MATH2349 Semester 2, 2018

Assignment 3

Manuel Matthew Gunadi

Read employment dataset

- The employment data comes from the Australian Bureau of Statistics (ABS) website. The title of the data is “6160.0 Table 1. JOBS and Employment income per job, by selected characteristics and by Regions and by Sex (2011-12 to 2015-16)”. The particular set used is the New South Wales data (Statistical area level 3).
- Variables include: number of jobs (’000) and median employment income per job($) in males, females or persons, SA2 region (ID and name) and years.
- The data can be obtained from: http://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/6160.02011-12%20to%202015-16?OpenDocument (http://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/6160.02011-12%20to%202015-16?OpenDocument)

```
Employment <- read_excel("ABS_Employment.xlsx", sheet = "Table 1.5", range = "A7:Q2305
")
colnames(Employment)
```

```
##  [1] "X__1"    "X__2"    "MALES"   "X__3"    "X__4"    "X__5"    "X__6"
##  [8] "FEMALES" "X__7"    "X__8"    "X__9"    "X__10"   "PERSONS" "X__11"
## [15] "X__12"   "X__13"   "X__14"
```

```
head(Employment)
```

```
## # A tibble: 6 x 17
##   X__1  X__2  MALES X__3  X__4  X__5  X__6  FEMALES X__7  X__8  X__9  X__~
##   <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>   <chr> <chr> <chr> <ch>
## 1 SA2   SA2 ~ 2011~ 2012~ 2013~ 2014~ 2015~ 2011-12 2012~ 2013~ 2014~ 201~
## 2 Aust~ <NA>  9474~ 9578~ 9539~ 9591~ 9637~ 8532.1  8679~ 8691~ 8769~ 886~
## 3 New ~ <NA>  2916~ 2949~ 2952~ 2977~ 3039~ 2633.9~ 2725~ 2718~ 2726~ 278~
## 4 1010~ Brai~ 1.54~ 1.583 1.56~ 1.55~ 1.52~ 1.3979~ 1.49~ 1.476 1.415 1.5~
## 5 1010~ Kara~ 3.97~ 3.98~ 3.65~ 3.6   3.68~ 3.746   3.68~ 3.415 3.31~ 3.3~
## 6 1010~ Quea~ 5.21~ 5.22~ 4.88  4.78~ 4.92~ 4.5199~ 4.48~ 4.18~ 4.20~ 4.3~
## # ... with 5 more variables: PERSONS <chr>, X__11 <chr>, X__12 <chr>,
## #   X__13 <chr>, X__14 <chr>
```

- Inspect/ understand Employment data structure:

```
class(Employment)
```

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

```
dim(Employment)
```

```
## [1] 2298   17
```

```
names(Employment)
```

```
##  [1] "X__1"    "X__2"    "MALES"   "X__3"    "X__4"    "X__5"    "X__6"
##  [8] "FEMALES" "X__7"    "X__8"    "X__9"    "X__10"   "PERSONS" "X__11"
## [15] "X__12"   "X__13"   "X__14"
```

```
sapply(Employment,class)
```

```
##        X__1        X__2       MALES        X__3        X__4        X__5
## "character" "character" "character" "character" "character" "character"
##        X__6     FEMALES        X__7        X__8        X__9       X__10
## "character" "character" "character" "character" "character" "character"
##     PERSONS       X__11       X__12       X__13       X__14
## "character" "character" "character" "character" "character"
```

Read income dataset

```
Income <- read_excel("ABS_Employment.xlsx", sheet = "Table 1.5", range = "R7:AF2305")
colnames(Income)
```

```
##  [1] "MALES"   "X__1"    "X__2"    "X__3"    "X__4"    "FEMALES" "X__5"
##  [8] "X__6"    "X__7"    "X__8"    "PERSONS" "X__9"    "X__10"   "X__11"
## [15] "X__12"
```

```
head(Income)
```

```
## # A tibble: 6 x 15
##   MALES X__1  X__2  X__3  X__4  FEMALES X__5  X__6  X__7  X__8  PERSONS
##   <chr> <chr> <chr> <chr> <chr> <chr>   <chr> <chr> <chr> <chr> <chr>
## 1 2011~ 2012~ 2013~ 2014~ 2015~ 2011-12 2012~ 2013~ 2014~ 2015~ 2011-12
## 2 27769 28799 29537 29963 30410 17247   18000 18735 19676 20538 21918
## 3 28223 28862 29767 30279 30875 18608.5 18517 19627 20672 21645 22938
## 4 17914 17784 19553 21523 2332~ 13103   1223~ 12000 14439 1349~ 15123
## 5 35269 33944 3681~ 38015 34614 23000   26229 27863 28725 3022~ 28614
## 6 3078~ 32952 33250 34091 33009 24030.5 27010 29618 29580 29654 27234.5
## # ... with 4 more variables: X__9 <chr>, X__10 <chr>, X__11 <chr>,
## #   X__12 <chr>
```

- Inspect/ understand Income data structure:

```
class(Income)
```

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

```
dim(Income)
```

```
## [1] 2298   15
```

```
names(Income)
```

```
##  [1] "MALES"   "X__1"    "X__2"    "X__3"    "X__4"    "FEMALES" "X__5"
##  [8] "X__6"    "X__7"    "X__8"    "PERSONS" "X__9"    "X__10"   "X__11"
## [15] "X__12"
```

Data tidying

- Clean employment data of all persons (male and female) into tidy format.
- First, subset the columns relating to ‘persons’
- Second, subset the rows which relate to observations for each region
- Gather the various columns containing year ranges into one long column
- convert into a data frame structure
- Convert “no. of jobs” variable from character to numeric, rounded to 3 digits.

```
#1
all_employment <-Employment[,c(2,13:17)]
colnames(all_employment)[1:6] <- all_employment[1,1:6]
#2
all_employment <- all_employment[4:nrow(all_employment),]
#3
all_emp <- all_employment %>% gather("2011-12", "2012-13", "2013-14", "2014-15", "2015
-16", key = "year", value = "no. of jobs")
#4
all_emp <- as.data.frame(all_emp)
#5
all_emp$`no. of jobs` <- round(as.numeric(all_emp$`no. of jobs`), digits = 3)
```

```
## Warning: NAs introduced by coercion
```

```
#6
head(all_emp)
```

```
##                            SA2 NAME    year no. of jobs
## 1                         Braidwood 2011-12       2.945
## 2                           Karabar 2011-12       7.719
## 3                        Queanbeyan 2011-12       9.732
## 4                 Queanbeyan - East 2011-12       4.516
## 5                 Queanbeyan Region 2011-12      12.794
## 6 Queanbeyan West - Jerrabomberra 2011-12      11.189
```

