ANZ@Data Task2

Sanjaya J Shetty

12/01/2021

```
### Load the library

library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(xlsx)
library(ggplot2)
library(fastDummies)
library(modelr)
library(rpart)
```

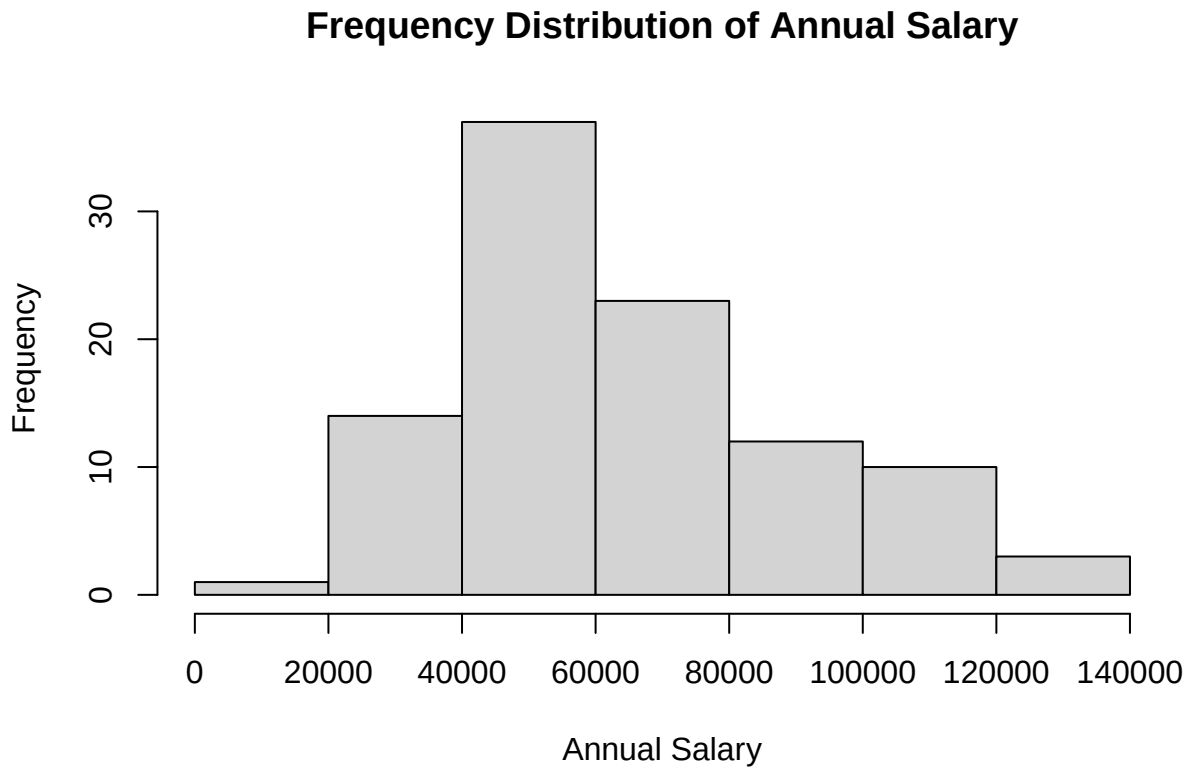
```
### load the data

data = read.xlsx('ANZ%20synthesised%20transaction%20dataset.xlsx; filename%2A.xlsx', as.data.frame = T,s
```

```
# Cleaning and pre-processing the data


data_ml <- filter(data, txn_description == 'PAY/SALARY')


# Finding number of unique values
```

```r
rapply(data_ml, function(x) length(unique(x)))

# Drop all the columns with just one entry and other irrelevant columns

data_ml <- data_ml[, c('account', 'first_name', 'balance', 'date', 'gender', 'age',
                       "amount", 'customer_id')]



# Calculate annual salary of each individual account holder

data_ml_sal <- data_ml[, c('account', 'balance', 'date',
                           "amount", 'customer_id','gender')]


data_ml_sal$date_diff <- 0
data_ml_sal$An_Salary <- 0


for (i in seq(nrow(data_ml_sal))){
  for (j in ((i+1): seq(nrow(data_ml_sal)))){
    if(i == j){
      next
    }
    if( j > 883 ){
      next
    }
    if(i !=j){
       if (data_ml_sal[i, 1]  == data_ml_sal[j,1]){
          data_ml_sal[i,7] <- abs(as.numeric(data_ml_sal[i,3]-data_ml_sal[j,3]))

          data_ml_sal[i,8] = data_ml_sal[i,4]*(364/abs(data_ml_sal[i,7]))
       }
      else{
        next
      }

    }
  }

}

# some account had some instance of recurring amount of salary on the same day, which resulted in Inf o
# 8 accounts, even though the the salary remained the same for previous time-frame and the next time-fr
# hence assuming that the recurring amount to be a clerical error or other early withdrawal of future s
# rather than a salary bump


## extract the annual salary of every account (ignore the Inf and zero value)

data_ml_sal[data_ml_sal == 0 | data_ml_sal == Inf] <-  NA

data_ml_sal_cleaned <- data_ml_sal[complete.cases(data_ml_sal),]
```

```r
data_ml_sal_cleaned <- data_ml_sal_cleaned[!duplicated(data_ml_sal_cleaned$account),]

# data_ml_sal_cleaned now contains the frequency of pay i.e, whether the payment of salary
# is done in weekly basis (7 days), bi weekly basis (14 days) etc.  and the annual salary

## checking all the various frequencies of pay

unique(data_ml_sal_cleaned$date_diff)

## the values contains 31, 30 and 32, this is because of the uneven distribution of number
## of days in a month.
```

```r
# frequency distribution of Salary

hist(data_ml_sal_cleaned$An_Salary, xlab = 'Annual Salary',
     main = 'Frequency Distribution of Annual Salary')
```




```r
## extracting spending from each account

data_ml_spend <- filter(data, txn_description != 'PAY/SALARY')

data_ml_spend_sum <- aggregate(amount~account+age+txn_description+gender, data_ml_spend, sum)
```

```r
