Assignment 2

Md Tauhidul Islam

2024-03-22

Setup

```
# Loading the required packages
library(tidyverse)
library(dplyr)
library(readr)
library(tidyr)
library(inspectdf)
library(e1071)
library(caret)
```

Q1. Loading the data

```
ysn = 1895813 # My student number
filenum <- (ysn+1) %% 3
filenum
```

```
## [1] 0
```

```
filename <- paste0("./data/gadget_",filenum,".csv")
filename
```

```
## [1] "./data/gadget_0.csv"
```

```
my_dataset <- read_csv(filename)
my_dataset # Output the first 10 lines of the dataset
```

```
## # A tibble: 1,001 x 4
##    name        population advertising    sales
##    <chr>       <chr>      <chr>          <dbl>
##  1 Fairhampton 415149     $1,467,743.39 141335
##  2 Ashwich     505932     $1,426,770.06  96672
##  3 Rockcaster  406564     $1,147,775.28  56084
##  4 Hogstead    494200     $1,114,590.85  43865
##  5 Snowgrad    323581     $888,153.32    66033
```

```
##   6 Sweetport   309610     $400,624.41    16066
##   7 Passcaster  780917     $3,209,478.58 327318
##   8 Parkdale    404280     $1,632,156.47 174979
##   9 Hogbury     499986     $1,445,273.80 105101
##  10 Readingford 242013     $615,491.11       NA
## # i 991 more rows
```

Q2. Adding a new column of row numbers

```
data_with_row <- my_dataset %>%
  mutate(row_num = row_number()) %>%
  relocate(row_num, .before = name) # Reordering row number to the leftmost column
data_with_row
```

```
## # A tibble: 1,001 x 5
##     row_num name        population advertising    sales
##       <int> <chr>       <chr>      <chr>          <dbl>
##   1       1 Fairhampton 415149     $1,467,743.39 141335
##   2       2 Ashwich     505932     $1,426,770.06  96672
##   3       3 Rockcaster  406564     $1,147,775.28  56084
##   4       4 Hogstead    494200     $1,114,590.85  43865
##   5       5 Snowgrad    323581     $888,153.32    66033
##   6       6 Sweetport   309610     $400,624.41    16066
##   7       7 Passcaster  780917     $3,209,478.58 327318
##   8       8 Parkdale    404280     $1,632,156.47 174979
##   9       9 Hogbury     499986     $1,445,273.80 105101
##  10      10 Readingford 242013     $615,491.11       NA
## # i 991 more rows
```

Q3. Types of variables

- Variable 1 (row_num): Quantitative Ordinal. This numeric variable represents the number of rows in an order.
- Variable 2 (name): Categorical Nominal. This variable represents names of the cities which has no order.
- Variable 3 (population): Quantitative Discrete. This variable counts the number of people in corresponding cities. Counts are discrete.
- Variable 4 (advertising): Quantitative Continuous. This variable represents the amount of money spent for advertising in corresponding cities. It can take on any value within a range, including fractional values which indicates precise values rather than discrete counts.
- Variable 5 (sales): Quantitative Discrete. This variable represents the number of sales in corresponding cities and this has no order. This is a discrete data as this is a count of gadget sale. We cannot sale gadgets in fraction.

Q4. Cleaning and taming the data

```r
# First let's check the summary
summary(data_with_row)
```

```
##     row_num          name            population         advertising
##  Min.   :   1   Length:1001        Length:1001        Length:1001
##  1st Qu.: 251   Class :character   Class :character   Class :character
##  Median : 501   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 501
##  3rd Qu.: 751
##  Max.   :1001
##
##      sales
##  Min.   :-3.285e+04
##  1st Qu.: 3.159e+04
##  Median : 6.162e+04
##  Mean   : 1.110e+12
##  3rd Qu.: 1.053e+05
##  Max.   : 1.110e+15
##  NA's   :1
```

```r
# We need to tame the data to be able to do clean the data

tamed_data <- data_with_row %>%
  mutate(
    row_num = as.integer(row_num),
    name = as.character(name),
    population = as.integer(population),
    advertising = as.numeric(str_replace_all(advertising, "\\$|,", "")),
    sales = as.integer(sales)
  )

# Let's inspect the missing values of tamed data now.

inspect_na(tamed_data)
```

```
## # A tibble: 5 x 3
##   col_name      cnt  pcnt
##   <chr>       <int> <dbl>
## 1 name            3 0.300
## 2 population      3 0.300
## 3 sales           2 0.200
## 4 row_num         0 0
## 5 advertising     0 0
```

Let's replace missing city names and removing other missing rows

```r
tamed_data <- tamed_data %>%
  mutate(name = if_else(is.na(name),
                        paste0("noname_ ", cumsum(is.na(name))), name)) %>%
  drop_na()

#Let's check missing values now

inspect_na(tamed_data)
```

```
## # A tibble: 5 x 3
##   col_name      cnt  pcnt
##   <chr>       <int> <dbl>
## 1 row_num         0     0
## 2 name            0     0
## 3 population      0     0
## 4 advertising     0     0
## 5 sales           0     0
```

Now, we can see the data has no more missing values.

Now, let's remove any duplicate rows

```r
# First, let's find if there are any duplicates
num_duplicated_rows <- sum(duplicated(tamed_data))
num_duplicated_rows
```

```
## [1] 0
```

There are no duplicated rows present in our dataset. Therefore, no actions are needed to remove duplicates.

Let's convert any negative values into positve values

```r
tamed_data <- tamed_data %>%
  mutate(
    sales = abs(sales),
    population = abs(population),
    advertising = abs(advertising)
  )

# Let's the the summary now to see some significant changes
summary(tamed_data)
```

```
##     row_num           name             population       advertising
##  Min.   :   1.0   Length:996         Min.   :      1   Min.   :1.000e+00
##  1st Qu.: 250.8   Class :character   1st Qu.: 225862   1st Qu.:5.402e+05
##  Median : 499.5   Mode  :character   Median : 334436   Median :8.854e+05
##  Mean   : 501.0                      Mean   : 361609   Mean   :2.011e+06
```

```
##  3rd Qu.: 751.2                      3rd Qu.: 468945   3rd Qu.:1.327e+06
##  Max.   :1001.0                      Max.   :1039360   Max.   :1.000e+09
##      sales
##  Min.   :     1
##  1st Qu.: 31618
##  Median : 61710
##  Mean   : 78557
##  3rd Qu.:105275
##  Max.   :595122
```

No more negative numbers in our data now.

Rmoving suspicious number

Now, we'll remove Suspiciously small or large numbers (in absolute value). let' plot our three main variables for a better understanding. We'll manually find out the range that we want to keep in our data from the plot.

