Home Credit Exploratory Data Analysis

Whitney Holt

2024-11-25

Contents

Introduction

Home Credit is an international consumer finance provider focused on responsibly lending to people with little to no credit history. To continue serving the unbanked, the company needs to confidently and accurately predict which prospective borrowers are likely to repay loans. Accurate loan repayment predictions enable Home Credit to foster financial inclusion while safeguarding the necessary enterprise profitability to sustain its mission.

The purpose of this project is to create a model to accurately predict which prospective borrowers are likely to repay loans. The specific target variable we will be predicting is called “target”, and represents each client’s ability to repay a loan (1 - client with payment difficulties: he/she had late payment more than X days on at least one of the first Y installments of the loan in our sample, 0 - all other cases).

The purpose of this exploratory data analysis (EDA) is to:

- Understand what data is available for the project
- Understand the scope of missing data and propose solutions
- Identify patterns within the available data and characteristics of each variable
- Understand relationships between variables

Questions about the data to explore:

- Is the data unbalanced with respect to the target?
- What would the accuracy be for a simple model consisting in a majority class classifier?
- Are there strong predictors that could be included later in a model?
- Which variables have missing data?
- What is the best solution for each variable with missing data?
- Do the values make sense? Are there mistaken values that should be cleaned or imputed?
- Are there columns with near-zero or zero variance?
- Will the input data need to be transformed in order to be used in a model?

Description of available data

Discuss the data available for the project.

```r
# Loading the data dictionary
HomeCredit_data_dictionary <- read.csv("HomeCredit_columns_description.csv")
```

```r
# Counting the number of columns in each data set
HomeCredit_data_dictionary %>%
  group_by(Table) %>%
  summarize(count = n())
```

```
## # A tibble: 7 x 2
##   Table                         count
##   <chr>                         <int>
## 1 POS_CASH_balance.csv              8
## 2 application_{train|test}.csv    122
## 3 bureau.csv                       17
## 4 bureau_balance.csv                3
## 5 credit_card_balance.csv          23
## 6 installments_payments.csv         8
## 7 previous_application.csv         38
```

There are 206 predictor variables available across the 7 available data sets:

- **120 predictors** in the **application train | test** data set (excluding ID and target variables: *SK_ID_CURR*, *TARGET*)
- **15 predictors** in the **bureau** data set (excluding ID variables: *SK_ID_CURR*, *SK_BUREAU_ID*)
- **2 predictors** in the **bureau balance** data set (excluding ID variables: *SK_BUREAU_ID*)
- **6 predictors** in the **POS CASH balance** data set (excluding ID variables: *SK_ID_PREV*, *SK_ID_CURR*)
- **21 predictors** in the **credit card balance** data set (excluding ID variables: *SK_ID_PREV*, *SK_ID_CURR*)
- **36 predictors** in the **previous application** data set (excluding ID variables: *SK_ID_PREV*, *SK_ID_CURR*)

- **6 predictors** in the **installments payments** data set (excluding ID variables: *SK_ID_PREV*, *SK_ID_CURR*)

The final model will likely not include all predictors from all available data sets. Some data sets are provided at various levels of granularity and will potentially be excluded for simplicity's sake.

Data Exploration

Starting with and potentially focusing on the application_{train|test}.csv data sets.

Loading the application_{train|test}.csv data sets:

```
# Loading the application train set
HomeCredit_application_train_data <- read.csv("application_train.csv")
```

Target Variable

Exploring the target variable in application_{train|test}.csv.

Questions of interest:

- Is the data unbalanced with respect to the target?
- What would the accuracy be for a simple model consisting in a majority class classifier?

```
# Creating a balance table of the target variable
HomeCredit_application_train_data %>%
  group_by(TARGET) %>%
  summarise(n  = n(),
            proportion = n / nrow(HomeCredit_application_train_data)) %>%
  round(digits = 2)
```

```
## # A tibble: 2 x 3
##   TARGET      n proportion
##    <dbl>  <dbl>      <dbl>
## 1      0 282686       0.92
## 2      1  24825       0.08
```

The data is **highly imbalanced** with respect to the target. A majority class classifier would have an accuracy of 92%.

Missing Data

Questions of interest:

- What is the scope of missing data in application_{train|test}.csv?
- What are possible solutions?
- Which solutions should be applied to which columns?

Scope of Missing Data

How many columns in application_{train|test}.csv have missing data?

```r
# Identifying columns with missing data in the train data
missing_values <- HomeCredit_application_train_data %>%
  summarise(across(everything(), ~ sum(is.na(.)))) %>%
  pivot_longer(everything(), names_to = "column",
               values_to = "missing_count") %>%
  filter(missing_count > 0) %>%
  arrange(desc(missing_count))

missing_values
```

```
## # A tibble: 61 x 2
##    column                   missing_count
##    <chr>                            <int>
##  1 COMMONAREA_AVG                  214865
##  2 COMMONAREA_MODE                 214865
##  3 COMMONAREA_MEDI                 214865
##  4 NONLIVINGAPARTMENTS_AVG         213514
##  5 NONLIVINGAPARTMENTS_MODE        213514
##  6 NONLIVINGAPARTMENTS_MEDI        213514
##  7 LIVINGAPARTMENTS_AVG            210199
##  8 LIVINGAPARTMENTS_MODE           210199
##  9 LIVINGAPARTMENTS_MEDI           210199
## 10 FLOORSMIN_AVG                   208642
## # i 51 more rows
```

The application_train.csv data set has missing data in 61 of the 122 columns.

Possible Solutions for Columns with Missing Data

Creating a new data frame, HomeCredit_application_train_data_clean to store cleaned variables in alongside variables that don't need cleaning while maintaining the integrity of the raw data.

```r
# Creating a new data frame, HomeCredit_application_train_data_clean
HomeCredit_application_train_data_clean <- HomeCredit_application_train_data
```

AMT_ANNUITY

AMT_ANNUITY is the loan annuity value.

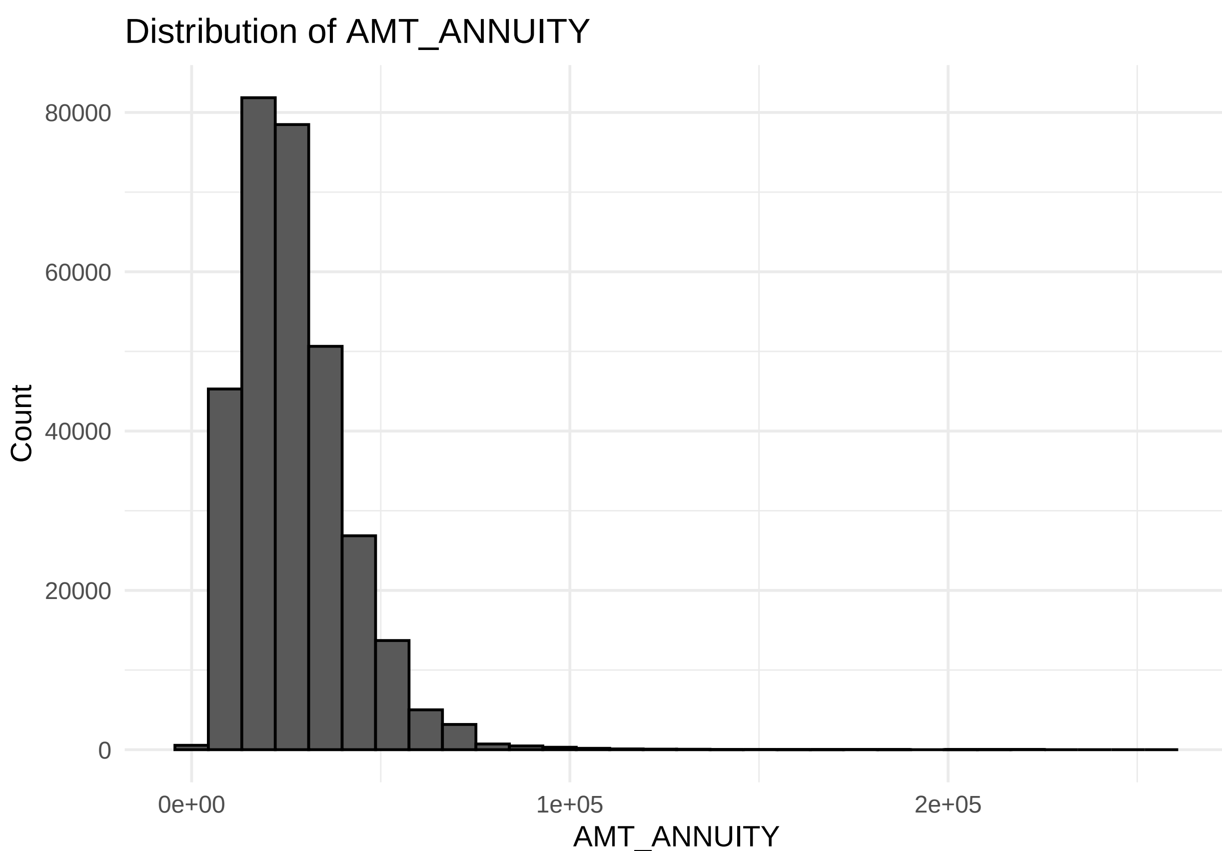
```r
# Viewing the distribution of of AMT_ANNUITY
summary(HomeCredit_application_train_data_clean$AMT_ANNUITY)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1616   16524   24903   27109   34596  258026      12
```

```r
ggplot(HomeCredit_application_train_data_clean, aes(x = AMT_ANNUITY)) +
  geom_histogram(color = "black") +
  labs(title = "Distribution of AMT_ANNUITY",
       x = "AMT_ANNUITY",
       y = "Count") +
  theme_minimal()
```

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

```
## Warning: Removed 12 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



Assumptions & Approach:

- This is a continuous, numeric variable representing the loan annuity value
- There are few (12) missing values in the training data set
- Since this should have been reported for every participant, we will impute missing values using the median since the data is skewed

```r
# Imputing missing values in AMT_ANNUITY using the Median
HomeCredit_application_train_data_clean <-
  HomeCredit_application_train_data_clean %>%
  mutate(AMT_ANNUITY = if_else(is.na(AMT_ANNUITY),
                               median(AMT_ANNUITY, na.rm = TRUE),
```

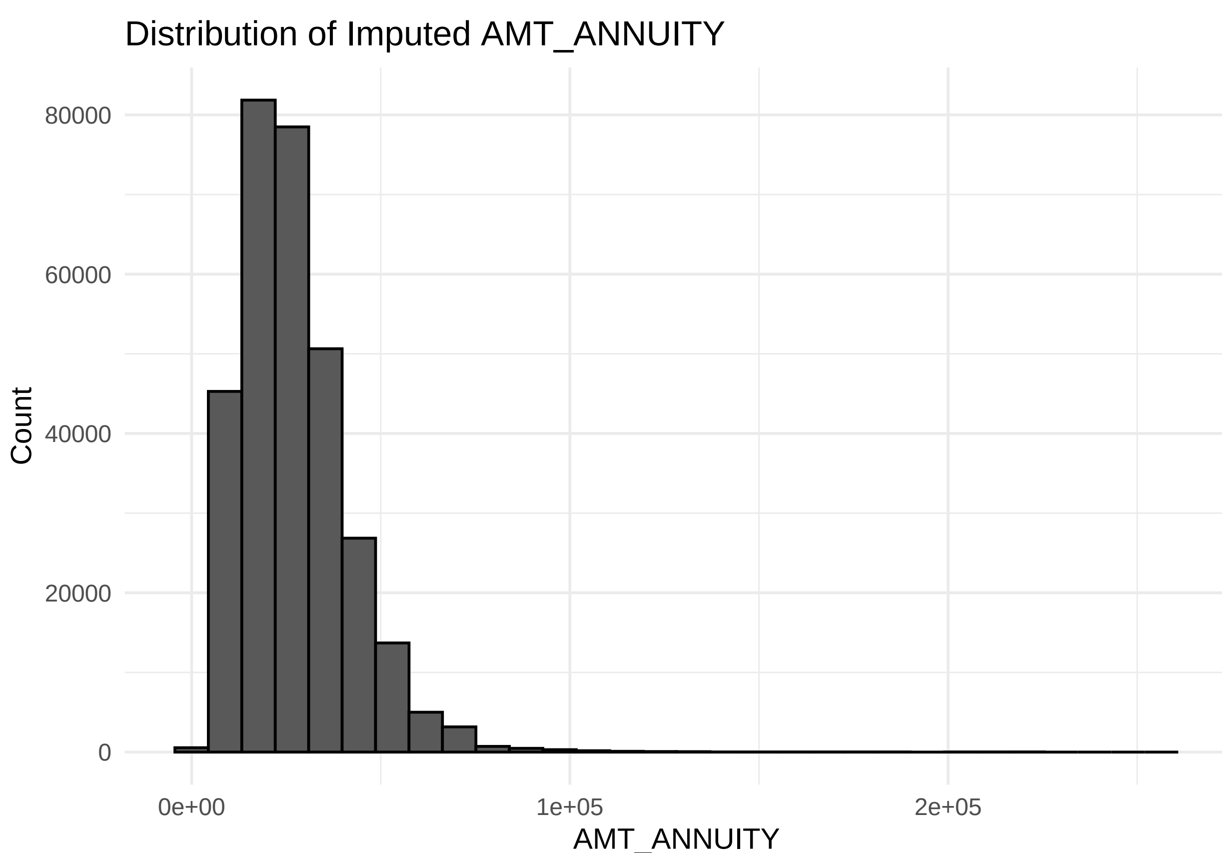
```
                                AMT_ANNUITY))

# Viewing the distribution of of AMT_ANNUITY after imputing
summary(HomeCredit_application_train_data_clean$AMT_ANNUITY)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1616   16524   24903   27108   34596  258026
```

```
ggplot(HomeCredit_application_train_data_clean, aes(x = AMT_ANNUITY)) +
  geom_histogram(color = "black") +
  labs(title = "Distribution of Imputed AMT_ANNUITY",
       x = "AMT_ANNUITY",
       y = "Count") +
  theme_minimal()
```

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



The distribution of AMT_ANNUITY after imputing looks very similar to the variable's distribution prior to imputing.

AMT_GOODS_PRICE

AMT_GOODS_PRICE is, for consumer loans, the price of the goods for which the loan is given.

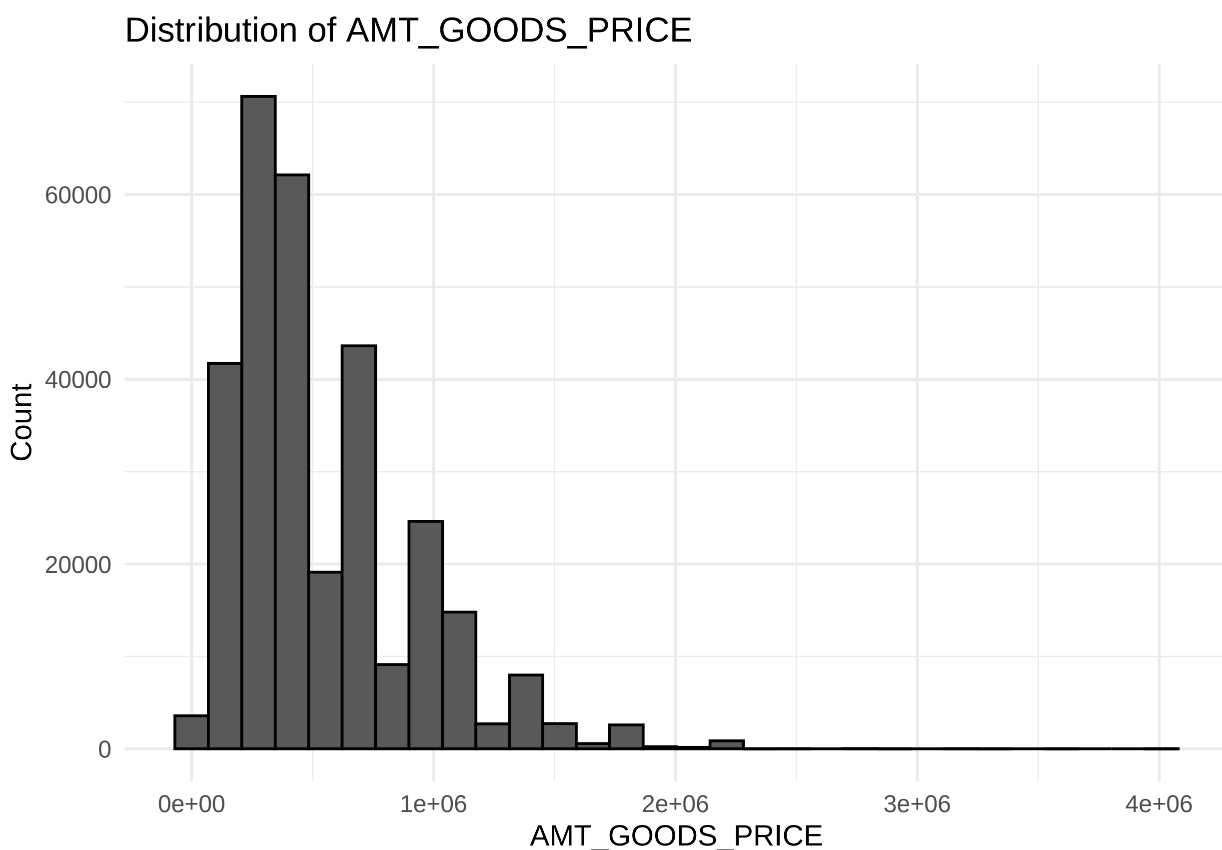
```r
# Viewing the distribution of of AMT_GOODS_PRICE
summary(HomeCredit_application_train_data_clean$AMT_GOODS_PRICE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   40500  238500  450000  538396  679500 4050000     278
```

```r
ggplot(HomeCredit_application_train_data_clean, aes(x = AMT_GOODS_PRICE)) +
  geom_histogram(color = "black") +
  labs(title = "Distribution of AMT_GOODS_PRICE",
       x = "AMT_GOODS_PRICE",
       y = "Count") +
  theme_minimal()
```

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

```
## Warning: Removed 278 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



Are the missing values here for non-consumer loans?

```r
# Quering unique values of NAME_CONTRACT_TYPE where AMT_GOODS_PRICE is NA
HomeCredit_application_train_data_clean %>%
  filter(is.na(AMT_GOODS_PRICE)) %>%
  distinct(NAME_CONTRACT_TYPE)
```

```
##   NAME_CONTRACT_TYPE
## 1    Revolving loans
```

The missing values have a contract type that is not a consumer loan. In this case, all 278 missing values are revolving loans.

Assumptions & Approach:

- If there is no value for an individual, they had a non-consumer loan
- Since AMT_GOODS_PRICE is skewed, we'll take the log transform of the variable
- Bin the log transformed variable into "low", "low-medium", "medium", "medium-high", "high", and "non-consumer loan"

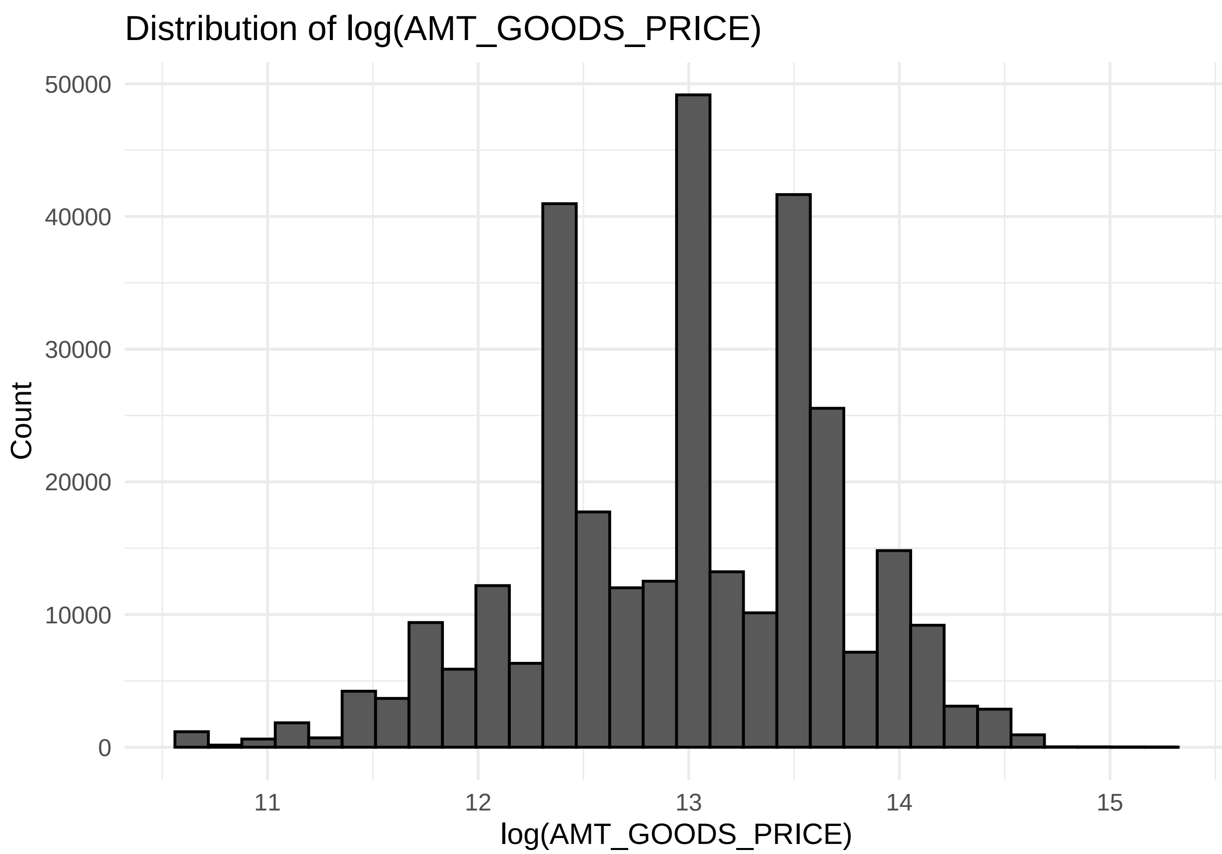
```r
# Viewing the distribution of of log(AMT_GOODS_PRICE)
summary(log(HomeCredit_application_train_data_clean$AMT_GOODS_PRICE))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   10.61   12.38   13.02   12.96   13.43   15.21     278
```

```r
ggplot(HomeCredit_application_train_data_clean,
       aes(x = log(AMT_GOODS_PRICE))) +
  geom_histogram(color = "black") +
  labs(title = "Distribution of log(AMT_GOODS_PRICE)",
       x = "log(AMT_GOODS_PRICE)",
       y = "Count") +
  theme_minimal()
```

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

```
## Warning: Removed 278 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



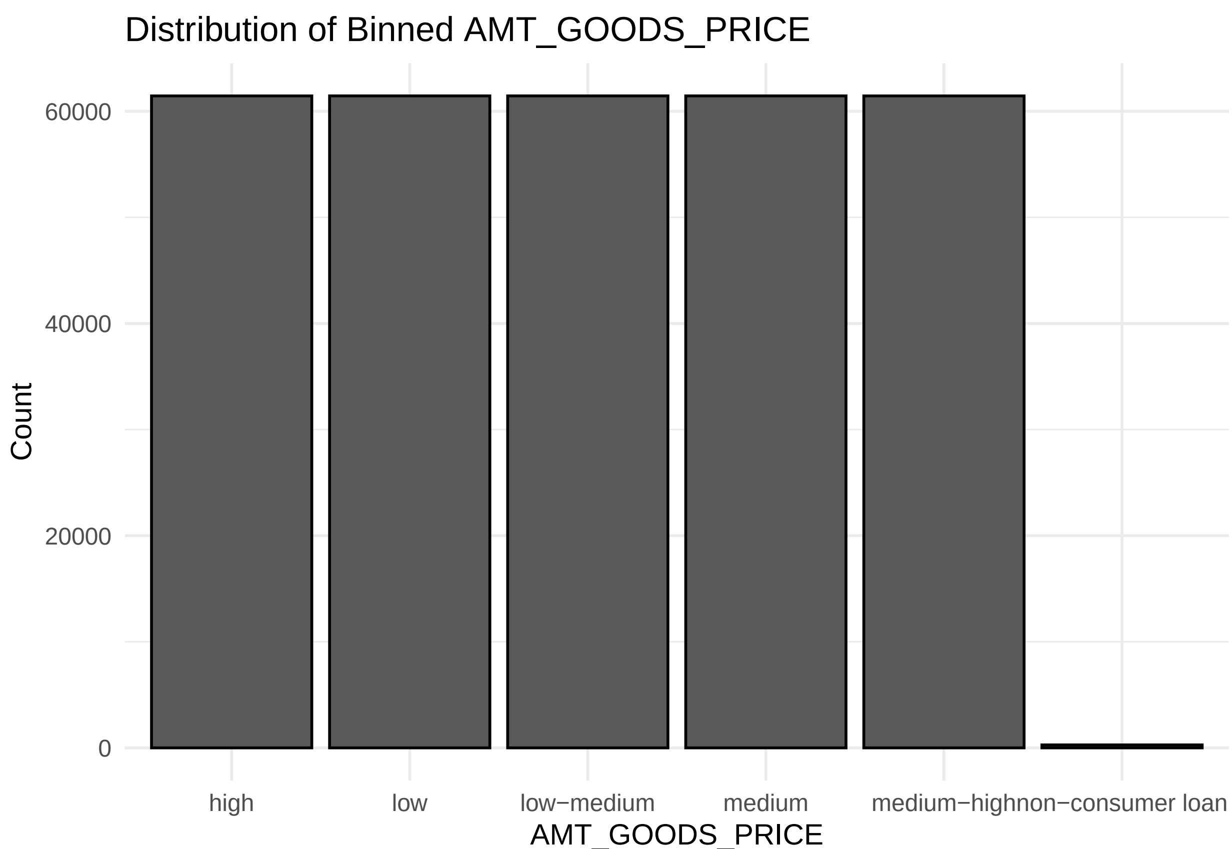
This the log transformed distribution looks much more normal, but appears to be multi-modal. We will move forward with binning the log transform of AMT_GOODS_PRICE.

