Data Wrangling (Data Preprocessing)

Code

Practical assessment 2

Wing Hang Chan

26-May-022

Setup

Hide

```
suppressMessages(library(dplyr))
suppressMessages(library(kableExtra))
suppressMessages(library(lubridate))
library(magrittr)
library(MVN)
library(readr)
library(readxl)
library(outliers)
library(stringr)
```

Student names, numbers and percentage of contributions

Group information

Student name	Student number	Percentage of contribution
Wing Hang, Chan	S3939713	100%

Executive Summary

Data set has been downloaded to the local machine and save to a directory which is near the R source code. The R code studio then has to locate the working directory with the data set saved.
read_csv and ***read_xlsx*** functions are used to read the data set which are provided by readr and readxl library respectively. ***merge*** function is used for joining two data frames to meet the minimum requirement 1. Then, ***as.factor*** function is chosen for transforming a row data from character to ordered factors. as.***date*** is also run for converting time to date for requirement 2-4. A function is created to replace value of 0 to na with ***mutate*** and ***across*** functions. ***unique*** is applied for checking the distinct ID. ***mutate*** is called to transform and create new columns. ***is.na*** function is used for checking na value in columns. Those functions are used to fulfill the requirement 5-7. ***boxplot*** function is used for plotting a visual graph to spot the outliers. MVN package is also applied for detecting outliers to meet requirement 9. ***log***, ***scale*** and ***hist*** are used for normalization of data and showing a balanced chart which achieves requirement 9. Requirement 10 must be satisfy by importing the libraries for those functions.
Finally, the step of pre-processing should be reshuffled. The separated dataset should be tidy up before they are being merged. It is because after merging, there may be more duplicated rows and processing time will be longer.

Data

Found a rental properties collaboration data set from kaggle.com (kaggle.com 2020) The data set contains of 3 files, 2 csv files and 1 xlsx file. Attribute **event_type** in user_activity table can be referred to abttribute **Event type** in event_types. **item_id** in user_activity can be referred to another **item_id** in user_activity. They are not merged in step 1, as to eliminate duplication data for next few steps. Detail data description describes as below tables.

property.csv

```
<tr>
```

Name	Description	Type
item_id	Unique property id	character(36)
deposit	Bond of the property	number
monthly_rent	Price of rent the property	number
district_uuid	Location area unique id	character(36)
room_qty	Number of rooms	number
unit_area	area of house	number
has_elevator	Property has elevator or not	boolean
building_floor_count	Number of units in same floor	number
unit_floor	nth floor of the house	number
has_storage_area	Property include storage location or not	boolean
property_age	Age of the property	number

user_activity.csv

Name	Description	Type
item_id	Unique property id	character(36)
user_id	Unique user id	character(36)
event_type	Log of user activity occured in each process step	Factor of event type
create_timestamp	Timestamp of creating records	Datetime

event_types.xlsx

Name	Description	Type
Step	Order of the series of events	number
Event type	Name of event	character
meaning	Description of event	character

Ref: kaggle.com 2020, Rental Properties Collaboration Data, Möbius, data file, AODC Public Domain Dedication and Licence (PDDL), https://www.kaggle.com/datasets/arashnic/property-data (https://www.kaggle.com/datasets/arashnic/property-data)

Hide

```
getwd()
```

```
[1] "/Users/chello/Desktop/RMIT/Math2349 - Data Wrangling/pratical assessment 2"
```

Hide

```
setwd('/Users/chello/Desktop/RMIT/Math2349 - Data Wrangling/pratical assessment 2')
property <- read_csv('data/property.csv')
```

```
Rows: 4930 Columns: 11
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (2): item_id, district_uuid
dbl (7): deposit, monthly_rent, room_qty, unit_area, building_floor_count, unit_floor,...
lgl (2): has_elevator, has_storage_area

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Hide

```
usr_act <- read_csv('data/user_activity.csv')
```

```
Rows: 323893 Columns: 4
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (3): item_id, user_id, event_type
dttm (1): create_timestamp

