BIN381 Milestone 2

Nosipho Precious Donkrag, Nontsikelelo Sharon Buhlungu, Tshepang Mogosi, Pitsi Pitsi

2024-10-12

Milestone 2: Data Fromatting and cleaning

libraries:

```
library(ggplot2)
```

Column headers

Form data understanding it was seen that the column names have been shifted resulting in some columns having 'NA' column names. Hence, when the csv is read it will be given default column names.

```
setwd("C:/Users/nosip/Documents/third Year/BIN381/milestones")
data <- read.csv("CustData2.csv",
header = FALSE,
na.strings = c("", "NA"),
fill = TRUE)
#Give columns a default name names
colnames(data) <- paste0("Default_col_Name", seq_len(ncol(data)))
```

The actual column names are stored in the second row of the dataset. these will be extracted next and the default column names will be replacced with the actual column names.

```
new_col_names <- as.character(data[1, ])
# drop the first row

data <- data[-1, ]
#replace the default column names with the correct ones
colnames(data) <- new_col_names
head(data, n=5)
```

```
##   ;Last.Name;First.Name;Middle.Initial;Title;Department.Name;Annual.Salary;Gross.Pay.Last.Paycheck;G
## 2                                                                                                1;
## 3
## 4                                                                                                 3
## 5
## 6                                                                                  5;THOMAS;MICHA
##   marital_status street_address postal_code        city state_province
## 2       619.76;2      501.62;48   025.48;46 616.58;1976        married
## 3       250.38;3      467.63;57   932.07;56 222.79;1964           <NA>
## 4       393.76;4      513.71;49   968.35;48 501.19;1942         single
## 5       735.10;1      561.67;35   469.59;34 432.85;1977        married
## 6       386.40;6     665.66;132  850.76;128 948.86;1949           <NA>
##                     Country_id phone_number        email      Education
## 2 27 North Sagadahoc Boulevard        60332          Ede     Gelderland
```

```
## 3        37 West Geneva Street        55406    Hoofddorp  Noord-Holland
## 4             47 Toa Alta Road        34077    Schimmert        Limburg
## 5        47 South Kanabec Road        72996 Scheveningen   Zuid-Holland
## 6           57 North 3rd Drive        67644    Joinville Santa Catarina
##   Occupation household_size     yrs_residence      NA    NA NA
## 2      52770   519-236-6123 Ruddy@company.com Masters Prof.  2
## 3      52770   327-194-5008 Ruddy@company.com Masters Prof.  2
## 4      52770   288-613-9676 Ruddy@company.com Masters Prof.  2
## 5      52770   222-269-1259 Ruddy@company.com Masters Prof.  2
## 6      52775   675-133-2226 Ruddy@company.com Masters Prof.  2
##                                                                                  NA
## 2           4
## 3      4
## 4 4
## 5               4
## 6    4
```

Correct column Mapping:

A mapping was created from data understanding that maps the shifted columns to their correct data.

```
##    Current.colum.name Correct.colum.name
## 1      state_province     marital_status
## 2          Country_id     street_address
## 3        phone_number        postal_code
## 4               email               city
## 5           Education     state_province
## 6          Occupation         Country_id
## 7      household_size       phone_number
## 8       yrs_residence              email
## 9                <NA>          Education
## 10               <NA>         Occupation
## 11               <NA>     household_size
## 12               <NA>      yrs_residence
```

create a new dataframe with the correct columns:

```r
data_df <- data.frame(
  marital_status = data$state_province,
  street_address   = data$Country_id,
  postal_code      = data$phone_number,
  city             = data$email,
  state_province   = data$Education,
  Country_id       = data$Occupation,
  phone_number     = data$household_size,
  email            = data$yrs_residence,
    Education        = data[[14]],
  Occupation       = data[[15]],
  household_size   = data[[16]],
  yrs_of_residence    = data[[17]]

)
head(data_df, n = 3)
```

```
##   marital_status              street_address postal_code      city
## 1        married 27 North Sagadahoc Boulevard       60332       Ede
```

```
## 2           <NA>        37 West Geneva Street       55406 Hoofddorp
## 3         single             47 Toa Alta Road       34077 Schimmert
##   state_province Country_id phone_number             email Education Occupation
## 1     Gelderland      52770 519-236-6123 Ruddy@company.com   Masters      Prof.
## 2  Noord-Holland      52770 327-194-5008 Ruddy@company.com   Masters      Prof.
## 3        Limburg      52770 288-613-9676 Ruddy@company.com   Masters      Prof.
##   household_size
## 1              2
## 2              2
## 3              2
##                                                                          yrs_of_residence
## 1           4
## 2      4
## 3 4
```

The above columns have been correctly associated with their data and stored in the new data frame: data_df.

The long column name (column 1)

The first column name in the data holds the column names of multiple columns. here are the column names in the first column of the data:

```
long_col_name <- colnames(data)[1]

split_names <- strsplit(long_col_name, ";")[[1]]
cat(split_names, sep = "\n")
```

```
##
## Last.Name
## First.Name
## Middle.Initial
## Title
## Department.Name
## Annual.Salary
## Gross.Pay.Last.Paycheck
## Gross.Year.To.Date
## Gross.Year.To.Date...FRS.Contribution
## year_of_birth
```

Create new columns from the first column

```
#split the data using a sep: ;
split_data <- strsplit(data[,1], ";")

#convert into a dataframe
split_df <- do.call(rbind, split_data)
```

```
## Warning in (function (..., deparse.level = 1) : number of columns of result is
## not a multiple of vector length (arg 1)
```

The above warning suggests that, in the semi-colon separated data there is some missing values; the semi-colon separated data in the first column holds the data of the following columns: (if there are missing values the data will not be symmetrical as it is being read).