```
# Binning the log transform of AMT_GOODS_PRICE into quintiles, keeping NAs as a separate class
HomeCredit_application_train_data_clean <-
  HomeCredit_application_train_data_clean %>%
  mutate(
    AMT_GOODS_PRICE = case_when(
      is.na(AMT_GOODS_PRICE) ~ "non-consumer loan",  # Handle missing values
      TRUE ~ case_when(
        ntile(log(AMT_GOODS_PRICE), 5) == 1 ~ "low",
        ntile(log(AMT_GOODS_PRICE), 5) == 2 ~ "low-medium",
        ntile(log(AMT_GOODS_PRICE), 5) == 3 ~ "medium",
        ntile(log(AMT_GOODS_PRICE), 5) == 4 ~ "medium-high",
        ntile(log(AMT_GOODS_PRICE), 5) == 5 ~ "high"
      )
    )
  )

# Viewing the distribution of of AMT_GOODS_PRICE after binning
summary(HomeCredit_application_train_data_clean$AMT_GOODS_PRICE)
```

```
##    Length     Class      Mode
##    307511 character character
```

```r
ggplot(HomeCredit_application_train_data_clean, aes(x = AMT_GOODS_PRICE)) +
  geom_bar(color = "black") +
  labs(title = "Distribution of Binned AMT_GOODS_PRICE",
       x = "AMT_GOODS_PRICE",
       y = "Count") +
  theme_minimal()
```



Existing AMT_GOODS_PRICE inputs have been binned into quintiles of their log-transformed value while the values that were previously missing have been categorized as non-consumer loans.

OWN_CAR_AGE

OWN_CAR_AGE is age of client's car.

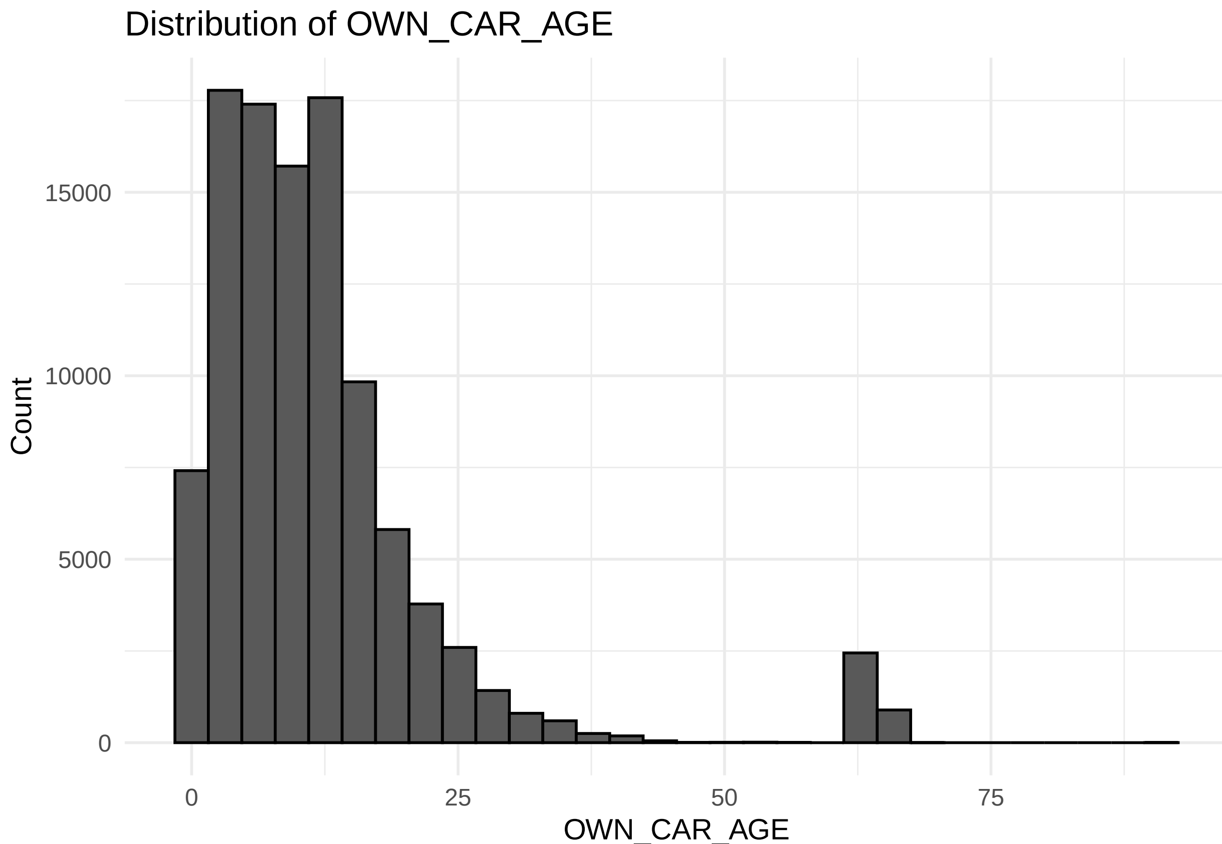
```r
# Viewing the distribution of of OWN_CAR_AGE
summary(HomeCredit_application_train_data_clean$OWN_CAR_AGE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    5.00    9.00   12.06   15.00   91.00  202929
```

```r
ggplot(HomeCredit_application_train_data_clean, aes(x = OWN_CAR_AGE)) +
  geom_histogram(color = "black") +
  labs(title = "Distribution of OWN_CAR_AGE",
       x = "OWN_CAR_AGE",
       y = "Count") +
  theme_minimal()
```

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

```
## Warning: Removed 202929 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



Assumptions & Approach:

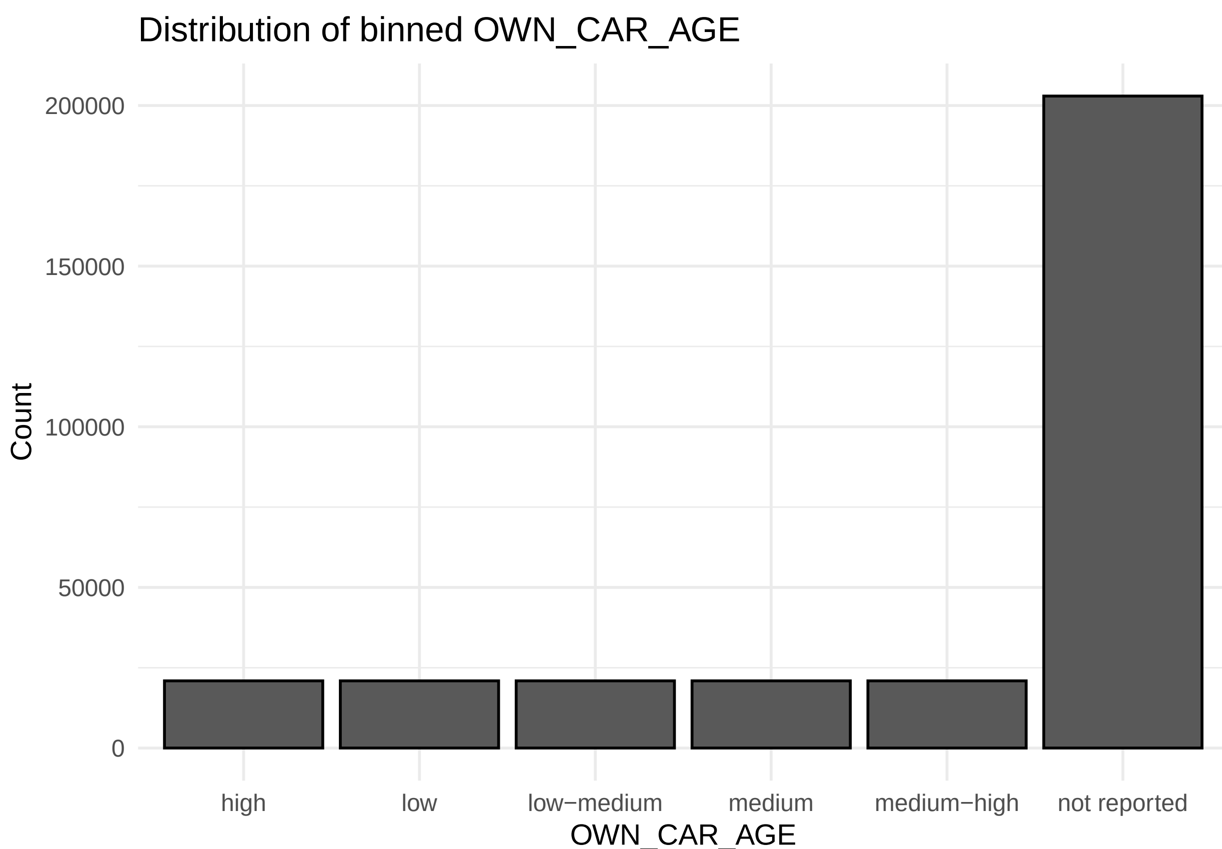
- If there is no value for an individual, we will assign them to the category "not reported"
- Bin variable into "low", "low-medium", "medium", "medium-high", "high", and "not reported"

```
# Binning the log transform of OWN_CAR_AGE into quintiles, keeping NAs as a separate class
HomeCredit_application_train_data_clean <-
  HomeCredit_application_train_data_clean %>%
  mutate(
    OWN_CAR_AGE = case_when(
      is.na(OWN_CAR_AGE) ~ "not reported",  # Handle missing values
      TRUE ~ case_when(
        ntile(OWN_CAR_AGE, 5) == 1 ~ "low",
        ntile(OWN_CAR_AGE, 5) == 2 ~ "low-medium",
        ntile(OWN_CAR_AGE, 5) == 3 ~ "medium",
        ntile(OWN_CAR_AGE, 5) == 4 ~ "medium-high",
        ntile(OWN_CAR_AGE, 5) == 5 ~ "high"
      )
    )
  )
```

```r
# Viewing the distribution of of AMT_GOODS_PRICE after binning
summary(HomeCredit_application_train_data_clean$OWN_CAR_AGE)
```

```
##    Length     Class      Mode 
##    307511 character character
```

```r
ggplot(HomeCredit_application_train_data_clean, aes(x = OWN_CAR_AGE)) +
  geom_bar(color = "black") +
  labs(title = "Distribution of binned OWN_CAR_AGE",
       x = "OWN_CAR_AGE",
       y = "Count") +
  theme_minimal()
```



Existing OWN_CAR_AGE inputs have been binned into quintiles while the values that were previously missing have been categorized as not reported. Over half of the data points did not report a value for OWN_CAR_AGE.

CNT_FAM_MEMBERS

CNT_FAM_MEMBERS is how many family members does client have.

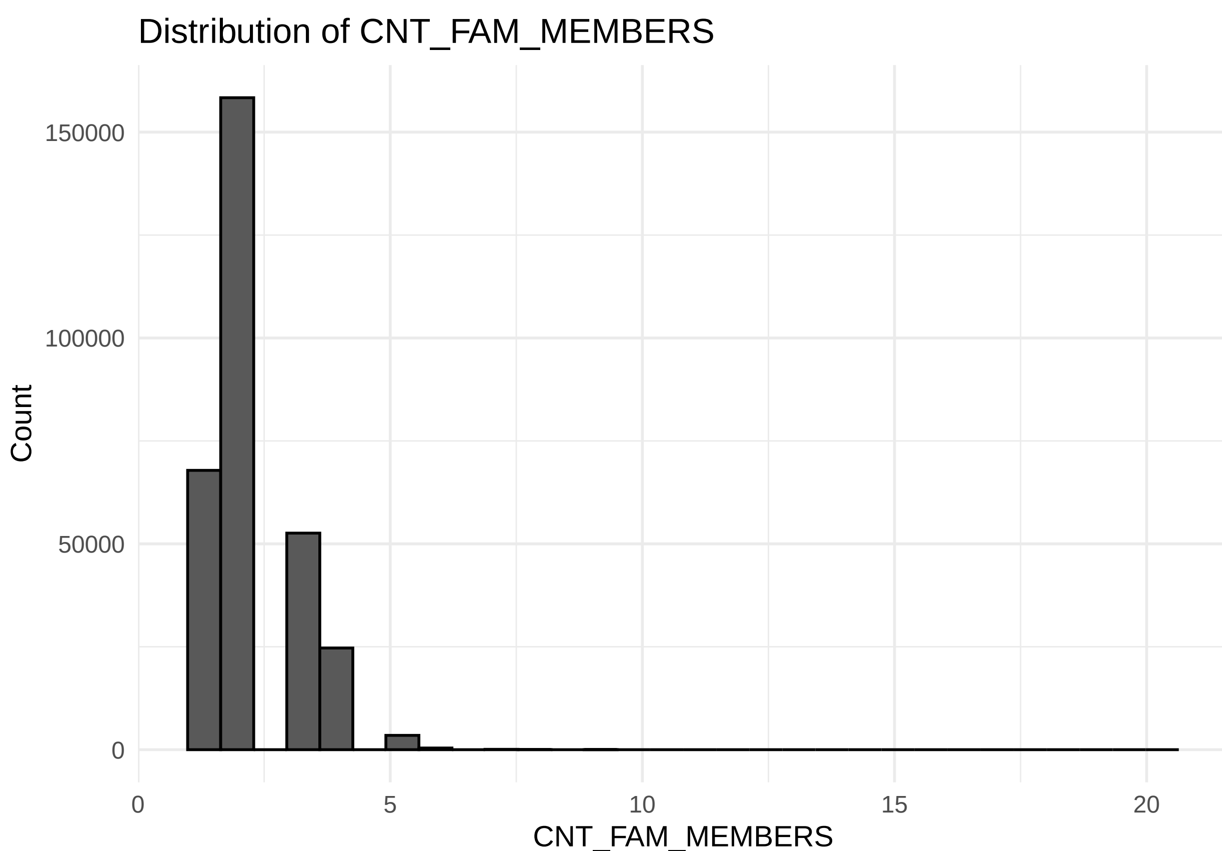
```r
# Viewing the distribution of of CNT_FAM_MEMBERS
summary(HomeCredit_application_train_data_clean$CNT_FAM_MEMBERS)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   1.000   2.000   2.000   2.153   3.000  20.000       2
```

```r
ggplot(HomeCredit_application_train_data_clean, aes(x = CNT_FAM_MEMBERS)) +
  geom_histogram(color = "black") +
  labs(title = "Distribution of CNT_FAM_MEMBERS",
       x = "CNT_FAM_MEMBERS",
       y = "Count") +
  theme_minimal()
```

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

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



Assumptions & Approach:

- Since the values range from 1 - 20, we'll assume that if there is no value for the individual, they have 0 family members
- Replace NAs with 0

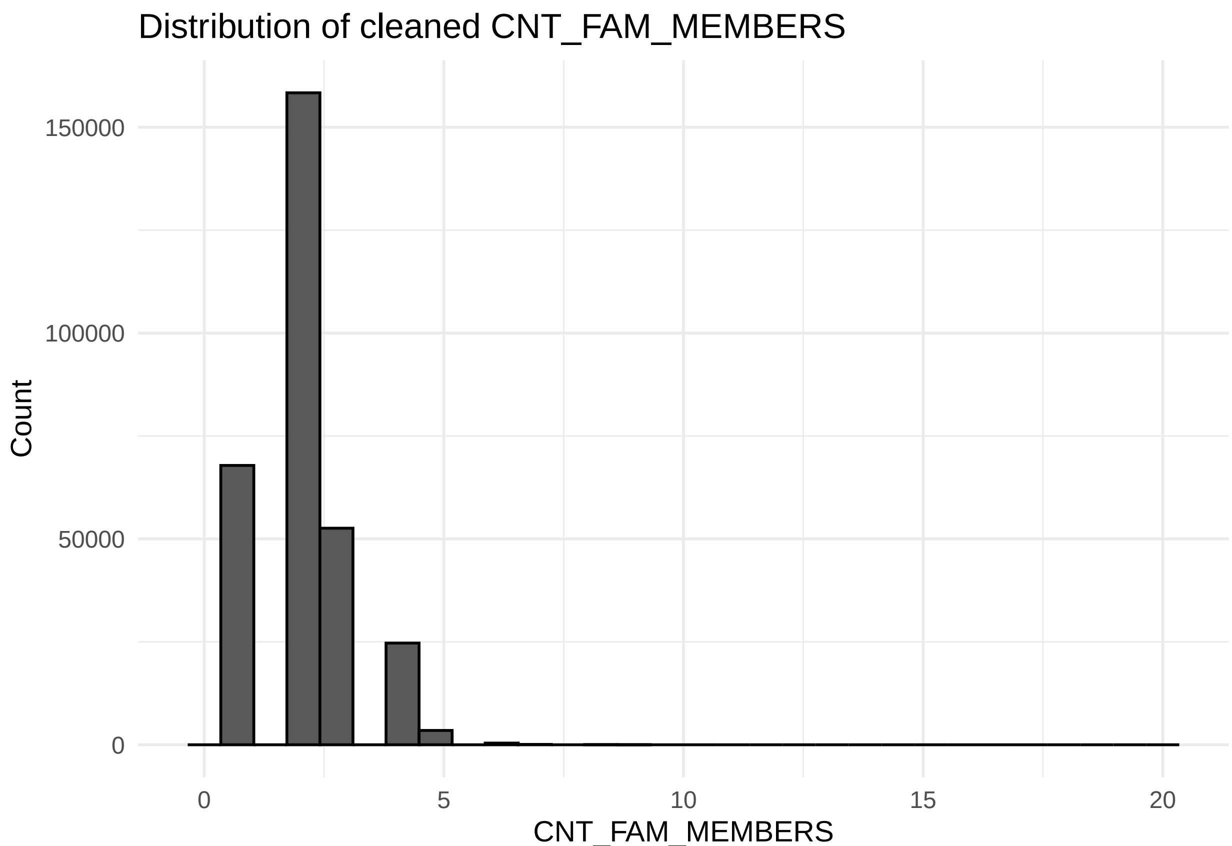
```r
# Replacing missing values in CNT_FAM_MEMBERS with 0
HomeCredit_application_train_data_clean <-
  HomeCredit_application_train_data_clean %>%
  mutate(CNT_FAM_MEMBERS = if_else(is.na(CNT_FAM_MEMBERS),
                                   0,
                                   CNT_FAM_MEMBERS))
```

```r
# Viewing the distribution of of CNT_FAM_MEMBERS after binning
summary(HomeCredit_application_train_data_clean$CNT_FAM_MEMBERS)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   2.000   2.153   3.000  20.000
```

```r
ggplot(HomeCredit_application_train_data_clean, aes(x = CNT_FAM_MEMBERS)) +
  geom_histogram(color = "black") +
  labs(title = "Distribution of cleaned CNT_FAM_MEMBERS",
       x = "CNT_FAM_MEMBERS",
       y = "Count") +
  theme_minimal()
```

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



Missing values in the CNT_FAM_MEMBERS column have been replaced with zeros, assuming the lack of input indicates the individual does not have any family members.

EXT_SOURCE variables

EXT_SOURCE_1, EXT_SOURCE_2, and EXT_SOURCE_3 are normalized scores from external data sources.

