HomeCredit EDA

Michael Tom

October 08, 2023

Contents

Introduction

Business Problem/Project Goal

HomeCredit is an international credit lender that supports customers with little to no credit history. With that in mind they are in need of a way to predict which of their applicants should be approved for loans based around alternative factors. The goal of this project is to create a classification model, using variables form the provided data, that will outperform a majority class classifier on prediction of our target variable (default).

This EDA Notebook will explore the structure of the provided data, prepare the data for modeling, address missing data and explore different attributes for their potential use in future modeling.

Questions to be answered

- What is our target variable?

- What is the current % of defaulting loans?
- What will we do with variables with missing data?
- What will we do with variables with very low variability?
- Which variables could be viable for a classification model (which show a difference between default and non default)?
- How can the additional 6 tables of data be used in our modeling?
- Will we need to do any transformation on the data to make them work for our models?
- Are there any indications of errors in the data? If so how will these be addressed?
- Are there any variables that maybe helpful, but might be discriminatory if used?

Load Packages, Import/Prep Data

In looking at the summary of the data we answer a few of our questions as well as find a few new ones. We found that we have a total sample size in our training data of 307,511. Of these our target variable is divided 282,686 (92%) for non default vs 24,825 (08%) Default. This shows us that our target for our model is to be able to predict better than a majority classifier of 92%. There are still many items that need addressing before modeling. There are many variables with a large amount of N/A's as well as a low amount of variability. There are also a few data points that need to be check/addressed for accuracy. With the additional 6 data sets we will need to address how they could possibly be brought into our data set to help us in our prediction model. Once these are addressed we should be able to start selecting variables with 2 different types of comparisons. Target vs Categorical check to see if there is a difference in distribution between te different categories. Target vs continuous with these we can compare the means between the target groups to determine if there is a difference which would denote a possible predictor.

```
# Load packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.3     v purrr   1.0.2
## v tibble  3.2.1     v dplyr   1.1.3
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(tidyr)
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
```

```
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
#import data
cloud_wd <- getwd()
setwd(cloud_wd)
application_test <- read.csv(file = "application_test.csv", stringsAsFactors = TRUE)
application_train <- read.csv(file = "application_train.csv", stringsAsFactors = TRUE)
bereau_balance <- read.csv(file = "bureau_balance.csv", stringsAsFactors = TRUE)
bureau <- read.csv(file = "bureau.csv", stringsAsFactors = TRUE)
credit_card_balance <- read.csv(file = "credit_card_balance.csv", stringsAsFactors = TRUE)
installments_payments <- read.csv(file = "installments_payments.csv", stringsAsFactors = TRUE)
POS_CASH_balance <- read.csv(file = "POS_CASH_balance.csv", stringsAsFactors = TRUE)
previous_application <- read.csv(file = "previous_application.csv", stringsAsFactors = TRUE)

#Check structure of target data
str(application_train, list.len = ncol(application_train))
```

```
## 'data.frame':    307511 obs. of  122 variables:
##  $ SK_ID_CURR                  : int  100002 100003 100004 100006 100007 100008 100009 100010 100011
##  $ TARGET                      : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ NAME_CONTRACT_TYPE          : Factor w/ 2 levels "Cash loans","Revolving loans": 1 1 2 1 1 1 1 1
##  $ CODE_GENDER                 : Factor w/ 3 levels "F","M","XNA": 2 1 2 1 2 2 1 2 1 2 ...
##  $ FLAG_OWN_CAR                : Factor w/ 2 levels "N","Y": 1 1 2 1 1 1 2 2 1 1 ...
##  $ FLAG_OWN_REALTY             : Factor w/ 2 levels "N","Y": 2 1 2 2 2 2 2 2 2 2 ...
##  $ CNT_CHILDREN                : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ AMT_INCOME_TOTAL            : num  202500 270000 67500 135000 121500 ...
##  $ AMT_CREDIT                  : num  406598 1293502 135000 312682 513000 ...
##  $ AMT_ANNUITY                 : num  24700 35698 6750 29686 21866 ...
##  $ AMT_GOODS_PRICE             : num  351000 1129500 135000 297000 513000 ...
##  $ NAME_TYPE_SUITE             : Factor w/ 8 levels "","Children",..: 8 3 8 8 8 7 8 8 2 8 ...
##  $ NAME_INCOME_TYPE            : Factor w/ 8 levels "Businessman",..: 8 5 8 8 8 5 2 5 4 8 ...
##  $ NAME_EDUCATION_TYPE         : Factor w/ 5 levels "Academic degree",..: 5 2 5 5 5 5 2 2 5 5 ...
##  $ NAME_FAMILY_STATUS          : Factor w/ 6 levels "Civil marriage",..: 4 2 4 1 4 2 2 2 2 4 ...
##  $ NAME_HOUSING_TYPE           : Factor w/ 6 levels "Co-op apartment",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ REGION_POPULATION_RELATIVE  : num  0.0188 0.00354 0.01003 0.00802 0.02866 ...
##  $ DAYS_BIRTH                  : int  -9461 -16765 -19046 -19005 -19932 -16941 -13778 -18850 -20099
##  $ DAYS_EMPLOYED               : int  -637 -1188 -225 -3039 -3038 -1588 -3130 -449 365243 -2019 ...
##  $ DAYS_REGISTRATION           : num  -3648 -1186 -4260 -9833 -4311 ...
##  $ DAYS_ID_PUBLISH             : int  -2120 -291 -2531 -2437 -3458 -477 -619 -2379 -3514 -3992 ...
##  $ OWN_CAR_AGE                 : int  NA NA 26 NA NA NA 17 8 NA NA ...
##  $ FLAG_MOBIL                  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_EMP_PHONE              : int  1 1 1 1 1 1 1 1 0 1 ...
##  $ FLAG_WORK_PHONE             : int  0 0 1 0 0 1 0 1 0 0 ...
##  $ FLAG_CONT_MOBILE            : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_PHONE                  : int  1 1 1 0 0 1 1 0 0 0 ...
##  $ FLAG_EMAIL                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ OCCUPATION_TYPE             : Factor w/ 19 levels "","Accountants",..: 10 5 10 10 5 10 2 12 1 10
##  $ CNT_FAM_MEMBERS             : int  1 2 1 2 1 2 3 2 2 1 ...
##  $ REGION_RATING_CLIENT        : int  2 1 2 2 2 2 2 3 2 2 ...
##  $ REGION_RATING_CLIENT_W_CITY : int  2 1 2 2 2 2 2 3 2 2 ...
##  $ WEEKDAY_APPR_PROCESS_START  : Factor w/ 7 levels "FRIDAY","MONDAY",..: 7 2 2 7 5 7 4 2 7 5 ...
```

```
##  $ HOUR_APPR_PROCESS_START     : int  10 11 9 17 11 16 16 16 14 8 ...
##  $ REG_REGION_NOT_LIVE_REGION  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ REG_REGION_NOT_WORK_REGION  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ LIVE_REGION_NOT_WORK_REGION : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ REG_CITY_NOT_LIVE_CITY      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ REG_CITY_NOT_WORK_CITY      : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ LIVE_CITY_NOT_WORK_CITY     : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ ORGANIZATION_TYPE           : Factor w/ 58 levels "Advertising",..: 6 40 12 6 38 34 6 34 58 10 ..
##  $ EXT_SOURCE_1                : num  0.083 0.311 NA NA NA ...
##  $ EXT_SOURCE_2                : num  0.263 0.622 0.556 0.65 0.323 ...
##  $ EXT_SOURCE_3                : num  0.139 NA 0.73 NA NA ...
##  $ APARTMENTS_AVG              : num  0.0247 0.0959 NA NA NA NA NA NA NA NA ...
##  $ BASEMENTAREA_AVG            : num  0.0369 0.0529 NA NA NA NA NA NA NA NA ...
##  $ YEARS_BEGINEXPLUATATION_AVG : num  0.972 0.985 NA NA NA ...
##  $ YEARS_BUILD_AVG             : num  0.619 0.796 NA NA NA ...
##  $ COMMONAREA_AVG              : num  0.0143 0.0605 NA NA NA NA NA NA NA NA ...
##  $ ELEVATORS_AVG               : num  0 0.08 NA NA NA NA NA NA NA NA ...
##  $ ENTRANCES_AVG               : num  0.069 0.0345 NA NA NA NA NA NA NA NA ...
##  $ FLOORSMAX_AVG               : num  0.0833 0.2917 NA NA NA ...
##  $ FLOORSMIN_AVG               : num  0.125 0.333 NA NA NA ...
##  $ LANDAREA_AVG                : num  0.0369 0.013 NA NA NA NA NA NA NA NA ...
##  $ LIVINGAPARTMENTS_AVG        : num  0.0202 0.0773 NA NA NA NA NA NA NA NA ...
##  $ LIVINGAREA_AVG              : num  0.019 0.0549 NA NA NA NA NA NA NA NA ...
##  $ NONLIVINGAPARTMENTS_AVG     : num  0 0.0039 NA NA NA NA NA NA NA NA ...
##  $ NONLIVINGAREA_AVG           : num  0 0.0098 NA NA NA NA NA NA NA NA ...
##  $ APARTMENTS_MODE             : num  0.0252 0.0924 NA NA NA NA NA NA NA NA ...
##  $ BASEMENTAREA_MODE           : num  0.0383 0.0538 NA NA NA NA NA NA NA NA ...
##  $ YEARS_BEGINEXPLUATATION_MODE: num  0.972 0.985 NA NA NA ...
##  $ YEARS_BUILD_MODE            : num  0.634 0.804 NA NA NA ...
##  $ COMMONAREA_MODE             : num  0.0144 0.0497 NA NA NA NA NA NA NA NA ...
##  $ ELEVATORS_MODE              : num  0 0.0806 NA NA NA NA NA NA NA NA ...
##  $ ENTRANCES_MODE              : num  0.069 0.0345 NA NA NA NA NA NA NA NA ...
##  $ FLOORSMAX_MODE              : num  0.0833 0.2917 NA NA NA ...
##  $ FLOORSMIN_MODE              : num  0.125 0.333 NA NA NA ...
##  $ LANDAREA_MODE               : num  0.0377 0.0128 NA NA NA NA NA NA NA NA ...
##  $ LIVINGAPARTMENTS_MODE       : num  0.022 0.079 NA NA NA NA NA NA NA NA ...
##  $ LIVINGAREA_MODE             : num  0.0198 0.0554 NA NA NA NA NA NA NA NA ...
##  $ NONLIVINGAPARTMENTS_MODE    : num  0 0 NA NA NA NA NA NA NA NA ...
##  $ NONLIVINGAREA_MODE          : num  0 0 NA NA NA NA NA NA NA NA ...
##  $ APARTMENTS_MEDI             : num  0.025 0.0968 NA NA NA NA NA NA NA NA ...
##  $ BASEMENTAREA_MEDI           : num  0.0369 0.0529 NA NA NA NA NA NA NA NA ...
##  $ YEARS_BEGINEXPLUATATION_MEDI: num  0.972 0.985 NA NA NA ...
##  $ YEARS_BUILD_MEDI            : num  0.624 0.799 NA NA NA ...
##  $ COMMONAREA_MEDI             : num  0.0144 0.0608 NA NA NA NA NA NA NA NA ...
##  $ ELEVATORS_MEDI              : num  0 0.08 NA NA NA NA NA NA NA NA ...
##  $ ENTRANCES_MEDI              : num  0.069 0.0345 NA NA NA NA NA NA NA NA ...
##  $ FLOORSMAX_MEDI              : num  0.0833 0.2917 NA NA NA ...
##  $ FLOORSMIN_MEDI              : num  0.125 0.333 NA NA NA ...
##  $ LANDAREA_MEDI               : num  0.0375 0.0132 NA NA NA NA NA NA NA NA ...
##  $ LIVINGAPARTMENTS_MEDI       : num  0.0205 0.0787 NA NA NA NA NA NA NA NA ...
##  $ LIVINGAREA_MEDI             : num  0.0193 0.0558 NA NA NA NA NA NA NA NA ...
##  $ NONLIVINGAPARTMENTS_MEDI    : num  0 0.0039 NA NA NA NA NA NA NA NA ...
##  $ NONLIVINGAREA_MEDI          : num  0 0.01 NA NA NA NA NA NA NA NA ...
##  $ FONDKAPREMONT_MODE          : Factor w/ 5 levels "","not specified",..: 4 4 1 1 1 1 1 1 1 1 ...
```

```
##  $ HOUSETYPE_MODE              : Factor w/ 4 levels "","block of flats",..: 2 2 1 1 1 1 1 1 1 1 ...
##  $ TOTALAREA_MODE              : num  0.0149 0.0714 NA NA NA NA NA NA NA NA ...
##  $ WALLSMATERIAL_MODE          : Factor w/ 8 levels "","Block","Mixed",..: 7 2 1 1 1 1 1 1 1 1 ...
##  $ EMERGENCYSTATE_MODE         : Factor w/ 3 levels "","No","Yes": 2 2 1 1 1 1 1 1 1 1 ...
##  $ OBS_30_CNT_SOCIAL_CIRCLE    : int  2 1 0 2 0 0 1 2 1 2 ...
##  $ DEF_30_CNT_SOCIAL_CIRCLE    : int  2 0 0 0 0 0 0 0 0 0 ...
##  $ OBS_60_CNT_SOCIAL_CIRCLE    : int  2 1 0 2 0 0 1 2 1 2 ...
##  $ DEF_60_CNT_SOCIAL_CIRCLE    : int  2 0 0 0 0 0 0 0 0 0 ...
##  $ DAYS_LAST_PHONE_CHANGE      : int  -1134 -828 -815 -617 -1106 -2536 -1562 -1070 0 -1673 ...
##  $ FLAG_DOCUMENT_2             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG_DOCUMENT_3             : int  1 1 0 1 0 1 0 1 1 0 ...
##  $ FLAG_DOCUMENT_4             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG_DOCUMENT_5             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG_DOCUMENT_6             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG_DOCUMENT_7             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG_DOCUMENT_8             : int  0 0 0 0 1 0 1 0 0 0 ...
##  $ FLAG_DOCUMENT_9             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG_DOCUMENT_10            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG_DOCUMENT_11            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG_DOCUMENT_12            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG_DOCUMENT_13            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG_DOCUMENT_14            : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ FLAG_DOCUMENT_15            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG_DOCUMENT_16            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG_DOCUMENT_17            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG_DOCUMENT_18            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG_DOCUMENT_19            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG_DOCUMENT_20            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FLAG_DOCUMENT_21            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AMT_REQ_CREDIT_BUREAU_HOUR  : int  0 0 0 NA 0 0 0 0 0 NA ...
##  $ AMT_REQ_CREDIT_BUREAU_DAY   : int  0 0 0 NA 0 0 0 0 0 NA ...
##  $ AMT_REQ_CREDIT_BUREAU_WEEK  : int  0 0 0 NA 0 0 0 0 0 NA ...
##  $ AMT_REQ_CREDIT_BUREAU_MON   : int  0 0 0 NA 0 0 1 0 0 NA ...
##  $ AMT_REQ_CREDIT_BUREAU_QRT   : int  0 0 0 NA 0 1 1 0 0 NA ...
##  $ AMT_REQ_CREDIT_BUREAU_YEAR  : int  1 0 0 NA 0 1 2 0 1 NA ...
```

```
###create working data set
clean_train <- application_train

###Add Factors to variables
factors <- c('TARGET','FLAG_MOBIL','FLAG_EMP_PHONE','FLAG_WORK_PHONE','FLAG_CONT_MOBILE','FLAG_PHONE','
factors
```

```
##  [1] "TARGET"                     "FLAG_MOBIL"
##  [3] "FLAG_EMP_PHONE"             "FLAG_WORK_PHONE"
##  [5] "FLAG_CONT_MOBILE"           "FLAG_PHONE"
##  [7] "FLAG_EMAIL"                 "REGION_RATING_CLIENT"
##  [9] "REGION_RATING_CLIENT_W_CITY" "REG_REGION_NOT_LIVE_REGION"
## [11] "REG_REGION_NOT_WORK_REGION"  "LIVE_REGION_NOT_WORK_REGION"
## [13] "REG_CITY_NOT_LIVE_CITY"      "REG_CITY_NOT_WORK_CITY"
## [15] "LIVE_CITY_NOT_WORK_CITY"     "FLAG_DOCUMENT_2"
## [17] "FLAG_DOCUMENT_3"             "FLAG_DOCUMENT_4"
## [19] "FLAG_DOCUMENT_5"             "FLAG_DOCUMENT_6"
## [21] "FLAG_DOCUMENT_7"             "FLAG_DOCUMENT_8"
```

```
## [23] "FLAG_DOCUMENT_9"             "FLAG_DOCUMENT_10"
## [25] "FLAG_DOCUMENT_11"            "FLAG_DOCUMENT_12"
## [27] "FLAG_DOCUMENT_13"            "FLAG_DOCUMENT_14"
## [29] "FLAG_DOCUMENT_15"            "FLAG_DOCUMENT_16"
## [31] "FLAG_DOCUMENT_17"            "FLAG_DOCUMENT_18"
## [33] "FLAG_DOCUMENT_19"            "FLAG_DOCUMENT_20"
## [35] "FLAG_DOCUMENT_21"
```

```
clean_train[factors] <- lapply(application_train[factors], factor)
###Check that factors applied
str(clean_train[factors], list.len =ncol(clean_train))
```

```
## 'data.frame':    307511 obs. of  35 variables:
##  $ TARGET                     : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_MOBIL                 : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ FLAG_EMP_PHONE             : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 1 2 ...
##  $ FLAG_WORK_PHONE            : Factor w/ 2 levels "0","1": 1 1 2 1 1 2 1 2 1 1 ...
##  $ FLAG_CONT_MOBILE           : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ FLAG_PHONE                 : Factor w/ 2 levels "0","1": 2 2 2 1 1 2 2 1 1 1 ...
##  $ FLAG_EMAIL                 : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ REGION_RATING_CLIENT       : Factor w/ 3 levels "1","2","3": 2 1 2 2 2 2 2 3 2 2 ...
##  $ REGION_RATING_CLIENT_W_CITY: Factor w/ 3 levels "1","2","3": 2 1 2 2 2 2 2 3 2 2 ...
##  $ REG_REGION_NOT_LIVE_REGION : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ REG_REGION_NOT_WORK_REGION : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ LIVE_REGION_NOT_WORK_REGION: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ REG_CITY_NOT_LIVE_CITY     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ REG_CITY_NOT_WORK_CITY     : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 2 1 1 ...
##  $ LIVE_CITY_NOT_WORK_CITY    : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 2 1 1 ...
##  $ FLAG_DOCUMENT_2            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_DOCUMENT_3            : Factor w/ 2 levels "0","1": 2 2 1 2 1 2 1 2 2 1 ...
##  $ FLAG_DOCUMENT_4            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_DOCUMENT_5            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_DOCUMENT_6            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_DOCUMENT_7            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_DOCUMENT_8            : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 2 1 1 1 ...
##  $ FLAG_DOCUMENT_9            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_DOCUMENT_10           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_DOCUMENT_11           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_DOCUMENT_12           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_DOCUMENT_13           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_DOCUMENT_14           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 1 ...
##  $ FLAG_DOCUMENT_15           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_DOCUMENT_16           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_DOCUMENT_17           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_DOCUMENT_18           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_DOCUMENT_19           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_DOCUMENT_20           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FLAG_DOCUMENT_21           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

```
#Check Summary of data
## Suppressed due to Length summary(clean_train)

#Check proportion of default
prop.table(table(clean_train$TARGET))
```

```
##
##          0          1
## 0.91927118 0.08072882
```

```r
#Check structure and summary of the bereau_blance data
str(bereau_balance)
```

```
## 'data.frame':    27299925 obs. of  3 variables:
##  $ SK_ID_BUREAU  : int  5715448 5715448 5715448 5715448 5715448 5715448 5715448 5715448 5715448 5715
##  $ MONTHS_BALANCE: int  0 -1 -2 -3 -4 -5 -6 -7 -8 -9 ...
##  $ STATUS        : Factor w/ 8 levels "0","1","2","3",..: 7 7 7 7 7 7 7 7 7 1 ...
```

```r
## Suppressed due to Length summary(bereau_balance)

#Check structure and summary of the bureau data
str(bureau)
```

```
## 'data.frame':    1716428 obs. of  17 variables:
##  $ SK_ID_CURR            : int  215354 215354 215354 215354 215354 215354 215354 162297 162297 16229
##  $ SK_ID_BUREAU          : int  5714462 5714463 5714464 5714465 5714466 5714467 5714468 5714469 5714
##  $ CREDIT_ACTIVE         : Factor w/ 4 levels "Active","Bad debt",..: 3 1 1 1 1 1 1 3 3 1 ...
##  $ CREDIT_CURRENCY       : Factor w/ 4 levels "currency 1","currency 2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ DAYS_CREDIT           : int  -497 -208 -203 -203 -629 -273 -43 -1896 -1146 -1146 ...
##  $ CREDIT_DAY_OVERDUE    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ DAYS_CREDIT_ENDDATE   : num  -153 1075 528 NA 1197 ...
##  $ DAYS_ENDDATE_FACT     : num  -153 NA NA NA NA NA NA -1710 -840 NA ...
##  $ AMT_CREDIT_MAX_OVERDUE: num  NA NA NA NA 77674 ...
##  $ CNT_CREDIT_PROLONG    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AMT_CREDIT_SUM        : num  91323 225000 464324 90000 2700000 ...
##  $ AMT_CREDIT_SUM_DEBT   : num  0 171342 NA NA NA ...
##  $ AMT_CREDIT_SUM_LIMIT  : num  NA NA NA NA NA ...
##  $ AMT_CREDIT_SUM_OVERDUE: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CREDIT_TYPE           : Factor w/ 15 levels "Another type of loan",..: 4 5 4 5 4 5 4 4 4 5 ...
##  $ DAYS_CREDIT_UPDATE    : int  -131 -20 -16 -16 -21 -31 -22 -1710 -840 -690 ...
##  $ AMT_ANNUITY           : num  NA NA NA NA NA NA NA NA NA NA ...
```

```r
## Suppressed due to Length summary(bureau)

#Check structure and summary of credit_card_balance
str(credit_card_balance)
```

```
## 'data.frame':    3840312 obs. of  23 variables:
##  $ SK_ID_PREV                : int  2562384 2582071 1740877 1389973 1891521 2646502 1079071 2095912 1
##  $ SK_ID_CURR                : int  378907 363914 371185 337855 126868 380010 171320 118650 367360 20
##  $ MONTHS_BALANCE            : int  -6 -1 -7 -4 -1 -7 -6 -7 -4 -5 ...
##  $ AMT_BALANCE               : num  57 63976 31815 236572 453919 ...
##  $ AMT_CREDIT_LIMIT_ACTUAL   : int  135000 45000 450000 225000 450000 270000 585000 45000 292500 225
##  $ AMT_DRAWINGS_ATM_CURRENT  : num  0 2250 0 2250 0 0 67500 45000 90000 76500 ...
##  $ AMT_DRAWINGS_CURRENT      : num  878 2250 0 2250 11547 ...
##  $ AMT_DRAWINGS_OTHER_CURRENT: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ AMT_DRAWINGS_POS_CURRENT  : num  878 0 0 0 11547 ...
##  $ AMT_INST_MIN_REGULARITY   : num  1700 2250 2250 11796 22925 ...
```

```
##  $ AMT_PAYMENT_CURRENT       : num  1800 2250 2250 11925 27000 ...
##  $ AMT_PAYMENT_TOTAL_CURRENT : num  1800 2250 2250 11925 27000 ...
##  $ AMT_RECEIVABLE_PRINCIPAL  : num  0 60175 26926 224949 443044 ...
##  $ AMT_RECIVABLE             : num  0 64876 31460 233049 453919 ...
##  $ AMT_TOTAL_RECEIVABLE      : num  0 64876 31460 233049 453919 ...
##  $ CNT_DRAWINGS_ATM_CURRENT  : num  0 1 0 1 0 0 1 1 3 3 ...
##  $ CNT_DRAWINGS_CURRENT      : int  1 1 0 1 1 0 1 1 8 9 ...
##  $ CNT_DRAWINGS_OTHER_CURRENT: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CNT_DRAWINGS_POS_CURRENT  : num  1 0 0 0 1 0 0 0 5 6 ...
##  $ CNT_INSTALMENT_MATURE_CUM : num  35 69 30 10 101 2 6 51 3 38 ...
##  $ NAME_CONTRACT_STATUS      : Factor w/ 7 levels "Active","Approved",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ SK_DPD                    : int  0 0 0 0 0 7 0 0 0 0 ...
##  $ SK_DPD_DEF                : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
## Suppressed due to Length summary(credit_card_balance)

#Check structure and summary of installments_payments
str(installments_payments)
```

```
## 'data.frame':    13605401 obs. of  8 variables:
##  $ SK_ID_PREV            : int  1054186 1330831 2085231 2452527 2714724 1137312 2234264 1818599 2723
##  $ SK_ID_CURR            : int  161674 151639 193053 199697 167756 164489 184693 111420 112102 10974
##  $ NUM_INSTALMENT_VERSION: num  1 0 2 1 1 1 4 2 0 1 ...
##  $ NUM_INSTALMENT_NUMBER : int  6 34 1 3 2 12 11 4 14 4 ...
##  $ DAYS_INSTALMENT       : num  -1180 -2156 -63 -2418 -1383 ...
##  $ DAYS_ENTRY_PAYMENT    : num  -1187 -2156 -63 -2426 -1366 ...
##  $ AMT_INSTALMENT        : num  6948 1717 25425 24350 2165 ...
##  $ AMT_PAYMENT           : num  6948 1717 25425 24350 2161 ...
```

```
## Suppressed due to Length summary(installments_payments)

#Check structure and summary of POS_CASH_balance
str(POS_CASH_balance)
```

```
## 'data.frame':    10001358 obs. of  8 variables:
##  $ SK_ID_PREV          : int  1803195 1715348 1784872 1903291 2341044 2207092 1110516 1387235 12205
##  $ SK_ID_CURR          : int  182943 367990 397406 269225 334279 342166 204376 153211 112740 274851
##  $ MONTHS_BALANCE      : int  -31 -33 -32 -35 -35 -32 -38 -35 -31 -32 ...
##  $ CNT_INSTALMENT      : num  48 36 12 48 36 12 48 36 12 24 ...
##  $ CNT_INSTALMENT_FUTURE: num  45 35 9 42 35 12 43 36 12 16 ...
##  $ NAME_CONTRACT_STATUS : Factor w/ 9 levels "Active","Amortized debt",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ SK_DPD              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SK_DPD_DEF          : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
## Suppressed due to Length summary(POS_CASH_balance)

#Check structure and summary of previous_application
str(previous_application)
```

```
## 'data.frame':    1670214 obs. of  37 variables:
##  $ SK_ID_PREV                : int  2030495 2802425 2523466 2819243 1784265 1383531 2315218 1656711
##  $ SK_ID_CURR                : int  271877 108129 122040 176158 202054 199383 175704 296299 342292 3
```

```
##  $ NAME_CONTRACT_TYPE         : Factor w/ 4 levels "Cash loans","Consumer loans",..: 2 1 1 1 1 1 1 1
##  $ AMT_ANNUITY                : num  1730 25189 15061 47041 31924 ...
##  $ AMT_APPLICATION            : num  17145 607500 112500 450000 337500 ...
##  $ AMT_CREDIT                 : num  17145 679671 136444 470790 404055 ...
##  $ AMT_DOWN_PAYMENT           : num  0 NA NA NA NA NA NA NA NA NA ...
##  $ AMT_GOODS_PRICE            : num  17145 607500 112500 450000 337500 ...
##  $ WEEKDAY_APPR_PROCESS_START : Factor w/ 7 levels "FRIDAY","MONDAY",..: 3 5 6 2 5 3 6 2 2 3 ...
##  $ HOUR_APPR_PROCESS_START    : int  15 11 11 7 9 8 11 7 15 15 ...
##  $ FLAG_LAST_APPL_PER_CONTRACT: Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ NFLAG_LAST_APPL_IN_DAY     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ RATE_DOWN_PAYMENT          : num  0 NA NA NA NA NA NA NA NA NA ...
##  $ RATE_INTEREST_PRIMARY      : num  0.183 NA NA NA NA ...
##  $ RATE_INTEREST_PRIVILEGED   : num  0.867 NA NA NA NA ...
##  $ NAME_CASH_LOAN_PURPOSE     : Factor w/ 25 levels "Building a house or an annex",..: 24 25 25 25 2
##  $ NAME_CONTRACT_STATUS       : Factor w/ 4 levels "Approved","Canceled",..: 1 1 1 1 3 1 2 2 2 2 ...
##  $ DAYS_DECISION              : int  -73 -164 -301 -512 -781 -684 -14 -21 -386 -57 ...
##  $ NAME_PAYMENT_TYPE          : Factor w/ 4 levels "Cash through the bank",..: 1 4 1 1 1 1 4 4 4 4 .
##  $ CODE_REJECT_REASON         : Factor w/ 9 levels "CLIENT","HC",..: 8 8 8 8 2 8 8 8 8 8 ...
##  $ NAME_TYPE_SUITE            : Factor w/ 8 levels "","Children",..: 1 8 7 1 1 3 1 1 1 1 ...
##  $ NAME_CLIENT_TYPE           : Factor w/ 4 levels "New","Refreshed",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ NAME_GOODS_CATEGORY        : Factor w/ 28 levels "Additional Service",..: 20 28 28 28 28 28 28 28
##  $ NAME_PORTFOLIO             : Factor w/ 5 levels "Cards","Cars",..: 4 3 3 3 3 3 5 5 5 5 ...
##  $ NAME_PRODUCT_TYPE          : Factor w/ 3 levels "walk-in","x-sell",..: 3 2 2 2 1 2 3 3 3 3 ...
##  $ CHANNEL_TYPE               : Factor w/ 8 levels "AP+ (Cash loan)",..: 5 4 6 6 6 6 6 6 6 6 ...
##  $ SELLERPLACE_AREA           : int  35 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ NAME_SELLER_INDUSTRY       : Factor w/ 11 levels "Auto technology",..: 3 11 11 11 11 11 11 11 11 1
##  $ CNT_PAYMENT                : num  12 36 12 12 24 18 NA NA NA NA ...
##  $ NAME_YIELD_GROUP           : Factor w/ 5 levels "high","low_action",..: 4 2 1 4 1 3 5 5 5 5 ...
##  $ PRODUCT_COMBINATION        : Factor w/ 18 levels "","Card Street",..: 15 9 8 10 5 9 4 4 4 4 ...
##  $ DAYS_FIRST_DRAWING         : num  365243 365243 365243 365243 NA ...
##  $ DAYS_FIRST_DUE             : num  -42 -134 -271 -482 NA -654 NA NA NA NA ...
##  $ DAYS_LAST_DUE_1ST_VERSION  : num  300 916 59 -152 NA -144 NA NA NA NA ...
##  $ DAYS_LAST_DUE              : num  -42 365243 365243 -182 NA ...
##  $ DAYS_TERMINATION           : num  -37 365243 365243 -177 NA ...
##  $ NFLAG_INSURED_ON_APPROVAL  : num  0 1 1 1 NA 1 NA NA NA NA ...
```

```
## Suppressed due to Length summary(previous_application)
```

N/A and Missing Data

In examining the missing data we find there are 45 attributes with 35% or more of their data listed as N/A. With such a large amount of the data missing we suggest these variables to be removed from model consideration. There are also 4 attributes with 35% or more data with blanks as with the N/A we would suggest removing these from consideration. After this we have 13 attributes which still contain N/A and 2 which contain blanks that will need addressing. For the remaining we would suggest:

- Anything less than 1% NA or blanks impute the mean vales for the missing. This will remove 6 of the 13 N/A and 1 of the Blanks
- For Ext_Source_3 though there is 19.83% missing data there does seem to be a difference of means between that could indicate as predictor for default. For this we would suggest testing models with this variable imputed.

- For the AMT_REQ group ()
- For HOUR, DAY, and WEEK 99% of values are either N/A or 0 for these we would suggest exuding these attributes from our models
- FOR MON, QTY, YEAR there are possible indicators of difference in the groups. For these we would suggest imputing the missing values.
- For OCCUPATION_TYPE/NAME_TYPE_SUITE This does seem to have potential as a predictor in our model. For these we would suggest keeping the blanks as their own category of "undisclosed" and using them in the model.

```
#Find % of NA's by attribute
sort(sapply(clean_train, function(x)
       round(100*sum(is.na(x))/length(x),2)),decreasing =TRUE)
```

```
##               COMMONAREA_AVG              COMMONAREA_MODE
##                        69.87                        69.87
##              COMMONAREA_MEDI       NONLIVINGAPARTMENTS_AVG
##                        69.87                        69.43
##     NONLIVINGAPARTMENTS_MODE      NONLIVINGAPARTMENTS_MEDI
##                        69.43                        69.43
##         LIVINGAPARTMENTS_AVG        LIVINGAPARTMENTS_MODE
##                        68.35                        68.35
##        LIVINGAPARTMENTS_MEDI               FLOORSMIN_AVG
##                        68.35                        67.85
##               FLOORSMIN_MODE              FLOORSMIN_MEDI
##                        67.85                        67.85
##              YEARS_BUILD_AVG             YEARS_BUILD_MODE
##                        66.50                        66.50
##             YEARS_BUILD_MEDI                 OWN_CAR_AGE
##                        66.50                        65.99
##                LANDAREA_AVG                LANDAREA_MODE
##                        59.38                        59.38
##                LANDAREA_MEDI             BASEMENTAREA_AVG
##                        59.38                        58.52
##            BASEMENTAREA_MODE            BASEMENTAREA_MEDI
##                        58.52                        58.52
##                EXT_SOURCE_1            NONLIVINGAREA_AVG
##                        56.38                        55.18
##           NONLIVINGAREA_MODE           NONLIVINGAREA_MEDI
##                        55.18                        55.18
##                ELEVATORS_AVG               ELEVATORS_MODE
##                        53.30                        53.30
##               ELEVATORS_MEDI               APARTMENTS_AVG
##                        53.30                        50.75
##              APARTMENTS_MODE              APARTMENTS_MEDI
##                        50.75                        50.75
##                ENTRANCES_AVG               ENTRANCES_MODE
##                        50.35                        50.35
##               ENTRANCES_MEDI               LIVINGAREA_AVG
##                        50.35                        50.19
##              LIVINGAREA_MODE              LIVINGAREA_MEDI
##                        50.19                        50.19
```

```
##                FLOORSMAX_AVG                FLOORSMAX_MODE
##                        49.76                         49.76
##              FLOORSMAX_MEDI  YEARS_BEGINEXPLUATATION_AVG
##                        49.76                         48.78
## YEARS_BEGINEXPLUATATION_MODE YEARS_BEGINEXPLUATATION_MEDI
##                        48.78                         48.78
##               TOTALAREA_MODE                 EXT_SOURCE_3
##                        48.27                         19.83
##   AMT_REQ_CREDIT_BUREAU_HOUR    AMT_REQ_CREDIT_BUREAU_DAY
##                        13.50                         13.50
##   AMT_REQ_CREDIT_BUREAU_WEEK    AMT_REQ_CREDIT_BUREAU_MON
##                        13.50                         13.50
##    AMT_REQ_CREDIT_BUREAU_QRT   AMT_REQ_CREDIT_BUREAU_YEAR
##                        13.50                         13.50
##     OBS_30_CNT_SOCIAL_CIRCLE     DEF_30_CNT_SOCIAL_CIRCLE
##                         0.33                          0.33
##     OBS_60_CNT_SOCIAL_CIRCLE     DEF_60_CNT_SOCIAL_CIRCLE
##                         0.33                          0.33
##                 EXT_SOURCE_2              AMT_GOODS_PRICE
##                         0.21                          0.09
##                   SK_ID_CURR                        TARGET
##                         0.00                          0.00
##           NAME_CONTRACT_TYPE                  CODE_GENDER
##                         0.00                          0.00
##                 FLAG_OWN_CAR              FLAG_OWN_REALTY
##                         0.00                          0.00
##                 CNT_CHILDREN             AMT_INCOME_TOTAL
##                         0.00                          0.00
##                   AMT_CREDIT                  AMT_ANNUITY
##                         0.00                          0.00
##              NAME_TYPE_SUITE             NAME_INCOME_TYPE
##                         0.00                          0.00
##          NAME_EDUCATION_TYPE           NAME_FAMILY_STATUS
##                         0.00                          0.00
##            NAME_HOUSING_TYPE   REGION_POPULATION_RELATIVE
##                         0.00                          0.00
##                   DAYS_BIRTH                DAYS_EMPLOYED
##                         0.00                          0.00
##            DAYS_REGISTRATION              DAYS_ID_PUBLISH
##                         0.00                          0.00
##                   FLAG_MOBIL               FLAG_EMP_PHONE
##                         0.00                          0.00
##              FLAG_WORK_PHONE             FLAG_CONT_MOBILE
##                         0.00                          0.00
##                   FLAG_PHONE                   FLAG_EMAIL
##                         0.00                          0.00
##              OCCUPATION_TYPE              CNT_FAM_MEMBERS
##                         0.00                          0.00
##         REGION_RATING_CLIENT  REGION_RATING_CLIENT_W_CITY
##                         0.00                          0.00
##   WEEKDAY_APPR_PROCESS_START      HOUR_APPR_PROCESS_START
##                         0.00                          0.00
##   REG_REGION_NOT_LIVE_REGION   REG_REGION_NOT_WORK_REGION
##                         0.00                          0.00
```

```
##  LIVE_REGION_NOT_WORK_REGION       REG_CITY_NOT_LIVE_CITY
##                         0.00                         0.00
##       REG_CITY_NOT_WORK_CITY      LIVE_CITY_NOT_WORK_CITY
##                         0.00                         0.00
##            ORGANIZATION_TYPE           FONDKAPREMONT_MODE
##                         0.00                         0.00
##               HOUSETYPE_MODE           WALLSMATERIAL_MODE
##                         0.00                         0.00
##          EMERGENCYSTATE_MODE       DAYS_LAST_PHONE_CHANGE
##                         0.00                         0.00
##              FLAG_DOCUMENT_2              FLAG_DOCUMENT_3
##                         0.00                         0.00
##              FLAG_DOCUMENT_4              FLAG_DOCUMENT_5
##                         0.00                         0.00
##              FLAG_DOCUMENT_6              FLAG_DOCUMENT_7
##                         0.00                         0.00
##              FLAG_DOCUMENT_8              FLAG_DOCUMENT_9
##                         0.00                         0.00
##             FLAG_DOCUMENT_10             FLAG_DOCUMENT_11
##                         0.00                         0.00
##             FLAG_DOCUMENT_12             FLAG_DOCUMENT_13
##                         0.00                         0.00
##             FLAG_DOCUMENT_14             FLAG_DOCUMENT_15
##                         0.00                         0.00
##             FLAG_DOCUMENT_16             FLAG_DOCUMENT_17
##                         0.00                         0.00
##             FLAG_DOCUMENT_18             FLAG_DOCUMENT_19
##                         0.00                         0.00
##             FLAG_DOCUMENT_20             FLAG_DOCUMENT_21
##                         0.00                         0.00
```

