* [About ANZ](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#about-anz)
* [Required packages](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#required-packages)
* [Executive Summary](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#executive-summary)
* [Dataset Description](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#dataset-description)
* [Understand](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#understand)
* [Tidy & Manipulate Data I](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#tidy-manipulate-data-i)
* [Tidy & Manipulate Data II](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#tidy-manipulate-data-ii)
  + [Creating New Column using setDT](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#creating-new-column-using-setdt)
  + [Creating New Column using Mutate](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#creating-new-column-using-mutate)
* [Scan I](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#scan-i)
  + [Scanning for Missing Values in Txn\_Description Column](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#scanning-for-missing-values-in-txn_description-column)
  + [Mutating the Missing Values](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#mutating-the-missing-values)
  + [Combining new rows with the existed one](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#combining-new-rows-with-the-existed-one)
* [Scan II](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#scan-ii)
  + [Detecting Outliers in Age Column](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#detecting-outliers-in-age-column)
  + [Detecting Outliers in Monthly Salary Column](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#detecting-outliers-in-monthly-salary-column)
  + [Dealing with Outliers in Monthly Salary Column](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#dealing-with-outliers-in-monthly-salary-column)
* [Transform](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#transform)
* [Resources](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#resources)

About ANZ

The Australia and New Zealand Banking Group Limited, commonly called ANZ, is an Australian multinational banking and financial services company headquartered in Melbourne, Australia. It is the second largest bank by assets and thirdlargest bank by market capitalisation in Australia.2

Required packages

**library**(readxl)

**library**(dplyr)

**library**(tidyr)

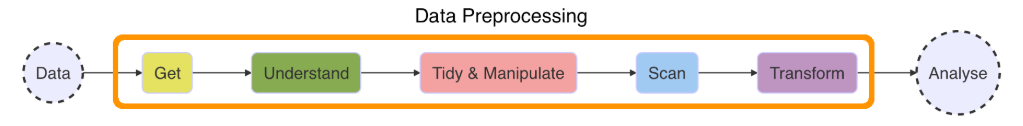
**library**(lubridate)

**library**(data.table)

**library**(outliers)

Executive Summary

In this assignment, applying all the data preprocessing steps were aimed. This process includes all the major tasks which are shown in the figure.



Resource: 3/

The dataset is provided from the ANZ’s Virtual Internship Program / InsideSherpa. The task is in this program, which seems really easy, gathering some interesting overall insights about the data. But the problem in this dataset is there are some data issues, and its needs to be cleaned. In this assignment, the collection of operations will be done to prepare all forms of untidy data (incomplete, noisy and inconsistent data) for statistical analysis.

This operations includes import(Refer.[Dataset Description](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#dataset-description)), manage(Refer.[Understand](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#understand)), manipulate(Refer.[Tidy & Manipulate Data I](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#tidy-manipulate-data-i),[Tidy & Manipulate Data II](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#tidy-manipulate-data-ii),[Scan I](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#scan-i),[Scan II](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#scan-ii)) and transform (Refer.[Transform](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#transform)) your data before performing any kind of data analysis.

The dataset originally has 23 variables and 12043 observations in total. During the assignment, some of columns were deleted and instead new columns were created and mutated. To manupulate the missing values, new dataset was created to make detailed analysis and after all manipulations, the new dataset and the original dataset had been merged. The main problem in this dataset were outliers. For monthly salary, the outliers were handled but even though the visiable bad effect of the outliers in the transaction amounts couldn’t be solved.

Dataset Description

This task is based on a synthesised transaction dataset containing *3 months’* worth of transactions for *100 hypothetical customers*. It contains purchases, recurring transactions, and salary transactions.

The dataset is designed to simulate realistic transaction behaviours that are observed in ANZ’s real transaction data,

ANZ <- read\_excel("C:/Users/serve/Desktop/ANZ/ANZ synthesised transaction dataset (4).xlsx")

head(ANZ)

## # A tibble: 6 x 23

## status card\_present\_fl~ bpay\_biller\_code account currency long\_lat

## <chr> <dbl> <dbl> <chr> <chr> <chr>

## 1 autho~ 1 NA ACC-15~ AUD 153.41 ~

## 2 autho~ 0 NA ACC-15~ AUD 153.41 ~

## 3 autho~ 1 NA ACC-12~ AUD 151.23 ~

## 4 autho~ 1 NA ACC-10~ AUD 153.10 ~

## 5 autho~ 1 NA ACC-15~ AUD 153.41 ~

## 6 posted NA NA ACC-16~ AUD 151.22 ~

## # ... with 17 more variables: txn\_description <chr>, merchant\_id <chr>,

## # merchant\_code <dbl>, first\_name <chr>, balance <dbl>, date <dttm>,

## # gender <chr>, age <dbl>, merchant\_suburb <chr>, merchant\_state <chr>,

## # extraction <chr>, amount <dbl>, transaction\_id <chr>, country <chr>,

## # customer\_id <chr>, merchant\_long\_lat <chr>, movement <chr>

dim(ANZ)

## [1] 12043 23

In the ANZ dataset, there are 23 different columns and 12043 observations.

To see the column names;

colnames(ANZ)

## [1] "status" "card\_present\_flag" "bpay\_biller\_code"

## [4] "account" "currency" "long\_lat"

## [7] "txn\_description" "merchant\_id" "merchant\_code"

## [10] "first\_name" "balance" "date"

## [13] "gender" "age" "merchant\_suburb"

## [16] "merchant\_state" "extraction" "amount"

## [19] "transaction\_id" "country" "customer\_id"

## [22] "merchant\_long\_lat" "movement"

The column descriptions;

* **status** stands for the way how the transactions happened as in authorized or posted.
* **card\_present\_flag** stands for a card not present transaction (CNP, MO/TO, Mail Order / Telephone Order, MOTOEC) is a paymentcard transaction made where the cardholder does not or cannot physically present the card for a merchant’s visual examination at the time that an order is given and payment effected.

summary(ANZ$card\_present\_flag)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

## 0.000 1.000 1.000 0.803 1.000 1.000 4326

As it seen above, this column has 0’s for which means the cards were not with the customers and 1’s stands for the cards were with the customers while they were doing their transaction. Also, this column has lots of missing values(4326) because of that this column will not be included for further analysis.