- id
- Last.Name
- First.Name

- Middle.Initial
- Title
- Department.Name
- Age

all these variables are unique and thus will not be used to train the machine learning model. the only variable that could be of interest is Age; however the age can be calculated from the column holding the year of birth. hence, to avoid offsetting the data, as the warning suggests this will happen, this whole column can be thrown away.

Numeric columns

the first numeric column holds the annual salary and the months related to the annual salary, seperated by a semi-colomn. these will be extracted.

```
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
```

```
#replace NA values with NA,NA to help with the splitting
data[[names(data)[2]]] <- ifelse(is.na(data[[names(data)[2]]]), "NA;NA", data[[names(data)[2]]])
```

split the values in this column into annual salary and annual months (months related to the annual salary).

```
split_values <- strsplit(data[[names(data)[2]]], ";")
```

```
#Inconsistent entries
problematic_entries <- which(lengths(split_values) != 2)
#data[problematic_entries, names(data)[2]]
head(length(problematic_entries))
```

```
## [1] 13567
```

There is inconsistent data that will affect the splitting, this data will be dropped from data.

```
valid_indices <- which(lengths(split_values) == 2)
data <- data[valid_indices, ]
```

size of new data:

```
print(nrow(data))
```

```
## [1] 187448
```

```
split_values <- strsplit(data[[names(data)[2]]], ";")
split_matrix <- matrix(unlist(split_values), ncol = 2, byrow = TRUE)
split_df <- as.data.frame(split_matrix, stringsAsFactors = FALSE)
```

```
colnames(split_df) <- c("Annual_Salary", "Months_Annual")
```

view the first 5 entries in split_df:

```
head(split_df, n = 5)
```

```
##   Annual_Salary Months_Annual
## 1        619.76             2
## 2        250.38             3
## 3        393.76             4
## 4        735.10             1
## 5        386.40             6
```

To ensure consistency drop these rows from data_df.

```
data_df <- data_df[valid_indices, ]
nrow(data)
```

```
## [1] 187448
```

```
nrow(data_df)
```

```
## [1] 187448
```

combine the 2 data frames.

```
data_df <- cbind(data_df, split_df)
```

```
head(data_df, n = 5)
```

```
##   marital_status               street_address postal_code         city
## 1        married 27 North Sagadahoc Boulevard       60332          Ede
## 2           <NA>        37 West Geneva Street       55406    Hoofddorp
## 3         single             47 Toa Alta Road       34077    Schimmert
## 4        married        47 South Kanabec Road       72996 Scheveningen
## 5           <NA>           57 North 3rd Drive       67644    Joinville
##   state_province Country_id phone_number             email Education Occupation
## 1     Gelderland      52770 519-236-6123 Ruddy@company.com   Masters      Prof.
## 2  Noord-Holland      52770 327-194-5008 Ruddy@company.com   Masters      Prof.
## 3        Limburg      52770 288-613-9676 Ruddy@company.com   Masters      Prof.
## 4   Zuid-Holland      52770 222-269-1259 Ruddy@company.com   Masters      Prof.
## 5 Santa Catarina      52775 675-133-2226 Ruddy@company.com   Masters      Prof.
##   household_size
## 1              2
## 2              2
## 3              2
## 4              2
## 5              2
##                                                                      yrs_of_residence
## 1           4
## 2      4
## 3 4
## 4               4
## 5    4
##   Annual_Salary Months_Annual
## 1        619.76             2
## 2        250.38             3
## 3        393.76             4
## 4        735.10             1
## 5        386.40             6
```

FRS.Contribution and Year of birth

the data in the column number 5, is the FRS contributions and year of birth.

```
print(data[1:5, 5])
```

```
## [1] "616.58;1976" "222.79;1964" "501.19;1942" "432.85;1977" "948.86;1949"
```

```
split_frs_yob <- strsplit(data[[5]], ";")
valid_rows <- sapply(split_frs_yob, function(x) length(x) == 2)
```

```
print(length(valid_rows))
```

```
## [1] 187448
```

drop inconsistent rows

```
data <- data[valid_rows, ]
data_df <- data_df[valid_rows, ]
```