- check the data classes of each column are correct to ensure further pre-processing

```
apply(all_emp, 2, class)
```

```
##    SA2 NAME        year no. of jobs
## "character" "character" "character"
```

```
class(all_emp$`no. of jobs`)
```

```
## [1] "numeric"
```

```
#1
all_Income <- bind_cols(Employment[,2],Income);
colnames(all_Income)
```

```
##  [1] "X__2"    "MALES"   "X__1"    "X__21"   "X__3"    "X__4"    "FEMALES"
##  [8] "X__5"    "X__6"    "X__7"    "X__8"    "PERSONS" "X__9"    "X__10"
## [15] "X__11"   "X__12"
```

```
all_Income <- all_Income[,c(1,12:16)]
colnames(all_Income)[1:6] <- all_Income[1,1:6]
#2
all_Income <- all_Income[4:nrow(all_Income),]
#3
all_Income <- all_Income %>% gather("2011-12", "2012-13", "2013-14", "2014-15", "2015-
16", key = "year", value = "Income")
#4
all_Income <- as.data.frame(all_Income)
#5
all_Income$Income <- round(as.numeric(all_Income$Income), digits = 0)
```

```
## Warning: NAs introduced by coercion
```

```
#6
head(all_Income)
```

```
##                             SA2 NAME    year Income
## 1                          Braidwood 2011-12  15123
## 2                            Karabar 2011-12  28614
## 3                         Queanbeyan 2011-12  27234
## 4                  Queanbeyan - East 2011-12  26528
## 5                  Queanbeyan Region 2011-12  29999
## 6 Queanbeyan West - Jerrabomberra 2011-12  37290
```

Merging employment and Income datasets

```
Employ_income <- bind_cols(all_emp, Income = all_Income$Income)
```

Filtering data and further tidying

- As we only have data from the 2016 census data, the most relevant time period for the employment figures is the 2015-2016 data set. Therefore, we filter the employment data for this time range.
- convert the year range, 2015-2016 into a single year, 2016, in numeric format.
- multiply the "no. of jobs" by 1000 as this data is thousands

```
Employ_income <- Employ_income %>% filter(year == "2015-16")
Employ_income <- Employ_income %>% mutate(year = str_replace(year, "15-", ""))
Employ_income$year = as.numeric(Employ_income$year)
Employ_income$`no. of jobs` <- Employ_income$`no. of jobs` * 1000
head(Employ_income)
```

```
##                             SA2 NAME year no. of jobs Income
## 1                          Braidwood 2016        3063  17882
## 2                            Karabar 2016        7067  31950
## 3                         Queanbeyan 2016        9310  31491
## 4                  Queanbeyan - East 2016        4480  29988
## 5                  Queanbeyan Region 2016       14061  37092
## 6 Queanbeyan West - Jerrabomberra 2016       11356  39012
```

Read mortgage dataset

- Read mortgage data from ABS: 2016 Census - Monthly Mortgage Repayments & dwellings location on census night
- The data is ABS census data from the 2016 Australian census. It was downloaded from TableBuilder (https://auth.censusdata.abs.gov.au/webapi/jsf/login.xhtml (https://auth.censusdata.abs.gov.au/webapi/jsf/login.xhtml)) using a public account.
- The fields selected were: * all SA2s within NSW * monthly mortgage repayments by dwelling
- This data is under a creative commons licence.

```
mortgage <- read_excel("NSW_SA2_MortgageRepayments.xlsx", range = "B9:X587")
head(mortgage)
```

```
## # A tibble: 6 x 23
##   X__1  `Nil repayments` `$1-$149` `$150-$299` `$300-$449` `$450-$599`
##   <chr>            <dbl>     <dbl>       <dbl>       <dbl>       <dbl>
## 1 SA2                 NA        NA          NA          NA          NA
## 2 Avoc~               17         8          12          17          14
## 3 Box ~               49        27          15          29          30
## 4 Calg~               27        12           7           8          12
## 5 Erin~               45        13          11          40          33
## 6 Gosf~               44        17          25          42          52
## # ... with 17 more variables: `$600-$799` <dbl>, `$800-$999` <dbl>,
## #   `$1,000-$1,199` <dbl>, `$1,200-$1,399` <dbl>, `$1,400-$1,599` <dbl>,
## #   `$1,600-$1,799` <dbl>, `$1,800-$1,999` <dbl>, `$2,000-$2,199` <dbl>,
## #   `$2,200-$2,399` <dbl>, `$2,400-$2,599` <dbl>, `$2,600-$2,999` <dbl>,
## #   `$3,000-$3,999` <dbl>, `$4,000-$4,999` <dbl>, `$5000 and over` <dbl>,
## #   `Not stated` <dbl>, `Not applicable` <dbl>, Total <dbl>
```

- Inspect/ understand mortgage data structure:

```
class(mortgage)
```

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

```
dim(mortgage)
```

```
## [1] 578  23
```

```
names(mortgage)
```

```
##  [1] "X__1"           "Nil repayments" "$1-$149"        "$150-$299"
##  [5] "$300-$449"      "$450-$599"      "$600-$799"      "$800-$999"
##  [9] "$1,000-$1,199"  "$1,200-$1,399"  "$1,400-$1,599"  "$1,600-$1,799"
## [13] "$1,800-$1,999"  "$2,000-$2,199"  "$2,200-$2,399"  "$2,400-$2,599"
## [17] "$2,600-$2,999"  "$3,000-$3,999"  "$4,000-$4,999"  "$5000 and over"
## [21] "Not stated"     "Not applicable" "Total"
```

```
sapply(mortgage,class)
```

```
##          X__1 Nil repayments       $1-$149      $150-$299      $300-$449
##    "character"      "numeric"      "numeric"      "numeric"      "numeric"
##      $450-$599      $600-$799      $800-$999  $1,000-$1,199  $1,200-$1,399
##      "numeric"      "numeric"      "numeric"      "numeric"      "numeric"
##  $1,400-$1,599  $1,600-$1,799  $1,800-$1,999  $2,000-$2,199  $2,200-$2,399
##      "numeric"      "numeric"      "numeric"      "numeric"      "numeric"
##  $2,400-$2,599  $2,600-$2,999  $3,000-$3,999  $4,000-$4,999 $5000 and over
##      "numeric"      "numeric"      "numeric"      "numeric"      "numeric"
##     Not stated Not applicable          Total
##      "numeric"      "numeric"      "numeric"
```

Tidying the mortgage data.

```
Repayments <- colnames(mortgage[2:22])
mortgage2 <- mortgage %>% gather(Repayments, key = "Most common mortgage repayments",
value = "Repayment reportings")
mortgage2 <- mortgage2[2:nrow(mortgage2),]
colnames(mortgage2)[1] <- "SA2 NAME"
colnames(mortgage2)[2] <- "Total dwellings in SA2"
head(mortgage2)
```

```
## # A tibble: 6 x 4
##   `SA2 NAME`     `Total dwellings ~ `Most common mortga~ `Repayment repor~
##   <chr>                       <dbl> <chr>                            <dbl>
## 1 Avoca Beach -~               3676 Nil repayments                      17
## 2 Box Head - Ma~               5374 Nil repayments                      49
## 3 Calga - Kulnu~               2205 Nil repayments                      27
## 4 Erina - Green~               5760 Nil repayments                      45
## 5 Gosford - Spr~               9213 Nil repayments                      44
## 6 Kariong                      2183 Nil repayments                      27
```