* **bpay\_biller\_code** stands for bill payments which are might be directly through the internet, mobile or phone banking. Customer has unique BPAY Biller Code, and this feature can be used with that code and Customers’ Reference Number to pay.

summary(ANZ$bpay\_biller\_code)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

## 0 0 0 0 0 0 11160

As it seen above, in this dataset either the customers doesnt have Bpay feature in their account or this information is missing. Because of these reasons, this column will not be used in the next steps.

* **account** is a unique code for each account.
* **currency** which is all the transactions used as a currency. For all the dataset the currency is AUD. Thats why the currency column will not be used for further analysis.
* **long\_lat** stands for where the customers live as in longitudes and latitudes.
* **txn\_description** stands for which way the transactions occured. For example, Inter Bank, Phone Bank etc.
* **merchant\_id** stands for a unique code given to a business by payment processors before a merchants begin processing credit cards.
* **merchant\_code** stands for a four-digit number listed in ISO 18245 for retail financial services.

summary(ANZ$merchant\_code)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

## 0 0 0 0 0 0 11160

As it seen above, in this dataset either the merchant code is 0 ,which is not possible, or this information is missing. Because of these reasons, this column will not be used in the next steps.

* **first\_name** stands for Customers’ first names.
* **balance** stands for how much money customers’ have in their account.
* **date** stands for which date the transaction occured.
* **gender** is Customers’ genders.
* **age** is Customers’ ages.
* **merchant** suburb stands for where the transaction occured as in suburbs.
* **merchant\_state** stands for where the transaction occured as in states.
* **extraction** stands for when the transactions occured, the date with the time(hms)
* **amount** stands for how much monet the customers spent in each transactions
* **transaction\_id** is a unique code for each transactions.
* **country** stands for where all the transactions occured. For all dataset the country is Australia. Thats why country column will not be in the next steps.
* **customer\_id** is a unique code for each customers.
* **merchant\_long\_lat** stands for where the transactions occured as in longitudes and latitudes.
* **movement** stands for the type of the account as in debit or credit card.

Understand

In this section, the types of the variables will be summarised. Before explaining the data structures, the new dataset will be created using one of the dplyr package function, also, some of the variables will be factorised and labelled.

The dplyr package is regarded as the “Grammar of Data Manipulation” in R and it originates from the popular plyr package, also developed by Hadley Wickham. 4

When working with a large data frame, often we want to only assess specific variables. The select() function allows us to select and/or rename variables.

ANZ1 <- ANZ %>% select(-currency, -country, -bpay\_biller\_code,-card\_present\_flag, -merchant\_code)

dim(ANZ1)

## [1] 12043 18

After eleminate the columns, the new dataset is called ANZ1. This dataset has 18 columns and 12043 rows.

summary(ANZ1)

## status account long\_lat

## Length:12043 Length:12043 Length:12043

## Class :character Class :character Class :character

## Mode :character Mode :character Mode :character

##

##

##

## txn\_description merchant\_id first\_name

## Length:12043 Length:12043 Length:12043

## Class :character Class :character Class :character

## Mode :character Mode :character Mode :character

##

##

##

## balance date gender

## Min. : 0.24 Min. :2018-08-01 00:00:00 Length:12043

## 1st Qu.: 3158.59 1st Qu.:2018-08-24 00:00:00 Class :character

## Median : 6432.01 Median :2018-09-16 00:00:00 Mode :character

## Mean : 14704.20 Mean :2018-09-15 21:27:39

## 3rd Qu.: 12465.94 3rd Qu.:2018-10-09 00:00:00

## Max. :267128.52 Max. :2018-10-31 00:00:00

## age merchant\_suburb merchant\_state extraction

## Min. :18.00 Length:12043 Length:12043 Length:12043

## 1st Qu.:22.00 Class :character Class :character Class :character

## Median :28.00 Mode :character Mode :character Mode :character

## Mean :30.58

## 3rd Qu.:38.00

## Max. :78.00

## amount transaction\_id customer\_id

## Min. : 0.10 Length:12043 Length:12043

## 1st Qu.: 16.00 Class :character Class :character

## Median : 29.00 Mode :character Mode :character

## Mean : 187.93

## 3rd Qu.: 53.66

## Max. :8835.98

## merchant\_long\_lat movement

## Length:12043 Length:12043

## Class :character Class :character

## Mode :character Mode :character

##

##

##

At first glance, we can see classes of the all the columns.

* The Status column needs to be ‘factor’ to make the analysis easier.

The function factor is used to encode a vector as a factor (the terms ‘category’ and ‘enumerated type’ are also used for factors). If argument ordered is TRUE, the factor levels are assumed to be ordered. 5

ANZ1$status <- factor(ANZ1$status, levels = c("authorized","posted"))

*Log\_lat column needs to be separated because of the tidy dataset rules. But in the next steps it will be handled.* (Refer.[Tidy & Manipulate Data I](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#tidy-manipulate-data-i))

* The txn\_description column needs to be factor to make the analysis easier.

ANZ1$txn\_description <- factor(ANZ1$txn\_description, levels = c("INTER BANK", "PAY/SALARY",

"PAYMENT","PHONEBANK","POS","SALES-POS"))

* The Balance column is already numeric, no need to change it.
* The Date column is imported as a numeric but it needs to converted to date using *lubridate* package *ymd()* function.

ANZ1$date <- ymd(ANZ1$date)

class(ANZ1$date)

## [1] "Date"

* The gender column needs to be factor.

ANZ1$gender <- factor(ANZ1$gender, levels=c("M","F"))

* As it known, the *Extraction* columns stands for the date and hour information for each transaction.