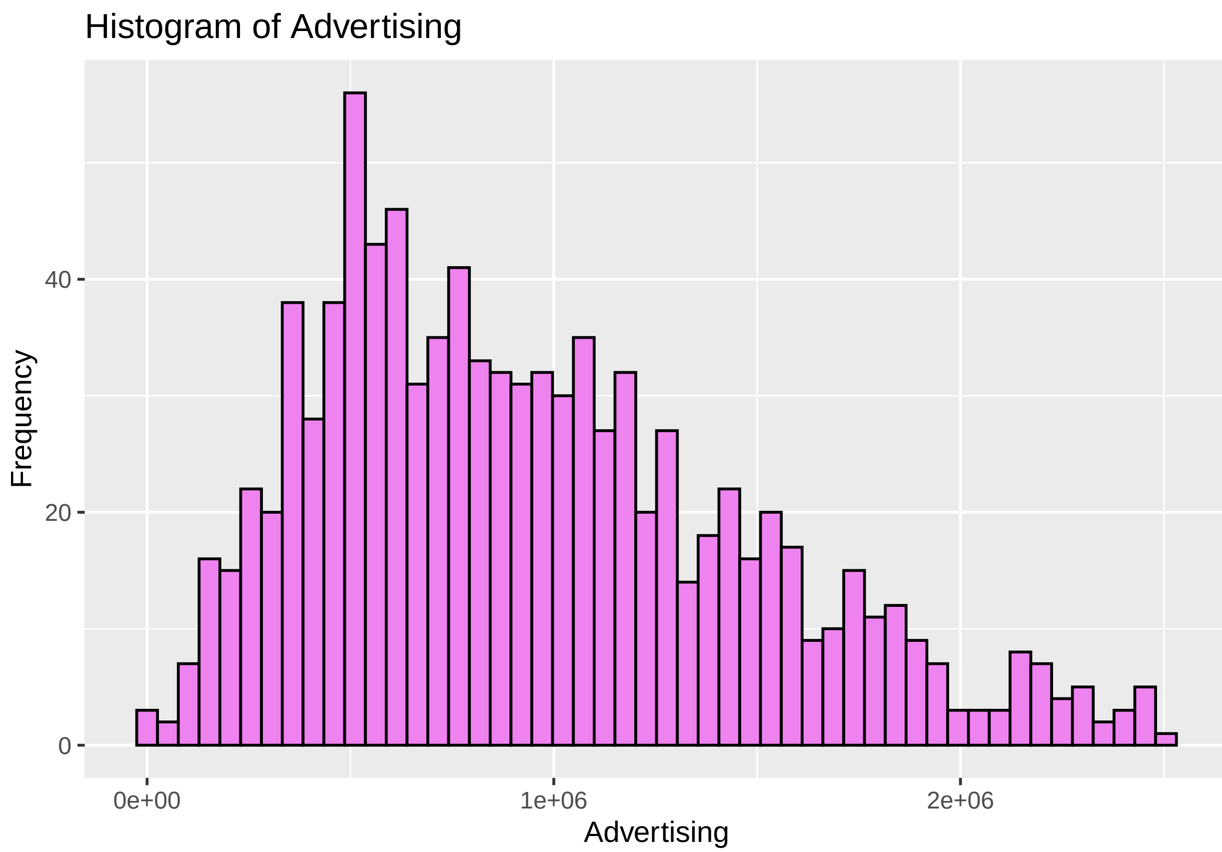
First of all let's find and remove the outliers.

```
q1_ad <- quantile(tamed_data$advertising, 0.25) # 1st quartile
q3_ad <- quantile(tamed_data$advertising, 0.75) # 3rd quartile

iqr_ad <- q3_ad - q1_ad # interquartile range

lower_data_point_ad <- q1_ad - 1.5 * iqr_ad
upper_data_point_ad <- q3_ad + 1.5 * iqr_ad

cleaned_data <- tamed_data %>%
  filter(tamed_data$advertising > lower_data_point_ad & tamed_data$advertising < upper_data_point_ad)

# Let's see the summary now

summary(cleaned_data)
```

```
##     row_num           name             population       advertising
##  Min.   :   1.0   Length:964         Min.   :      2   Min.   :      1
##  1st Qu.: 249.8   Class :character   1st Qu.: 222174   1st Qu.: 532027
##  Median : 499.5   Mode  :character   Median : 325308   Median : 855684
##  Mean   : 500.8                      Mean   : 349629   Mean   : 947041
##  3rd Qu.: 750.5                      3rd Qu.: 454133   3rd Qu.:1267294
##  Max.   :1001.0                      Max.   :1039360   Max.   :2505400
##      sales
##  Min.   :     3
##  1st Qu.: 31438
##  Median : 60126
##  Mean   : 71836
##  3rd Qu.:101128
##  Max.   :318068
```

```
# Let's Plot histogram for 'advertising' for visualisation

ggplot(cleaned_data, aes(x = advertising)) +
```

```r
  geom_histogram(bins = 50, fill = "violet", color = "black") +
  labs(x = "Advertising", y = "Frequency", title = "Histogram of Advertising")
```



We are still having suspiciously low numbers, as lower whisker is not able to eliminate the absolute low number in this case.

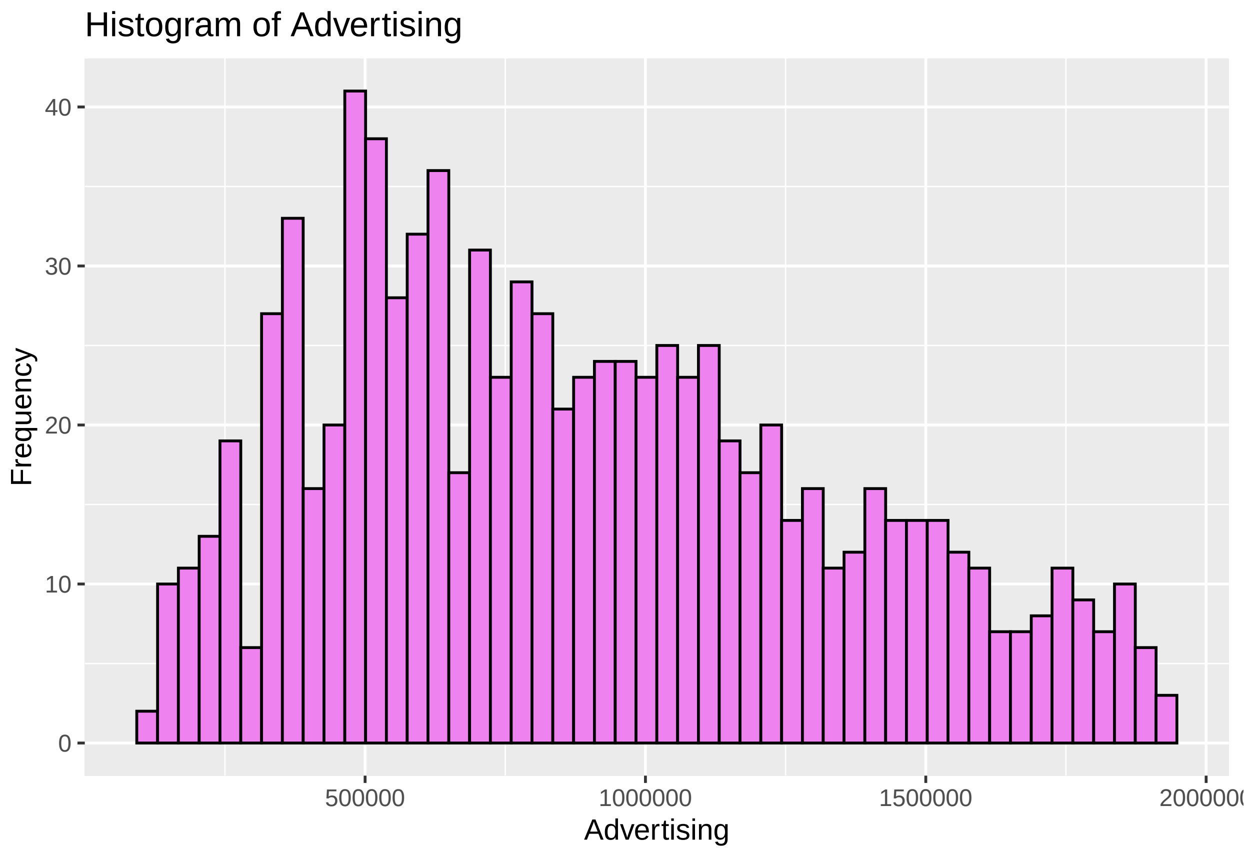
Therefore, based on the summary and histogram let's consider the data below 1st percentile and above 95 percentile is suspicious. We we'll remove the suspicious data based on our consideration.

```r
lower_lim_ad <- quantile(cleaned_data$advertising, 0.01)
upper_lim_ad  <- quantile(cleaned_data$advertising, 0.95)

cleaned_data <- cleaned_data %>%
  filter(cleaned_data$advertising > lower_lim_ad & cleaned_data$advertising < upper_lim_ad)

summary(cleaned_data)
```

```
##     row_num          name             population      advertising
##  Min.   :   1.0   Length:905         Min.   :     2   Min.   : 106160
##  1st Qu.: 248.0   Class :character   1st Qu.:219282   1st Qu.: 529683
##  Median : 499.0   Mode  :character   Median :313478   Median : 832230
##  Mean   : 501.5                      Mean   :334698   Mean   : 889032
##  3rd Qu.: 754.0                      3rd Qu.:435525   3rd Qu.:1197159
##  Max.   :1001.0                      Max.   :974181   Max.   :1924145
##      sales
##  Min.   :     3
```

```
##  1st Qu.: 31025
##  Median : 57557
##  Mean   : 65732
##  3rd Qu.: 93157
##  Max.   :210717
```

```
ggplot(cleaned_data, aes(x = advertising)) +
  geom_histogram(bins = 50, fill = "violet", color = "black") +
  labs(x = "Advertising", y = "Frequency", title = "Histogram of Advertising")
```