- Convert the mortgage monthly repayments into an ordered factor
- Take out the fators, “Not applicable” and “Not stated” as we are more interested and concerned about knowing the repayment ranges that were stated in the census.

```
mortgage2$`Most common mortgage repayments` <- factor(mortgage2$`Most common mortgage
repayments`, levels = Repayments)
levels(mortgage2$`Most common mortgage repayments`)
```

```
##  [1] "Nil repayments" "$1-$149"        "$150-$299"      "$300-$449"
##  [5] "$450-$599"      "$600-$799"      "$800-$999"      "$1,000-$1,199"
##  [9] "$1,200-$1,399"  "$1,400-$1,599"  "$1,600-$1,799"  "$1,800-$1,999"
## [13] "$2,000-$2,199"  "$2,200-$2,399"  "$2,400-$2,599"  "$2,600-$2,999"
## [17] "$3,000-$3,999"  "$4,000-$4,999"  "$5000 and over" "Not stated"
## [21] "Not applicable"
```

```
clean_mortgage <- mortgage2 %>% filter(!(`Most common mortgage repayments` %in% c("Not
applicable", "Not stated")))
# table(clean_mortgage$`Most common mortgage repayments`)
head(mortgage2)
```

```
## # A tibble: 6 x 4
##   `SA2 NAME`     `Total dwellings ~ `Most common mortga~ `Repayment repor~
##   <chr>                       <dbl> <fct>                            <dbl>
## 1 Avoca Beach -~               3676 Nil repayments                      17
## 2 Box Head - Ma~               5374 Nil repayments                      49
## 3 Calga - Kulnu~               2205 Nil repayments                      27
## 4 Erina - Green~               5760 Nil repayments                      45
## 5 Gosford - Spr~               9213 Nil repayments                      44
## 6 Kariong                      2183 Nil repayments                      27
```

- Find the most commonly occuring repayment range for each region by filtering for the max number of frequency in each SA2.

```
mortgage_common <- clean_mortgage %>% group_by(`SA2 NAME`) %>% filter(`Repayment repor
tings` == max(`Repayment reportings`))
head(mortgage_common)
```

```
## # A tibble: 6 x 4
## # Groups:   SA2 NAME [6]
##   `SA2 NAME`     `Total dwellings ~ `Most common mortga~ `Repayment repor~
##   <chr>                       <dbl> <fct>                            <dbl>
## 1 Prospect Rese~                  7 Nil repayments                       0
## 2 Banksmeadow                     4 Nil repayments                       0
## 3 Port Botany I~                  6 Nil repayments                       0
## 4 Sydney Airport                  7 Nil repayments                       0
## 5 Centennial Pa~                  0 Nil repayments                       0
## 6 Holsworthy Mi~                  0 Nil repayments                       0
```

- Exclude duplicated regions where all repayment reportings are “0”:

```
mort_common_clean <- mortgage_common[!duplicated(mortgage_common$`SA2 NAME`),]
head(mort_common_clean)
```

```
## # A tibble: 6 x 4
## # Groups:   SA2 NAME [6]
##   `SA2 NAME`     `Total dwellings ~ `Most common mortga~ `Repayment repor~
##   <chr>                       <dbl> <fct>                            <dbl>
## 1 Prospect Rese~                  7 Nil repayments                       0
## 2 Banksmeadow                     4 Nil repayments                       0
## 3 Port Botany I~                  6 Nil repayments                       0
## 4 Sydney Airport                  7 Nil repayments                       0
## 5 Centennial Pa~                  0 Nil repayments                       0
## 6 Holsworthy Mi~                  0 Nil repayments                       0
```

Merging employment and mortgage datasets

- The mortgage data does not capture as many regions as the employment data (eg. mortgage_cleaned contain 577 observations compared with all_emp_clean with 2295 observations) If we are using the combined dataset forthe purpose of records, we can join all these variables. However, if pre-processing is for analysis purposes, we should subset only the regions where we have both mortgage and employment data. The next part does this merge.
- Merge the employment dataset with the mortgage dataset by SA2 name.

```
full_data <- Employ_income %>% inner_join(mort_common_clean, by="SA2 NAME")
head(full_data)
```

```
##                           SA2 NAME year no. of jobs Income
## 1                        Braidwood 2016        3063  17882
## 2                          Karabar 2016        7067  31950
## 3                       Queanbeyan 2016        9310  31491
## 4                Queanbeyan - East 2016        4480  29988
## 5                Queanbeyan Region 2016       14061  37092
## 6 Queanbeyan West - Jerrabomberra 2016       11356  39012
##   Total dwellings in SA2 Most common mortgage repayments
## 1                   2297                   $1,600-$1,799
## 2                   3387                   $2,000-$2,199
## 3                   5652                   $2,000-$2,199
## 4                   2458                   $2,000-$2,199
## 5                   6295                   $3,000-$3,999
## 6                   4616                   $3,000-$3,999
##   Repayment reportings
## 1                   42
## 2                  149
## 3                  161
## 4                   66
## 5                  588
## 6                  351
```

Treat missing values

- Scan for missing values in SA2 name, no.of jobs and total dwellings

```
colSums(is.na(full_data))
```

```
##                         SA2 NAME                             year
##                                0                                0
##                      no. of jobs                           Income
##                               13                               13
##            Total dwellings in SA2 Most common mortgage repayments
##                                0                                0
##              Repayment reportings
##                                0
```

- Impute the median number of jobs for missing values here as there are only such cases. The number of missing values is <5% of the data so we can be safe to exclude these observations.

```
imputed_jobs <- Hmisc::impute(full_data$`no. of jobs`, fun=median)
full_data$`no. of jobs` <- imputed_jobs
colSums(is.na(full_data))
```

```
##                         SA2 NAME                             year
##                                0                                0
##                      no. of jobs                           Income
##                                0                               13
##            Total dwellings in SA2 Most common mortgage repayments
##                                0                                0
##              Repayment reportings
##                                0
```

- Impute the median number of jobs for missing values here as there are only such cases. The number of missing values is <5% of the data so we can be safe to exclude these observations.

```
imputed_income <- Hmisc::impute(full_data$Income, fun=median)
full_data$Income <- imputed_income
colSums(is.na(full_data))
```