Check if the data frames are still aligned.

```
nrow(data)
```

```
## [1] 177874
```

```
nrow(data_df)
```

```
## [1] 177874
```

```
split_frs_yob_matrix <- matrix(unlist(split_frs_yob[valid_rows]), ncol = 2, byrow = TRUE)
data_df$FRS.Contribution <- split_frs_yob_matrix[, 1]
data_df$Year_of_Birth <- split_frs_yob_matrix[, 2]
```

```
head(data_df)
```

```
##   marital_status               street_address postal_code         city
## 1        married 27 North Sagadahoc Boulevard       60332          Ede
## 2           <NA>        37 West Geneva Street       55406    Hoofddorp
## 3         single             47 Toa Alta Road       34077    Schimmert
## 4        married        47 South Kanabec Road       72996 Scheveningen
## 5           <NA>           57 North 3rd Drive       67644    Joinville
## 6         single      67 East Mcintosh Avenue       83786       Nagoya
##   state_province Country_id phone_number             email Education Occupation
## 1     Gelderland      52770 519-236-6123 Ruddy@company.com   Masters      Prof.
## 2  Noord-Holland      52770 327-194-5008 Ruddy@company.com   Masters      Prof.
## 3        Limburg      52770 288-613-9676 Ruddy@company.com   Masters      Prof.
## 4   Zuid-Holland      52770 222-269-1259 Ruddy@company.com   Masters      Prof.
## 5 Santa Catarina      52775 675-133-2226 Ruddy@company.com   Masters      Prof.
## 6          Aichi      52782 183-207-2933 Ruddy@company.com   Masters      Prof.
##   household_size
## 1              2
## 2              2
## 3              2
## 4              2
## 5              2
## 6              2
##                                                                      yrs_of_residence
## 1           4
## 2      4
```

```
## 3 4
## 4               4
## 5    4
## 6   4
##   Annual_Salary Months_Annual FRS.Contribution Year_of_Birth
## 1        619.76             2           616.58          1976
## 2        250.38             3           222.79          1964
## 3        393.76             4           501.19          1942
## 4        735.10             1           432.85          1977
## 5        386.40             6           948.86          1949
## 6        621.22             3           047.65          1950
```

```
#nrow(data_df)
```

Net salary column

the net salary to date column:

```
split_net_months <- strsplit(data[[3]], ";")
valid_rows <- sapply(split_net_months, function(x) length(x) == 2)
```

```
print(sum(valid_rows))
```

```
## [1] 171390
```

drop inconsistent rows

```
data <- data[valid_rows, ]
data_df <- data_df[valid_rows, ]
```

```
nrow(data)
```

```
## [1] 171390
```

```
nrow(data_df)
```

```
## [1] 171390
```

```
split_net_nMonths_matrix <- matrix(unlist(split_net_months[valid_rows]), ncol = 2, byrow = TRUE)

data_df$Net_Salary <- split_net_nMonths_matrix[, 1]
data_df$Net_months <- split_net_nMonths_matrix[, 2]
```

```
nrow(data_df)
```

```
## [1] 171390
```

```
head(data_df[16:18])
```

```
##   Year_of_Birth Net_Salary Net_months
## 1          1976     501.62         48
## 2          1964     467.63         57
## 3          1942     513.71         49
## 4          1977     561.67         35
## 5          1949     665.66        132
## 6          1950     802.71         97
```

the annual salary, net salary and the FRS contributions have been accounted for, there is still the gross salary to go. we expect the gross to be the highest for the 3 numeric columns.

```r
print(data[1:10,4])
```

```
##  [1] "025.48;46"  "932.07;56"  "968.35;48"  "469.59;34"  "850.76;128"
##  [6] "945.90;95"  "182.33;173" "738.62;44"  "025.77;65"  "574.10;22"
```

```r
split_gross_months <- strsplit(data[[4]], ";")

valid_gross_rows <- sapply(split_gross_months, function(x) length(x) == 2)

print(sum(valid_gross_rows))
```

```
## [1] 171390
```

```r
data <- data[valid_gross_rows, ]
data_df <- data_df[valid_gross_rows, ]
```

```r
nrow(data)
```

```
## [1] 171390
```

```r
nrow(data_df)
```

```
## [1] 171390
```

```r
split_gross_matrix <- matrix(unlist(split_gross_months[valid_gross_rows]), ncol = 2, byrow = TRUE)

data_df$Gross_Salary <- split_gross_matrix[, 1]
data_df$Gross_Months <- split_gross_matrix[, 2]
```

```r
head(data_df)
```

```
##   marital_status               street_address postal_code         city
## 1        married 27 North Sagadahoc Boulevard       60332          Ede
## 2           <NA>        37 West Geneva Street       55406    Hoofddorp
## 3         single             47 Toa Alta Road       34077    Schimmert
## 4        married        47 South Kanabec Road       72996 Scheveningen
## 5           <NA>           57 North 3rd Drive       67644    Joinville
## 6         single      67 East Mcintosh Avenue       83786       Nagoya
##   state_province Country_id phone_number             email Education Occupation
## 1     Gelderland      52770 519-236-6123 Ruddy@company.com   Masters      Prof.
## 2  Noord-Holland      52770 327-194-5008 Ruddy@company.com   Masters      Prof.
## 3        Limburg      52770 288-613-9676 Ruddy@company.com   Masters      Prof.
## 4   Zuid-Holland      52770 222-269-1259 Ruddy@company.com   Masters      Prof.
## 5 Santa Catarina      52775 675-133-2226 Ruddy@company.com   Masters      Prof.
## 6          Aichi      52782 183-207-2933 Ruddy@company.com   Masters      Prof.
##   household_size
## 1              2
## 2              2
## 3              2
## 4              2
## 5              2
## 6              2
##                                                                      yrs_of_residence
## 1           4
## 2      4
## 3 4
## 4               4
```

```
## 5    4
## 6   4
##   Annual_Salary Months_Annual FRS.Contribution Year_of_Birth Net_Salary
## 1        619.76             2           616.58          1976     501.62
## 2        250.38             3           222.79          1964     467.63
## 3        393.76             4           501.19          1942     513.71
## 4        735.10             1           432.85          1977     561.67
## 5        386.40             6           948.86          1949     665.66
## 6        621.22             3           047.65          1950     802.71
##   Net_months Gross_Salary Gross_Months
## 1         48       025.48           46
## 2         57       932.07           56
## 3         49       968.35           48
## 4         35       469.59           34
## 5        132       850.76          128
## 6         97       945.90           95
```

```r
names(data_df)
```

```
##  [1] "marital_status"   "street_address"   "postal_code"      "city"
##  [5] "state_province"   "Country_id"       "phone_number"     "email"
##  [9] "Education"        "Occupation"       "household_size"   "yrs_of_residence"
## [13] "Annual_Salary"    "Months_Annual"    "FRS.Contribution" "Year_of_Birth"
## [17] "Net_Salary"       "Net_months"       "Gross_Salary"     "Gross_Months"
```