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Hide

```
evt_types <- read_excel('data/event_types.xlsx')

head(property)
```

item_id <chr>	deposit <dbl>	monthly_rent <dbl>	district_uuid <chr>
91c0e569-bddd-4128-9720-2550bb85580e	64800000	0	263682f6-d0cd-4569-aeec-e727b76b7665
b00b7919-06be-4d26-98b8-1971787e1d46	72000000	4320000	97c9535e-3985-47ce-a84c-a962c838a76b
9eddb6bc-e424-4774-b55f-bfd54366d627	50400000	1440000	b790f536-c274-4147-86e0-94d9b6d7352d
12cf6b07-5d56-4126-94d2-ce9cbfe2214f	36000000	864000	93d06676-4975-4cc5-919b-3a0c29c7ad43
929eb20c-3694-46b2-b96c-91117b995d1b	28800000	1296000	58e59fa9-9947-478f-9cef-bc6a2cbe49a9
834d0738-9820-43bd-90b3-69e61e833201	122400000	0	93d06676-4975-4cc5-919b-3a0c29c7ad43

6 rows | 1-4 of 11 columns

Hide

```
head(usr_act)
```

item_id <chr>	user_id <chr>	event_type <chr>
00062bc5-2535-4b1e-bbcb-228526c990b8	182aa519-83a8-848f-84a1-8697046d84c2	seen
00062bc5-2535-4b1e-bbcb-228526c990b8	189a081a-ae0f-499d-9092-01758d93fa7f	seen
00062bc5-2535-4b1e-bbcb-228526c990b8	189a081a-ae0f-499d-9092-01758d93fa7f	sent_catalog_link
00062bc5-2535-4b1e-bbcb-228526c990b8	189a081a-ae0f-499d-9092-01758d93fa7f	visit_request-canceled
00062bc5-2535-4b1e-bbcb-228526c990b8	189a081a-ae0f-499d-9092-01758d93fa7f	visit_request-new
00062bc5-2535-4b1e-bbcb-228526c990b8	18e0a74e-8418-854e-8a77-84aa7577d551	seen_in_list

6 rows | 1-3 of 4 columns

Hide

```
head(evt_types)
```

Step <dbl>	Event type <chr>	meaning <chr>
1	seen_in_list	Home impression on website
2	seen	Home view on Address website
3	suggest-new	Home link sent via text message, suggested by Address team
4	suggest_similar	Similar homes sent to tenant
5	sent_catalog_link	Home link sent via text message
6	visit_request-new	User requested to visit the home

6 rows

Hide

```
# requirement 1
usr_act_detail <- merge(x=usr_act, y=evt_types,
                  by.x = c("event_type"), by.y = c("Event type") )
head(usr_act_detail)
```

	event_type <chr>	item_id <chr>	user_id <chr>
1	deal-success	8e0e5341-5758-4209-bc39-02d1cb20c1e5	d2208de5-19d2-49f1-a501-3a389e9f9e69
2	deal-success	3376d4b2-36b7-4d0b-bc55-f189bbe6bb4f	dd0ea5f3-44a4-440d-aed1-e01560ead4d6
3	deal-success	59b1416e-4206-4cb5-9bb8-9b2554db2b43	4a7308e7-a54f-4ef2-a39c-069988694fa3
4	deal-success	57d9b528-3708-4c39-9e83-4f6eb25fdc26	58663c8d-5aa2-407a-8892-0146875ce67e
5	deal-success	74788b97-36b2-4e2b-b7b0-0b444676fcb5	fee6c60a-c5ac-4fe7-aaf8-df7f59e959ad
6	deal-success	0b8b1b1f-0102-41c5-b7c6-8228f5d023f0	f6aa23d5-9ae9-415d-98c5-13d3f08acc9a

6 rows | 1-4 of 6 columns

Hide

```
# Do not merge here to avoid duplicate values for next steps
# merged <- merge(x=property, y=usr_act, by = "item_id")
```