# Merging spending pattern with annual salary

data_ml_Sal_Sp <- merge(data_ml_sal_cleaned, data_ml_spend_sum,  by = 'account')

# drop costumer ID, Balance, and date and amount.x columns

data_ml_Sal_Sp <- data_ml_Sal_Sp %>%
                            select(-date, -amount.x, -balance, -customer_id, -gender.y,-account)

head(data_ml_Sal_Sp)
```

```
##   gender.x date_diff An_Salary age txn_description amount.y
## 1        F         7  46388.68  40         PAYMENT   844.00
## 2        F         7  46388.68  40       SALES-POS  3445.86
## 3        F         7  46388.68  40             POS  3399.41
## 4        M        28  41535.13  22             POS   675.54
## 5        M        28  41535.13  22      INTER BANK   909.00
## 6        M        28  41535.13  22       SALES-POS  1299.99
```

```r
## Build a predictive model


## Linear Model

fit <- lm(formula = An_Salary~., data = data_ml_Sal_Sp)

summary(fit)
```

```
##
## Call:
## lm(formula = An_Salary ~ ., data = data_ml_Sal_Sp)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -40734 -19537  -3986  15382  66807
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                77965.911   5691.199  13.699  < 2e-16 ***
## gender.xM                   6177.084   2575.259   2.399 0.016946 *
## date_diff                   -713.799    155.681  -4.585  6.2e-06 ***
## age                         -137.097    115.413  -1.188 0.235632
## txn_descriptionPAYMENT    -10557.212   4115.710  -2.565 0.010704 *
## txn_descriptionPHONE BANK   2306.672   6947.112   0.332 0.740050
## txn_descriptionPOS         -8630.160   3978.729  -2.169 0.030706 *
## txn_descriptionSALES-POS   -8601.700   3998.100  -2.151 0.032081 *
## amount.y                      3.938      1.166   3.377 0.000811 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24930 on 374 degrees of freedom
## Multiple R-squared:  0.107,  Adjusted R-squared:  0.08792
## F-statistic: 5.603 on 8 and 374 DF,  p-value: 1.013e-06
```

```
rmse(fit, data_ml_Sal_Sp)
```

```
## [1] 24633.08
```

```
## Build a decision-tree based model


data_ml_Sal_Sp <- dummy_cols(data_ml_Sal_Sp, select_columns = c('txn_description', 'gender.x'),
                                  remove_selected_columns = T)


inTrain <- createDataPartition(data_ml_Sal_Sp, p = 0.70, list = F)
```

```
## Error: cannot allocate vector of size 4.1 Gb
```

```
data_train <- data_ml_Sal_Sp[inTrain,]
```

```
## Error in `[.data.frame`(data_ml_Sal_Sp, inTrain, ): object 'inTrain' not found
```

```
data_test <- data_ml_Sal_Sp[-inTrain,]
```

```
## Error in `[.data.frame`(data_ml_Sal_Sp, -inTrain, ): object 'inTrain' not found
```

```
fit_rpart <- rpart(An_Salary~., data = data_test, control = rpart.control(minsplit = 1, minbucket = 1,cp
```

```
## Error in is.data.frame(data): object 'data_test' not found
```

```
rmse(fit_rpart, data_train)
```

```
## Error in response_var(model): object 'fit_rpart' not found
```