Convert Expected numeric columns to numeric:

```r
numeric_columns <- c("Annual_Salary", "Months_Annual", "FRS.Contribution",
                     "Year_of_Birth", "Net_Salary", "Net_months",
                     "Gross_Salary", "Gross_Months",
                     "household_size", "yrs_of_residence", "postal_code")

data_df[numeric_columns] <- lapply(data_df[numeric_columns], function(x) as.numeric(as.character(x)))
```

```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
```

```r
# Check the structure of the data to verify conversion
summary(data_df)
```

```
##  marital_status     street_address      postal_code        city
##  Length:171390      Length:171390      Min.   :30000   Length:171390
##  Class :character   Class :character   1st Qu.:45704   Class :character
##  Mode  :character   Mode  :character   Median :60994   Mode  :character
##                                        Mean   :60620
##                                        3rd Qu.:75023
##                                        Max.   :92330
##                                        NA's   :7426
##  state_province      Country_id        phone_number          email
##  Length:171390      Length:171390      Length:171390      Length:171390
##  Class :character   Class :character   Class :character   Class :character
```

```
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   Education          Occupation        household_size yrs_of_residence
##  Length:171390      Length:171390      Min.   :2.00   Min.   :2.000
##  Class :character   Class :character   1st Qu.:2.00   1st Qu.:2.000
##  Mode  :character   Mode  :character   Median :2.00   Median :3.000
##                                        Mean   :2.13   Mean   :3.208
##                                        3rd Qu.:2.00   3rd Qu.:4.000
##                                        Max.   :3.00   Max.   :5.000
##                                        NA's   :8483   NA's   :109
##  Annual_Salary    Months_Annual     FRS.Contribution  Year_of_Birth
##  Min.   :   0.0   Min.   :  1.000   Min.   :   0.08   Min.   :   1
##  1st Qu.: 255.4   1st Qu.:  2.000   1st Qu.: 253.01   1st Qu.:1945
##  Median : 492.6   Median :  2.000   Median : 498.91   Median :1955
##  Mean   : 493.1   Mean   :  4.571   Mean   : 500.75   Mean   :1874
##  3rd Qu.: 740.2   3rd Qu.:  4.000   3rd Qu.: 749.78   3rd Qu.:1969
##  Max.   :1000.0   Max.   :156.000   Max.   : 999.98   Max.   :1990
##  NA's   :7426     NA's   :72
##    Net_Salary       Net_months      Gross_Salary     Gross_Months
##  Min.   :   0.0   Min.   :  1.00   Min.   :   0.0   Min.   :  1.00
##  1st Qu.: 268.2   1st Qu.: 39.00   1st Qu.: 251.6   1st Qu.: 39.00
##  Median : 506.5   Median : 57.00   Median : 495.4   Median : 56.00
##  Mean   : 506.5   Mean   : 60.59   Mean   : 498.2   Mean   : 60.55
##  3rd Qu.: 746.5   3rd Qu.: 80.00   3rd Qu.: 744.5   3rd Qu.: 78.00
##  Max.   :1000.0   Max.   :322.00   Max.   :1000.0   Max.   :322.00
##  NA's   :72
```

The above code has introduced NA values, these will be dealt with accordingly.

Missing Values

Lets view the total number of missing values for each column:

```
missing_values_summary <- list()

for (col_name in names(data_df)) {
  total_missing <- sum(is.na(data_df[[col_name]]))

  missing_values_summary[[col_name]] <- total_missing
}

missing_values_df <- data.frame(
  Column = names(missing_values_summary),
  Total_Missing_Values = unlist(missing_values_summary)
)

print(missing_values_df)
```

```
##                           Column Total_Missing_Values
## marital_status     marital_status                52185
## street_address     street_address                 2281
## postal_code           postal_code                 7426
```

```
## city                          city                    0
## state_province     state_province                     0
## Country_id             Country_id                     0
## phone_number         phone_number                     0
## email                       email                     0
## Education               Education                    0
## Occupation              Occupation                   0
## household_size     household_size                 8483
## yrs_of_residence yrs_of_residence                  109
## Annual_Salary       Annual_Salary                 7426
## Months_Annual       Months_Annual                   72
## FRS.Contribution FRS.Contribution                    0
## Year_of_Birth       Year_of_Birth                    0
## Net_Salary             Net_Salary                   72
## Net_months             Net_months                    0
## Gross_Salary         Gross_Salary                    0
## Gross_Months         Gross_Months                    0
```

Imputation of missing values

Continuous columns that will be filled with the mean: - Annual_Salary - Months_Annual (12.5 months is a valid entry and refers to 1 year and 6 months hence it is continuous) - Net_Salary