Understand

Hide

```
# requirement 2
c(evt_types['Event type'])
```

```
$`Event type`
 [1] "seen_in_list"              "seen"                      "suggest-new"
 [4] "suggest_similar"           "sent_catalog_link"         "visit_request-new"
 [7] "visit_request-canceled"    "visit-new"                 "visit-canceled"
[10] "visit-unsuccess"           "visit-success"             "meeting_request-new"
[13] "meeting_request-canceled" "meeting-new"               "meeting-canceled"
[16] "meeting-unsuccess"         "meeting-success"           "deal-success"
```

Hide

```
evt_types_lv = evt_types[['Event type']]

# requirement 3 & 4
usr_act_detail$event_type <-
  factor(usr_act_detail$event_type,
         levels = evt_types_lv,
         ordered = TRUE)
head(usr_act_detail)
```

	event_type <ord>	item_id <chr>	user_id <chr>
1	deal-success	8e0e5341-5758-4209-bc39-02d1cb20c1e5	d2208de5-19d2-49f1-a501-3a389e9f9e69
2	deal-success	3376d4b2-36b7-4d0b-bc55-f189bbe6bb4f	dd0ea5f3-44a4-440d-aed1-e01560ead4d6
3	deal-success	59b1416e-4206-4cb5-9bb8-9b2554db2b43	4a7308e7-a54f-4ef2-a39c-069988694fa3
4	deal-success	57d9b528-3708-4c39-9e83-4f6eb25fdc26	58663c8d-5aa2-407a-8892-0146875ce67e
5	deal-success	74788b97-36b2-4e2b-b7b0-0b444676fcb5	fee6c60a-c5ac-4fe7-aaf8-df7f59e959ad
6	deal-success	0b8b1b1f-0102-41c5-b7c6-8228f5d023f0	f6aa23d5-9ae9-415d-98c5-13d3f08acc9a

6 rows | 1-4 of 6 columns

Hide

```
str(usr_act_detail['event_type'])
```

```
'data.frame':   323893 obs. of  1 variable:
 $ event_type: Ord.factor w/ 18 levels "seen_in_list"<..: 18 18 18 18 18 18 18 18 18 18 ...
```

Hide

```
# create a new column as create_date from create_timestamp
usr_act_detail$create_date <- as.Date(usr_act_detail$create_timestamp)
# another data type conversions done in section 4 for event_type
```

Data information of usr_act_detail (event_types.xlsx)

Hide

```
typeof(usr_act_detail)
```

```
[1] "list"
```

Hide

```
str(usr_act_detail)
```

```
'data.frame':   323893 obs. of  7 variables:
 $ event_type      : Ord.factor w/ 18 levels "seen_in_list"<..: 18 18 18 18 18 18 18 18 18 18 ...
 $ item_id         : chr  "8e0e5341-5758-4209-bc39-02d1cb20c1e5" "3376d4b2-36b7-4d0b-bc55-f189bbe6bb4f" "59b1416e
-4206-4cb5-9bb8-9b2554db2b43" "57d9b528-3708-4c39-9e83-4f6eb25fdc26" ...
 $ user_id         : chr  "d2208de5-19d2-49f1-a501-3a389e9f9e69" "dd0ea5f3-44a4-440d-aed1-e01560ead4d6" "4a7308e7
-a54f-4ef2-a39c-069988694fa3" "58663c8d-5aa2-407a-8892-0146875ce67e" ...
 $ create_timestamp: POSIXct, format: "2020-02-09 17:30:12" "2020-02-10 20:16:47" ...
 $ Step            : num  18 18 18 18 18 18 18 18 18 18 ...
 $ meaning         : chr  "Successful rent" "Successful rent" "Successful rent" "Successful rent" ...
 $ create_date     : Date, format: "2020-02-09" "2020-02-10" ...
```

Hide

```
class(usr_act_detail)
```

```
[1] "data.frame"
```

Hide

```
dim(usr_act_detail)
```

```
[1] 323893      7
```

Data information of property (property.csv)

