PA I Final

EDA

Predictive Final. With the goal of...

Load Data

Prosper data set containing...

```
library(ggplot2)
```

```
Warning: package 'ggplot2' was built under R version 4.4.2
```

```
library(dplyr)
```

```
Warning: package 'dplyr' was built under R version 4.4.2
```

```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(pracma)
```

```
Warning: package 'pracma' was built under R version 4.4.2
```

```
library(corrplot)
```

```
Warning: package 'corrplot' was built under R version 4.4.2

corrplot 0.95 loaded
```

```
library(pROC)
```

```
Type 'citation("pROC")' for a citation.


Attaching package: 'pROC'
```

```
The following objects are masked from 'package:stats':

    cov, smooth, var
```

```
library(coefplot)
```

```
Warning: package 'coefplot' was built under R version 4.4.2
```

```
library(MASS)
```

```

Attaching package: 'MASS'

The following object is masked from 'package:dplyr':

    select
```

```
library(tidyverse)
```

```
Warning: package 'tidyverse' was built under R version 4.4.2

Warning: package 'readr' was built under R version 4.4.1

Warning: package 'forcats' was built under R version 4.4.2

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✓ forcats   1.0.0     ✓ stringr   1.5.1
✓ lubridate 1.9.3     ✓ tibble    3.2.1
✓ purrr     1.0.2     ✓ tidyr     1.3.1
✓ readr     2.1.5

── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ purrr::cross()  masks pracma::cross()
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
✖ MASS::select()  masks dplyr::select()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
errors
```

```
library(tidymodels)
```

```
Warning: package 'tidymodels' was built under R version 4.4.2

── Attaching packages ─────────────────────────────────────── tidymodels 1.2.0 ──
✓ broom        1.0.7      ✓ rsample      1.2.1
✓ dials        1.3.0      ✓ tune         1.2.1
✓ infer        1.0.7      ✓ workflows    1.1.4
✓ modeldata    1.4.0      ✓ workflowsets 1.1.0
```

```
✓ parsnip      1.2.1      ✓ yardstick    1.3.2
✓ recipes      1.0.10
Warning: package 'broom' was built under R version 4.4.2

Warning: package 'dials' was built under R version 4.4.2

Warning: package 'infer' was built under R version 4.4.2

Warning: package 'modeldata' was built under R version 4.4.2

Warning: package 'parsnip' was built under R version 4.4.2

Warning: package 'rsample' was built under R version 4.4.2

Warning: package 'tune' was built under R version 4.4.2

Warning: package 'workflows' was built under R version 4.4.2

Warning: package 'workflowsets' was built under R version 4.4.2

Warning: package 'yardstick' was built under R version 4.4.2

── Conflicts ──────────────────────────────────────── tidymodels_conflicts() ──
✖ purrr::cross()    masks pracma::cross()
✖ scales::discard() masks purrr::discard()
✖ dplyr::filter()   masks stats::filter()
✖ recipes::fixed()  masks stringr::fixed()
✖ dplyr::lag()      masks stats::lag()
✖ MASS::select()    masks dplyr::select()
✖ yardstick::spec() masks readr::spec()
✖ recipes::step()   masks stats::step()
• Learn how to get started at https://www.tidymodels.org/start/
```

```
library(caret)
```

```
Warning: package 'caret' was built under R version 4.4.2

Loading required package: lattice

Attaching package: 'caret'

The following objects are masked from 'package:yardstick':

    precision, recall, sensitivity, specificity

The following object is masked from 'package:purrr':

    lift
```

```
library(explore)
```

```
Warning: package 'explore' was built under R version 4.4.2
```

```
library(outliers)
library(tidyr)
library(stats)
library(Rprofet)
```

```
Warning: package 'Rprofet' was built under R version 4.4.2
```

```
# library(earth)
library(faraway)
```

```
Warning: package 'faraway' was built under R version 4.4.2


Attaching package: 'faraway'

The following object is masked from 'package:lattice':

    melanoma

The following object is masked from 'package:pracma':

    logit
```

```
library(lattice)
library(gmodels)
```

```
Warning: package 'gmodels' was built under R version 4.4.2


Attaching package: 'gmodels'

The following object is masked from 'package:pROC':

    ci
```

```
library(gains)
library(maps)
```

```
Warning: package 'maps' was built under R version 4.4.2


Attaching package: 'maps'

The following object is masked from 'package:faraway':

    ozone
```