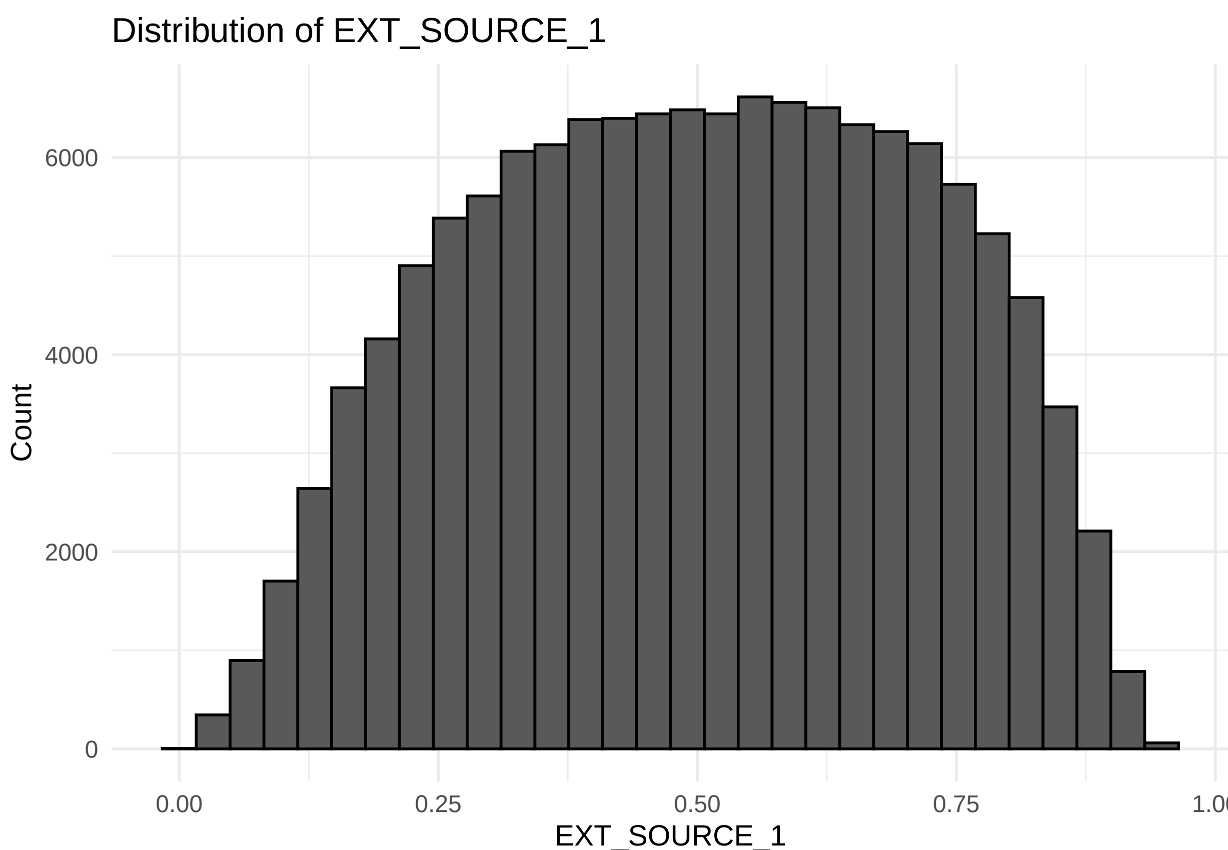
```r
# Viewing the distribution of of EXT_SOURCE_1
summary(HomeCredit_application_train_data_clean$EXT_SOURCE_1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.01    0.33    0.51    0.50    0.68    0.96  173378
```

```r
ggplot(HomeCredit_application_train_data_clean, aes(x = EXT_SOURCE_1)) +
  geom_histogram(color = "black") +
  labs(title = "Distribution of EXT_SOURCE_1",
       x = "EXT_SOURCE_1",
       y = "Count") +
  theme_minimal()
```

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

## Warning: Removed 173378 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



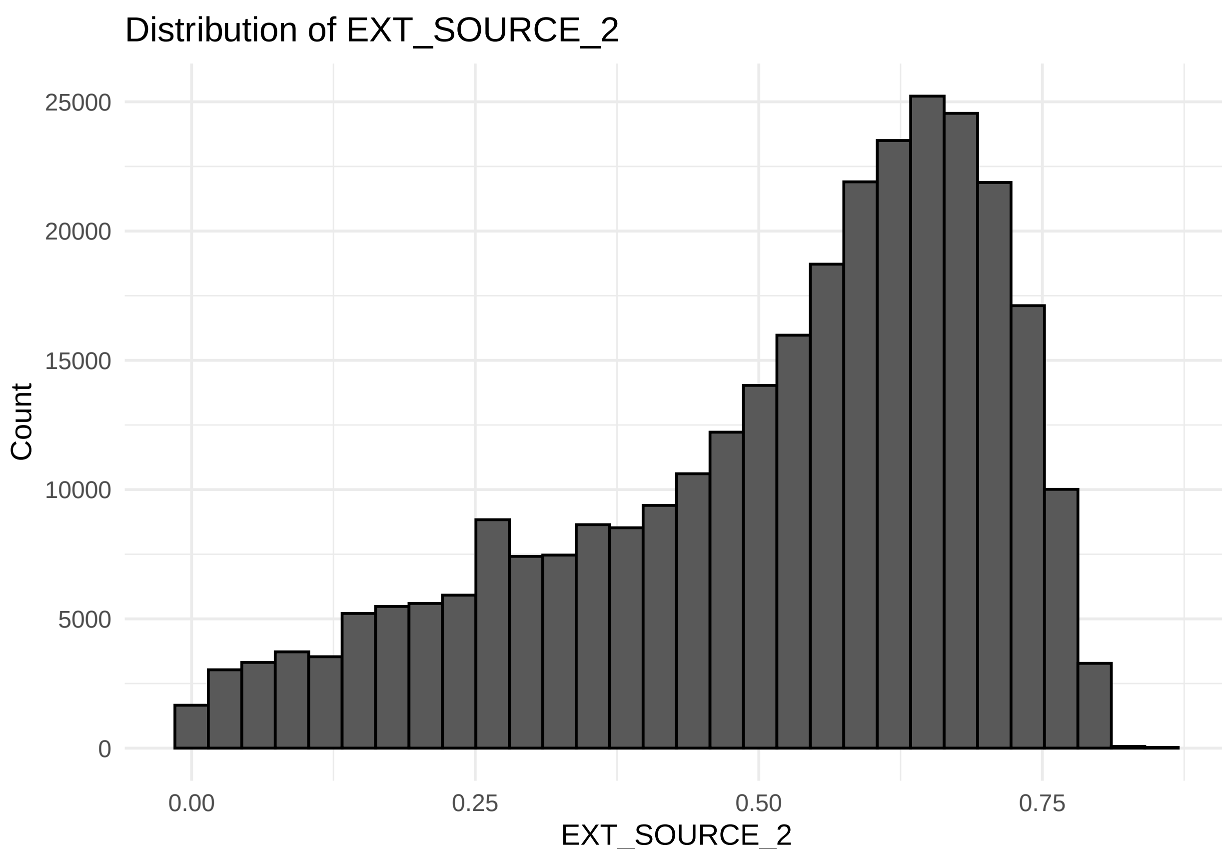

```r
# Viewing the distribution of of EXT_SOURCE_2
summary(HomeCredit_application_train_data_clean$EXT_SOURCE_2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.0000  0.3925  0.5660  0.5144  0.6636  0.8550     660
```

```r
ggplot(HomeCredit_application_train_data_clean, aes(x = EXT_SOURCE_2)) +
  geom_histogram(color = "black") +
  labs(title = "Distribution of EXT_SOURCE_2",
       x = "EXT_SOURCE_2",
       y = "Count") +
  theme_minimal()
```

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

```
## Warning: Removed 660 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



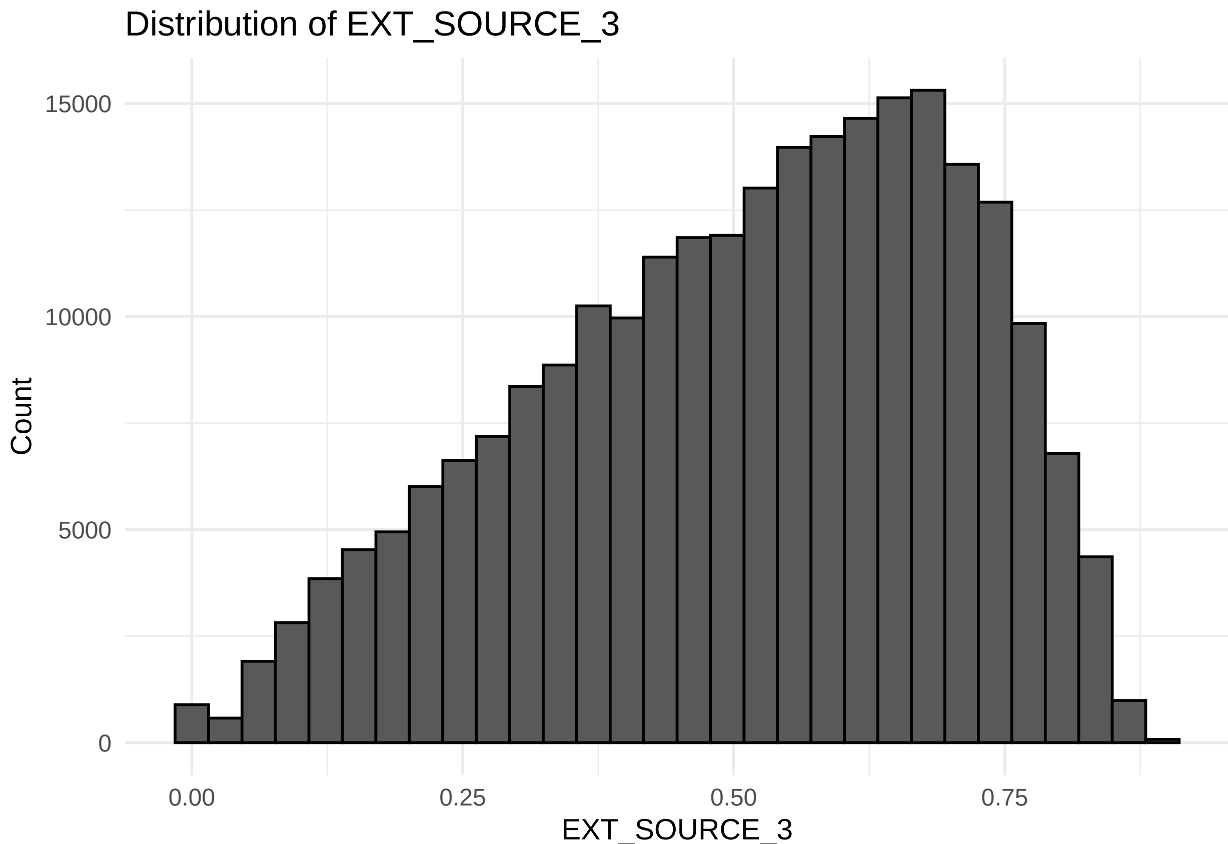

```r
# Viewing the distribution of of EXT_SOURCE_3
summary(HomeCredit_application_train_data_clean$EXT_SOURCE_3)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.37    0.54    0.51    0.67    0.90   60965
```

```r
ggplot(HomeCredit_application_train_data_clean, aes(x = EXT_SOURCE_3)) +
  geom_histogram(color = "black") +
  labs(title = "Distribution of EXT_SOURCE_3",
       x = "EXT_SOURCE_3",
       y = "Count") +
  theme_minimal()
```

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

## Warning: Removed 60965 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



Assumptions & Approach:

- If there is no value for an individual, they don't have a credit score meaning they haven't had a loan before
- Bin these scores, keeping a category for those without scores

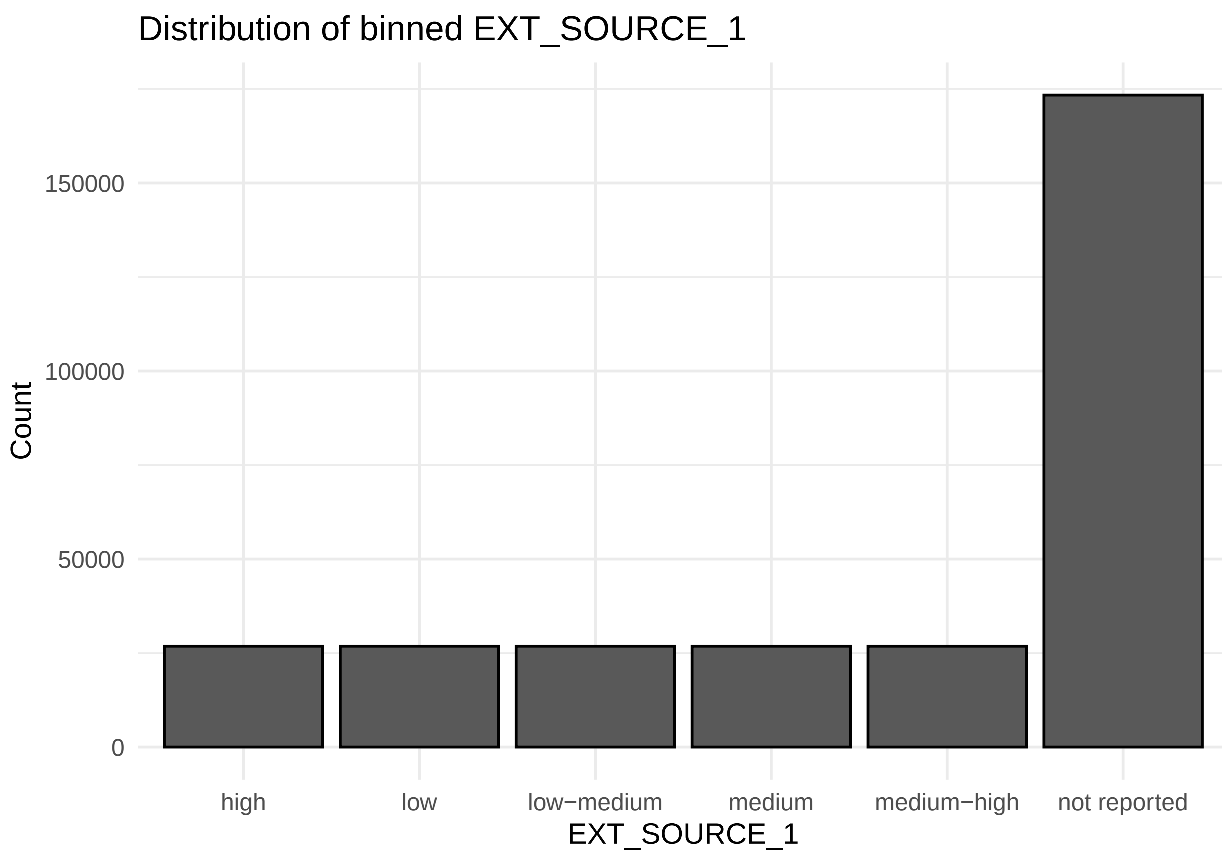
```r
# Binning the EXT_SOURCE variables into quintiles, keeping NAs
HomeCredit_application_train_data_clean <-
  HomeCredit_application_train_data_clean %>%
  mutate(
    EXT_SOURCE_1 = case_when(
      is.na(EXT_SOURCE_1) ~ "not reported",  # Handle missing values
      TRUE ~ case_when(
        ntile(EXT_SOURCE_1, 5) == 1 ~ "low",
        ntile(EXT_SOURCE_1, 5) == 2 ~ "low-medium",
        ntile(EXT_SOURCE_1, 5) == 3 ~ "medium",
        ntile(EXT_SOURCE_1, 5) == 4 ~ "medium-high",
        ntile(EXT_SOURCE_1, 5) == 5 ~ "high")),
    EXT_SOURCE_2 = case_when(
      is.na(EXT_SOURCE_2) ~ "not reported",  # Handle missing values
```

```r
      TRUE ~ case_when(
        ntile(EXT_SOURCE_2, 5) == 1 ~ "low",
        ntile(EXT_SOURCE_2, 5) == 2 ~ "low-medium",
        ntile(EXT_SOURCE_2, 5) == 3 ~ "medium",
        ntile(EXT_SOURCE_2, 5) == 4 ~ "medium-high",
        ntile(EXT_SOURCE_2, 5) == 5 ~ "high")),
    EXT_SOURCE_3 = case_when(
      is.na(EXT_SOURCE_3) ~ "not reported",  # Handle missing values
      TRUE ~ case_when(
        ntile(EXT_SOURCE_3, 5) == 1 ~ "low",
        ntile(EXT_SOURCE_3, 5) == 2 ~ "low-medium",
        ntile(EXT_SOURCE_3, 5) == 3 ~ "medium",
        ntile(EXT_SOURCE_3, 5) == 4 ~ "medium-high",
        ntile(EXT_SOURCE_3, 5) == 5 ~ "high"))
  )

# Viewing the distribution of of EXT_SOURCE_1
summary(HomeCredit_application_train_data_clean$EXT_SOURCE_1)
```

```
##    Length     Class      Mode
##    307511 character character
```

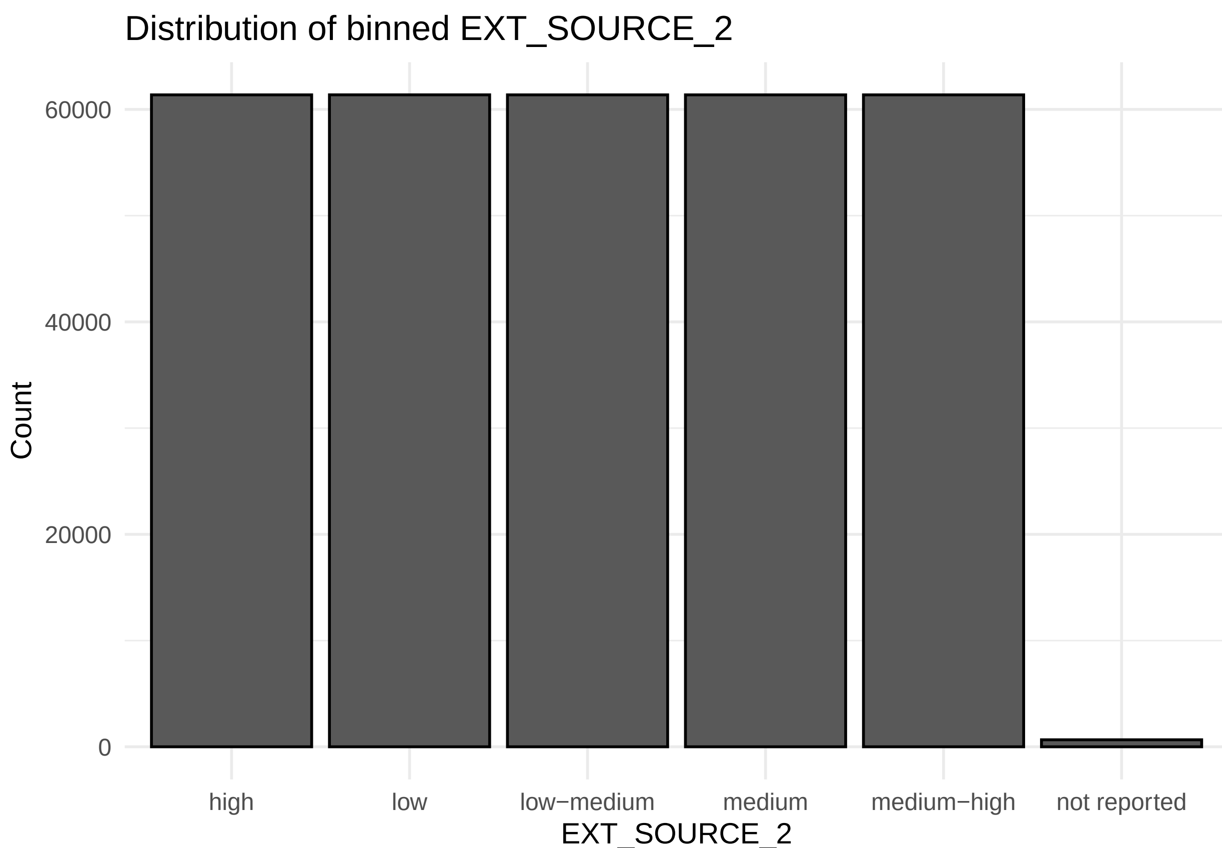
```r
ggplot(HomeCredit_application_train_data_clean, aes(x = EXT_SOURCE_1)) +
  geom_bar(color = "black") +
  labs(title = "Distribution of binned EXT_SOURCE_1",
       x = "EXT_SOURCE_1",
       y = "Count") +
  theme_minimal()
```




```r
# Viewing the distribution of of EXT_SOURCE_2
summary(HomeCredit_application_train_data_clean$EXT_SOURCE_2)
```

```
##    Length     Class      Mode 
##    307511 character character
```

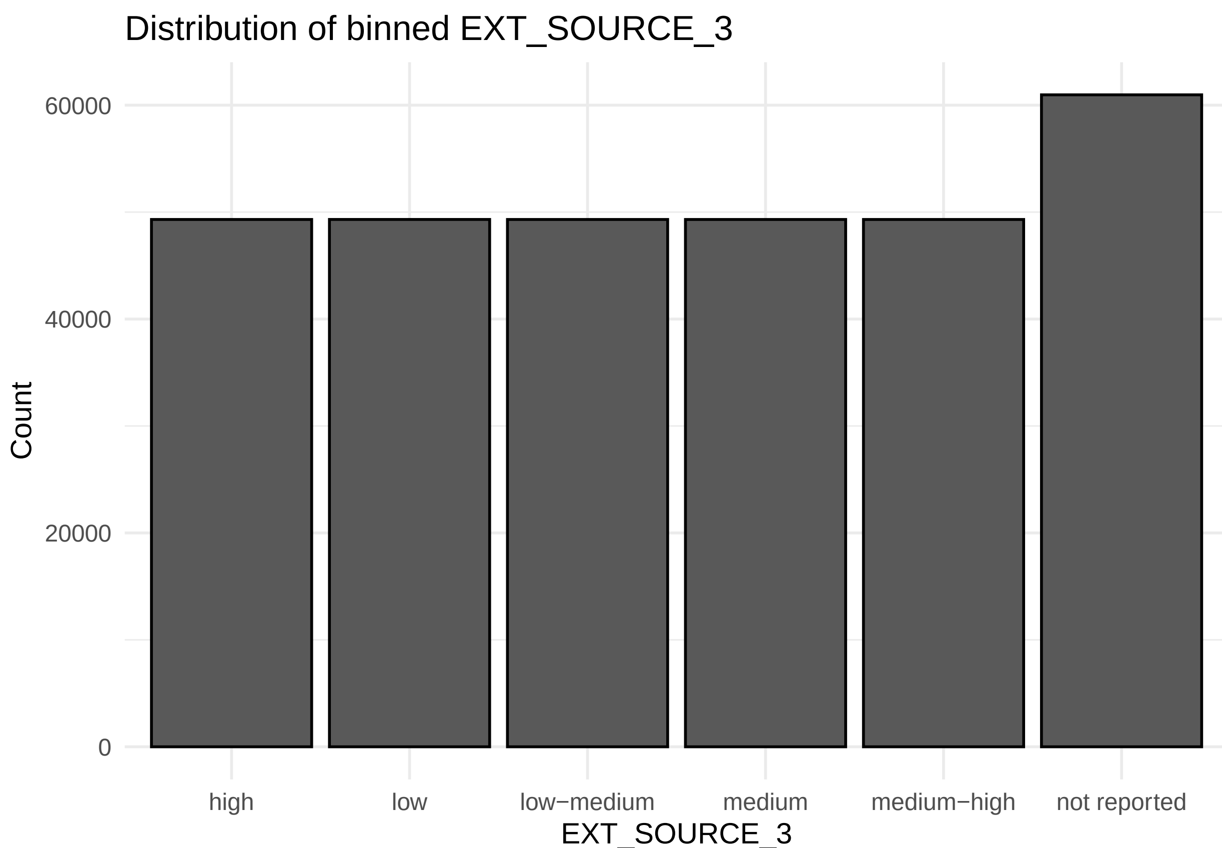
```r
ggplot(HomeCredit_application_train_data_clean, aes(x = EXT_SOURCE_2)) +
  geom_bar(color = "black") +
  labs(title = "Distribution of binned EXT_SOURCE_2",
       x = "EXT_SOURCE_2",
       y = "Count") +
  theme_minimal()
```




```r
# Viewing the distribution of of EXT_SOURCE_3
summary(HomeCredit_application_train_data_clean$EXT_SOURCE_3)
```

```
##    Length     Class      Mode
##    307511 character character
```

```r
ggplot(HomeCredit_application_train_data_clean, aes(x = EXT_SOURCE_3)) +
  geom_bar(color = "black") +
  labs(title = "Distribution of binned EXT_SOURCE_3",
       x = "EXT_SOURCE_3",
       y = "Count") +
  theme_minimal()
```



Normalized Information about the building where the client lives

43 columns with missing data fit this description:

- APARTMENTS_AVG
- APARTMENTS_MEDI
- APARTMENTS_MODE
- BASEMENTAREA_AVG
- BASEMENTAREA_MEDI
- BASEMENTAREA_MODE
- COMMONAREA_AVG
- COMMONAREA_MEDI
- COMMONAREA_MODE
- ELEVATORS_AVG
- ELEVATORS_MEDI
- ELEVATORS_MODE
- ENTRANCES_AVG
- ENTRANCES_MEDI
- ENTRANCES_MODE
- FLOORSMAX_AVG
- FLOORSMAX_MEDI
- FLOORSMAX_MODE
- FLOORSMIN_AVG
- FLOORSMIN_MEDI
- FLOORSMIN_MODE
- LANDAREA_AVG
- LANDAREA_MEDI

- LANDAREA_MODE
- LIVINGAPARTMENTS_AVG
- LIVINGAPARTMENTS_MEDI
- LIVINGAPARTMENTS_MODE
- LIVINGAREA_AVG
- LIVINGAREA_MEDI
- LIVINGAREA_MODE
- NONLIVINGAPARTMENTS_AVG
- NONLIVINGAPARTMENTS_MEDI
- NONLIVINGAPARTMENTS_MODE
- NONLIVINGAREA_AVG
- NONLIVINGAREA_MEDI
- NONLIVINGAREA_MODE
- TOTALAREA_MODE
- YEARS_BEGINEXPLUATATION_AVG
- YEARS_BEGINEXPLUATATION_MEDI
- YEARS_BEGINEXPLUATATION_MODE
- YEARS_BUILD_AVG
- YEARS_BUILD_MEDI
- YEARS_BUILD_MODE

What are the various values of HOUSETYPE_MODE?