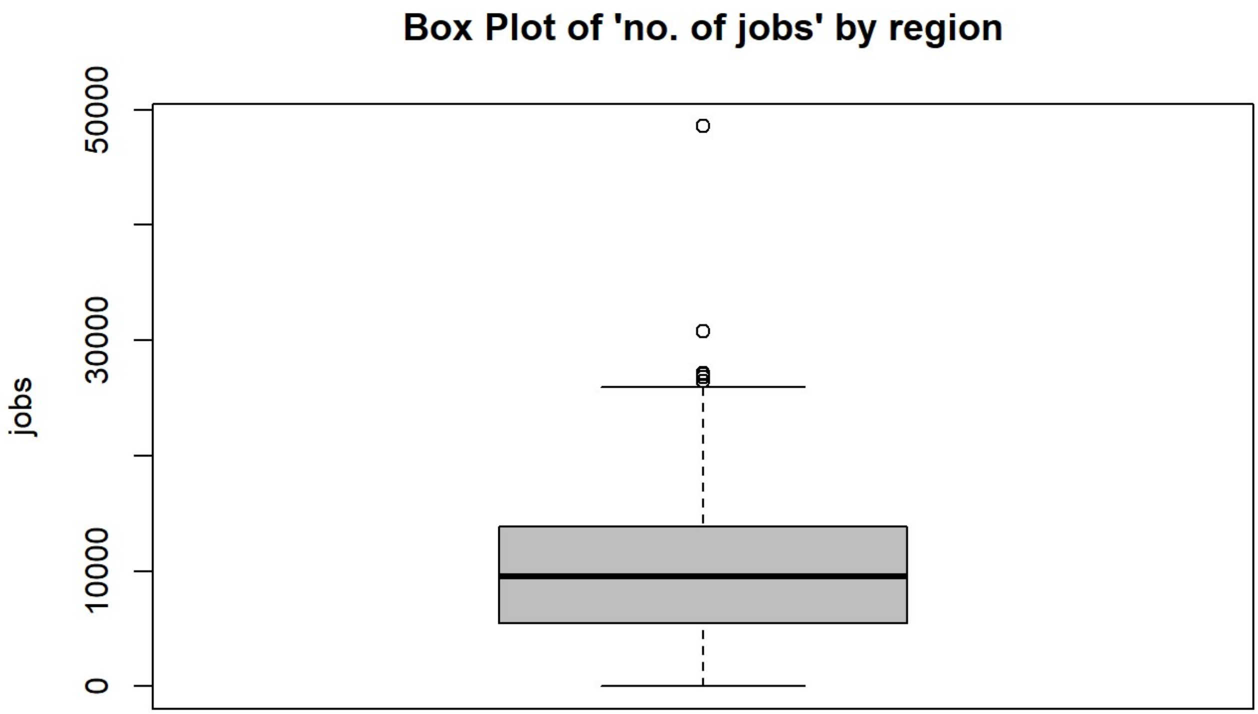
```
##                         SA2 NAME                             year
##                                0                                0
##                      no. of jobs                           Income
##                                0                                0
##            Total dwellings in SA2 Most common mortgage repayments
##                                0                                0
##              Repayment reportings
##                                0
```

Treating univariate and multivariate outliers.

Univariate outliers:

- Detect any outliers in either jobs, income or total dwellings

```
boxplot(as.numeric(full_data$`no. of jobs`), main = "Box Plot of 'no. of jobs' by regi
on", ylab = "jobs", col = "grey")
```




```
z_score_job <- full_data$`no. of jobs` %>% scores(type ="z")
z_score_job %>% summary()
```

```
##
##  13 values imputed to -0.1243648
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.7009 -0.7944 -0.1244  0.0000  0.5899  6.3429
```

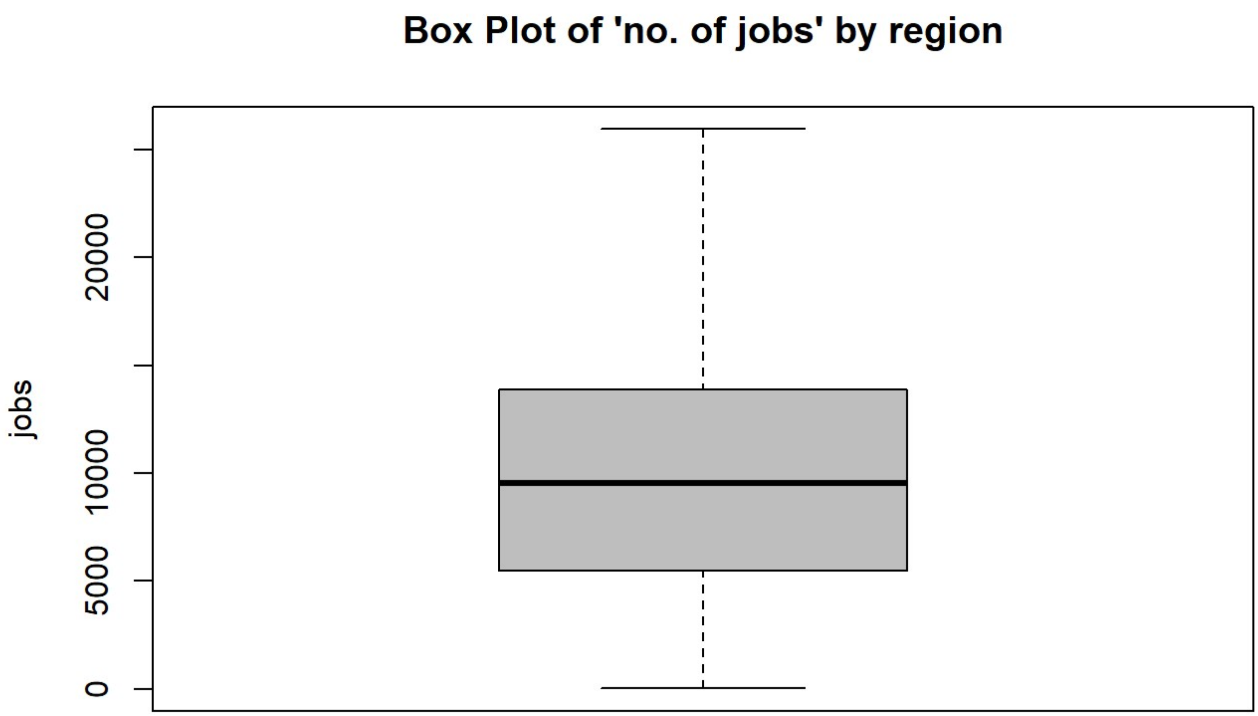
```
which(abs(z_score_job) > 3)
```

```
## [1] 348 349
```

- Handling the outliers by capping

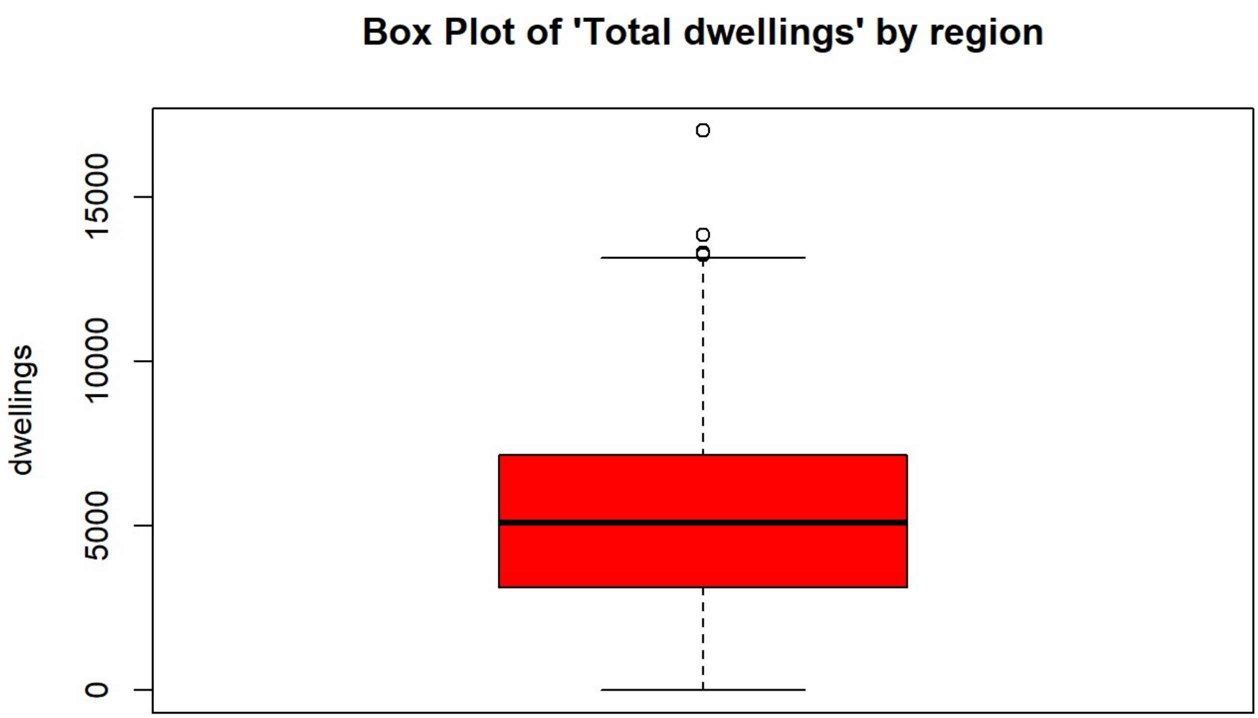
```
cap <- function(x){
    quantiles <- quantile( x, c(.05, 0.25, 0.75, .95 ) )
    x[ x < quantiles[2] - 1.5*IQR(x) ] <- quantiles[1]
    x[ x > quantiles[3] + 1.5*IQR(x) ] <- quantiles[4]
    x
}

jobs_capped <- full_data$`no. of jobs` %>% cap()
boxplot(as.numeric(jobs_capped), main = "Box Plot of 'no. of jobs' by region", ylab =
"jobs", col = "grey")
```