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

    map
```

```
library(mapproj)
```

```
Warning: package 'mapproj' was built under R version 4.4.2
```

```
prosper_data <- read.csv("C:/Users/logan/OneDrive - South Dakota State University - SDSU/Archive/S
```

EDA

1. View Structure of dataset
    1. 25,606 NA's (227,844 total observations within dataset) (0.112% of dataset is missing)
    2. Bad rate: 4952/18987 = 0.26081%
2. Make a subset for model input variables
    1. Input variables are: "AmountRemaining", "BorrowerCity", "BorrowerState", "DebtToIncomeRatio", "IsBorrowerHomeowner", "ListingKey", "ListingNumber", "MemberKey", "LoanKey", "CurrentDelinquencies", "DelinquenciesLast7Years", "PublicRecordsLast10Years", "FirstRecordedCreditLine", "TotalCreditLines", "InquiriesLast6Months", "AmountDelinquent", "PublicRecordsLast12Months", "CurrentCreditLines", "OpenCreditLines", "RevolvingCreditBalance", "BankcardUtilization", "EmploymentStatus", "LengthStatusMonths", "Income", "BorrowerOccupation"
3. FirstRecordedCreditLine
    1. Extract year from FirstRecordedCreditLine by formatting it from "yyyy-mm-ddThh:mm:ss" to "yyyy"
4. Change IsBorrowerHomeowner from "TRUE/FALSE" to "0/1"
5. BorrowerOccupation
    1. Combine "Profession that is not part of t" and "Professional" into "Other"
    2. Bin Occupation into Small, Medium, Large bins??
    3. 57/300 students bad
        1. Student College - Freshman: 6/15 Bad
        2. Student College - Sophomore: 11/33 Bad
        3. Student College - Junior: 12/61 Bad
        4. Student College - Senior: 12/85 Bad

5. Student College - Grad School: 9/84 Bad

6. Student College - Technical School: 3/10 Bad

7. Student College - Community College: 4/12 Bad

```
#1.View Structure of dataset
names(prosper_data)
```

```
 [1] "AmountRemaining"           "BorrowerCity"
 [3] "BorrowerState"             "DebtToIncomeRatio"
 [5] "IsBorrowerHomeowner"       "ListingKey"
 [7] "ListingNumber"             "MemberKey"
 [9] "AmountBorrowed"            "LoanKey"
[11] "CurrentDelinquencies"      "DelinquenciesLast7Years"
[13] "PublicRecordsLast10Years"  "FirstRecordedCreditLine"
[15] "TotalCreditLines"          "InquiriesLast6Months"
[17] "AmountDelinquent"          "PublicRecordsLast12Months"
[19] "CurrentCreditLines"        "OpenCreditLines"
[21] "RevolvingCreditBalance"    "BankcardUtilization"
[23] "EmploymentStatus"          "LengthStatusMonths"
[25] "Income"                    "BorrowerOccupation"
[27] "LpStatus"                  "DPD"
[29] "PrincipalBalance"          "Bad"
```

```
summary(prosper_data)
```

```
 AmountRemaining BorrowerCity       BorrowerState      DebtToIncomeRatio
 Min.   :0       Length:18987       Length:18987       Min.   : 0.0000
 1st Qu.:0       Class :character   Class :character   1st Qu.: 0.1300
 Median :0       Mode  :character   Mode  :character   Median : 0.2000
 Mean   :0                                             Mean   : 0.3441
 3rd Qu.:0                                             3rd Qu.: 0.3200
 Max.   :0                                             Max.   :10.0100
                                                       NA's   :659
 IsBorrowerHomeowner  ListingKey         ListingNumber     MemberKey
 Mode :logical       Length:18987       Min.   : 16558   Length:18987
 FALSE:10560         Class :character   1st Qu.:107487   Class :character
 TRUE :8427          Mode  :character   Median :200535   Mode  :character
                                        Mean   :199731
                                        3rd Qu.:295787
                                        Max.   :375669