Distinct columns that will be filled with the mode: - yrs_of_residence - Education - household_size - marital_status (mist convert to numeric first)

```
data_df$marital_status[1:10]
```

```
##  [1] "married"     NA            "single"      "married"     NA
##  [6] "single"      "married"     NA            "single"      "378.11;1951"
```

this column still has inconsistent values that need to be removed:

```
valid_marital_status <- c("married", "single", "divorced", NA)

# valid marital status
valid_rows <- data_df$marital_status %in% valid_marital_status

# Drop inconsistent rows
data_df <- data_df[valid_rows, ]
```

```
# Check the result to see if there is still enough
#data to train the model
nrow(data_df)
```

```
## [1] 152062
```

convert this column to numeric

```
marital_mapping <- c("married" = 2, "single" = 1)

data_df$marital_status <- as.numeric(factor(data_df$marital_status,
                                            levels = names(marital_mapping),
                                            labels = marital_mapping))
```

```
# Check the result
head(data_df[, c("marital_status")], n = 5)
```

```
## [1]  1 NA  2  1 NA
```

replace missing values with the mode

```
mode_marital_status <- data_df %>%
  summarise(mode = as.numeric(names(which.max(table(marital_status))))) %>%
  pull(mode)
```

```
#Replace NA values
data_df$marital_status[is.na(data_df$marital_status)] <- mode_marital_status
```

```
head(data_df[, c("marital_status")], n = 5)
```

```
## [1] 1 2 2 1 2
```

Replace missing values with the mean for the following columns:

- "Annual_Salary"
- "Months_Annual"
- "Net_Salary"

```
# Function to calculate mean and replace NA values
replace_na_with_mean <- function(column) {
  mean_value <- mean(column, na.rm = TRUE)
  column[is.na(column)] <- mean_value
  return(column)
}

# Apply to the numeric columns
data_df$Annual_Salary <- replace_na_with_mean(data_df$Annual_Salary)
data_df$Months_Annual <- replace_na_with_mean(data_df$Months_Annual)
data_df$Net_Salary <- replace_na_with_mean(data_df$Net_Salary)
```

```
head(data_df[, c("Annual_Salary", "Months_Annual", "Net_Salary")], n = 5)
```

```
##   Annual_Salary Months_Annual Net_Salary
## 1        619.76             2     501.62
## 2        250.38             3     467.63
## 3        393.76             4     513.71
## 4        735.10             1     561.67
## 5        386.40             6     665.66
```

Replace missing values with the mode for the following columns:

- "household_size"
- "Education"
- "yrs_of_residence"

```
calculate_mode <- function(column) {
  as.numeric(names(which.max(table(column))))
}


data_df$yrs_of_residence[is.na(data_df$yrs_of_residence)] <- calculate_mode(data_df$yrs_of_residence)
data_df$Education[is.na(data_df$Education)] <- calculate_mode(data_df$Education)
```

```
## Warning in calculate_mode(data_df$Education): NAs introduced by coercion
```

```
data_df$household_size[is.na(data_df$household_size)] <- calculate_mode(data_df$household_size)
```

```r
head(data_df[, c("yrs_of_residence", "Education", "household_size")], n = 5)
```

```
##   yrs_of_residence Education household_size
## 1                4   Masters              2
## 2                4   Masters              2
## 3                4   Masters              2
## 4                4   Masters              2
## 5                4   Masters              2
```

NA are introduced because of inconsistent data, so we have to check for them:

```r
sum(is.na(data_df$yrs_of_residence))
```

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

```r
sum(is.na(data_df$Education))
```

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

```r
sum(is.na(data_df$household_size))
```

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

Viewing the final Structure of the data

```r
data_types <- data.frame(
  Column = names(data_df),
  Data_Type = sapply(data_df, class)
)

# Print the table of data types
print(data_types)
```

```
##                           Column Data_Type
## marital_status     marital_status   numeric
## street_address     street_address character
## postal_code           postal_code   numeric
## city                         city character
## state_province     state_province character
## Country_id             Country_id character
## phone_number         phone_number character
## email                       email character
## Education               Education character
## Occupation             Occupation character
## household_size     household_size   numeric
## yrs_of_residence yrs_of_residence   numeric
## Annual_Salary       Annual_Salary   numeric
## Months_Annual       Months_Annual   numeric
## FRS.Contribution FRS.Contribution   numeric
## Year_of_Birth       Year_of_Birth   numeric
## Net_Salary             Net_Salary   numeric
## Net_months             Net_months   numeric
## Gross_Salary         Gross_Salary   numeric
## Gross_Months         Gross_Months   numeric
```

Target variable

The data is almost cleaned, it is missing the target variable, from the milestone outline the target will be numeric (1 or 0) and will be calculated as following: - if customer earns more than 50000 then qualify = 1 - else: qualify = 0.

The "Net_Salary", "Annual_Salary", "Gross_Salary" will be explored to identify the most suitable variable to calculate the target.

