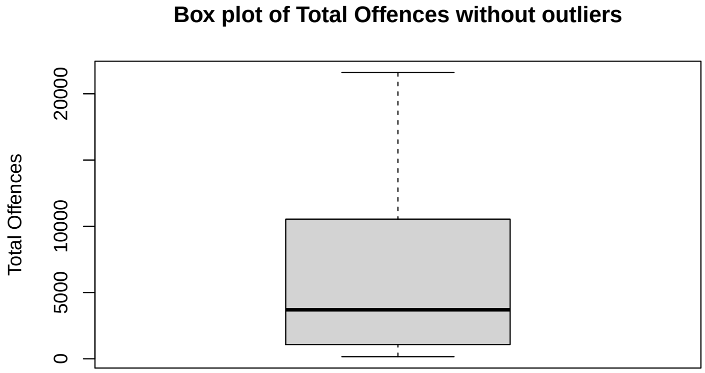
```
which(abs(z.scores)>3)
```

```
## [1] 7
```

```
length(which(abs(z.scores)>3))
```

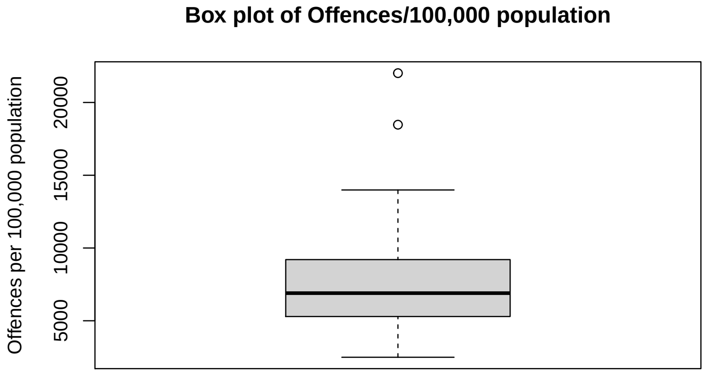
```
## [1] 1
```

```
alcohol_offence_clean1 <- alcohol_offence$`Offence Count` %>% cap()
alcohol_offence_clean1 %>% boxplot(main="Box plot of Total Offences without outliers", ylab="Total Offen
```



- Observing the boxplot below, there are two univariate outliers for the Offence per 100,000 population variable as indicated by the 2 dots that are out of the range of the maximum whisker of the boxplot.
- The outliers presented are from Melbourne and Latrobe which has a offence per 100,000 population of 22,016 and 18,473 respectively much higher than the average offence of 7,338 per 100,000 population.

```
alcohol_offence$`Offence per 100,000 population` %>% boxplot(main="Box plot of Offences/100,000 populati
```



```
z.scores1 <- alcohol_offence$`Offence per 100,000 population` %>% scores(type="z")
z.scores1 %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## -1.4870 -0.6610 -0.1876  0.0000  0.4925  4.2719
```

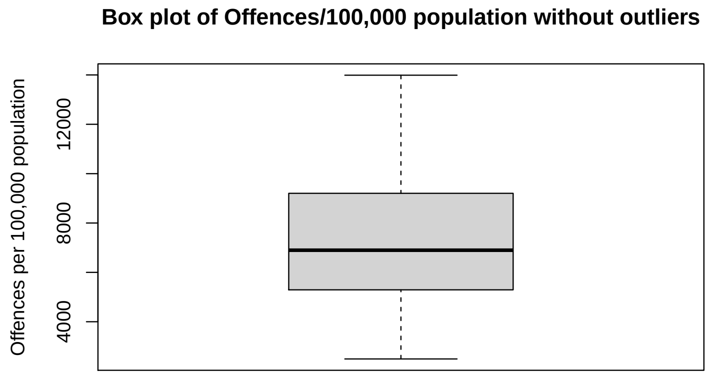
```
which(abs(z.scores1)>3)
```

```
## [1]  7 24
```

```
length(which(abs(z.scores1)>3))
```

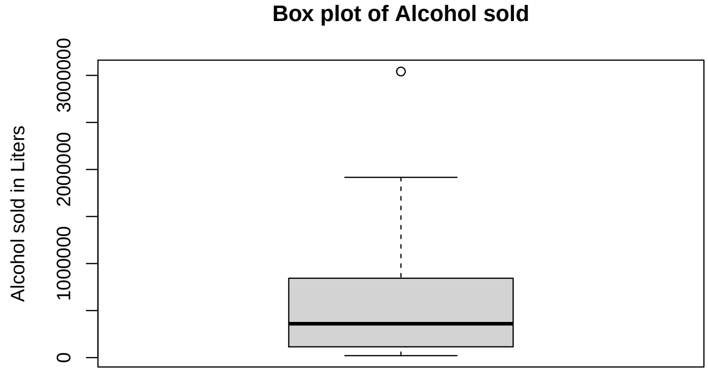
```
## [1] 2
```

```
alcohol_offence_clean2 <- alcohol_offence$‘Offence per 100,000 population‘ %>% cap()
alcohol_offence_clean2 %>% boxplot(main="Box plot of Offences/100,000 population without outliers", yla
```



- Observing the boxplot below, there is an univariate outliers for the Total alcohol sold population variable as indicated by the single dot that is out of the range of the maximum whisker of the boxplot.
- The outliers presented in this data is from Melbourne which has a total alcohol sold of 3,041,394 liters almost 6 times the average of 504,264 liters.

```
alcohol_offence$‘Total alcohol sold (L)‘ %>% boxplot(main="Box plot of Alcohol sold",
                                                       ylab="Alcohol sold in Liters")
```



```
z.scores2 <- alcohol_offence$`Total alcohol sold (L)` %>% scores(type="z")
z.scores2 %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.9744 -0.7935 -0.3190  0.0000  0.6155  4.8634
```