 AmountBorrowed    LoanKey           CurrentDelinquencies
 Min.   : 1000   Length:18987       Min.   : 0.000
 1st Qu.: 2551   Class :character   1st Qu.: 0.000
 Median : 5000   Mode  :character   Median : 0.000
 Mean   : 6476                      Mean   : 1.423
 3rd Qu.: 8000                      3rd Qu.: 1.000
 Max.   :25000                      Max.   :64.000
```

```
                                  NA's   :61
DelinquenciesLast7Years PublicRecordsLast10Years FirstRecordedCreditLine
Min.   : 0.000          Min.   : 0.000           Length:18987
1st Qu.: 0.000          1st Qu.: 0.000           Class :character
Median : 0.000          Median : 0.000           Mode  :character
Mean   : 6.197          Mean   : 0.433
3rd Qu.: 7.000          3rd Qu.: 1.000
Max.   :99.000          Max.   :30.000
NA's   :61              NA's   :61
TotalCreditLines InquiriesLast6Months AmountDelinquent
Min.   :  2.00   Min.   :  0.000      Min.   :     0.0
1st Qu.: 14.00   1st Qu.:  0.000      1st Qu.:     0.0
Median : 22.00   Median :  2.000      Median :     0.0
Mean   : 24.19   Mean   :  2.893      Mean   :  1281.9
3rd Qu.: 33.00   3rd Qu.:  4.000      3rd Qu.:   120.8
Max.   :129.00   Max.   :105.000      Max.   :444745.0
NA's   :61       NA's   :61           NA's   :4117
PublicRecordsLast12Months CurrentCreditLines OpenCreditLines
Min.   :0.000             Min.   : 0.000     Min.   : 0.000
1st Qu.:0.000             1st Qu.: 5.000     1st Qu.: 4.000
Median :0.000             Median : 8.000     Median : 7.000
Mean   :0.043             Mean   : 9.449     Mean   : 8.102
3rd Qu.:0.000             3rd Qu.:13.000     3rd Qu.:11.000
Max.   :7.000             Max.   :52.000     Max.   :48.000
NA's   :4105              NA's   :4105       NA's   :4105
RevolvingCreditBalance BankcardUtilization EmploymentStatus
Min.   :      0        Min.   :0.000       Length:18987
1st Qu.:   1162        1st Qu.:0.230       Class :character
Median :   5023        Median :0.630       Mode  :character
Mean   :  15914        Mean   :0.568
3rd Qu.:  15146        3rd Qu.:0.890
Max.   :1435667        Max.   :5.950
NA's   :4105           NA's   :4105
LengthStatusMonths     Income      BorrowerOccupation   LpStatus
Min.   :  0.00     Min.   :0.000   Length:18987       Length:18987
1st Qu.:  0.00     1st Qu.:2.000   Class :character   Class :character
Median :  0.00     Median :3.000   Mode  :character   Mode  :character
Mean   : 22.79     Mean   :2.795
3rd Qu.: 15.00     3rd Qu.:4.000
Max.   :554.00     Max.   :7.000

     DPD         PrincipalBalance      Bad
Min.   :  0.00   Min.   :    0    Min.   :0.0000
1st Qu.:  0.00   1st Qu.: 1514    1st Qu.:0.0000
Median :  0.00   Median : 2752    Median :0.0000
Mean   : 62.71   Mean   : 3917    Mean   :0.2608
3rd Qu.: 47.00   3rd Qu.: 4942    3rd Qu.:1.0000
Max.   :516.00   Max.   :25000    Max.   :1.0000
```