```
#Set NA threshold
NAthreshold <- 35

#count the number of attributes above threshold
sum(sapply(clean_train, function(x)
       round(100*sum(is.na(x))/length(x),2))>NAthreshold)
```

```
## [1] 45
```

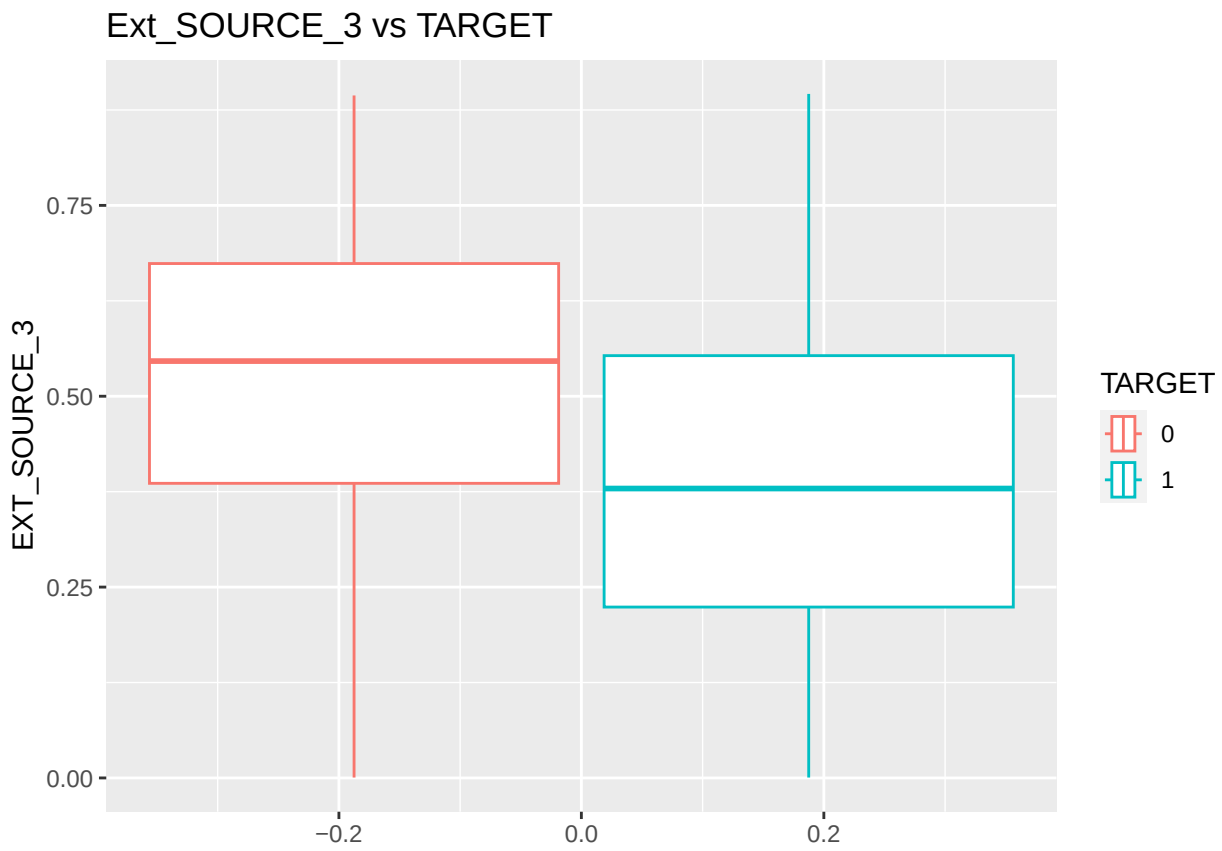
```
#summarize remaining attributes with NAs below threshold

summary(clean_train[c("EXT_SOURCE_3", "AMT_REQ_CREDIT_BUREAU_HOUR","AMT_REQ_CREDIT_BUREAU_DAY",  "AMT_R
```

```
##   EXT_SOURCE_3   AMT_REQ_CREDIT_BUREAU_HOUR AMT_REQ_CREDIT_BUREAU_DAY
##  Min.   :0.00    Min.   :0.00               Min.   :0.00
##  1st Qu.:0.37    1st Qu.:0.00               1st Qu.:0.00
##  Median :0.54    Median :0.00               Median :0.00
##  Mean   :0.51    Mean   :0.01               Mean   :0.01
##  3rd Qu.:0.67    3rd Qu.:0.00               3rd Qu.:0.00
##  Max.   :0.90    Max.   :4.00               Max.   :9.00
##  NA's   :60965   NA's   :41519              NA's   :41519
##  AMT_REQ_CREDIT_BUREAU_WEEK AMT_REQ_CREDIT_BUREAU_MON AMT_REQ_CREDIT_BUREAU_QRT
```

```
##  Min.   :0.00              Min.   : 0.00            Min.   :  0.00
##  1st Qu.:0.00              1st Qu.: 0.00            1st Qu.:  0.00
##  Median :0.00              Median : 0.00            Median :  0.00
##  Mean   :0.03              Mean   : 0.27            Mean   :  0.27
##  3rd Qu.:0.00              3rd Qu.: 0.00            3rd Qu.:  0.00
##  Max.   :8.00              Max.   :27.00            Max.   :261.00
##  NA's   :41519             NA's   :41519            NA's   :41519
##  AMT_REQ_CREDIT_BUREAU_YEAR OBS_30_CNT_SOCIAL_CIRCLE DEF_30_CNT_SOCIAL_CIRCLE
##  Min.   : 0.0               Min.   :  0.000          Min.   : 0.0000
##  1st Qu.: 0.0               1st Qu.:  0.000          1st Qu.: 0.0000
##  Median : 1.0               Median :  0.000          Median : 0.0000
##  Mean   : 1.9               Mean   :  1.422          Mean   : 0.1434
##  3rd Qu.: 3.0               3rd Qu.:  2.000          3rd Qu.: 0.0000
##  Max.   :25.0               Max.   :348.000          Max.   :34.0000
##  NA's   :41519              NA's   :1021             NA's   :1021
##  OBS_60_CNT_SOCIAL_CIRCLE DEF_60_CNT_SOCIAL_CIRCLE  EXT_SOURCE_2
##  Min.   :  0.000          Min.   : 0.0             Min.   :0.0000
##  1st Qu.:  0.000          1st Qu.: 0.0             1st Qu.:0.3925
##  Median :  0.000          Median : 0.0             Median :0.5660
##  Mean   :  1.405          Mean   : 0.1             Mean   :0.5144
##  3rd Qu.:  2.000          3rd Qu.: 0.0             3rd Qu.:0.6636
##  Max.   :344.000          Max.   :24.0             Max.   :0.8550
##  NA's   :1021             NA's   :1021             NA's   :660
##  AMT_GOODS_PRICE
##  Min.   :  40500
##  1st Qu.: 238500
##  Median : 450000
##  Mean   : 538396
##  3rd Qu.: 679500
##  Max.   :4050000
##  NA's   :278
```

```
#Visualize EXT_SOURCE_3 Boxplot
ggplot(data = clean_train, aes(x=EXT_SOURCE_3, color= TARGET)) + geom_boxplot() + labs(title = "Ext_SOU
```

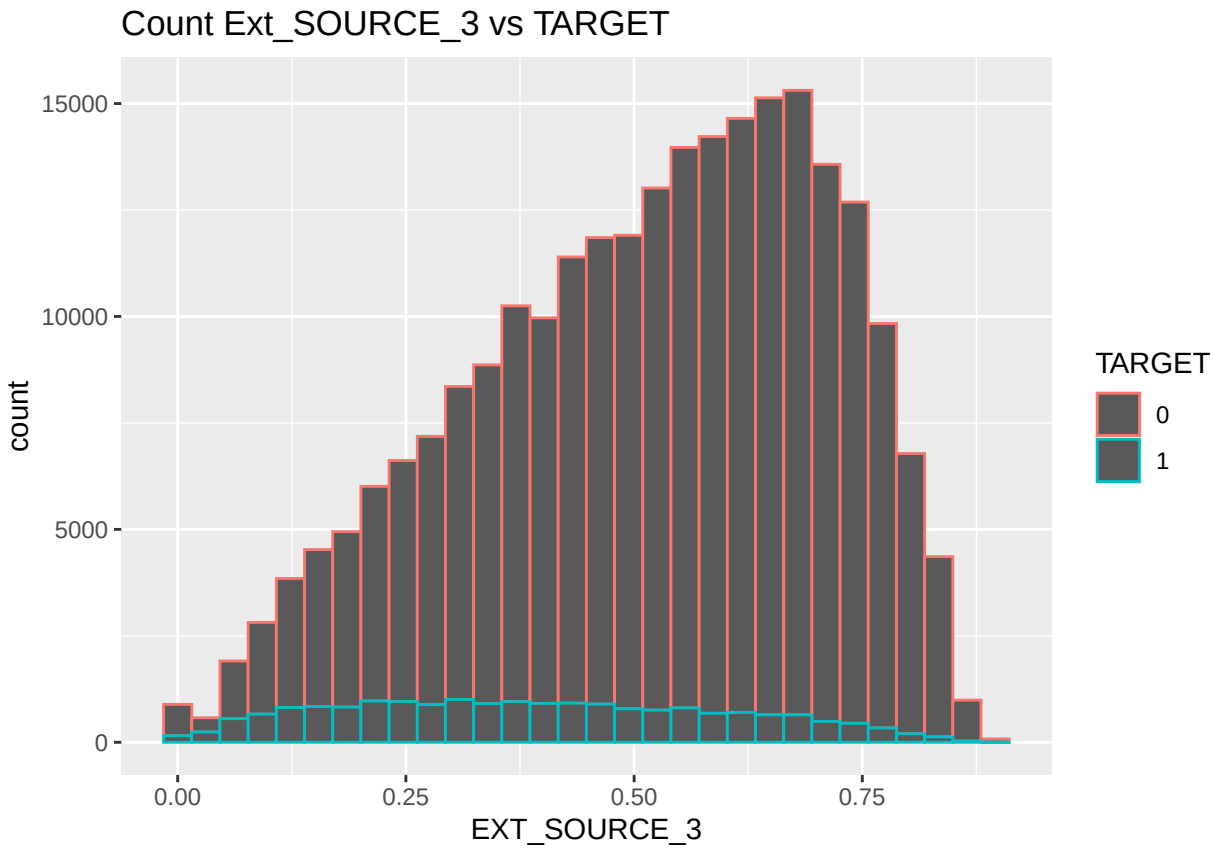
```
## Warning: Removed 60965 rows containing non-finite values (`stat_boxplot()`).
```




```
#Visualize EXT_SOURCE_3 histogram
ggplot(data = clean_train, aes(x=EXT_SOURCE_3, color= TARGET)) + geom_histogram() + labs(title = "Count
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 60965 rows containing non-finite values (`stat_bin()`).
```




```
#Check EXT_SOURCE_3 distributor vs Target
clean_train %>%
    group_by(TARGET) %>%
    summarise(mean = mean(EXT_SOURCE_3, na.rm = TRUE))
```

```
## # A tibble: 2 x 2
##   TARGET  mean
##   <fct>  <dbl>
## 1 0      0.521
## 2 1      0.391
```

```
#Visualize AMT_REQ Group
clean_train %>%
  group_by(AMT_REQ_CREDIT_BUREAU_HOUR) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100))
```

```
## # A tibble: 6 x 3
##   AMT_REQ_CREDIT_BUREAU_HOUR      n      freq
##                        <int>  <int>     <dbl>
## 1                          0 264366 86.0
## 2                          1   1560  0.507
## 3                          2     56  0.0182
## 4                          3      9  0.00293
## 5                          4      1  0.000325
## 6                         NA  41519 13.5
```

```
clean_train %>%
  group_by(AMT_REQ_CREDIT_BUREAU_DAY) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100))
```

```
## # A tibble: 10 x 3
##    AMT_REQ_CREDIT_BUREAU_DAY      n      freq
##                        <int>  <int>     <dbl>
##  1                         0 264503 86.0
##  2                         1   1292  0.420
##  3                         2    106  0.0345
##  4                         3     45  0.0146
##  5                         4     26  0.00845
##  6                         5      9  0.00293
##  7                         6      8  0.00260
##  8                         8      1  0.000325
##  9                         9      2  0.000650
## 10                        NA  41519 13.5
```

```
clean_train %>%
  group_by(AMT_REQ_CREDIT_BUREAU_WEEK) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100))
```

```
## # A tibble: 10 x 3
##    AMT_REQ_CREDIT_BUREAU_WEEK      n      freq
##                         <int>  <int>     <dbl>
##  1                          0 257456 83.7
##  2                          1   8208  2.67
##  3                          2    199  0.0647
##  4                          3     58  0.0189
##  5                          4     34  0.0111
##  6                          5     10  0.00325
##  7                          6     20  0.00650
##  8                          7      2  0.000650
##  9                          8      5  0.00163
## 10                         NA  41519 13.5
```

```
clean_train %>%
  group_by(AMT_REQ_CREDIT_BUREAU_MON) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## # A tibble: 25 x 3
##    AMT_REQ_CREDIT_BUREAU_MON      n      freq
##                        <int>  <int>     <dbl>
##  1                         0 222233 72.3
##  2                         1  33147 10.8
##  3                         2   5386  1.75
##  4                         3   1991  0.647
##  5                         4   1076  0.350
```

```
##  6                          5    602  0.196
##  7                          6    343  0.112
##  8                          7    298  0.0969
##  9                          8    185  0.0602
## 10                          9    206  0.0670
## 11                         10    132  0.0429
## 12                         11    119  0.0387
## 13                         12     77  0.0250
## 14                         13     72  0.0234
## 15                         14     40  0.0130
## 16                         15     35  0.0114
## 17                         16     23  0.00748
## 18                         17     14  0.00455
## 19                         18      6  0.00195
## 20                         19      3  0.000976
## 21                         22      1  0.000325
## 22                         23      1  0.000325
## 23                         24      1  0.000325
## 24                         27      1  0.000325
## 25                         NA  41519 13.5
```

```
clean_train %>%
  group_by(AMT_REQ_CREDIT_BUREAU_QRT) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## # A tibble: 12 x 3
##    AMT_REQ_CREDIT_BUREAU_QRT      n      freq
##                        <int>  <int>     <dbl>
##  1                         0 215417 70.1
##  2                         1  33862 11.0
##  3                         2  14412  4.69
##  4                         3   1717  0.558
##  5                         4    476  0.155
##  6                         5     64  0.0208
##  7                         6     28  0.00911
##  8                         7      7  0.00228
##  9                         8      7  0.00228
## 10                        19      1  0.000325
## 11                       261      1  0.000325
## 12                        NA  41519 13.5
```

```
clean_train %>%
  group_by(AMT_REQ_CREDIT_BUREAU_YEAR,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'AMT_REQ_CREDIT_BUREAU_YEAR'. You can
## override using the `.groups` argument.
```

```
## # A tibble: 43 x 4
```

```
## # Groups:   AMT_REQ_CREDIT_BUREAU_YEAR [26]
##    AMT_REQ_CREDIT_BUREAU_YEAR TARGET     n   freq
##                         <int> <fct>  <int>  <dbl>
##  1                          0 0      66678  92.9
##  2                          0 1       5123   7.13
##  3                          1 0      58755  92.7
##  4                          1 1       4650   7.33
##  5                          2 0      46124  91.9
##  6                          2 1       4068   8.10
##  7                          3 0      30952  92.0
##  8                          3 1       2676   7.96
##  9                          4 0      19004  91.7
## 10                          4 1       1710   8.26
## 11                          5 0      11049  91.7
## 12                          5 1       1003   8.32
## 13                          6 0       6335  90.9
## 14                          6 1        632   9.07
## 15                          7 0       3513  90.8
## 16                          7 1        356   9.20
## 17                          8 0       1944  91.4
## 18                          8 1        183   8.60
## 19                          9 0        977  89.1
## 20                          9 1        119  10.9
## 21                         10 0         19  86.4
## 22                         10 1          3  13.6
## 23                         11 0         29  93.5
## 24                         11 1          2   6.45
## 25                         12 0         28  93.3
## 26                         12 1          2   6.67
## 27                         13 0         18  94.7
## 28                         13 1          1   5.26
## 29                         14 0          7  70
## 30                         14 1          3  30
## 31                         15 0          6 100
## 32                         16 0          2  66.7
## 33                         16 1          1  33.3
## 34                         17 0          7 100
## 35                         18 0          4 100
## 36                         19 0          4 100
## 37                         20 0          1 100
## 38                         21 0          1 100
## 39                         22 1          1 100
## 40                         23 0          1 100
## 41                         25 0          1 100
## 42                         NA 0      37227  89.7
## 43                         NA 1       4292  10.3
```

```
#Find % of blanks by attribute
sort(sapply(clean_train, function(x)
       round(100*sum(x=="")/length(x),2)),decreasing =TRUE)
```

```
##          FONDKAPREMONT_MODE           WALLSMATERIAL_MODE
##                       68.39                        50.84
##              HOUSETYPE_MODE         EMERGENCYSTATE_MODE
```

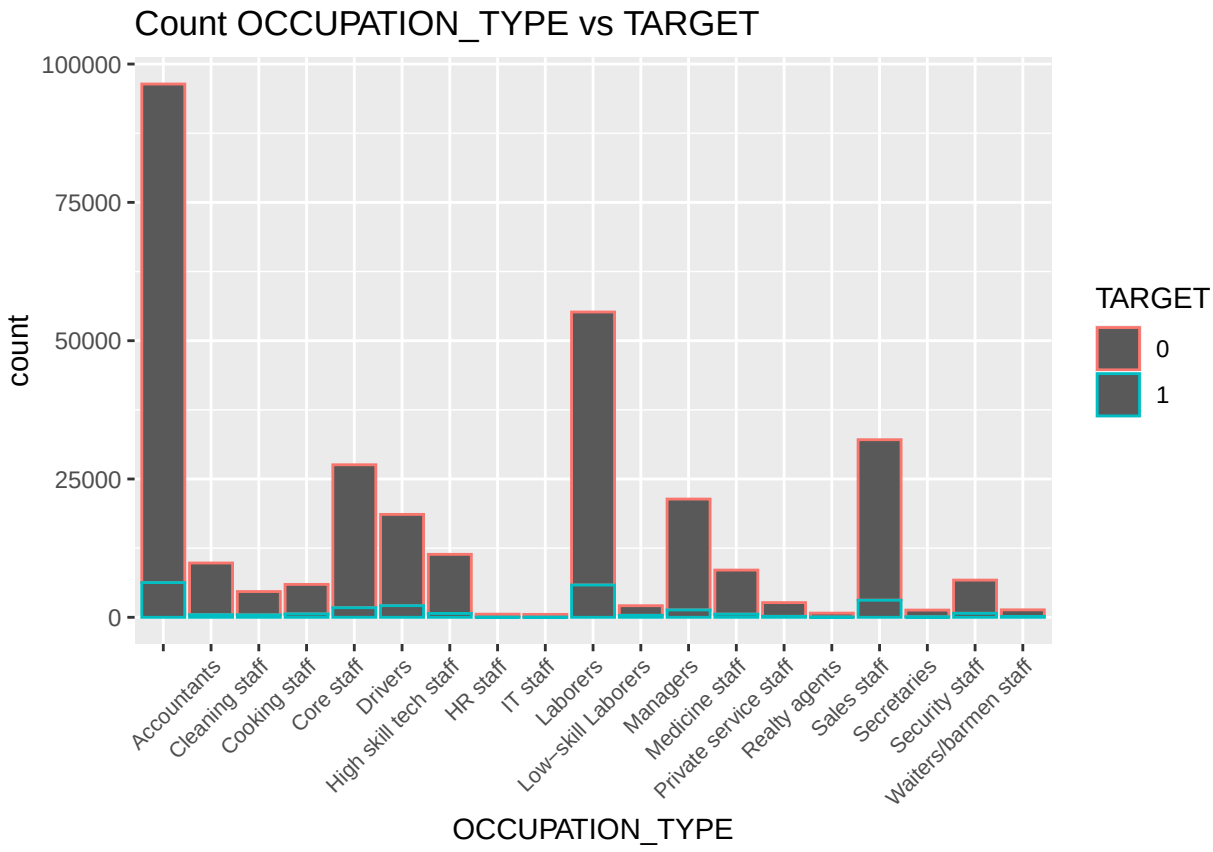
```
##                     50.18                     47.40
##            OCCUPATION_TYPE            NAME_TYPE_SUITE
##                     31.35                      0.42
##                SK_ID_CURR                    TARGET
##                      0.00                      0.00
##         NAME_CONTRACT_TYPE                CODE_GENDER
##                      0.00                      0.00
##               FLAG_OWN_CAR            FLAG_OWN_REALTY
##                      0.00                      0.00
##               CNT_CHILDREN           AMT_INCOME_TOTAL
##                      0.00                      0.00
##                 AMT_CREDIT           NAME_INCOME_TYPE
##                      0.00                      0.00
##        NAME_EDUCATION_TYPE         NAME_FAMILY_STATUS
##                      0.00                      0.00
##          NAME_HOUSING_TYPE  REGION_POPULATION_RELATIVE
##                      0.00                      0.00
##                DAYS_BIRTH              DAYS_EMPLOYED
##                      0.00                      0.00
##          DAYS_REGISTRATION            DAYS_ID_PUBLISH
##                      0.00                      0.00
##                FLAG_MOBIL             FLAG_EMP_PHONE
##                      0.00                      0.00
##            FLAG_WORK_PHONE           FLAG_CONT_MOBILE
##                      0.00                      0.00
##                FLAG_PHONE                 FLAG_EMAIL
##                      0.00                      0.00
##       REGION_RATING_CLIENT REGION_RATING_CLIENT_W_CITY
##                      0.00                      0.00
##  WEEKDAY_APPR_PROCESS_START    HOUR_APPR_PROCESS_START
##                      0.00                      0.00
##  REG_REGION_NOT_LIVE_REGION  REG_REGION_NOT_WORK_REGION
##                      0.00                      0.00
## LIVE_REGION_NOT_WORK_REGION     REG_CITY_NOT_LIVE_CITY
##                      0.00                      0.00
##     REG_CITY_NOT_WORK_CITY    LIVE_CITY_NOT_WORK_CITY
##                      0.00                      0.00
##           ORGANIZATION_TYPE            FLAG_DOCUMENT_2
##                      0.00                      0.00
##            FLAG_DOCUMENT_3            FLAG_DOCUMENT_4
##                      0.00                      0.00
##            FLAG_DOCUMENT_5            FLAG_DOCUMENT_6
##                      0.00                      0.00
##            FLAG_DOCUMENT_7            FLAG_DOCUMENT_8
##                      0.00                      0.00
##            FLAG_DOCUMENT_9           FLAG_DOCUMENT_10
##                      0.00                      0.00
##           FLAG_DOCUMENT_11           FLAG_DOCUMENT_12
##                      0.00                      0.00
##           FLAG_DOCUMENT_13           FLAG_DOCUMENT_14
##                      0.00                      0.00
##           FLAG_DOCUMENT_15           FLAG_DOCUMENT_16
##                      0.00                      0.00
##           FLAG_DOCUMENT_17           FLAG_DOCUMENT_18
```

```
##                         0.00                         0.00
##            FLAG_DOCUMENT_19            FLAG_DOCUMENT_20
##                         0.00                         0.00
##            FLAG_DOCUMENT_21
##                         0.00
```

```
#summarize remaining attributes with blanks below threshold
summary(clean_train[c("OCCUPATION_TYPE", "NAME_TYPE_SUITE")])
```

```
##     OCCUPATION_TYPE         NAME_TYPE_SUITE
##             :96391   Unaccompanied  :248526
##  Laborers   :55186   Family         : 40149
##  Sales staff:32102   Spouse, partner: 11370
##  Core staff :27570   Children       :  3267
##  Managers   :21371   Other_B        :  1770
##  Drivers    :18603                  :  1292
##  (Other)    :56288   (Other)        :  1137
```

```
#Visualize OCCUPATION_TYPE
ggplot(data = clean_train, aes(x=OCCUPATION_TYPE, color = TARGET)) + geom_bar() + labs(title = "Count O
```



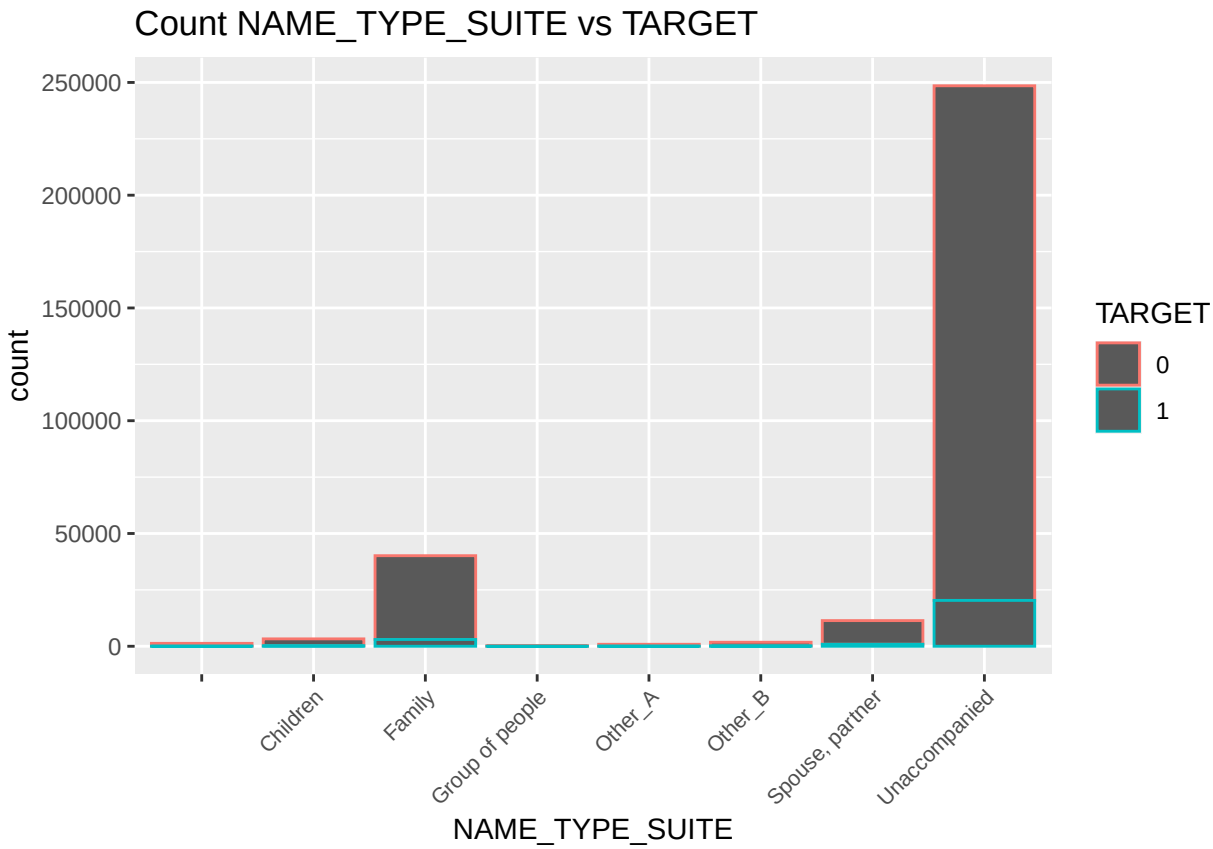

```
clean_train %>%
  group_by(OCCUPATION_TYPE,TARGET) %>%
    summarise(n=n()) %>%
```

```
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'OCCUPATION_TYPE'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 38 x 4
## # Groups:   OCCUPATION_TYPE [19]
##    OCCUPATION_TYPE         TARGET     n  freq
##    <fct>                   <fct>  <int> <dbl>
##  1 ""                      0      90113 93.5
##  2 ""                      1       6278  6.51
##  3 "Accountants"           0       9339 95.2
##  4 "Accountants"           1        474  4.83
##  5 "Cleaning staff"        0       4206 90.4
##  6 "Cleaning staff"        1        447  9.61
##  7 "Cooking staff"         0       5325 89.6
##  8 "Cooking staff"         1        621 10.4
##  9 "Core staff"            0      25832 93.7
## 10 "Core staff"            1       1738  6.30
## 11 "Drivers"               0      16496 88.7
## 12 "Drivers"               1       2107 11.3
## 13 "High skill tech staff" 0      10679 93.8
## 14 "High skill tech staff" 1        701  6.16
## 15 "HR staff"              0        527 93.6
## 16 "HR staff"              1         36  6.39
## 17 "IT staff"              0        492 93.5
## 18 "IT staff"              1         34  6.46
## 19 "Laborers"              0      49348 89.4
## 20 "Laborers"              1       5838 10.6
## 21 "Low-skill Laborers"    0       1734 82.8
## 22 "Low-skill Laborers"    1        359 17.2
## 23 "Managers"              0      20043 93.8
## 24 "Managers"              1       1328  6.21
## 25 "Medicine staff"        0       7965 93.3
## 26 "Medicine staff"        1        572  6.70
## 27 "Private service staff" 0       2477 93.4
## 28 "Private service staff" 1        175  6.60
## 29 "Realty agents"         0        692 92.1
## 30 "Realty agents"         1         59  7.86
## 31 "Sales staff"           0      29010 90.4
## 32 "Sales staff"           1       3092  9.63
## 33 "Secretaries"           0       1213 93.0
## 34 "Secretaries"           1         92  7.05
## 35 "Security staff"        0       5999 89.3
## 36 "Security staff"        1        722 10.7
## 37 "Waiters/barmen staff"  0       1196 88.7
## 38 "Waiters/barmen staff"  1        152 11.3
```

```
#Visualize NAME_TYPE_SUITE
ggplot(data = clean_train, aes(x=NAME_TYPE_SUITE, color = TARGET)) + geom_bar() + labs(title = "Count N
```




```
clean_train %>%
  group_by(NAME_TYPE_SUITE,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'NAME_TYPE_SUITE'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 16 x 4
## # Groups:   NAME_TYPE_SUITE [8]
##    NAME_TYPE_SUITE   TARGET      n  freq
##    <fct>             <fct>   <int> <dbl>
##  1 ""                0        1222 94.6
##  2 ""                1          70  5.42
##  3 "Children"        0        3026 92.6
##  4 "Children"        1         241  7.38
##  5 "Family"          0       37140 92.5
##  6 "Family"          1        3009  7.49
##  7 "Group of people" 0         248 91.5
##  8 "Group of people" 1          23  8.49
##  9 "Other_A"         0         790 91.2
## 10 "Other_A"         1          76  8.78
## 11 "Other_B"         0        1596 90.2
## 12 "Other_B"         1         174  9.83
```

```
## 13 "Spouse, partner" 0       10475 92.1
## 14 "Spouse, partner" 1         895  7.87
## 15 "Unaccompanied"   0      228189 91.8
## 16 "Unaccompanied"   1       20337  8.18
```

Low Variance

To address low variance within a variable we use a filter to select any variables with 5% or less variance. From this we find 35 variables with less than 5% variance, many of these were also addressed with the N/A group. We would suggest removing all of these from model consideration, with the exception of DAYS_EOMPLOYED which will be addressed in the potential errors section.

```
# Run Filter for low variance
nearzero <- nearZeroVar(clean_train, freqCut = 95/5 )

# Check the summary of each of the Low variance variables
summary(clean_train[c(nearzero)])
```

```
##  DAYS_EMPLOYED    FLAG_MOBIL FLAG_CONT_MOBILE REG_REGION_NOT_LIVE_REGION
##  Min.   :-17912   0:     1   0:   574         0:302854
##  1st Qu.: -2760   1:307510   1:306937         1:  4657
##  Median : -1213
##  Mean   : 63815
##  3rd Qu.:  -289
##  Max.   :365243
##
##  LIVE_REGION_NOT_WORK_REGION BASEMENTAREA_AVG  LANDAREA_AVG
##  0:295008                    Min.   :0.00     Min.   :0.00
##  1: 12503                    1st Qu.:0.04     1st Qu.:0.02
##                              Median :0.08     Median :0.05
##                              Mean   :0.09     Mean   :0.07
##                              3rd Qu.:0.11     3rd Qu.:0.09
##                              Max.   :1.00     Max.   :1.00
##                              NA's   :179943   NA's   :182590
##  NONLIVINGAREA_AVG BASEMENTAREA_MODE LANDAREA_MODE    NONLIVINGAREA_MODE
##  Min.   :0.00      Min.   :0.00      Min.   :0.00     Min.   :0.00
##  1st Qu.:0.00      1st Qu.:0.04      1st Qu.:0.02     1st Qu.:0.00
##  Median :0.00      Median :0.07      Median :0.05     Median :0.00
##  Mean   :0.03      Mean   :0.09      Mean   :0.06     Mean   :0.03
##  3rd Qu.:0.03      3rd Qu.:0.11      3rd Qu.:0.08     3rd Qu.:0.02
##  Max.   :1.00      Max.   :1.00      Max.   :1.00     Max.   :1.00
##  NA's   :169682    NA's   :179943    NA's   :182590   NA's   :169682
##  BASEMENTAREA_MEDI LANDAREA_MEDI    NONLIVINGAREA_MEDI FLAG_DOCUMENT_2
##  Min.   :0.00      Min.   :0.00     Min.   :0.00       0:307498
##  1st Qu.:0.04      1st Qu.:0.02     1st Qu.:0.00       1:    13
##  Median :0.08      Median :0.05     Median :0.00
##  Mean   :0.09      Mean   :0.07     Mean   :0.03
##  3rd Qu.:0.11      3rd Qu.:0.09     3rd Qu.:0.03
##  Max.   :1.00      Max.   :1.00     Max.   :1.00
##  NA's   :179943    NA's   :182590   NA's   :169682
##  FLAG_DOCUMENT_4 FLAG_DOCUMENT_5 FLAG_DOCUMENT_7 FLAG_DOCUMENT_9
##  0:307486        0:302863        0:307452        0:306313
```

```
##  1:    25        1:  4648        1:    59        1:  1198
##
##
##
##
##
##  FLAG_DOCUMENT_10 FLAG_DOCUMENT_11 FLAG_DOCUMENT_12 FLAG_DOCUMENT_13
##  0:307504         0:306308         0:307509         0:306427
##  1:     7         1:  1203         1:     2         1:  1084
##
##
##
##
##
##  FLAG_DOCUMENT_14 FLAG_DOCUMENT_15 FLAG_DOCUMENT_16 FLAG_DOCUMENT_17
##  0:306608         0:307139         0:304458         0:307429
##  1:   903         1:   372         1:  3053         1:    82
##
##
##
##
##
##  FLAG_DOCUMENT_18 FLAG_DOCUMENT_19 FLAG_DOCUMENT_20 FLAG_DOCUMENT_21
##  0:305011         0:307328         0:307355         0:307408
##  1:  2500         1:   183         1:   156         1:   103
##
##
##
##
##
##  AMT_REQ_CREDIT_BUREAU_HOUR AMT_REQ_CREDIT_BUREAU_DAY
##  Min.   :0.00               Min.   :0.00
##  1st Qu.:0.00               1st Qu.:0.00
##  Median :0.00               Median :0.00
##  Mean   :0.01               Mean   :0.01
##  3rd Qu.:0.00               3rd Qu.:0.00
##  Max.   :4.00               Max.   :9.00
##  NA's   :41519              NA's   :41519
##  AMT_REQ_CREDIT_BUREAU_WEEK
##  Min.   :0.00
##  1st Qu.:0.00
##  Median :0.00
##  Mean   :0.03
##  3rd Qu.:0.00
##  Max.   :8.00
##  NA's   :41519
```

Outliars and Potential Errors

A number of the Outliars/Potential Errors in variables that have not been addressed previously are:

- 4 XNA in CODE_GENDER. There are 4 observations listed as XNA in gender as this number is extremely small we would suggest imputing these 2 with male 2 with female.

- AMT_TOTAL_INCOME. In amount total income we have a few very large incomes that maybe errors or just very high incomes. With them drawing up the average we would suggest using a logged version of this variable to help prediction.
- DAYS_EMPLOYED in this variable it appears to be if no date is put the data is counting as 365243. For this we would need to check to make sure this assumption is correct. If it is replacing this value with 0 is a possibility.
- AMT_REQ_BUREAU_QRT in this case it appears to be an error outliar with a gap between the max value of 261 to the next value of 19, suggest removing or replacing with second max of 19.
- Social_Circle outliars in each group there is one observation that is significantly higher than other observations. Since in each group it is only one observation with this removing or replacing with the mean is possible.