```
# Querying unique values of APARTMENTS_AVG where HOUSETYPE_MODE is NA
HomeCredit_application_train_data_clean %>%
  distinct(HOUSETYPE_MODE)
```

```
##      HOUSETYPE_MODE
## 1    block of flats
## 2
## 3    terraced house
## 4 specific housing
```

Assumptions & Approach:

- None of the applicants are un-housed
- If the variable's distribution includes 0 as a possible value, then we will assume the missing values do not indicate additional information
- In the case that missing values do not indicate additional information, we will impute missing values using the median

```
# Viewing the distribution of the variables
## APARTMENTS_AVG
summary(HomeCredit_application_train_data_clean$APARTMENTS_AVG)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.06    0.09    0.12    0.15    1.00  156061
```

```
## APARTMENTS_MEDI
summary(HomeCredit_application_train_data_clean$APARTMENTS_MEDI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.06    0.09    0.12    0.15    1.00  156061
```

```r
## APARTMENTS_MODE
summary(HomeCredit_application_train_data_clean$APARTMENTS_MODE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.05    0.08    0.11    0.14    1.00  156061
```

```r
## BASEMENTAREA_AVG
summary(HomeCredit_application_train_data_clean$BASEMENTAREA_AVG)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.04    0.08    0.09    0.11    1.00  179943
```

```r
## BASEMENTAREA_MEDI
summary(HomeCredit_application_train_data_clean$BASEMENTAREA_MEDI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.04    0.08    0.09    0.11    1.00  179943
```

```r
## BASEMENTAREA_MODE
summary(HomeCredit_application_train_data_clean$BASEMENTAREA_MODE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.04    0.07    0.09    0.11    1.00  179943
```

```r
## COMMONAREA_AVG
summary(HomeCredit_application_train_data_clean$COMMONAREA_AVG)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.01    0.02    0.04    0.05    1.00  214865
```

```r
## COMMONAREA_MEDI
summary(HomeCredit_application_train_data_clean$COMMONAREA_MEDI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.01    0.02    0.04    0.05    1.00  214865
```

```r
## COMMONAREA_MODE
summary(HomeCredit_application_train_data_clean$COMMONAREA_MODE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.01    0.02    0.04    0.05    1.00  214865
```

```r
## ELEVATORS_AVG
summary(HomeCredit_application_train_data_clean$ELEVATORS_AVG)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.00    0.08    0.12    1.00  163891
```

```r
## ELEVATORS_MEDI
summary(HomeCredit_application_train_data_clean$ELEVATORS_MEDI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.00    0.08    0.12    1.00  163891
```

```r
## ELEVATORS_MODE
summary(HomeCredit_application_train_data_clean$ELEVATORS_MODE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.00    0.07    0.12    1.00  163891
```

```r
## ENTRANCES_AVG
summary(HomeCredit_application_train_data_clean$ENTRANCES_AVG)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.07    0.14    0.15    0.21    1.00  154828
```

```r
## ENTRANCES_MEDI
summary(HomeCredit_application_train_data_clean$ENTRANCES_MEDI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.07    0.14    0.15    0.21    1.00  154828
```

```r
## ENTRANCES_MODE
summary(HomeCredit_application_train_data_clean$ENTRANCES_MODE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.07    0.14    0.15    0.21    1.00  154828
```

```r
## FLOORSMAX_AVG
summary(HomeCredit_application_train_data_clean$FLOORSMAX_AVG)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.17    0.17    0.23    0.33    1.00  153020
```

```r
## FLOORSMAX_MEDI
summary(HomeCredit_application_train_data_clean$FLOORSMAX_MEDI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.17    0.17    0.23    0.33    1.00  153020
```

```r
## FLOORSMAX_MODE
summary(HomeCredit_application_train_data_clean$FLOORSMAX_MODE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.17    0.17    0.22    0.33    1.00  153020
```

```r
## FLOORSMIN_AVG
summary(HomeCredit_application_train_data_clean$FLOORSMIN_AVG)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.08    0.21    0.23    0.38    1.00  208642
```

```r
## FLOORSMIN_MEDI
summary(HomeCredit_application_train_data_clean$FLOORSMIN_MEDI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.08    0.21    0.23    0.38    1.00  208642
```

```r
## FLOORSMIN_MODE
summary(HomeCredit_application_train_data_clean$FLOORSMIN_MODE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.08    0.21    0.23    0.38    1.00  208642
```

```r
## LANDAREA_AVG
summary(HomeCredit_application_train_data_clean$LANDAREA_AVG)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.02    0.05    0.07    0.09    1.00  182590
```

```r
## LANDAREA_MEDI
summary(HomeCredit_application_train_data_clean$LANDAREA_MEDI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.02    0.05    0.07    0.09    1.00  182590
```

```r
## LANDAREA_MODE
summary(HomeCredit_application_train_data_clean$LANDAREA_MODE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.02    0.05    0.06    0.08    1.00  182590
```

```r
## LIVINGAPARTMENTS_AVG
summary(HomeCredit_application_train_data_clean$LIVINGAPARTMENTS_AVG)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.05    0.08    0.10    0.12    1.00  210199
```

```r
## LIVINGAPARTMENTS_MEDI
summary(HomeCredit_application_train_data_clean$LIVINGAPARTMENTS_MEDI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.05    0.08    0.10    0.12    1.00  210199
```

```r
## LIVINGAPARTMENTS_MODE
summary(HomeCredit_application_train_data_clean$LIVINGAPARTMENTS_MODE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.05    0.08    0.11    0.13    1.00  210199
```

```r
## LIVINGAREA_AVG
summary(HomeCredit_application_train_data_clean$LIVINGAREA_AVG)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.05    0.07    0.11    0.13    1.00  154350
```

```r
## LIVINGAREA_MEDI
summary(HomeCredit_application_train_data_clean$LIVINGAREA_MEDI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.05    0.07    0.11    0.13    1.00  154350
```

```r
## LIVINGAREA_MODE
summary(HomeCredit_application_train_data_clean$LIVINGAREA_MODE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.04    0.07    0.11    0.13    1.00  154350
```

```r
## NONLIVINGAPARTMENTS_AVG
summary(HomeCredit_application_train_data_clean$NONLIVINGAPARTMENTS_AVG)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.00    0.01    0.00    1.00  213514
```

```r
## NONLIVINGAPARTMENTS_MEDI
summary(HomeCredit_application_train_data_clean$NONLIVINGAPARTMENTS_MEDI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.00    0.01    0.00    1.00  213514
```

```r
## NONLIVINGAPARTMENTS_MODE
summary(HomeCredit_application_train_data_clean$NONLIVINGAPARTMENTS_MODE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.00    0.01    0.00    1.00  213514
```

```r
## NONLIVINGAREA_AVG
summary(HomeCredit_application_train_data_clean$NONLIVINGAREA_AVG)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.00    0.03    0.03    1.00  169682
```

```r
## NONLIVINGAREA_MEDI
summary(HomeCredit_application_train_data_clean$NONLIVINGAREA_MEDI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.00    0.03    0.03    1.00  169682
```

```r
## NONLIVINGAREA_MODE
summary(HomeCredit_application_train_data_clean$NONLIVINGAREA_MODE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.00    0.03    0.02    1.00  169682
```

```r
## TOTALAREA_MODE
summary(HomeCredit_application_train_data_clean$TOTALAREA_MODE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.04    0.07    0.10    0.13    1.00  148431
```

```r
## YEARS_BEGINEXPLUATATION_AVG
summary(HomeCredit_application_train_data_clean$YEARS_BEGINEXPLUATATION_AVG)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.98    0.98    0.98    0.99    1.00  150007
```

```r
## YEARS_BEGINEXPLUATATION_MEDI
summary(HomeCredit_application_train_data_clean$YEARS_BEGINEXPLUATATION_MEDI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.98    0.98    0.98    0.99    1.00  150007
```

```r
## YEARS_BEGINEXPLUATATION_MODE
summary(HomeCredit_application_train_data_clean$YEARS_BEGINEXPLUATATION_MODE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.98    0.98    0.98    0.99    1.00  150007
```

```r
## YEARS_BUILD_AVG
summary(HomeCredit_application_train_data_clean$YEARS_BUILD_AVG)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.69    0.76    0.75    0.82    1.00  204488
```

```r
## YEARS_BUILD_MEDI
summary(HomeCredit_application_train_data_clean$YEARS_BUILD_MEDI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.69    0.76    0.76    0.83    1.00  204488
```

```r
## YEARS_BUILD_MODE
summary(HomeCredit_application_train_data_clean$YEARS_BUILD_MODE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.70    0.76    0.76    0.82    1.00  204488
```

Since each of the variable's distributions include 0, we will impute missing values for each variable using the median.

```r
# Imputing missing values in AMT_ANNUITY using the Median
HomeCredit_application_train_data_clean <-
  HomeCredit_application_train_data_clean %>%
  mutate(across(
    c(APARTMENTS_AVG,
      APARTMENTS_MEDI,
      APARTMENTS_MODE,
      BASEMENTAREA_AVG,
      BASEMENTAREA_MEDI,
      BASEMENTAREA_MODE,
      COMMONAREA_AVG,
      COMMONAREA_MEDI,
      COMMONAREA_MODE,
      ELEVATORS_AVG,
      ELEVATORS_MEDI,
      ELEVATORS_MODE,
      ENTRANCES_AVG,
      ENTRANCES_MEDI,
      ENTRANCES_MODE,
      FLOORSMAX_AVG,
      FLOORSMAX_MEDI,
      FLOORSMAX_MODE,
      FLOORSMIN_AVG,
      FLOORSMIN_MEDI,
      FLOORSMIN_MODE,
      LANDAREA_AVG,
      LANDAREA_MEDI,
      LANDAREA_MODE,
      LIVINGAPARTMENTS_AVG,
      LIVINGAPARTMENTS_MEDI,
      LIVINGAPARTMENTS_MODE,
      LIVINGAREA_AVG,
      LIVINGAREA_MEDI,
      LIVINGAREA_MODE,
      NONLIVINGAPARTMENTS_AVG,
      NONLIVINGAPARTMENTS_MEDI,
      NONLIVINGAPARTMENTS_MODE,
      NONLIVINGAREA_AVG,
      NONLIVINGAREA_MEDI,
      NONLIVINGAREA_MODE,
      TOTALAREA_MODE,
      YEARS_BEGINEXPLUATATION_AVG,
      YEARS_BEGINEXPLUATATION_MEDI,
      YEARS_BEGINEXPLUATATION_MODE,
```

```
      YEARS_BUILD_AVG,
      YEARS_BUILD_MEDI,
      YEARS_BUILD_MODE),
    ~ if_else(is.na(.), median(., na.rm = TRUE), .)
  ))
```

How many observation of client's social surroundings

Observable

- OBS_30_CNT_SOCIAL_CIRCLE: How many observation of client's social surroundings with observable 30 DPD (days past due) default
- OBS_60_CNT_SOCIAL_CIRCLE: How many observation of client's social surroundings with observable 60 DPD (days past due) default

Defaulted

- DEF_30_CNT_SOCIAL_CIRCLE: How many observation of client's social surroundings defaulted on 30 DPD (days past due)
- DEF_60_CNT_SOCIAL_CIRCLE: How many observation of client's social surroundings defaulted on 60 (days past due) DPD

```
# Viewing the distribution of the variables
## OBS_30_CNT_SOCIAL_CIRCLE
summary(HomeCredit_application_train_data_clean$OBS_30_CNT_SOCIAL_CIRCLE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   0.000   0.000   1.422   2.000 348.000    1021
```

```
## OBS_60_CNT_SOCIAL_CIRCLE
summary(HomeCredit_application_train_data_clean$OBS_60_CNT_SOCIAL_CIRCLE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   0.000   0.000   1.405   2.000 344.000    1021
```

```
## DEF_30_CNT_SOCIAL_CIRCLE
summary(HomeCredit_application_train_data_clean$DEF_30_CNT_SOCIAL_CIRCLE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.0000  0.0000  0.0000  0.1434  0.0000 34.0000    1021
```

```
## DEF_60_CNT_SOCIAL_CIRCLE
summary(HomeCredit_application_train_data_clean$DEF_60_CNT_SOCIAL_CIRCLE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0     0.0     0.0     0.1     0.0    24.0    1021
```

Assumptions & Approach:

- Assuming the missing values do not indicate additional information
- Impute missing values using the median due to skewness

```r
# Imputing missing values in AMT_ANNUITY using the Median
HomeCredit_application_train_data_clean <-
  HomeCredit_application_train_data_clean %>%
  mutate(across(
    c(OBS_30_CNT_SOCIAL_CIRCLE,
      OBS_60_CNT_SOCIAL_CIRCLE,
      DEF_30_CNT_SOCIAL_CIRCLE,
      DEF_60_CNT_SOCIAL_CIRCLE),
    ~ if_else(is.na(.), median(., na.rm = TRUE), .)
  ))
```

DAYS_LAST_PHONE_CHANGE

DAYS_LAST_PHONE_CHANGE is how many days before application did client change phones.

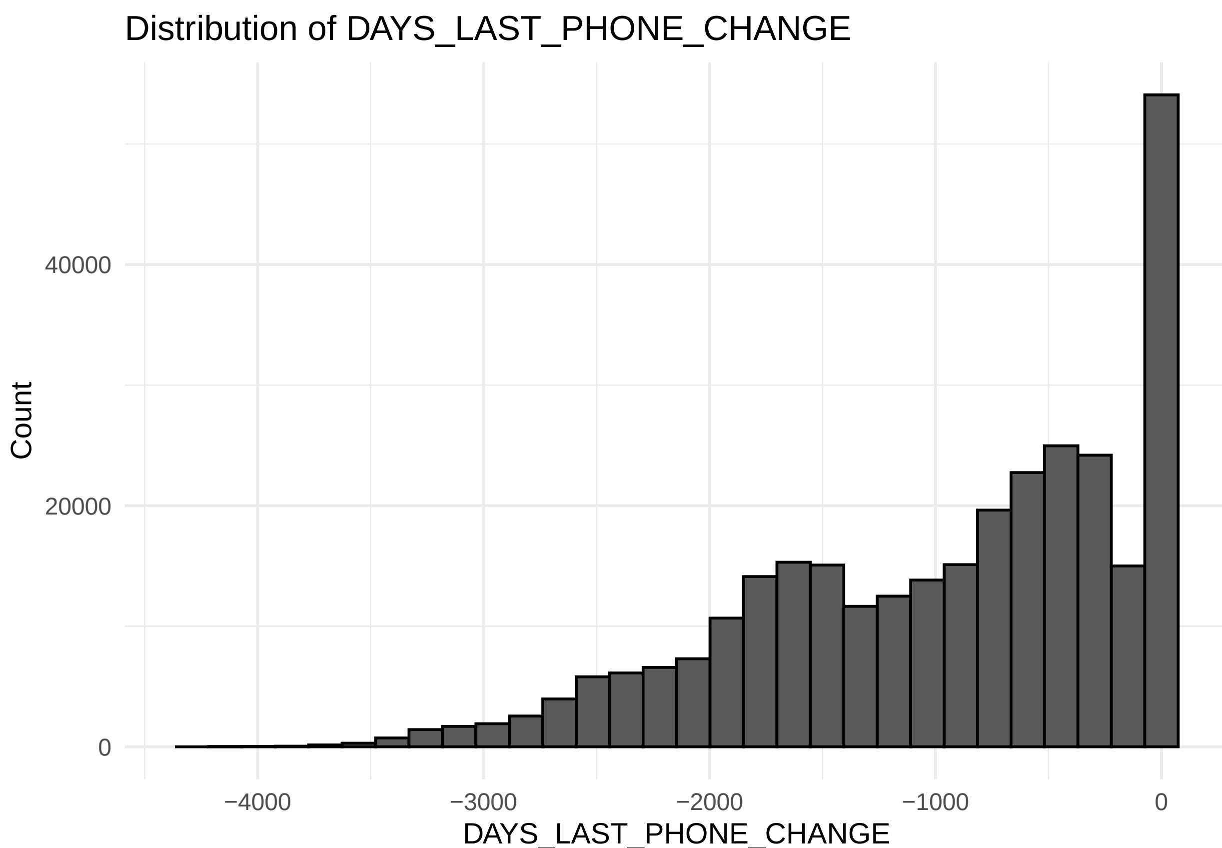
```r
# Viewing the distribution of DAYS_LAST_PHONE_CHANGE
summary(HomeCredit_application_train_data_clean$DAYS_LAST_PHONE_CHANGE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## -4292.0 -1570.0  -757.0  -962.9  -274.0     0.0       1
```

```r
ggplot(HomeCredit_application_train_data_clean,
       aes(x = DAYS_LAST_PHONE_CHANGE)) +
  geom_histogram(color = "black") +
  labs(title = "Distribution of DAYS_LAST_PHONE_CHANGE",
       x = "DAYS_LAST_PHONE_CHANGE",
       y = "Count") +
  theme_minimal()
```

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

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_bin()`).
```



Assumptions & Approach:

- Assuming the missing values do not indicate additional information
- Impute missing values using the median due to skewness

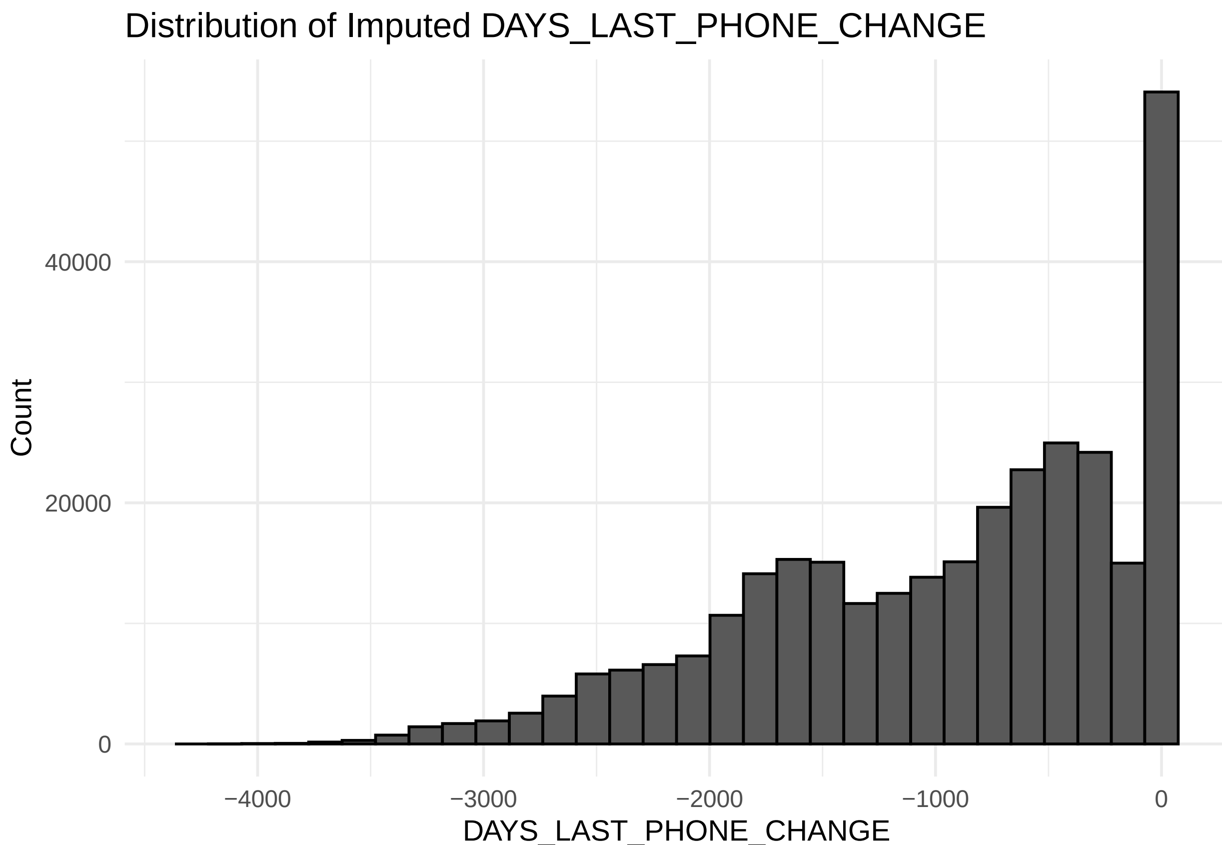
```
# Imputing missing values in DAYS_LAST_PHONE_CHANGE using the Median
HomeCredit_application_train_data_clean <-
  HomeCredit_application_train_data_clean %>%
  mutate(DAYS_LAST_PHONE_CHANGE = if_else(is.na(DAYS_LAST_PHONE_CHANGE),
                                 median(DAYS_LAST_PHONE_CHANGE, na.rm = TRUE),
                                 DAYS_LAST_PHONE_CHANGE))

# Viewing the distribution of of DAYS_LAST_PHONE_CHANGE after imputing
summary(HomeCredit_application_train_data_clean$DAYS_LAST_PHONE_CHANGE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -4292.0 -1570.0  -757.0  -962.9  -274.0     0.0
```

```
ggplot(HomeCredit_application_train_data_clean,
       aes(x = DAYS_LAST_PHONE_CHANGE)) +
  geom_histogram(color = "black") +
  labs(title = "Distribution of Imputed DAYS_LAST_PHONE_CHANGE",
       x = "DAYS_LAST_PHONE_CHANGE",
       y = "Count") +
  theme_minimal()
```

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



The distribution of DAYS_LAST_PHONE_CHANGE after imputing looks very similar to the variable’s distribution prior to imputing.