```
str(prosper_data)
```

```
'data.frame':   18987 obs. of  30 variables:
 $ AmountRemaining          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ BorrowerCity             : chr  "" "" "" "" ...
 $ BorrowerState            : chr  "HI" "ID" "CA" "MD" ...
 $ DebtToIncomeRatio        : num  NA 0.49 0.93 0.37 0.27 0.01 0.19 0.09 0.54 0.26 ...
 $ IsBorrowerHomeowner      : logi  TRUE TRUE FALSE FALSE FALSE FALSE ...
 $ ListingKey               : chr  "C7483419663390942D38AC4" "61E933658067906368B91C1"
"1EB33391336918886852BF6" "1254342491044057712 7E4F" ...
 $ ListingNumber            : int  324575 32910 148076 353292 293990 345240 117853 39417 262045
308313 ...
 $ MemberKey                : chr  "E8E53419992364577414546" "8EBF3364961007766E2B997"
"A4A23383316370317A81F8C" "A9493423025490065F2C7E7" ...
 $ AmountBorrowed           : num  2000 2000 9900 2700 5000 1500 5000 9000 3000 15000 ...
 $ LoanKey                  : chr  "00013421083473792D70F75" "000B3366346245964D6187E"
"000E3392089465002A7DBA0" "1.33E+20" ...
 $ CurrentDelinquencies     : int  0 0 0 0 0 1 4 8 0 0 ...
 $ DelinquenciesLast7Years  : int  0 0 0 15 0 0 13 24 0 0 ...
 $ PublicRecordsLast10Years : int  0 1 0 1 0 1 0 0 0 0 ...
 $ FirstRecordedCreditLine  : chr  "1998-02-25T00:00:00" "1997-04-01T00:00:00" "1988-09-
08T00:00:00" "1990-03-22T00:00:00" ...
 $ TotalCreditLines         : int  31 32 30 52 4 14 33 21 9 28 ...
 $ InquiriesLast6Months     : int  3 2 1 1 2 1 4 5 1 0 ...
 $ AmountDelinquent         : int  0 NA 0 0 0 476 5720 NA 0 0 ...
 $ PublicRecordsLast12Months: int  0 NA 0 0 0 0 0 NA 0 0 ...
 $ CurrentCreditLines       : int  10 NA 22 19 3 0 10 NA 9 15 ...
 $ OpenCreditLines          : int  7 NA 19 18 3 0 9 NA 9 15 ...
 $ RevolvingCreditBalance   : int  2792 NA 28861 15093 832 0 2664 NA 6942 26186 ...
 $ BankcardUtilization      : num  0.22 NA 0.84 0.88 0.36 0 0.39 NA 0.62 0.41 ...
 $ EmploymentStatus         : chr  "Full-time" "Not available" "Full-time" "Full-time" ...
 $ LengthStatusMonths       : int  0 0 44 0 7 0 43 0 0 0 ...
 $ Income                   : int  1 0 2 5 2 4 3 0 2 4 ...
 $ BorrowerOccupation       : chr  "Military Enlisted" "Profession that is not part of t"
"Teacher" "Professional" ...
 $ LpStatus                 : chr  "Current" "2 months late" "Current" "Current" ...
 $ DPD                      : int  0 77 0 0 0 0 0 0 0 0 ...
 $ PrincipalBalance         : num  948 1359 5725 1531 2784 ...
 $ Bad                      : int  0 1 0 0 0 0 0 0 0 0 ...
```

```
class(prosper_data)
```

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

```
#2.Make a subset for model input variables
input_vars <- prosper_data[, c("AmountRemaining", "BorrowerCity", "BorrowerState", "DebtToIncomeRa
# Create ID
```

```
input_vars$ID <- c(1:nrow(input_vars))
sapply(input_vars, function(x) sum(is.na(x)))
```

```
         AmountRemaining             BorrowerCity            BorrowerState
                       0                        0                        0
       DebtToIncomeRatio       IsBorrowerHomeowner                ListingKey
                     659                        0                        0
           ListingNumber                 MemberKey             AmountBorrowed
                       0                        0                        0
                LoanKey      CurrentDelinquencies   DelinquenciesLast7Years
                       0                       61                       61
PublicRecordsLast10Years   FirstRecordedCreditLine         TotalCreditLines
                      61                        0                       61
    InquiriesLast6Months          AmountDelinquent PublicRecordsLast12Months
                      61                     4117                     4105
      CurrentCreditLines          OpenCreditLines    RevolvingCreditBalance
                    4105                     4105                     4105
     BankcardUtilization          EmploymentStatus       LengthStatusMonths
                    4105                        0                        0
                  Income       BorrowerOccupation                      Bad
                       0                        0                        0
                      ID
                       0
```

```
# 3.FirstRecordedCreditLine (format)
format_date <- function(date_column) {
  return(substr(date_column, 1, 4))
}
input_vars$FirstRecordedCreditLine <- format_date(input_vars$FirstRecordedCreditLine)

# 4.Change IsBorrowerHomeowner from "TRUE/FALSE" to "0/1"
input_vars$IsBorrowerHomeowner <- ifelse(
  input_vars$IsBorrowerHomeowner == TRUE, 1, 0)