**To understand what type of data extraction column has,**

ANZ$extraction[1:6]

## [1] "2018-08-01T01:01:15.000+0000" "2018-08-01T01:13:45.000+0000"

## [3] "2018-08-01T01:26:15.000+0000" "2018-08-01T01:38:45.000+0000"

## [5] "2018-08-01T01:51:15.000+0000" "2018-08-01T02:00:00.000+0000"

ANZ1$extraction <- ymd\_hms(ANZ1$extraction)

class(ANZ1$extraction)

## [1] "POSIXct" "POSIXt"

ANZ1$extraction[1:6]

## [1] "2018-08-01 01:01:15 UTC" "2018-08-01 01:13:45 UTC"

## [3] "2018-08-01 01:26:15 UTC" "2018-08-01 01:38:45 UTC"

## [5] "2018-08-01 01:51:15 UTC" "2018-08-01 02:00:00 UTC"

As it seen above, *ymd\_hms* function has worked very well on the column.

*merchant\_long\_lat column needs to be separated, it will be handled in the next steps. (Refer.*[*Tidy & Manipulate Data I*](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#tidy-manipulate-data-i)*)*

* The movement column needs to be factor.

ANZ1$movement <- factor(ANZ1$movement, levels=c("credit","debit"))

Tidy & Manipulate Data I

In this section, the scope of the tidy data principles, the dataset will be examined. For manupulations the *Tidyr* package will be used.

Most messy datasets can be tidied with a small set of tools. The tidyr package is a very useful package that reshapes the layout of data sets.3

It is often said that 80% of data analysis is spent on the process of cleaning and preparing the data (Dasu and Johnson 2003). Data preparation is not just a first step, but must be repeated many over the course of analysis as new problems come to light or new data is collected. 6

There are three interrelated rules which make a dataset tidy:

* Each variable must have its own column.
* Each observation must have its own row.
* Each value must have its own cell. 7

**Long\_lat and Merchant long\_lat information keeps in the same column it violates the rule which is stated above.**

ANZ$long\_lat[1:6]

## [1] "153.41 -27.95" "153.41 -27.95" "151.23 -33.94" "153.10 -27.66"

## [5] "153.41 -27.95" "151.22 -33.87"

ANZ$merchant\_long\_lat[1:6]

## [1] "153.38 -27.99" "151.21 -33.87" "151.21 -33.87" "153.05 -26.68"

## [5] "153.44 -28.06" NA

separate() pulls apart one column into multiple columns, by splitting wherever a separator character appears.

ANZ1 <- ANZ1 %>% separate(long\_lat, into = c("longitude","latitude"), sep=6)

ANZ1 <- ANZ1 %>% separate(merchant\_long\_lat, into = c("m\_longitude","m\_latitude"), sep=6)

head(ANZ1$longitude)

## [1] "153.41" "153.41" "151.23" "153.10" "153.41" "151.22"

head(ANZ1$latitude)

## [1] " -27.95" " -27.95" " -33.94" " -27.66" " -27.95" " -33.87"

head(ANZ1$m\_latitude)

## [1] " -27.99" " -33.87" " -33.87" " -26.68" " -28.06" NA

head(ANZ1$m\_longitude)

## [1] "153.38" "151.21" "151.21" "153.05" "153.44" NA

As it seen above, separate function had made this column to store only one observation in each cell.

Tidy & Manipulate Data II

In R, there are multiple way to create a new column, in this section it will shown how to create new columns using setDT function and using mutate funtion with conditions.

Creating New Column using setDT

setDT helps to retains the new data.frame’s row names under a new column.

In this step, to create a new column for age ranges, setTD function will be used. For this step, first of all, agebreaks and agelabels were created and in setTD function, under the agegroup column name, these vectors were merged.

agebreaks <- c(0,1,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,500)

agelabels <- c("0-1","1-4","5-9","10-14","15-19","20-24","25-29","30-34",

"35-39","40-44","45-49","50-54","55-59","60-64","65-69",

"70-74","75-79","80-84","85+")

setDT(ANZ1)[ , agegroups := cut(age,

breaks = agebreaks,

right = FALSE,

labels = agelabels)]

head(ANZ1$agegroups)

## [1] 25-29 25-29 35-39 40-44 25-29 20-24

## 19 Levels: 0-1 1-4 5-9 10-14 15-19 20-24 25-29 30-34 35-39 40-44 ... 85+

Creating New Column using Mutate

In this step, transaction column will be created. This column has all the transaction includes Internet Banking, Paymnets, Phone Banking, POS and SALES-POS transactions. In this command, the column will be created using ifelse statement.

mutate + ifelse helps to create new conditional variables.

ANZ1<- mutate(ANZ1,transaction= ifelse(txn\_description=="INTER BANK" | txn\_description=="PAYMENT" |

txn\_description=="PHONE BANK" | txn\_description=="POS" |

txn\_description == "SALES-POS", ANZ1$amount, 0))

Normally, Transaction volume is calculation based on the aggregate number of transactions (inclusive of credit, debit and prepaid) from a merchant. In cases where a merchant corporation has more than one DBA, merchant banks must consider the aggregate volume of transactions stored, processed or transmitted by the corporate entity to determine the validation level. Credit card authorizations would be considered as transactions processed and therefore count toward determining your transaction totals. 8

But in this assignment, to classify the transaction volumes, the basic summary statistics will be used.