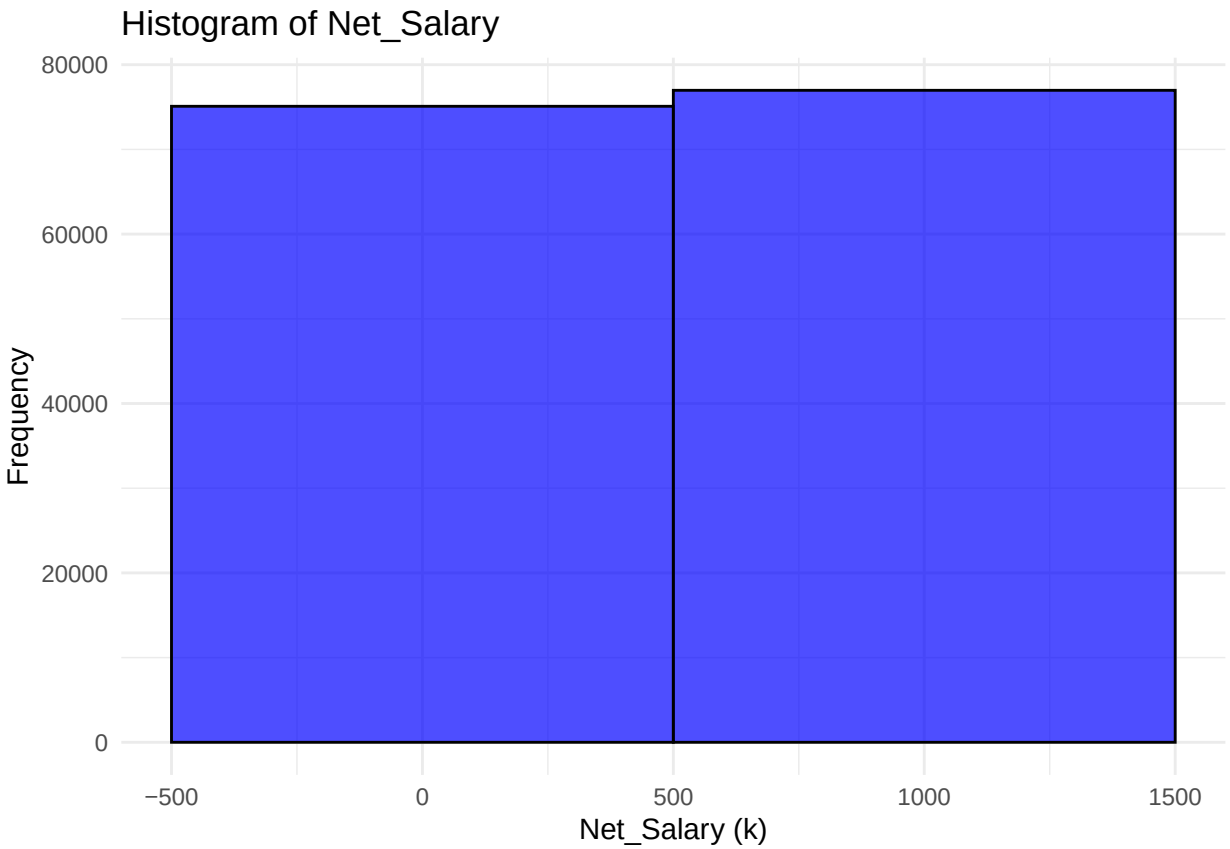
Histogram of the Net salary:

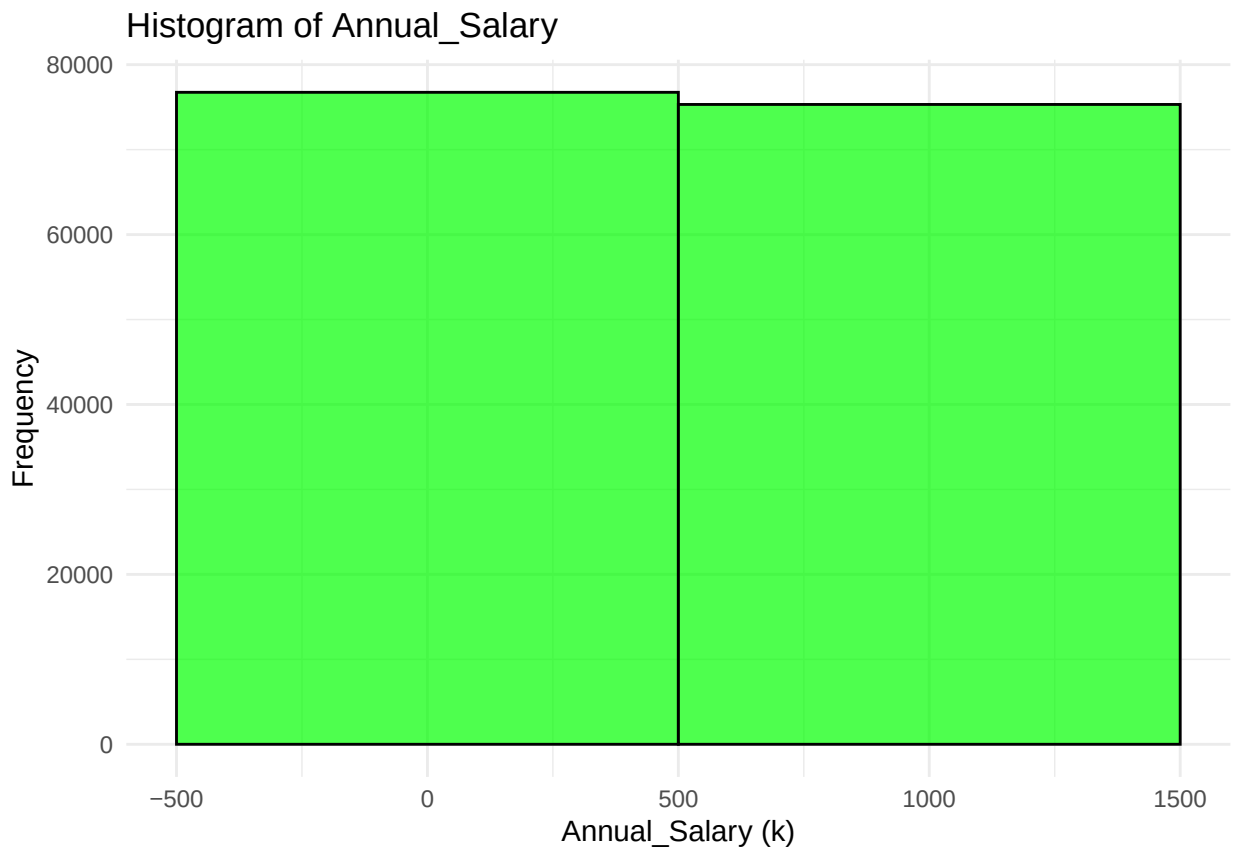
```
data_df$Net_Salary <- abs(data_df$Net_Salary)
data_df$Annual_Salary <- abs(data_df$Annual_Salary)
data_df$Gross_Salary <- abs(data_df$Gross_Salary)
```