Hide

```
typeof(property)
```

```
[1] "list"
```

Hide

```
str(property)
```

```
spec_tbl_df [4,930 × 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ item_id             : chr [1:4930] "91c0e569-bddd-4128-9720-2550bb85580e" "b00b7919-06be-4d26-98b8-1971787e1d4
6" "9eddb6bc-e424-4774-b55f-bfd54366d627" "12cf6b07-5d56-4126-94d2-ce9cbfe2214f" ...
 $ deposit             : num [1:4930] 64800000 72000000 50400000 36000000 28800000 ...
 $ monthly_rent        : num [1:4930] 0 4320000 1440000 864000 1296000 ...
 $ district_uuid       : chr [1:4930] "263682f6-d0cd-4569-aeec-e727b76b7665" "97c9535e-3985-47ce-a84c-a962c838a76
b" "b790f536-c274-4147-86e0-94d9b6d7352d" "93d06676-4975-4cc5-919b-3a0c29c7ad43" ...
 $ room_qty            : num [1:4930] 1 2 1 1 1 2 1 1 1 1 ...
 $ unit_area           : num [1:4930] 42 116 74 60 45 86 58 68 68 42 ...
 $ has_elevator        : logi [1:4930] FALSE TRUE FALSE TRUE TRUE TRUE ...
 $ building_floor_count: num [1:4930] 3 NA 2 NA NA NA NA 3 2 4 ...
 $ unit_floor          : num [1:4930] 0 1 0 2 1 5 0 2 2 1 ...
 $ has_storage_area    : logi [1:4930] TRUE TRUE TRUE FALSE TRUE FALSE ...
 $ property_age        : num [1:4930] 23 16 19 6 4 5 25 18 27 0 ...
 - attr(*, "spec")=
  .. cols(
  ..   item_id = col_character(),
  ..   deposit = col_double(),
  ..   monthly_rent = col_double(),
  ..   district_uuid = col_character(),
  ..   room_qty = col_double(),
  ..   unit_area = col_double(),
  ..   has_elevator = col_logical(),
  ..   building_floor_count = col_double(),
  ..   unit_floor = col_double(),
  ..   has_storage_area = col_logical(),
  ..   property_age = col_double()
  .. )
 - attr(*, "problems")=<externalptr>
```

Hide

```
class(property)
```

```
[1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

Hide

```
dim(property)
```

```
[1] 4930   11
```

Data information of usr_act (user_activity.csv)

Hide

```
typeof(usr_act)
```

```
[1] "list"
```

Hide

```
str(usr_act)
```

```
spec_tbl_df [323,893 × 4] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ item_id         : chr [1:323893] "00062bc5-2535-4b1e-bbcb-228526c990b8" "00062bc5-2535-4b1e-bbcb-228526c990b8"
"00062bc5-2535-4b1e-bbcb-228526c990b8" "00062bc5-2535-4b1e-bbcb-228526c990b8" ...
 $ user_id         : chr [1:323893] "182aa519-83a8-848f-84a1-8697046d84c2" "189a081a-ae0f-499d-9092-01758d93fa7f"
"189a081a-ae0f-499d-9092-01758d93fa7f" "189a081a-ae0f-499d-9092-01758d93fa7f" ...
 $ event_type      : chr [1:323893] "seen" "seen" "sent_catalog_link" "visit_request-canceled" ...
 $ create_timestamp: POSIXct[1:323893], format: "2020-02-03 15:47:25" "2020-02-04 20:19:31" ...
 - attr(*, "spec")=
  .. cols(
  ..   item_id = col_character(),
  ..   user_id = col_character(),
  ..   event_type = col_character(),
  ..   create_timestamp = col_datetime(format = "")
  .. )
 - attr(*, "problems")=<externalptr>
```

Hide

```
class(usr_act)
```

```
[1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

Hide

```
dim(usr_act)
```

```
[1] 323893      4
```

Data information of evt_types (event_types.xlsx)