summary(ANZ1$transaction)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

## 0.00 12.77 24.21 48.23 42.78 7081.09 101

IQR = 42.78 - 12.77

UpperLimit = 1.5 \*(IQR) + 42.78

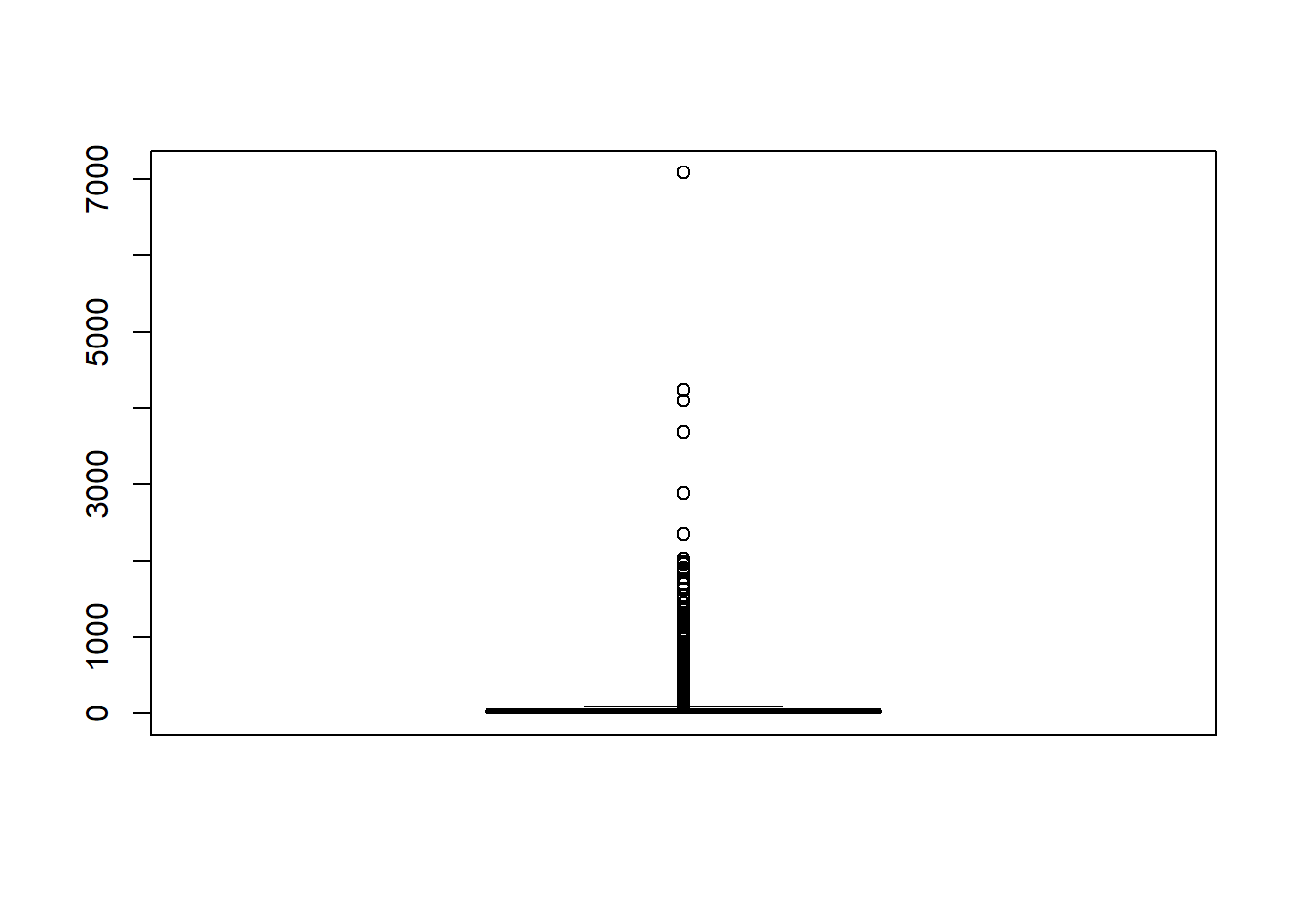
UpperLimit

## [1] 87.795

To classify the transaction volumes,

* “Low” category’s limit is: 0 - 12.77(which is 1. Quartile)
* “Low-Medium” category’s limit is: 12.77(which is 1. Quartile) - 24.21(which is median)
* “Medium” category’s limit is: 24.21(which is median) - 48.23 (which is mean)
* “Medium - High” category’s limit is: 48.23 (which is mean) - 87.79(Which is upper limit)
* “High” category’s limit is: 87.79(Which is upper limit) - 7081.09(max value)

boxplot(ANZ1$transaction)



This classification doesnt make sence in some cases, because it has many outliers in it.

ANZ1 <- ANZ1 %>%

mutate(transaction\_volume =

case\_when(transaction >= 0 & transaction <= 12.87 ~ "Low",

transaction > 12.87 & transaction <= 24.44 ~ "Low-Medium",

transaction > 24.44 & transaction <= 48.72 ~ "Medium",

transaction > 48.72 & transaction <= 94.495 ~ "Medium-High",

transaction > 94.495 & transaction <= 7081.09 ~ "High"))

ANZ1$transaction\_volume <- factor(ANZ1$transaction\_volume)

summary(ANZ1$transaction\_volume)

## High Low Low-Medium Medium Medium-High NA's

## 1093 3017 2993 3522 1317 101

As it seen there are 101 missing values, and they will be handled in the next steps. (Refer. [Scan])

* To analyse the customers’ salaries,

**The salary table will be created using group by, filter, mutate, and select functions.**

Salary\_Table <- ANZ1 %>%

group\_by(customer\_id) %>%

filter(txn\_description=="PAY/SALARY") %>%

mutate(monthlysalary = (sum(amount))/3) %>%

select(account, customer\_id, merchant\_id, txn\_description,

first\_name, gender,balance, date, age, merchant\_suburb,

amount, agegroups, monthlysalary, transaction)

To see the months clearly,

Salary\_Table <- Salary\_Table %>%

mutate(month =

case\_when(date >= "2018-08-01" & date <= "2018-08-31" ~ "August",

date >= "2018-09-01" & date <= "2018-09-30" ~ "September",

date >= "2018-10-01" & date <= "2018-10-31" ~ "October"))

Salary\_Table$month <- factor(Salary\_Table$month)

This kind of manipulations help to make further analysis easier. For instance, to check where there is a correlation between months and salaries, or age and salaries.

Scan I

In this section, the missing values will be scanned, and dealt by using filtering and capping method.

Scanning for Missing Values in Txn\_Description Column

sum(is.na(ANZ1$txn\_description))

## [1] 101

As it seen above, there are 101 values which are NA.

The new dataset will be created to give a detailed look, this dataset will be used missing data mutations.

Missing\_Txn <- ANZ1 %>% filter(is.na(txn\_description))

dim(Missing\_Txn)

## [1] 101 23

Mutating the Missing Values

After creating another table called Missing\_Txn, helped to examine which transactions has missing description. it is concluded that for 17 customers there are 101 transactions which are not labeled. It is conducted that 13 customers has monthly or weekly these transactions because of that these transactions will be labeled as *‘PaymentT’*. The other accounts which are respectively, ACC-966140392(which is 68th row), ACC-3536132544(which is 18th row), ACC-90814749(which is 58th row) will be labeled as a *possible fraud risk*.