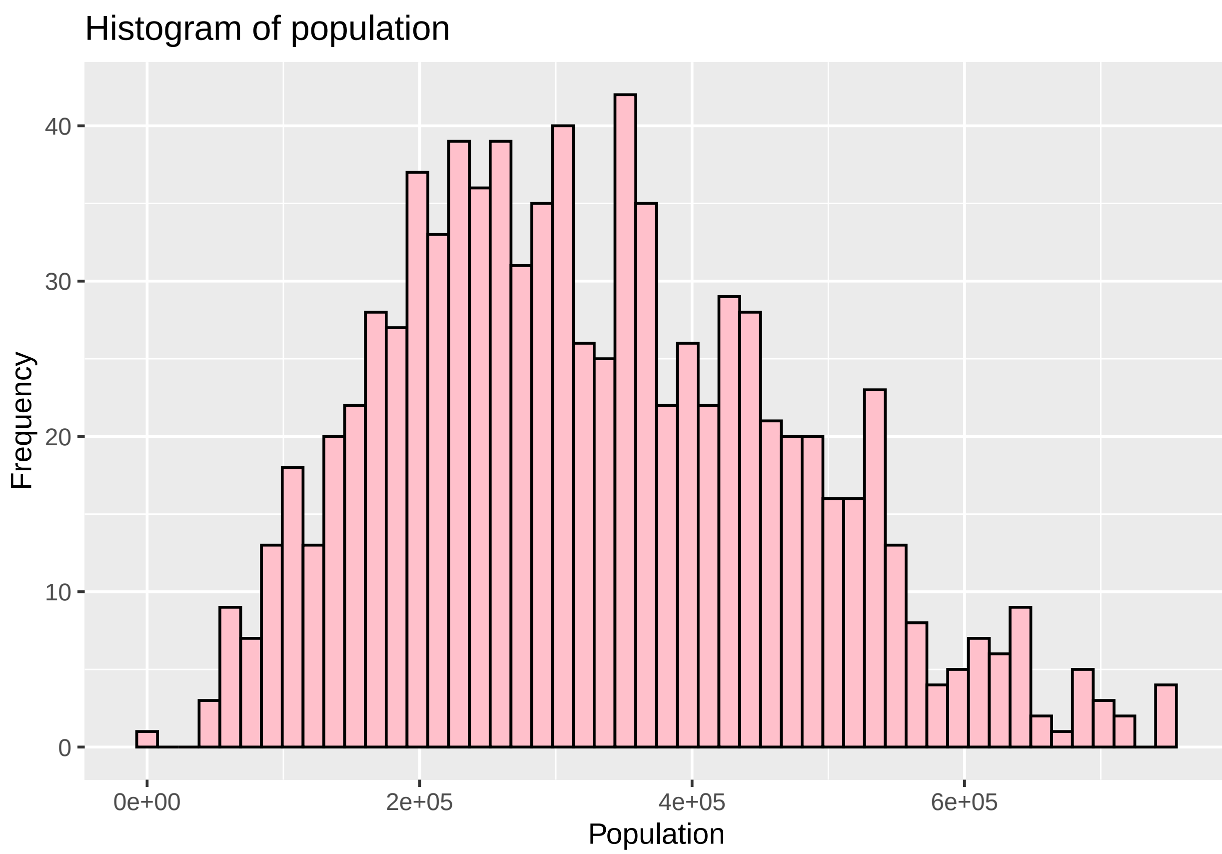


Now the plot looks much better. We'll do the same with population and sales.

```
q1_pop <- quantile(cleaned_data$population, 0.25) # 1st quartile
q3_pop <- quantile(cleaned_data$population, 0.75) # 3rd quartile

iqr_pop <- q3_pop - q1_pop # interquartile range

lower_data_point_pop <- q1_pop - 1.5 * iqr_pop
upper_data_point_pop <- q3_pop + 1.5 * iqr_pop

cleaned_data <- cleaned_data %>%
  filter(cleaned_data$population > lower_data_point_pop & cleaned_data$population < upper_data_point_pop

# Let's see the summary now

summary(cleaned_data)
```

```
##     row_num          name             population      advertising
##  Min.   :   1.0   Length:891         Min.   :     2   Min.   : 106160
##  1st Qu.: 247.5   Class :character   1st Qu.:216534   1st Qu.: 522676
##  Median : 497.0   Mode  :character   Median :310477   Median : 818251
##  Mean   : 500.5                      Mean   :327484   Mean   : 877153
##  3rd Qu.: 753.5                      3rd Qu.:428936   3rd Qu.:1174515
##  Max.   :1001.0                      Max.   :747870   Max.   :1913454
##      sales
##  Min.   :   698
##  1st Qu.: 30976
##  Median : 57115
##  Mean   : 65434
##  3rd Qu.: 92720
##  Max.   :210717
```

```
# Plot histogram for 'population' for visualisation
ggplot(cleaned_data, aes(x = population)) +
  geom_histogram(bins = 50, fill = "pink", color = "black") +
  labs(x = "Population", y = "Frequency", title = "Histogram of population")
```



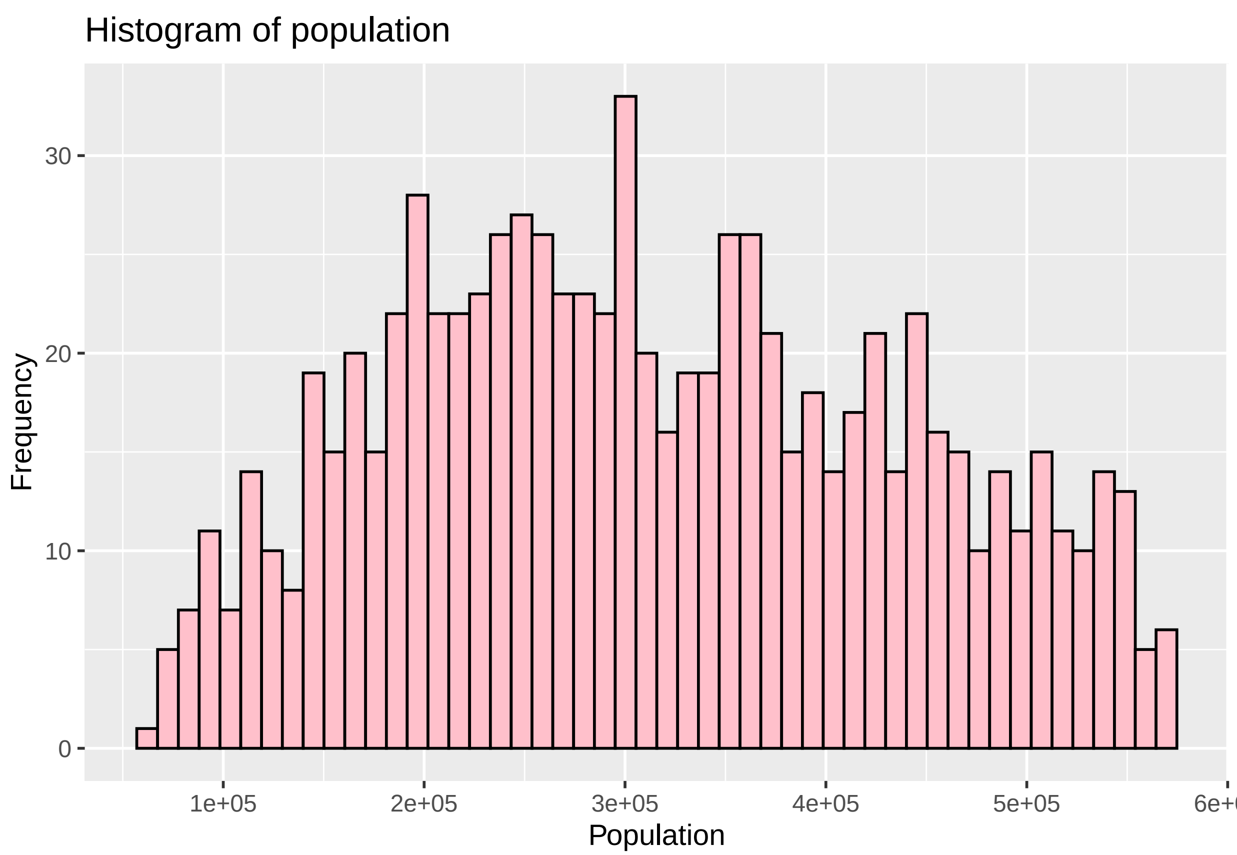
Same case happened with population as the whisker is not able to remove the suspiciously lower number in absoulate value. Therefore, we will proceed with the same process as advertising.

```
lower_lim_pop <- quantile(cleaned_data$population, 0.01)
upper_lim_pop  <- quantile(cleaned_data$population, 0.95)
```

```r
cleaned_data <- cleaned_data %>%
  filter(cleaned_data$population > lower_lim_pop & cleaned_data$population < upper_lim_pop)

summary(cleaned_data)
```

```
##     row_num           name             population       advertising
##  Min.   :   1.0   Length:837         Min.   : 66832   Min.   : 106160
##  1st Qu.: 243.0   Class :character   1st Qu.:214368   1st Qu.: 518957
##  Median : 501.0   Mode  :character   Median :302353   Median : 792100
##  Mean   : 502.9                      Mean   :313099   Mean   : 849836
##  3rd Qu.: 758.0                      3rd Qu.:411127   3rd Qu.:1128470
##  Max.   :1001.0                      Max.   :574234   Max.   :1913454
##      sales
##  Min.   :   698
##  1st Qu.: 30938
##  Median : 54816
##  Mean   : 64220
##  3rd Qu.: 89860
##  Max.   :210717
```

```r
ggplot(cleaned_data, aes(x = population)) +
  geom_histogram(bins = 50, fill = "pink", color = "black") +
  labs(x = "Population", y = "Frequency", title = "Histogram of population")
```