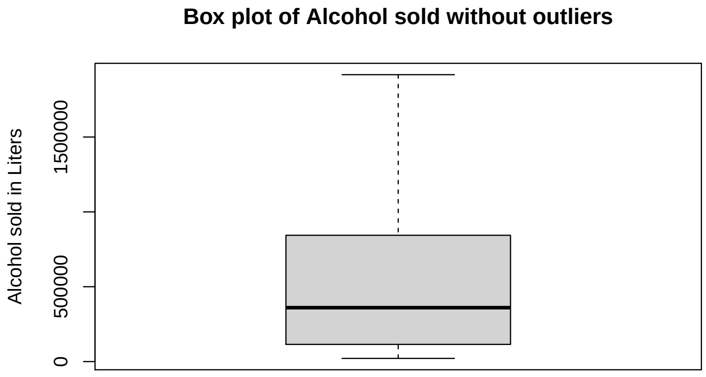
```
which(abs(z.scores2)>3)
```

```
## [1] 7
```

```
length(which(abs(z.scores2)>3))
```

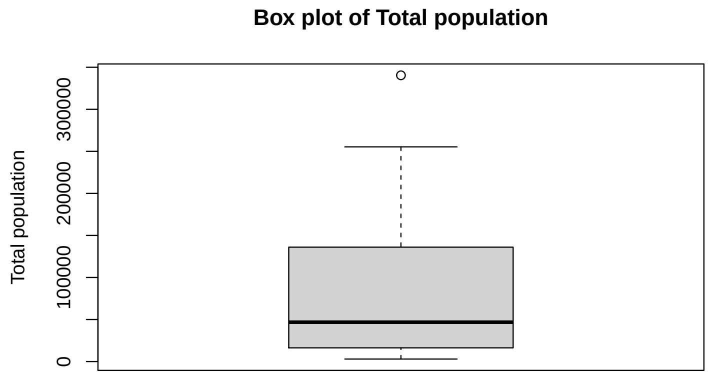
```
## [1] 1
```

```
alcohol_offence_clean3 <- alcohol_offence$`Total alcohol sold (L)` %>% cap()
alcohol_offence_clean3 %>% boxplot(main="Box plot of Alcohol sold without outliers",
                                                              ylab="Alcohol sold in Liters")
```



- Observing the boxplot below, there is an univariate outliers for the Total population variable as indicated by the single dot that is out of the range of the maximum whisker of the boxplot.
- The outliers presented in this data is from Local Government Area, Casey which has a total population of 340,443 greater than 5 times the average of 80,306 for each Local Government Area.

```
alcohol_offence$`Total Population` %>% boxplot(main="Box plot of Total population",
                                                ylab="Total population")
```



```
z.scores3 <- alcohol_offence$‘Total Population‘ %>% scores(type="z")
z.scores3 %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.0229 -0.8498 -0.4539  0.0000  0.7043  3.3574
```

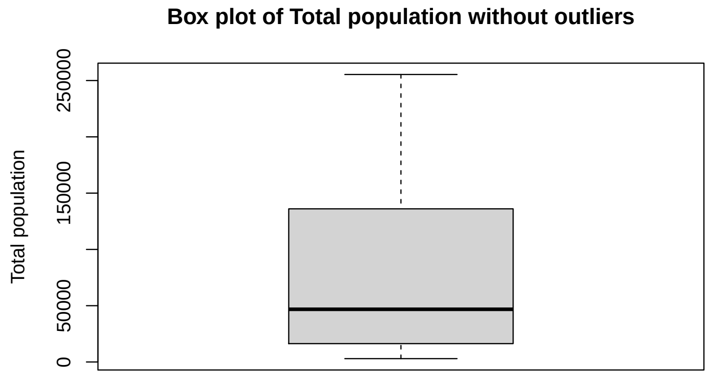
```
which(abs(z.scores3)>3)
```

```
## [1] 42
```

```
length(which(abs(z.scores3)>3))
```

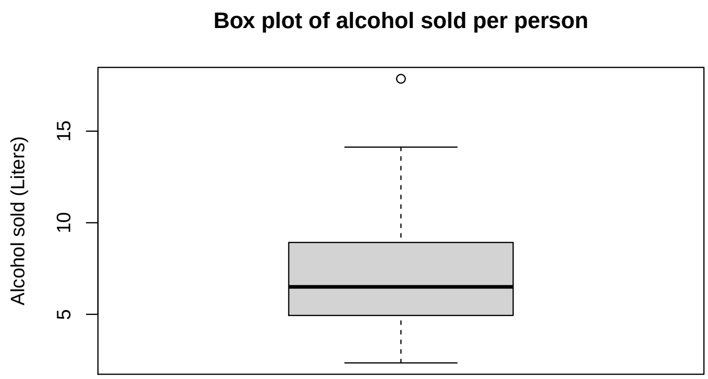
```
## [1] 1
```

```
alcohol_offence_clean4 <- alcohol_offence$‘Total Population‘ %>% cap()
alcohol_offence_clean4 %>%  boxplot(main="Box plot of Total population without outliers",
                                    ylab="Total population")
```