Hide

```
typeof(evt_types)
```

```
[1] "list"
```

Hide

```
str(evt_types)
```

```
tibble [18 × 3] (S3: tbl_df/tbl/data.frame)
 $ Step      : num [1:18] 1 2 3 4 5 6 7 8 9 10 ...
 $ Event type: chr [1:18] "seen_in_list" "seen" "suggest-new" "suggest_similar" ...
 $ meaning   : chr [1:18] "Home impression on website" "Home view on Address website" "Home link sent via text me
ssage, suggested by Address team" "Similar homes sent to tenant" ...
```

Hide

```
class(evt_types)
```

```
[1] "tbl_df"     "tbl"        "data.frame"
```

Hide

```
dim(evt_types)
```

```
[1] 18  3
```

usr_act_detail is a data frame with 323893 obs of 7 vars. *event_type* is an ordered factor with 18 levels. The order is from **event_type**s' *step*. **property** is a data frame with 4930 obs of 17 vars. **usr_act** is a data frame with 323893 of 4 vars. **evt_types** is a data frame with 18 obs of 3 vars. *create_date* of **usr_act_detail** is converted from *create_timestamp* as date.

Tidy & Manipulate Data I

Hide

```
replace_0_na <- function(x) {
  ifelse(x == 0, NA, x)
}

property <- property %>%
  mutate(across(
    c(
      'deposit',
      'monthly_rent',
      'unit_area',
      'building_floor_count'
    ),
    replace_0_na
  ))


# unit floor 225 outliers

# check uniqueness for item_id
length(unique(property$item_id)) == dim(property)[1]
```

```
[1] TRUE
```

Hide

```
length(unique(property$district_uuid)) == dim(property)[1]
```

```
[1] FALSE
```

Hide

```
length(unique(usr_act_detail$uesr_id)) == dim(usr_act_detail)[1]
```

```
[1] FALSE
```

Hide

```
# unit floor -2 -1 0 may represent underground or ground floor
```

A function is created to replace 0 to NA which is applied to 4 cols (*deposit, monthly_rent, unit_area and building_floor_count*). Their values of 0 are replaced as those values are impossible to be 0 by definition. The 0-value will lead to calculating the mean wrongly. There is another check with the uniqueness for *item_id* which the result is passed as it is equal to the dimension of **property** data frame. Another two ID distinct no. is not equal to **property** and **usr_act_detail** dimension. The result is acceptable as they can be duplicated as the many-to-many relationship. Unit floor (-2, -1, 0) may represent underground or ground floor. There is another unit floor of 225. It may be an outlier which will be handled in Scan II.

Tidy & Manipulate Data II

Hide

```
property <- property %>%
  mutate(
    weekly_rent = monthly_rent * 12 / 52,
  )

property <- property %>%
  mutate(
    deposit_rent_area_ratio = deposit / (monthly_rent * unit_area),
  )
```

weekly_rent is calculated from monthly rent. *deposit_rent_area_ratio* is also calculated from *deposit* divided by product of *monthly_rent* and *unit_area*. It is used in step 9.