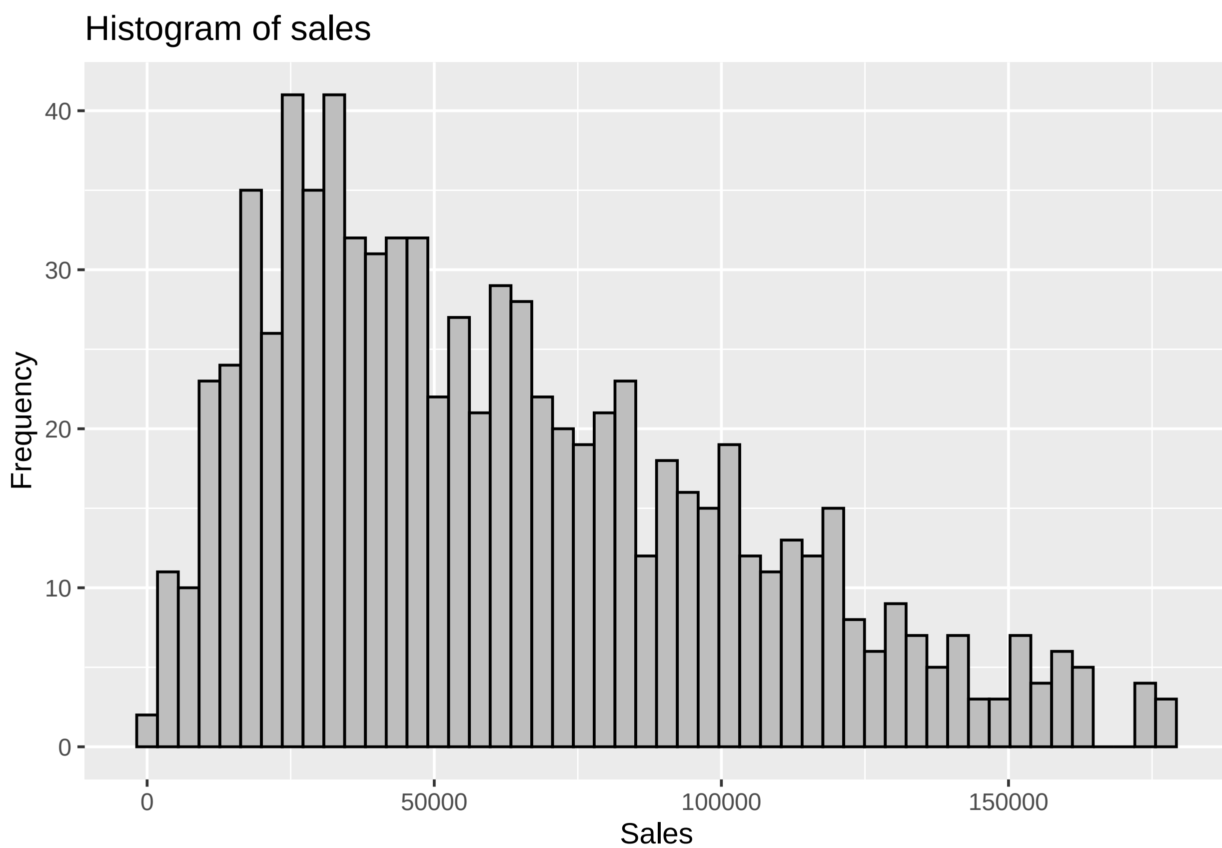


Let's proceed with sales now.

```r
q1_sales <- quantile(cleaned_data$sales, 0.25) # 1st quartile
q3_sales <- quantile(cleaned_data$sales, 0.75) # 3rd quartile
iqr_sales <- q3_sales - q1_sales # Interquartile range

lower_data_point_sales <- q1_sales - 1.5 * iqr_sales
upper_data_point_sales <- q3_sales + 1.5 * iqr_sales

cleaned_data <- cleaned_data %>%
  filter(cleaned_data$sales > lower_data_point_sales & cleaned_data$sales < upper_data_point_sales)

summary(cleaned_data)
```

```
##     row_num          name             population      advertising
##  Min.   :   1.0   Length:827         Min.   : 66832   Min.   : 106160
##  1st Qu.: 245.5   Class :character   1st Qu.:213361   1st Qu.: 516656
##  Median : 506.0   Mode  :character   Median :300826   Median : 787483
##  Mean   : 504.4                      Mean   :311266   Mean   : 837936
##  3rd Qu.: 760.0                      3rd Qu.:407451   3rd Qu.:1115089
##  Max.   :1001.0                      Max.   :574234   Max.   :1856640
##      sales
##  Min.   :   698
##  1st Qu.: 30752
##  Median : 54317
##  Mean   : 62588
##  3rd Qu.: 88809
##  Max.   :178119
```

```r
# Plot histogram for 'sales' for visualisation
ggplot(cleaned_data, aes(x = sales)) +
  geom_histogram(bins = 50, fill = "gray", color = "black") +
  labs(x = "Sales", y = "Frequency", title = "Histogram of sales")
```

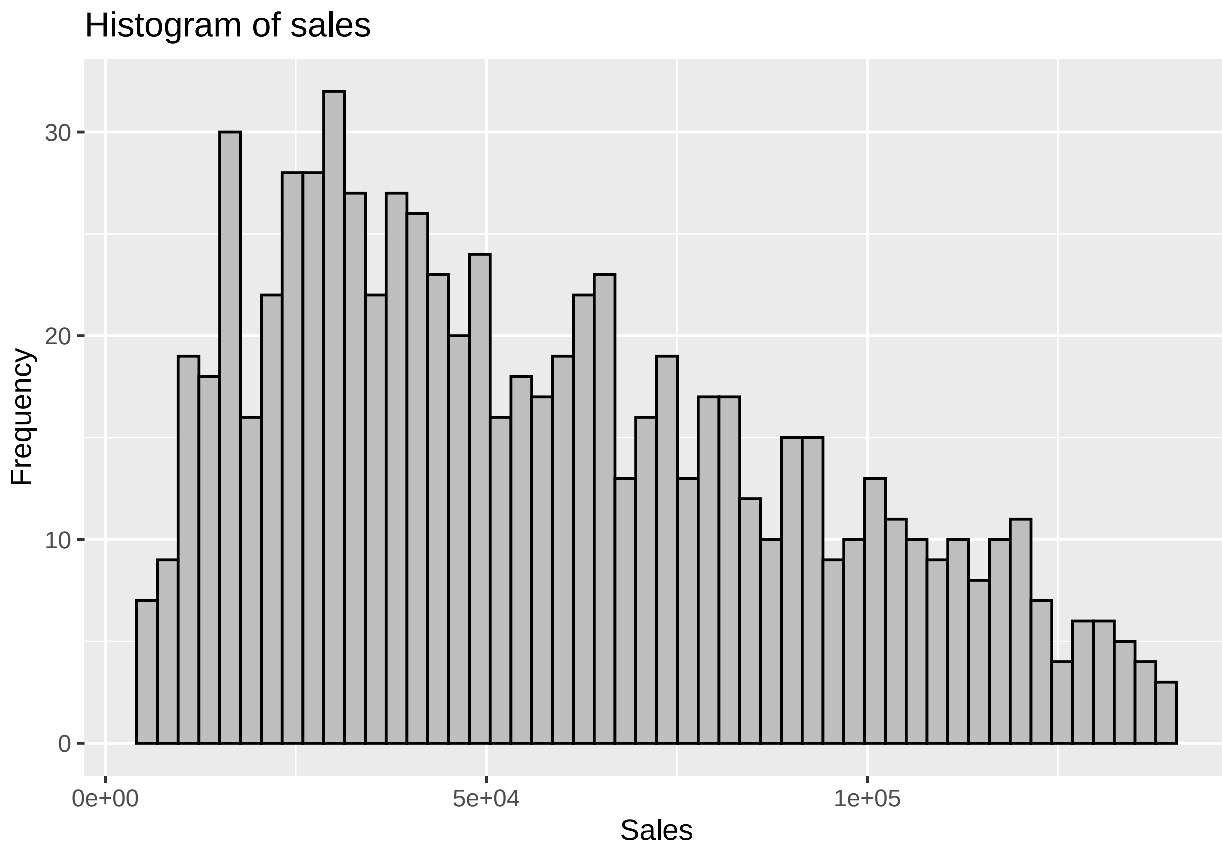


We'll proceed with 1 and 95 percentile for further filtering.

```
lower_lim_sales <- quantile(cleaned_data$sales, 0.01)
upper_lim_sales  <- quantile(cleaned_data$sales, 0.95)

cleaned_data <- cleaned_data %>%
  filter(cleaned_data$sales > lower_lim_sales & cleaned_data$sales < upper_lim_sales)

summary(cleaned_data)
```

```
##     row_num           name             population      advertising
##  Min.   :   2.0   Length:776         Min.   : 66832   Min.   : 131176
##  1st Qu.: 240.8   Class :character   1st Qu.:210134   1st Qu.: 513058
##  Median : 497.5   Mode  :character   Median :295935   Median : 763579
##  Mean   : 500.5                      Mean   :306292   Mean   : 806150
##  3rd Qu.: 757.2                      3rd Qu.:396949   3rd Qu.:1074083
##  Max.   :1001.0                      Max.   :574234   Max.   :1751698
##      sales
##  Min.   :  5261
##  1st Qu.: 30255
##  Median : 52464
##  Mean   : 58204
##  3rd Qu.: 82655
##  Max.   :139015
```

```r
ggplot(cleaned_data, aes(x = sales)) +
  geom_histogram(bins = 50, fill = "gray", color = "black") +
  labs(x = "Sales", y = "Frequency", title = "Histogram of sales")
```




```r
# Displaying data
cleaned_data
```

```
## # A tibble: 776 x 5
##    row_num name        population advertising  sales
##      <int> <chr>            <int>       <dbl>  <int>
##  1       2 Ashwich         505932    1426770.  96672
##  2       3 Rockcaster      406564    1147775.  56084
##  3       4 Hogstead        494200    1114591.  43865
##  4       5 Snowgrad        323581     888153.  66033
##  5       6 Sweetport       309610     400624.  16066
##  6       9 Hogbury         499986    1445274. 105101
##  7      11 Hallcester      177261     495314.  49104
##  8      12 Farmhampton     274909     475015.  16465
##  9      13 Norness         233164     687985.  53140
## 10      15 Princemouth     189518     308733.  17092
## # i 766 more rows
```

```r
# Let' see the final summary of our data
summary(cleaned_data)
```

```
##     row_num            name             population      advertising    
##  Min.   :   2.0   Length:776         Min.   : 66832   Min.   : 131176  
##  1st Qu.: 240.8   Class :character   1st Qu.:210134   1st Qu.: 513058  
##  Median : 497.5   Mode  :character   Median :295935   Median : 763579  
##  Mean   : 500.5                      Mean   :306292   Mean   : 806150  
##  3rd Qu.: 757.2                      3rd Qu.:396949   3rd Qu.:1074083  
##  Max.   :1001.0                      Max.   :574234   Max.   :1751698  
##      sales       
##  Min.   :  5261  
##  1st Qu.: 30255  
##  Median : 52464  
##  Mean   : 58204  
##  3rd Qu.: 82655  
##  Max.   :139015
```