# write.csv(input_vars, file = "C:\\Users\\logan\\OneDrive - South Dakota State University - SDSU

# 5.BorrowerOccupation
input_vars$BorrowerOccupation <- ifelse(input_vars$BorrowerOccupation %in% c("Profession that is

# Categorization based on occupation
input_vars <- input_vars %>%
  mutate(BorrowerOccupation = case_when(
    BorrowerOccupation %in% c("Nurse's Aide", "Nurse - Licensed Practical Nurse",
                              "Nurse - Registered Nurse (RN)", "Medical Technician",
                              "Dentist", "Doctor", "Pharmacist") ~ "Medical and Healthcare",
    BorrowerOccupation %in% c("Engineer - Chemical", "Engineer - Electrical",
                              "Engineer - Mechanical") ~ "Engineering",
    BorrowerOccupation %in% c("Teacher", "Professor", "Teacher's Aide",
                              startsWith(BorrowerOccupation, "Student")) ~ "Education",
```

```
    BorrowerOccupation %in% c("Accountant/CPA", "Investor", "Executive") ~ "Business and Finance",
    BorrowerOccupation %in% c("Civil Service", "Military Enlisted", "Military Officer",
                              "Police Officer/Correction Office", "Fireman", "Postal Service") ~ "
    BorrowerOccupation %in% c("Computer Programmer", "Scientist", "Chemist", "Biologist") ~ "Scie
    BorrowerOccupation %in% c("Attorney", "Secretary/Administrative Assista", "Clerical") ~ "Lega
    BorrowerOccupation %in% c("Tradesman - Carpenter", "Tradesman - Electrician",
                              "Tradesman - Mechanic", "Tradesman - Plumber", "Construction") ~ "Tr
    BorrowerOccupation %in% c("Food Service", "Food Service Management",
                              "Retail Management", "Sales - Commission", "Sales - Retail",
                              "Waiter/Waitress") ~ "Service Industry",
    TRUE ~ "Other"
  ))
```

Bin Data

ListingKey: An ID to uniquely identify each loan or listing, and distinguish it from others.

ListingNumber: It may serve a similar purpose as ListingKey, providing a unique identifier for each listing, but the exact difference likely is dependent on a company specific database query structure.

MemberKey: A unique identifier for each member or borrower on the lending platform.

LoanKey: Similar to ListingKey, LoanKey is a unique identifier for each loan or loan application on the platform. It helps differentiate one loan from another.

```
#Use BinProfet function to bin the data
binData = BinProfet(data = input_vars, id = "ID", target = "Bad", num.bins = 12)

names(binData)
```

```
 [1] "ID"                          "Bad"
 [3] "AmountRemaining_Bins"        "BorrowerCity_Bins"
 [5] "BorrowerState_Bins"          "DebtToIncomeRatio_Bins"
 [7] "IsBorrowerHomeowner_Bins"    "ListingKey_Bins"
 [9] "ListingNumber_Bins"          "MemberKey_Bins"
[11] "AmountBorrowed_Bins"         "LoanKey_Bins"
[13] "CurrentDelinquencies_Bins"   "DelinquenciesLast7Years_Bins"
[15] "PublicRecordsLast10Years_Bins" "FirstRecordedCreditLine_Bins"
[17] "TotalCreditLines_Bins"       "InquiriesLast6Months_Bins"
[19] "AmountDelinquent_Bins"       "PublicRecordsLast12Months_Bins"
[21] "CurrentCreditLines_Bins"     "OpenCreditLines_Bins"
[23] "RevolvingCreditBalance_Bins" "BankcardUtilization_Bins"
[25] "EmploymentStatus_Bins"       "LengthStatusMonths_Bins"
[27] "Income_Bins"                 "BorrowerOccupation_Bins"
```