Scan I

Hide

```
sum(is.na(property$deposit))
```

```
[1] 13
```

Hide

```
sum(is.na(property$monthly_rent))
```

```
[1] 1846
```

Hide

```
sum(is.na(property$room_qty))
```

```
[1] 3
```

Hide

```
sum(is.na(property$unit_area))
```

```
[1] 2
```

Hide

```
sum(is.na(property$has_elevator))
```

```
[1] 17
```

Hide

```
sum(is.na(property$building_floor_count))
```

```
[1] 1650
```

Hide

```
sum(is.na(property$unit_floor))
```

```
[1] 37
```

Hide

```
sum(is.na(property$has_storage_area))
```

```
[1] 10
```

Hide

```
sum(is.na(property$property_age))
```

```
[1] 4
```

Hide

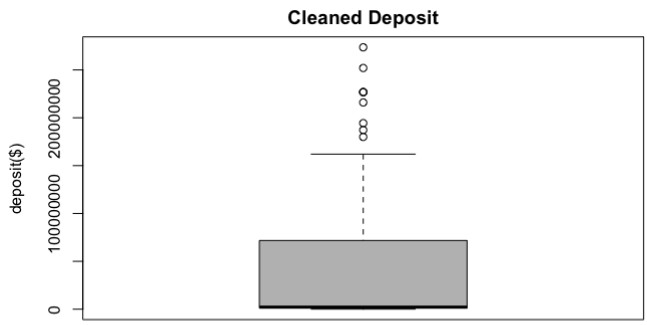
```
district_deposit_mean <- property %>%
  group_by(district_uuid) %>%
  summarise(across(deposit, mean, .names="district_deposit_mean", na.rm = TRUE))

property <- merge(x=property, y=district_deposit_mean,
                  by = "district_uuid" )

property <- property %>%
  mutate(cleaned_deposit = ifelse(
    is.na(monthly_rent),
    `district_deposit_mean`,
    `monthly_rent`
  ))

district_monthly_rent_mean <- property %>%
  group_by(district_uuid) %>%
  summarise(across(monthly_rent,
                   mean,
                   .names = "district_monthly_rent_mean",
                   na.rm = TRUE))

property <- merge(x = property, y = district_monthly_rent_mean,
                  by = "district_uuid")

property <- property %>%
  mutate(cleaned_monthly_rent = ifelse(
    is.na(monthly_rent),
    `district_monthly_rent_mean`,
    `monthly_rent`
  ))
```

There are na values in *deposit, monthly_rent, room_qty, unit_area, has_elevator, building_floor_count, unit_floor, has_storage_area and property*. However, all values (except *deposit* and *monthly_rent*) can not be guessed or replaced by a calculated value as they should be from input. For *deposit* and *monthly_rent*, there are a *cleaned_deposit* and *cleaned_monthly_rent* by replacing na value with average value. They do not replace the original value because it is to keep na value aswe do not know why the na/0 value is filled.

Scan II

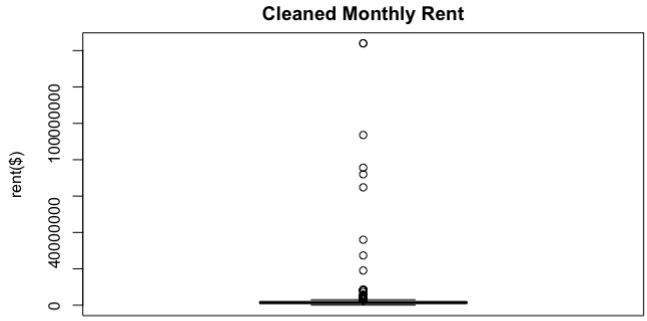
Hide

```
options(scipen = 999)
property$cleaned_deposit %>%
  boxplot(main="Cleaned Deposit", ylab="deposit($)", col = "grey")
```



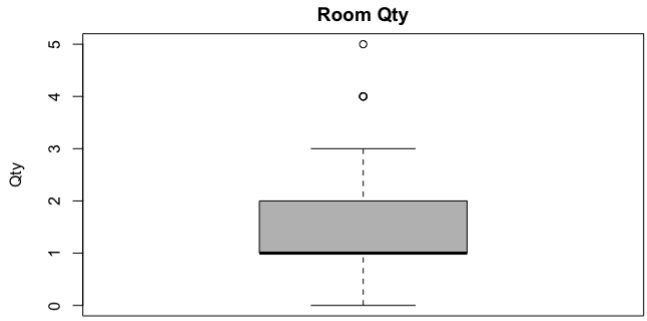
Hide

```
property$cleaned_monthly_rent %>%
  boxplot(main="Cleaned Monthly Rent", ylab="rent($)", col = "grey")
```



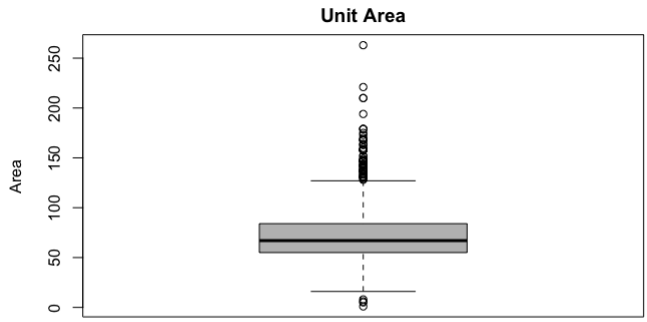
Hide

```
property$room_qty %>%
  boxplot(main="Room Qty", ylab="Qty", col = "grey")
```