- Observing the boxplot below, there is an univariate outliers for the Average consumption per person in liters variable as indicated by the single dot is above the interquartile range.
- The outliers presented in this data is from Melbourne which has a Average alcohol sold of 17.86 liters almost 2.5 times the average of 7.19 liters.

```
alcohol_offence$‘Average sold per person in Liters‘ %>% boxplot(main="Box plot of alcohol sold per pers
```



```
z.scores4 <- alcohol_offence$‘Average sold per person in Liters‘ %>% scores(type="z")
z.scores4 %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.6347 -0.7710 -0.2507  0.0000  0.5580  3.5377
```

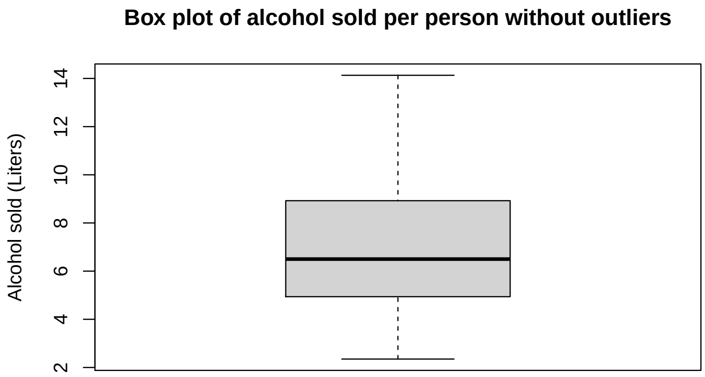
```
which(abs(z.scores4)>3)
```

```
## [1] 7
```

```
length(which(abs(z.scores4)>3))
```

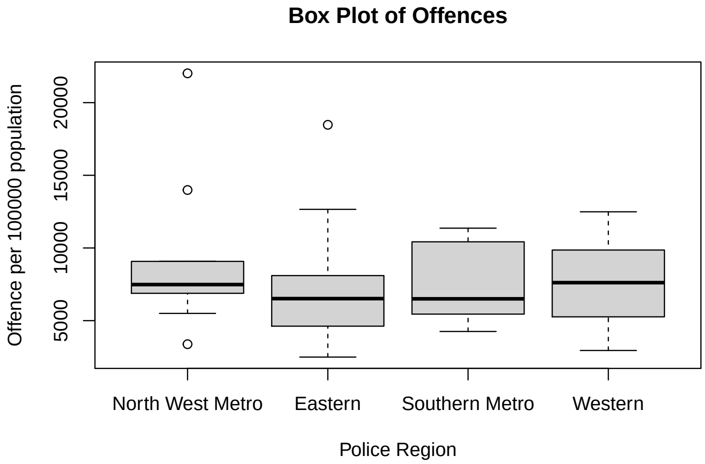
```
## [1] 1
```

```
alcohol_offence_clean5 <- alcohol_offence$‘Average sold per person in Liters‘ %>% cap()
alcohol_offence_clean5 %>% boxplot(main="Box plot of alcohol sold per person without outliers", ylab="A
```



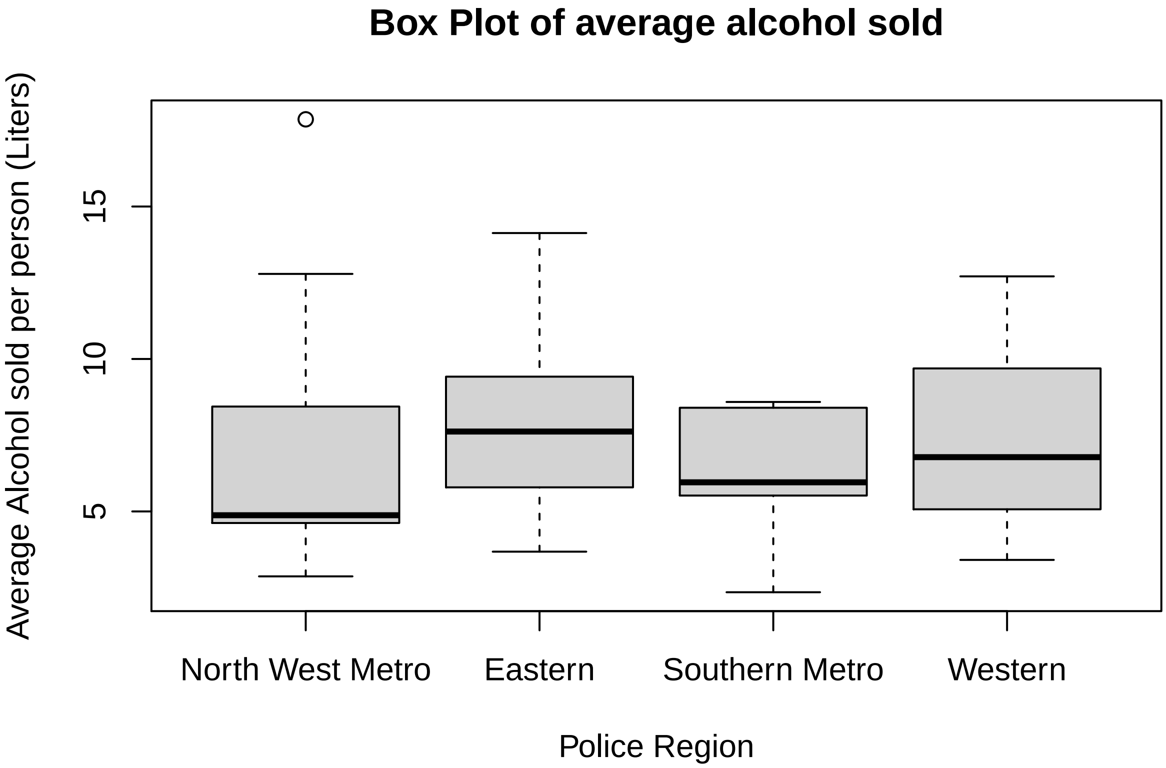
- Using bivariate box plots and scatter plots, the offence per 100,000 population based on the Police region is examined. Observing the boxplot, there are 4 outliers for the offence per 100,000 population, 3 from North West Metro and 1 from Eastern Police region. It is worth pointing out that one of the outlier for the North West Metro is a lower than the minimum whisker of the box plot.
- The Western region appears to have the highest mean offence per 100,000 population. In addition, the interquartile range for Southern Metro appears to be the biggest which indicates a greater variability in the offence committed, while the Eastern police region has the biggest upper whisker which indicates that there is more variability in the offences for the most positive quartile group.

```
# multivariate method
boxplot(alcohol_offence$`Offence per 100,000 population` ~ alcohol_offence$`Police Region`,
        ylab="Offence per 100000 population", xlab= "Police Region",
        main="Box Plot of Offences")
```