```
summary(binData)
```

```
      ID             Bad         AmountRemaining_Bins   BorrowerCity_Bins
Min.   :    1   Min.   :0.0000   0:18987                        :15043
1st Qu.: 4748   1st Qu.:0.0000                        CHICAGO    :   96
Median : 9494   Median :0.0000                        ATLANTA    :   60
Mean   : 9494   Mean   :0.2608                        HOUSTON    :   51
3rd Qu.:14240   3rd Qu.:1.0000                        LOS ANGELES:   49
Max.   :18987   Max.   :1.0000                        SAN DIEGO  :   47
                                                      (Other)    : 3641
BorrowerState_Bins DebtToIncomeRatio_Bins IsBorrowerHomeowner_Bins
CA     :3315       [0.17,0.2) :1897       0:10560
IL     :1403       [0.23,0.27):1832       1: 8427
GA     :1378       [0.27,0.32):1805
TX     :1165       [0.14,0.17):1785
FL     :1020       [0.07,0.11):1677
MI     : 830       [0,0.07)   :1607
(Other):9876       (Other)    :8384
ListingKey_Bins        ListingNumber_Bins                MemberKey_Bins
1.61E+22:    2   [16558,43506)  :1582     4.77E+22               :    3
2.73E+22:    2   [43506,79869)  :1582     6.63E+22               :    3
3.50E+22:    2   [79869,107482) :1582     002B33814311723535519BD:    2
3.75E+22:    2   [107482,133064):1582     006C3373804016872128132:    2
3.93E+22:    2   [133064,163846):1582     00C43387968070538859D91:    2
5.13E+22:    2   [163846,200491):1582     01A43393831401155034A50:    2
(Other) :18975   (Other)        :9495     (Other)                :18973
   AmountBorrowed_Bins   LoanKey_Bins   CurrentDelinquencies_Bins