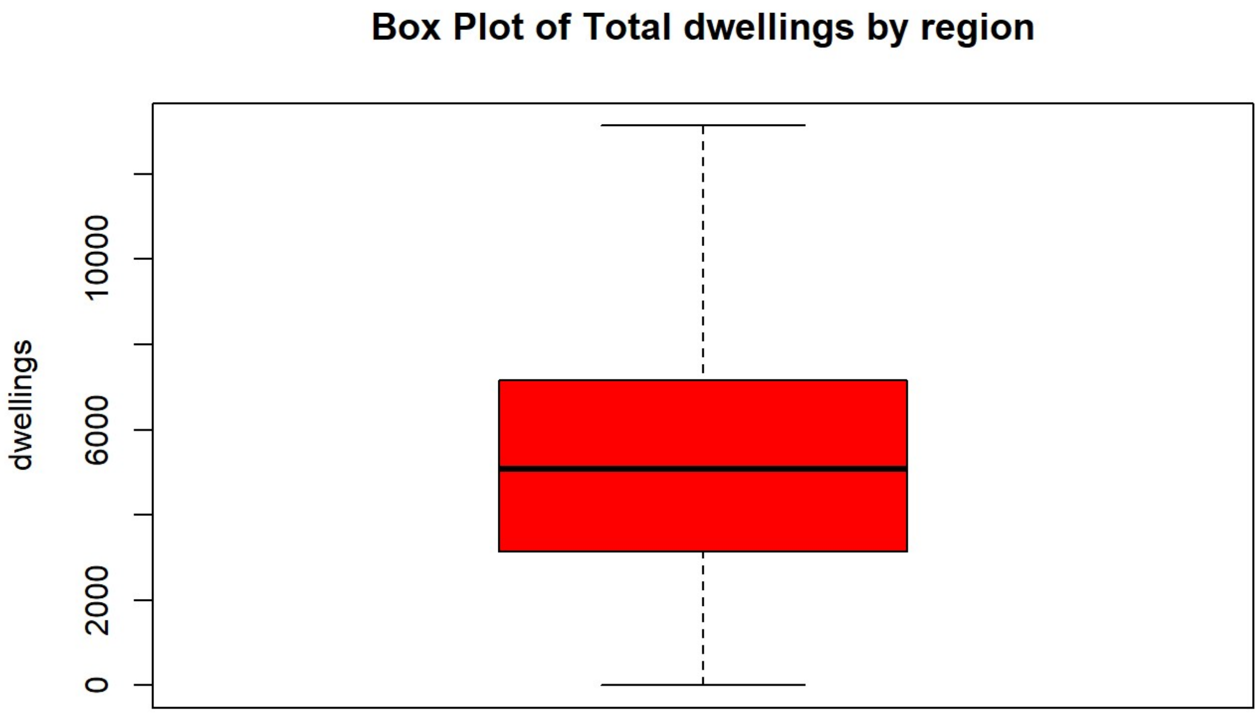

```
full_data$`no. of jobs` <- jobs_capped
```

```
boxplot(as.numeric(full_data$`Total dwellings in SA2`), main = "Box Plot of 'Total dwe
llings' by region", ylab = "dwellings", col = "red")
```



- handle the outliers by capping.

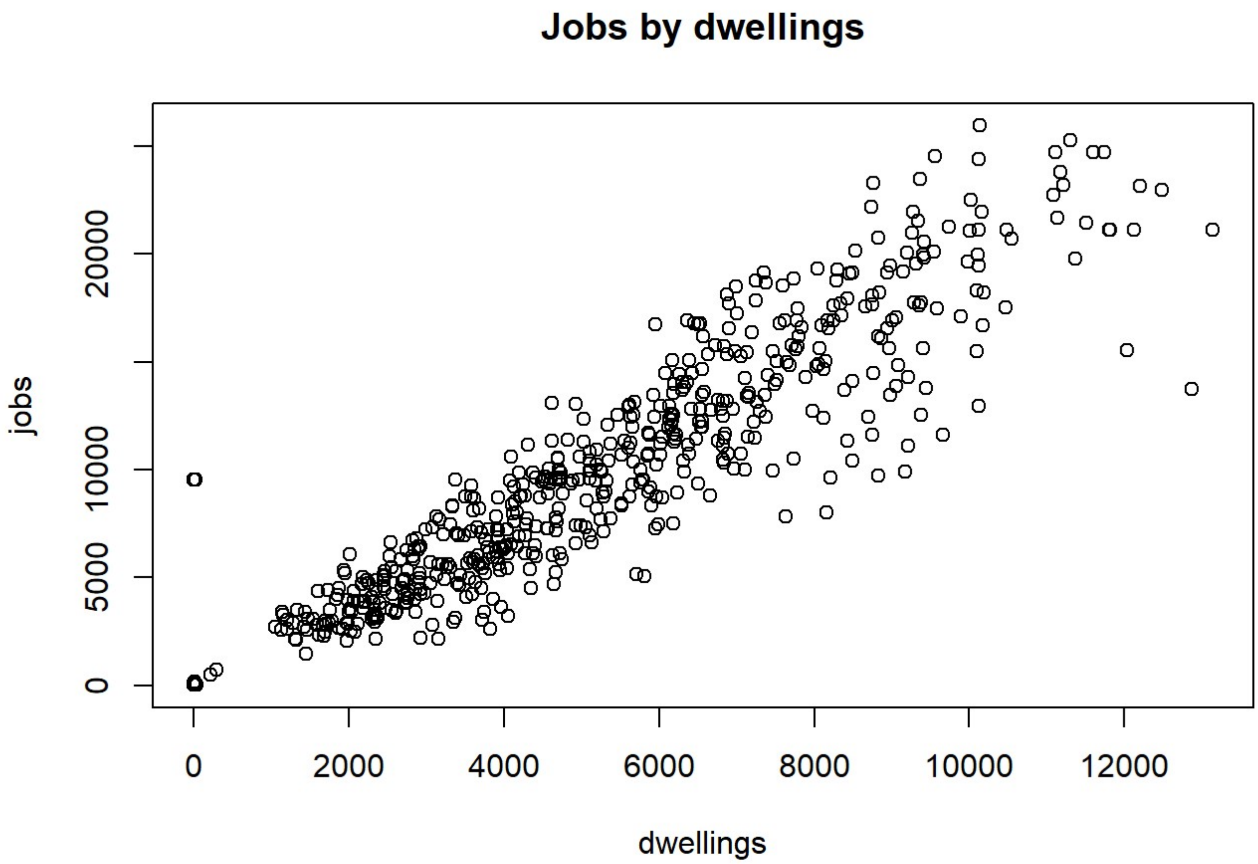
```
dwellings_capped <- full_data$`Total dwellings in SA2` %>% cap()
boxplot(as.numeric(dwellings_capped), main = "Box Plot of Total dwellings by region",
ylab = "dwellings", col = "red")
```




```
full_data$`Total dwellings in SA2` <- dwellings_capped
```