Now, the data looks perfect for our further analysis.

Q5. Creating new variables

```
cleaned_data <- cleaned_data %>%
  mutate(
    sales_pct = (sales/population)*100,
    adv_exp_pp = advertising/population
  )
cleaned_data
```

```
## # A tibble: 776 x 7
##    row_num name        population advertising  sales sales_pct adv_exp_pp
##      <int> <chr>            <int>       <dbl>  <int>     <dbl>      <dbl>
##  1       2 Ashwich         505932    1426770.  96672     19.1        2.82
##  2       3 Rockcaster      406564    1147775.  56084     13.8        2.82
##  3       4 Hogstead        494200    1114591.  43865      8.88       2.26
##  4       5 Snowgrad        323581     888153.  66033     20.4        2.74
##  5       6 Sweetport       309610     400624.  16066      5.19       1.29
##  6       9 Hogbury         499986    1445274. 105101     21.0        2.89
##  7      11 Hallcester      177261     495314.  49104     27.7        2.79
##  8      12 Farmhampton     274909     475015.  16465      5.99       1.73
##  9      13 Norness         233164     687985.  53140     22.8        2.95
## 10      15 Princemouth     189518     308733.  17092      9.02       1.63
## # i 766 more rows
```

Clasification of the new variables:

- Both of these new variables are floating numbers and can be considered as ratio variables which is a type of quantitative variable. These variables have a clear definition of being 0 which satisfies the characteristics of ratio variable. 0% of sales_pct indicates no sales at all, and 0 adv_exp_pp indicates no amount being spend on advertisement per person.

Q6. Taking random sample

```
set.seed(ysn)
sampled_data <- cleaned_data %>%
  sample_n(700)
sampled_data
```

```
## # A tibble: 700 x 7
##    row_num name        population advertising sales sales_pct adv_exp_pp
##      <int> <chr>            <int>       <dbl> <int>     <dbl>      <dbl>
##  1     875 Aubury           66832     212081. 16459     24.6        3.17
##  2     739 Princeville     181109     472336. 33356     18.4        2.61
##  3     466 Summerbury      337958     789674. 58494     17.3        2.34
##  4     997 Faybury         193460     562506. 54317     28.1        2.91
##  5      75 Kettletown      206810     500767. 30020     14.5        2.42
##  6     474 Sagefield       201878     573511. 37479     18.6        2.84
##  7     103 Passview        229325     835440. 75046     32.7        3.64
##  8     709 Frostworth      150259     481705. 38772     25.8        3.21
##  9     318 Stonepool       235810     693764. 56007     23.8        2.94
## 10     669 Pineton         237290     350202.  8380      3.53       1.48
## # i 690 more rows
```

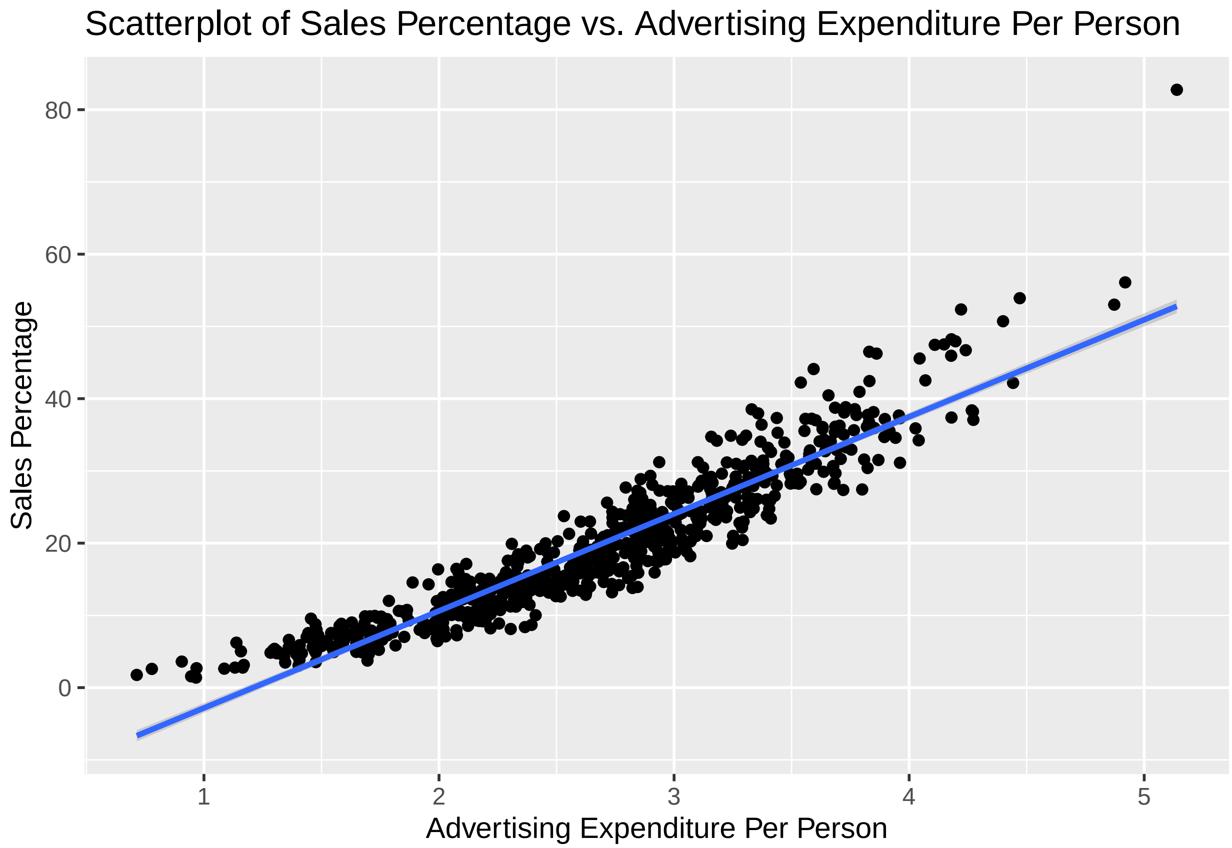
Q7. Producing summary statistics

```
inspect_num(sampled_data)
```

```
## # A tibble: 6 x 10
##   col_name       min      q1 median   mean      q3     max      sd pcnt_na hist
##   <chr>        <dbl>   <dbl>  <dbl>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <named >
## 1 row_num    2    e+0 2.43e2 5.07e2 5.05e2 7.57e2 1.00e3 2.91e+2       0 <tibble>
## 2 population 6.68e+4 2.09e5 2.93e5 3.04e5 3.92e5 5.74e5 1.21e+5       0 <tibble>
## 3 advertisi~ 1.31e+5 5.12e5 7.59e5 7.99e5 1.06e6 1.75e6 3.57e+5       0 <tibble>
## 4 sales      5.26e+3 3.00e4 5.10e4 5.76e4 8.17e4 1.39e5 3.36e+4       0 <tibble>
## 5 sales_pct  1.39e+0 1.23e1 1.87e1 1.97e1 2.60e1 8.28e1 1.03e+1       0 <tibble>
## 6 adv_exp_pp 7.14e-1 2.19e0 2.73e0 2.68e0 3.15e0 5.14e0 7.19e-1       0 <tibble>
```