Missing\_Txn$txn\_description <- as.character(Missing\_Txn$txn\_description)

Missing\_Txn[68,] <- Missing\_Txn[68,] %>% mutate(txn\_description = replace(txn\_description,

is.na(txn\_description),

"PossibleFroud"))

Missing\_Txn[18,] <- Missing\_Txn[68,] %>% mutate(txn\_description = replace(txn\_description,

is.na(txn\_description),

"PossibleFroud"))

Missing\_Txn[58,] <- Missing\_Txn[68,] %>% mutate(txn\_description = replace(txn\_description,

is.na(txn\_description),

"PossibleFroud"))

Missing\_Txn <- Missing\_Txn %>% mutate(txn\_description = replace(txn\_description,

is.na(txn\_description),

"PaymentT"))

table(Missing\_Txn$txn\_description)

##

## PaymentT PossibleFroud

## 98 3

sum(is.na(Missing\_Txn$txn\_description))

## [1] 0

As it seen above, there is no missing value anymore.

Combining new rows with the existed one

Last of all, Missing\_Txn table and ANZ1 table will be combined. But before combination, the null txn\_description rows will be filtered and instead of that column, Missing\_Txn table will be inserted.

ANZ2 <- ANZ1 %>% filter(!is.na(ANZ1$txn\_description))

ANZ3 <- bind\_rows(ANZ2, Missing\_Txn)

table(ANZ3$txn\_description)

##

## INTER BANK PAY/SALARY PAYMENT PaymentT POS

## 742 883 2600 98 3783

## PossibleFroud SALES-POS

## 3 3934

Scan II

In this section, the dataset will be scanned for outliers.

Detecting Outliers in Age Column

summary(Salary\_Table$age)