```
#Address CODE_GENDER
clean_train %>%
  group_by(CODE_GENDER,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100))
```

```
## `summarise()` has grouped output by 'CODE_GENDER'. You can override using the
## `.groups` argument.
```

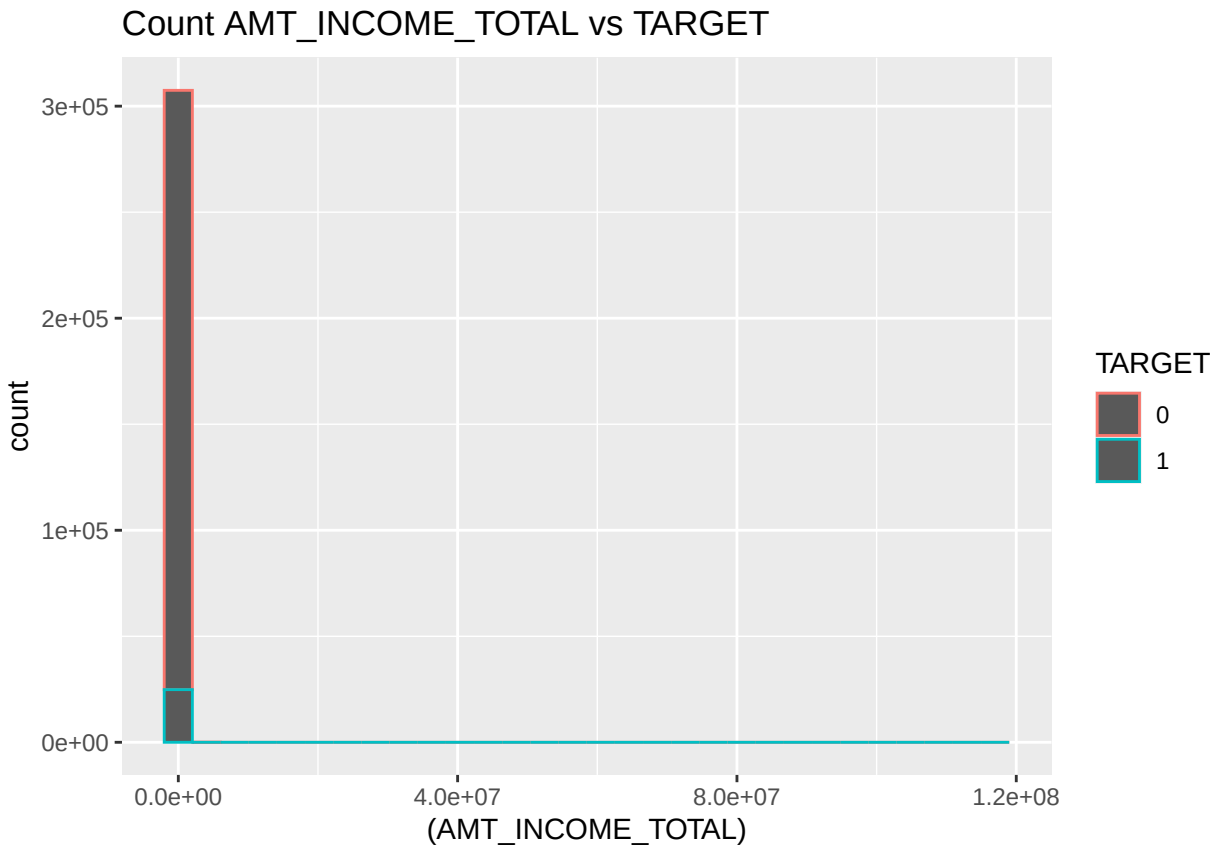
```
## # A tibble: 5 x 4
## # Groups:   CODE_GENDER [3]
##   CODE_GENDER TARGET      n   freq
##   <fct>       <fct>   <int>  <dbl>
## 1 F           0      188278  93.0
## 2 F           1       14170   7.00
## 3 M           0       94404  89.9
## 4 M           1       10655  10.1
## 5 XNA         0           4 100
```

```
#Address AMT_INCOME_TOTAL

##Check top 25 values of AMT_INCOME_TOTAL
format(head(sort(clean_train$AMT_INCOME_TOTAL,decreasing=TRUE), n = 25), big.mark = ",")
```

```
##  [1] "117,000,000" " 18,000,090" " 13,500,000" "  9,000,000" "  6,750,000"
##  [6] "  4,500,000" "  4,500,000" "  4,500,000" "  4,500,000" "  3,950,060"
## [11] "  3,825,000" "  3,600,000" "  3,600,000" "  3,375,000" "  3,375,000"
## [16] "  3,150,000" "  3,150,000" "  2,930,026" "  2,700,000" "  2,475,000"
## [21] "  2,250,000" "  2,250,000" "  2,250,000" "  2,250,000" "  2,250,000"
```

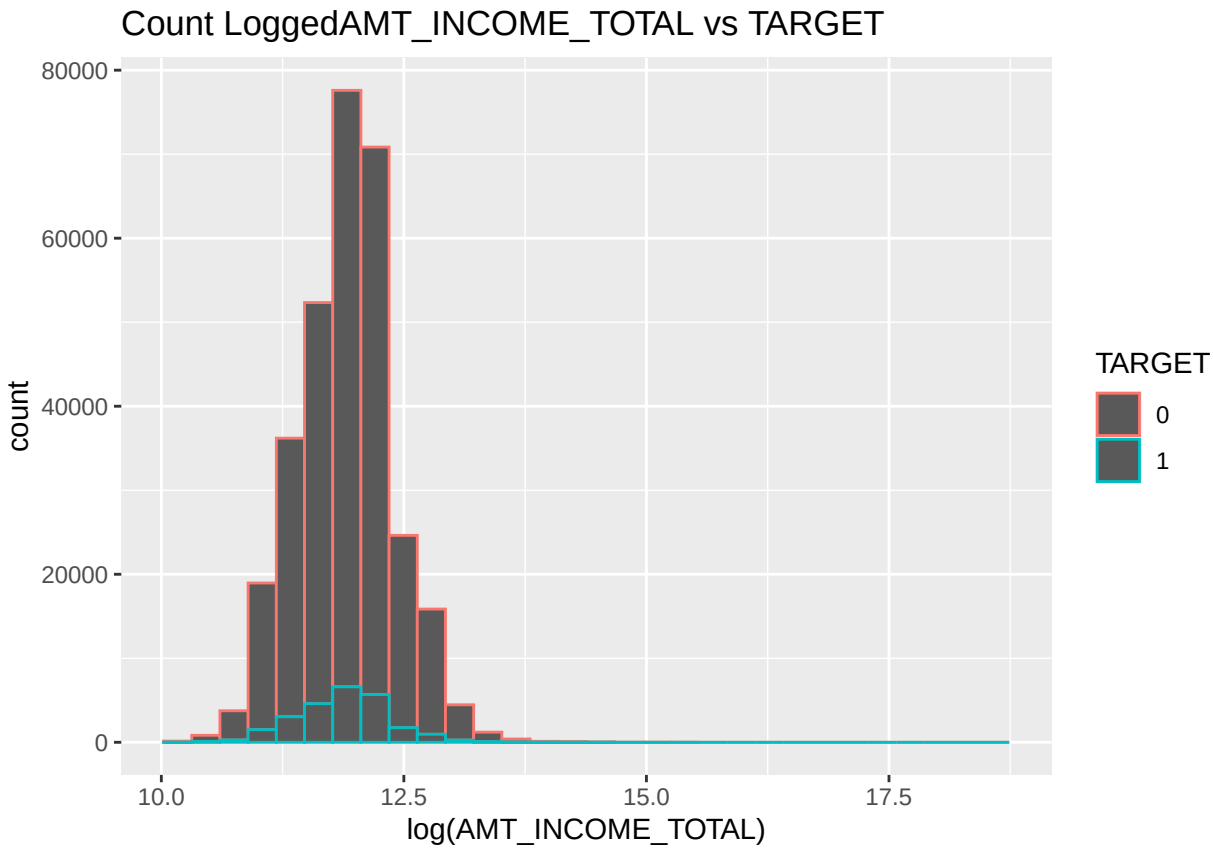
```
##Plot AMT_INCOME_TOTAL
ggplot(data = clean_train, aes(x=(AMT_INCOME_TOTAL), color = TARGET)) + geom_histogram() +labs(title =
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
##Plot same data with Logged Incomes
ggplot(data = clean_train, aes(x=log(AMT_INCOME_TOTAL), color = TARGET)) + geom_histogram() +labs(title
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

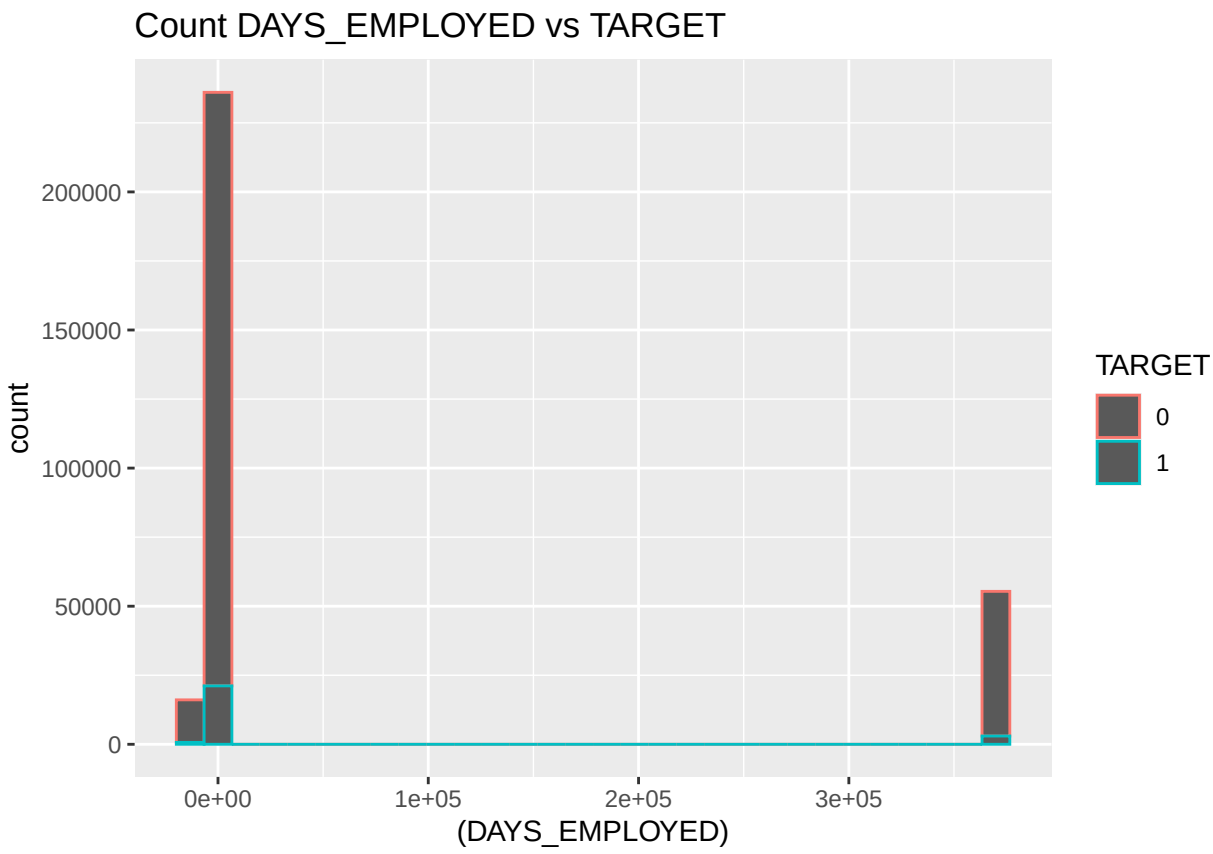



```
#Address DAYS_EMPLOYED

## Show Summary for Days Employed
summary(clean_train$DAYS_EMPLOYED)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -17912   -2760   -1213   63815    -289  365243
```

```
##Plot Days Employed
ggplot(data = clean_train, aes(x=(DAYS_EMPLOYED), color = TARGET)) + geom_histogram() +labs(title = "Co
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

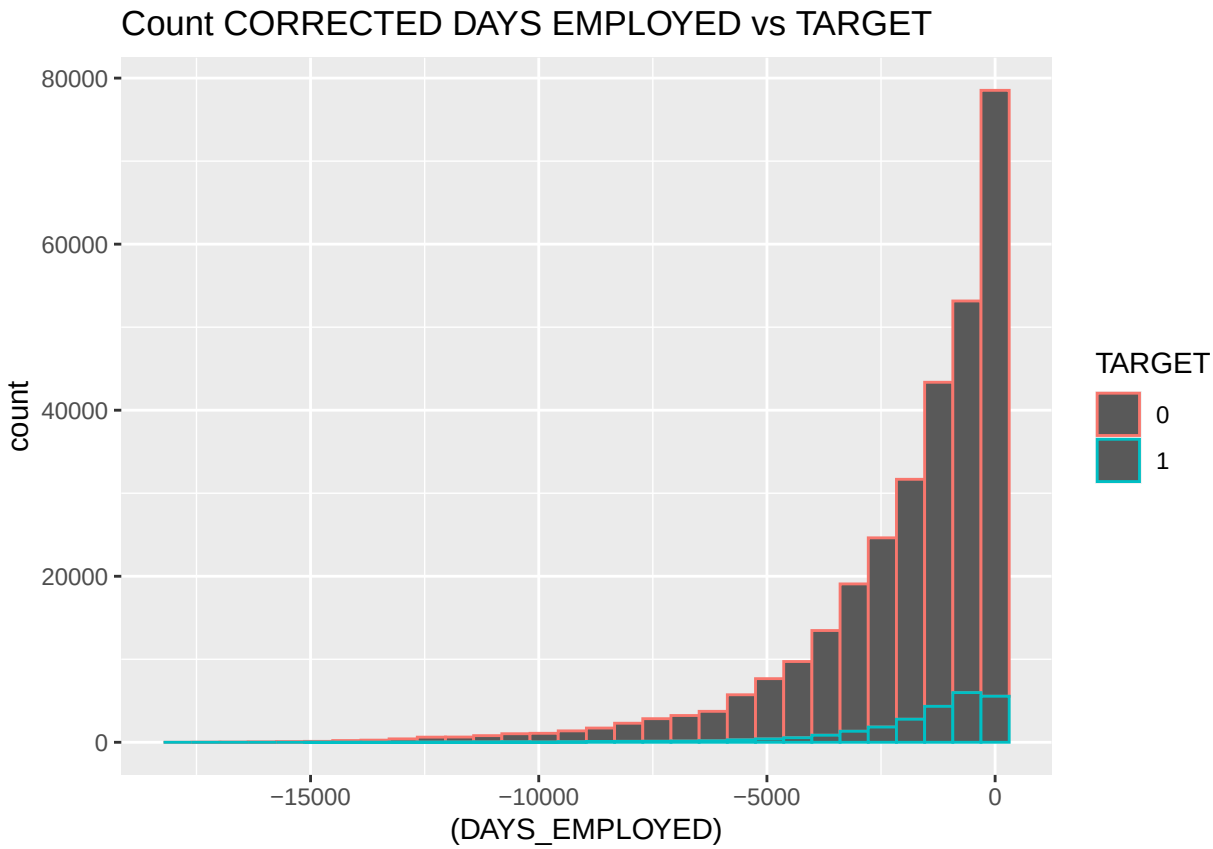



```
##Replace 365243 with 0
DEtest <- clean_train %>%
    mutate(DAYS_EMPLOYED = replace(DAYS_EMPLOYED, DAYS_EMPLOYED == 365243, 0))

##Recheck summary and plot
summary(DEtest$DAYS_EMPLOYED)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -17912   -2760   -1213   -1955    -289       0
```

```
ggplot(data = DEtest, aes(x=(DAYS_EMPLOYED), color = TARGET)) + geom_histogram() +labs(title = "Count C
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



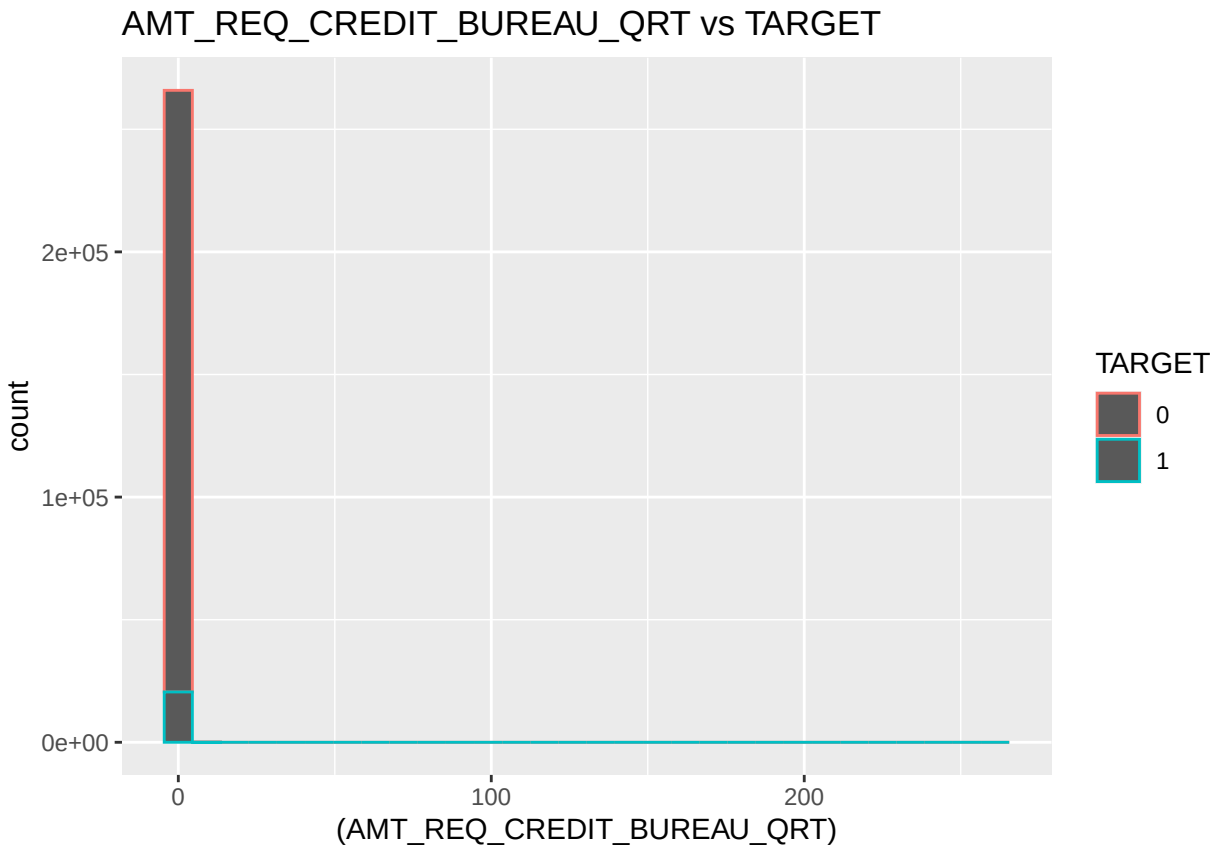

```
#Address AMT_REQ_BUREAU_QRT max

##Check Summary of AMT_REQ_BUREAU_QRT
summary(clean_train$AMT_REQ_CREDIT_BUREAU_QRT)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.00    0.27    0.00  261.00   41519
```

```
##Plot to check for outliar
ggplot(data = DEtest, aes(x=(AMT_REQ_CREDIT_BUREAU_QRT), color = TARGET)) + geom_histogram()+labs(title
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 41519 rows containing non-finite values (`stat_bin()`).
```




```
##Check top values
head(sort(clean_train$AMT_REQ_CREDIT_BUREAU_QRT, decreasing =TRUE))
```

```
## [1] 261  19   8   8   8   8
```

```
#Address outliars in OBS_30_CNT_SOCIAL_CIRCLE, DEF_30_CNT_SOCIAL_CIRCLE, OBS_60_CNT_SOCIAL_CIRCLE, DEF_

##Check summary
summary(clean_train[c('OBS_30_CNT_SOCIAL_CIRCLE', 'DEF_30_CNT_SOCIAL_CIRCLE', 'OBS_60_CNT_SOCIAL_CIRCLE
```

```
##  OBS_30_CNT_SOCIAL_CIRCLE DEF_30_CNT_SOCIAL_CIRCLE OBS_60_CNT_SOCIAL_CIRCLE
##  Min.   :  0.000          Min.   : 0.0000          Min.   :  0.000
##  1st Qu.:  0.000          1st Qu.: 0.0000          1st Qu.:  0.000
##  Median :  0.000          Median : 0.0000          Median :  0.000
##  Mean   :  1.422          Mean   : 0.1434          Mean   :  1.405
##  3rd Qu.:  2.000          3rd Qu.: 0.0000          3rd Qu.:  2.000
##  Max.   :348.000          Max.   :34.0000          Max.   :344.000
##  NA's   :1021             NA's   :1021             NA's   :1021
##  DEF_60_CNT_SOCIAL_CIRCLE
##  Min.   : 0.0
##  1st Qu.: 0.0
##  Median : 0.0
##  Mean   : 0.1
##  3rd Qu.: 0.0
##  Max.   :24.0
```

```
##  NA's   :1021
```

```
##Check top Values
format(head(sort(clean_train$OBS_30_CNT_SOCIAL_CIRCLE,decreasing=TRUE), n = 10), big.mark = ",")
```

```
##  [1] "348" " 47" " 30" " 30" " 29" " 28" " 27" " 27" " 27" " 27"
```

```
format(head(sort(clean_train$DEF_30_CNT_SOCIAL_CIRCLE,decreasing=TRUE), n = 10), big.mark = ",")
```

```
##  [1] "34" " 8" " 7" " 6" " 6" " 6" " 6" " 6" " 6" " 6"
```

```
format(head(sort(clean_train$OBS_60_CNT_SOCIAL_CIRCLE,decreasing=TRUE), n = 10), big.mark = ",")
```

```
##  [1] "344" " 47" " 30" " 29" " 29" " 28" " 27" " 27" " 27" " 27"
```

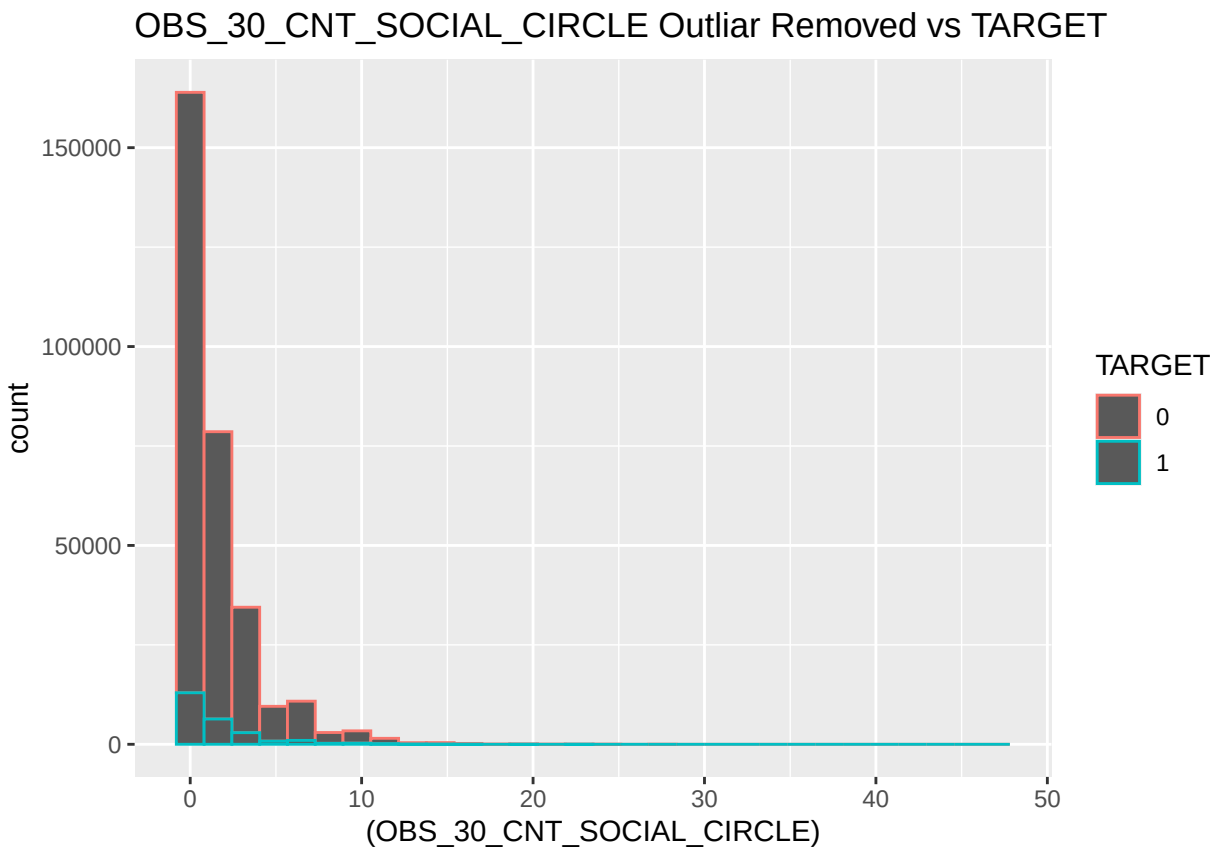
```
format(head(sort(clean_train$DEF_60_CNT_SOCIAL_CIRCLE,decreasing=TRUE), n = 10), big.mark = ",")
```

```
##  [1] "24" " 7" " 6" " 6" " 6" " 5" " 5" " 5" " 5" " 5"
```

```
#Test impact of imputing top value
SCtest <- clean_train %>%
    mutate(OBS_30_CNT_SOCIAL_CIRCLE = replace(OBS_30_CNT_SOCIAL_CIRCLE, OBS_30_CNT_SOCIAL_CIRCLE == 348

##Plot with outliar removed
ggplot(data = SCtest, aes(x=(OBS_30_CNT_SOCIAL_CIRCLE), color = TARGET)) + geom_histogram() + labs(titl
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 1021 rows containing non-finite values (`stat_bin()`).
```




```
##Check Summary
summary(SCtest$OBS_30_CNT_SOCIAL_CIRCLE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   0.000   0.000   1.421   2.000  47.000    1021
```

Additional Data Sets

In this section we look at some possibilities for additional data to be added to our training data. Here we use 2 examples, one from the bureau data and one from the credit card balance data. From the bureau data the data we are interested in is if a loan had credit that was overdue, what was the highest number of days that was overdue. When we pull this in we found a difference in means between our target group, making it a possible predictor for our model.

From the credit card balance DF the variable of interest is latest credit card balance before application. When pulling this in we found a large amount of Na's (220,606 or 71.7%). The amount of NA's would make it a possibility for excluding, but with the observations reported we do see a difference in means between our target groups.

When modeling other interesting variable options could be:

- bureau, AMT_CREDIT_SUM, total credit amount (total credit)
- POS_CASH_balance, MONTHS_BALANCE, Month of balance relative to application (cash on hand)

- POS_CASH_balance, SK_DPD, days past due (overdue credit)
- credit_card_balance, AMT_CREDIT_LIMIT_ACTUAL, Credit card limit
- previous_application, CODE_REJECT_REASON, reason for previous application rejection (any previous rejections?)
- previous_application, AMT_CREDIT, Final credit amount from previous application

```
#Add Credit Day Overdue

##Check Structure and Summary
str(bureau)
```

```
## 'data.frame':    1716428 obs. of  17 variables:
##  $ SK_ID_CURR            : int  215354 215354 215354 215354 215354 215354 215354 162297 162297 16229
##  $ SK_ID_BUREAU          : int  5714462 5714463 5714464 5714465 5714466 5714467 5714468 5714469 5714
##  $ CREDIT_ACTIVE         : Factor w/ 4 levels "Active","Bad debt",..: 3 1 1 1 1 1 1 3 3 1 ...
##  $ CREDIT_CURRENCY       : Factor w/ 4 levels "currency 1","currency 2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ DAYS_CREDIT           : int  -497 -208 -203 -203 -629 -273 -43 -1896 -1146 -1146 ...
##  $ CREDIT_DAY_OVERDUE    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ DAYS_CREDIT_ENDDATE   : num  -153 1075 528 NA 1197 ...
##  $ DAYS_ENDDATE_FACT     : num  -153 NA NA NA NA NA NA -1710 -840 NA ...
##  $ AMT_CREDIT_MAX_OVERDUE: num  NA NA NA NA 77674 ...
##  $ CNT_CREDIT_PROLONG    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AMT_CREDIT_SUM        : num  91323 225000 464324 90000 2700000 ...
##  $ AMT_CREDIT_SUM_DEBT   : num  0 171342 NA NA NA ...
##  $ AMT_CREDIT_SUM_LIMIT  : num  NA NA NA NA NA ...
##  $ AMT_CREDIT_SUM_OVERDUE: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CREDIT_TYPE           : Factor w/ 15 levels "Another type of loan",..: 4 5 4 5 4 5 4 4 4 5 ...
##  $ DAYS_CREDIT_UPDATE    : int  -131 -20 -16 -16 -21 -31 -22 -1710 -840 -690 ...
##  $ AMT_ANNUITY           : num  NA NA NA NA NA NA NA NA NA NA ...
```

```
summary(bureau)
```

```
##    SK_ID_CURR      SK_ID_BUREAU      CREDIT_ACTIVE       CREDIT_CURRENCY
##  Min.   :100001   Min.   :5000000   Active  : 630607   currency 1:1715020
##  1st Qu.:188867   1st Qu.:5463954   Bad debt:     21   currency 2:   1224
##  Median :278055   Median :5926304   Closed  :1079273   currency 3:    174
##  Mean   :278215   Mean   :5924434   Sold    :   6527   currency 4:     10
##  3rd Qu.:367426   3rd Qu.:6385681
##  Max.   :456255   Max.   :6843457
##
##   DAYS_CREDIT    CREDIT_DAY_OVERDUE  DAYS_CREDIT_ENDDATE DAYS_ENDDATE_FACT
##  Min.   :-2922   Min.   :   0.0000   Min.   :-42060.0    Min.   :-42023
##  1st Qu.:-1666   1st Qu.:   0.0000   1st Qu.: -1138.0    1st Qu.: -1489
##  Median : -987   Median :   0.0000   Median :  -330.0    Median :  -897
##  Mean   :-1142   Mean   :   0.8182   Mean   :   510.5    Mean   : -1017
##  3rd Qu.: -474   3rd Qu.:   0.0000   3rd Qu.:   474.0    3rd Qu.:  -425
##  Max.   :    0   Max.   :2792.0000   Max.   : 31199.0    Max.   :     0
##                                      NA's   :105553      NA's   :633653
##  AMT_CREDIT_MAX_OVERDUE CNT_CREDIT_PROLONG AMT_CREDIT_SUM
##  Min.   :        0      Min.   :0.00000    Min.   :       0
##  1st Qu.:        0      1st Qu.:0.00000    1st Qu.:   51300
```

```
##  Median :        0      Median :0.00000    Median :   125518
##  Mean   :     3825      Mean   :0.00641    Mean   :   354995
##  3rd Qu.:        0      3rd Qu.:0.00000    3rd Qu.:   315000
##  Max.   :115987185      Max.   :9.00000    Max.   :585000000
##  NA's   :1124488                           NA's   :13
##  AMT_CREDIT_SUM_DEBT AMT_CREDIT_SUM_LIMIT AMT_CREDIT_SUM_OVERDUE
##  Min.   : -4705600   Min.   :-586406      Min.   :      0
##  1st Qu.:        0   1st Qu.:      0      1st Qu.:      0
##  Median :        0   Median :      0      Median :      0
##  Mean   :   137085   Mean   :   6230      Mean   :     38
##  3rd Qu.:    40154   3rd Qu.:      0      3rd Qu.:      0
##  Max.   :170100000   Max.   :4705600      Max.   :3756681
##  NA's   :257669      NA's   :591780
##                         CREDIT_TYPE      DAYS_CREDIT_UPDATE  AMT_ANNUITY
##  Consumer credit               :1251615   Min.   :-41947.0   Min.   :        0
##  Credit card                   : 402195   1st Qu.:  -908.0   1st Qu.:        0
##  Car loan                      :  27690   Median :  -395.0   Median :        0
##  Mortgage                      :  18391   Mean   :  -593.8   Mean   :    15713
##  Microloan                     :  12413   3rd Qu.:   -33.0   3rd Qu.:    13500
##  Loan for business development:   1975   Max.   :   372.0   Max.   :118453424
##  (Other)                       :   2149                      NA's   :1226791
```

```
summary(bureau$CREDIT_DAY_OVERDUE)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##    0.0000    0.0000    0.0000    0.8182    0.0000 2792.0000
```

```
##Create new DF with max line of Credit Days Overdue for each SK_ID
overdue_credit_max <- bureau %>%
    group_by(SK_ID_CURR) %>%
    slice(which.max(CREDIT_DAY_OVERDUE))

##Create new DF with Credit Days Overdue added
overdue <- clean_train %>%
    left_join(select(overdue_credit_max, CREDIT_DAY_OVERDUE), by="SK_ID_CURR")
```

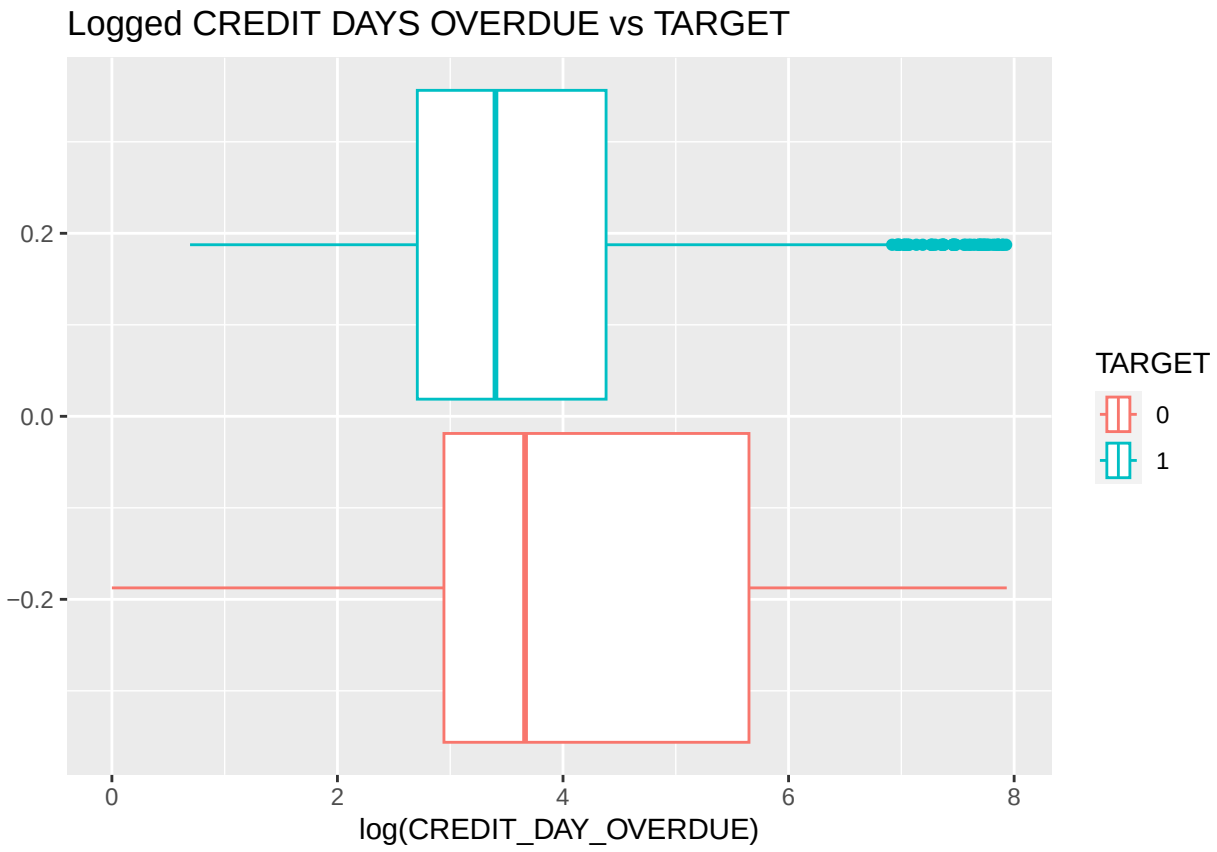
```
## Adding missing grouping variables: `SK_ID_CURR`
```

```
##CHECK Summary
summary(overdue$CREDIT_DAY_OVERDUE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.00    4.77    0.00 2792.00   44020
```

```
##Create Boxplot to check for mean variation
ggplot(data = overdue, aes(x=log(CREDIT_DAY_OVERDUE), color = TARGET)) + geom_boxplot() + labs(title =
```

```
## Warning: Removed 304114 rows containing non-finite values (`stat_boxplot()`).
```




```
#Add AMT Balance

##Check structure and summary of credit_card_balance
str(credit_card_balance)
```

```
## 'data.frame':    3840312 obs. of  23 variables:
##  $ SK_ID_PREV                : int  2562384 2582071 1740877 1389973 1891521 2646502 1079071 2095912 1
##  $ SK_ID_CURR                : int  378907 363914 371185 337855 126868 380010 171320 118650 367360 2
##  $ MONTHS_BALANCE            : int  -6 -1 -7 -4 -1 -7 -6 -7 -4 -5 ...
##  $ AMT_BALANCE               : num  57 63976 31815 236572 453919 ...
##  $ AMT_CREDIT_LIMIT_ACTUAL   : int  135000 45000 450000 225000 450000 270000 585000 45000 292500 225
##  $ AMT_DRAWINGS_ATM_CURRENT  : num  0 2250 0 2250 0 0 67500 45000 90000 76500 ...
##  $ AMT_DRAWINGS_CURRENT      : num  878 2250 0 2250 11547 ...
##  $ AMT_DRAWINGS_OTHER_CURRENT: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ AMT_DRAWINGS_POS_CURRENT  : num  878 0 0 0 11547 ...
##  $ AMT_INST_MIN_REGULARITY   : num  1700 2250 2250 11796 22925 ...
##  $ AMT_PAYMENT_CURRENT       : num  1800 2250 2250 11925 27000 ...
##  $ AMT_PAYMENT_TOTAL_CURRENT : num  1800 2250 2250 11925 27000 ...
##  $ AMT_RECEIVABLE_PRINCIPAL  : num  0 60175 26926 224949 443044 ...
##  $ AMT_RECIVABLE             : num  0 64876 31460 233049 453919 ...
##  $ AMT_TOTAL_RECEIVABLE      : num  0 64876 31460 233049 453919 ...
##  $ CNT_DRAWINGS_ATM_CURRENT  : num  0 1 0 1 0 0 1 1 3 3 ...
##  $ CNT_DRAWINGS_CURRENT      : int  1 1 0 1 1 0 1 1 8 9 ...
##  $ CNT_DRAWINGS_OTHER_CURRENT: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CNT_DRAWINGS_POS_CURRENT  : num  1 0 0 0 1 0 0 0 5 6 ...
##  $ CNT_INSTALMENT_MATURE_CUM : num  35 69 30 10 101 2 6 51 3 38 ...
```

```
##  $ NAME_CONTRACT_STATUS      : Factor w/ 7 levels "Active","Approved",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ SK_DPD                    : int  0 0 0 0 0 7 0 0 0 0 ...
##  $ SK_DPD_DEF                : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
summary(credit_card_balance)
```

```
##    SK_ID_PREV        SK_ID_CURR     MONTHS_BALANCE    AMT_BALANCE
##  Min.   :1000018   Min.   :100006   Min.   :-96.00   Min.   :-420250
##  1st Qu.:1434385   1st Qu.:189517   1st Qu.:-55.00   1st Qu.:      0
##  Median :1897122   Median :278396   Median :-28.00   Median :      0
##  Mean   :1904504   Mean   :278324   Mean   :-34.52   Mean   :  58300
##  3rd Qu.:2369328   3rd Qu.:367580   3rd Qu.:-11.00   3rd Qu.:  89047
##  Max.   :2843496   Max.   :456250   Max.   : -1.00   Max.   :1505902
##
##  AMT_CREDIT_LIMIT_ACTUAL AMT_DRAWINGS_ATM_CURRENT AMT_DRAWINGS_CURRENT
##  Min.   :      0         Min.   :  -6827          Min.   :  -6212
##  1st Qu.:  45000         1st Qu.:      0          1st Qu.:      0
##  Median : 112500         Median :      0          Median :      0
##  Mean   : 153808         Mean   :   5961          Mean   :   7433
##  3rd Qu.: 180000         3rd Qu.:      0          3rd Qu.:      0
##  Max.   :1350000         Max.   :2115000          Max.   :2287098
##                          NA's   :749816
##  AMT_DRAWINGS_OTHER_CURRENT AMT_DRAWINGS_POS_CURRENT AMT_INST_MIN_REGULARITY
##  Min.   :      0.0          Min.   :      0          Min.   :     0
##  1st Qu.:      0.0          1st Qu.:      0          1st Qu.:     0
##  Median :      0.0          Median :      0          Median :     0
##  Mean   :    288.2          Mean   :   2969          Mean   :  3540
##  3rd Qu.:      0.0          3rd Qu.:      0          3rd Qu.:  6634
##  Max.   :1529847.0          Max.   :2239274          Max.   :202882
##  NA's   :749816             NA's   :749816           NA's   :305236
##  AMT_PAYMENT_CURRENT AMT_PAYMENT_TOTAL_CURRENT AMT_RECEIVABLE_PRINCIPAL
##  Min.   :      0     Min.   :      0           Min.   :-423306
##  1st Qu.:    152     1st Qu.:      0           1st Qu.:      0
##  Median :   2703     Median :      0           Median :      0
##  Mean   :  10281     Mean   :   7589           Mean   :  55966
##  3rd Qu.:   9000     3rd Qu.:   6750           3rd Qu.:  85359
##  Max.   :4289207     Max.   :4278316           Max.   :1472317
##  NA's   :767988
##  AMT_RECIVABLE     AMT_TOTAL_RECEIVABLE CNT_DRAWINGS_ATM_CURRENT
##  Min.   :-420250   Min.   :-420250      Min.   : 0.0
##  1st Qu.:      0   1st Qu.:      0      1st Qu.: 0.0
##  Median :      0   Median :      0      Median : 0.0
##  Mean   :  58089   Mean   :  58098      Mean   : 0.3
##  3rd Qu.:  88900   3rd Qu.:  88914      3rd Qu.: 0.0
##  Max.   :1493338   Max.   :1493338      Max.   :51.0
##                                         NA's   :749816
##  CNT_DRAWINGS_CURRENT CNT_DRAWINGS_OTHER_CURRENT CNT_DRAWINGS_POS_CURRENT
##  Min.   :  0.0000     Min.   : 0                 Min.   :  0.0
##  1st Qu.:  0.0000     1st Qu.: 0                 1st Qu.:  0.0
##  Median :  0.0000     Median : 0                 Median :  0.0
##  Mean   :  0.7031     Mean   : 0                 Mean   :  0.6
##  3rd Qu.:  0.0000     3rd Qu.: 0                 3rd Qu.:  0.0
##  Max.   :165.0000     Max.   :12                 Max.   :165.0
##                       NA's   :749816             NA's   :749816
```

```
##  CNT_INSTALMENT_MATURE_CUM    NAME_CONTRACT_STATUS     SK_DPD
##  Min.   :  0.00            Active       :3698436   Min.   :   0.000
##  1st Qu.:  4.00            Approved     :      5   1st Qu.:   0.000
##  Median : 15.00            Completed    : 128918   Median :   0.000
##  Mean   : 20.83            Demand       :   1365   Mean   :   9.284
##  3rd Qu.: 32.00            Refused      :     17   3rd Qu.:   0.000
##  Max.   :120.00            Sent proposal:    513   Max.   :3260.000
##  NA's   :305236            Signed       :  11058
##    SK_DPD_DEF
##  Min.   :   0.000
##  1st Qu.:   0.000
##  Median :   0.000
##  Mean   :   0.332
##  3rd Qu.:   0.000
##  Max.   :3260.000
##
```

```
##Create new DF with one line per CC balance for each SK_ID
newccbalance <- credit_card_balance %>%
    group_by(SK_ID_CURR) %>%
    slice(which.max(MONTHS_BALANCE))

##Create new DF joining new data
ccbalance <- clean_train %>%
    left_join(select(newccbalance, AMT_BALANCE), by="SK_ID_CURR")
```