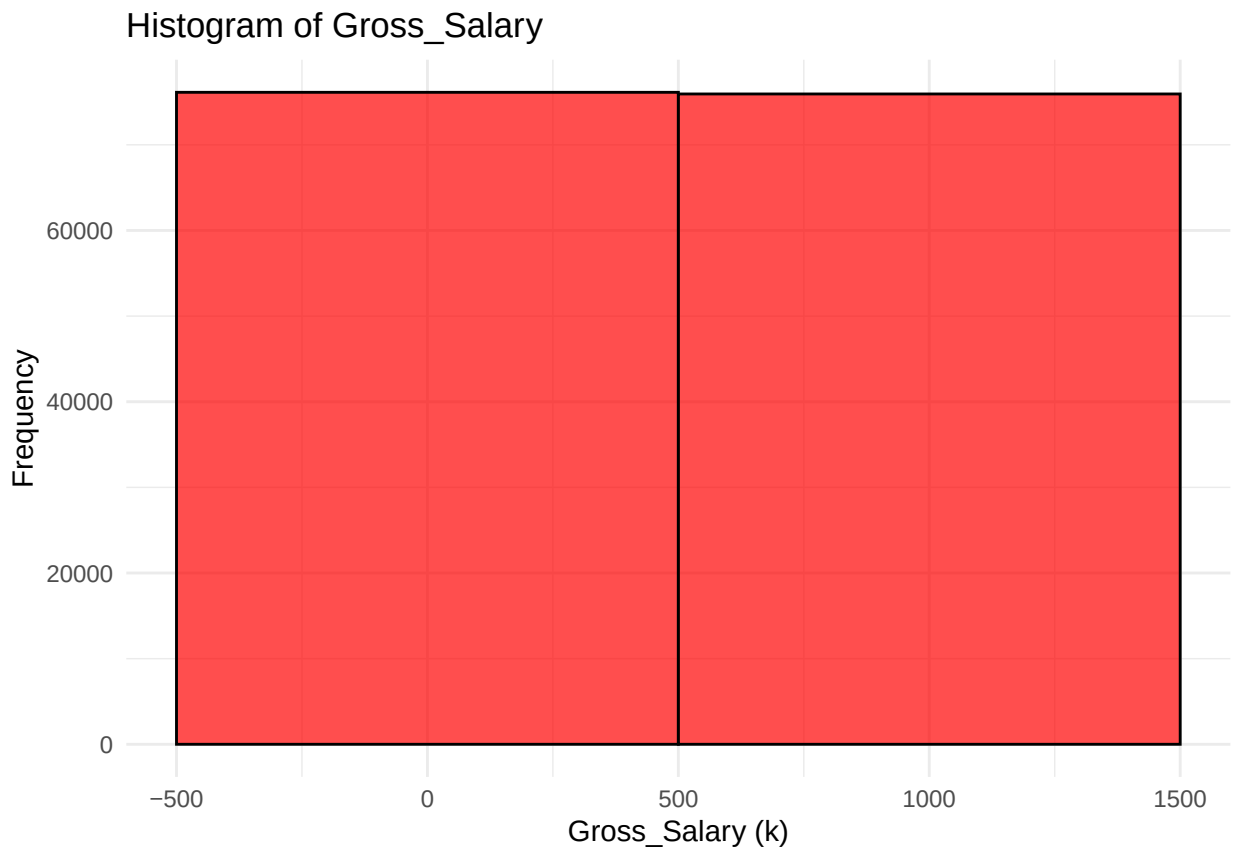
```
salary_columns <- c("Net_Salary", "Annual_Salary", "Gross_Salary")

colors <- c("blue", "green", "red")
```

```
for (i in seq_along(salary_columns)) {
  p <- ggplot(data_df, aes_string(x = salary_columns[i])) +
    geom_histogram(binwidth = 1000, fill = colors[i], color = "black", alpha = 0.7) +
    labs(title = paste("Histogram of", salary_columns[i]),
         x =  paste(salary_columns[i], "(k)"),
         y = "Frequency") +
    theme_minimal()

  print(p)
}
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```


Histogram of Net_Salary
80000
60000
40000
20000
0
Frequency
−500
0
500
1000
1500
Net_Salary (k)


Histogram of Annual_Salary
80000
60000
40000
20000
0
Frequency
−500
0
500
1000
1500
Annual_Salary (k)



The annual and gross salary seem almost identical, this is to be expected as they are very similar in value. hence the net salary will be used to calculate the target.