Q8. Producing scatterplot

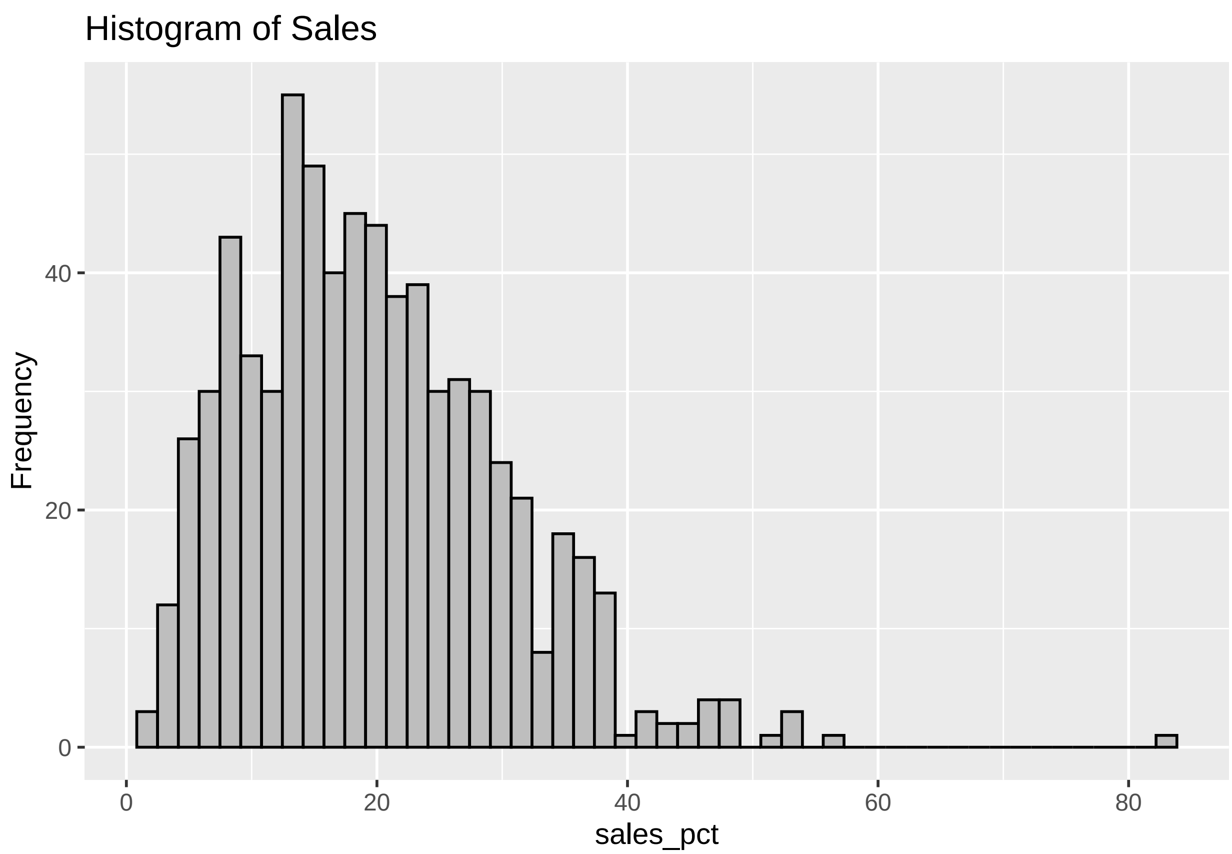
```
ggplot(sampled_data, aes(x = adv_exp_pp, y = sales_pct)) +
  geom_point()+
  geom_smooth(method ="lm") +
  labs(title = "Scatterplot of Sales Percentage vs. Advertising Expenditure Per Person",
       x = "Advertising Expenditure Per Person",
       y = "Sales Percentage")
```



As we can see from the plot, the sales percentage increases when the expenditure on advertisement increase. The best fit line suggests no curvature or strong deviation. This indicates a linear relationship between two variables.

Q9. Producing histogram

```
ggplot(sampled_data, aes(x = sales_pct)) +
  geom_histogram(bins = 50, fill = "gray", color = "black")+
  labs(x = "sales_pct", y = "Frequency", title = "Histogram of Sales")
```




```
skewness_sales_pct <- skewness(sampled_data$sales_pct)
print(skewness_sales_pct)
```

```
## [1] 0.888268
```

The data does not look like a standard normal distribution as it has a positive skewness of 0.888 which indicates a right skewed distribution. To be a normal distribution the plot should be symmetric to the mean and skewness be closer to 0.

Q10. Applying Box-Cox transformation

(a) Finding lambda value

```
box_cox  <- BoxCoxTrans(sampled_data$sales_pct)
box_cox$lambda
```

```
## [1] 0.5
```

(b) Apply the transformation to create a new column

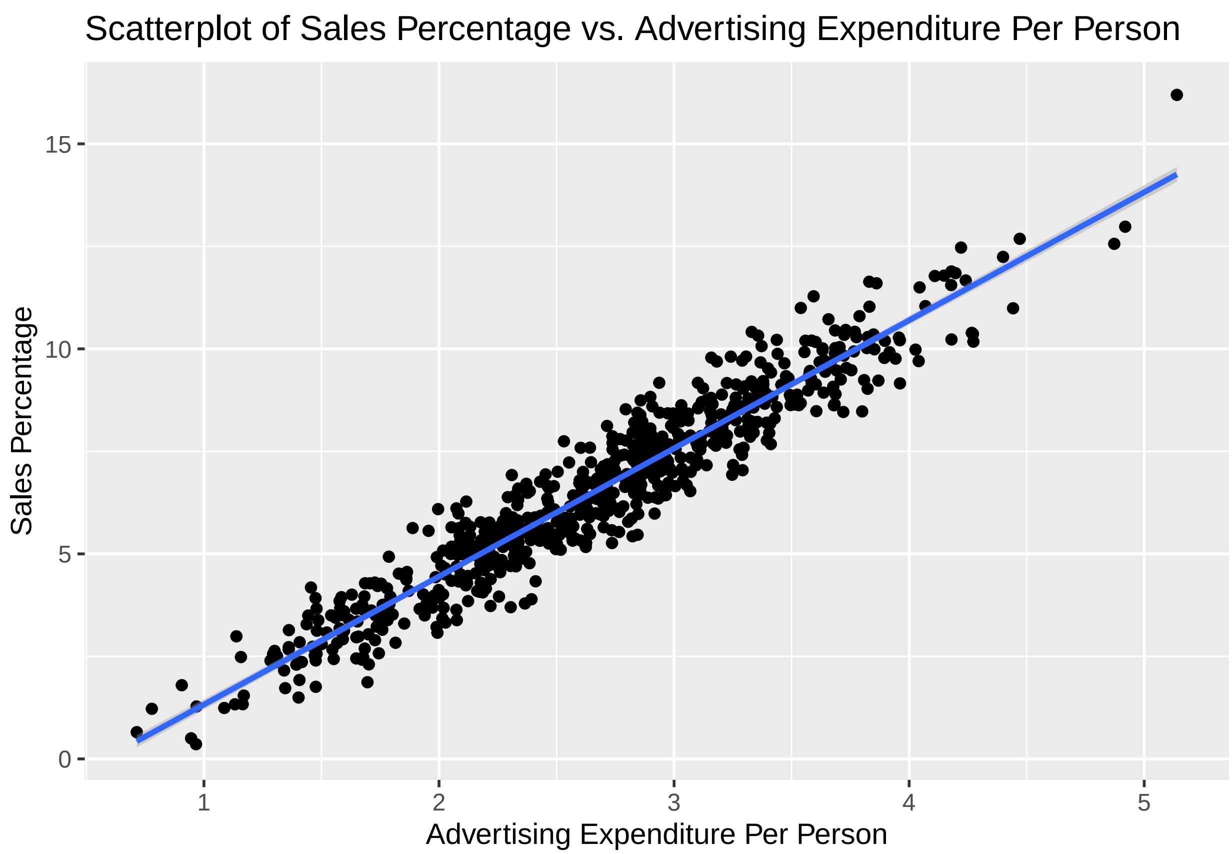
```r
sampled_data <- sampled_data %>%
  mutate(sales_pct_trans = predict(box_cox, sampled_data$sales_pct))
sampled_data
```

```
## # A tibble: 700 x 8
##    row_num name        population advertising sales sales_pct adv_exp_pp
##      <int> <chr>            <int>       <dbl> <int>     <dbl>      <dbl>
##  1     875 Aubury           66832     212081. 16459     24.6        3.17
##  2     739 Princeville     181109     472336. 33356     18.4        2.61
##  3     466 Summerbury      337958     789674. 58494     17.3        2.34
##  4     997 Faybury         193460     562506. 54317     28.1        2.91
##  5      75 Kettletown      206810     500767. 30020     14.5        2.42
##  6     474 Sagefield       201878     573511. 37479     18.6        2.84
##  7     103 Passview        229325     835440. 75046     32.7        3.64
##  8     709 Frostworth      150259     481705. 38772     25.8        3.21
##  9     318 Stonepool       235810     693764. 56007     23.8        2.94
## 10     669 Pineton         237290     350202.  8380      3.53       1.48
## # i 690 more rows
## # i 1 more variable: sales_pct_trans <dbl>
```