## Min. 1st Qu. Median Mean 3rd Qu. Max.

## 18.00 22.00 30.00 32.63 40.00 78.00

IQR\_Age = 40 - 22

UpperLimit\_Age = 1.5 \*(IQR) + 22

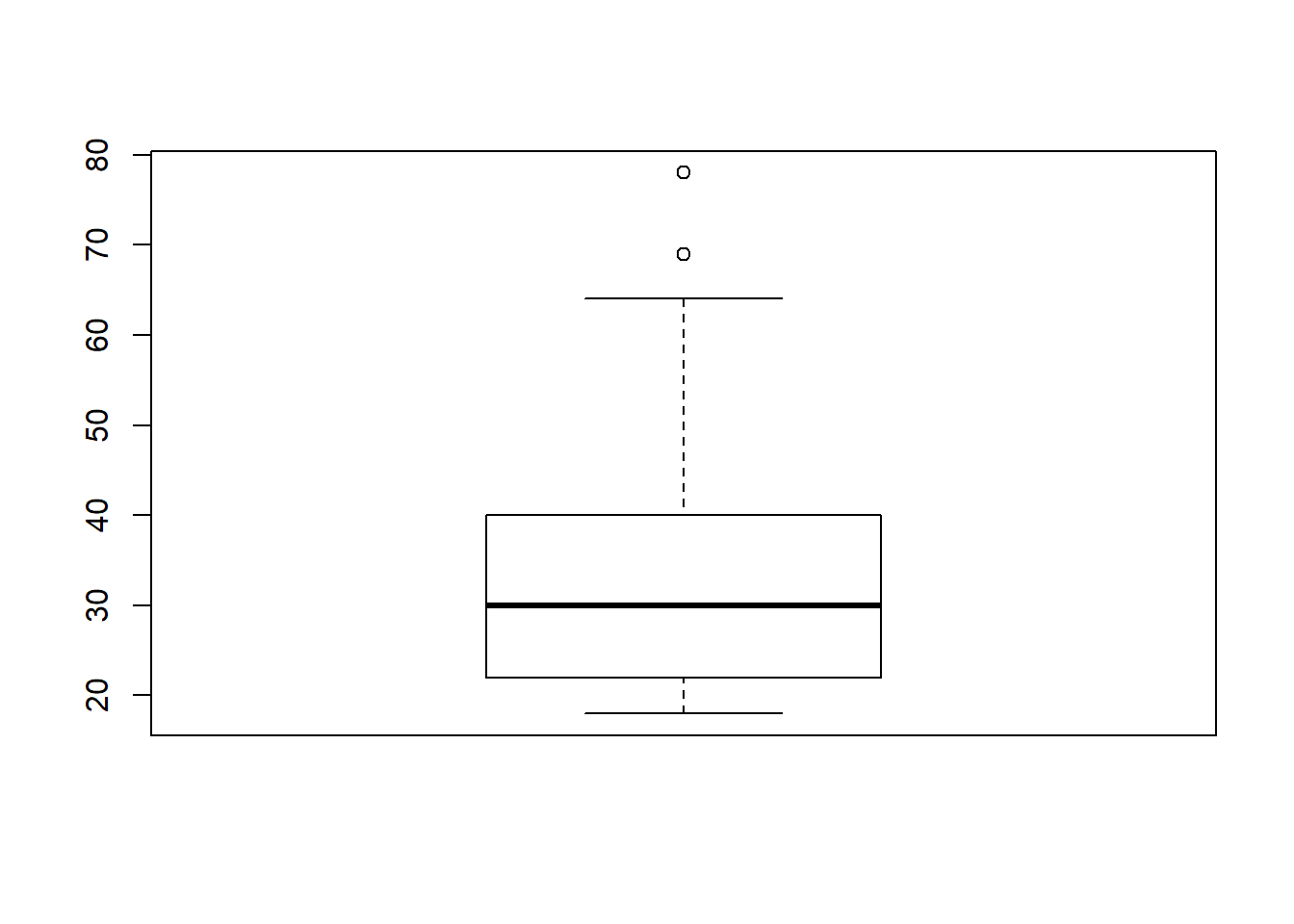
UpperLimit\_Age

## [1] 67.015

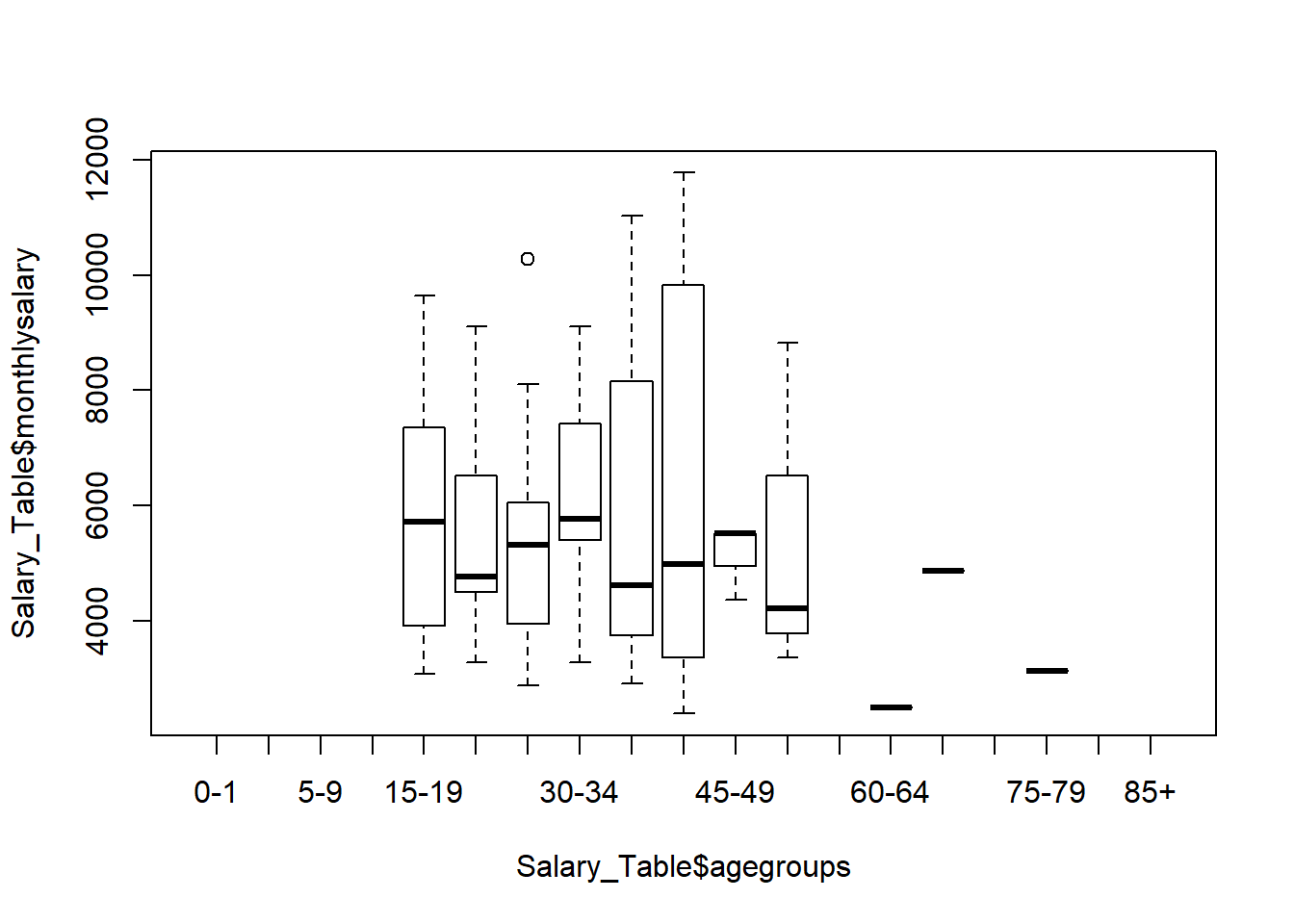
As it seen above, after age 67, the box plot is going to show as a outlier. But in the real world, the bank can have customers who are above 67 years old, hence, any manipulation will be used to deal with them.

To visualise the boxplot,

boxplot(Salary\_Table$age)