Number of inquiries to Credit Bureau about the client before application

- AMT_REQ_CREDIT_BUREAU_HOUR: Number of inquiries to Credit Bureau about the client one hour before application
- AMT_REQ_CREDIT_BUREAU_DAY: Number of inquiries to Credit Bureau about the client one day before application (excluding one hour before application)
- AMT_REQ_CREDIT_BUREAU_WEEK: Number of inquiries to Credit Bureau about the client one week before application (excluding one day before application)
- AMT_REQ_CREDIT_BUREAU_MON: Number of inquiries to Credit Bureau about the client one month before application (excluding one week before application)
- AMT_REQ_CREDIT_BUREAU_QRT: Number of inquiries to Credit Bureau about the client 3 month before application (excluding one month before application)
- AMT_REQ_CREDIT_BUREAU_YEAR: Number of inquiries to Credit Bureau about the client one day year (excluding last 3 months before application)

```
# Viewing the distribution of the variables
## AMT_REQ_CREDIT_BUREAU_HOUR
summary(HomeCredit_application_train_data_clean$AMT_REQ_CREDIT_BUREAU_HOUR)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.00    0.01    0.00    4.00   41519
```

```r
## AMT_REQ_CREDIT_BUREAU_DAY
summary(HomeCredit_application_train_data_clean$AMT_REQ_CREDIT_BUREAU_DAY)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00    0.00    0.00    0.01    0.00    9.00   41519
```

```r
## AMT_REQ_CREDIT_BUREAU_WEEK
summary(HomeCredit_application_train_data_clean$AMT_REQ_CREDIT_BUREAU_WEEK)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00    0.00    0.00    0.03    0.00    8.00   41519
```

```r
## AMT_REQ_CREDIT_BUREAU_MON
summary(HomeCredit_application_train_data_clean$AMT_REQ_CREDIT_BUREAU_MON)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00    0.00    0.00    0.27    0.00   27.00   41519
```

```r
## AMT_REQ_CREDIT_BUREAU_QRT
summary(HomeCredit_application_train_data_clean$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
```

```r
## AMT_REQ_CREDIT_BUREAU_YEAR
summary(HomeCredit_application_train_data_clean$AMT_REQ_CREDIT_BUREAU_YEAR)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##     0.0     0.0     1.0     1.9     3.0    25.0   41519
```

Assumptions & Approach:

- Assuming the missing values do not indicate additional information
- Impute missing values using the median due to skewness

```r
# Imputing missing values in AMT_ANNUITY using the Median
HomeCredit_application_train_data_clean <-
  HomeCredit_application_train_data_clean %>%
  mutate(across(
    c(AMT_REQ_CREDIT_BUREAU_HOUR,
      AMT_REQ_CREDIT_BUREAU_DAY,
      AMT_REQ_CREDIT_BUREAU_WEEK,
      AMT_REQ_CREDIT_BUREAU_MON,
      AMT_REQ_CREDIT_BUREAU_QRT,
      AMT_REQ_CREDIT_BUREAU_YEAR),
    ~ if_else(is.na(.), median(., na.rm = TRUE), .)
  ))
```

Final Missing Data Evaluation

```r
# Identifying columns with missing data in the train data
clean_missing_values <- HomeCredit_application_train_data_clean %>%
  summarise(across(everything(), ~ sum(is.na(.)))) %>%
  pivot_longer(everything(),
               names_to = "column",
               values_to = "missing_count") %>%
  filter(missing_count > 0) %>%
  arrange(desc(missing_count))

clean_missing_values
```

```
## # A tibble: 0 x 2
## # i 2 variables: column <chr>, missing_count <int>
```

All missing values have been adjusted for through various customized solutions.

Near Zero Variance

The goal of this section is to detect variables that have very little variation or are mostly constant, which are often uninformative in predictive modeling sometimes leading to over fitting or instability.

```r
# Identifying variables with near-zero or zero variance
nzv_vars <- nearZeroVar(HomeCredit_application_train_data_clean,
                        saveMetrics = TRUE)

# Identifying Column indices of near-zero variance variables
nzv_cols <- nearZeroVar(HomeCredit_application_train_data_clean)

# Removing near-zero variance variables from the data set
HomeCredit_application_train_data_clean <-
  HomeCredit_application_train_data_clean[, -nzv_cols]
```

50 near zero variance variables were detected and removed from the data set.

Predictor-Target Relationships

The goal of this section is to explore the relationship between target and predictors, looking for potentially strong predictors that could be included later in a model.

Categorical variables

How many categorical predictor variables are there?

```r
# Identifying remaining categorical variables
colnames(select_if(HomeCredit_application_train_data_clean, is.character))
```

```
##  [1] "NAME_CONTRACT_TYPE"         "CODE_GENDER"
##  [3] "FLAG_OWN_CAR"               "FLAG_OWN_REALTY"
##  [5] "AMT_GOODS_PRICE"            "NAME_TYPE_SUITE"
```

```
##  [7] "NAME_INCOME_TYPE"           "NAME_EDUCATION_TYPE"
##  [9] "NAME_FAMILY_STATUS"         "NAME_HOUSING_TYPE"
## [11] "OWN_CAR_AGE"                "OCCUPATION_TYPE"
## [13] "WEEKDAY_APPR_PROCESS_START" "ORGANIZATION_TYPE"
## [15] "EXT_SOURCE_1"               "EXT_SOURCE_2"
## [17] "EXT_SOURCE_3"               "FONDKAPREMONT_MODE"
## [19] "HOUSETYPE_MODE"             "WALLSMATERIAL_MODE"
## [21] "EMERGENCYSTATE_MODE"
```

```
# Converting character categorical variables to factor variables
HomeCredit_application_train_data_clean <-
  HomeCredit_application_train_data_clean %>%
  mutate(across(c(NAME_CONTRACT_TYPE,
                  CODE_GENDER,
                  FLAG_OWN_CAR,
                  FLAG_OWN_REALTY,
                  AMT_GOODS_PRICE,
                  NAME_TYPE_SUITE,
                  NAME_INCOME_TYPE,
                  NAME_EDUCATION_TYPE,
                  NAME_FAMILY_STATUS,
                  NAME_HOUSING_TYPE,
                  OWN_CAR_AGE,
                  OCCUPATION_TYPE,
                  WEEKDAY_APPR_PROCESS_START,
                  ORGANIZATION_TYPE,
                  EXT_SOURCE_1,
                  EXT_SOURCE_2,
                  EXT_SOURCE_3,
                  FONDKAPREMONT_MODE,
                  HOUSETYPE_MODE,
                  WALLSMATERIAL_MODE,
                  EMERGENCYSTATE_MODE),
                as.factor))

# Converting additional variables to factor variables
HomeCredit_application_train_data_clean <-
  HomeCredit_application_train_data_clean %>%
  mutate(across(c(FLAG_EMP_PHONE,
                  FLAG_WORK_PHONE,
                  FLAG_PHONE,
                  FLAG_EMAIL,
                  FLAG_DOCUMENT_3,
                  FLAG_DOCUMENT_6,
                  FLAG_DOCUMENT_8,
                  REGION_RATING_CLIENT,
                  REGION_RATING_CLIENT_W_CITY,
                  HOUR_APPR_PROCESS_START,
                  REG_REGION_NOT_WORK_REGION,
                  REG_CITY_NOT_LIVE_CITY,
                  REG_CITY_NOT_WORK_CITY,
                  LIVE_CITY_NOT_WORK_CITY),
                as.factor))
```

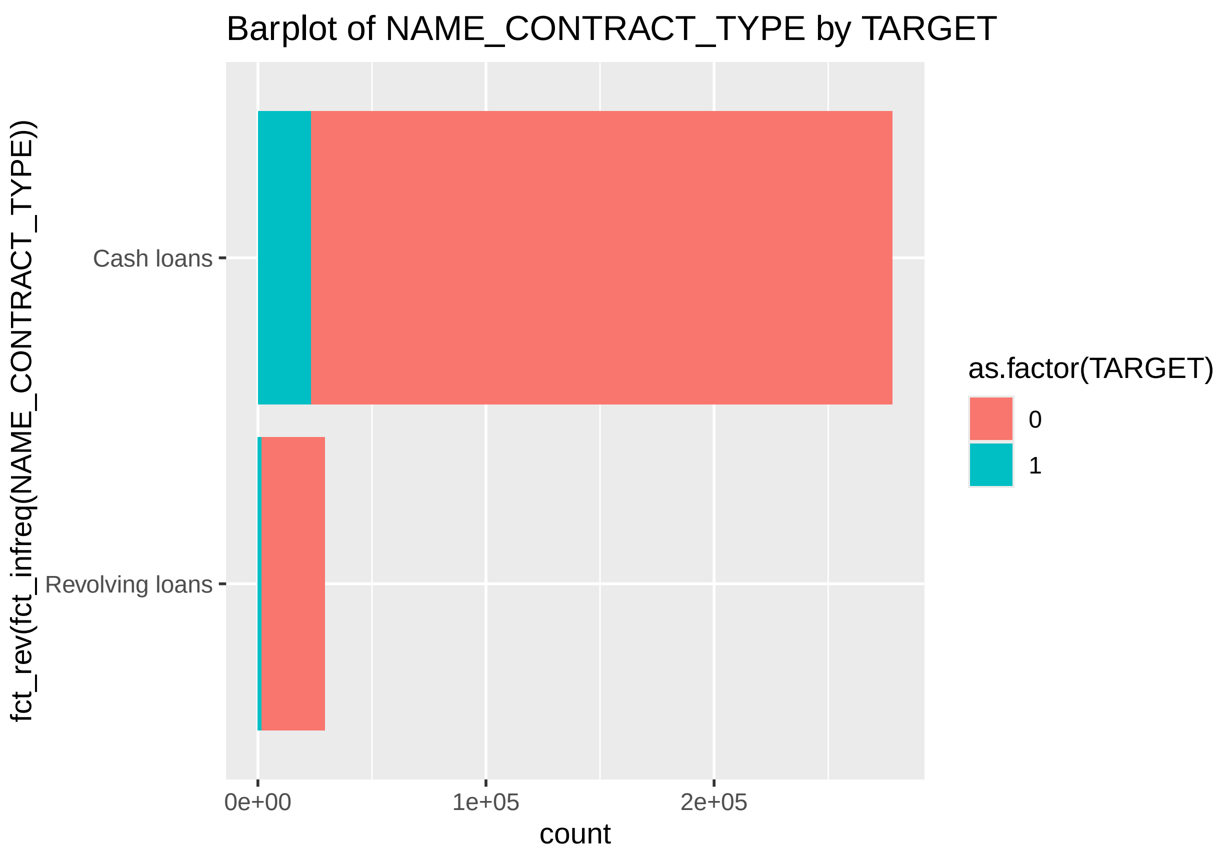
```r
# Identifying factor variables
colnames(select_if(HomeCredit_application_train_data_clean, is.factor ))
```

```
##  [1] "NAME_CONTRACT_TYPE"          "CODE_GENDER"
##  [3] "FLAG_OWN_CAR"                "FLAG_OWN_REALTY"
##  [5] "AMT_GOODS_PRICE"             "NAME_TYPE_SUITE"
##  [7] "NAME_INCOME_TYPE"            "NAME_EDUCATION_TYPE"
##  [9] "NAME_FAMILY_STATUS"          "NAME_HOUSING_TYPE"
## [11] "OWN_CAR_AGE"                 "FLAG_EMP_PHONE"
## [13] "FLAG_WORK_PHONE"             "FLAG_PHONE"
## [15] "FLAG_EMAIL"                  "OCCUPATION_TYPE"
## [17] "REGION_RATING_CLIENT"        "REGION_RATING_CLIENT_W_CITY"
## [19] "WEEKDAY_APPR_PROCESS_START"  "HOUR_APPR_PROCESS_START"
## [21] "REG_REGION_NOT_WORK_REGION"  "REG_CITY_NOT_LIVE_CITY"
## [23] "REG_CITY_NOT_WORK_CITY"      "LIVE_CITY_NOT_WORK_CITY"
## [25] "ORGANIZATION_TYPE"           "EXT_SOURCE_1"
## [27] "EXT_SOURCE_2"                "EXT_SOURCE_3"
## [29] "FONDKAPREMONT_MODE"          "HOUSETYPE_MODE"
## [31] "WALLSMATERIAL_MODE"          "EMERGENCYSTATE_MODE"
## [33] "FLAG_DOCUMENT_3"             "FLAG_DOCUMENT_6"
## [35] "FLAG_DOCUMENT_8"
```

NAME_CONTRACT_TYPE

NAME_CONTRACT_TYPE: Identification if loan is cash or revolving

```r
# NAME_CONTRACT_TYPE barplot
HomeCredit_application_train_data_clean %>% ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(NAME_CONTRACT_TYPE)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of NAME_CONTRACT_TYPE by TARGET") +
  coord_flip()
```




```r
# NAME_CONTRACT_TYPE proportion table
kable(t(HomeCredit_application_train_data_clean %>%
            select(TARGET, NAME_CONTRACT_TYPE) %>%
            table() %>%
            prop.table(margin = 2) %>%
            round(2)), format = "markdown")
```

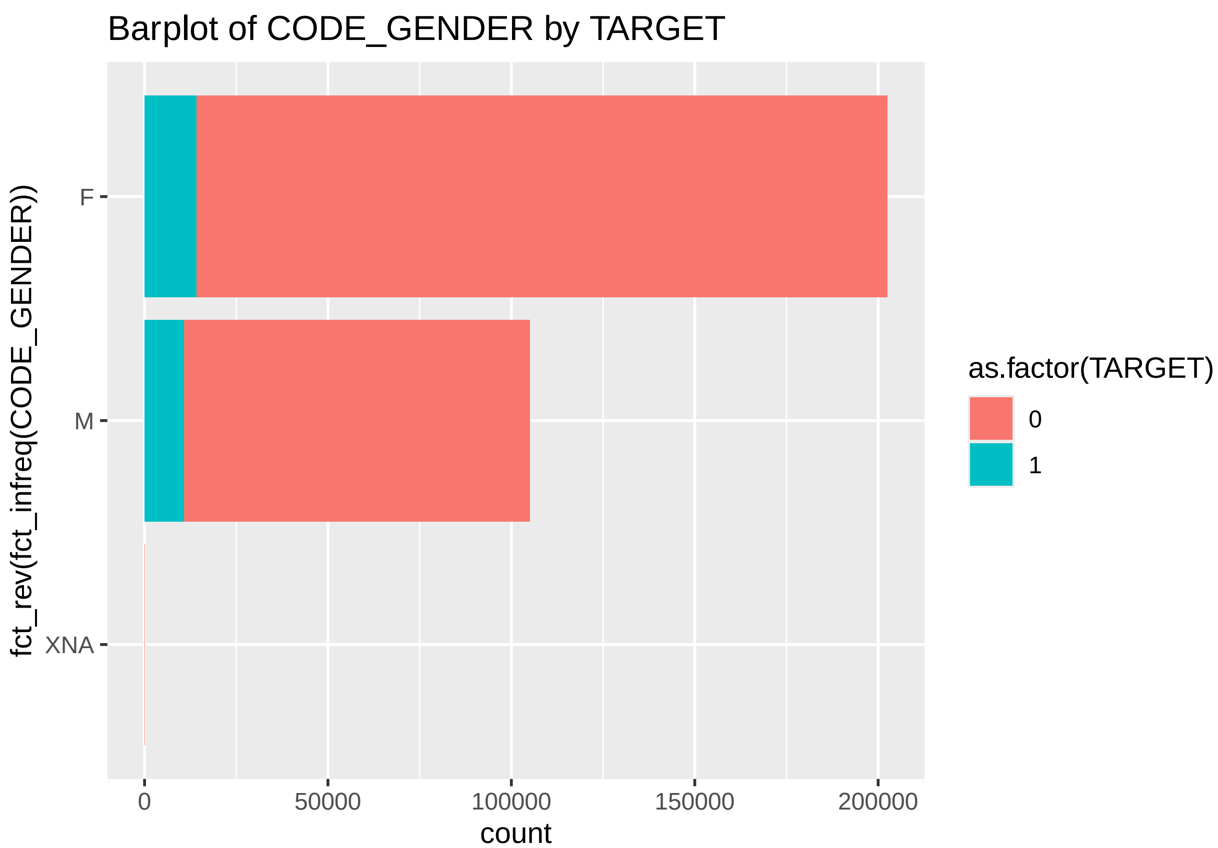
	0	1
Cash loans	0.92	0.08
Revolving loans	0.95	0.05

Most NAME_CONTRACT_TYPEs are Cash Loans. This group is also more likely to default (8%) compared to revolving loans (5%).

CODE_GENDER

CODE_GENDER: Gender of the client

```r
# CODE_GENDER barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(CODE_GENDER)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of CODE_GENDER by TARGET") +
  coord_flip()
```




```
# CODE_GENDER proportion table
kable(t(HomeCredit_application_train_data_clean %>%
          select(TARGET, CODE_GENDER) %>%
          table() %>%
          prop.table(margin = 2) %>%
          round(2)), format = "markdown")
```

	0	1
F	0.93	0.07
M	0.90	0.10
XNA	1.00	0.00

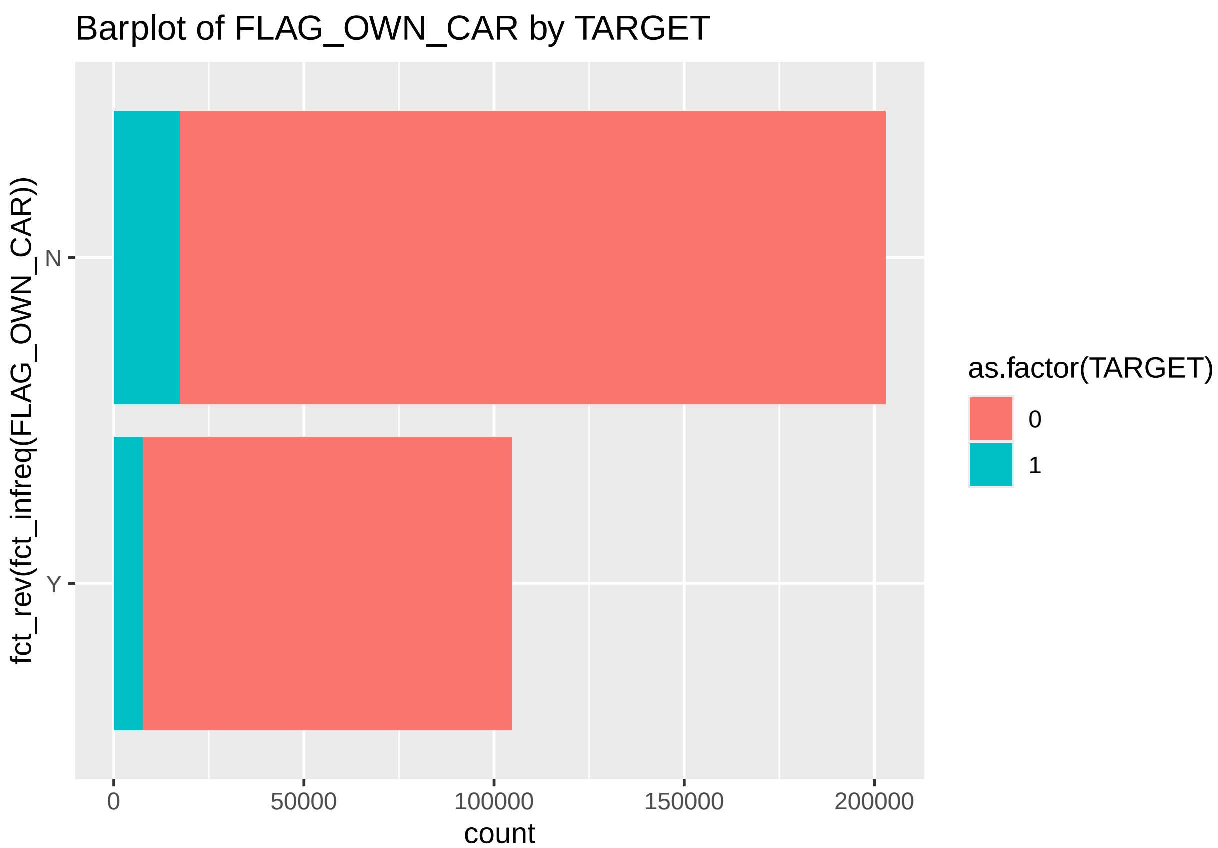
There aer more female than male observations in the dataset, but default rate for males (10%) is slightly higher than that for females (7%).

FLAG_OWN_CAR

FLAG_OWN_CAR: Flag if the client owns a car

```
# FLAG_OWN_CAR barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(FLAG_OWN_CAR)),
               fill = as.factor(TARGET))) +
```

```r
  ggtitle("Barplot of FLAG_OWN_CAR by TARGET") +
  coord_flip()
```




```r
# FLAG_OWN_CAR proportion table
kable(t(HomeCredit_application_train_data_clean %>%
          select(TARGET, FLAG_OWN_CAR) %>%
          table() %>%
          prop.table(margin = 2) %>%
          round(2)), format = "markdown")
```

	0	1
N	0.91	0.09
Y	0.93	0.07

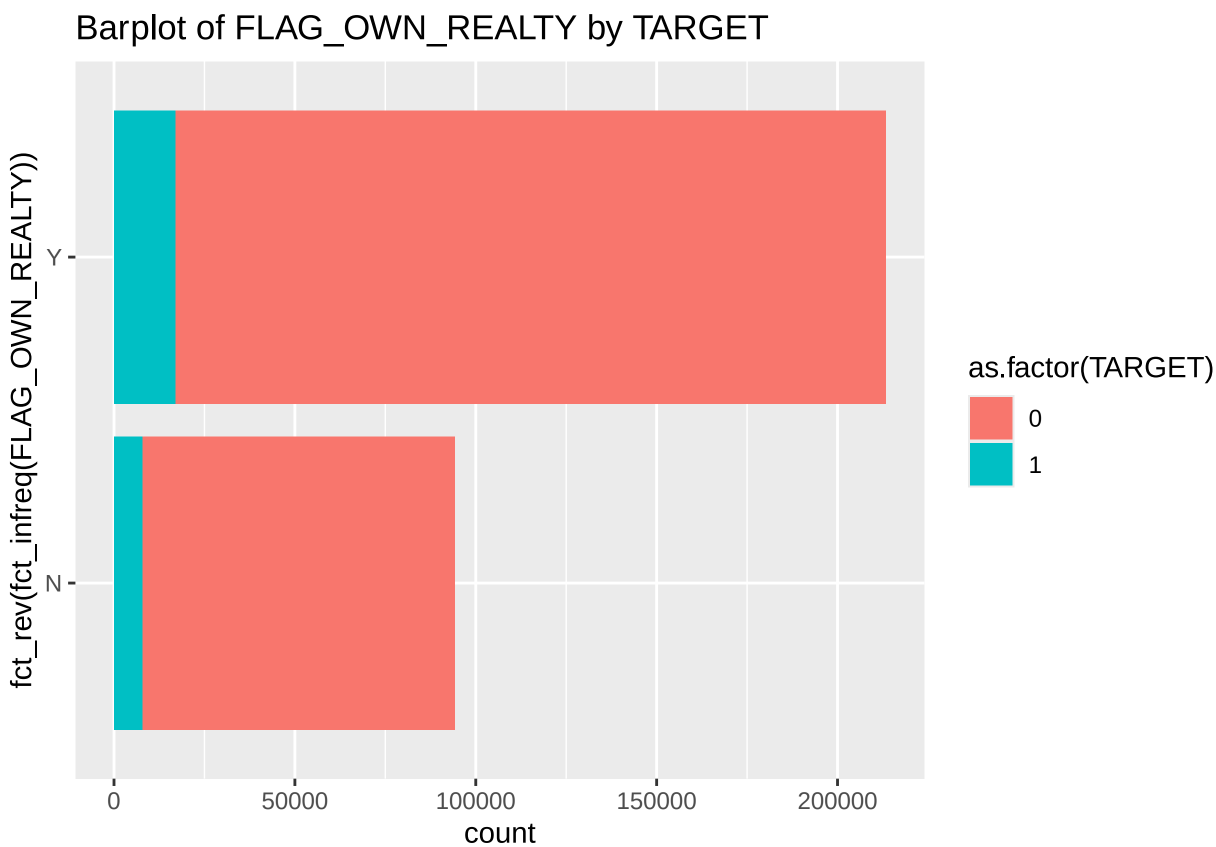
More clients don't own cars than do, but the default rate seems to be higher for those who do not own a car (9%) than for those that do (7%).