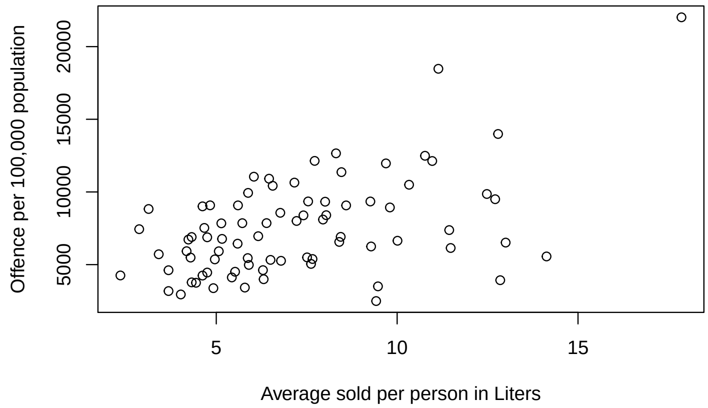
- In terms of average alcohol sold per person, the outlier, Melbourne, in North West Metro tops the chart at 17.86 liters. The highest mean for average alcohol sold per person is the Eastern region while the lowest mean is the North West Metro region. In terms of variability, the Southern Metro appears to have a lower variability in alcohol sold indicated by the smaller interquartile range however there is a greater variability in the average alcohol sold in the most negative quartile group as seen from a very big lower whisker.

```
boxplot(alcohol_offence$‘Average sold per person in Liters‘ ~ alcohol_offence$‘Police Region‘, ylab="Av
        main="Box Plot of average alcohol sold")
```



- Using the multivariate method for comparing two numeric variables, a scatter plot is useful in dealing with two quantitative variables, offence per 100,000 population and average alcohol consumption per person.
- The scatter plot below indicates there are some outliers in the data as seen from the dots that are far away from the cluster.

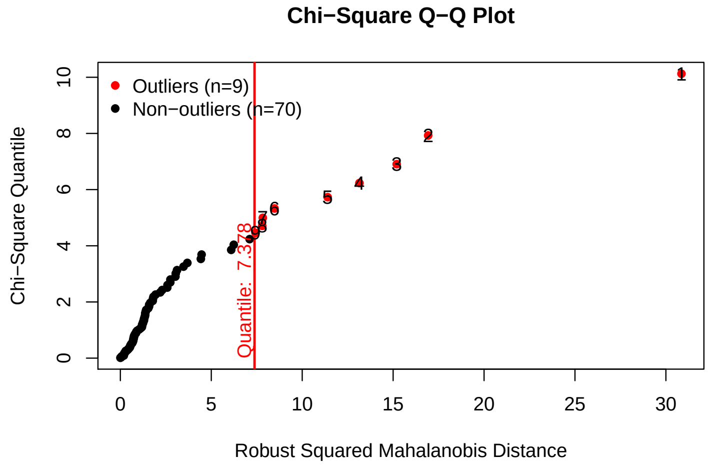
```
alcohol_offence %>% plot(‘Offence per 100,000 population‘ ~ ‘Average sold per person in Liters‘,data=.)
```



- Using the Mahalanobis distance the multivariate outliers is detected using the chi square distribution method.
- From the QQ plot the Mahalanobis distance suggests that there are 9 outliers in the subset offence_alcohol dataset.

```
offence_alcohol <- alcohol_offence %>% select(‘Average sold per person in Liters‘, ‘Offence per 100,000

results <- mvn(data= offence_alcohol,
               multivariateOutlierMethod = "quan",
               showOutliers = TRUE)
```