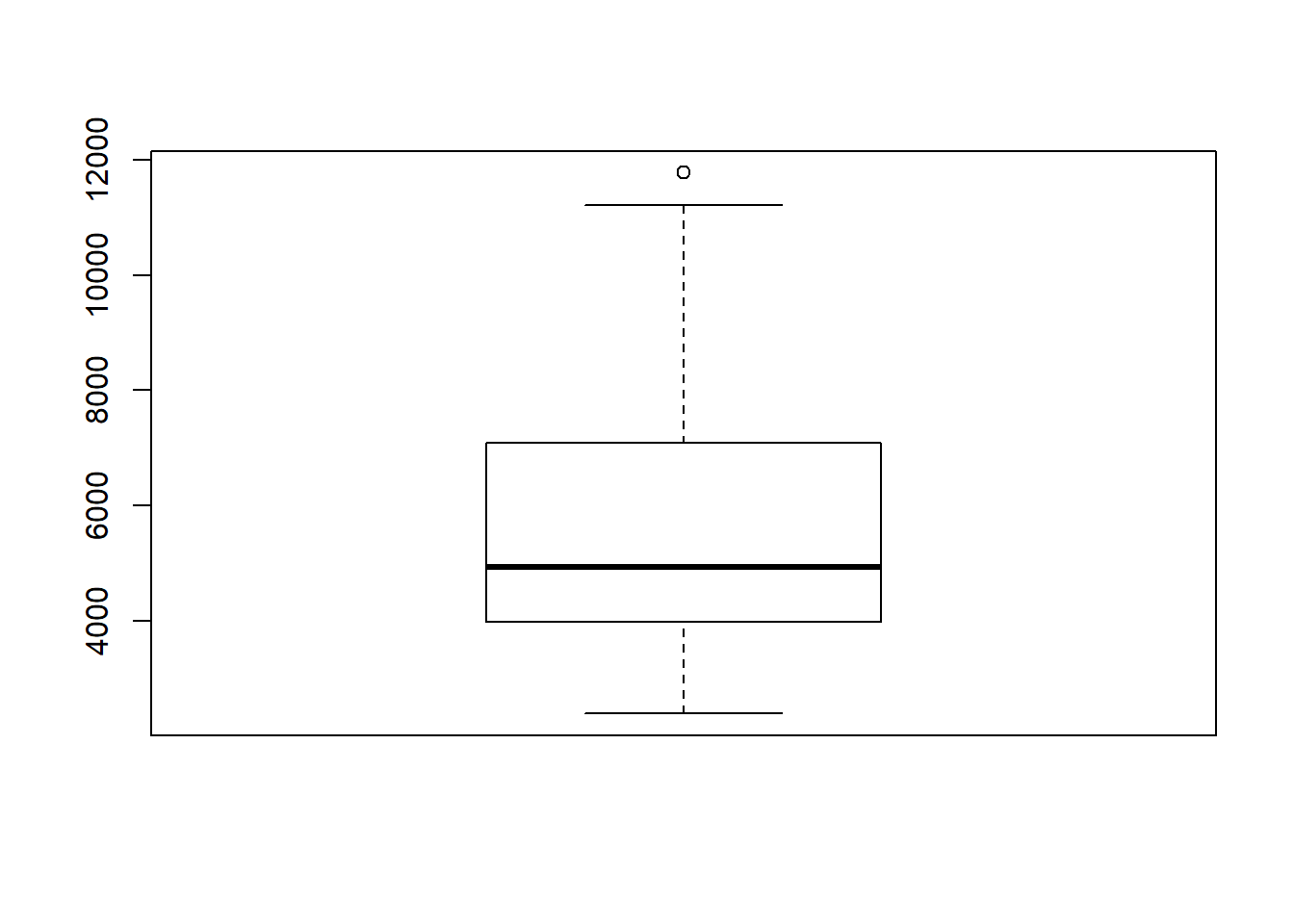


plot(Salary\_Table$monthlysalary~Salary\_Table$agegroups)



Detecting Outliers in Monthly Salary Column

boxplot(Salary\_Table$monthlysalary)



Dealing with Outliers in Monthly Salary Column

cap <- **function**(x){

quantiles <- quantile( x, c(.05, 0.25, 0.75, .95 ) )

x[ x < quantiles[2] - 1.5\*IQR(x) ] <- quantiles[1]

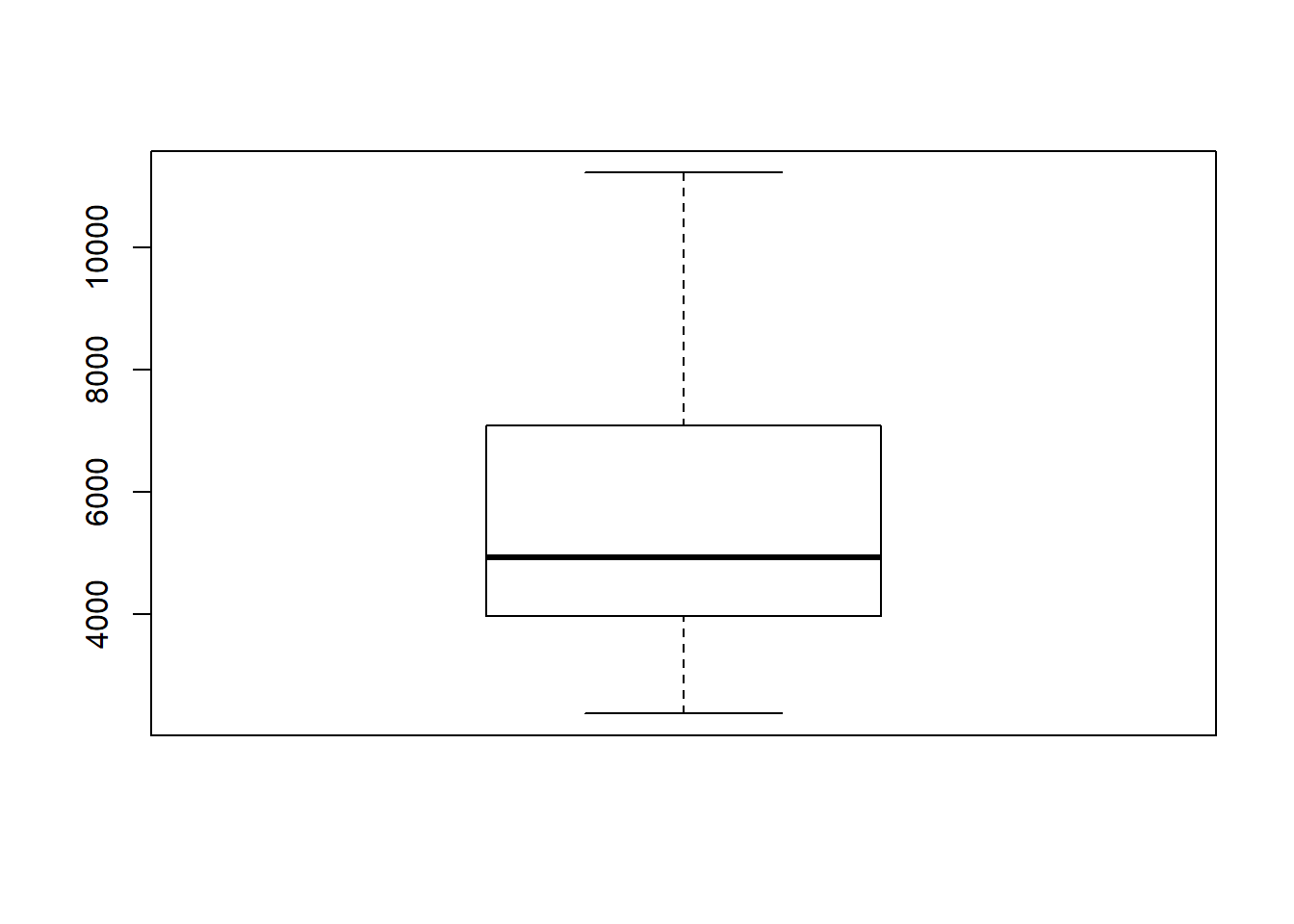
x[ x > quantiles[3] + 1.5\*IQR(x) ] <- quantiles[4]

x

}

Salary\_Table$monthlysalary <- Salary\_Table$monthlysalary %>% cap()

boxplot(Salary\_Table$monthlysalary)



Transform

The assumption of normality is important for hypothesis testing and in regression models. In general linear models, the assumption comes in to play with regards to residuals (aka errors). In both cases it is useful to test for normality. 9

In this section, for the salary information, different method of the normalisations will be applied. They will be compared using histogram plots.