[5050,7070)  :2012     1.19E+22:    2   [0,1)  :11971
[1201,2025)  :1987     1.34E+22:    2   [1,2)  : 2490
[5000,5050)  :1885     1.79E+22:    2   [2,3)  : 1237
[3002,4001)  :1809     2.59E+22:    2   [3,4)  :  743
[9090,12015) :1743     2.60E+22:    2   [6,8)  :  556
[12015,20250):1652     2.94E+22:    2   [4,5)  :  546
(Other)      :7899     (Other) :18975   (Other): 1444
DelinquenciesLast7Years_Bins PublicRecordsLast10Years_Bins
[0,1)  :10579                [0,1)   :13454
[3,5)  : 1251                [1,2)   : 4021
[7,10) : 1035                [2,3)   :  851
[13,18):  872                [3,4)   :  337
[1,2)  :  830                [4,6)   :  189
[5,7)  :  811                [6, Inf):   74
(Other): 3609                Missing :   61
FirstRecordedCreditLine_Bins TotalCreditLines_Bins InquiriesLast6Months_Bins
1994   : 1329                [24,28):1895          [0,1)  :4835
1995   : 1280                [2,8)  :1857          [1,2)  :4122
1996   : 1131                [8,12) :1832          [2,3)  :2794
1998   : 1069                [32,37):1716          [3,4)  :1853
1993   : 1031                [15,18):1698          [4,5)  :1353
1997   : 1005                [28,32):1670          [5,6)  : 968
(Other):12142                (Other):8319          (Other):3062
AmountDelinquent_Bins PublicRecordsLast12Months_Bins CurrentCreditLines_Bins
[0,1)     :10648      [0,1)   :14339                  Missing:4105
```

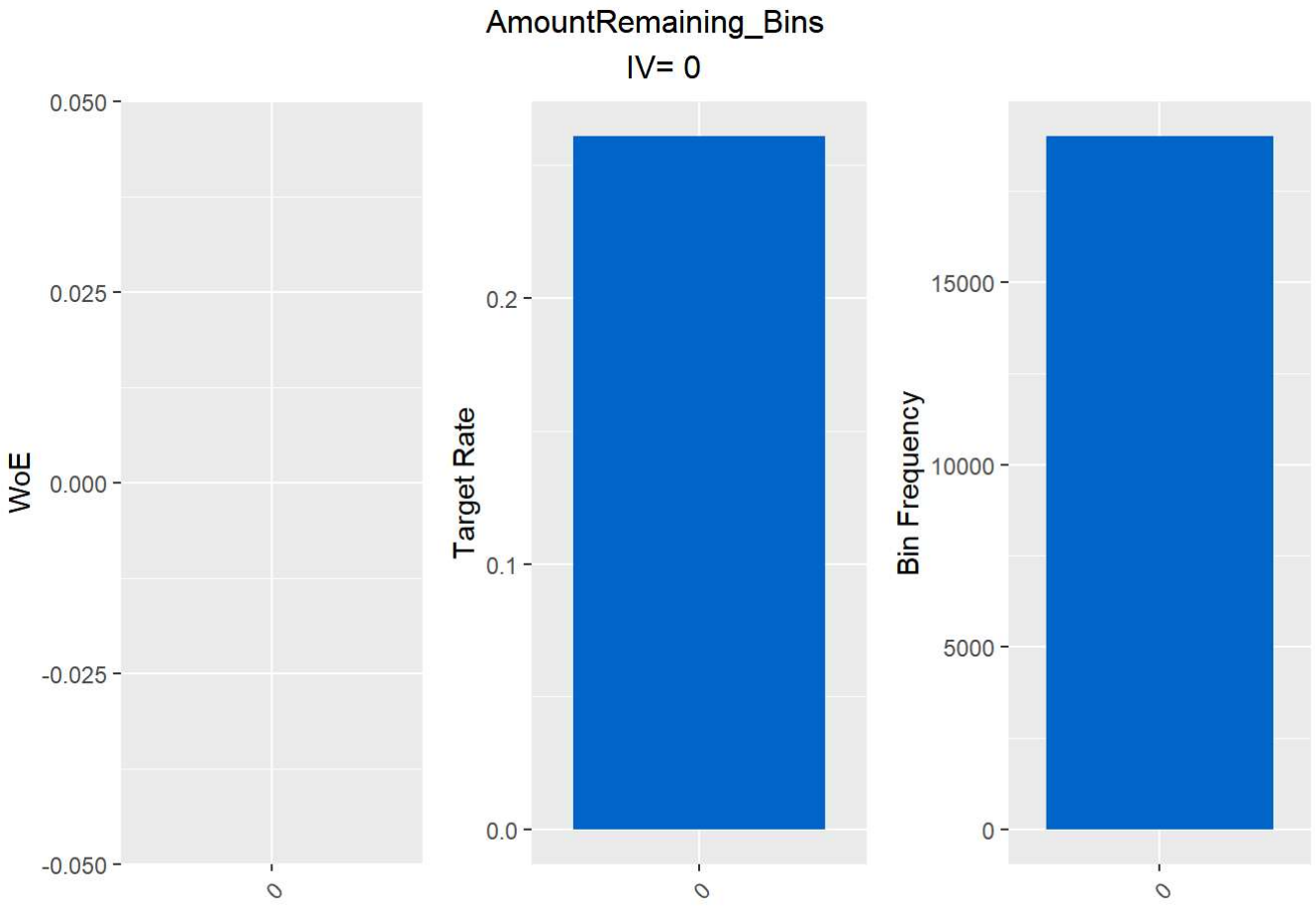
```
 Missing   : 4117      [1,2)   :  482                [9,11) :1992
 [91,207)  :  388      [2,4)   :   54                [3,5)  :1820