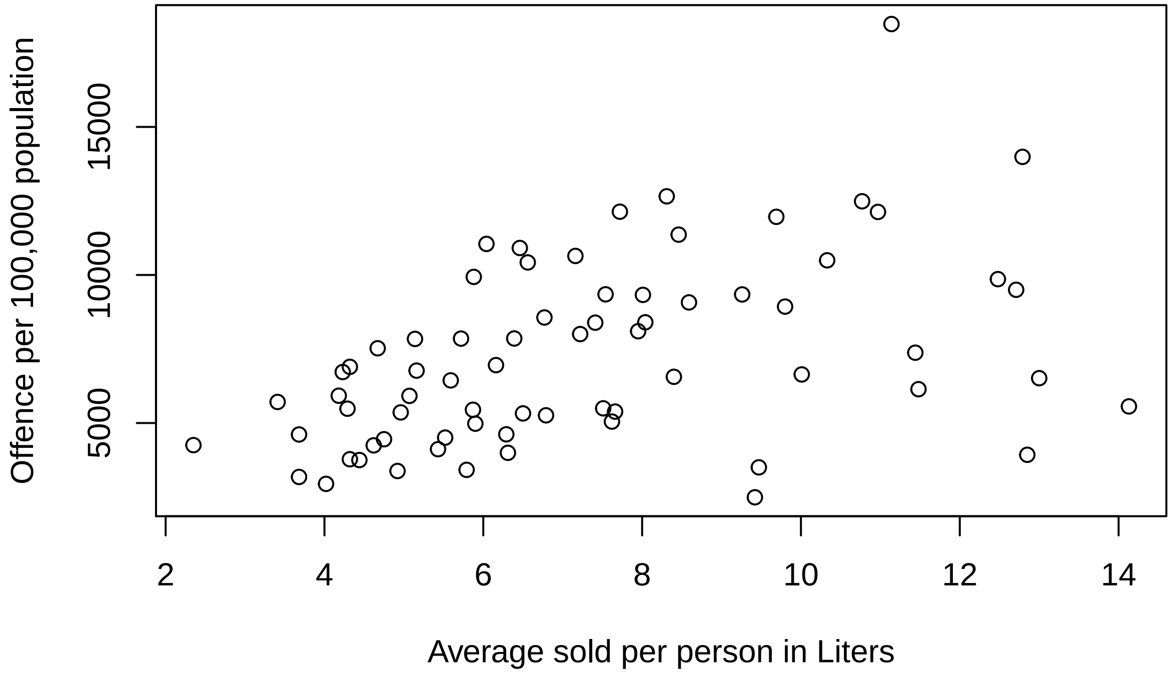
- Therefore the 9 outliers are excluded into the alcohol_clean3 dataset and after putting the data into a scatter plot there are less outliers from the cluster and the dataset from 79 observation have when subsetted to 70 observations after the 9 outliers were removed.

```
results$multivariateOutliers
```

```
##    Observation Mahalanobis Distance Outlier
## 1           1               30.856    TRUE
## 2           2               16.927    TRUE
## 3           3               15.197    TRUE
## 4           4               13.147    TRUE
## 5           5               11.398    TRUE
## 6           6                8.476    TRUE
## 7           7                7.834    TRUE
## 8           8                7.800    TRUE
## 9           9                7.431    TRUE
```

```
alcohol_clean3 <- offence_alcohol[-(1:9),]

alcohol_clean3 %>% plot(`Offence per 100,000 population`~`Average sold per person in Liters`,data=.)
```



```
dim(alcohol_clean3)
```

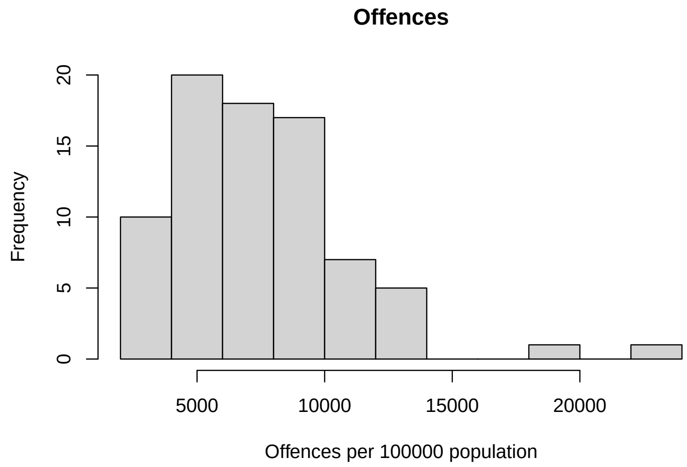
```
## [1] 70  2
```

Transformation

- The offence per 100,000 population is plotted using a histogram and by observation the distribution appears to be right skewed. Using the skewness function, the data for offence per 100,000 population produces 1.45. A skewness of more than 1 is considered to be highly skewed and in this case to the right.

- Therefore, the offences variable will be transformed using various methods to reduce the skewness of the distribution. Data that is normally distributed allows for easier and more robust statistical analysis.

```
hist(alcohol_offence$`Offence per 100,000 population`, xlab="Offences per 100000 population", main="Off
```

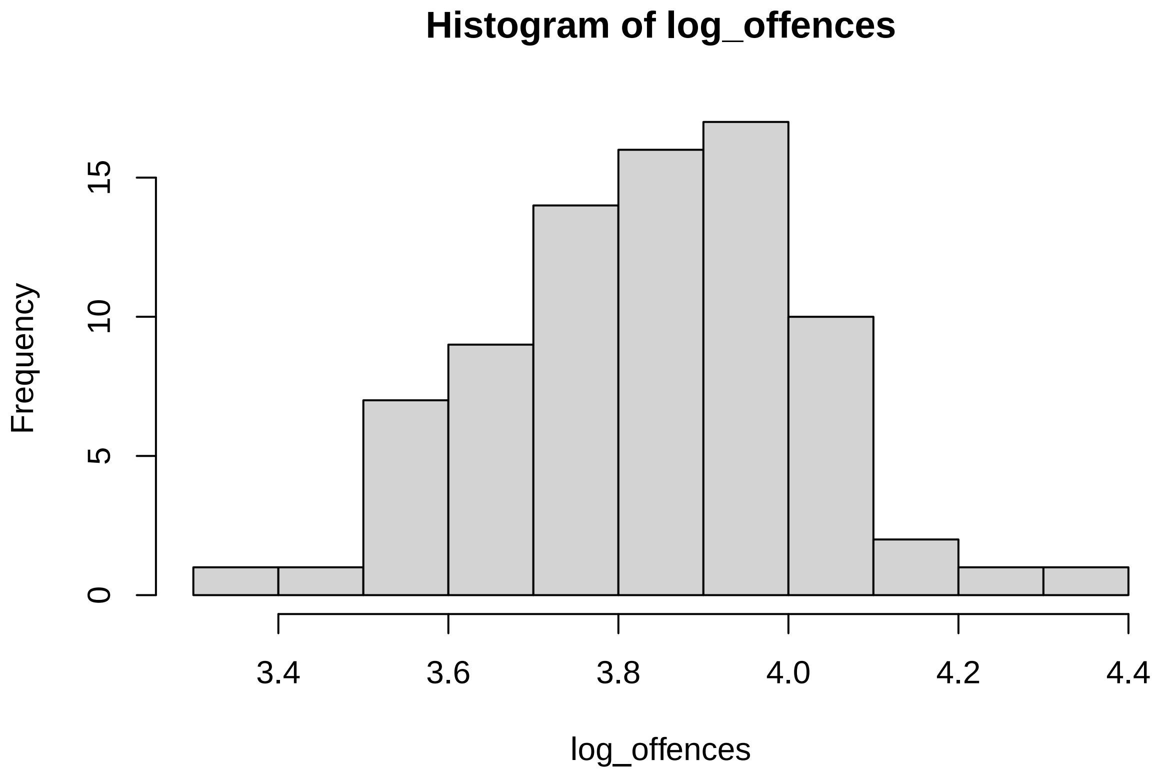


```
skewness(alcohol_offence$`Offence per 100,000 population`)
```

```
## [1] 1.451044
```