- Look for multivariate outliers by inspection:

```
full_data %>% plot(`no. of jobs`~ `Total dwellings in SA2`, data = ., ylab = "jobs", x
lab = "dwellings", main = "Jobs by dwellings")
```



- Look for multivariate outliers with the mvn package

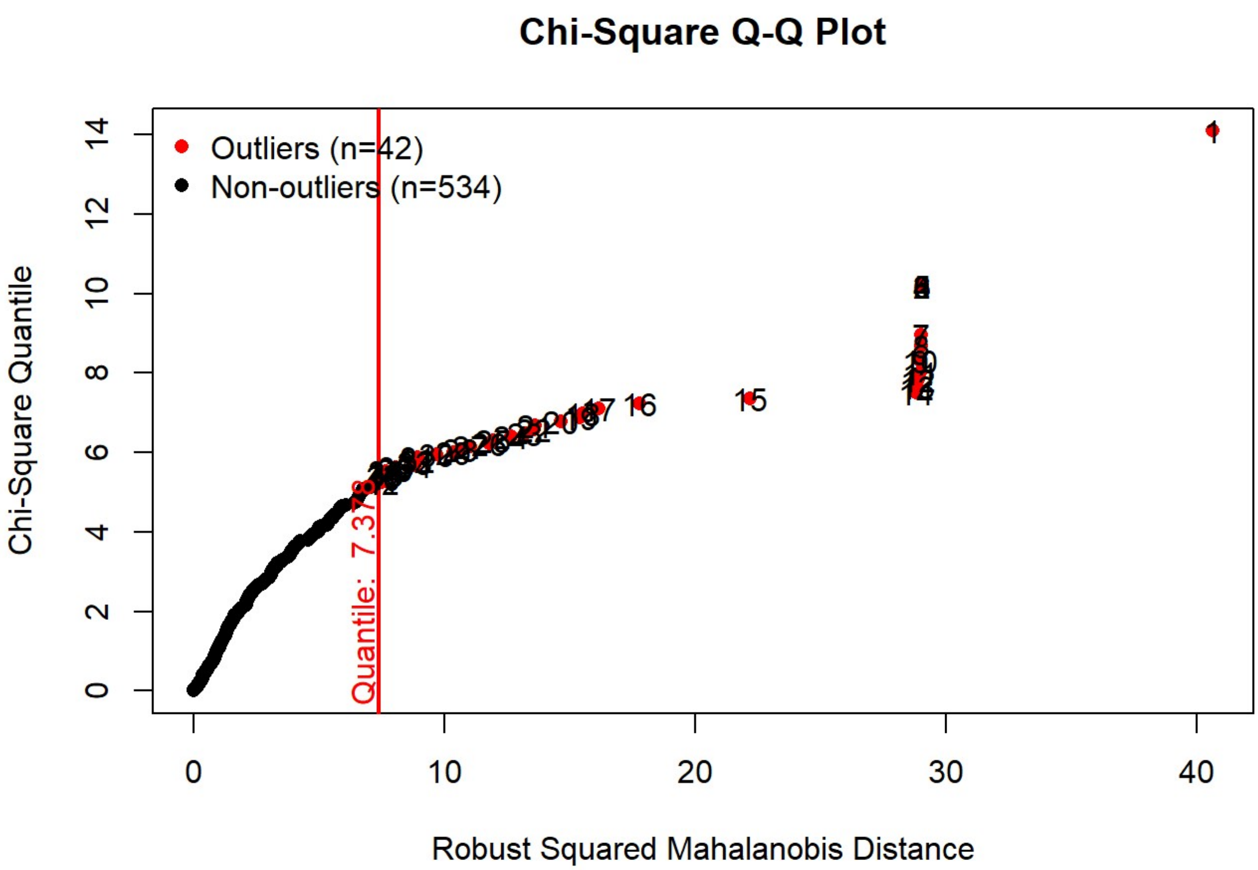
```
class(full_data)
```

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

```
colnames(full_data)
```

```
## [1] "SA2 NAME"                        "year"
## [3] "no. of jobs"                     "Income"
## [5] "Total dwellings in SA2"          "Most common mortgage repayments"
## [7] "Repayment reportings"
```