Q11. Producing scatterplot and histogram of the transformed data

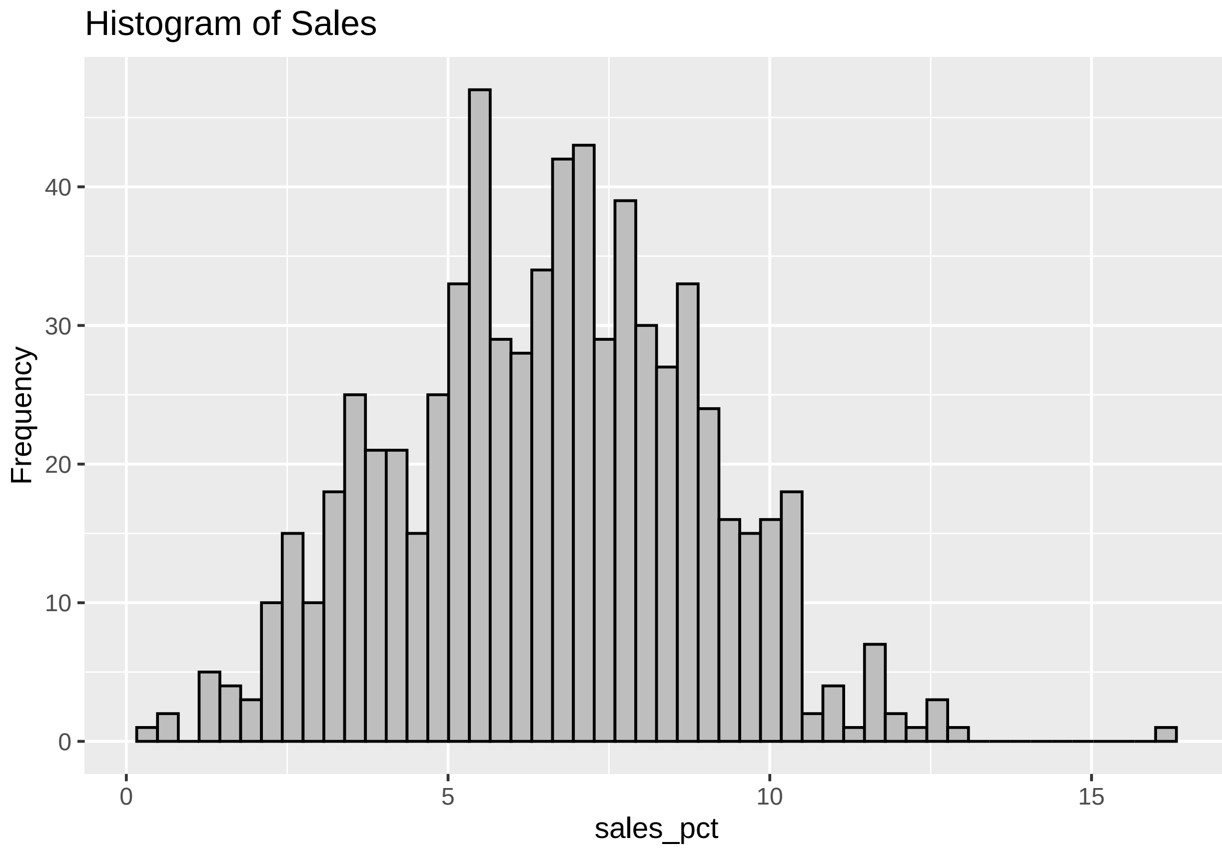
Scatterplot

```r
ggplot(sampled_data, aes(x = adv_exp_pp, y = sales_pct_trans)) +
  geom_point()+
  geom_smooth(method ="lm") +
  labs(title = "Scatterplot of Sales Percentage vs. Advertising Expenditure Per Person",
       x = "Advertising Expenditure Per Person",
       y = "Sales Percentage")
```



Histogram

```
ggplot(sampled_data, aes(x = sales_pct_trans)) +
  geom_histogram(bins = 50, fill = "gray", color = "black")+
  labs(x = "sales_pct", y = "Frequency", title = "Histogram of Sales")
```




```r
skewness_sales_pct_trans <- skewness(sampled_data$sales_pct_trans)
print(skewness_sales_pct_trans)
```

```
## [1] 0.05767271
```

The scatterplot shows increase in sales percentage with the increase of advertise expenditure along the line of best fit. This suggest strong linear relationship between these two variables. The deviation of the data points from the line is decreased after the transformation, suggesting a more linear relationship.

The skewness of the histogram is decreased now which suggests a slightly left skewed distribution with a skewness of 0.0577. This is very close to 0. Therefore, we can say that this a very close to a standard normal distribution.

With these transformation, the data is now more suitable for linear modeling.

Q12. Finding general equations and building linear model

(a) General equation of a linear model for the transformed data

The general equation for a simple linear model is:

$y = \beta_0 + \beta_1 X + \epsilon$ where, y is the response variable, x is the explanatory variable β_0 is the y-intercept of the regression line, β_1 is the slope of the regression line, and ϵ is the error term

For our transformed data, 'sales__pct__trans' is our response variable and 'adv__exp__pp' is our explanatory variable. Therefore, the equation is -

sales_pct_trans = β_0 + β_1(adv_exp_pp) + ϵ

(a) Formula for the line of best fit

The formula for the line of best fit is - $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 X$

Where, $\hat{y}$ is the predicted value of the response variable based on the regression line. $\hat{\beta}_0$ is the estimated y-intercept and $\hat{\beta}_1$ is the estimated slope of the regression line. x is the value of the predictor variable.

sales_pct_trans_pred = $\hat{\beta}_0$ + $\hat{\beta}_1$(adv_exp_pp)

(c) Building the linear model

```
lm_model <- lm(sales_pct_trans ~ adv_exp_pp, data = sampled_data)

# Summarising to get the coefficients
model_summary <- summary(lm_model)
model_summary
```

```
##
## Call:
## lm(formula = sales_pct_trans ~ adv_exp_pp, data = sampled_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.80083 -0.46251  0.00545  0.46168  1.93917
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.79806    0.10015  -17.95   <2e-16 ***
## adv_exp_pp   3.12368    0.03614   86.42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6868 on 698 degrees of freedom
## Multiple R-squared:  0.9145, Adjusted R-squared:  0.9144
## F-statistic:  7469 on 1 and 698 DF,  p-value: < 2.2e-16
```

From the summary we have, $\hat{\beta}_0$ = -1.80 $\hat{\beta}_1$ = 3.12

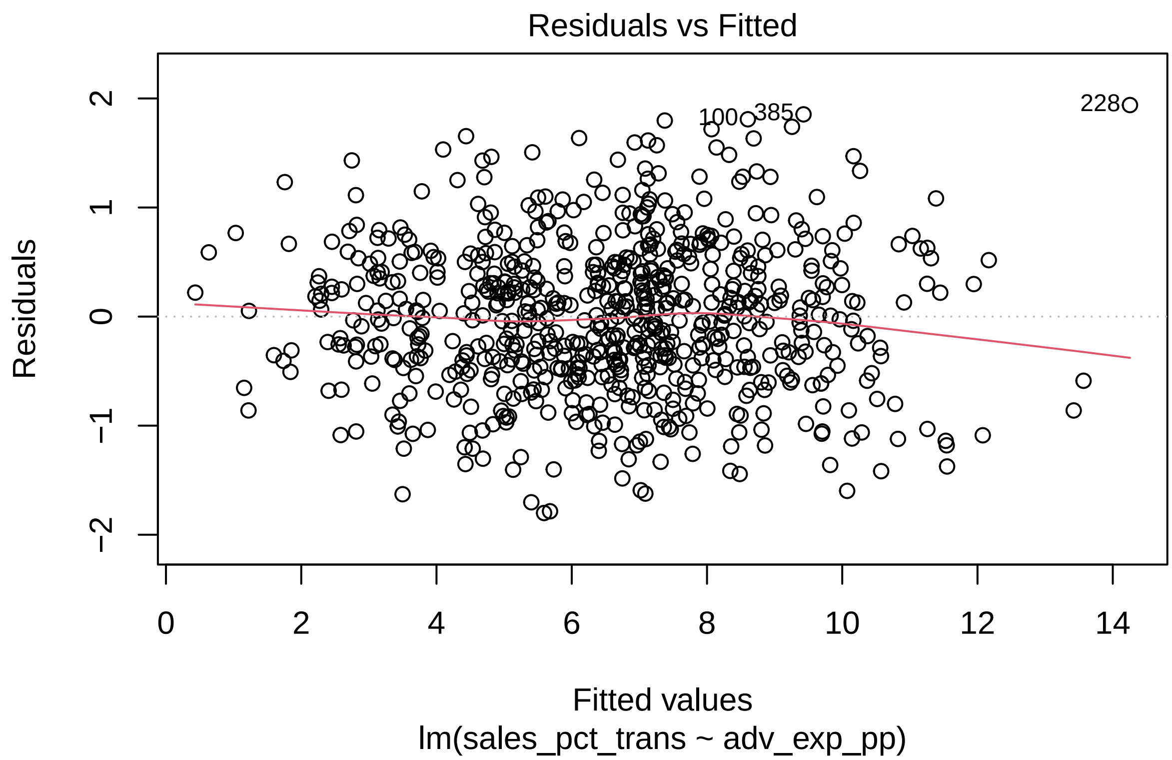
Therefore, our equation of the line becomes,

sales_pct_trans_pred = -1.80 + 3.12(adv_exp_pp)