- Using the base 10 logarithmic transformation reduces right skewness, As seen from the histogram the distribution appears to be less skewed and more normal than before.
- Using the base e logarithmic transformation also reduces the right skewness of the offence data.
- Based on appearance, the base e logarithmic transformation looks more normal compared to the base 10 logarthmic transformation. However using the skewness function, the skewness is reduce from 1.45 to 0.013 for both transformation methods. In this scenario the log transformation is extremely useful in transforming the data into a normal distribution and is the most effective method.

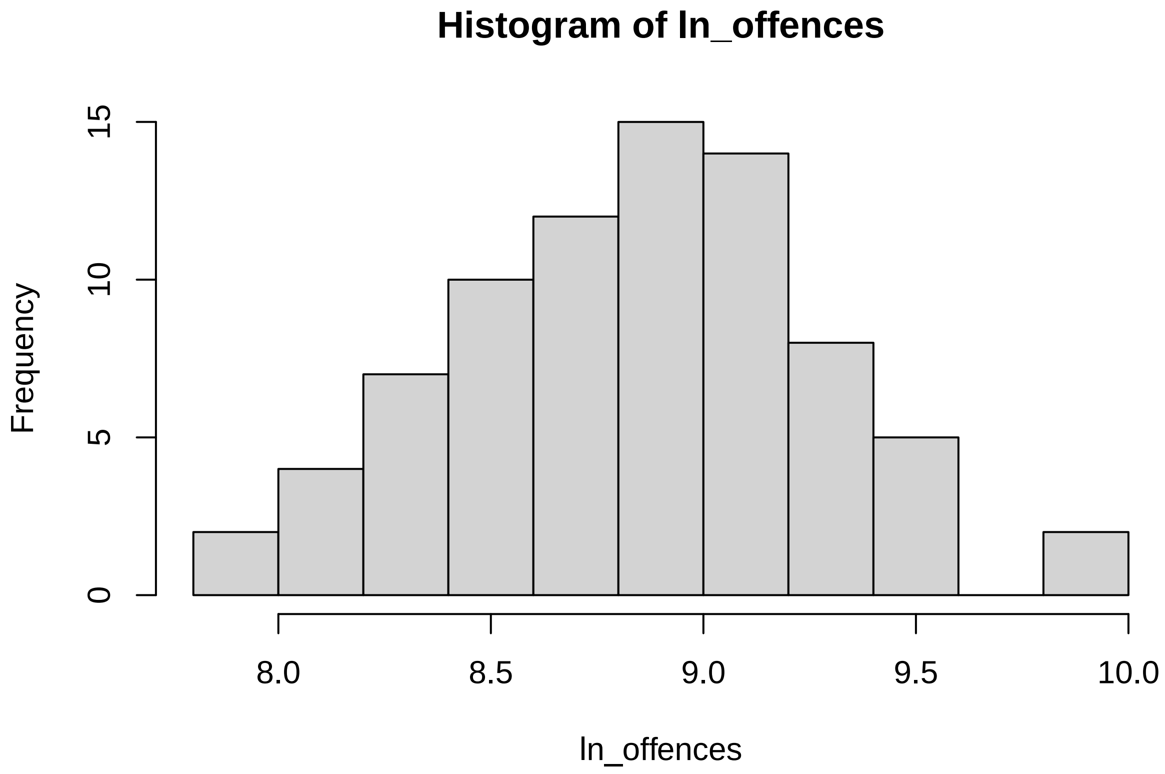
```
log_offences <- log10(alcohol_offence$`Offence per 100,000 population`)
hist(log_offences)
```



```r
skewness(log_offences)
```

```
## [1] 0.01334287
```

```r
ln_offences <- log(alcohol_offence$`Offence per 100,000 population`)
hist(ln_offences)
```