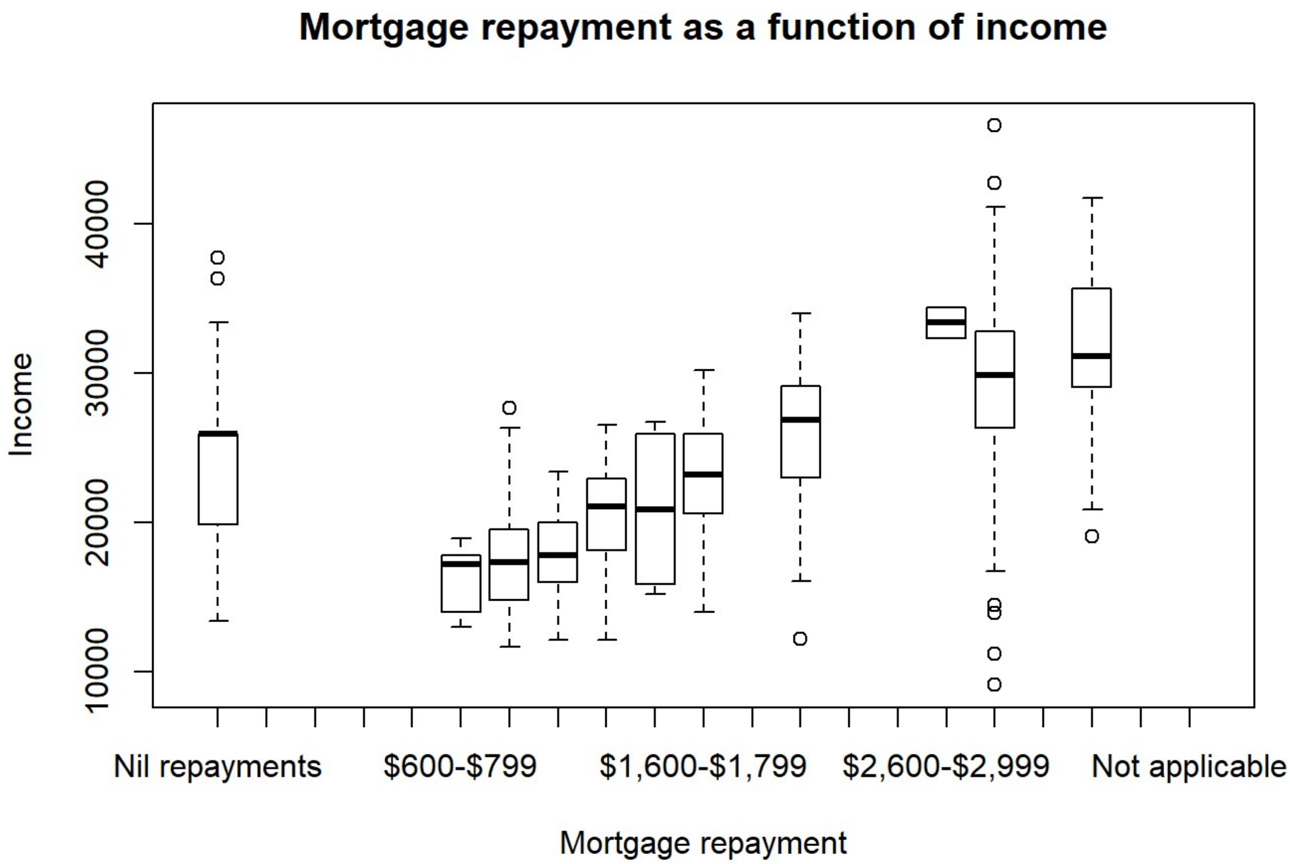
```
full_data_sub <- full_data %>% dplyr::select(`no. of jobs`, `Total dwellings in SA2`)
job_dwelling_clean <- mvn(data = full_data_sub, multivariateOutlierMethod = "quan", sh
owOutliers = TRUE, showNewData = TRUE)
```




```
full_data2 <- job_dwelling_clean$newData
head(full_data2)
```

```
##     no. of jobs Total dwellings in SA2
## 100        8760                   4208
## 101       13965                   7486
## 102        6784                   3760
## 103        4690                   2166
## 104        3442                   2037
## 105        4978                   2918
```

```
full_data %>% plot(Income ~ `Most common mortgage repayments`, data = ., ylab = "Incom
e", xlab = "Mortgage repayment", main = "Mortgage repayment as a function of income")
```



- Look for multivariate outliers with the mvn package

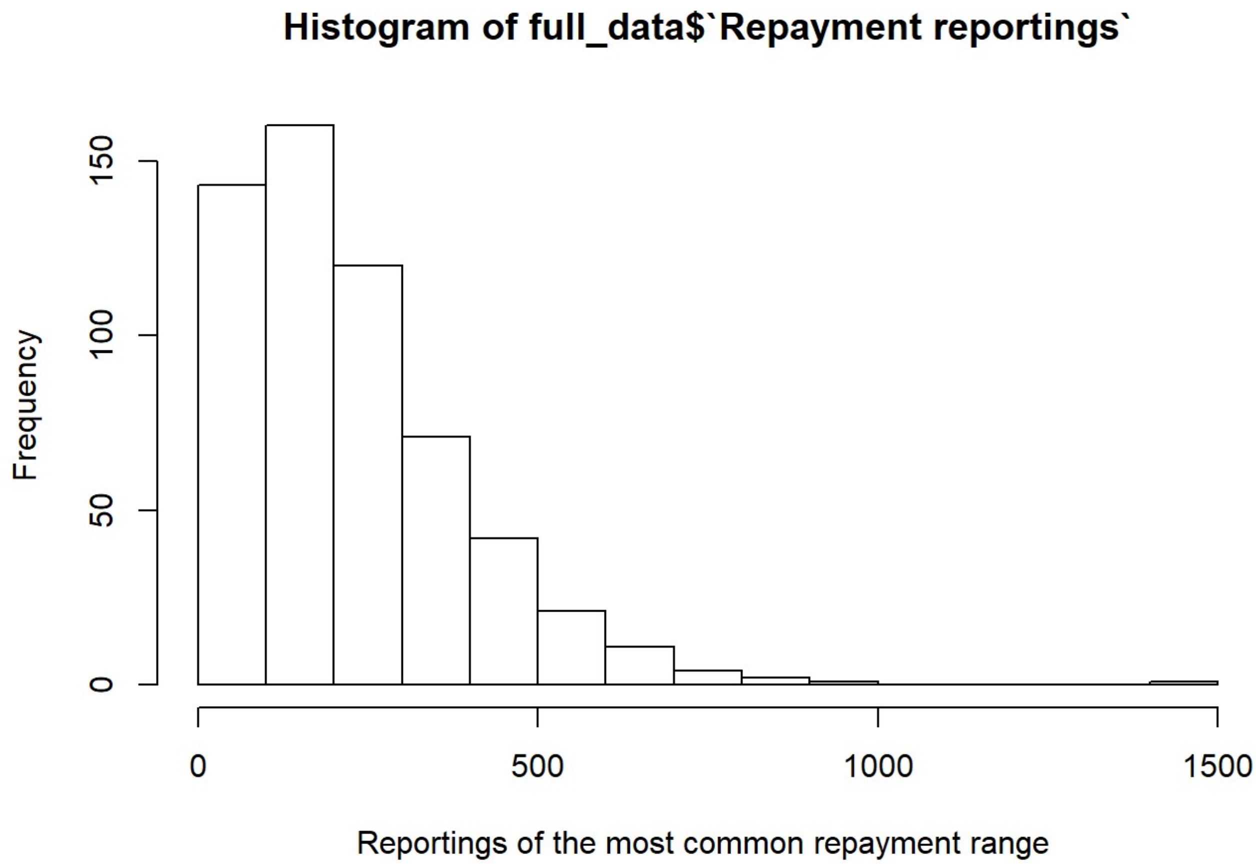
```
full_data_sub2 <- full_data %>% dplyr::select(Income, `Most common mortgage repayments
`)
#Income_Mortgage_clean <- mvn(data = full_data_sub2, multivariateOutlierMethod = "quan
", showOutliers = TRUE, showNewData = TRUE)
```

- Multivariate outliers detected between categorical variable (mortgage repayments) and numeric variable (Income). Futher preprocessing will be required to treat these.