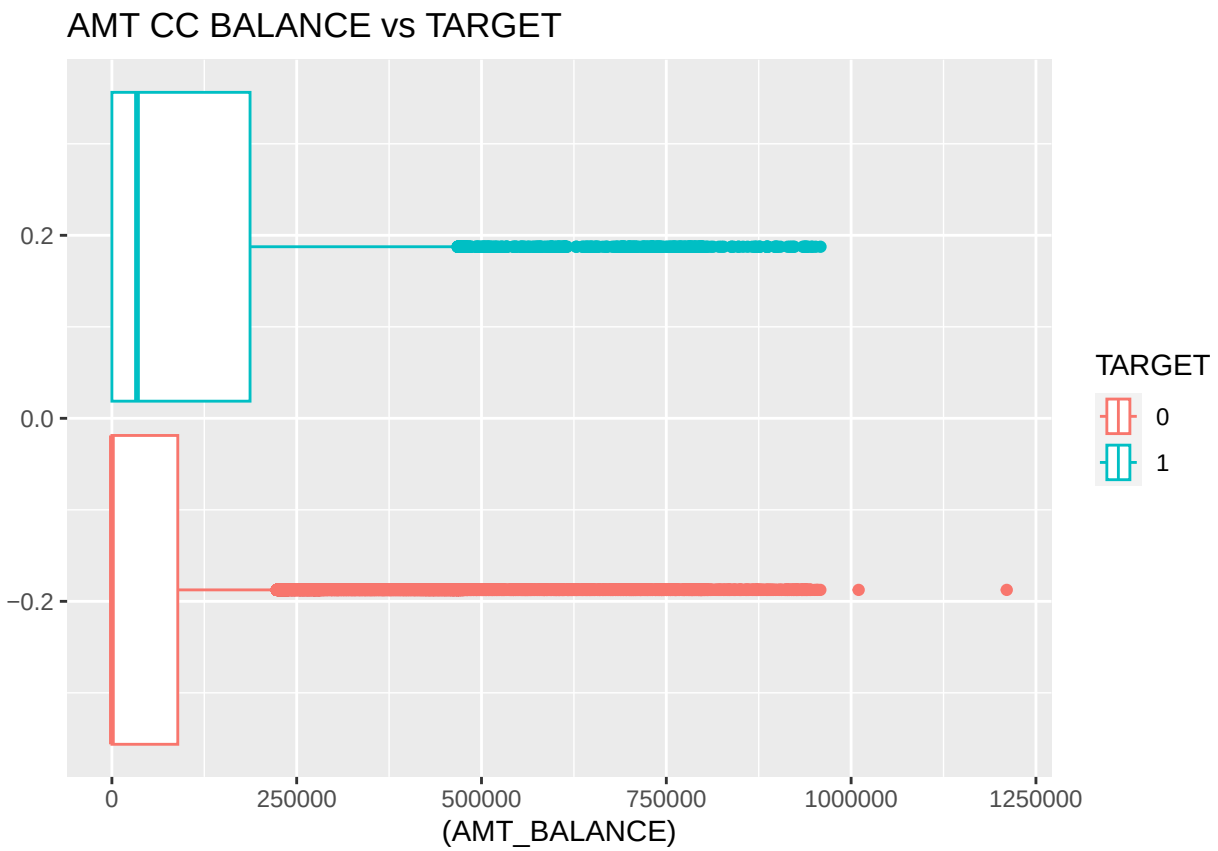
```
## Adding missing grouping variables: `SK_ID_CURR`
```

```
##Check summary of added data
summary(ccbalance$AMT_BALANCE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0       0       0   76634  105528 1210511  220606
```

```
##Plot Data
ggplot(data = ccbalance, aes(x=(AMT_BALANCE), color = TARGET)) + geom_boxplot() + labs(title = "AMT CC
```

```
## Warning: Removed 220606 rows containing non-finite values (`stat_boxplot()`).
```



Predictors

After removing the large NAs, blanks, and low variance there were 46 remaining variables that needed testing for potential prediction power. For these we placed each of the variables into 1 of 3 groups Strong difference between groups, some difference between groups, no difference between groups. To classify these we compared a % of default vs non default for each group in a categorical data, and mean values between continuous variables.

The results gave us 10 Strong Differences, 16 Some Differences, 20 No Differences. Our 10 Strong Differences variables are:

1. NAME_INCOME_TYPE
2. NAME_HOUSING_TYPE
3. DAYS_BIRTH
4. DAYS_EMPLOYED
5. REGION_RATING_CLIENT
6. REGION_RATING_CLIENT_W_CITY
7. REG_CITY_NOT_LIVE_CITY
8. EXT_SOURCE_1
9. EXT_SOURCE_2
10. EXT_SOURCE_3

These would be our starting point for variable selection in our models.

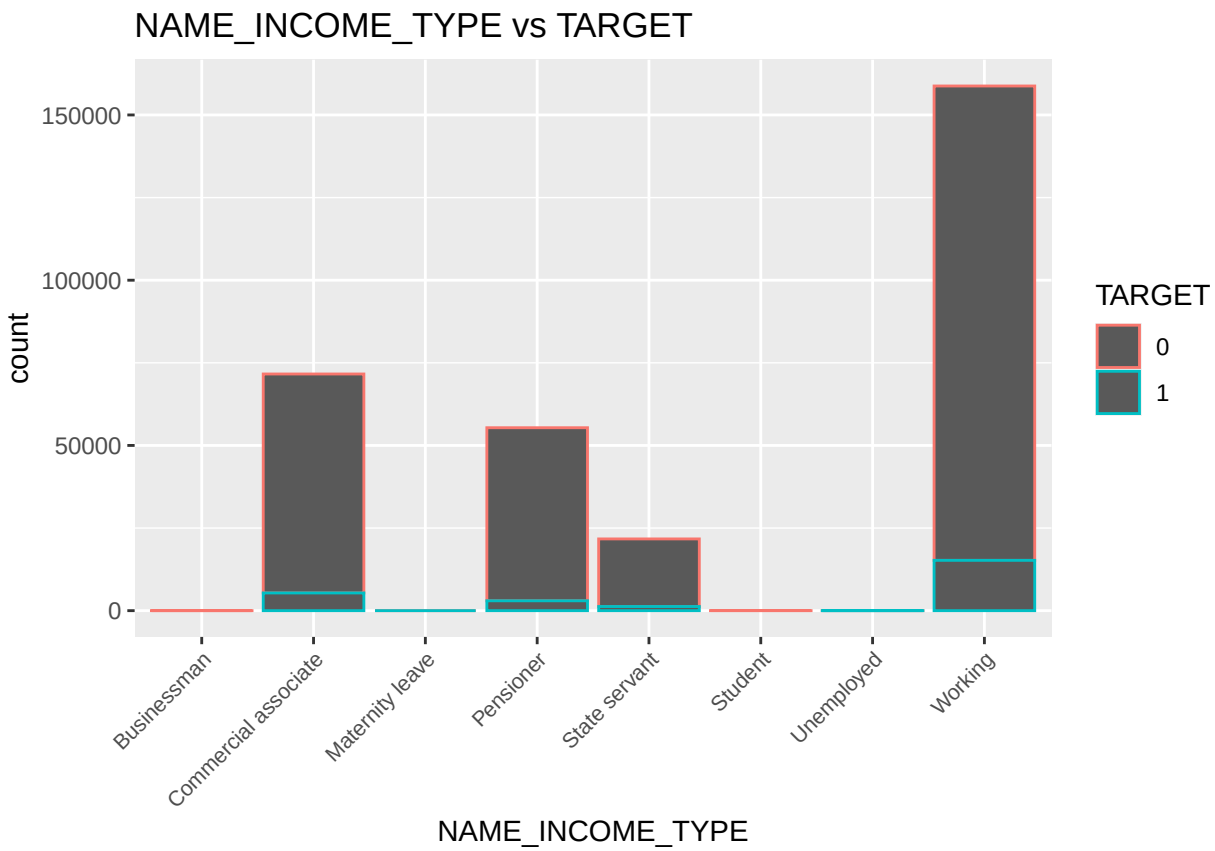
Strong Predictors

```
#NAME_INCOME_TYPE Large difference between gruops, potential strong predictor
clean_train %>%
  group_by(NAME_INCOME_TYPE,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'NAME_INCOME_TYPE'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 14 x 4
## # Groups:   NAME_INCOME_TYPE [8]
##    NAME_INCOME_TYPE     TARGET      n   freq
##    <fct>                <fct>   <int>  <dbl>
##  1 Businessman          0          10 100
##  2 Commercial associate 0       66257  92.5
##  3 Commercial associate 1        5360   7.48
##  4 Maternity leave      0           3  60
##  5 Maternity leave      1           2  40
##  6 Pensioner            0       52380  94.6
##  7 Pensioner            1        2982   5.39
##  8 State servant        0       20454  94.2
##  9 State servant        1        1249   5.75
## 10 Student              0          18 100
## 11 Unemployed           0          14  63.6
## 12 Unemployed           1           8  36.4
## 13 Working              0      143550  90.4
## 14 Working              1       15224   9.59
```

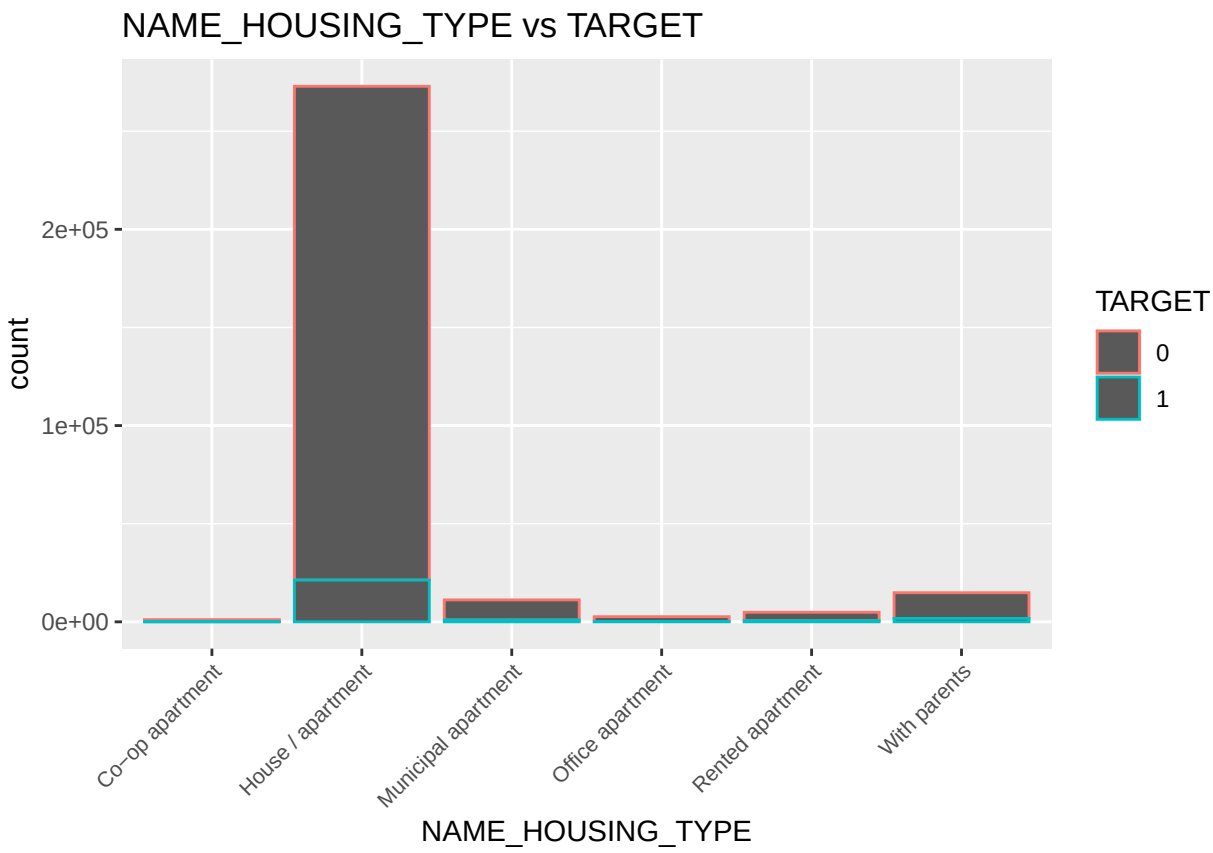
```
ggplot(data = clean_train, aes(x=NAME_INCOME_TYPE, color = TARGET)) + geom_bar() + labs(title = "NAME_I
```



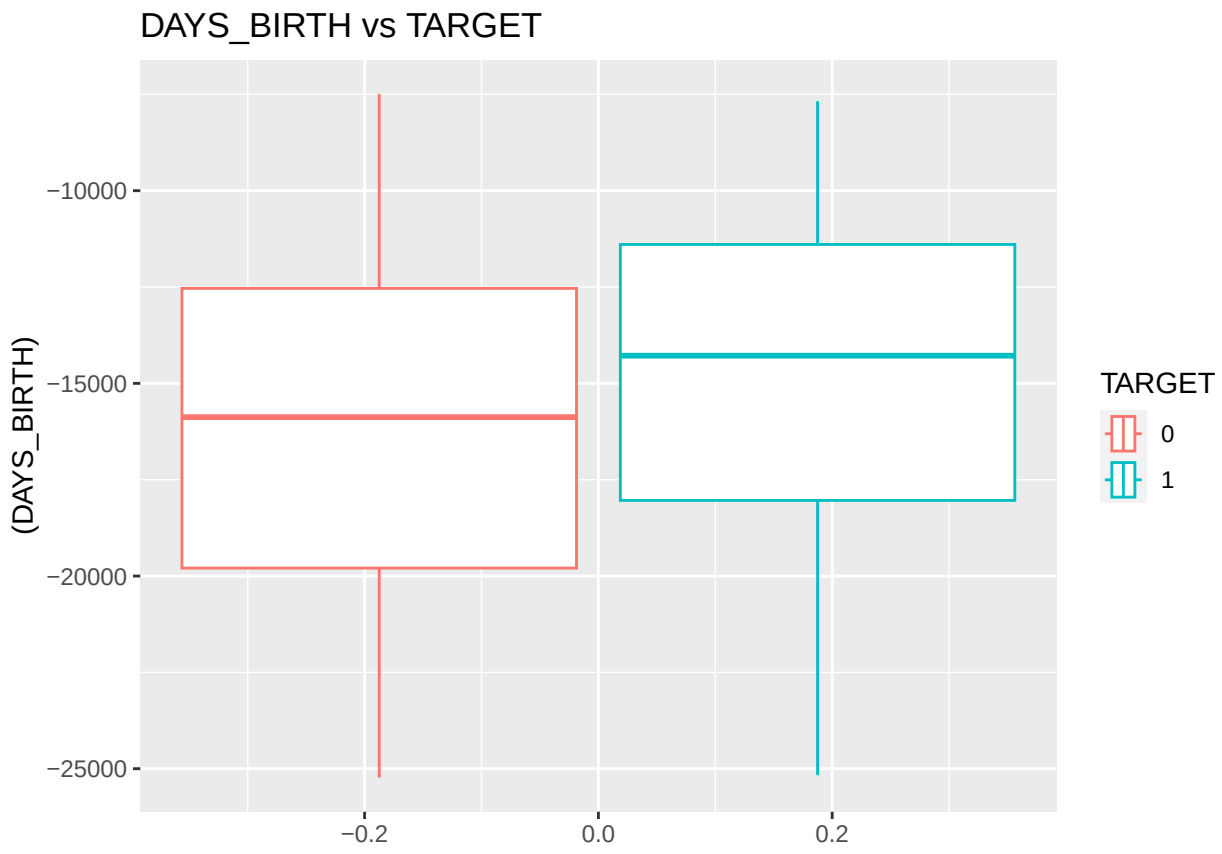

```
#NAME_HOUSING_TYPE LARGE difference between groups, potential Strong predictor
clean_train %>%
  group_by(NAME_HOUSING_TYPE,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'NAME_HOUSING_TYPE'. You can override using
## the `.groups` argument.

## # A tibble: 12 x 4
## # Groups:   NAME_HOUSING_TYPE [6]
##    NAME_HOUSING_TYPE   TARGET      n  freq
##    <fct>               <fct>   <int> <dbl>
##  1 Co-op apartment     0        1033 92.1
##  2 Co-op apartment     1          89  7.93
##  3 House / apartment   0      251596 92.2
##  4 House / apartment   1       21272  7.80
##  5 Municipal apartment 0       10228 91.5
##  6 Municipal apartment 1         955  8.54
##  7 Office apartment    0        2445 93.4
##  8 Office apartment    1         172  6.57
##  9 Rented apartment    0        4280 87.7
## 10 Rented apartment    1         601 12.3
## 11 With parents        0       13104 88.3
## 12 With parents        1        1736 11.7
```

```
ggplot(data = clean_train, aes(x=NAME_HOUSING_TYPE, color = TARGET)) + geom_bar() + labs(title = "NAME_
```



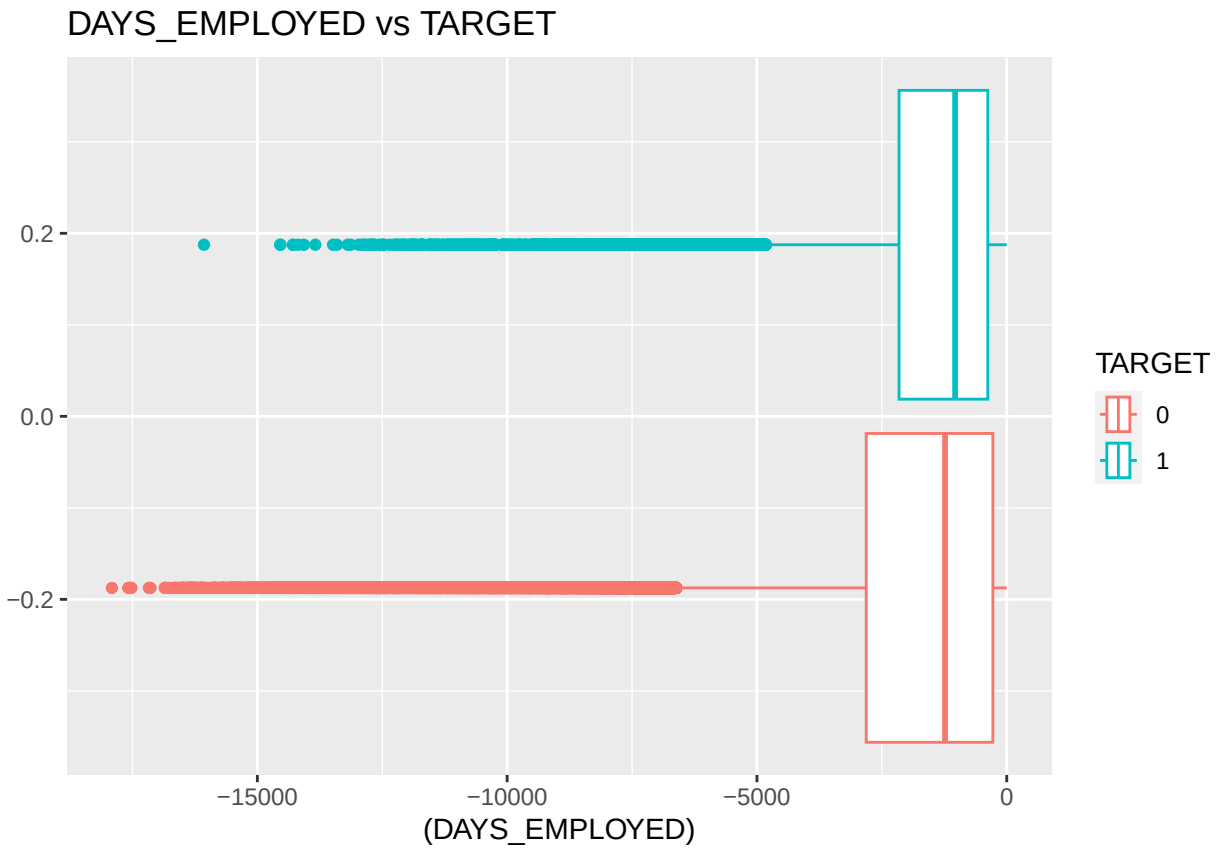

```
#DAYS_BIRTH LARGE difference in Means strong possibility of prediction
ggplot(data = clean_train, aes(x=(DAYS_BIRTH), color = TARGET)) + geom_boxplot() + labs(title = "DAYS_B
```




```
clean_train %>%
  group_by(TARGET) %>%
    summarise(mean = mean((DAYS_BIRTH)))
```

```
## # A tibble: 2 x 2
##   TARGET    mean
##   <fct>    <dbl>
## 1 0      -16138.
## 2 1      -14885.
```

```
#DAYS_EMPLOYED LARGE difference in Means strong possibility of prediction, after transformation of data
ggplot(data = DEtest, aes(x=(DAYS_EMPLOYED), color = TARGET)) + geom_boxplot() + labs(title = "DAYS_EMPL
```




```
clean_train %>%
  group_by(TARGET) %>%
     mutate(DAYS_EMPLOYED = replace(DAYS_EMPLOYED, DAYS_EMPLOYED == 365243, 0)) %>%
    summarise(mean = mean((DAYS_EMPLOYED)))
```

```
## # A tibble: 2 x 2
##   TARGET   mean
##   <fct>   <dbl>
## 1 0      -1986.
## 2 1      -1596.
```

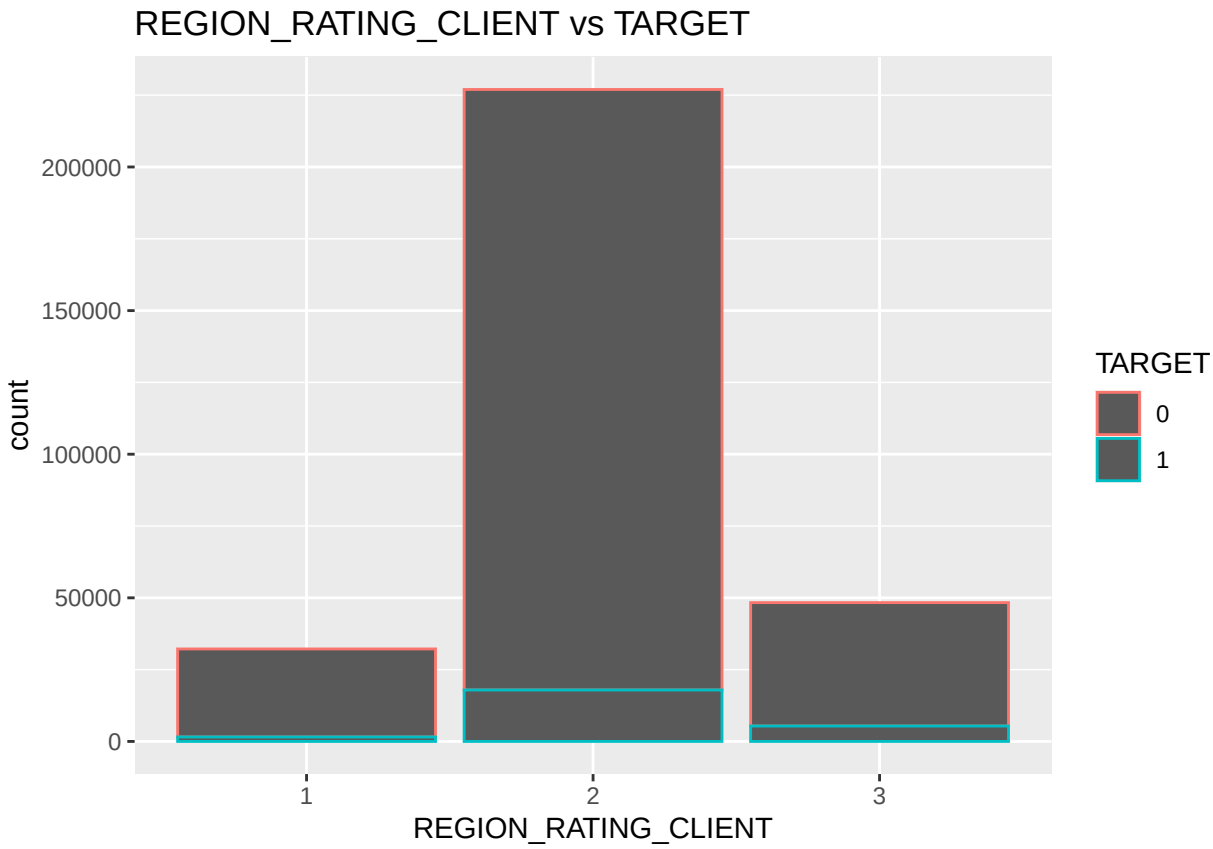
```
#REGION_RATING_CLIENT Large differences between groups Strong possibility for prediction
clean_train %>%
  group_by(REGION_RATING_CLIENT,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'REGION_RATING_CLIENT'. You can override
## using the `.groups` argument.
```

```
## # A tibble: 6 x 4
## # Groups:   REGION_RATING_CLIENT [3]
##   REGION_RATING_CLIENT TARGET      n  freq
```

```
##   <fct>                <fct>   <int> <dbl>
## 1 1                    0       30645 95.2
## 2 1                    1        1552  4.82
## 3 2                    0      209077 92.1
## 4 2                    1       17907  7.89
## 5 3                    0       42964 88.9
## 6 3                    1        5366 11.1
```

```
ggplot(data = clean_train, aes(x=REGION_RATING_CLIENT, color = TARGET)) + geom_bar() + labs(title = "RE
```



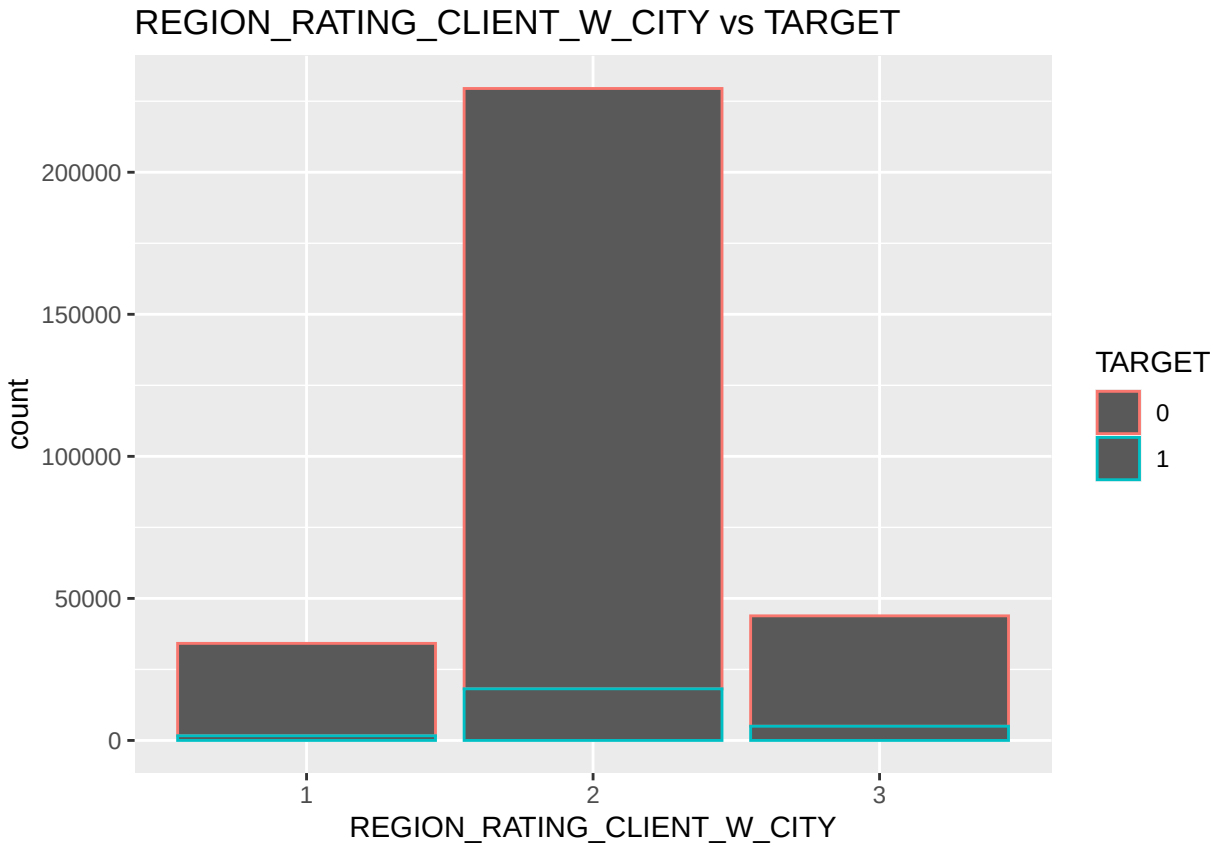

```
#REGION_RATING_CLIENT_W_CITY Large differences between groups Strong possibility for prediction
clean_train %>%
  group_by(REGION_RATING_CLIENT_W_CITY,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'REGION_RATING_CLIENT_W_CITY'. You can
## override using the `.groups` argument.
```

```
## # A tibble: 6 x 4
## # Groups:   REGION_RATING_CLIENT_W_CITY [3]
##   REGION_RATING_CLIENT_W_CITY TARGET      n  freq
##   <fct>                       <fct>   <int> <dbl>
## 1 1                           0       32513 95.2
```

```
## 2 1                          1        1654  4.84
## 3 2                          0      211314 92.1
## 4 2                          1       18170  7.92
## 5 3                          0       38859 88.6
## 6 3                          1        5001 11.4
```

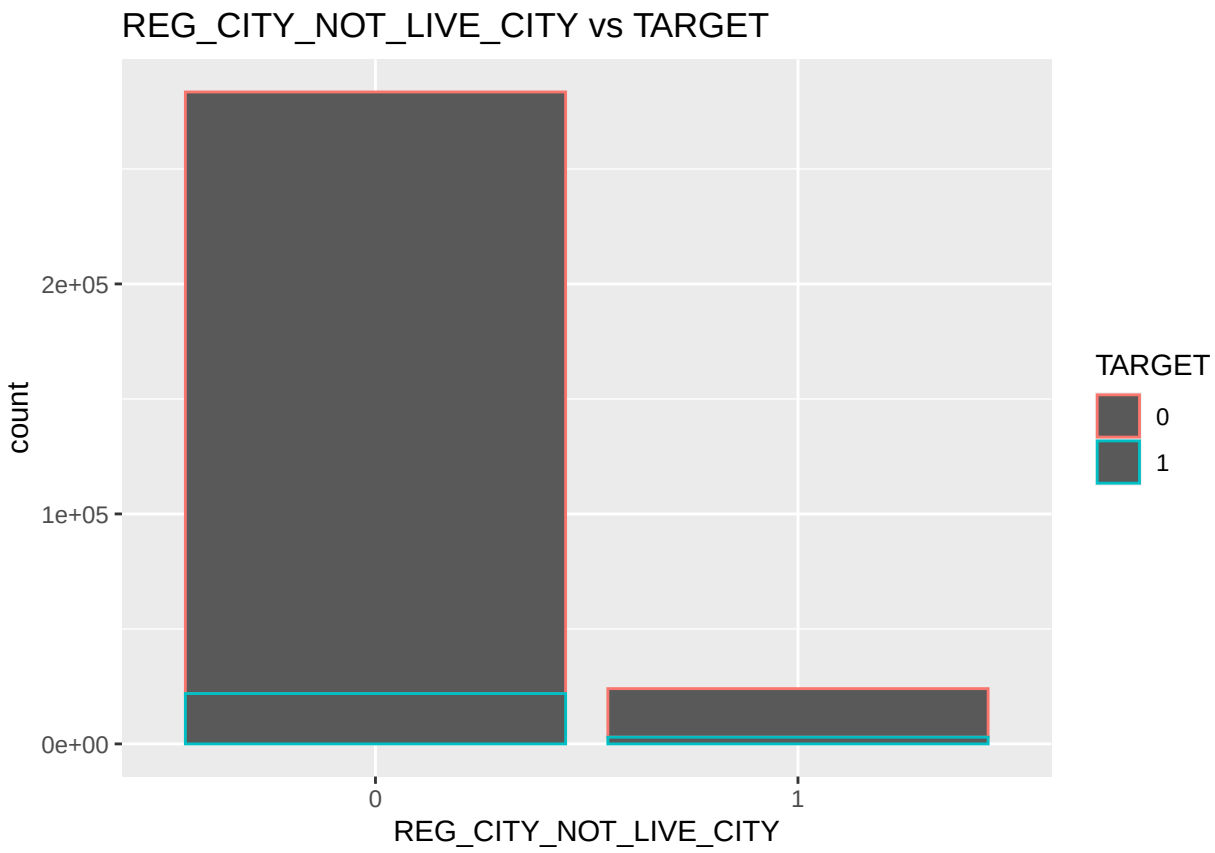
```
ggplot(data = clean_train, aes(x=REGION_RATING_CLIENT_W_CITY, color = TARGET)) + geom_bar() + labs(titl
```




```
#REG_CITY_NOT_LIVE_CITY large difference for those not living in city 12.2% default Strong possible for
clean_train %>%
  group_by(REG_CITY_NOT_LIVE_CITY,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

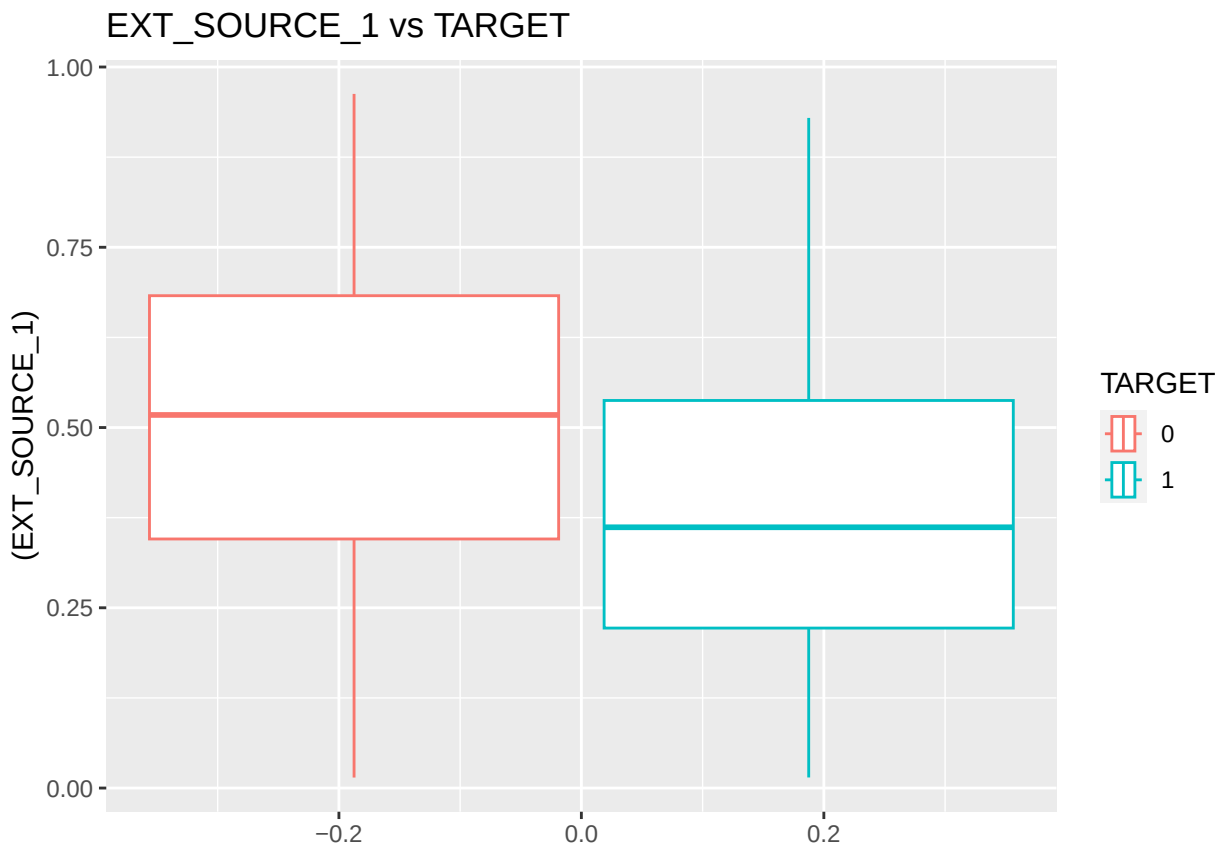
```
## `summarise()` has grouped output by 'REG_CITY_NOT_LIVE_CITY'. You can override
## using the `.groups` argument.