```
names(data_df)
```

TTarget variable:

```
##  [1] "marital_status"   "street_address"   "postal_code"      "city"
##  [5] "state_province"   "Country_id"       "phone_number"     "email"
##  [9] "Education"        "Occupation"       "household_size"   "yrs_of_residence"
## [13] "Annual_Salary"    "Months_Annual"    "FRS.Contribution" "Year_of_Birth"
## [17] "Net_Salary"       "Net_months"       "Gross_Salary"     "Gross_Months"
```

```
# Calculate the Qualify variable
data_df$Qualify <- ifelse((data_df$Net_Salary * 1000 / 12) >= 50000, 1, 0)
```

```
head(data_df[, c("Net_Salary", "Qualify")], n = 10)
```

```
##    Net_Salary Qualify
## 1      501.62       0
## 2      467.63       0
## 3      513.71       0
## 4      561.67       0
## 5      665.66       1
## 6      802.71       1
## 7      725.53       1
## 8      360.64       0
```

```
## 9       92.69       0
## 12     251.40       0
```

Bar plot of the final count

```
qualify_counts <- table(data_df$Qualify)
qualify_percentages <- qualify_counts / sum(qualify_counts) * 100
qualify_percentages_df <- as.data.frame(qualify_percentages)

colnames(qualify_percentages_df) <- c("Qualify", "Percentage")
```

```
ggplot(qualify_percentages_df, aes(x = factor(Qualify), y = Percentage)) +
  geom_bar(stat = "identity", fill = "lightblue", color = "black") +
  labs(title = "Percentage of Qualify Values",
       x = "Qualify",
       y = "Percentage (%)") +
  theme_minimal() +
  scale_x_discrete(labels = c("0" = "Not Qualified", "1" = "Qualified")) +
  geom_text(aes(label = round(Percentage, 1)), vjust = -0.5)
```

A 40% to almost 60% in class percentages, suggests that class balancing techniques will not be required; as the two classes are somewhat almost balanced.

Final structure of the dataset:

```
column_types <- sapply(data_df, class)

column_info <- data.frame(Column_Name = names(column_types),
                          Data_Type = column_types,
                          stringsAsFactors = FALSE)
print(column_info)
```

```
##                       Column_Name Data_Type
## marital_status     marital_status   numeric
## street_address     street_address character
## postal_code           postal_code   numeric
## city                         city character
## state_province     state_province character
## Country_id             Country_id character
## phone_number         phone_number character
## email                       email character
## Education               Education character
## Occupation              Occupation character
## household_size     household_size   numeric
## yrs_of_residence yrs_of_residence   numeric
## Annual_Salary       Annual_Salary   numeric
## Months_Annual       Months_Annual   numeric
## FRS.Contribution FRS.Contribution   numeric
## Year_of_Birth       Year_of_Birth   numeric
## Net_Salary             Net_Salary   numeric
## Net_months             Net_months   numeric
## Gross_Salary         Gross_Salary   numeric
## Gross_Months         Gross_Months   numeric
## Qualify                   Qualify   numeric
```

the data appears to be ready for transformation. The data will be saved in a csv file.

```
write.csv(data_df, file = "cleaned_cust.csv", row.names = FALSE)
```

Final notes on what was done so far

- Adding data type constraints on the column data
  - The columns that are expected to be numeric are now numeric; some columns such as "marital_status" are represented numerically as they are categorical.
- missing values have been imputed using mean for continuous variables and mode for categorical variables.
- Inconsistent and noisy data has been removed.
- Some categorical variables that will be used to train the model have been discretized.

```
missing_values_summary <- list()

for (col_name in names(data_df)) {
  total_missing <- sum(is.na(data_df[[col_name]]))

  missing_values_summary[[col_name]] <- total_missing
}
```

```r
missing_values_df <- data.frame(
  Column = names(missing_values_summary),
  Total_Missing_Values = unlist(missing_values_summary)
)

missing_values_df
```

Missing values have been delt with

```
##                           Column Total_Missing_Values
## marital_status     marital_status                    0
## street_address     street_address                    0
## postal_code           postal_code                    0
## city                         city                    0
## state_province     state_province                    0
## Country_id            Country_id                     0
## phone_number         phone_number                    0
## email                       email                    0
## Education              Education                     0
## Occupation             Occupation                    0
## household_size     household_size                    0
## yrs_of_residence yrs_of_residence                    0
## Annual_Salary       Annual_Salary                    0
## Months_Annual       Months_Annual                    0
## FRS.Contribution FRS.Contribution                    0
## Year_of_Birth       Year_of_Birth                    0
## Net_Salary             Net_Salary                    0
## Net_months             Net_months                    0
## Gross_Salary         Gross_Salary                    0
## Gross_Months         Gross_Months                    0
## Qualify                   Qualify                    0
```