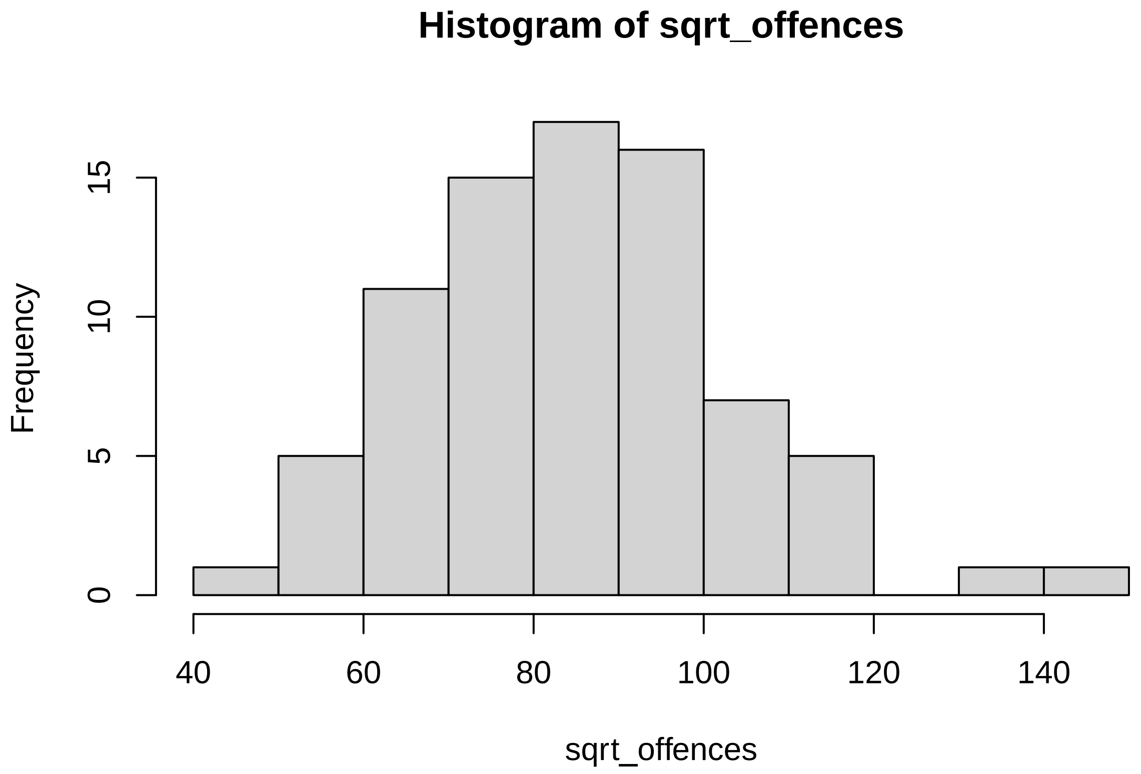


```
skewness(ln_offences)
```

```
## [1] 0.01334287
```

- The square root transformation also reduces the right skewness in the offence data and this method allows the transformation to be applied to zero values although there are none in this case. The skewness is reduced from 1.45 to 0.66 and is significant reduction in the right skewness of data.

```
sqrt_offences <- sqrt(alcohol_offence$`Offence per 100,000 population`)
hist(sqrt_offences)
```

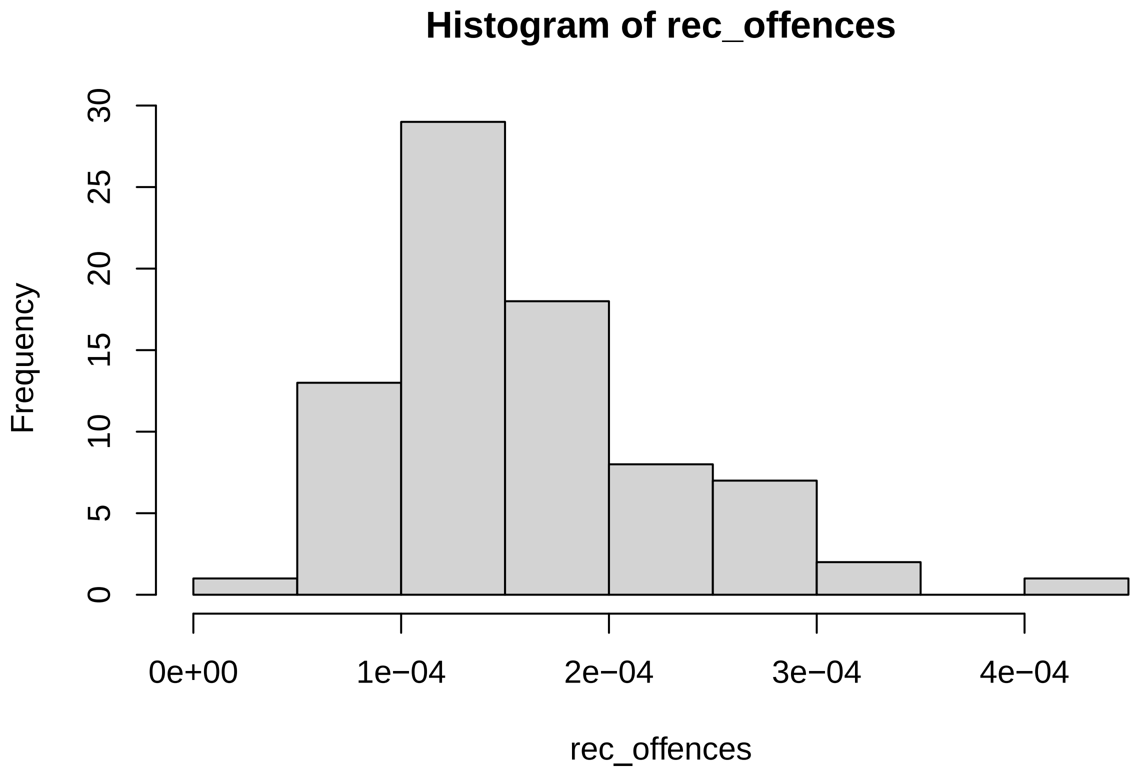


```
skewness(sqrt_offences)
```

```
## [1] 0.6562218
```