## # A tibble: 4 x 4
## # Groups:   REG_CITY_NOT_LIVE_CITY [2]
##   REG_CITY_NOT_LIVE_CITY TARGET      n  freq
##   <fct>                  <fct>   <int> <dbl>
## 1 0                      0      261586 92.3
## 2 0                      1       21886  7.72
## 3 1                      0       21100 87.8
## 4 1                      1        2939 12.2
```

```
ggplot(data = clean_train, aes(x=REG_CITY_NOT_LIVE_CITY, color = TARGET)) + geom_bar() + labs(title = "
```




```
#EXT_SOURCE_1 Strong difference in means after replacing NA with mean
ggplot(data = clean_train, aes(x=(EXT_SOURCE_1), color = TARGET)) + geom_boxplot() + labs(title = "EXT_S
```

```
## Warning: Removed 173378 rows containing non-finite values (`stat_boxplot()`).
```

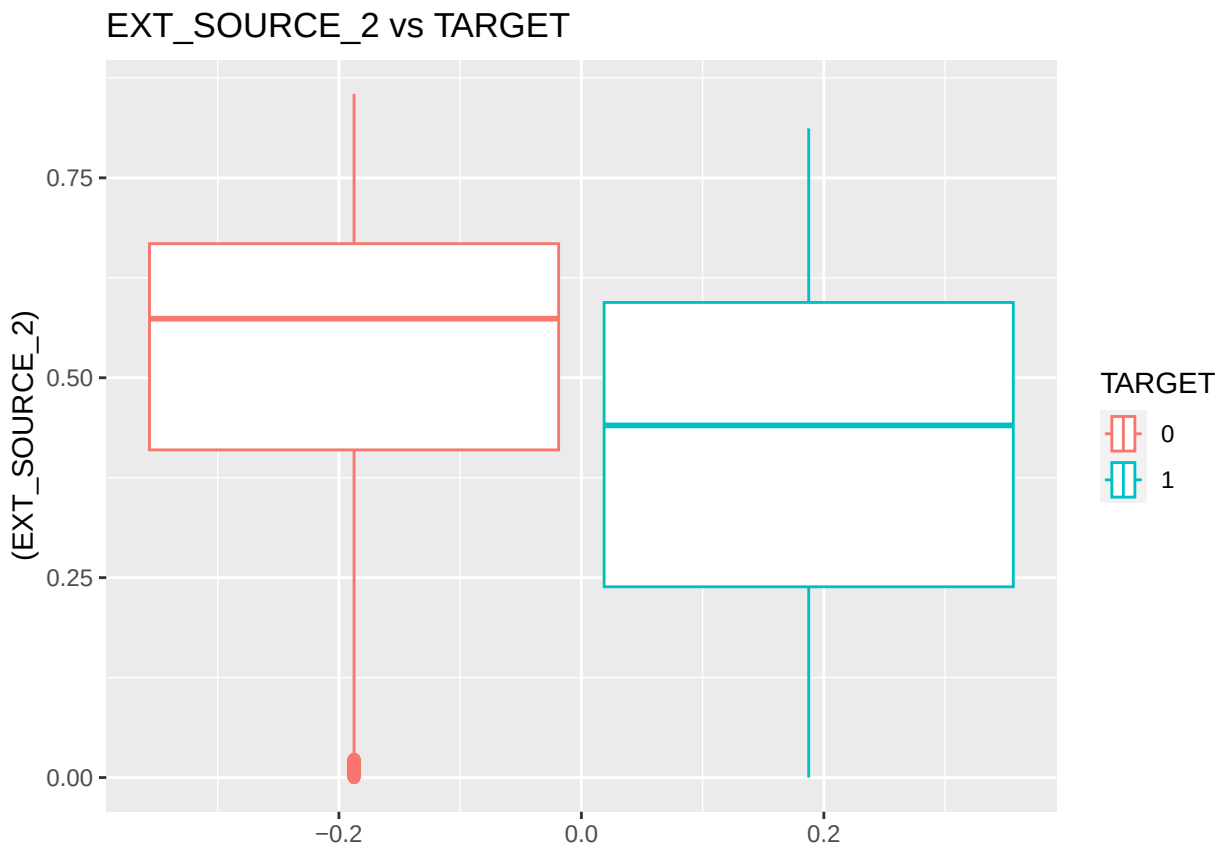



```
clean_train %>%
  group_by(TARGET) %>%
    mutate(across(EXT_SOURCE_1, ~replace_na(., mean(., na.rm=TRUE)))) %>%
    summarise(mean = mean((EXT_SOURCE_1)))
```

```
## # A tibble: 2 x 2
##   TARGET  mean
##   <fct>  <dbl>
## 1 0      0.511
## 2 1      0.387
```

```
#EXT_SOURCE_2 Strong difference in means after replacing NA with mean
ggplot(data = clean_train, aes(x=(EXT_SOURCE_2), color = TARGET)) + geom_boxplot() + labs(title = "EXT_S
```

```
## Warning: Removed 660 rows containing non-finite values (‘stat_boxplot()‘).
```

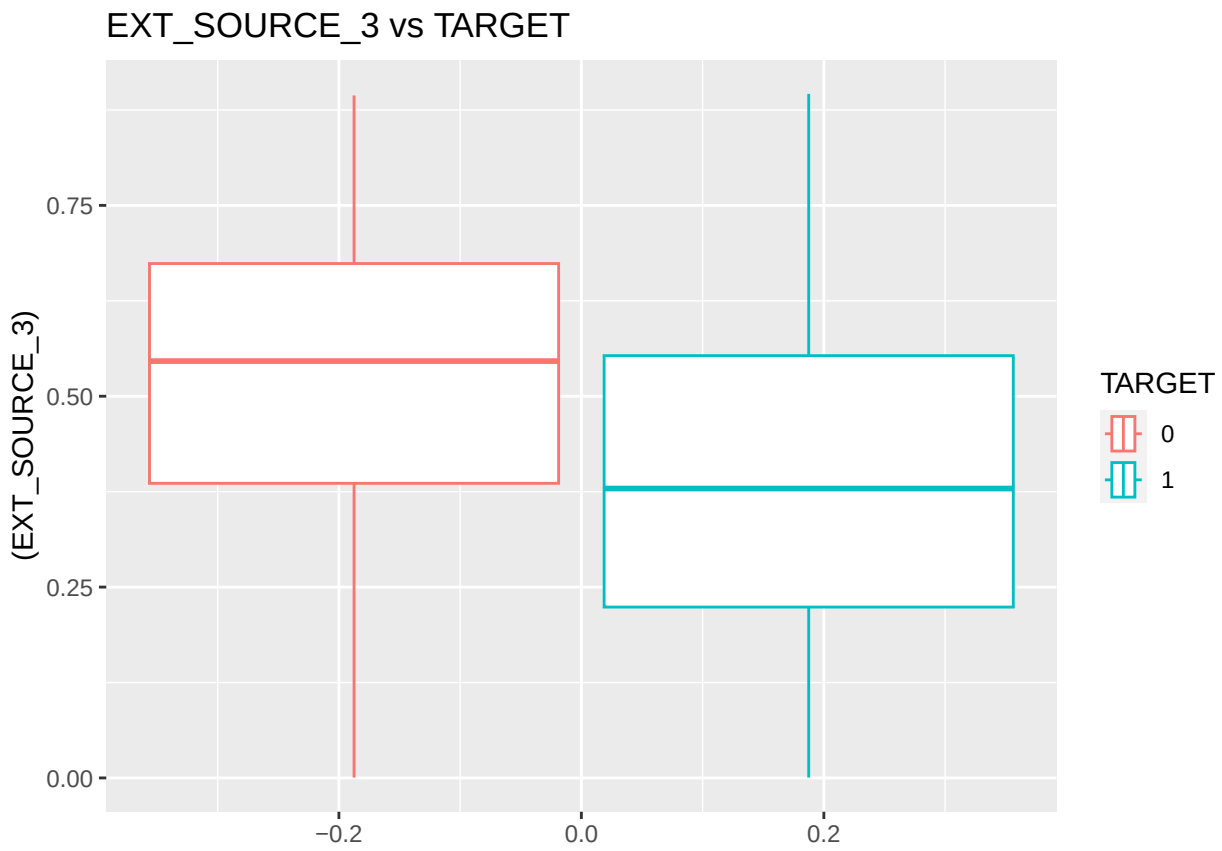



```
clean_train %>%
  group_by(TARGET) %>%
    mutate(across(EXT_SOURCE_2, ~replace_na(., mean(., na.rm=TRUE)))) %>%
    summarise(mean = mean((EXT_SOURCE_2)))
```

```
## # A tibble: 2 x 2
##   TARGET  mean
##   <fct>  <dbl>
## 1 0      0.523
## 2 1      0.411
```

```
#EXT_SOURCE_3 Strong difference in means after replacing NA with mean
ggplot(data = clean_train, aes(x=(EXT_SOURCE_3), color = TARGET)) + geom_boxplot() + labs(title = "EXT_S
```

```
## Warning: Removed 60965 rows containing non-finite values (‘stat_boxplot()‘).
```




```
clean_train %>%
  group_by(TARGET) %>%
    mutate(across(EXT_SOURCE_3, ~replace_na(., mean(., na.rm=TRUE)))) %>%
    summarise(mean = mean((EXT_SOURCE_3)))
```

```
## # A tibble: 2 x 2
##   TARGET  mean
##   <fct>  <dbl>
## 1 0      0.521
## 2 1      0.391
```

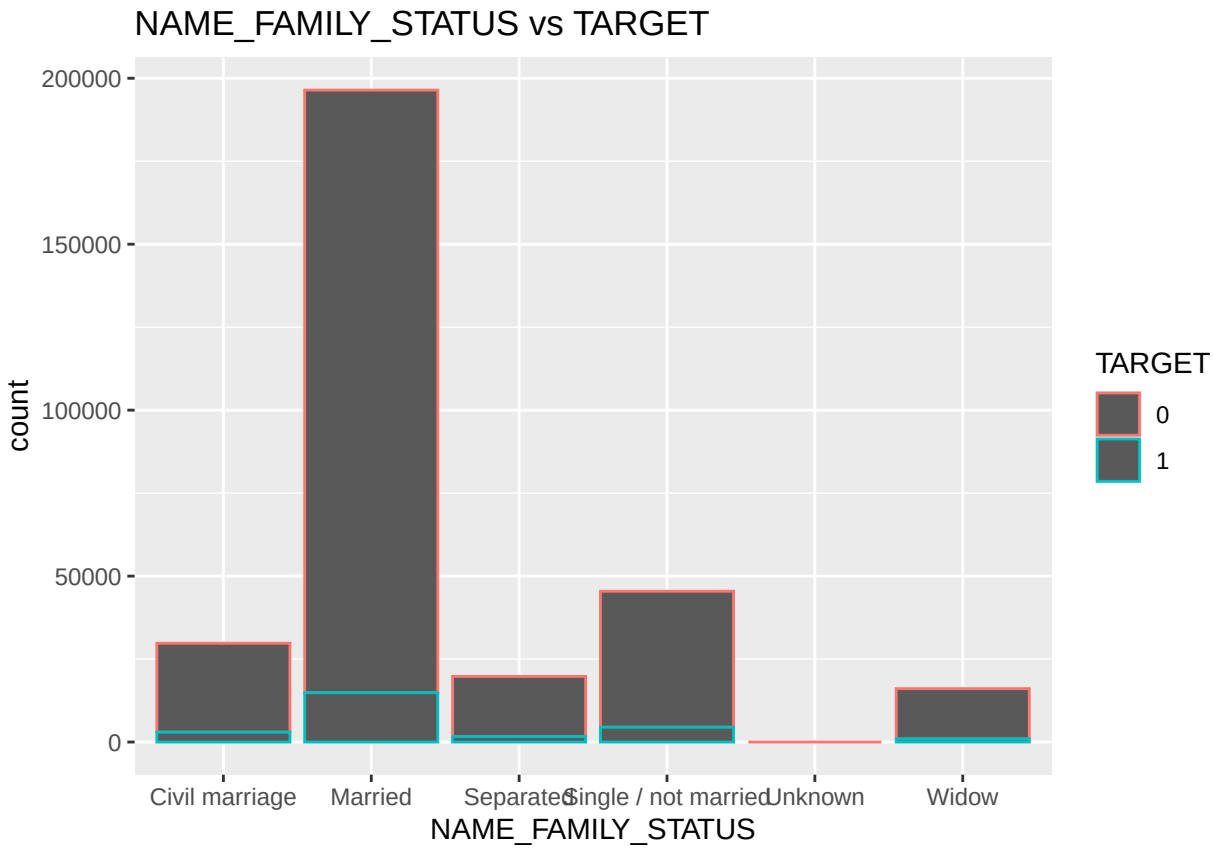
```
#NAME_FAMILY_STATUS difference between groups, potential predictor
clean_train %>%
  group_by(NAME_FAMILY_STATUS,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'NAME_FAMILY_STATUS'. You can override
## using the `.groups` argument.
```

```
## # A tibble: 11 x 4
## # Groups:   NAME_FAMILY_STATUS [6]
##    NAME_FAMILY_STATUS   TARGET      n   freq
```

```
##    <fct>                <fct>   <int>  <dbl>
##  1 Civil marriage       0       26814  90.1
##  2 Civil marriage       1        2961   9.94
##  3 Married              0      181582  92.4
##  4 Married              1       14850   7.56
##  5 Separated            0       18150  91.8
##  6 Separated            1        1620   8.19
##  7 Single / not married 0       40987  90.2
##  8 Single / not married 1        4457   9.81
##  9 Unknown              0           2 100
## 10 Widow                0       15151  94.2
## 11 Widow                1         937   5.82
```

```
ggplot(data = clean_train, aes(x=NAME_FAMILY_STATUS, color = TARGET)) + geom_bar() + labs(title = "NAME_
```



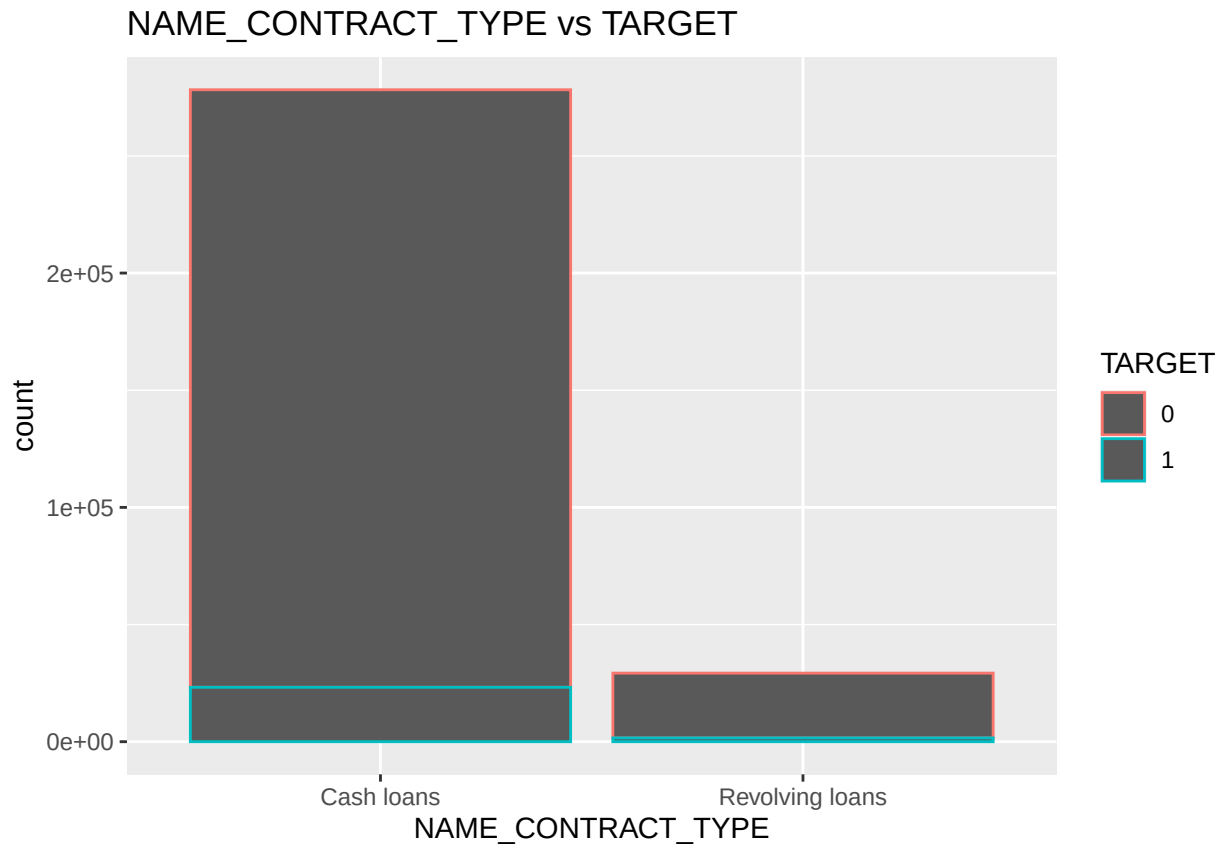
Some Difference

```
#NAME_CONTRACT_TYPE 3% difference between default on cash loans
clean_train %>%
  group_by(NAME_CONTRACT_TYPE,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'NAME_CONTRACT_TYPE'. You can override
## using the `.groups` argument.
```

```
## # A tibble: 4 x 4
## # Groups:   NAME_CONTRACT_TYPE [2]
##   NAME_CONTRACT_TYPE TARGET      n  freq
##   <fct>              <fct>   <int> <dbl>
## 1 Cash loans         0      255011 91.7
## 2 Cash loans         1       23221  8.35
## 3 Revolving loans    0       27675 94.5
## 4 Revolving loans    1        1604  5.48
```

```
ggplot(data = clean_train, aes(x=NAME_CONTRACT_TYPE, color = TARGET)) + geom_bar() + labs(title = "NAME_
```



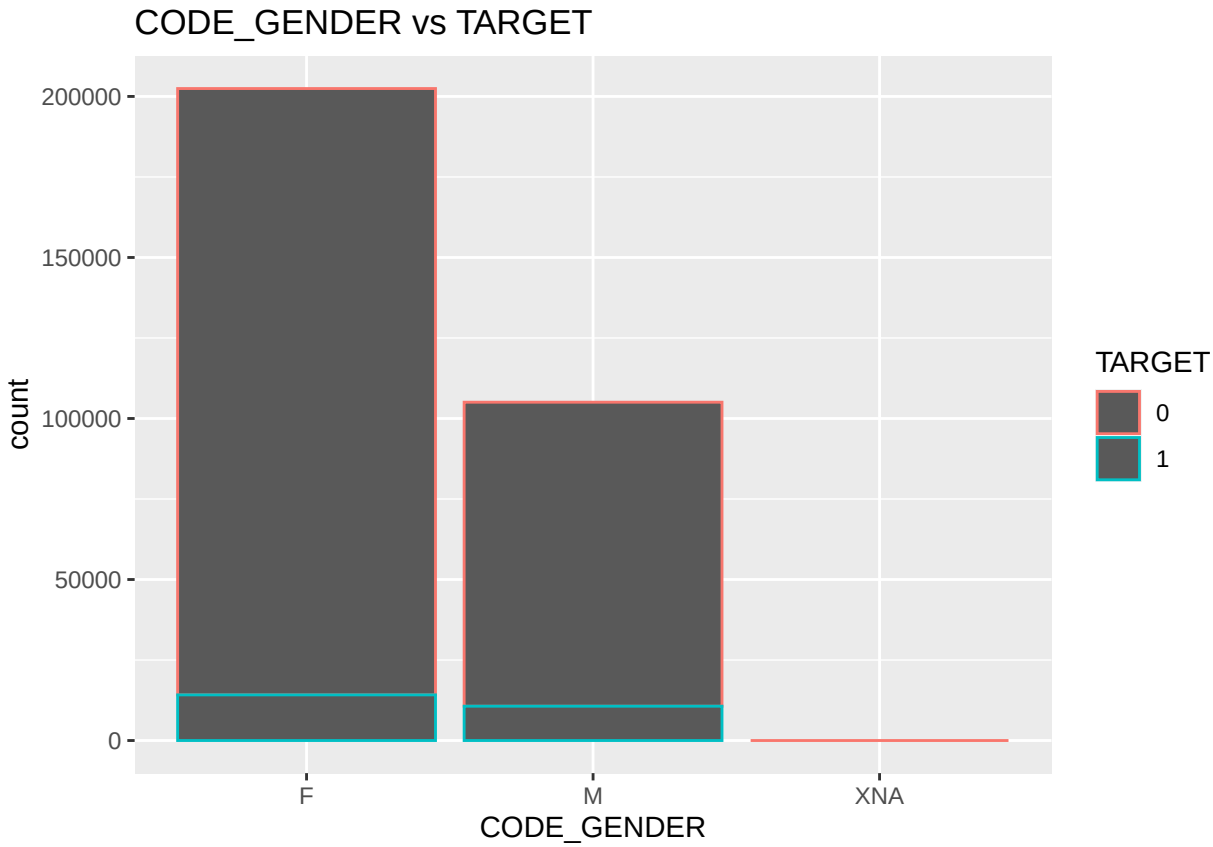

```
#CODE_GENDER 2% difference between M & F
clean_train %>%
  group_by(CODE_GENDER,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100))
```

```
## `summarise()` has grouped output by 'CODE_GENDER'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 5 x 4
```

```
## # Groups:   CODE_GENDER [3]
##   CODE_GENDER TARGET      n   freq
##   <fct>       <fct>   <int>  <dbl>
## 1 F           0      188278  93.0
## 2 F           1       14170   7.00
## 3 M           0       94404  89.9
## 4 M           1       10655  10.1
## 5 XNA         0           4 100
```

```
ggplot(data = clean_train, aes(x=CODE_GENDER, color = TARGET)) + geom_bar() + labs(title = "CODE_GENDER
```



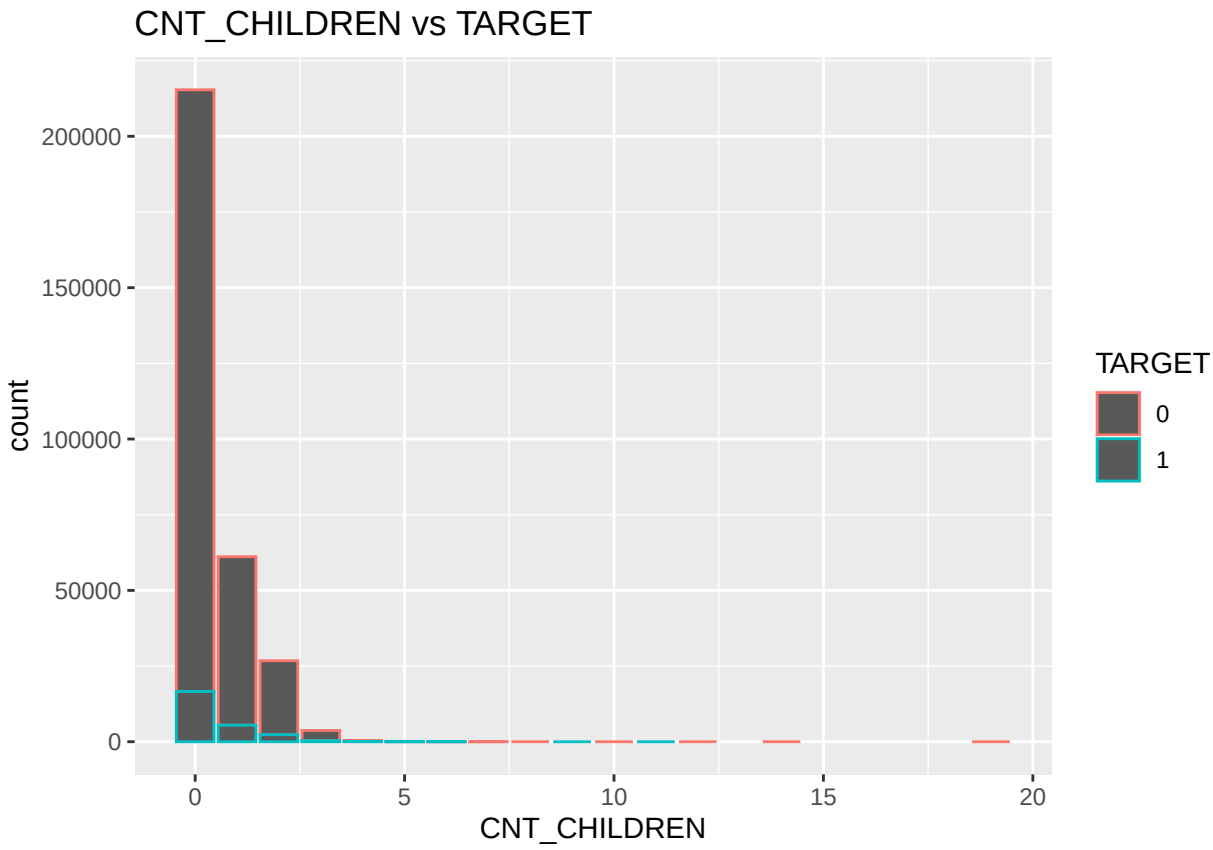

```
#CNT_CHILDREN Higher default as more children max at 28% for 6 children
clean_train %>%
  group_by(CNT_CHILDREN,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'CNT_CHILDREN'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 22 x 4
## # Groups:   CNT_CHILDREN [15]
##    CNT_CHILDREN TARGET      n   freq
```

```
##           <int> <fct>   <int>  <dbl>
##  1            0 0      198762  92.3
##  2            0 1       16609   7.71
##  3            1 0       55665  91.1
##  4            1 1        5454   8.92
##  5            2 0       24416  91.3
##  6            2 1        2333   8.72
##  7            3 0        3359  90.4
##  8            3 1         358   9.63
##  9            4 0         374  87.2
## 10            4 1          55  12.8
## 11            5 0          77  91.7
## 12            5 1           7   8.33
## 13            6 0          15  71.4
## 14            6 1           6  28.6
## 15            7 0           7 100
## 16            8 0           2 100
## 17            9 1           2 100
## 18           10 0           2 100
## 19           11 1           1 100
## 20           12 0           2 100
## 21           14 0           3 100
## 22           19 0           2 100
```

```
ggplot(data = clean_train, aes(x=CNT_CHILDREN, color = TARGET)) + geom_bar() + labs(title = "CNT_CHILDR
```



```
#AMT_GOODS_PRICE Possible significant difference in means. Needed to replace 278 NA with Avg of Group
ggplot(data = clean_train, aes(x=(AMT_GOODS_PRICE), color = TARGET)) + geom_boxplot() + labs(title = "Al
```

```
## Warning: Removed 278 rows containing non-finite values (‘stat_boxplot()‘).
```


```
clean_train %>%
  group_by(TARGET) %>%
    mutate(across(AMT_GOODS_PRICE, ~replace_na(., mean(., na.rm=TRUE)))) %>%
    summarise(mean = mean((AMT_GOODS_PRICE)))
```

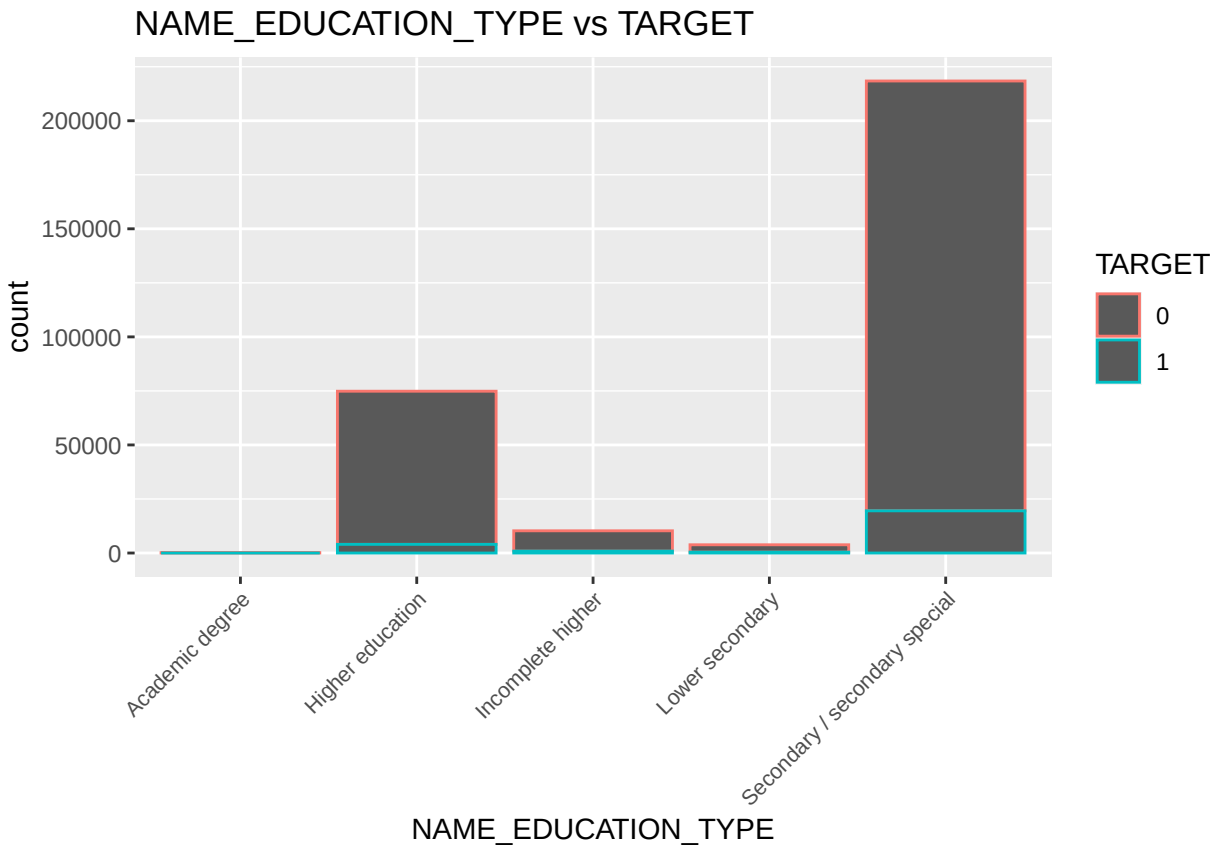
```
## # A tibble: 2 x 2
##   TARGET    mean
##   <fct>    <dbl>
## 1 0      542737.
## 2 1      488972.
```

```
#NAME_EDUCATION_TYPE small difference between groups, potential predictor
clean_train %>%
  group_by(NAME_EDUCATION_TYPE,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## ‘summarise()‘ has grouped output by ’NAME_EDUCATION_TYPE’. You can override
## using the ‘.groups‘ argument.
```

```
## # A tibble: 10 x 4
## # Groups:   NAME_EDUCATION_TYPE [5]
##    NAME_EDUCATION_TYPE           TARGET      n  freq
##    <fct>                         <fct>   <int> <dbl>
##  1 Academic degree               0         161 98.2
##  2 Academic degree               1           3  1.83
##  3 Higher education              0       70854 94.6
##  4 Higher education              1        4009  5.36
##  5 Incomplete higher             0        9405 91.5
##  6 Incomplete higher             1         872  8.48
##  7 Lower secondary               0        3399 89.1
##  8 Lower secondary               1         417 10.9
##  9 Secondary / secondary special 0      198867 91.1
## 10 Secondary / secondary special 1       19524  8.94
```

```
ggplot(data = clean_train, aes(x=NAME_EDUCATION_TYPE, color = TARGET)) + geom_bar() + labs(title = "NAM
```

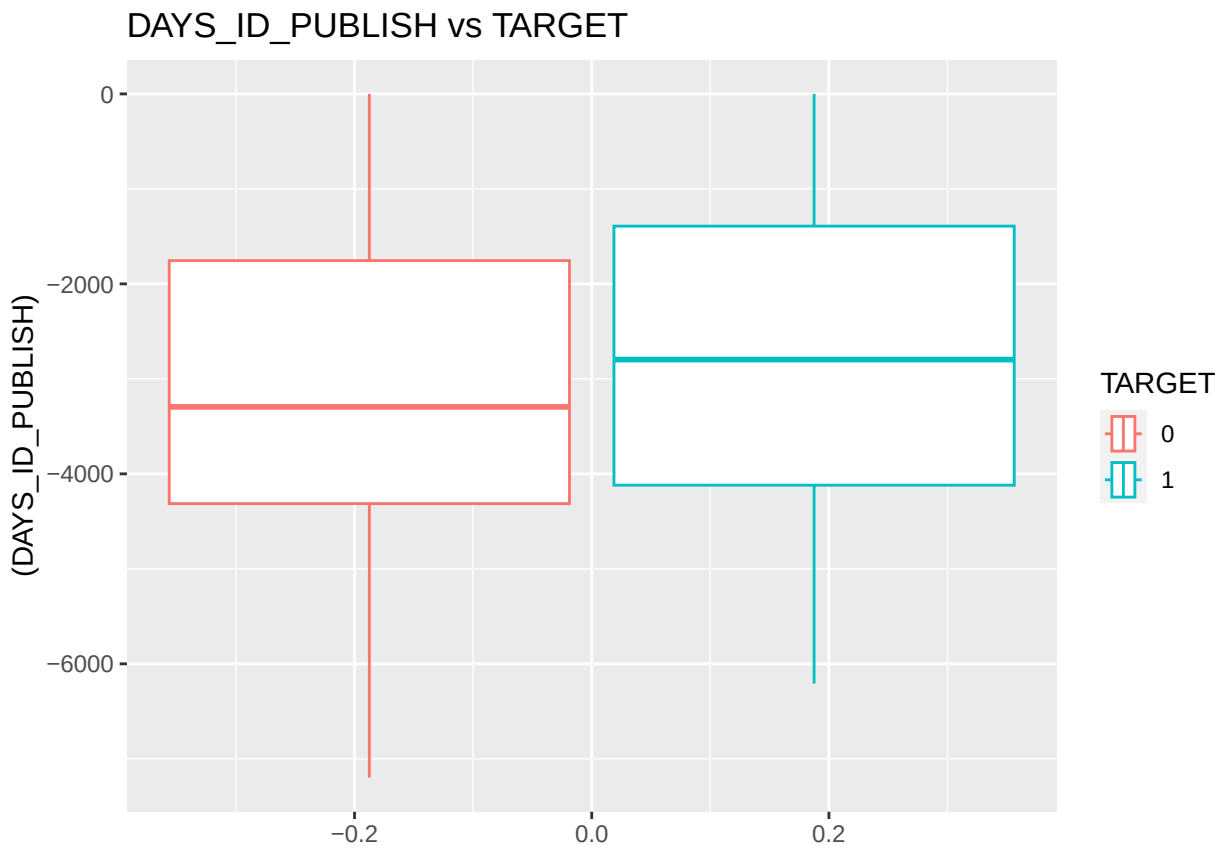



```
#DAYS_REGISTRATION slight difference in means possibility of prediction
ggplot(data = DEtest, aes(x=(DAYS_REGISTRATION), color = TARGET)) + geom_boxplot() + labs(title = "DAYS_
```


```
clean_train %>%
  group_by(TARGET) %>%
    summarise(mean = mean((DAYS_REGISTRATION)))
```

```
## # A tibble: 2 x 2
##   TARGET   mean
##   <fct>   <dbl>
## 1 0      -5030.
## 2 1      -4487.
```

```
#DAYS_ID_PUBLISH slight difference in means possibility of prediction
ggplot(data = DEtest, aes(x=(DAYS_ID_PUBLISH), color = TARGET)) + geom_boxplot() + labs(title = "DAYS_I
```




```
clean_train %>%
  group_by(TARGET) %>%
    summarise(mean = mean((DAYS_ID_PUBLISH)))
```