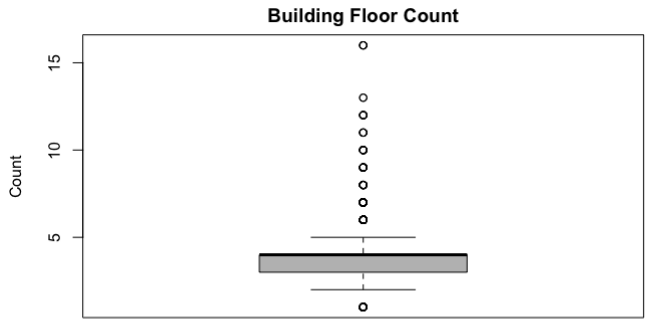
Hide

```
property$unit_area %>%
  boxplot(main="Unit Area", ylab="Area", col = "grey")
```



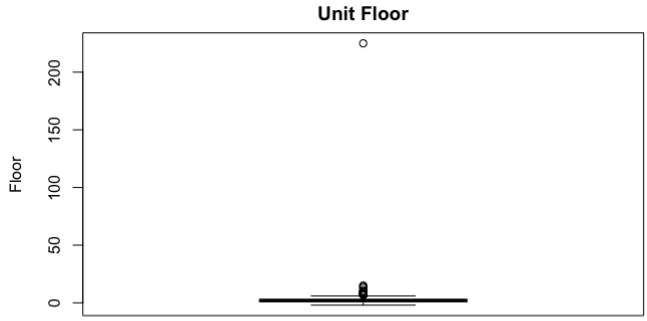
Hide

```
property$building_floor_count %>%
  boxplot(main="Building Floor Count", ylab="Count", col = "grey")
```



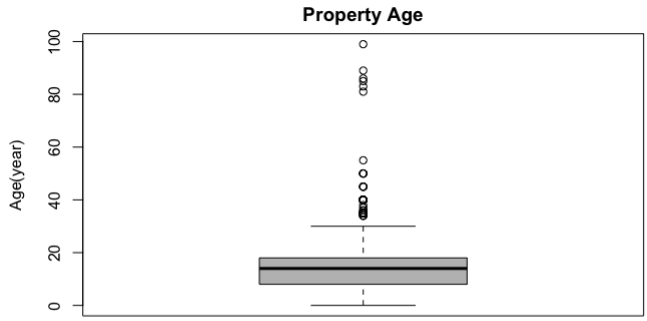
Hide

```
property$unit_floor %>%
  boxplot(main="Unit Floor", ylab="Floor", col = "grey")
```



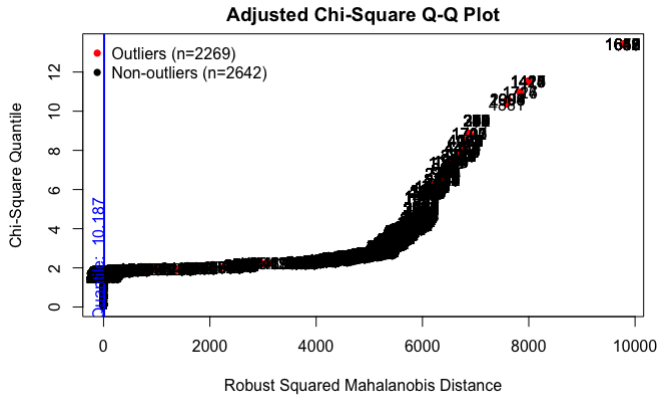
Hide

```
property$property_age %>%
  boxplot(main="Property Age", ylab="Age(year)", col = "grey")
```