- The reciprocal transformation reduces the skewness from 1.45 to 1.04. As the data does not have many large values in offence, the most drastic effect is not the most drastic compared to the log transformations. For reciprocal transformation the skewness is reduce from 1.45 to 1.04.

```
rec_offences <- 1/alcohol_offence$`Offence per 100,000 population`
hist(rec_offences)
```

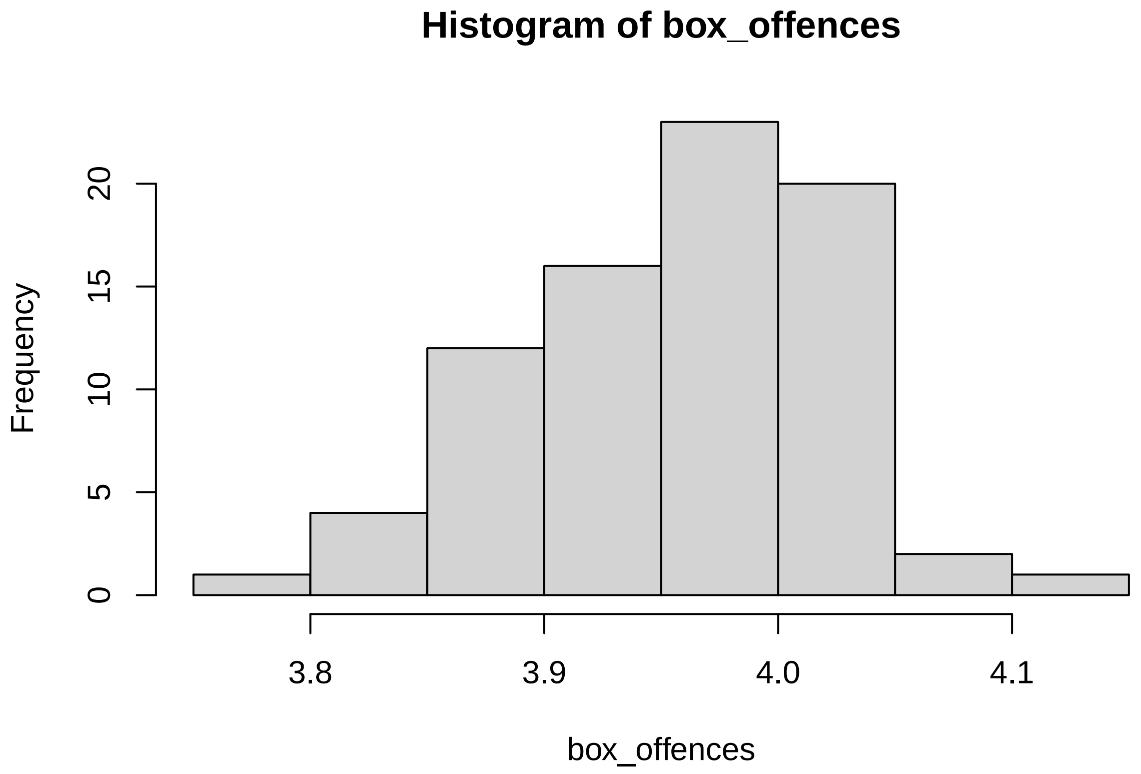


```
skewness(rec_offences)
```

```
## [1] 1.041724
```

- The box cox transformation gives us a skewness of -0.23 which is a left skewness and is not the most preferred method as there are no negative values in the offence per 100,000 population.

```
box_offences <- BoxCox(alcohol_offence$`Offence per 100,000 population`, lambda="auto")
hist(box_offences)
```



```
skewness(box_offences)
```

```
## [1] -0.229151
```

Conclusion

- The two dataset alcohol and offences were tidied and manipulated using various technique such as gsub, rounding, rename, group by, summarise, factor to ensure that the data structure and variables were appropriate and understandable. The alcohol_offence dataset was created using a join function and new variables were added in for a more robust approach to the investigation such as total population and average alcohol sold per person.
- In addition, there were missing data in the form of the totals of each police region and another two regions that were not part of the investigation. These missing values were scanned, extracted and removed accordingly to ensure that the data is tidy.
- Outliers were also discovered through the boxplot that was used on each numeric variable. This investigation points to the high number of alcohol sold and offence committed in Melbourne due to the high population being the most liveable city with a exponentially large number of nightclubs and bars.
- Lastly, the offence variable was transformed appropriately using the logarithmic transformation that have turned a heavily right skewed data into one that is normal. When the data have been transformed into a distribution that is more normal, the data can be used for more statistically analysis such as the Student t test and Chi square test of association.

References

Crime Statistics Agency (2020). Crime Statistics Agency: Recorded Offences. https://www.crimestatistics.vic.gov.au/crime-statistics/latest-victorian-crime-data/recorded-offences

Changyong Feng, Hongyue Wang, Naiji Lu, Tian Chen, Hua He, Ying Lu, and Xin M. Tu. Log-transformation and its implications for data analysis. Shanghai Arch Psychiatry. 2014 Apr; 26(2): 105-109. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4120293/#:~:text=Using%20the%20log%20transformation%20to%20make%20data%20conform%20to%20normality&text=If%20the%20original%20data%20follows,does%20remove%20or%20reduce%20skewness.