```
## # A tibble: 2 x 2
##   TARGET   mean
##   <fct>   <dbl>
## 1 0      -3017.
## 2 1      -2732.
```

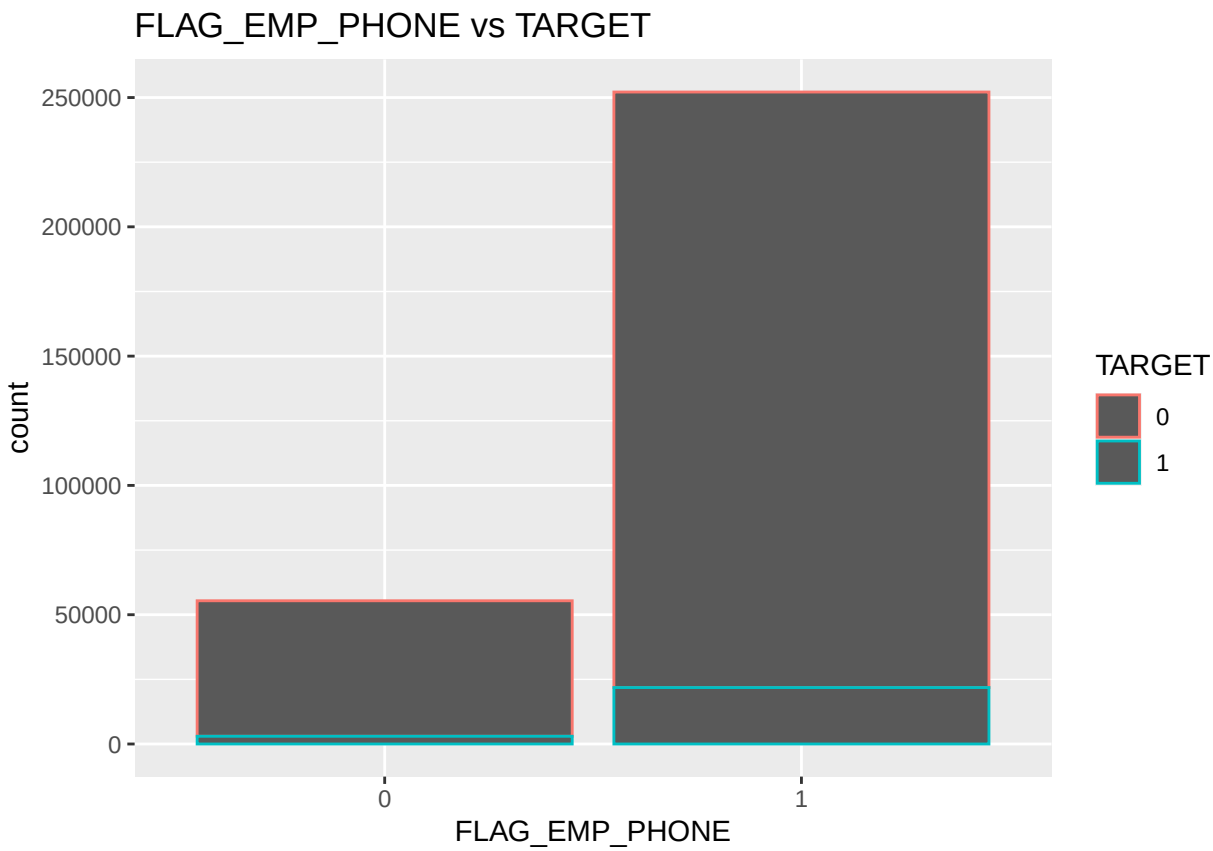
```
#FLAG_EMP_PHONE 3% difference between groups possibility of prediction
clean_train %>%
  group_by(FLAG_EMP_PHONE,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'FLAG_EMP_PHONE'. You can override using
## the `.groups` argument.
```

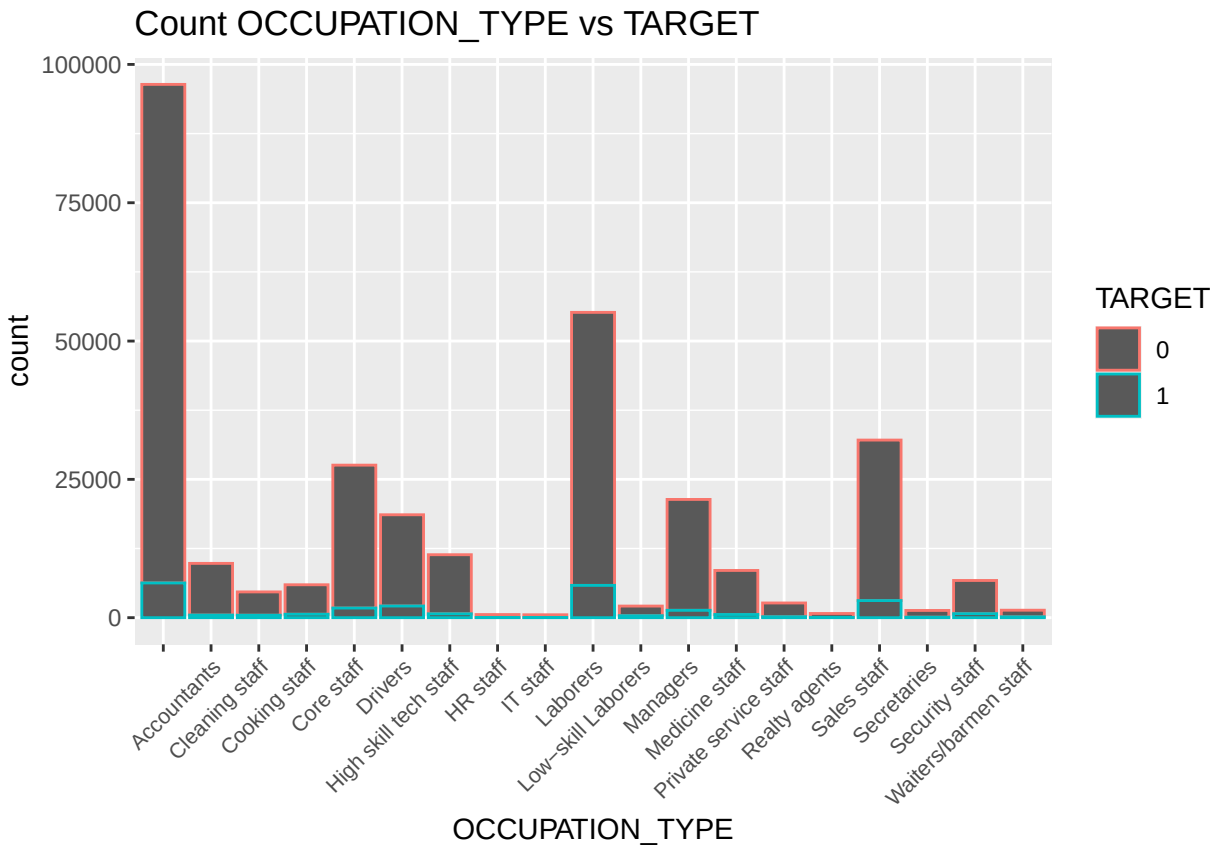
```
## # A tibble: 4 x 4
## # Groups:   FLAG_EMP_PHONE [2]
##   FLAG_EMP_PHONE TARGET      n  freq
##   <fct>          <fct>   <int> <dbl>
```

```
## 1 0              0       52395 94.6
## 2 0              1        2991  5.40
## 3 1              0      230291 91.3
## 4 1              1       21834  8.66
```

```
ggplot(data = clean_train, aes(x=FLAG_EMP_PHONE, color = TARGET)) + geom_bar() + labs(title = "FLAG_EMP
```




```
#OCCUPATION_TYPE Visualize Varying differences between groups
ggplot(data = clean_train, aes(x=OCCUPATION_TYPE, color = TARGET)) + geom_bar() + labs(title = "Count O
```




```
clean_train %>%
  group_by(OCCUPATION_TYPE,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'OCCUPATION_TYPE'. You can override using
## the `.groups` argument.

## # A tibble: 38 x 4
## # Groups:   OCCUPATION_TYPE [19]
##    OCCUPATION_TYPE         TARGET     n  freq
##    <fct>                   <fct>  <int> <dbl>
##  1 ""                      0      90113 93.5
##  2 ""                      1       6278  6.51
##  3 "Accountants"           0       9339 95.2
##  4 "Accountants"           1        474  4.83
##  5 "Cleaning staff"        0       4206 90.4
##  6 "Cleaning staff"        1        447  9.61
##  7 "Cooking staff"         0       5325 89.6
##  8 "Cooking staff"         1        621 10.4
##  9 "Core staff"            0      25832 93.7
## 10 "Core staff"            1       1738  6.30
## 11 "Drivers"               0      16496 88.7
## 12 "Drivers"               1       2107 11.3
```

```
## 13 "High skill tech staff" 0      10679 93.8
## 14 "High skill tech staff" 1        701  6.16
## 15 "HR staff"              0        527 93.6
## 16 "HR staff"              1         36  6.39
## 17 "IT staff"              0        492 93.5
## 18 "IT staff"              1         34  6.46
## 19 "Laborers"              0      49348 89.4
## 20 "Laborers"              1       5838 10.6
## 21 "Low-skill Laborers"    0       1734 82.8
## 22 "Low-skill Laborers"    1        359 17.2
## 23 "Managers"              0      20043 93.8
## 24 "Managers"              1       1328  6.21
## 25 "Medicine staff"        0       7965 93.3
## 26 "Medicine staff"        1        572  6.70
## 27 "Private service staff" 0       2477 93.4
## 28 "Private service staff" 1        175  6.60
## 29 "Realty agents"         0        692 92.1
## 30 "Realty agents"         1         59  7.86
## 31 "Sales staff"           0      29010 90.4
## 32 "Sales staff"           1       3092  9.63
## 33 "Secretaries"           0       1213 93.0
## 34 "Secretaries"           1         92  7.05
## 35 "Security staff"        0       5999 89.3
## 36 "Security staff"        1        722 10.7
## 37 "Waiters/barmen staff"  0       1196 88.7
## 38 "Waiters/barmen staff"  1        152 11.3
```

```
#CNT_FAM_MEMBERS Same distribution as cnt_children possible for prediction
clean_train %>%
  group_by(CNT_FAM_MEMBERS,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

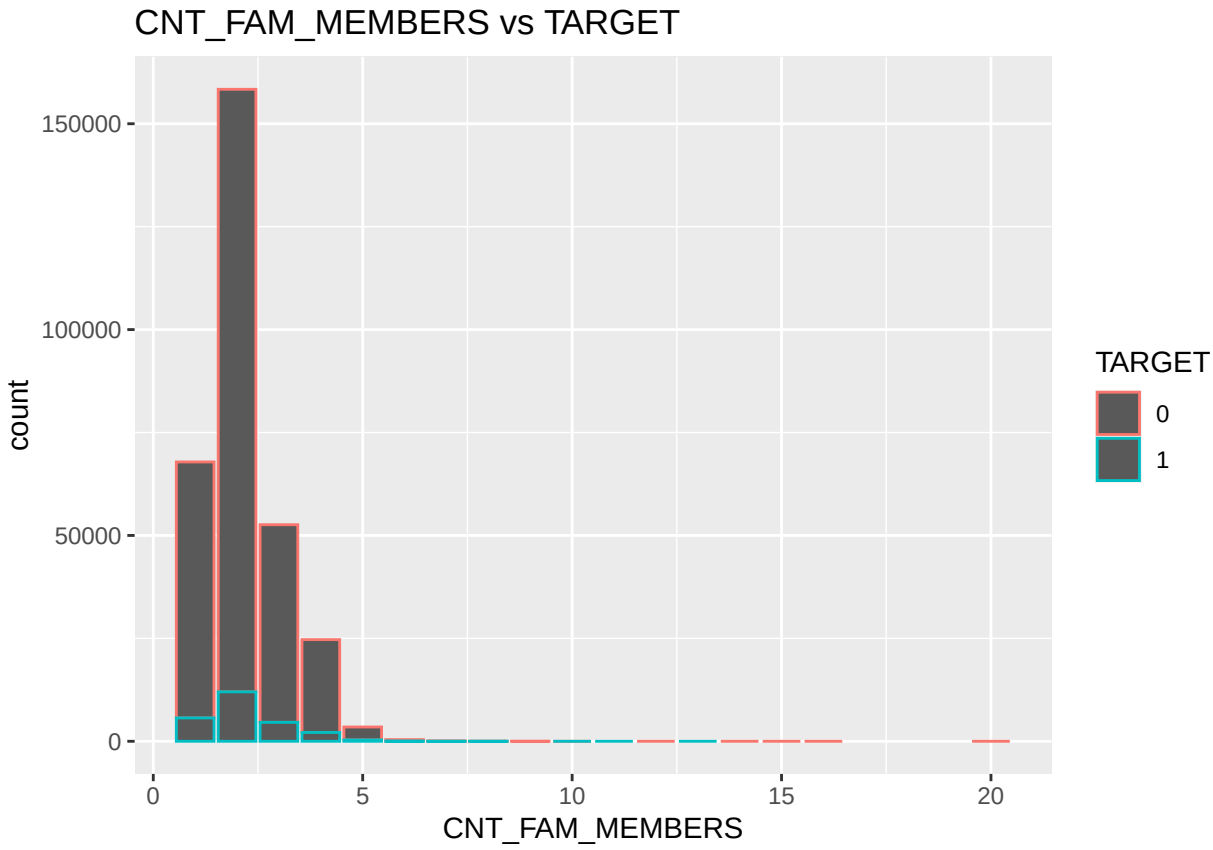
```
## `summarise()` has grouped output by 'CNT_FAM_MEMBERS'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 27 x 4
## # Groups:   CNT_FAM_MEMBERS [18]
##    CNT_FAM_MEMBERS TARGET      n   freq
##              <int> <fct>   <int>  <dbl>
##  1               1 0       62172  91.6
##  2               1 1        5675   8.36
##  3               2 0      146348  92.4
##  4               2 1       12009   7.58
##  5               3 0       47993  91.2
##  6               3 1        4608   8.76
##  7               4 0       22561  91.4
##  8               4 1        2136   8.65
##  9               5 0        3151  90.6
## 10               5 1         327   9.40
## 11               6 0         353  86.5
## 12               6 1          55  13.5
```

```
## 13               7 0          75  92.6
## 14               7 1           6   7.41
## 15               8 0          14  70
## 16               8 1           6  30
## 17               9 0           6 100
## 18              10 0           2  66.7
## 19              10 1           1  33.3
## 20              11 1           1 100
## 21              12 0           2 100
## 22              13 1           1 100
## 23              14 0           2 100
## 24              15 0           1 100
## 25              16 0           2 100
## 26              20 0           2 100
## 27              NA 0           2 100
```

```
ggplot(data = clean_train, aes(x=CNT_FAM_MEMBERS, color = TARGET)) + geom_bar() + labs(title = "CNT_FAM
```

```
## Warning: Removed 2 rows containing non-finite values (`stat_count()`).
```



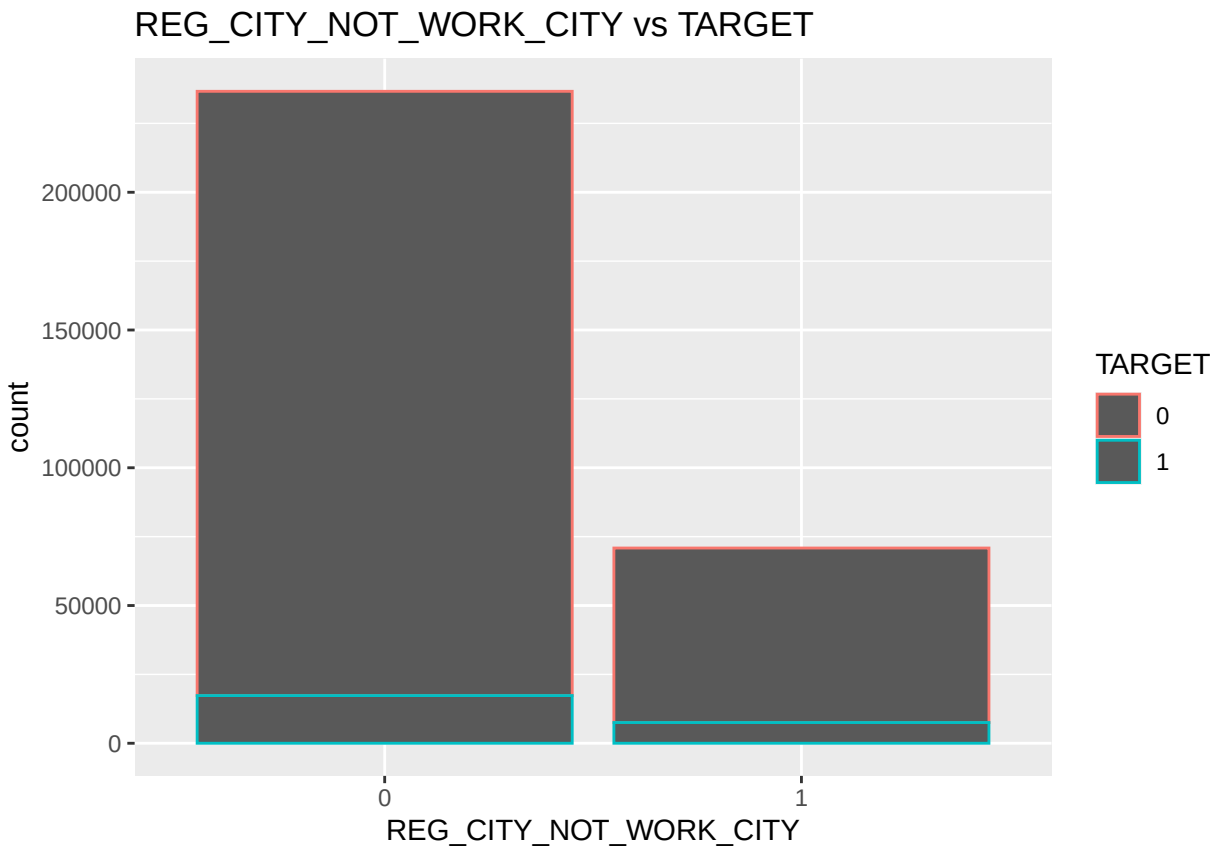

```
#REG_CITY_NOT_WORK_CITY large difference for those not living in city 10.6% default possible for predic
clean_train %>%
  group_by(REG_CITY_NOT_WORK_CITY,TARGET) %>%
    summarise(n=n()) %>%
```

```
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'REG_CITY_NOT_WORK_CITY'. You can override
## using the `.groups` argument.
```

```
## # A tibble: 4 x 4
## # Groups:   REG_CITY_NOT_WORK_CITY [2]
##   REG_CITY_NOT_WORK_CITY TARGET      n  freq
##   <fct>                  <fct>   <int> <dbl>
## 1 0                      0      219339 92.7
## 2 0                      1       17305  7.31
## 3 1                      0       63347 89.4
## 4 1                      1        7520 10.6
```

```
ggplot(data = clean_train, aes(x=REG_CITY_NOT_WORK_CITY, color = TARGET)) + geom_bar() + labs(title = "
```

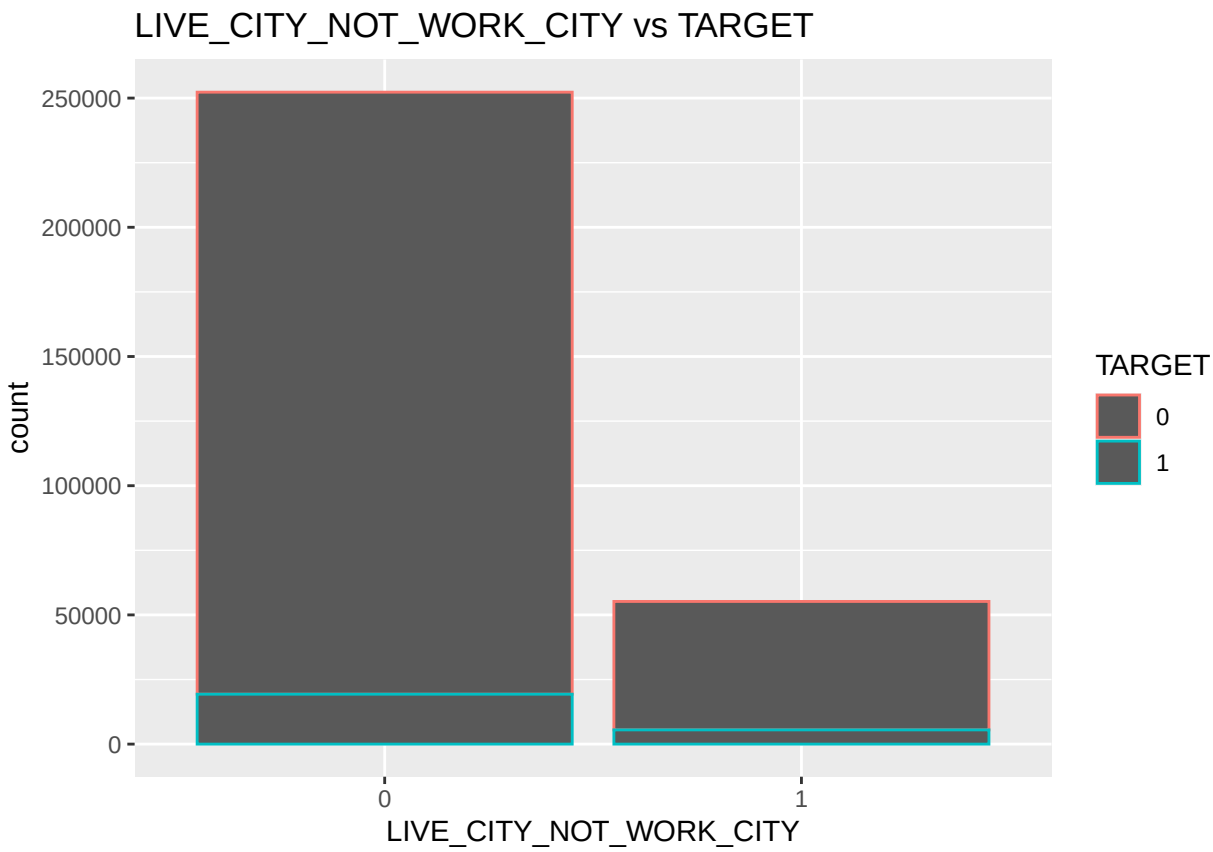



```
#LIVE_CITY_NOT_WORK_CITY large difference for those not living in city 9.97% default possible for predi
clean_train %>%
  group_by(LIVE_CITY_NOT_WORK_CITY,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'LIVE_CITY_NOT_WORK_CITY'. You can override
## using the `.groups` argument.
```

```
## # A tibble: 4 x 4
## # Groups:   LIVE_CITY_NOT_WORK_CITY [2]
##   LIVE_CITY_NOT_WORK_CITY TARGET      n  freq
##   <fct>                   <fct>   <int> <dbl>
## 1 0                       0      232974 92.3
## 2 0                       1       19322  7.66
## 3 1                       0       49712 90.0
## 4 1                       1        5503  9.97
```

```
ggplot(data = clean_train, aes(x=LIVE_CITY_NOT_WORK_CITY, color = TARGET)) + geom_bar() + labs(title =
```



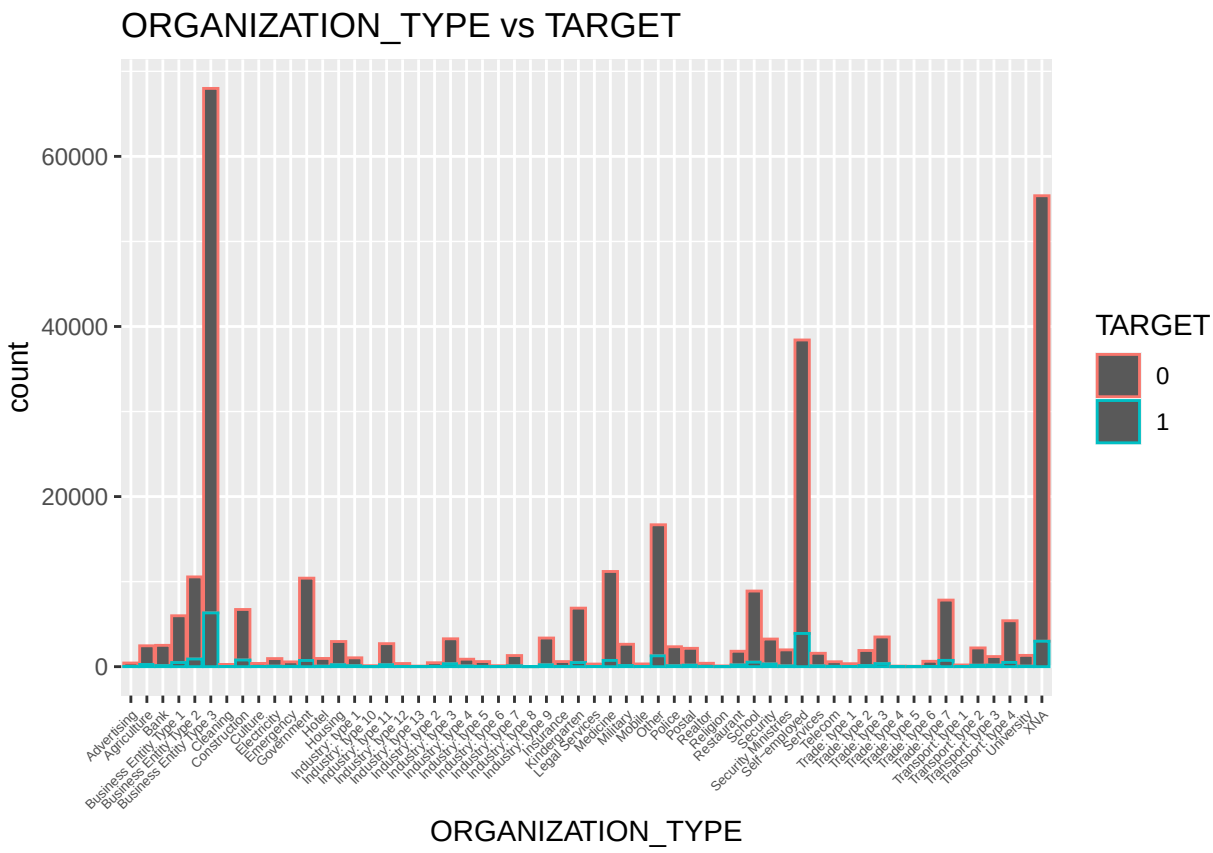

```
#ORGANIZATION_TYPE lots of variation between groups possible for prediction
clean_train %>%
  group_by(ORGANIZATION_TYPE,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'ORGANIZATION_TYPE'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 116 x 4
```

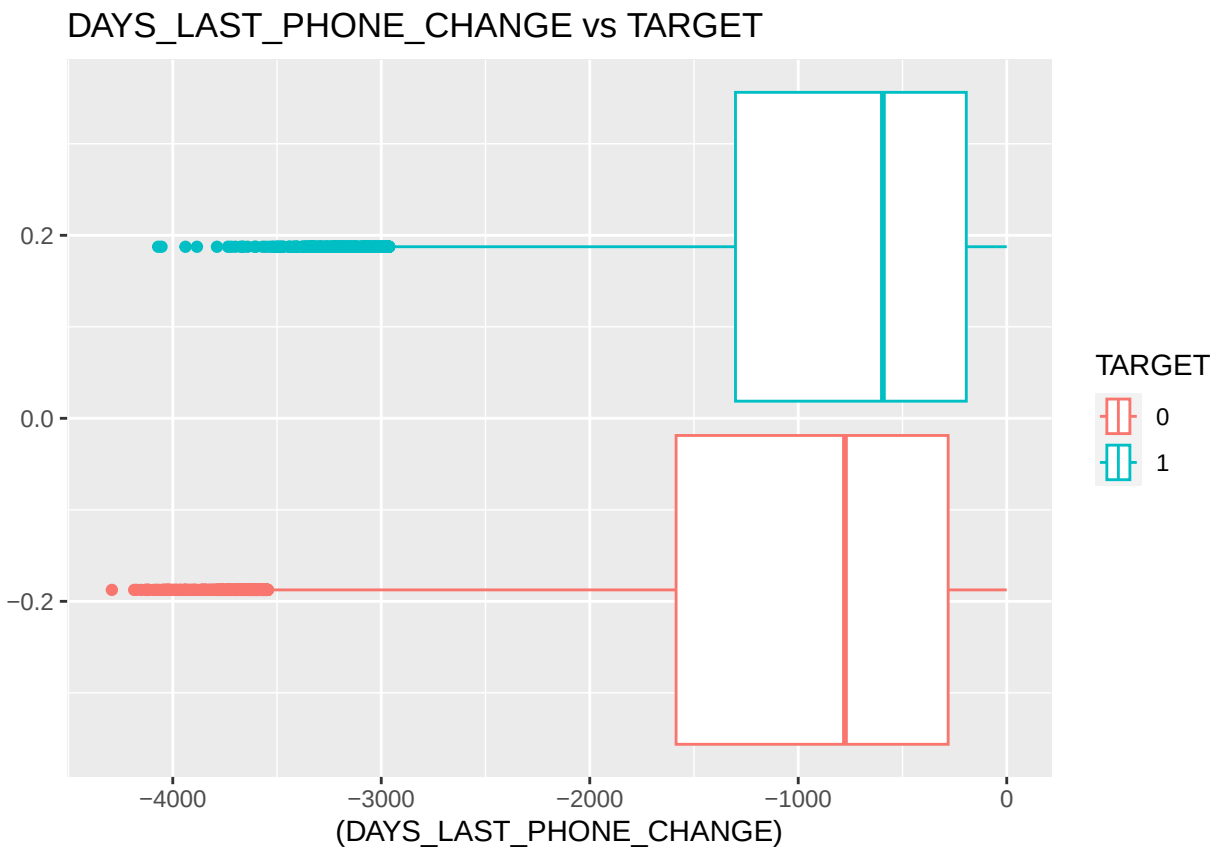
```
## # Groups:   ORGANIZATION_TYPE [58]
##    ORGANIZATION_TYPE      TARGET     n  freq
##    <fct>                  <fct>  <int> <dbl>
##  1 Advertising            0        394 91.8
##  2 Advertising            1         35  8.16
##  3 Agriculture            0       2197 89.5
##  4 Agriculture            1        257 10.5
##  5 Bank                   0       2377 94.8
##  6 Bank                   1        130  5.19
##  7 Business Entity Type 1 0       5497 91.9
##  8 Business Entity Type 1 1        487  8.14
##  9 Business Entity Type 2 0       9653 91.5
## 10 Business Entity Type 2 1        900  8.53
## 11 Business Entity Type 3 0      61669 90.7
## 12 Business Entity Type 3 1       6323  9.30
## 13 Cleaning               0        231 88.8
## 14 Cleaning               1         29 11.2
## 15 Construction           0       5936 88.3
## 16 Construction           1        785 11.7
## 17 Culture                0        358 94.5
## 18 Culture                1         21  5.54
## 19 Electricity            0        887 93.4
## 20 Electricity            1         63  6.63
## 21 Emergency              0        520 92.9
## 22 Emergency              1         40  7.14
## 23 Government             0       9678 93.0
## 24 Government             1        726  6.98
## 25 Hotel                  0        904 93.6
## 26 Hotel                  1         62  6.42
## 27 Housing                0       2723 92.1
## 28 Housing                1        235  7.94
## 29 Industry: type 1       0        924 88.9
## 30 Industry: type 1       1        115 11.1
## 31 Industry: type 10      0        102 93.6
## 32 Industry: type 10      1          7  6.42
## 33 Industry: type 11      0       2470 91.3
## 34 Industry: type 11      1        234  8.65
## 35 Industry: type 12      0        355 96.2
## 36 Industry: type 12      1         14  3.79
## 37 Industry: type 13      0         58 86.6
## 38 Industry: type 13      1          9 13.4
## 39 Industry: type 2       0        425 92.8
## 40 Industry: type 2       1         33  7.21
## 41 Industry: type 3       0       2930 89.4
## 42 Industry: type 3       1        348 10.6
## 43 Industry: type 4       0        788 89.9
## 44 Industry: type 4       1         89 10.1
## 45 Industry: type 5       0        558 93.2
## 46 Industry: type 5       1         41  6.84
## 47 Industry: type 6       0        104 92.9
## 48 Industry: type 6       1          8  7.14
## 49 Industry: type 7       0       1202 92.0
## 50 Industry: type 7       1        105  8.03
## # i 66 more rows
```

```
ggplot(data = clean_train, aes(x=ORGANIZATION_TYPE, color = TARGET)) + geom_bar() + labs(title = "ORGANI
```




```
#DAYS_LAST_PHONE_CHANGE difference in means possible predictor
ggplot(data = clean_train, aes(x=(DAYS_LAST_PHONE_CHANGE), color = TARGET)) + geom_boxplot() + labs(titl
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_boxplot()`).
```

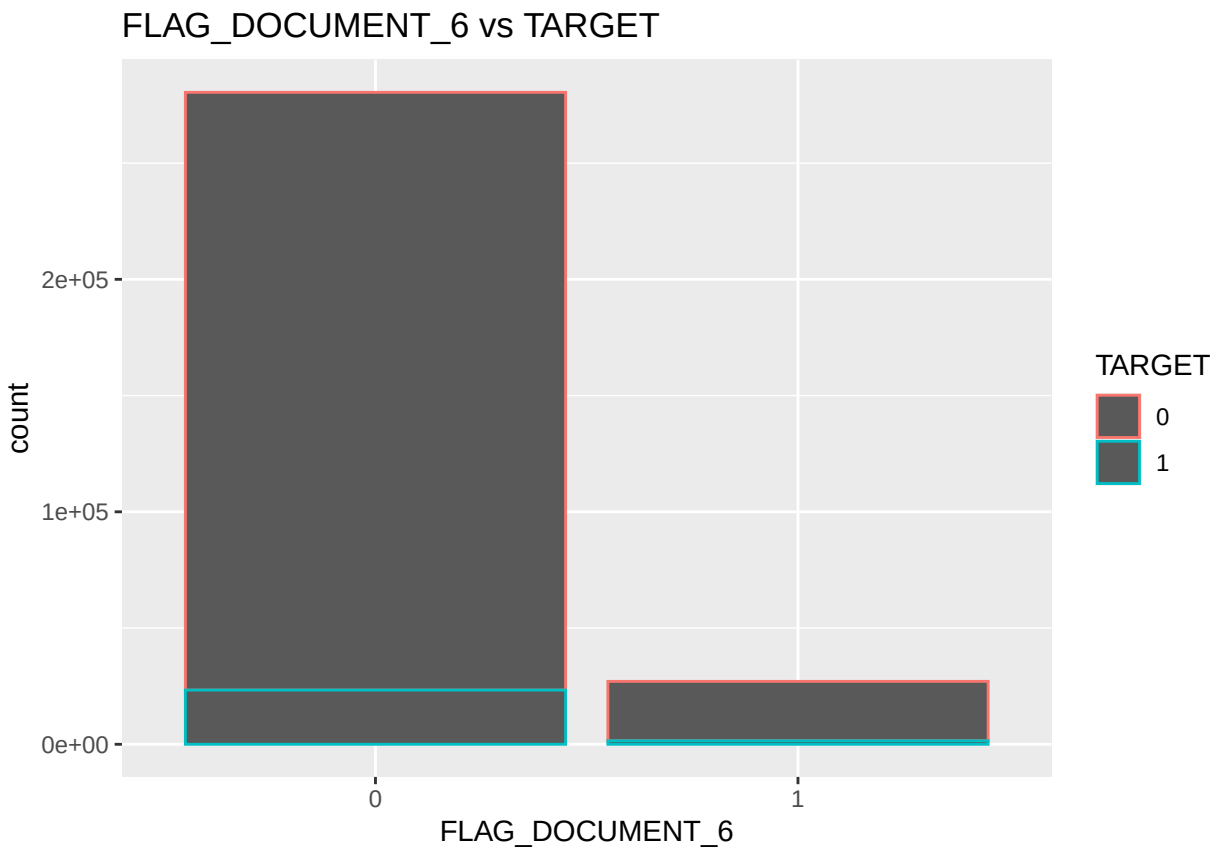



```
#FLAG_DOCUMENT_6 large difference between default in those who provided the doc. possible
clean_train %>%
  group_by(FLAG_DOCUMENT_6,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'FLAG_DOCUMENT_6'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 4 x 4
## # Groups:   FLAG_DOCUMENT_6 [2]
##   FLAG_DOCUMENT_6 TARGET      n  freq
##   <fct>           <fct>   <int> <dbl>
## 1 0               0      257115 91.7
## 2 0               1       23318  8.31
## 3 1               0       25571 94.4
## 4 1               1        1507  5.57
```

```
ggplot(data = clean_train, aes(x=FLAG_DOCUMENT_6, color = TARGET)) + geom_bar() + labs(title = "FLAG_DO
```