FLAG_OWN_REALTY

FLAG_OWN_REALTY: Flag if client owns a house or flat

```r
# FLAG_OWN_REALTY barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
```

```r
geom_bar(aes(x = fct_rev(fct_infreq(FLAG_OWN_REALTY)),
             fill = as.factor(TARGET))) +
ggtitle("Barplot of FLAG_OWN_REALTY by TARGET") +
coord_flip()
```




```r
# FLAG_OWN_REALTY proportion table
kable(t(HomeCredit_application_train_data_clean %>%
          select(TARGET, FLAG_OWN_REALTY) %>%
          table() %>%
          prop.table(margin = 2) %>%
          round(2)), format = "markdown")
```

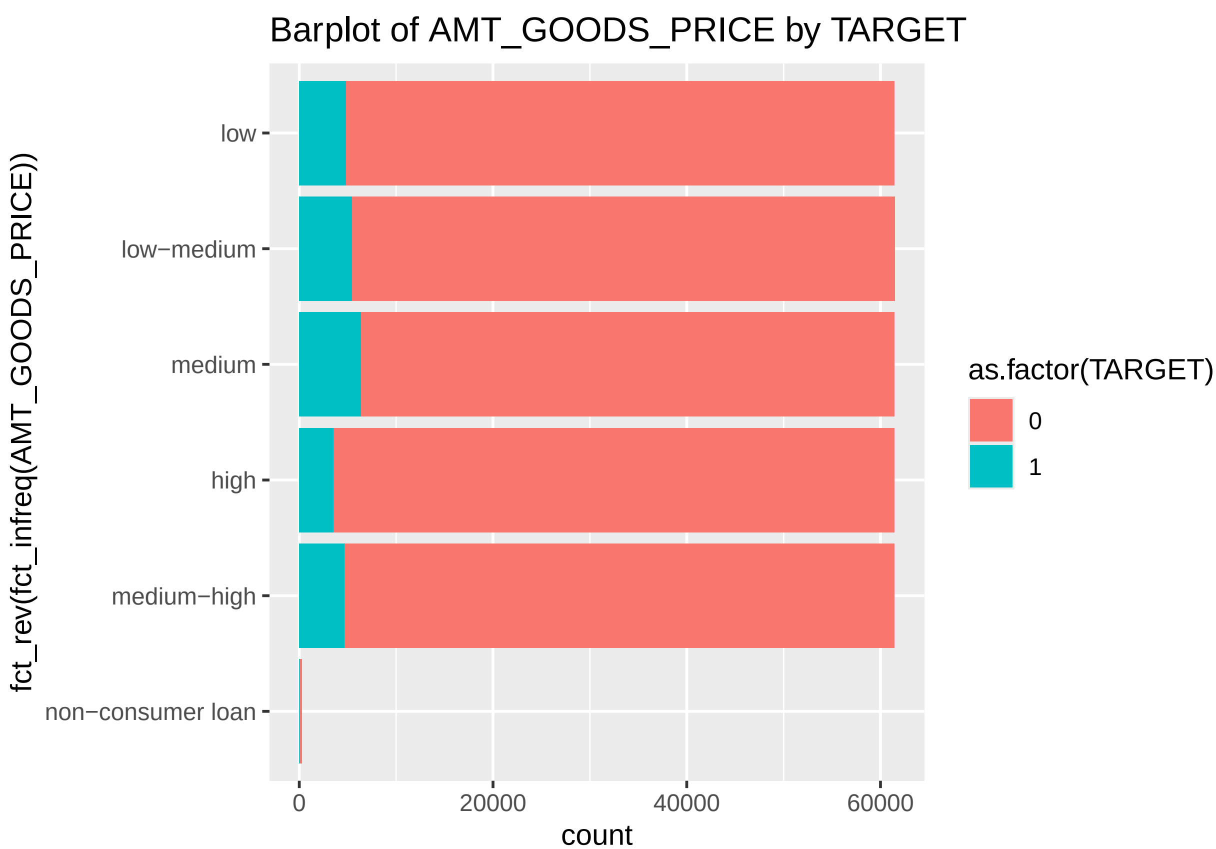
	0	1
N	0.92	0.08
Y	0.92	0.08

More clients in the data set own a house or flat than done, but there is no difference in default rate between the two groups.

AMT_GOODS_PRICE

AMT_GOODS_PRICE: For consumer loans it is the price of the goods for which the loan is given

```r
# AMT_GOODS_PRICE barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(AMT_GOODS_PRICE)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of AMT_GOODS_PRICE by TARGET") +
  coord_flip()
```




```r
# AMT_GOODS_PRICE proportion table
kable(t(HomeCredit_application_train_data_clean %>%
            select(TARGET, AMT_GOODS_PRICE) %>%
            table() %>%
            prop.table(margin = 2) %>%
            round(2)), format = "markdown")
```

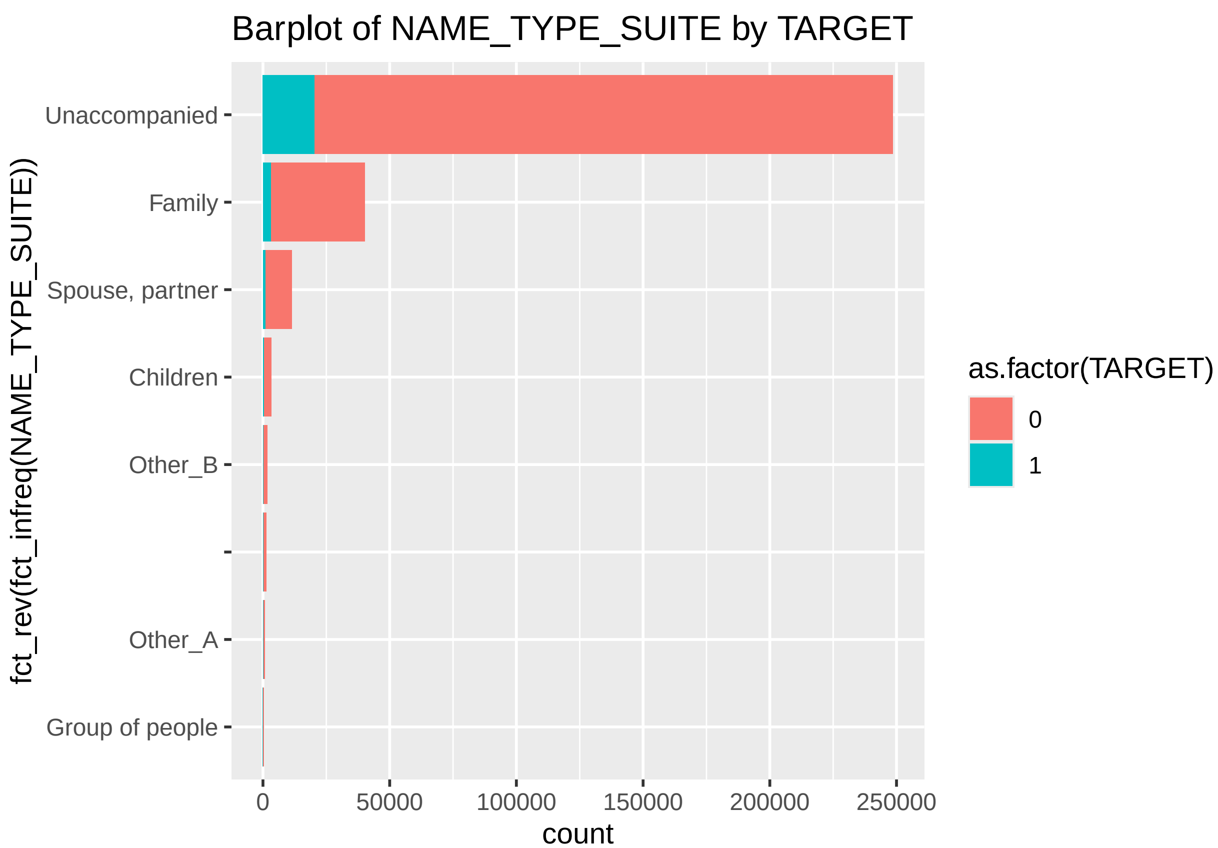
	0	1
high	0.94	0.06
low	0.92	0.08
low-medium	0.91	0.09
medium	0.90	0.10
medium-high	0.92	0.08
non-consumer loan	0.92	0.08

Default rates seem to be highest among those with a medium AMT_GOODS_PRICE, but it's pretty equal across groups.

NAME_TYPE_SUITE

NAME_TYPE_SUITE: Who was accompanying client when he was applying for the loan

```
# NAME_TYPE_SUITE barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(NAME_TYPE_SUITE)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of NAME_TYPE_SUITE by TARGET") +
  coord_flip()
```




```
# NAME_TYPE_SUITE proportion table
kable(t(HomeCredit_application_train_data_clean %>%
            select(TARGET, NAME_TYPE_SUITE) %>%
            table() %>%
            prop.table(margin = 2) %>%
            round(2)), format = "markdown")
```

	0	1
	0.95	0.05
Children	0.93	0.07

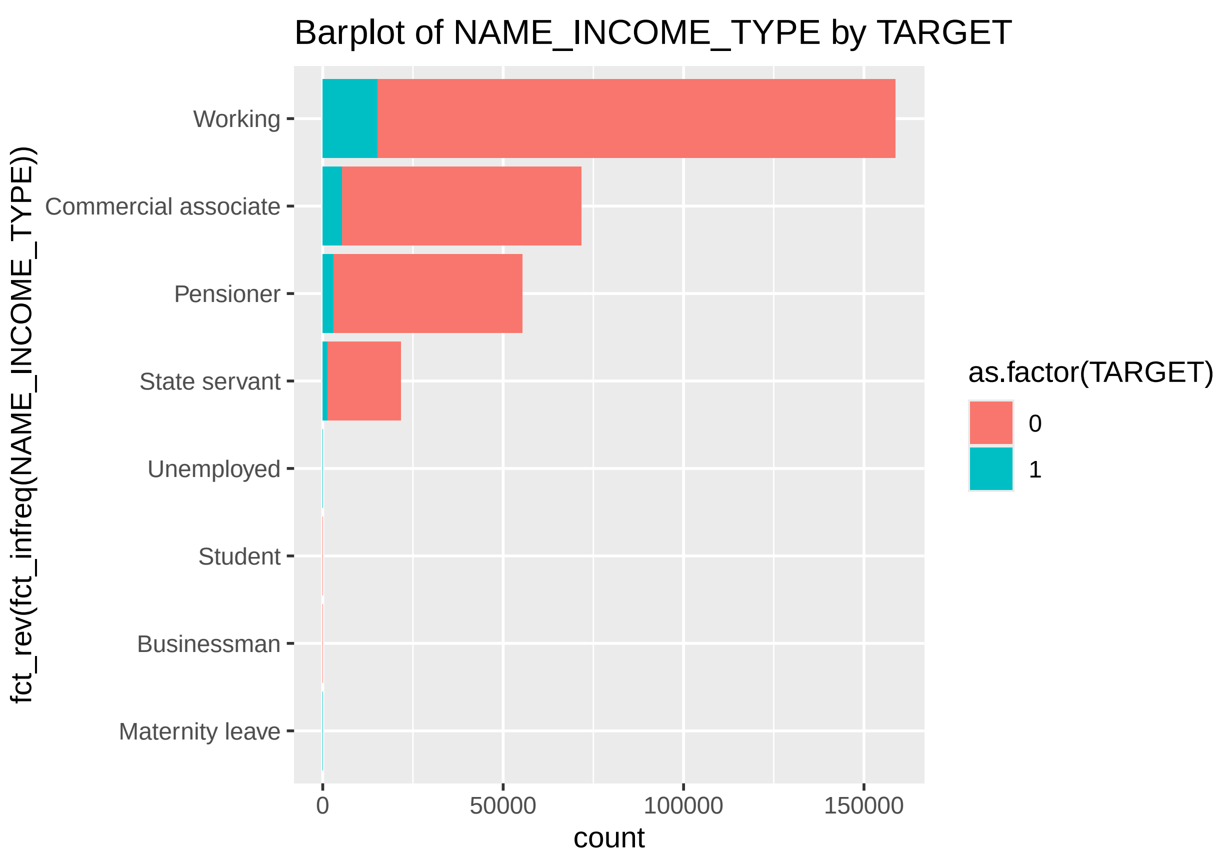
	0	1
Family	0.93	0.07
Group of people	0.92	0.08
Other_A	0.91	0.09
Other_B	0.90	0.10
Spouse, partner	0.92	0.08
Unaccompanied	0.92	0.08

Most clients in the data set were unaccompanied when applying for the loan, and the default rate is not highest in this group.

NAME_INCOME_TYPE

NAME_INCOME_TYPE: Clients income type (businessman, working, maternity leave,…)

```
# NAME_INCOME_TYPE barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(NAME_INCOME_TYPE)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of NAME_INCOME_TYPE by TARGET") +
  coord_flip()
```



```r
# NAME_INCOME_TYPE proportion table
kable(t(HomeCredit_application_train_data_clean %>%
            select(TARGET, NAME_INCOME_TYPE) %>%
            table() %>%
            prop.table(margin = 2) %>%
            round(2)), format = "markdown")
```

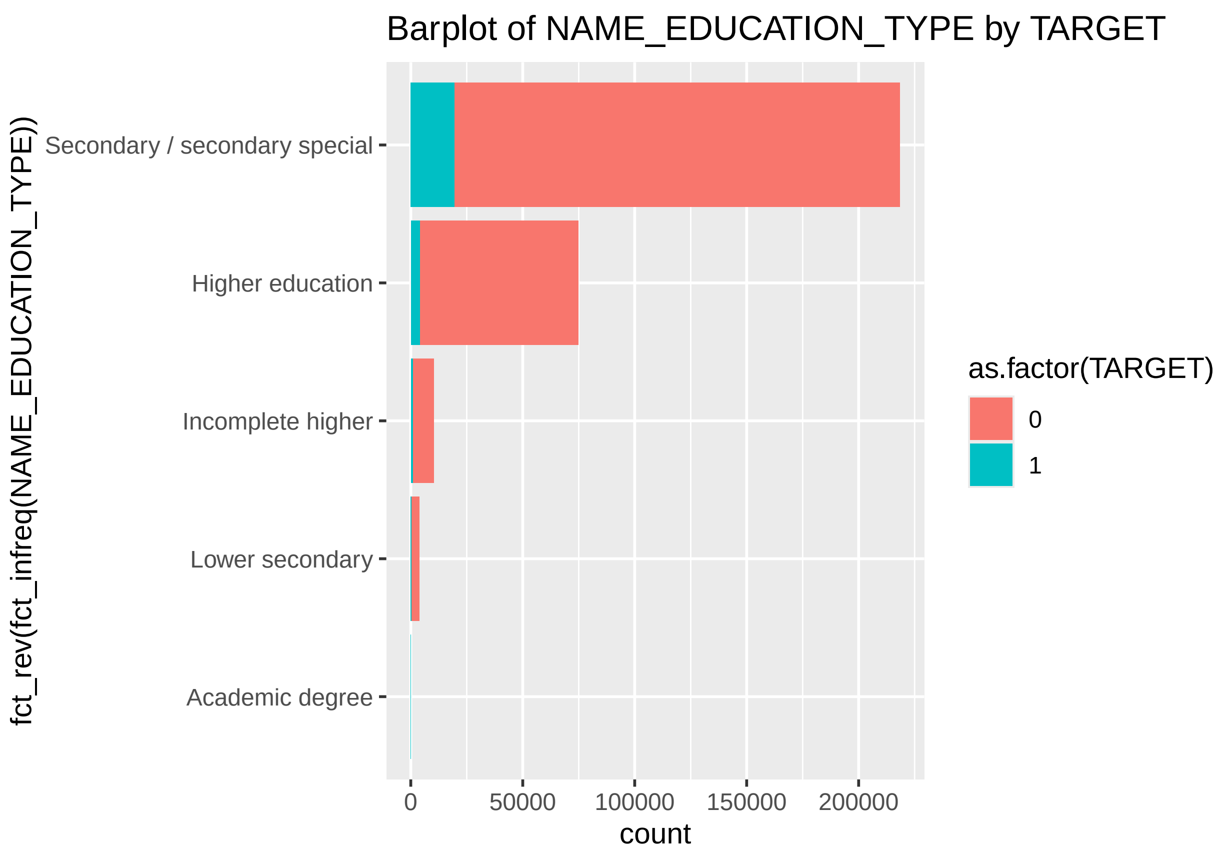
	0	1
Businessman	1.00	0.00
Commercial associate	0.93	0.07
Maternity leave	0.60	0.40
Pensioner	0.95	0.05
State servant	0.94	0.06
Student	1.00	0.00
Unemployed	0.64	0.36
Working	0.90	0.10

Most clients in the data set have a NAME_INCOME_TYPE of "Working", but this group did not have the highest default rate. Clients with a NAME_INCOME_TYPE of "Maternity leave" defaulted 40% of the time and those with a NAME_INCOME_TYPE of "Unemployed" defaulted 36% of the time.

NAME_EDUCATION_TYPE

NAME_EDUCATION_TYPE: Level of highest education the client achieved

```r
# NAME_EDUCATION_TYPE barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(NAME_EDUCATION_TYPE)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of NAME_EDUCATION_TYPE by TARGET") +
  coord_flip()
```




```
# NAME_EDUCATION_TYPE proportion table
kable(t(HomeCredit_application_train_data_clean %>%
            select(TARGET, NAME_EDUCATION_TYPE) %>%
            table() %>%
            prop.table(margin = 2) %>%
            round(2)), format = "markdown")
```

	0	1
Academic degree	0.98	0.02
Higher education	0.95	0.05
Incomplete higher	0.92	0.08
Lower secondary	0.89	0.11
Secondary / secondary special	0.91	0.09

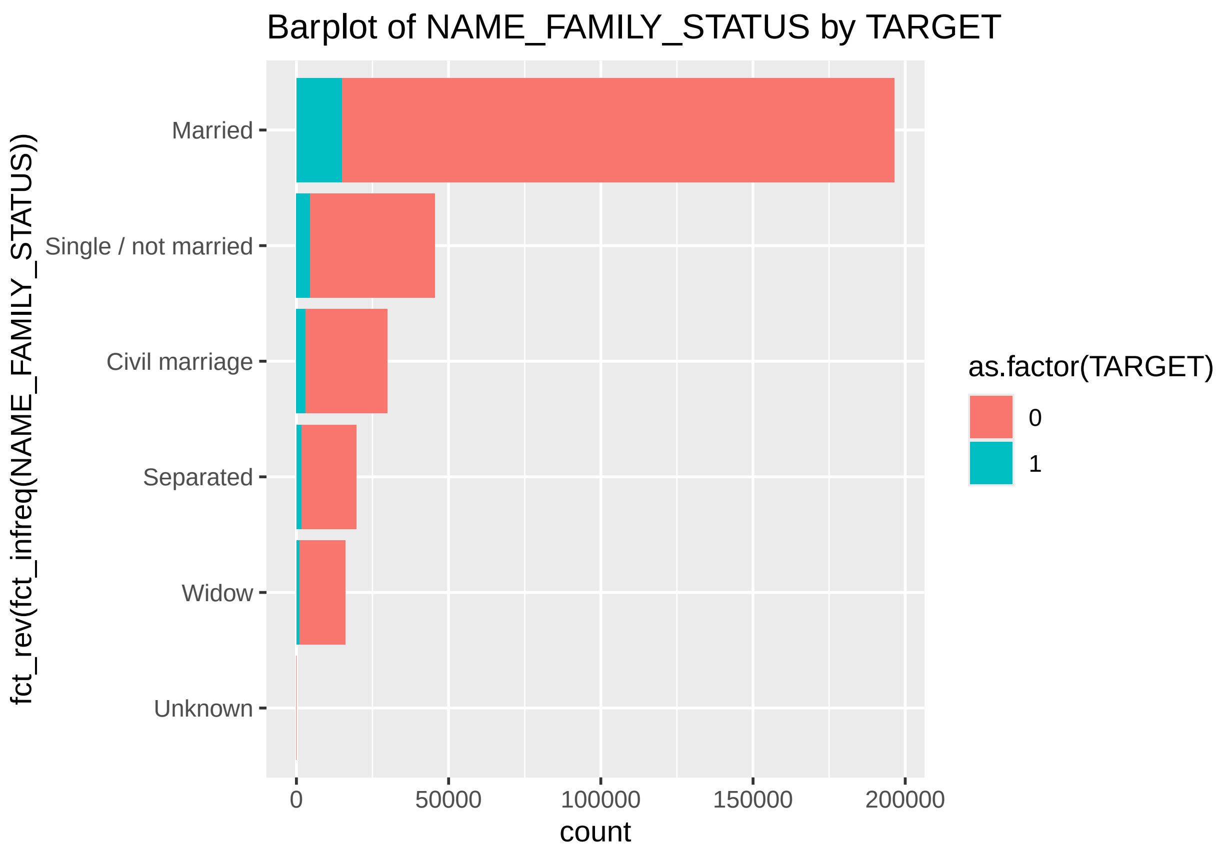
Most clients in the data set have a NAME_EDUCATION_TYPE of "Secondary/ secondary special", but the group with "Lower secondary" defaulted the most often at 11%.