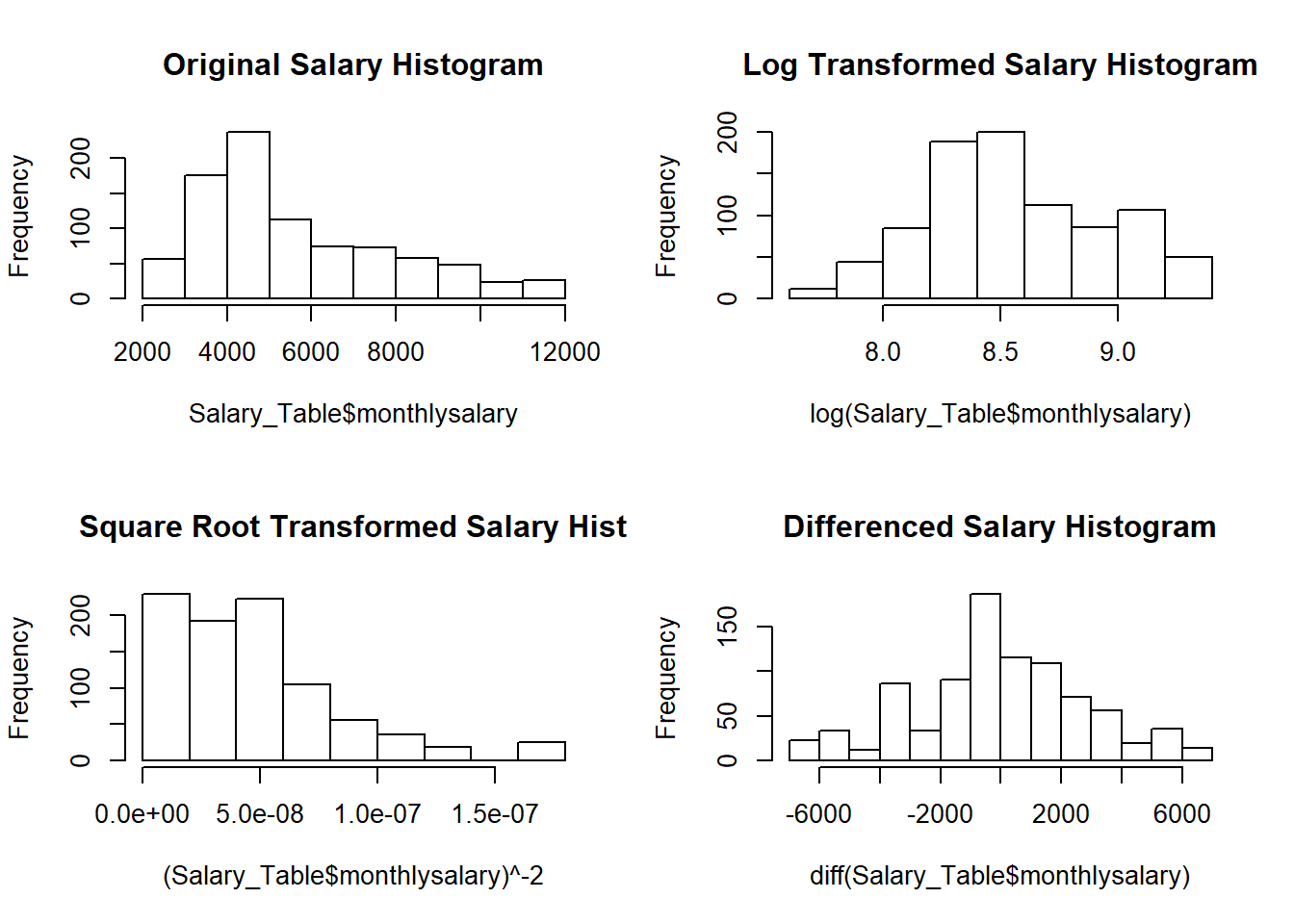
par(mfrow=c(2,2))

hist(Salary\_Table$monthlysalary, main="Original Salary Histogram")

hist(log(Salary\_Table$monthlysalary), main="Log Transformed Salary Histogram")

hist((Salary\_Table$monthlysalary)^-2, main="Square Root Transformed Salary Hist")

hist(diff(Salary\_Table$monthlysalary), main="Differenced Salary Histogram")



As it seen above, differencing helped the age distribution to distribute normally. After that assumption, further statistical analysises can be applied. For instance, to analyse whether the mean difference between male and female’s salaries is zero or not, paired sample t-test can be used. For this procedure, there are 4 main assumptions and one of them is the dependent variable should be approximately normally distributed and the other one is the dependent variable should not contain any outliers. Because of these assumptions, last two sections[Transform](http://rstudio-pubs-static.s3.amazonaws.com/503514_7c937cc6d00d4b6882b221774a4214ae.html#transform),[Detecting Outliers] have vital importance.

Resources

1- <https://camberwellshopping.com.au/trader/anz-bank/>

2- <https://en.wikipedia.org/wiki/Australia_and_New_Zealand_Banking_Group>

3- <http://rare-phoenix-161610.appspot.com/secured/Module_01.html>

4- <http://rare-phoenix-161610.appspot.com/secured/Module_04.html#select>()\_function

5- <https://www.rdocumentation.org/packages/base/versions/3.6.0/topics/factor>

6- <https://vita.had.co.nz/papers/tidy-data.pdf>

7- <https://r4ds.had.co.nz/tidy-data.html>

8- (<https://www.pcicomplianceguide.org/how-are-transaction-volumes-calculated-to-determine-merchant-level/>)

9- <https://r4ds.had.co.nz/tidy-data.html>