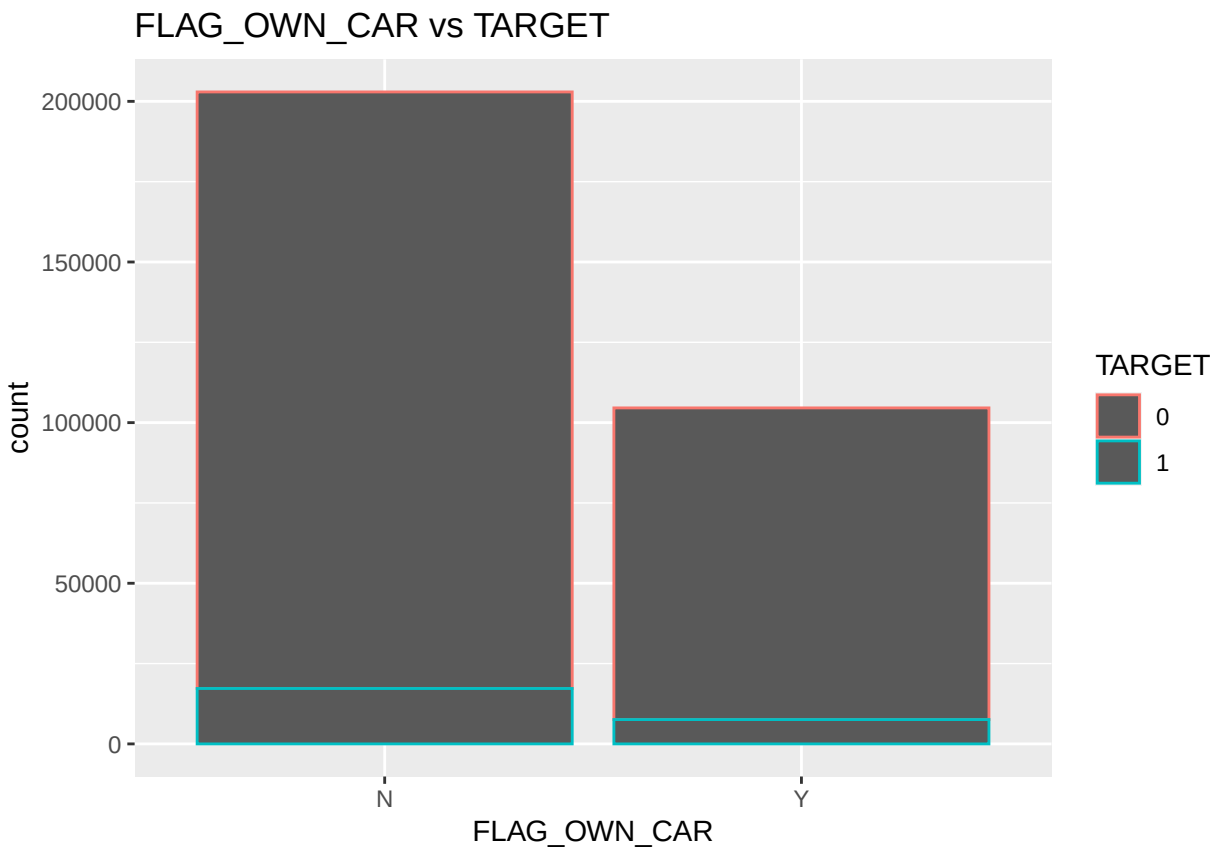
No Difference

```
#FLAG_OWN_CAR 1.25% difference between default on No vs Yes
clean_train %>%
  group_by(FLAG_OWN_CAR,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'FLAG_OWN_CAR'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 4 x 4
## # Groups:   FLAG_OWN_CAR [2]
##   FLAG_OWN_CAR TARGET      n  freq
##   <fct>        <fct>   <int> <dbl>
## 1 N            0      185675 91.5
## 2 N            1       17249  8.50
## 3 Y            0       97011 92.8
## 4 Y            1        7576  7.24
```

```
ggplot(data = clean_train, aes(x=FLAG_OWN_CAR, color = TARGET)) + geom_bar() + labs(title = "FLAG_OWN_CA
```

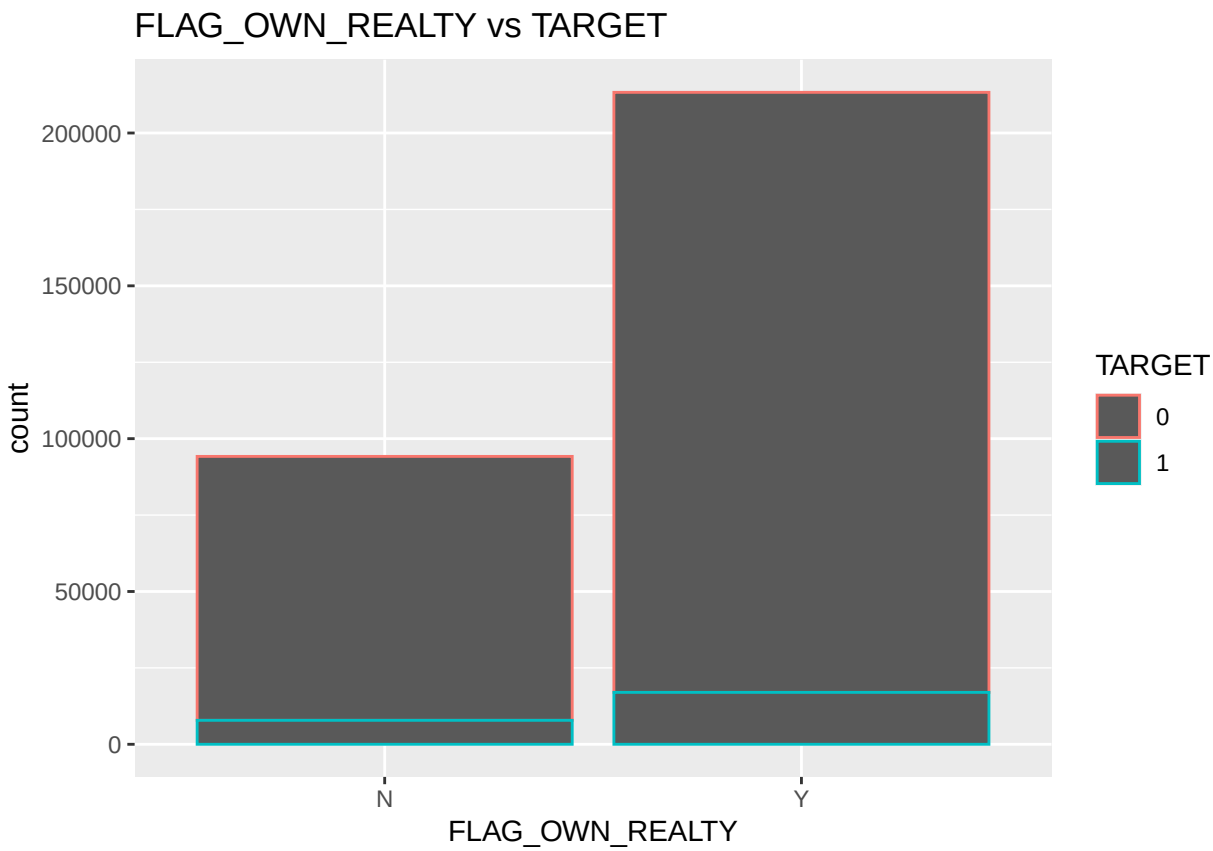



```
#FLAG_OWN_REALTY >1% difference between default on No vs Yes
clean_train %>%
  group_by(FLAG_OWN_REALTY,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

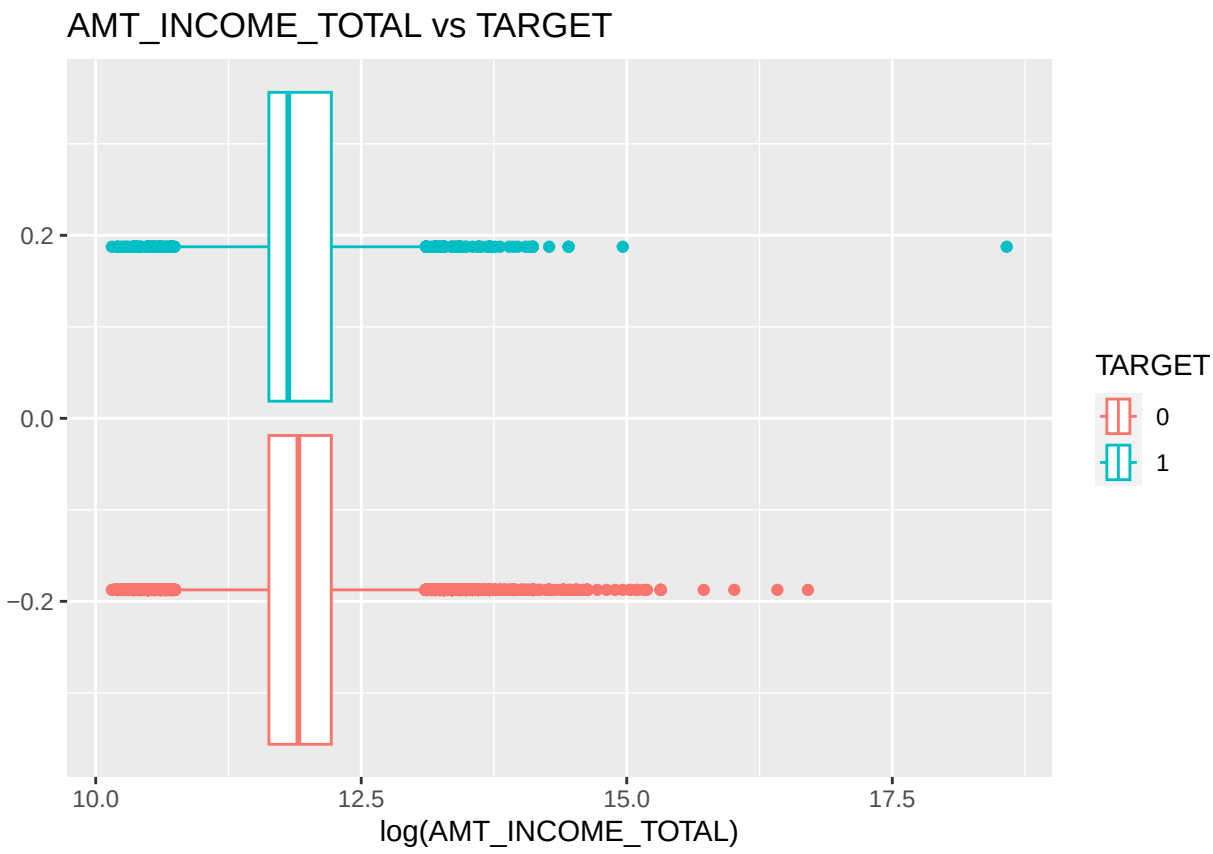
```
## `summarise()` has grouped output by 'FLAG_OWN_REALTY'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 4 x 4
## # Groups:   FLAG_OWN_REALTY [2]
##   FLAG_OWN_REALTY TARGET      n  freq
##   <fct>           <fct>   <int> <dbl>
## 1 N               0       86357 91.7
## 2 N               1        7842  8.32
## 3 Y               0      196329 92.0
## 4 Y               1       16983  7.96
```

```
ggplot(data = clean_train, aes(x=FLAG_OWN_REALTY, color = TARGET)) + geom_bar() + labs(title = "FLAG_OWN
```




```
#AMT_INCOME_TOTAL no significant difference in values or logged values. Need to address outliars
ggplot(data = clean_train, aes(x=log(AMT_INCOME_TOTAL), color = TARGET)) + geom_boxplot() + labs(title =
```



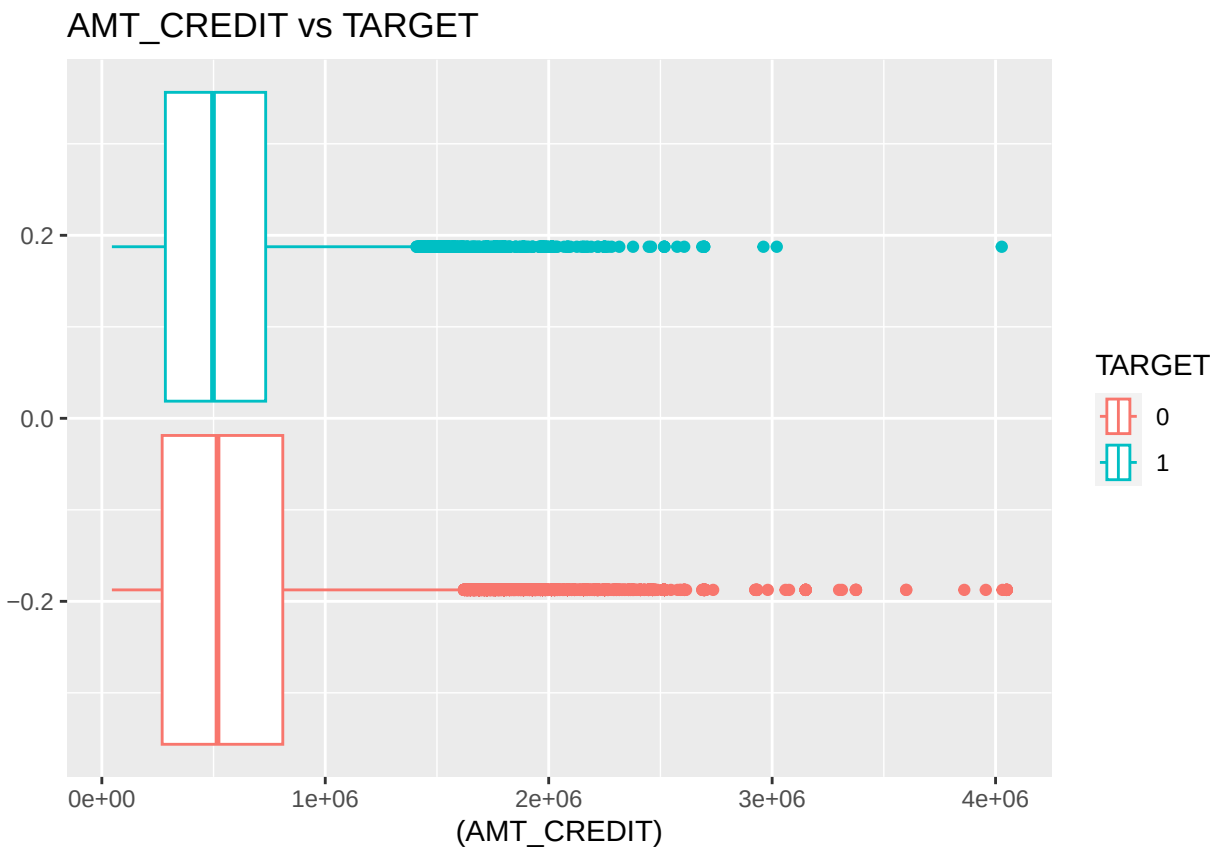

```
clean_train %>%
  group_by(TARGET) %>%
    summarise(mean = mean((AMT_INCOME_TOTAL)))
```

```
## # A tibble: 2 x 2
##   TARGET    mean
##   <fct>    <dbl>
## 1 0      169078.
## 2 1      165612.
```

```
clean_train %>%
  group_by(TARGET) %>%
    summarise(mean = mean(log(AMT_INCOME_TOTAL)))
```

```
## # A tibble: 2 x 2
##   TARGET  mean
##   <fct>  <dbl>
## 1 0       11.9
## 2 1       11.9
```

```
#AMT_CREDIT no significant difference in means.
ggplot(data = clean_train, aes(x=(AMT_CREDIT), color = TARGET)) + geom_boxplot() + labs(title = "AMT_CR
```

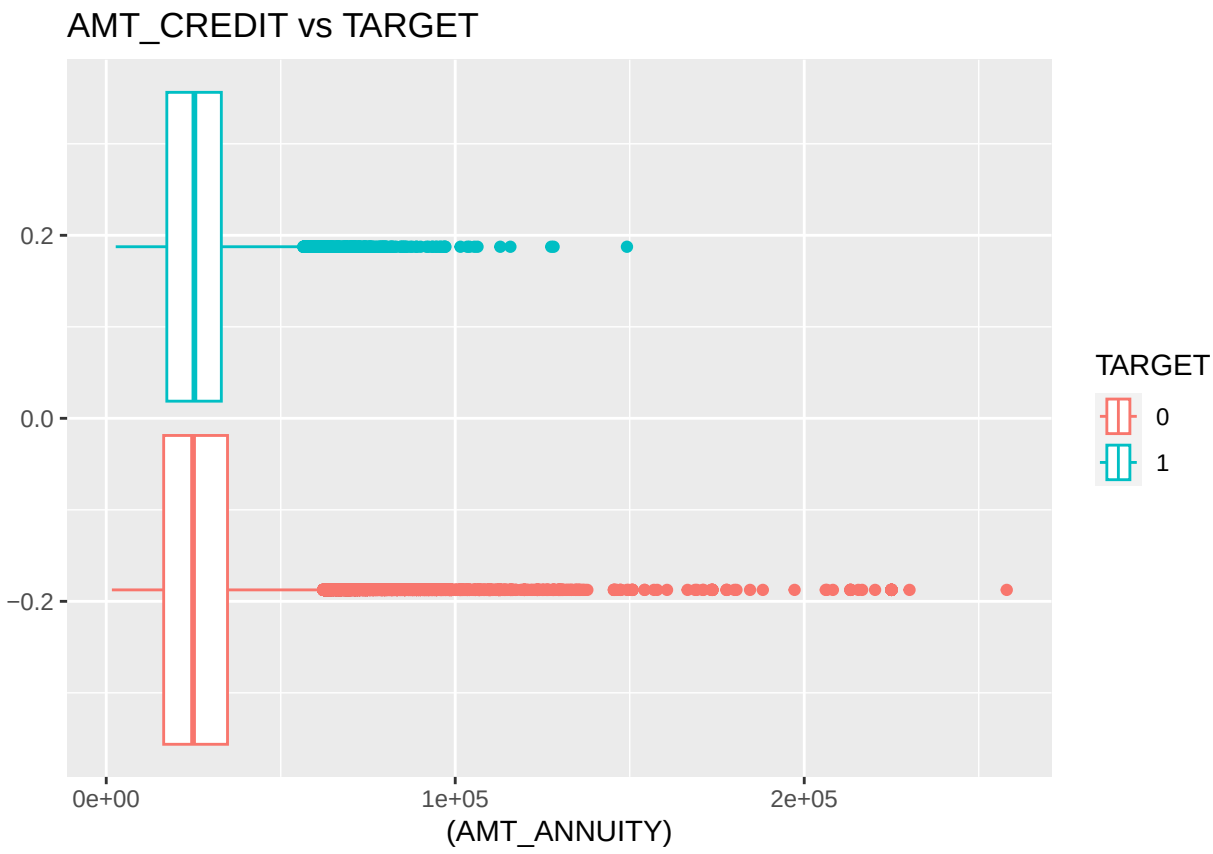



```
clean_train %>%
  group_by(TARGET) %>%
    summarise(mean = mean((AMT_CREDIT)))
```

```
## # A tibble: 2 x 2
##   TARGET    mean
##   <fct>    <dbl>
## 1 0      602648.
## 2 1      557779.
```

```
#AMT_ANNUITY no significant difference in means.needed to replace 12 NA with Avg of Group
ggplot(data = clean_train, aes(x=(AMT_ANNUITY), color = TARGET)) + geom_boxplot() + labs(title = "AMT_C
```

```
## Warning: Removed 12 rows containing non-finite values (‘stat_boxplot()‘).
```




```
clean_train %>%
    mutate(across(AMT_ANNUITY, ~replace_na(., mean(., na.rm=TRUE)))) %>%
  group_by(TARGET) %>%
    summarise(mean = mean((AMT_ANNUITY)))
```

```
## # A tibble: 2 x 2
##   TARGET   mean
##   <fct>   <dbl>
## 1 0      27164.
## 2 1      26482.
```

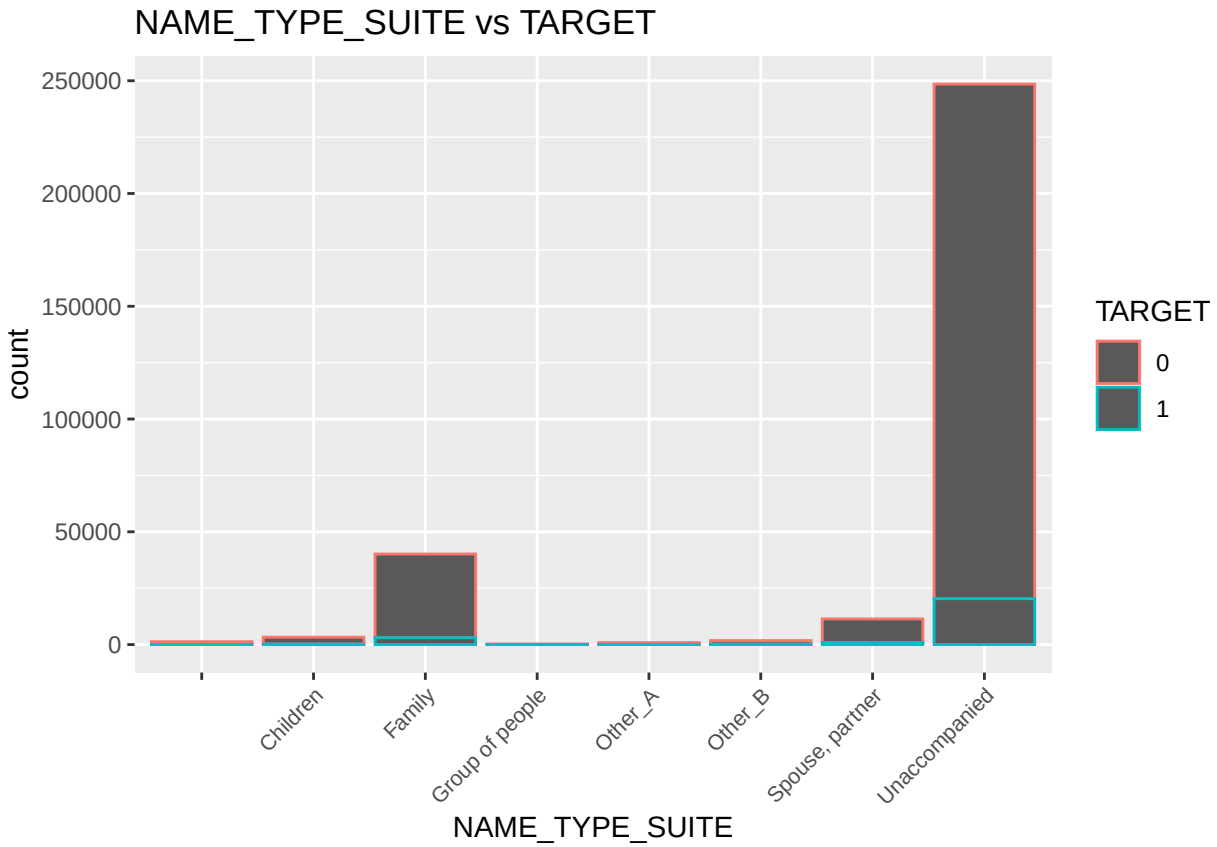
```
#NAME_TYPE_SUITE No significant difference between groups
clean_train %>%
  group_by(NAME_TYPE_SUITE,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'NAME_TYPE_SUITE'. You can override using
## the `.groups` argument.
```

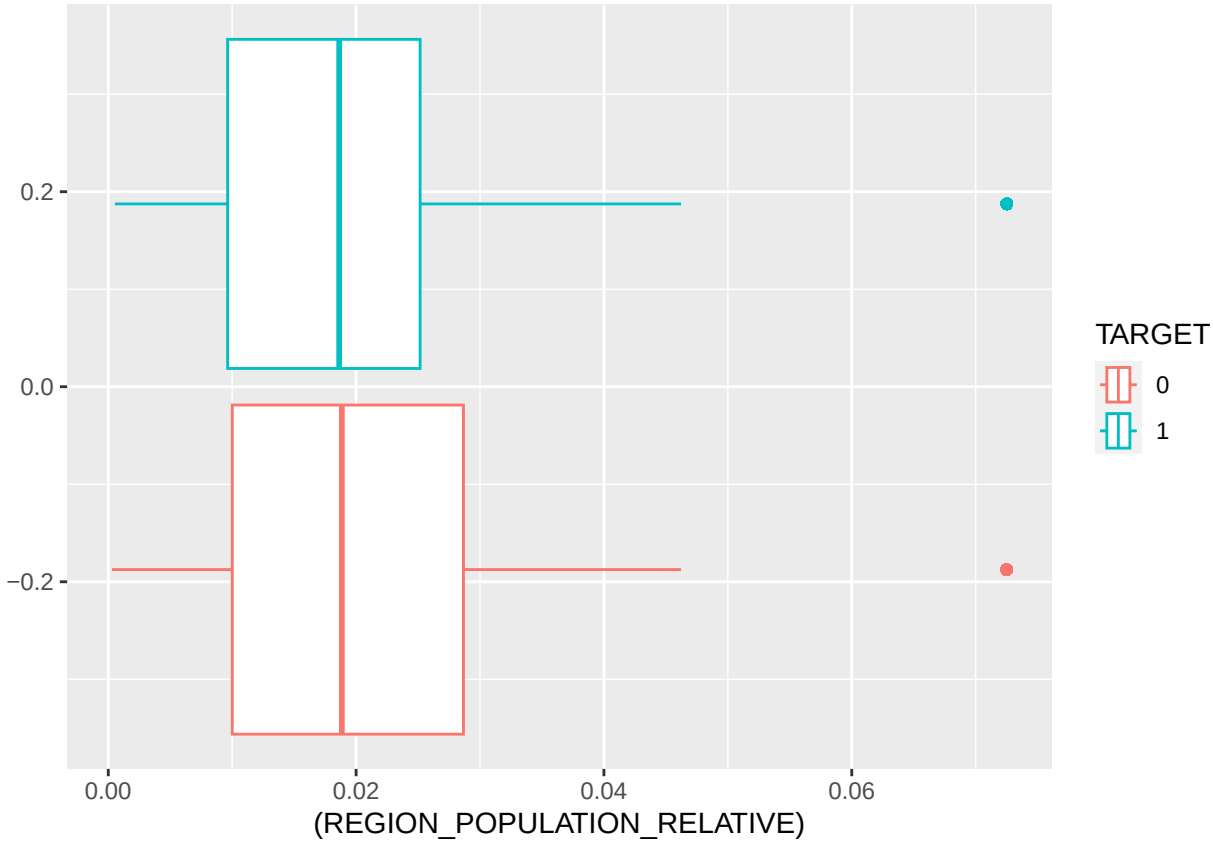
```
## # A tibble: 16 x 4
## # Groups:   NAME_TYPE_SUITE [8]
##    NAME_TYPE_SUITE   TARGET      n  freq
```

```
##    <fct>             <fct>   <int> <dbl>
##  1 ""                0        1222 94.6
##  2 ""                1          70  5.42
##  3 "Children"        0        3026 92.6
##  4 "Children"        1         241  7.38
##  5 "Family"          0       37140 92.5
##  6 "Family"          1        3009  7.49
##  7 "Group of people" 0         248 91.5
##  8 "Group of people" 1          23  8.49
##  9 "Other_A"         0         790 91.2
## 10 "Other_A"         1          76  8.78
## 11 "Other_B"         0        1596 90.2
## 12 "Other_B"         1         174  9.83
## 13 "Spouse, partner" 0       10475 92.1
## 14 "Spouse, partner" 1         895  7.87
## 15 "Unaccompanied"   0      228189 91.8
## 16 "Unaccompanied"   1       20337  8.18
```

```
ggplot(data = clean_train, aes(x=NAME_TYPE_SUITE, color = TARGET)) + geom_bar() + labs(title = "NAME_TY
```




```
#REGION_POPULATION_RELATIVE No large differnce in means
ggplot(data = clean_train, aes(x=(REGION_POPULATION_RELATIVE), color = TARGET)) + geom_boxplot()
```




```
clean_train %>%
  group_by(TARGET) %>%
    summarise(mean = mean((REGION_POPULATION_RELATIVE)))
```

```
## # A tibble: 2 x 2
##   TARGET   mean
##   <fct>   <dbl>
## 1 0      0.0210
## 2 1      0.0191
```

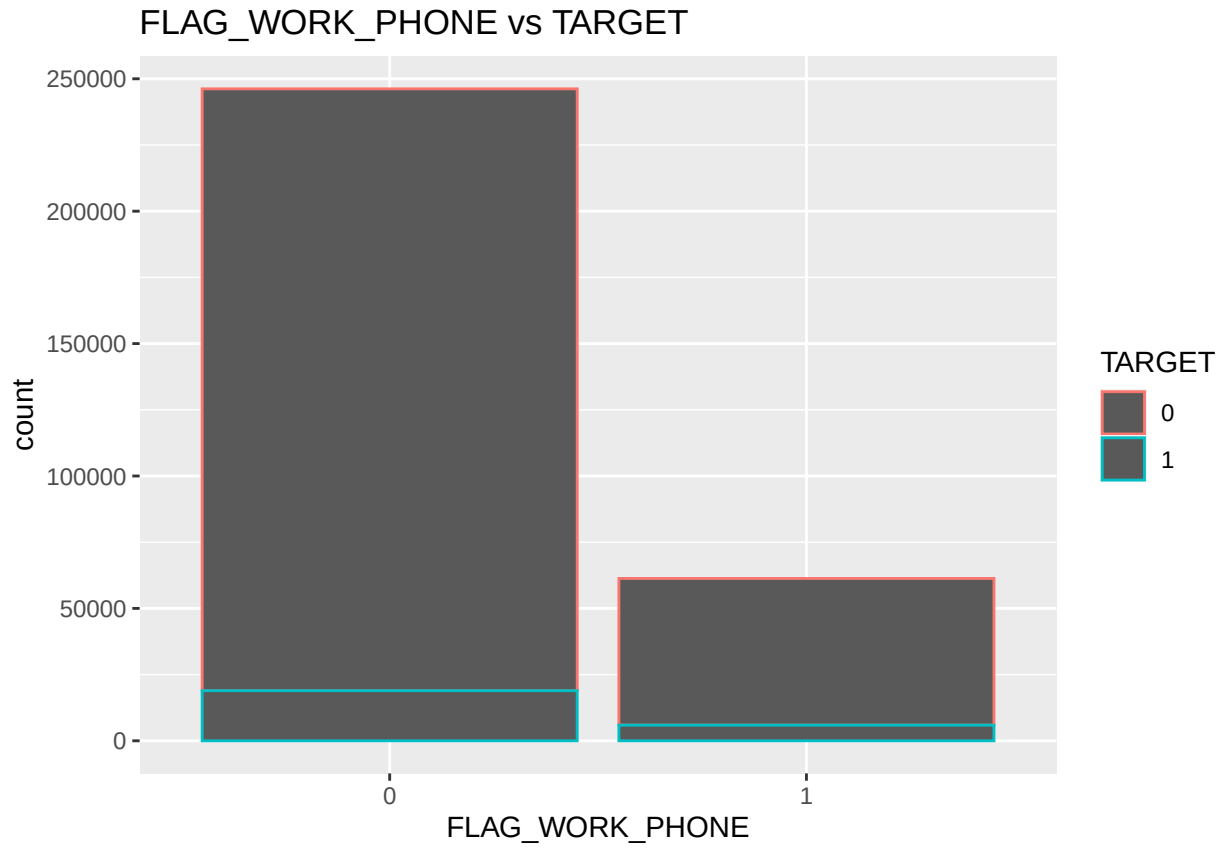
```
#FLAG_WORK_PHONE no large difference in groups
clean_train %>%
  group_by(FLAG_WORK_PHONE,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'FLAG_WORK_PHONE'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 4 x 4
## # Groups:   FLAG_WORK_PHONE [2]
##   FLAG_WORK_PHONE TARGET      n  freq
##   <fct>           <fct>   <int> <dbl>
```

```
## 1 0               0      227282 92.3
## 2 0               1       18921  7.69
## 3 1               0       55404 90.4
## 4 1               1        5904  9.63
```

```
ggplot(data = clean_train, aes(x=FLAG_WORK_PHONE, color = TARGET)) + geom_bar() + labs(title = "FLAG_WO
```

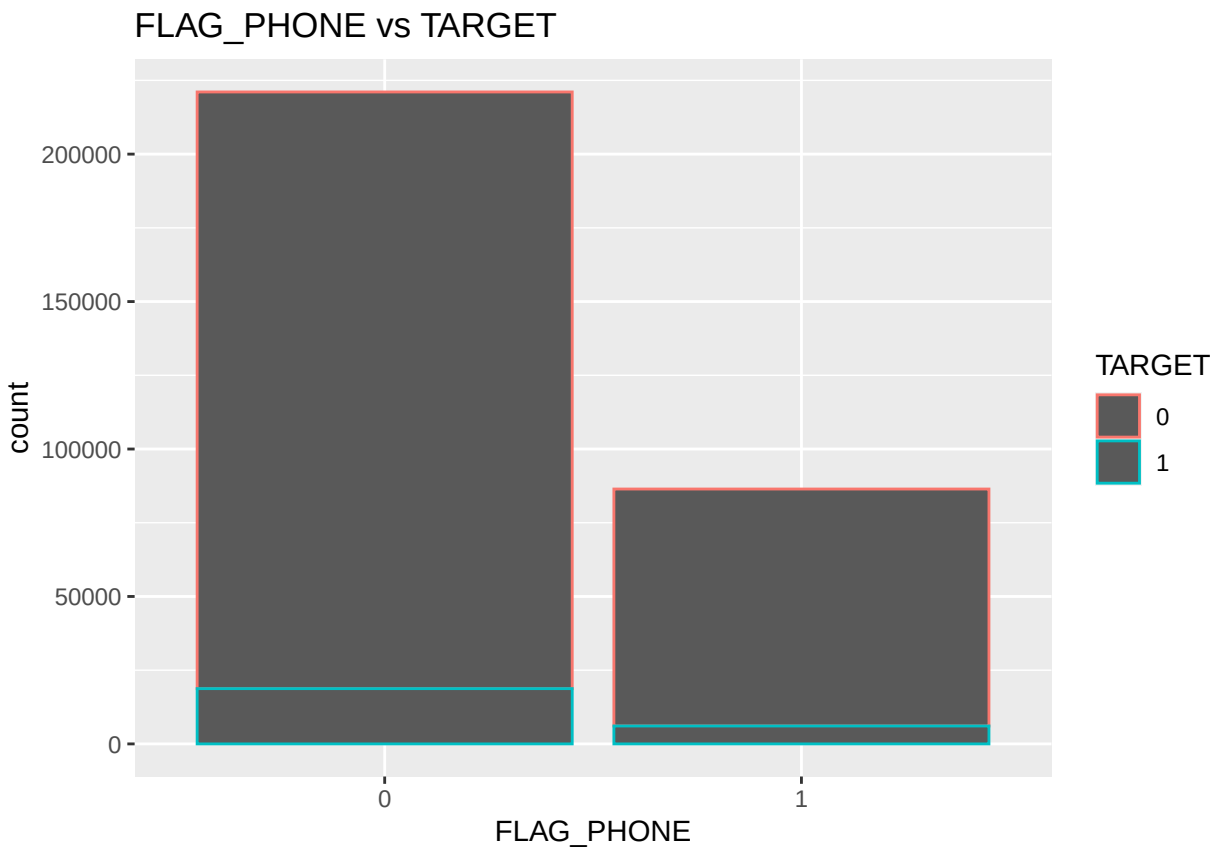



```
#FLAG_PHONE no large difference in groups
clean_train %>%
  group_by(FLAG_PHONE,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'FLAG_PHONE'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 4 x 4
## # Groups:   FLAG_PHONE [2]
##   FLAG_PHONE TARGET      n  freq
##   <fct>      <fct>   <int> <dbl>
## 1 0          0      202336 91.5
## 2 0          1       18744  8.48
## 3 1          0       80350 93.0
## 4 1          1        6081  7.04
```

```
ggplot(data = clean_train, aes(x=FLAG_PHONE, color = TARGET)) + geom_bar() + labs(title = "FLAG_PHONE vs
```

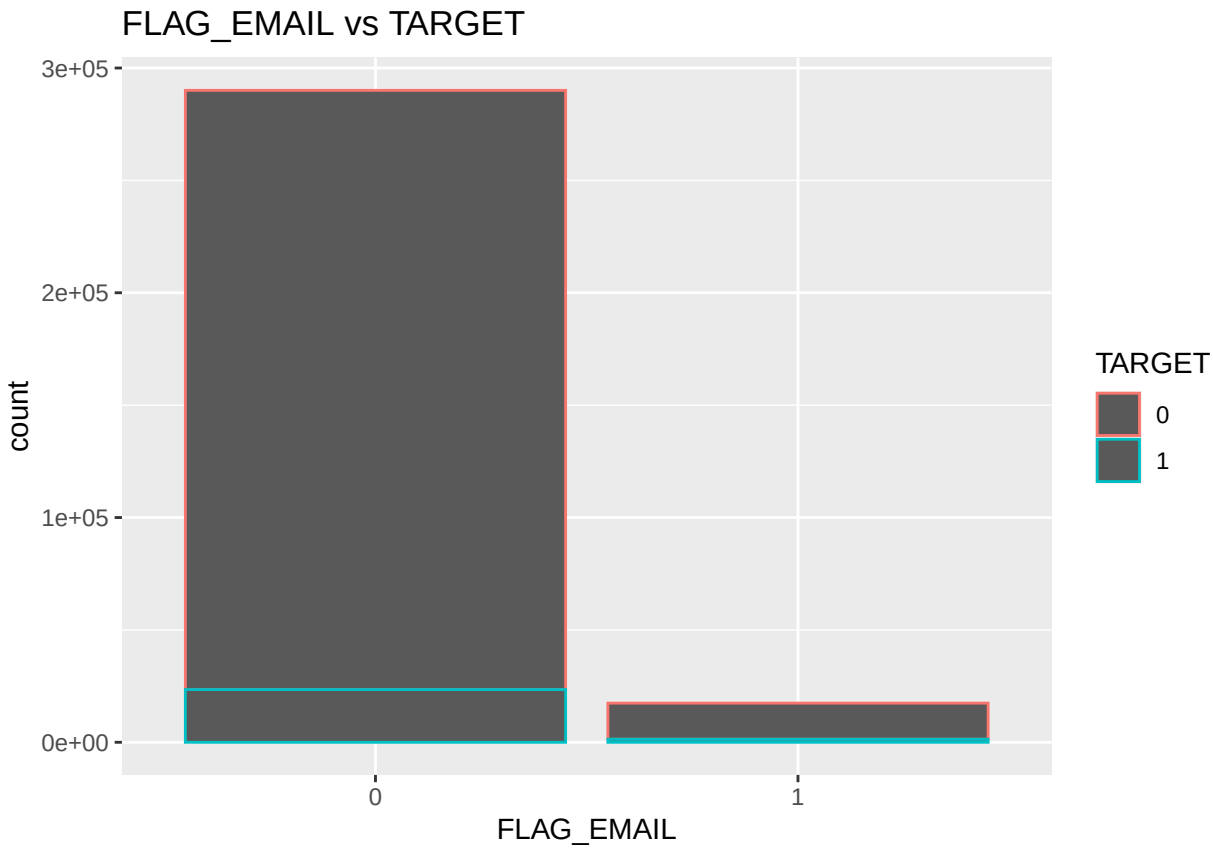



```
#FLAG_EMAIL no large difference in groups
clean_train %>%
  group_by(FLAG_EMAIL,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'FLAG_EMAIL'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 4 x 4
## # Groups:   FLAG_EMAIL [2]
##   FLAG_EMAIL TARGET      n  freq
##   <fct>      <fct>   <int> <dbl>
## 1 0          0      266618 91.9
## 2 0          1       23451  8.08
## 3 1          0       16068 92.1
## 4 1          1        1374  7.88
```