NAME_FAMILY_STATUS

NAME_FAMILY_STATUS: Family status of the client

```
# NAME_FAMILY_STATUS barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(NAME_FAMILY_STATUS)),
```

```
                fill = as.factor(TARGET))) +
  ggtitle("Barplot of NAME_FAMILY_STATUS by TARGET") +
  coord_flip()
```




```
# NAME_FAMILY_STATUS proportion table
kable(t(HomeCredit_application_train_data_clean %>%
            select(TARGET, NAME_FAMILY_STATUS) %>%
            table() %>%
            prop.table(margin = 2) %>%
            round(2)), format = "markdown")
```

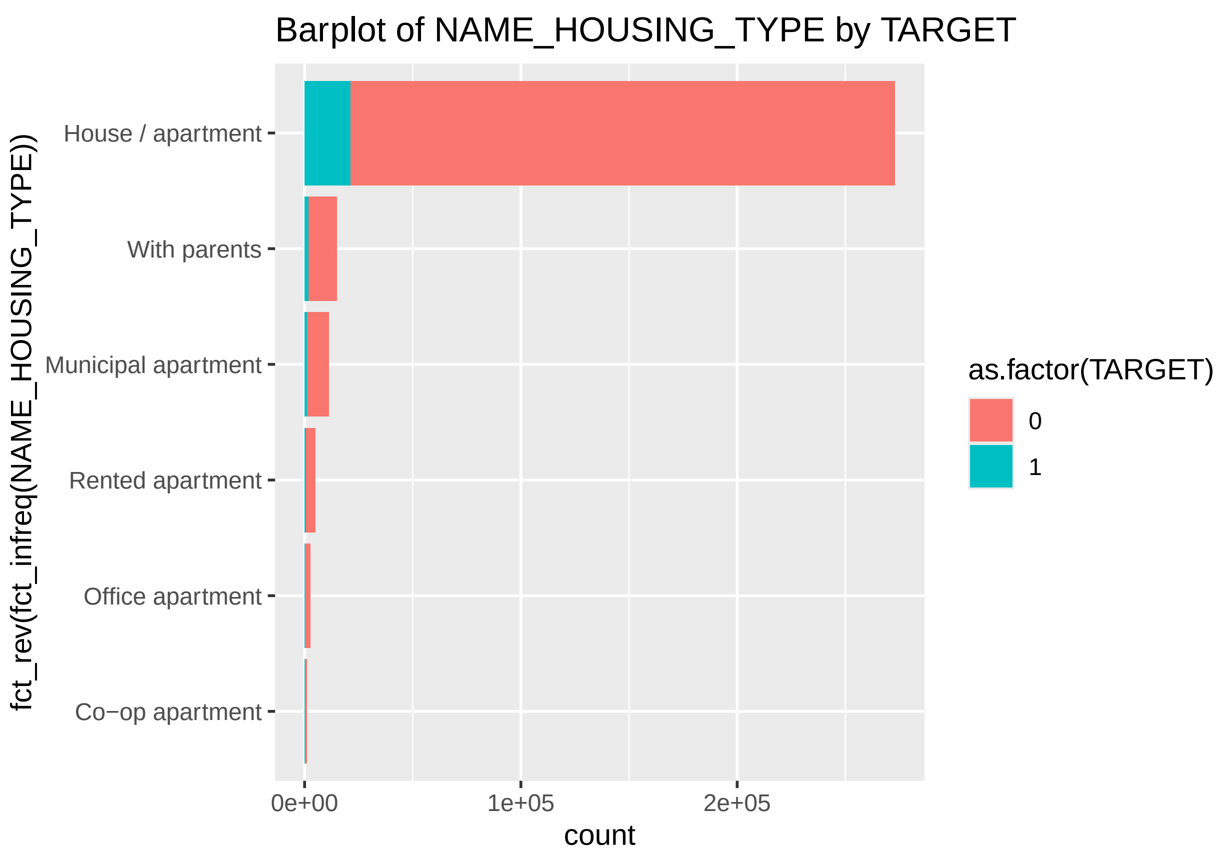
	0	1
Civil marriage	0.90	0.10
Married	0.92	0.08
Separated	0.92	0.08
Single / not married	0.90	0.10
Unknown	1.00	0.00
Widow	0.94	0.06

Most clients in the data set have a NAME_FAMILY_STATUS of "Married". The "Married" and "Civil Marriage" groups had the highest default rates at 10%.

NAME_HOUSING_TYPE

NAME_HOUSING_TYPE: What is the housing situation of the client (renting, living with parents, ...)

```r
# NAME_HOUSING_TYPE barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(NAME_HOUSING_TYPE)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of NAME_HOUSING_TYPE by TARGET") +
  coord_flip()
```




```r
# NAME_HOUSING_TYPE proportion table
kable(t(HomeCredit_application_train_data_clean %>%
          select(TARGET, NAME_HOUSING_TYPE) %>%
          table() %>%
          prop.table(margin = 2) %>%
          round(2)), format = "markdown")
```

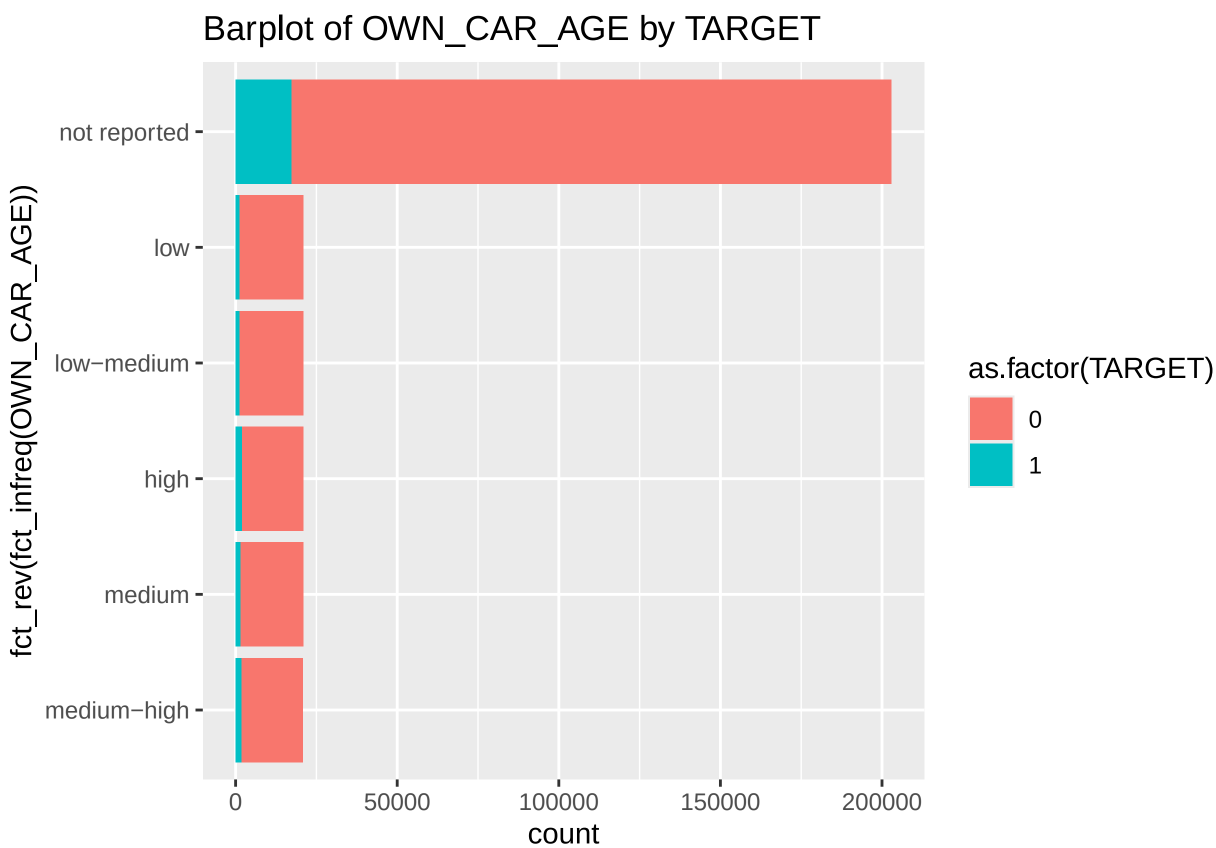
	0	1
Co-op apartment	0.92	0.08
House / apartment	0.92	0.08
Municipal apartment	0.91	0.09
Office apartment	0.93	0.07
Rented apartment	0.88	0.12
With parents	0.88	0.12

Most clients in the data set have a NAME_HOUSING_TYPE of “House/ apartment”, but the “Rented apartment” and “With parents” groups had the highest default rate at 12% each.

OWN_CAR_AGE

OWN_CAR_AGE: Age of client’s car

```
# OWN_CAR_AGE barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(OWN_CAR_AGE)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of OWN_CAR_AGE by TARGET") +
  coord_flip()
```




```
# OWN_CAR_AGE proportion table
kable(t(HomeCredit_application_train_data_clean %>%
            select(TARGET, OWN_CAR_AGE) %>%
            table() %>%
            prop.table(margin = 2) %>%
            round(2)), format = "markdown")
```

	0	1
high	0.91	0.09
low	0.94	0.06

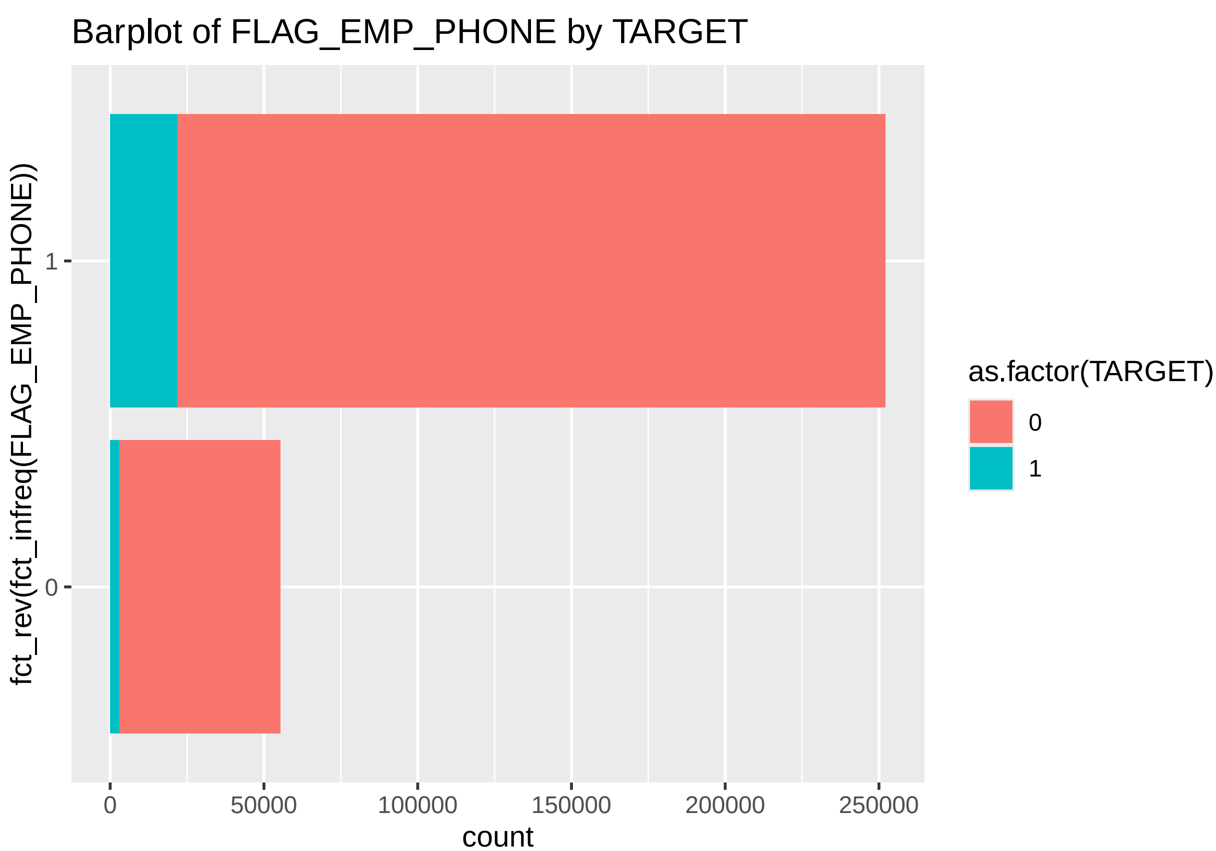
	0	1
low-medium	0.95	0.05
medium	0.93	0.07
medium-high	0.92	0.08
not reported	0.91	0.09

Most clients in the data set did not report an OWN_CAR_AGE. The "high" and "not reported" groups had the highest default rates at 9%.

FLAG_EMP_PHONE

FLAG_EMP_PHONE: Did client provide work phone (1=YES, 0=NO)

```
# FLAG_EMP_PHONE barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(FLAG_EMP_PHONE)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of FLAG_EMP_PHONE by TARGET") +
  coord_flip()
```




```
# FLAG_EMP_PHONE proportion table
kable(t(HomeCredit_application_train_data_clean %>%
          select(TARGET, FLAG_EMP_PHONE) %>%
          table() %>%
```

```
          prop.table(margin = 2) %>%
          round(2)), format = "markdown")
```

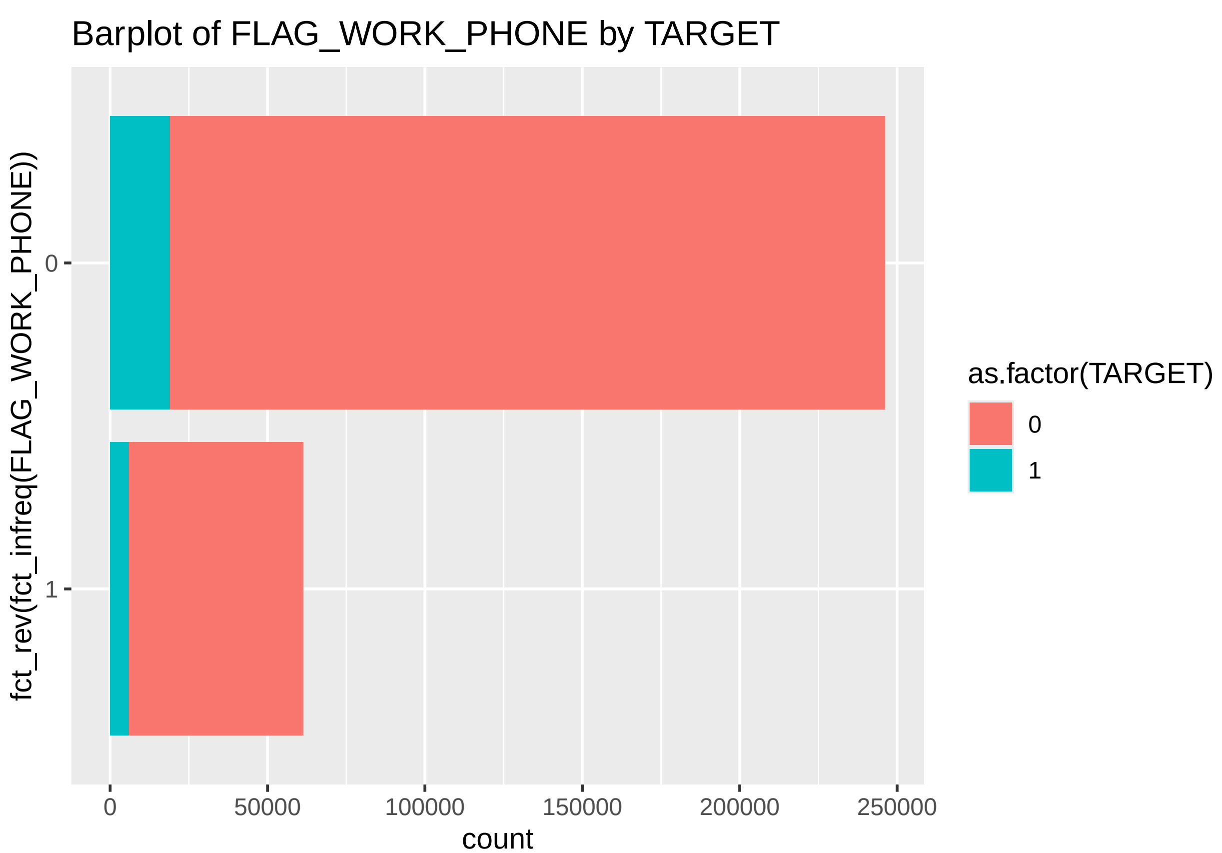
	0	1
0	0.95	0.05
1	0.91	0.09

Most clients in the data set provided a work phone number, this group also has the highest default rate at 9%.

FLAG_WORK_PHONE

FLAG_WORK_PHONE: Did client provide home phone (1=YES, 0=NO)

```
# FLAG_WORK_PHONE barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(FLAG_WORK_PHONE)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of FLAG_WORK_PHONE by TARGET") +
  coord_flip()
```



```r
# FLAG_WORK_PHONE proportion table
kable(t(HomeCredit_application_train_data_clean %>%
            select(TARGET, FLAG_WORK_PHONE) %>%
            table() %>%
            prop.table(margin = 2) %>%
            round(2)), format = "markdown")
```

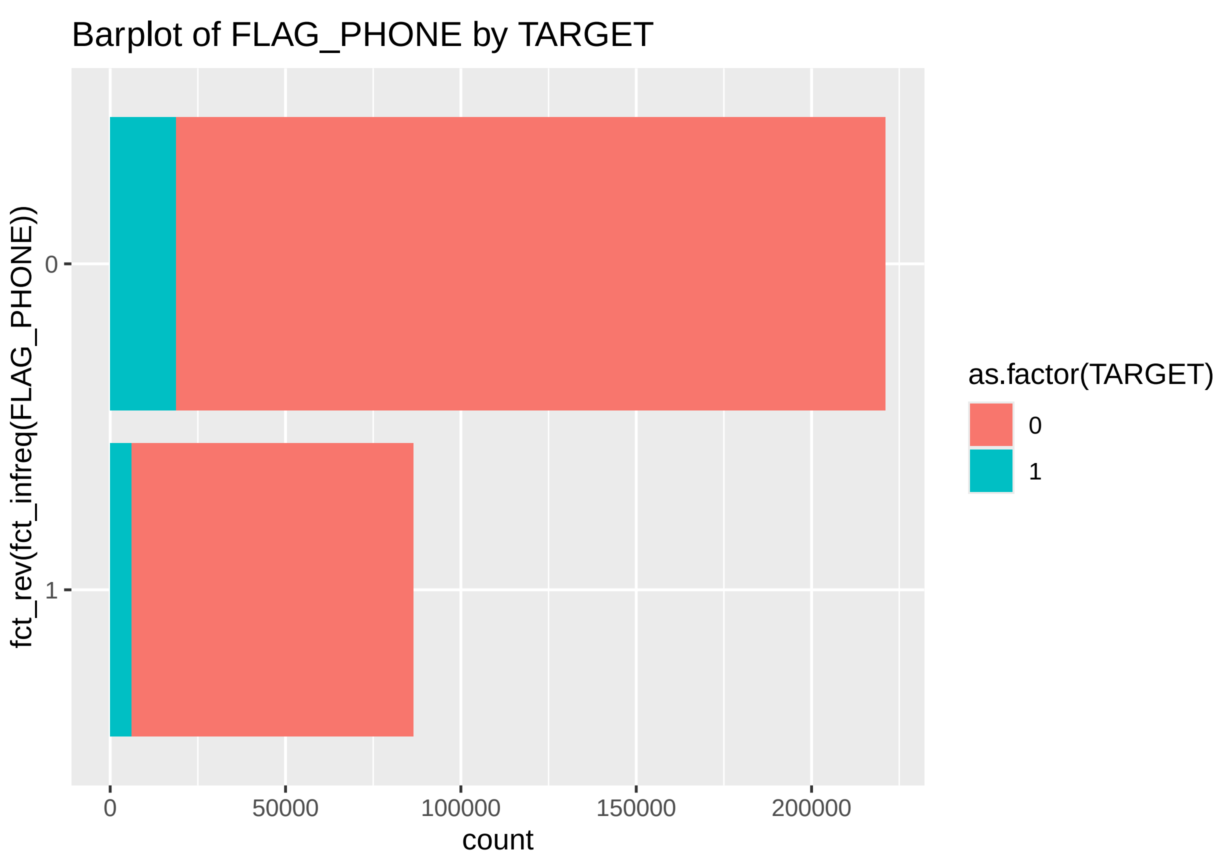
	0	1
0	0.92	0.08
1	0.90	0.10

Most clients in the data set provided a work phone number, this group also has the highest default rate at 10%.

FLAG_PHONE

FLAG_PHONE: Did client provide home phone (1=YES, 0=NO)

```r
# FLAG_PHONE barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(FLAG_PHONE)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of FLAG_PHONE by TARGET") +
  coord_flip()
```



```r
# FLAG_PHONE proportion table
kable(t(HomeCredit_application_train_data_clean %>%
            select(TARGET, FLAG_PHONE) %>%
            table() %>%
            prop.table(margin = 2) %>%
            round(2)), format = "markdown")
```

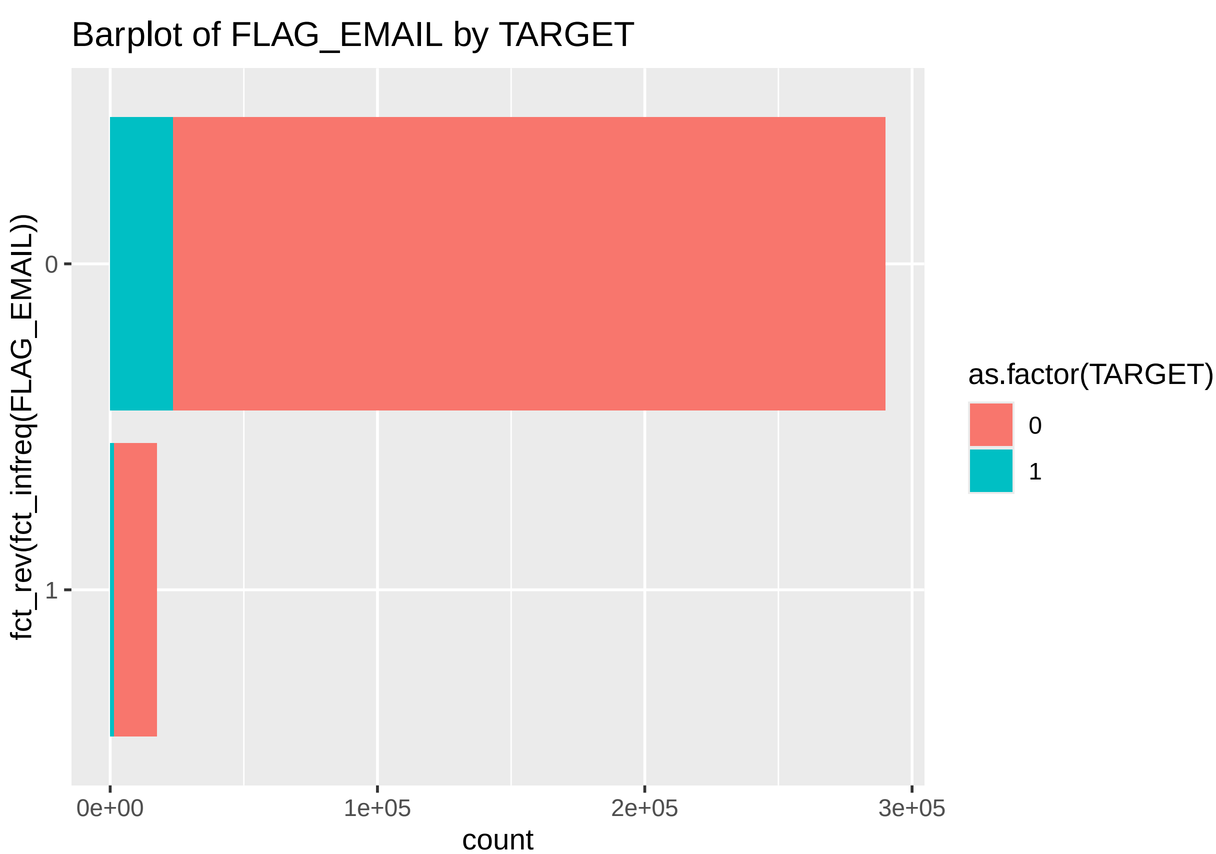
	0	1
0	0.92	0.08
1	0.93	0.07

Most clients in the data set provided a home phone number, this group also has the highest default rate at 8%.