Hide

```
p <- property %>% select('cleaned_deposit',
                         'cleaned_monthly_rent',
                         'district_uuid') %>% group_by(district_uuid)

results <-
  mvn(
    data = property %>% select('cleaned_deposit', 'cleaned_monthly_rent'),
    multivariateOutlierMethod = "adj",
    showOutliers = TRUE,
    bc = TRUE, bcType="optimal"
  )
```



Hide

```
results$multivariateOutliers
```

	Observation <chr>	Mahalanobis Distance <dbl>	Outlier <chr>
1638	1638	9770.335	TRUE
1639	1639	9770.335	TRUE
1640	1640	9770.335	TRUE
1642	1642	9770.335	TRUE
1643	1643	9770.335	TRUE
1645	1645	9770.335	TRUE
1647	1647	9770.335	TRUE
1648	1648	9770.335	TRUE
1649	1649	9770.335	TRUE
1651	1651	9770.335	TRUE

1-10 of 2,269 rows Previous **1** 2 3 4 5 6 … 100 Next

Hide

```
NA
```

There are 7 numeric attribute values. Using box spot can be spotted. They are *cleaned_deposit, cleaned_monthly_rent, room_qty, unit_area, building_floor_count, unit_floor and property_age*. Those outliers cannot be deleted, as there may be references for user activity tables. They cannot be replaced by mean value as well, because those values have their own meaning and the mean tells nothing to those attributes. Therefore, no action can be applied for those outliers.
There are so many outliners for *clean_deposit* and *clean_monthly_rent* because they are in different districts. There may be a chance of higher value with the city districts. Also the higher unit area, the higher the deposit.

Transform

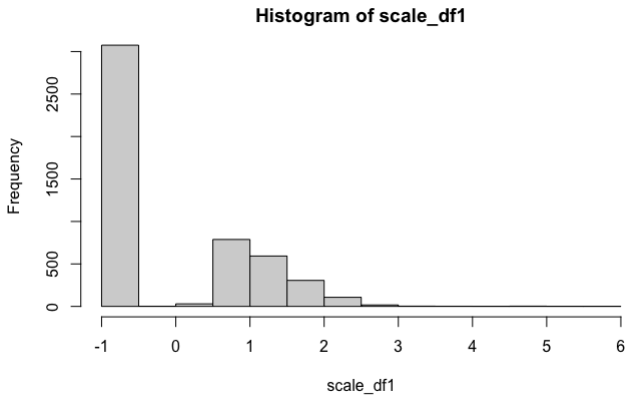
Hide

```
scale_df1 <- scale(property['cleaned_deposit'], center = TRUE, scale = TRUE)
head(scale_df1)
```

```
     cleaned_deposit
[1,]       1.2010999
[2,]       1.2010999
[3,]      -0.7012336
[4,]      -0.7336500
[5,]      -0.7353562
[6,]      -0.6927029
```

Hide

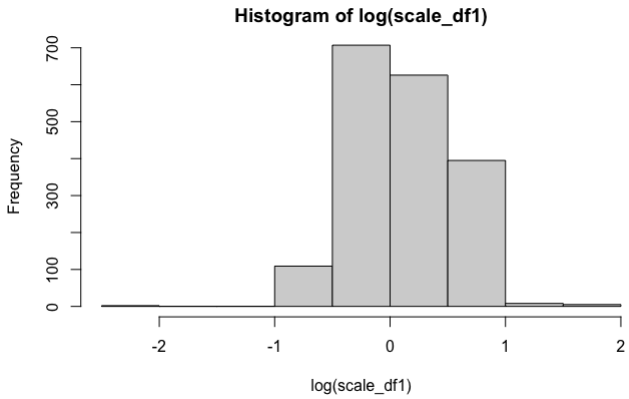
```
hist(scale_df1)
```



Hide

```
hist(log(scale_df1))
```

```
Warning in log(scale_df1) : NaNs produced
```



cleaned_deposit was chosen to be normalised. Scale function is used and centre and scale are true to perform z-score transformation. A histogram shows for the z-scored value, but it is left swifted. Therefore, a log operation was done to centre the histogram.