```
ggplot(data = clean_train, aes(x=FLAG_EMAIL, color = TARGET)) + geom_bar() + labs(title = "FLAG_EMAIL vs
```



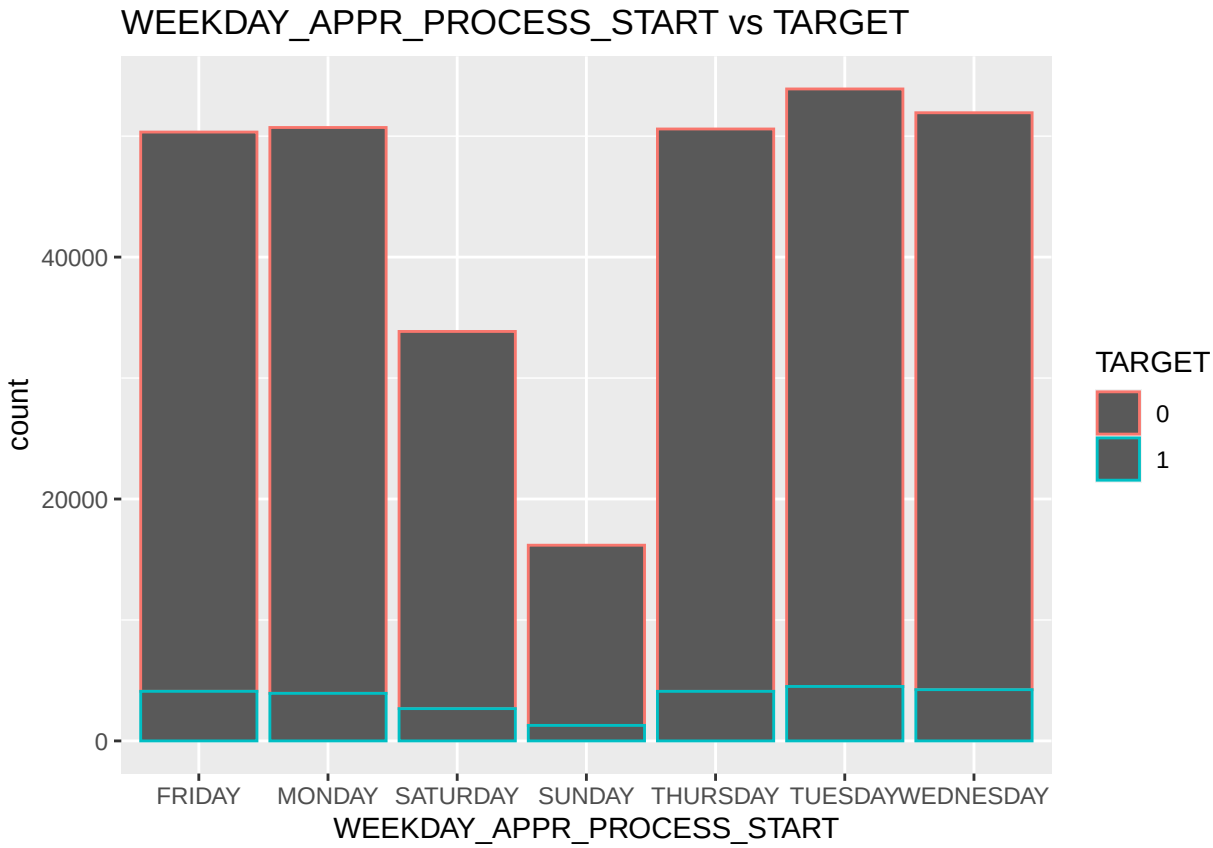

```
#WEEKDAY_APPR_PROCESS_START No Large difference in groups.
clean_train %>%
  group_by(WEEKDAY_APPR_PROCESS_START,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'WEEKDAY_APPR_PROCESS_START'. You can
## override using the `.groups` argument.
```

```
## # A tibble: 14 x 4
## # Groups:   WEEKDAY_APPR_PROCESS_START [7]
##    WEEKDAY_APPR_PROCESS_START TARGET     n  freq
##    <fct>                      <fct>  <int> <dbl>
##  1 FRIDAY                     0      46237 91.9
##  2 FRIDAY                     1       4101  8.15
##  3 MONDAY                     0      46780 92.2
##  4 MONDAY                     1       3934  7.76
##  5 SATURDAY                   0      31182 92.1
##  6 SATURDAY                   1       2670  7.89
##  7 SUNDAY                     0      14898 92.1
##  8 SUNDAY                     1       1283  7.93
##  9 THURSDAY                   0      46493 91.9
## 10 THURSDAY                   1       4098  8.10
## 11 TUESDAY                    0      49400 91.6
```

```
## 12 TUESDAY                   1       4501  8.35
## 13 WEDNESDAY                 0      47696 91.8
## 14 WEDNESDAY                 1       4238  8.16
```

```
ggplot(data = clean_train, aes(x=WEEKDAY_APPR_PROCESS_START, color = TARGET)) + geom_bar() + labs(title
```



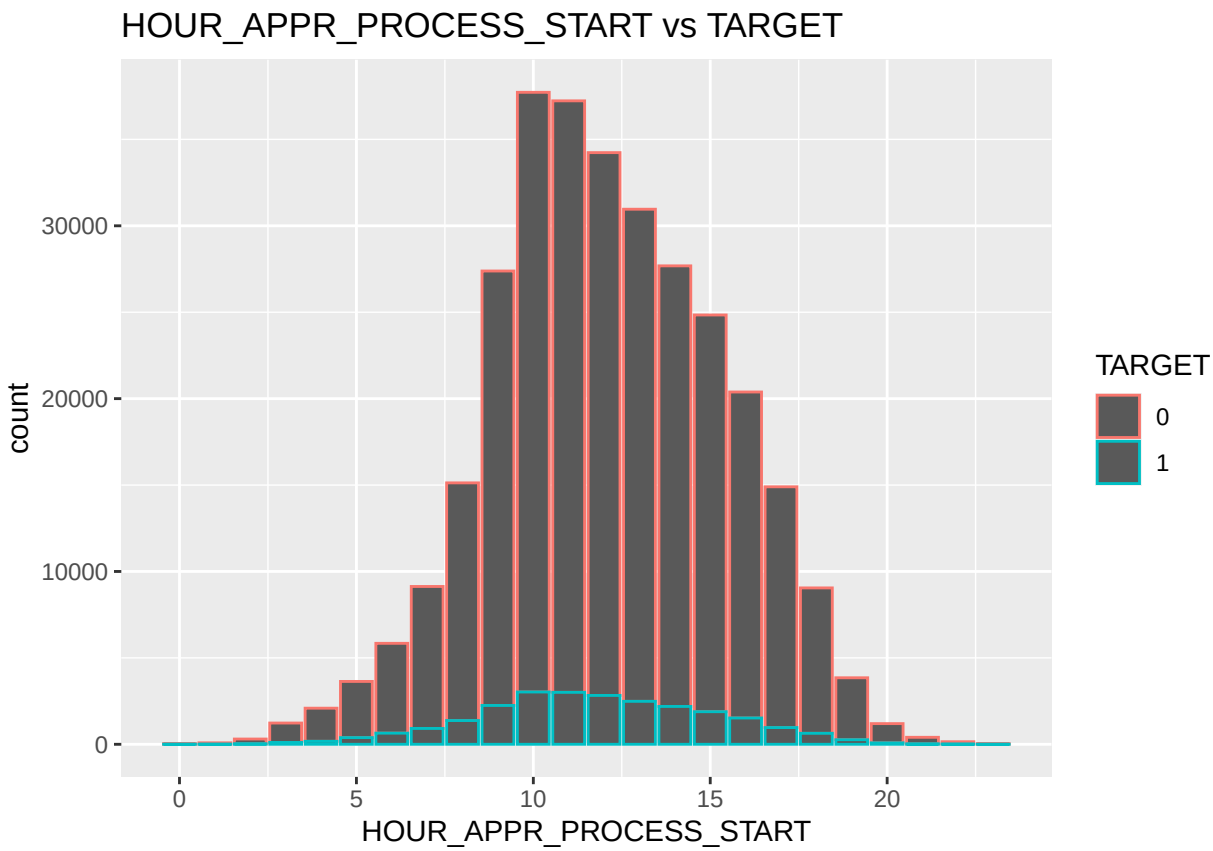

```
#HOUR_APPR_PROCESS_START Higher default rates in the early morning and late evening but much smaller gr
clean_train %>%
  group_by(HOUR_APPR_PROCESS_START,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'HOUR_APPR_PROCESS_START'. You can override
## using the `.groups` argument.
```

```
## # A tibble: 48 x 4
## # Groups:   HOUR_APPR_PROCESS_START [24]
##    HOUR_APPR_PROCESS_START TARGET     n  freq
##                      <int> <fct>  <int> <dbl>
##  1                       0 0         34 85
##  2                       0 1          6 15
##  3                       1 0         79 91.9
##  4                       1 1          7  8.14
##  5                       2 0        275 90.2
```

```
##  6                      2 1         30  9.84
##  7                      3 0       1123 91.3
##  8                      3 1        107  8.70
##  9                      4 0       1917 91.7
## 10                      4 1        173  8.28
## 11                      5 0       3253 89.4
## 12                      5 1        385 10.6
## 13                      6 0       5197 89.0
## 14                      6 1        645 11.0
## 15                      7 0       8214 90.0
## 16                      7 1        917 10.0
## 17                      8 0      13754 90.9
## 18                      8 1       1373  9.08
## 19                      9 0      25137 91.8
## 20                      9 1       2247  8.21
## 21                     10 0      34696 92.0
## 22                     10 1       3026  8.02
## 23                     11 0      34223 91.9
## 24                     11 1       3006  8.07
## 25                     12 0      31406 91.7
## 26                     12 1       2827  8.26
## 27                     13 0      28474 92.0
## 28                     13 1       2485  8.03
## 29                     14 0      25493 92.1
## 30                     14 1       2189  7.91
## 31                     15 0      22953 92.4
## 32                     15 1       1886  7.59
## 33                     16 0      18864 92.5
## 34                     16 1       1521  7.46
## 35                     17 0      13933 93.5
## 36                     17 1        967  6.49
## 37                     18 0       8414 93.0
## 38                     18 1        635  7.02
## 39                     19 0       3584 93.1
## 40                     19 1        264  6.86
## 41                     20 0       1112 93.0
## 42                     20 1         84  7.02
## 43                     21 0        380 93.8
## 44                     21 1         25  6.17
## 45                     22 0        135 90
## 46                     22 1         15 10
## 47                     23 0         36 87.8
## 48                     23 1          5 12.2
```

```
ggplot(data = clean_train, aes(x=HOUR_APPR_PROCESS_START, color = TARGET)) + geom_bar() + labs(title =
```

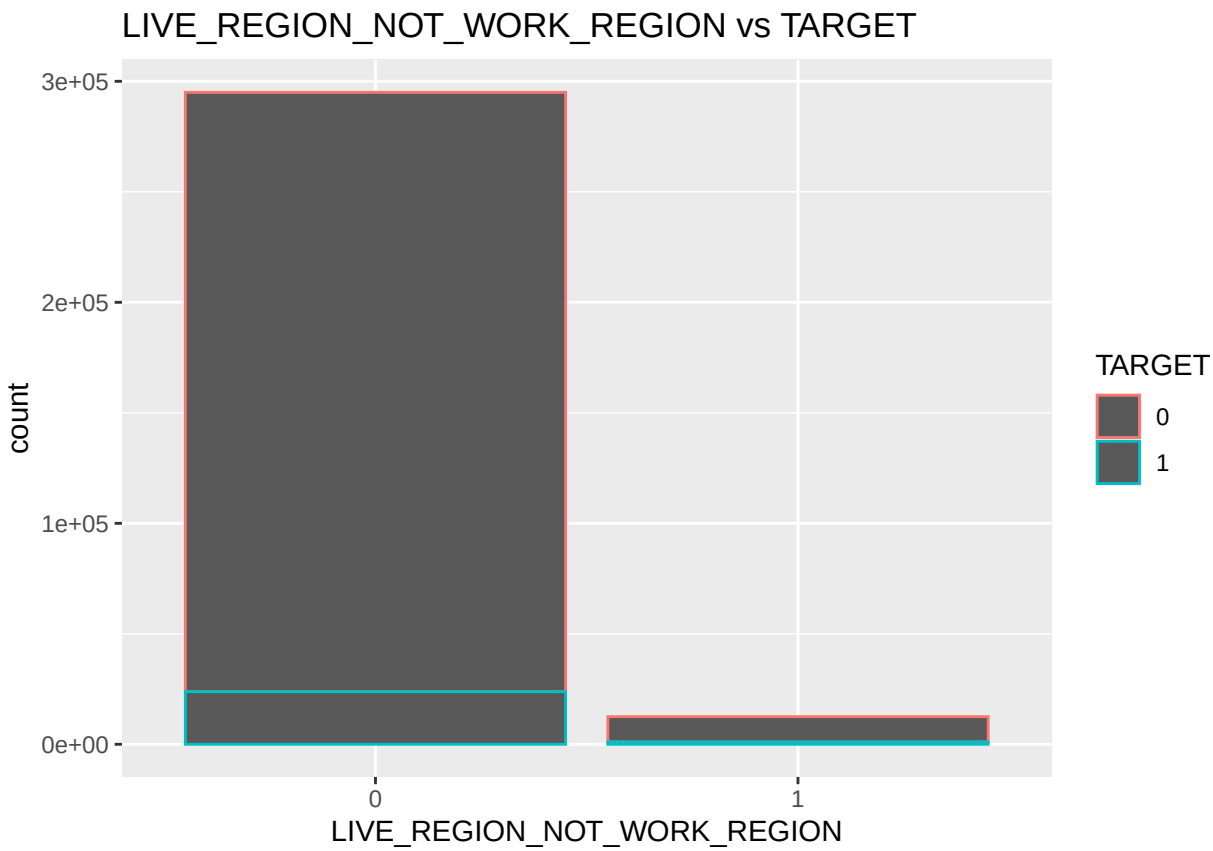



```
#LIVE_REGION_NOT_WORK_REGION No large difference in groups
clean_train %>%
  group_by(LIVE_REGION_NOT_WORK_REGION,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'LIVE_REGION_NOT_WORK_REGION'. You can
## override using the `.groups` argument.
```

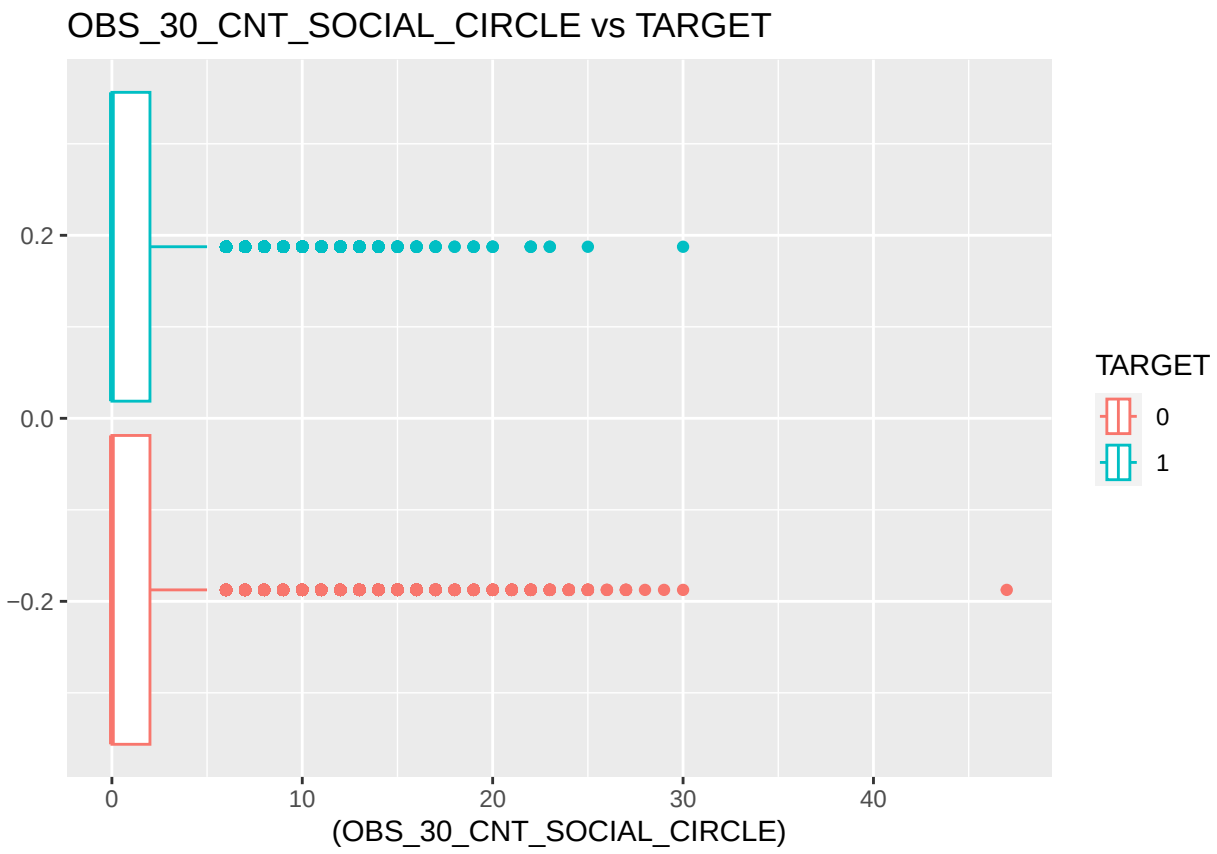
```
## # A tibble: 4 x 4
## # Groups:   LIVE_REGION_NOT_WORK_REGION [2]
##   LIVE_REGION_NOT_WORK_REGION TARGET      n  freq
##   <fct>                       <fct>   <int> <dbl>
## 1 0                           0      271239 91.9
## 2 0                           1       23769  8.06
## 3 1                           0       11447 91.6
## 4 1                           1        1056  8.45
```

```
ggplot(data = clean_train, aes(x=LIVE_REGION_NOT_WORK_REGION, color = TARGET)) + geom_bar() + labs(titl
```



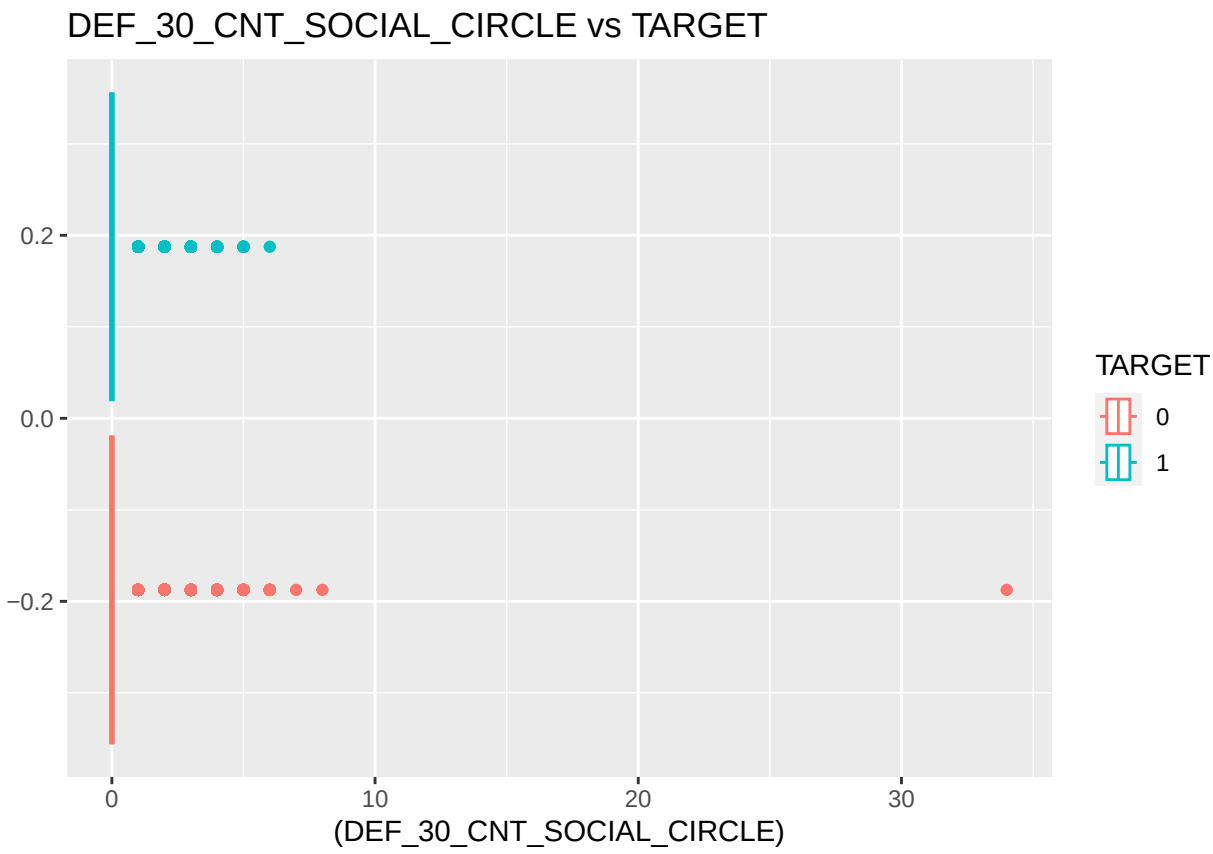

```
#OBS_30_CNT_SOCIAL_CIRCLE No difference in Means even with removed outliars
ggplot(data = SCtest, aes(x=(OBS_30_CNT_SOCIAL_CIRCLE), color = TARGET)) + geom_boxplot() + labs(title =
```

```
## Warning: Removed 1021 rows containing non-finite values (‘stat_boxplot()‘).
```



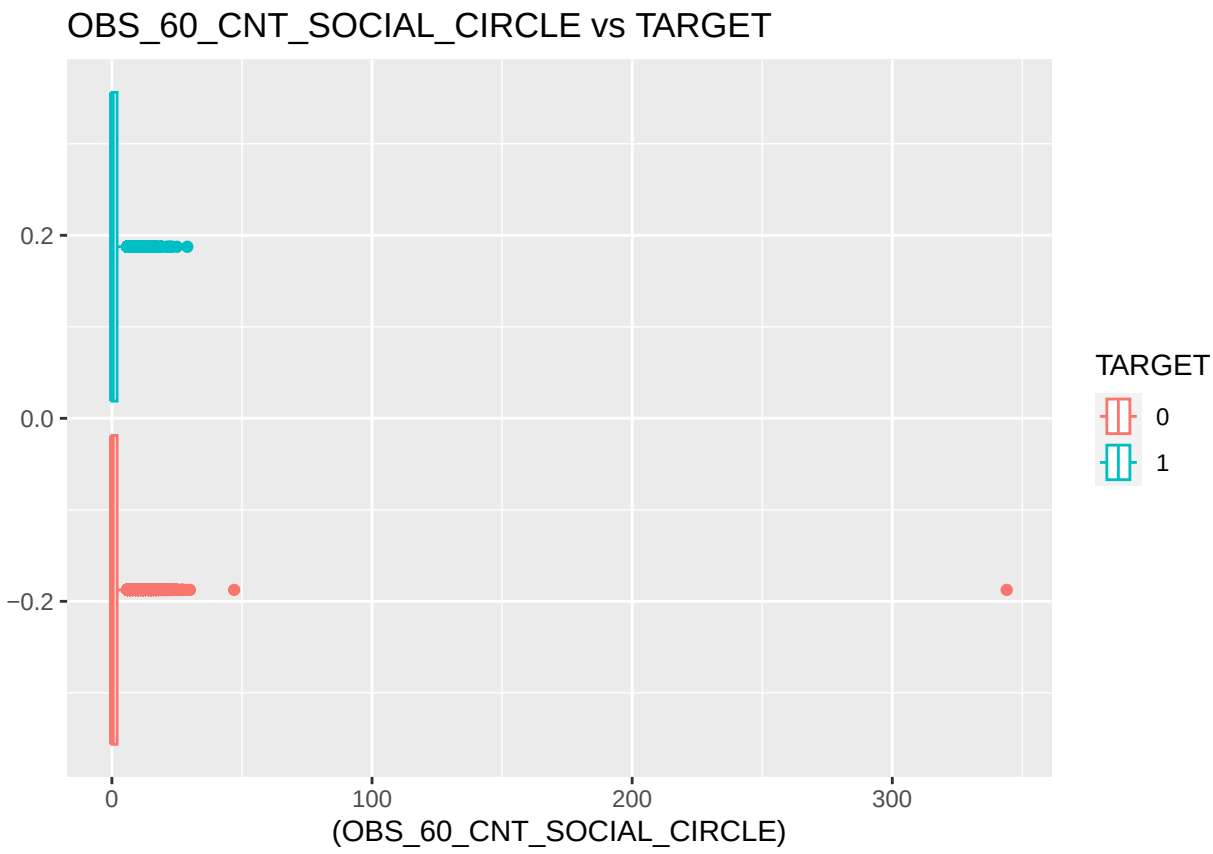

```
#DEF_30_CNT_SOCIAL_CIRCLE No difference in Means even with removed outliars
ggplot(data = clean_train, aes(x=(DEF_30_CNT_SOCIAL_CIRCLE), color = TARGET)) + geom_boxplot() + labs(ti
```

```
## Warning: Removed 1021 rows containing non-finite values (‘stat_boxplot()‘).
```



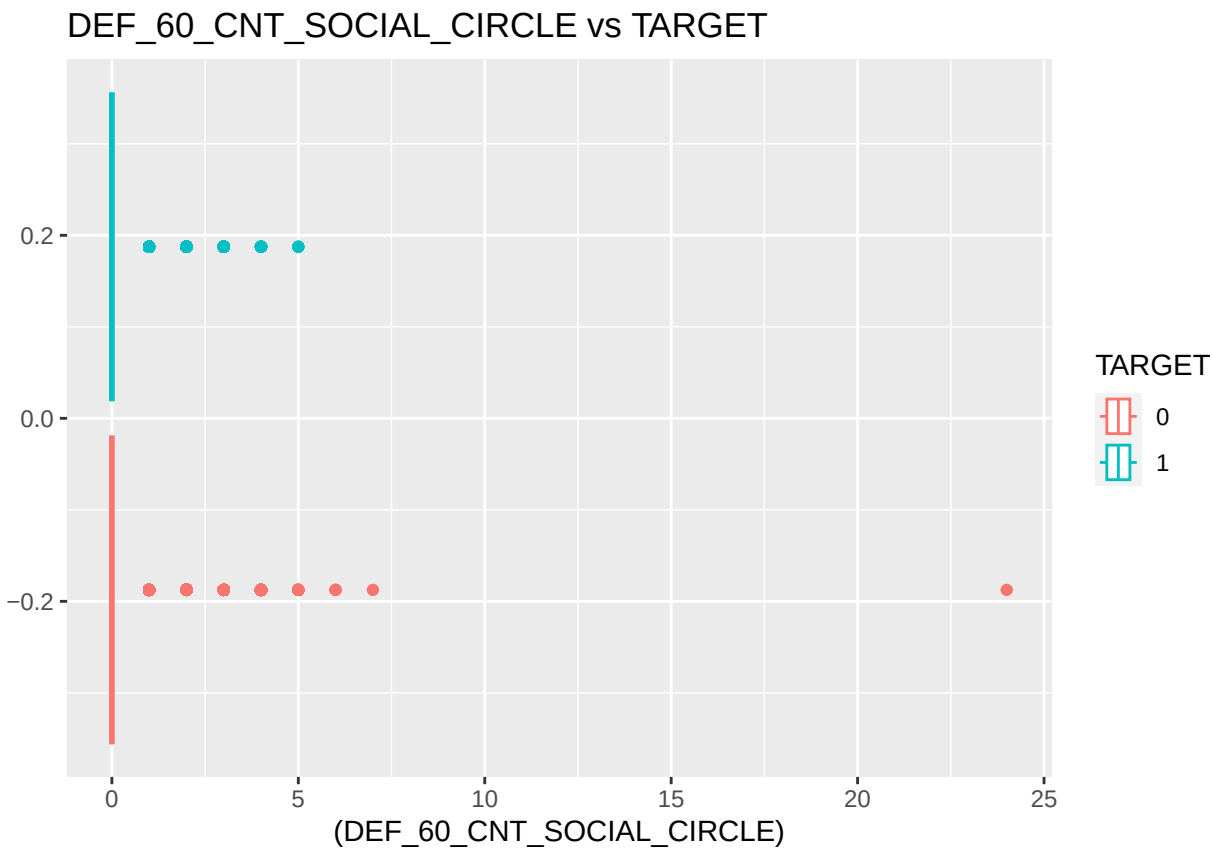

```
#OBS_60_CNT_SOCIAL_CIRCLE No difference in Means even with removed outliars
ggplot(data = clean_train, aes(x=(OBS_60_CNT_SOCIAL_CIRCLE), color = TARGET)) + geom_boxplot() + labs(ti
```

```
## Warning: Removed 1021 rows containing non-finite values (‘stat_boxplot()‘).
```




```
#DEF_60_CNT_SOCIAL_CIRCLE No difference in Means even with removed outliars
ggplot(data = clean_train, aes(x=(DEF_60_CNT_SOCIAL_CIRCLE), color = TARGET)) + geom_boxplot() + labs(ti
```

```
## Warning: Removed 1021 rows containing non-finite values (‘stat_boxplot()‘).
```

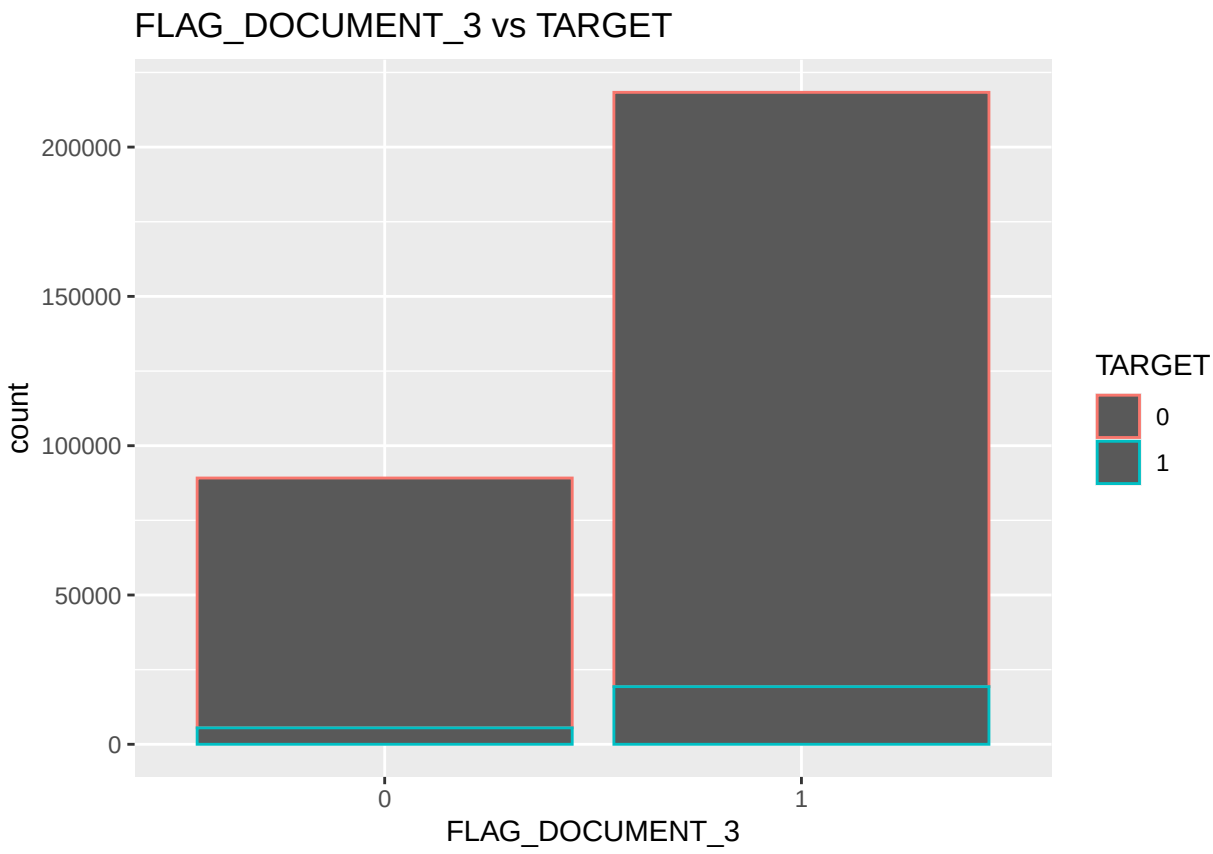



```
#FLAG_DOCUMENT_3 no large difference between groups
clean_train %>%
  group_by(FLAG_DOCUMENT_3,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'FLAG_DOCUMENT_3'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 4 x 4
## # Groups:   FLAG_DOCUMENT_3 [2]
##   FLAG_DOCUMENT_3 TARGET      n  freq
##   <fct>           <fct>   <int> <dbl>
## 1 0               0       83658 93.8
## 2 0               1        5513  6.18
## 3 1               0      199028 91.2
## 4 1               1       19312  8.84
```

```
ggplot(data = clean_train, aes(x=FLAG_DOCUMENT_3, color = TARGET)) + geom_bar() + labs(title = "FLAG_DO
```

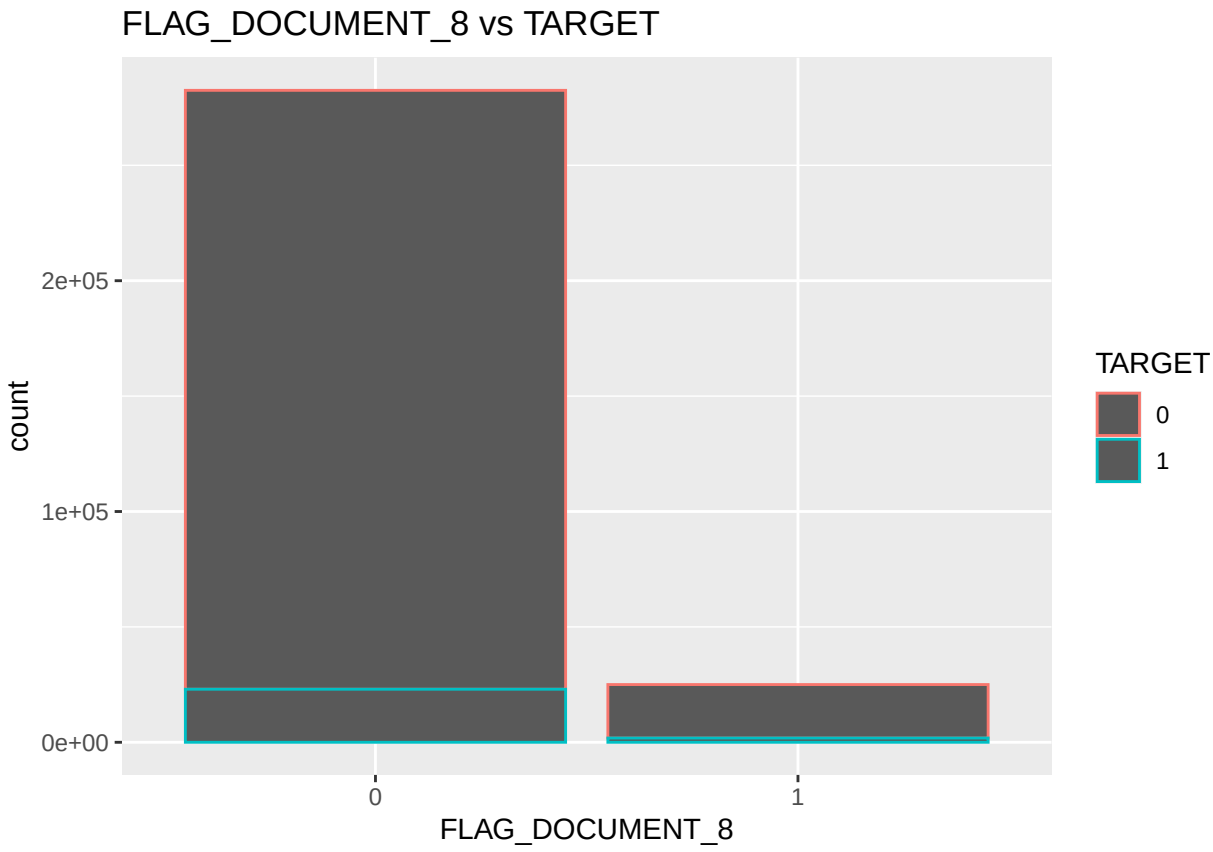



```
#FLAG_DOCUMENT_8 no large difference between default in those who provided the doc. possible
clean_train %>%
  group_by(FLAG_DOCUMENT_8,TARGET) %>%
    summarise(n=n()) %>%
    mutate(freq = (n/ sum(n)*100)) %>%
    print( n = 50)
```

```
## `summarise()` has grouped output by 'FLAG_DOCUMENT_8'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 4 x 4
## # Groups:   FLAG_DOCUMENT_8 [2]
##   FLAG_DOCUMENT_8 TARGET      n  freq
##   <fct>           <fct>   <int> <dbl>
## 1 0               0      259498 91.9
## 2 0               1       22989  8.14
## 3 1               0       23188 92.7
## 4 1               1        1836  7.34
```

```
ggplot(data = clean_train, aes(x=FLAG_DOCUMENT_8, color = TARGET)) + geom_bar() + labs(title = "FLAG_DO
```



EDA RESULTS

In this EDA notebook we have made great strides in understanding and preparing our data for modeling. We found our target variable has a 92% non default to 08% default rate. Our definition for success will be out performing this baseline. We were able to address many of the potential issues with the data around how to handle N/As and blanks. For these a suggestion of anything with 35% or more NAs or Blanks be removed from model consideration (49 total variables suggested). We also addressed all variables between 35% and 0% NAs or blanks and provided suggestions for each. We used a low variance filter to suggest and variables with less than 5% variability be removed from model consideration (35 total variables suggested). We then made suggestions for potential errors and outliars within the data. We looked at the 6 other provided data sources and provided 2 examples of how we could use these data sets as well as other suggestions for the modeling phase. Lastly we identified 10 potential strong predictors and 16 moderate predictors from our remaining variables. Though many of this work was more subjective than I would have liked it, I think that this exercise has yielded many great observations into the HomeCredit datasets and set us up for success in our upcoming modeling stage.