FLAG_EMAIL

FLAG_EMAIL: Did client provide email (1=YES, 0=NO)

```r
# FLAG_EMAIL barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(FLAG_EMAIL)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of FLAG_EMAIL by TARGET") +
  coord_flip()
```



```r
# FLAG_EMAIL proportion table
kable(t(HomeCredit_application_train_data_clean %>%
          select(TARGET, FLAG_EMAIL) %>%
          table() %>%
          prop.table(margin = 2) %>%
          round(2)), format = "markdown")
```

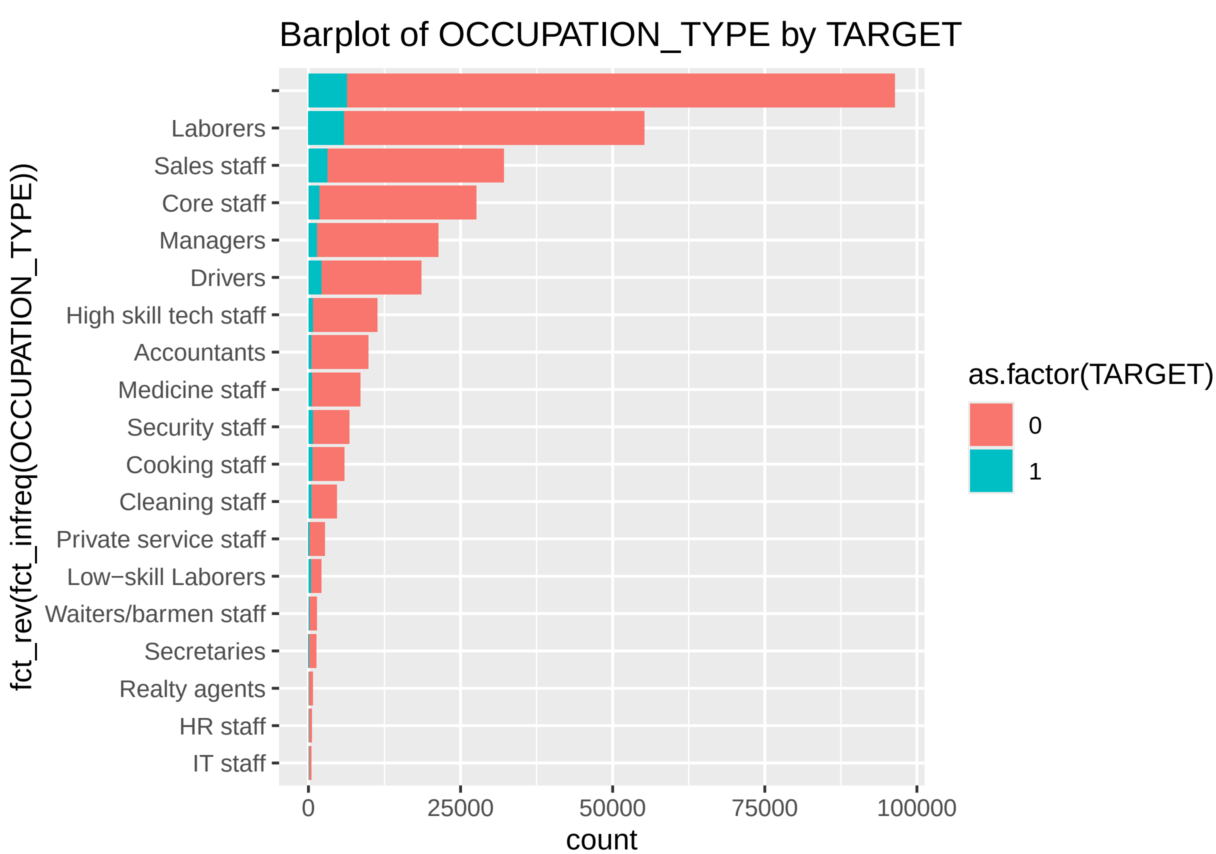
	0	1
0	0.92	0.08
1	0.92	0.08

Most clients in the data set did not provide an email address, but both groups had an equal default rate of 8%.

OCCUPATION_TYPE

OCCUPATION_TYPE: What kind of occupation does the client have

```r
# OCCUPATION_TYPE barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(OCCUPATION_TYPE)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of OCCUPATION_TYPE by TARGET") +
  coord_flip()
```



```r
# OCCUPATION_TYPE proportion table
kable(t(HomeCredit_application_train_data_clean %>%
            select(TARGET, OCCUPATION_TYPE) %>%
            table() %>%
            prop.table(margin = 2) %>%
            round(2)), format = "markdown")
```

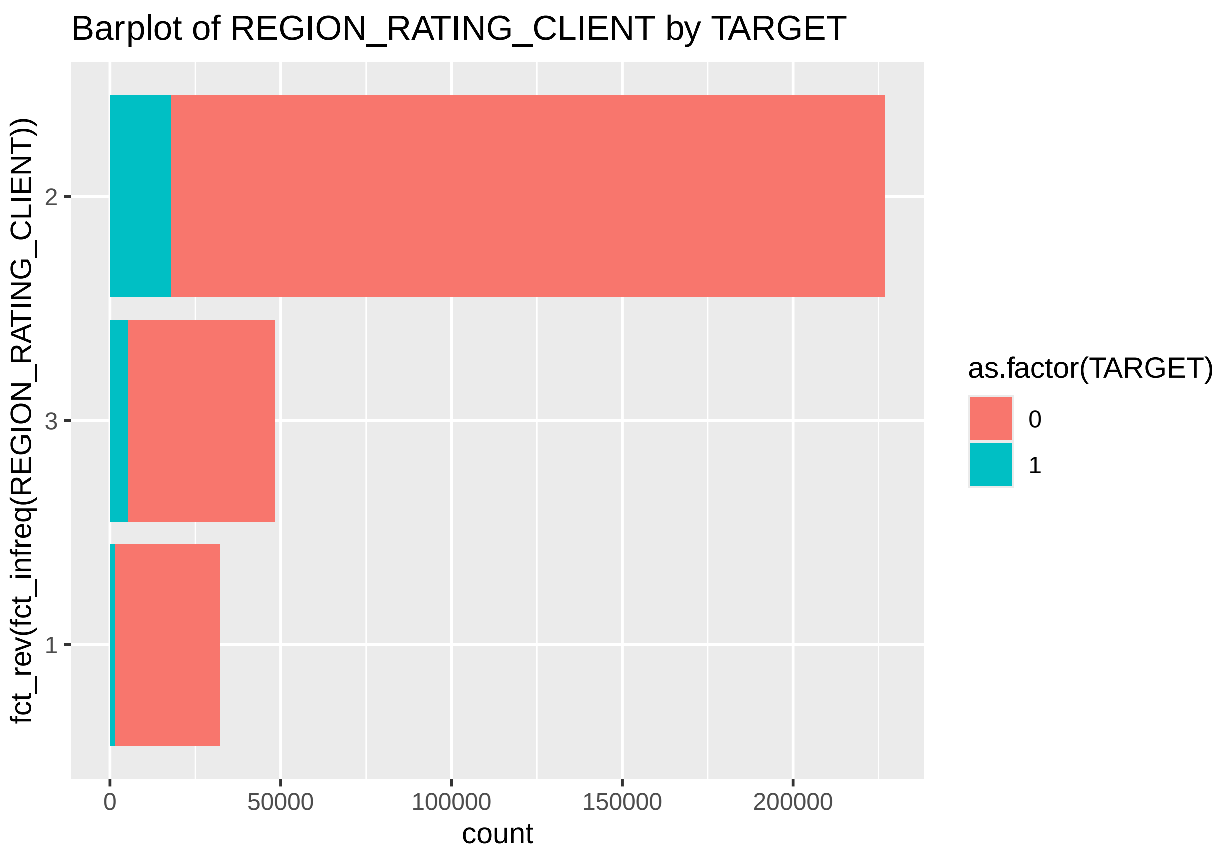
	0	1
	0.93	0.07
Accountants	0.95	0.05
Cleaning staff	0.90	0.10
Cooking staff	0.90	0.10
Core staff	0.94	0.06
Drivers	0.89	0.11
High skill tech staff	0.94	0.06
HR staff	0.94	0.06
IT staff	0.94	0.06
Laborers	0.89	0.11
Low-skill Laborers	0.83	0.17
Managers	0.94	0.06
Medicine staff	0.93	0.07
Private service staff	0.93	0.07
Realty agents	0.92	0.08
Sales staff	0.90	0.10
Secretaries	0.93	0.07
Security staff	0.89	0.11
Waiters/barmen staff	0.89	0.11

The highest default rate was among low-skill laborers.

REGION_RATING_CLIENT

REGION_RATING_CLIENT: Our rating of the region where client lives (1,2,3)

```r
# REGION_RATING_CLIENT barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(REGION_RATING_CLIENT)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of REGION_RATING_CLIENT by TARGET") +
  coord_flip()
```




```r
# REGION_RATING_CLIENT proportion table
kable(t(HomeCredit_application_train_data_clean %>%
          select(TARGET, REGION_RATING_CLIENT) %>%
          table() %>%
          prop.table(margin = 2) %>%
          round(2)), format = "markdown")
```

0	1
0.95	0.05
0.92	0.08
0.89	0.11

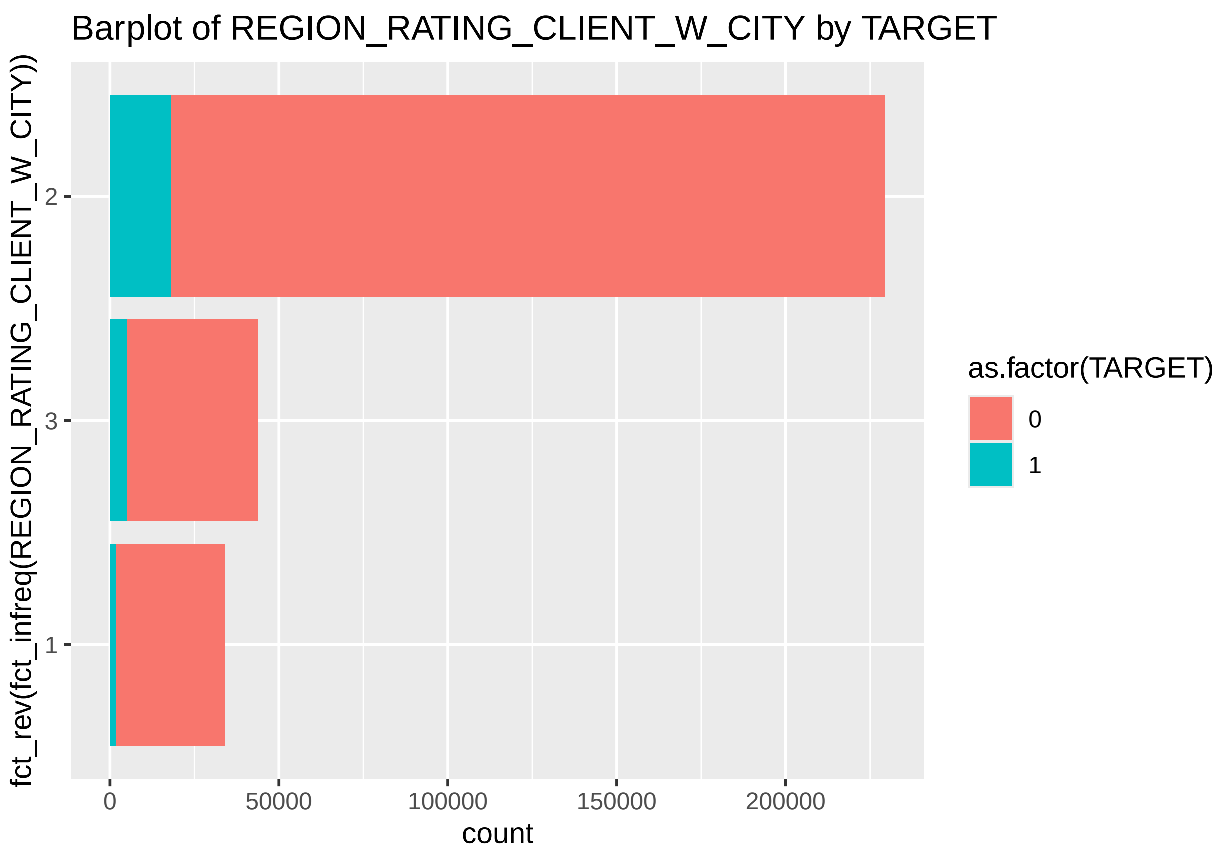
Clients in the REGION_RATING_CLIENT = 2 group had the highest default rate at 11%.

REGION_RATING_CLIENT_W_CITY

REGION_RATING_CLIENT_W_CITY: Our rating of the region where client lives with taking city into account (1,2,3)

```r
# REGION_RATING_CLIENT_W_CITY barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(REGION_RATING_CLIENT_W_CITY)),
               fill = as.factor(TARGET))) +
```

```r
  ggtitle("Barplot of REGION_RATING_CLIENT_W_CITY by TARGET") +
  coord_flip()
```




```r
# REGION_RATING_CLIENT_W_CITY proportion table
kable(t(HomeCredit_application_train_data_clean %>%
          select(TARGET, REGION_RATING_CLIENT_W_CITY) %>%
          table() %>%
          prop.table(margin = 2) %>%
          round(2)), format = "markdown")
```

0	1
0.95	0.05
0.92	0.08
0.89	0.11

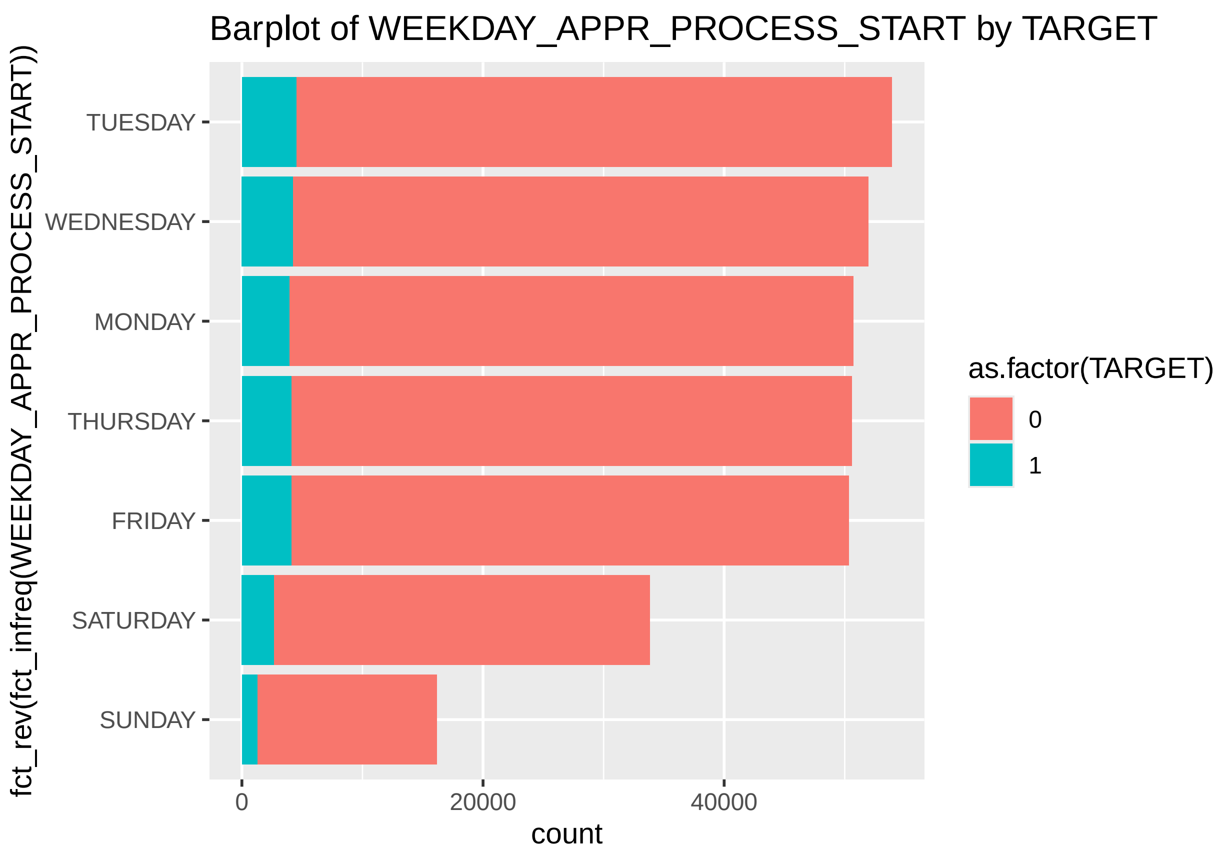
Clients in the REGION_RATING_CLIENT_W_CITY = 2 group had the highest default rate at 11%.

WEEKDAY_APPR_PROCESS_START

WEEKDAY_APPR_PROCESS_START: On which day of the week did the client apply for the loan

```r
# WEEKDAY_APPR_PROCESS_START barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
```

```
  geom_bar(aes(x = fct_rev(fct_infreq(WEEKDAY_APPR_PROCESS_START)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of WEEKDAY_APPR_PROCESS_START by TARGET") +
  coord_flip()
```




```
# WEEKDAY_APPR_PROCESS_START proportion table
kable(t(HomeCredit_application_train_data_clean %>%
          select(TARGET, WEEKDAY_APPR_PROCESS_START) %>%
          table() %>%
          prop.table(margin = 2) %>%
          round(2)), format = "markdown")
```

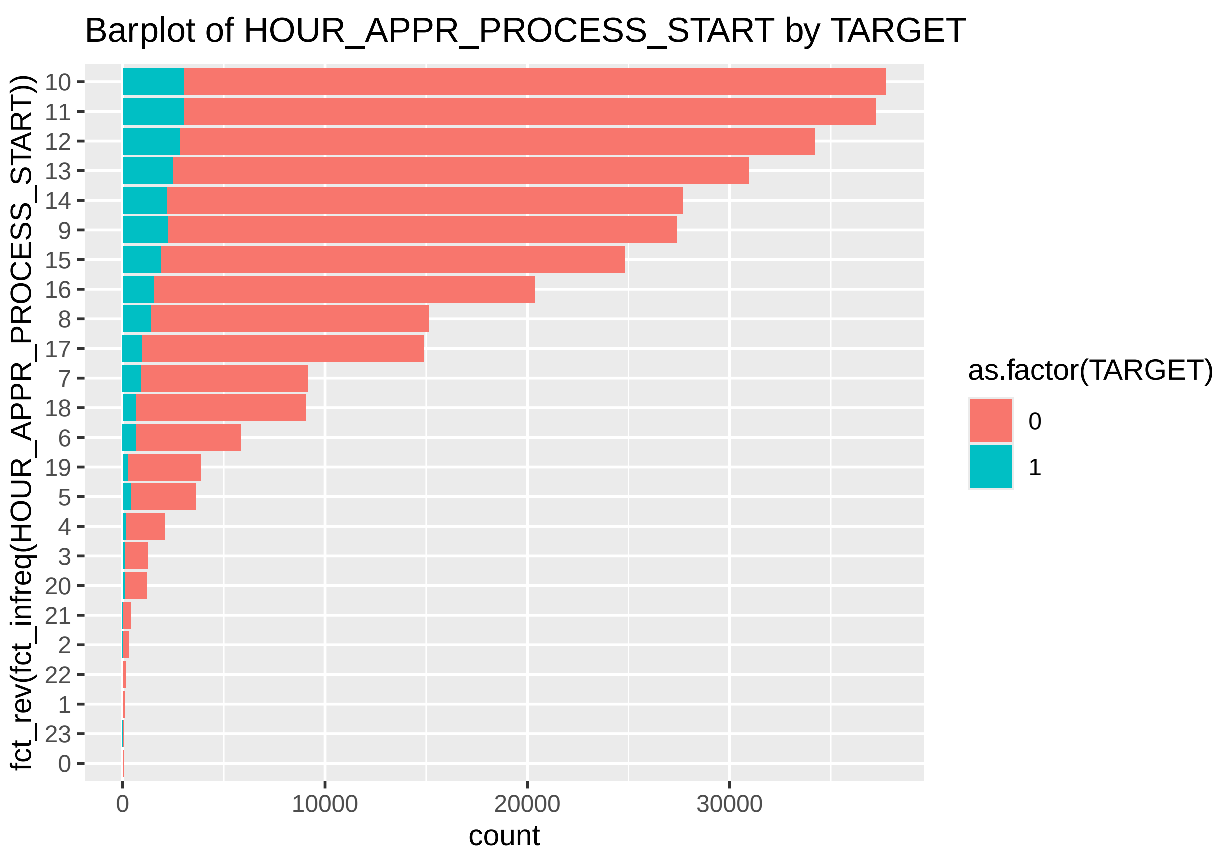
	0	1
FRIDAY	0.92	0.08
MONDAY	0.92	0.08
SATURDAY	0.92	0.08
SUNDAY	0.92	0.08
THURSDAY	0.92	0.08
TUESDAY	0.92	0.08
WEDNESDAY	0.92	0.08

Default rate doesn't really vary among WEEKDAY_APPR_PROCESS_STARTs.

HOUR_APPR_PROCESS_START

HOUR_APPR_PROCESS_START: Approximately at what hour did the client apply for the loan

```
# HOUR_APPR_PROCESS_START barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(HOUR_APPR_PROCESS_START)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of HOUR_APPR_PROCESS_START by TARGET") +
  coord_flip()
```




```
# HOUR_APPR_PROCESS_START proportion table
kable(t(HomeCredit_application_train_data_clean %>%
          select(TARGET, HOUR_APPR_PROCESS_START) %>%
          table() %>%
          prop.table(margin = 2) %>%
          round(2)), format = "markdown")
```

	0	1
0	0.85	0.15
1	0.92	0.08
2	0.90	0.10
3	0.91	0.09
4	0.92	0.08
5	0.89	0.11

	0	1
6	0.89	0.11
7	0.90	0.10
8	0.91	0.09
9	0.92	0.08
10	0.92	0.08
11	0.92	0.08
12	0.92	0.08
13	0.92	0.08
14	0.92	0.08
15	0.92	0.08
16	0.93	0.07
17	0.94	0.06
18	0.93	0.07
19	0.93	0.07
20	0.93	0.07
21	0.94	0.06
22	0.90	0.10
23	0.88	0.12

Applications started in hour 0 had the highest default rate at 15%, but had the fewest applications started in that hour.

REG_REGION_NOT_WORK_REGION

REG_REGION_NOT_WORK_REGION: Flag if client's permanent address does not match work address (1=different, 0=same, at region level)

```
# REG_REGION_NOT_WORK_REGION barplot
HomeCredit_application_train_data_clean %>%
  ggplot() +
  geom_bar(aes(x = fct_rev(fct_infreq(REG_REGION_NOT_WORK_REGION)),
               fill = as.factor(TARGET))) +
  ggtitle("Barplot of REG_REGION_NOT_WORK_REGION by TARGET") +
  coord_flip()
```