Q13. Producing scatterplot and histogram of the transformed data

Checking linearity

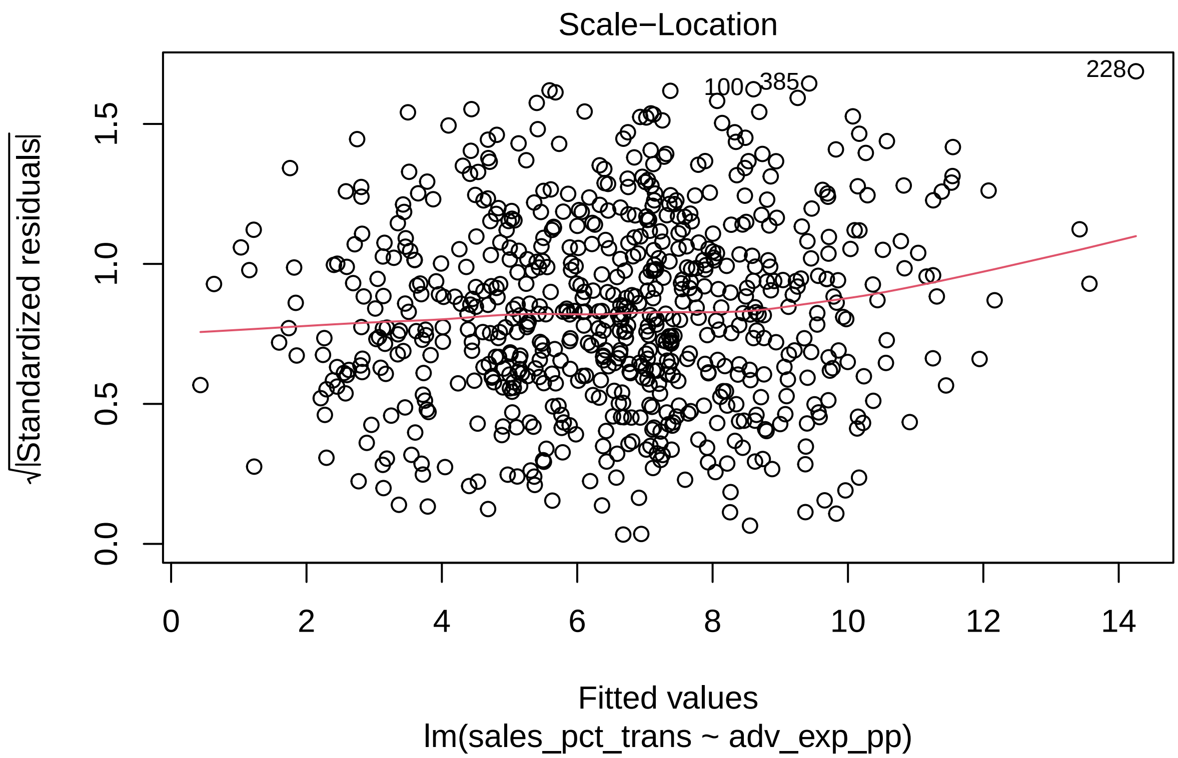
```r
plot(lm_model, which = 1)
```



Findings: We can see a roughly straight (no trends) line which indicates that the assumptions are satisfied.

Checking homoscedasticity

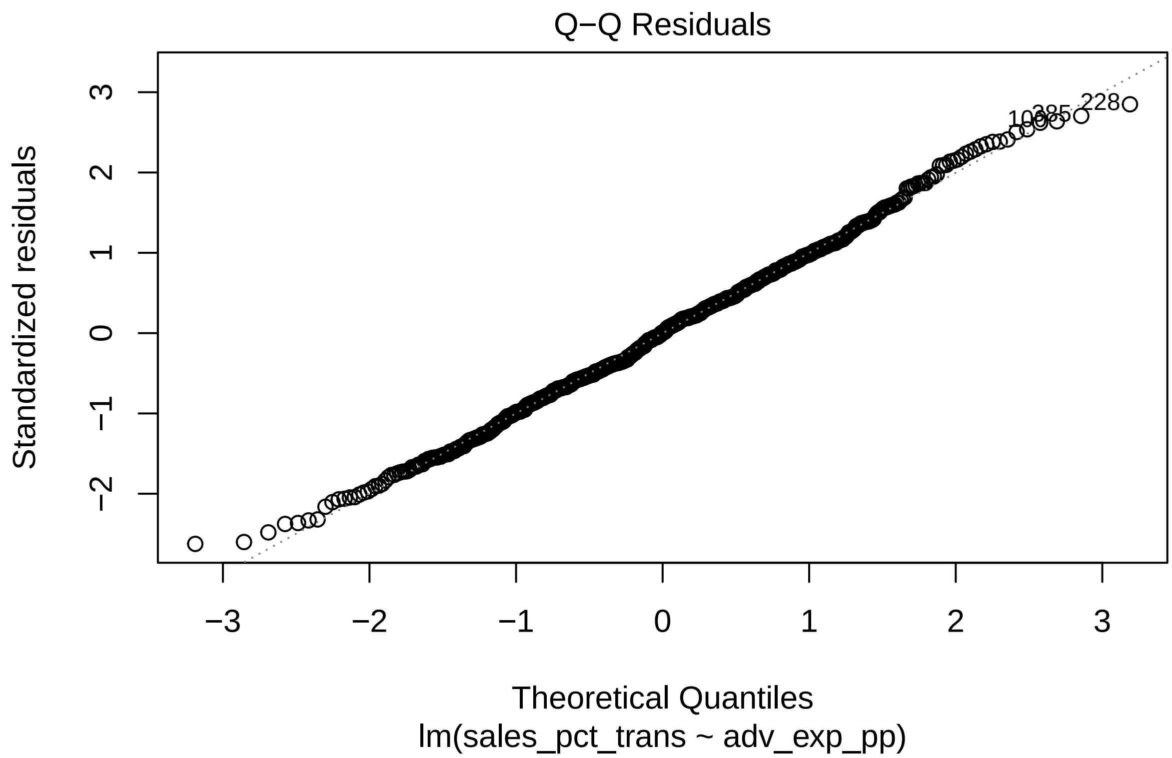
```r
plot(lm_model, which = 3)
```



Findings: Again, We can see a roughly straight and flat line with no trends, which indicates that the model has constant spread.

Checking normality

```
plot(lm_model, which = 2)
```



Findings: From the normal QQ plot, we can notice that points do not drift away from the line within -2 and +2. Therefore, we are confident in our normality assumption as the points between -2 and +2 is our main concern.

Checking independence

One potential problem with independence assumption could be the method of data collection. If the data is collected from closely related cities, their errors could be correlated because of their shared economic conditions or market trends.

Q14. Predicting the percentage of a city's population that will buy a Gadget 2® the given scenarios

```
new_data  <- tibble(
  adv_exp_pp = c(0.05, 3.14, 6.00)
)
predictions_transformed <- predict(lm_model, new_data, interval = "prediction", level = 0.90)

predictions_transformed
```

```
##         fit       lwr        upr
## 1 -1.641879 -2.784646 -0.4991118
```

```
## 2  8.010295  6.877937  9.1426536
## 3 16.944023 15.794839 18.0932069
```

```r
# Let's inverse the box-cox transformation to get the original scaled data for appropriate prediction

inverse_boxcox <- function(x, lambda)
  if(lambda == 0) exp(x) else (lambda * x + 1)^(1/lambda)

output <- as_tibble(predictions_transformed) %>%
  mutate(
    adv_exp_pp = new_data$adv_exp_pp,
    fit = inverse_boxcox(fit, box_cox$lambda),
    lwr = inverse_boxcox(lwr, box_cox$lambda),
    upr = inverse_boxcox(upr, box_cox$lambda)
  ) %>%
  relocate(adv_exp_pp, .before = fit)
output
```

```
## # A tibble: 3 x 4
##   adv_exp_pp     fit    lwr     upr
##        <dbl>   <dbl>  <dbl>   <dbl>
## 1       0.05  0.0321  0.154   0.563
## 2       3.14 25.1    19.7    31.0
## 3       6    89.7    79.2   101.
```

Q15. Interpretation

For an advertising expenditure of $0.05, the model predicts that only 0.0321% people are expected to make a purchase. However, the fitted value is outside of the prediction interval (0.154% - 0.563%) which suggests a discrepancy. This could have happened due to to the model being extrapolating beyond the range of data it was trained on, or it might indicate that the linear model is not suitable for extremely low levels of advertising expenditure.

The model predicts 25.1% of purchase if the company wishes to spend $3.14 per person. The model is 90% confident that the true mean response is expected to lie within 19.7% - 31.0%. This might help the company to make decision.

Finally, if the company wishes to spend a higher amount of money for the advertising, in our case it was $6.00 per person, the model predicts a significant increase in purchase percentage which is 89.7% with an interval of $79.2 to 101%. The upper interval is exceeding 100% is not possible in the context of percentages. This anomaly suggests that for a high value of adv_exp_pp the model may not be completely reliable.

However, it is clear that a higher investment in advertisement could potentially lead to a higher sales. For practical purposes, we suggest that, the company should focus on moderate or high level of advertising expenditure which could be in between 3.14-6.00 in order to generate great sales.