Data transformations:

*Histogram of Repayment reporting numbers

```
hist(full_data$`Repayment reportings`, xlab = "Reportings of the most common repayment
range ")
```

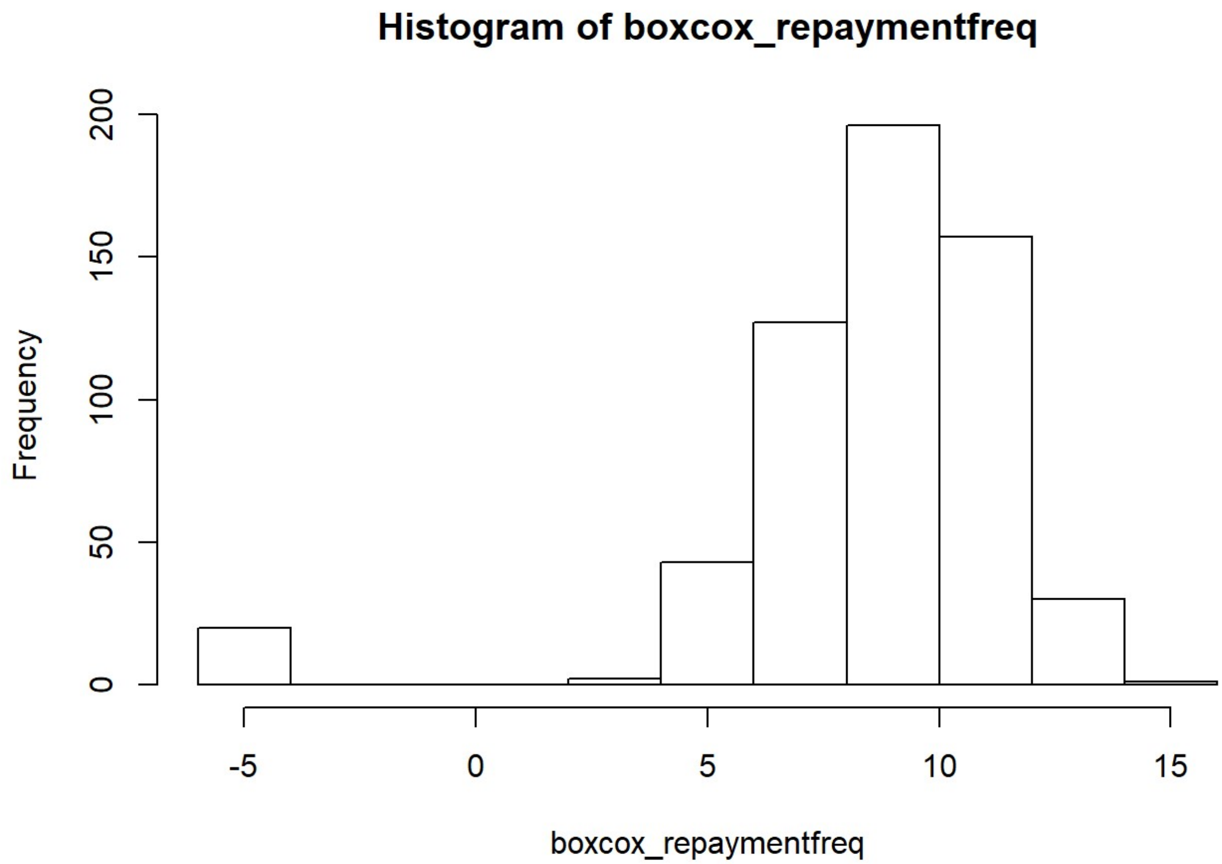


- The counts of the most common repayment option per region is positively skewed. It would be beneficial to transform the data,
- BoxCox transformation.

```
boxcox_repaymentfreq <- BoxCox(full_data$`Repayment reportings`, lambda = "auto")
head(boxcox_repaymentfreq)
```

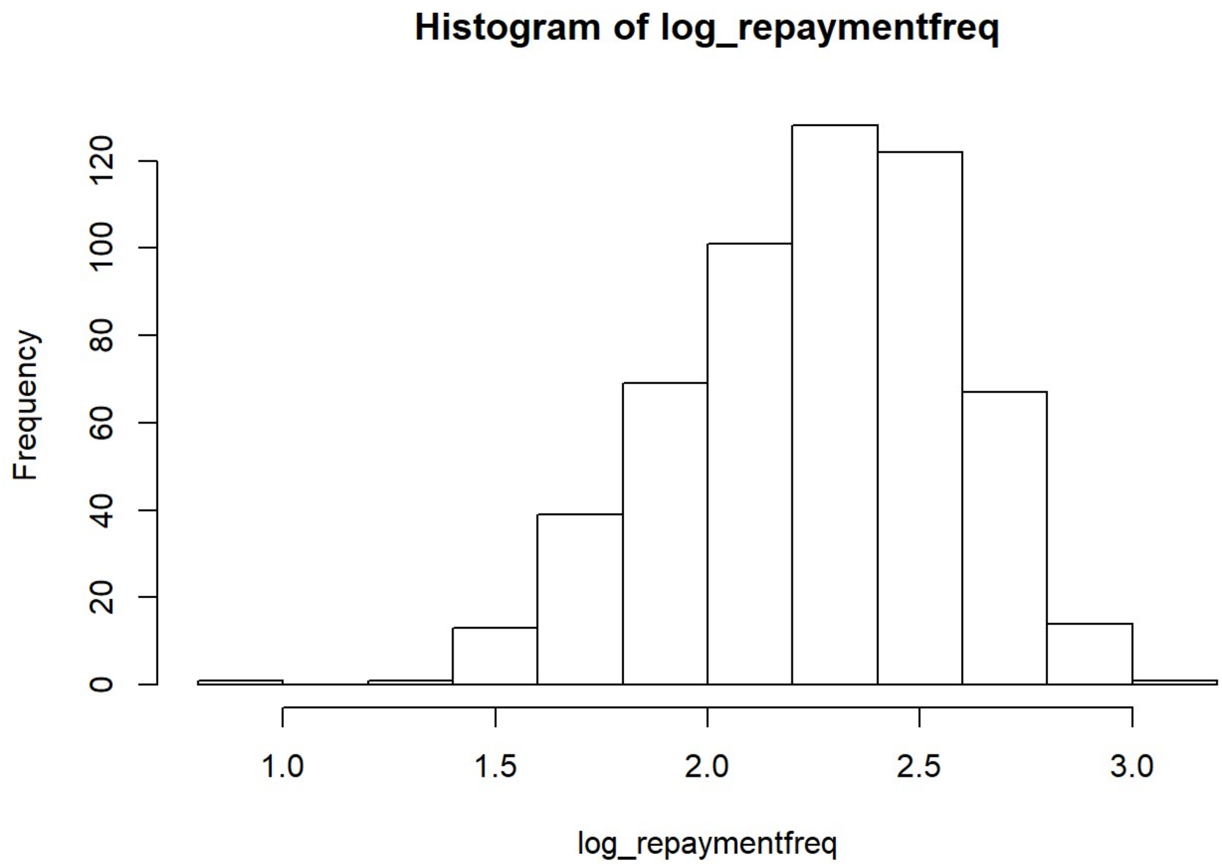
```
## [1]  5.429006  8.324734  8.525332  6.384187 12.353288 10.715659
```

```
hist(boxcox_repaymentfreq)
```



*log10 transformation

```
log_repaymentfreq <- log10(full_data$`Repayment reportings`)
hist(log_repaymentfreq)
```



Histogram of log_repaymentfreq
Frequency
0
20
40
60
80
100
120
1.0
1.5
2.0
2.5
3.0
log_repaymentfreq