 [1,91)    :  386      [4, Inf):    7                [11,13):1585
 [635,1011):  384      Missing : 4105                [15,19):1485
 [207,371) :  383                                    [0,3)  :1314
 (Other)   : 2681                                    (Other):6686
 OpenCreditLines_Bins RevolvingCreditBalance_Bins BankcardUtilization_Bins
 Missing:4105         Missing    :4105            Missing    :4105
 [9,11) :1926         [0,1)      :1407            [0,0.01)   :1985
 [0,3)  :1680         [1,550)    :1230            [0.94,0.99):1454
 [11,13):1382         [5172,7425):1226            [0.8,0.88) :1382
 [7,8)  :1300         [550,1248) :1225            [0.7,0.8)  :1314
 [13,16):1300         [1248,2222):1225            [0.46,0.59):1274
 (Other):7294         (Other)    :8569            (Other)    :7473
   EmploymentStatus_Bins LengthStatusMonths_Bins  Income_Bins
              :  665     [0,1)  :12694           [3,4)  :5662
 Full-time    :12750     [19,27):  630           [0,1)  :4202
 Not available: 3440     [6,12) :  611           [4,5)  :3732
 Not employed :  117     [12,19):  592           [2,3)  :1871
 Part-time    :  586     [1,6)  :  591           [5,6)  :1601
 Retired      :  312     [27,38):  587           [6,7)  :1385
 Self-employed: 1117     (Other): 3282           (Other): 534
             BorrowerOccupation_Bins
 Other                   :10262
 Service Industry        : 2176
 Legal and Administrative: 1567
 Business and Finance    :  980
 Public Service          :  887
 Science and Technology  :  864
 (Other)                 : 2251
```

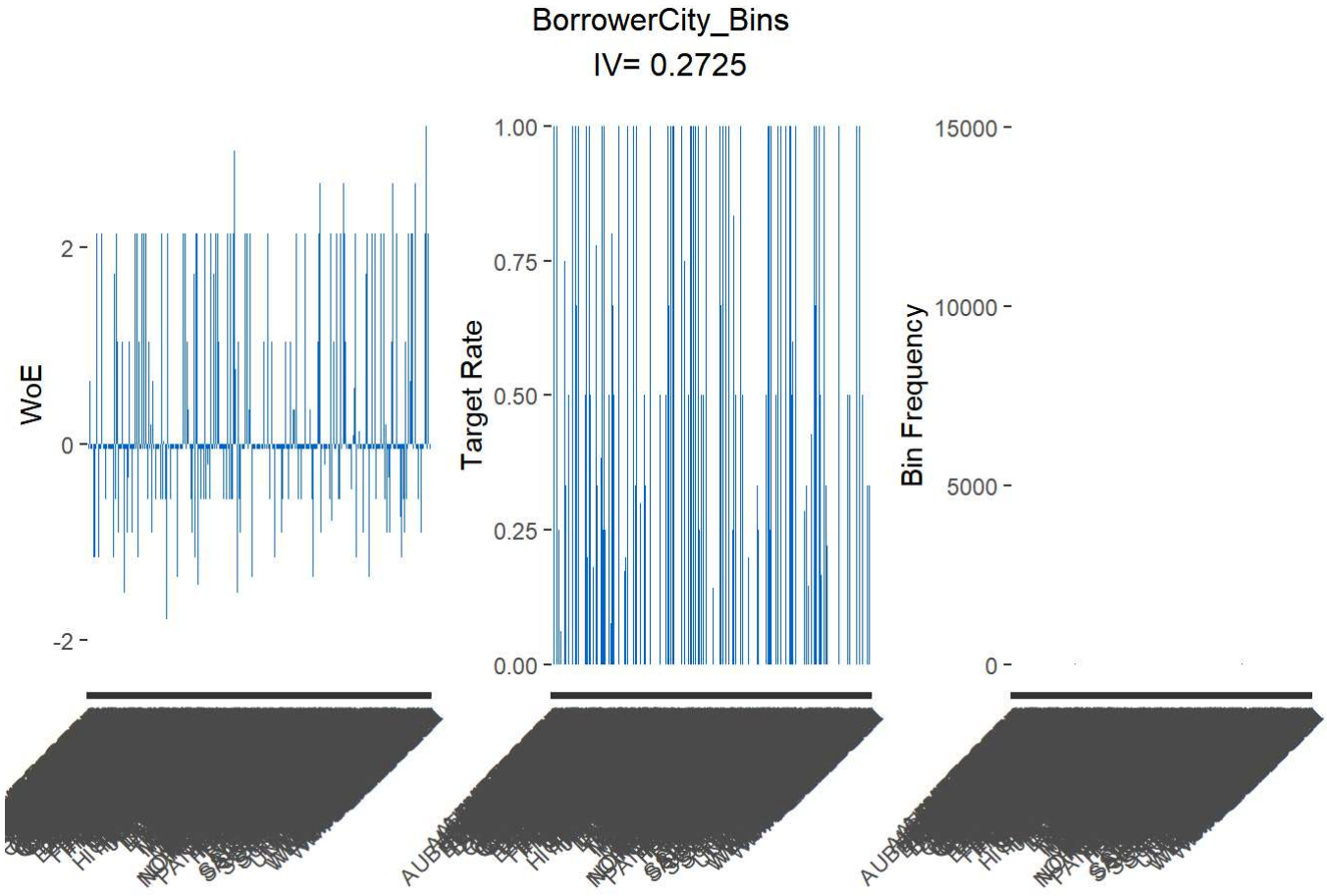
```
str(binData)
```

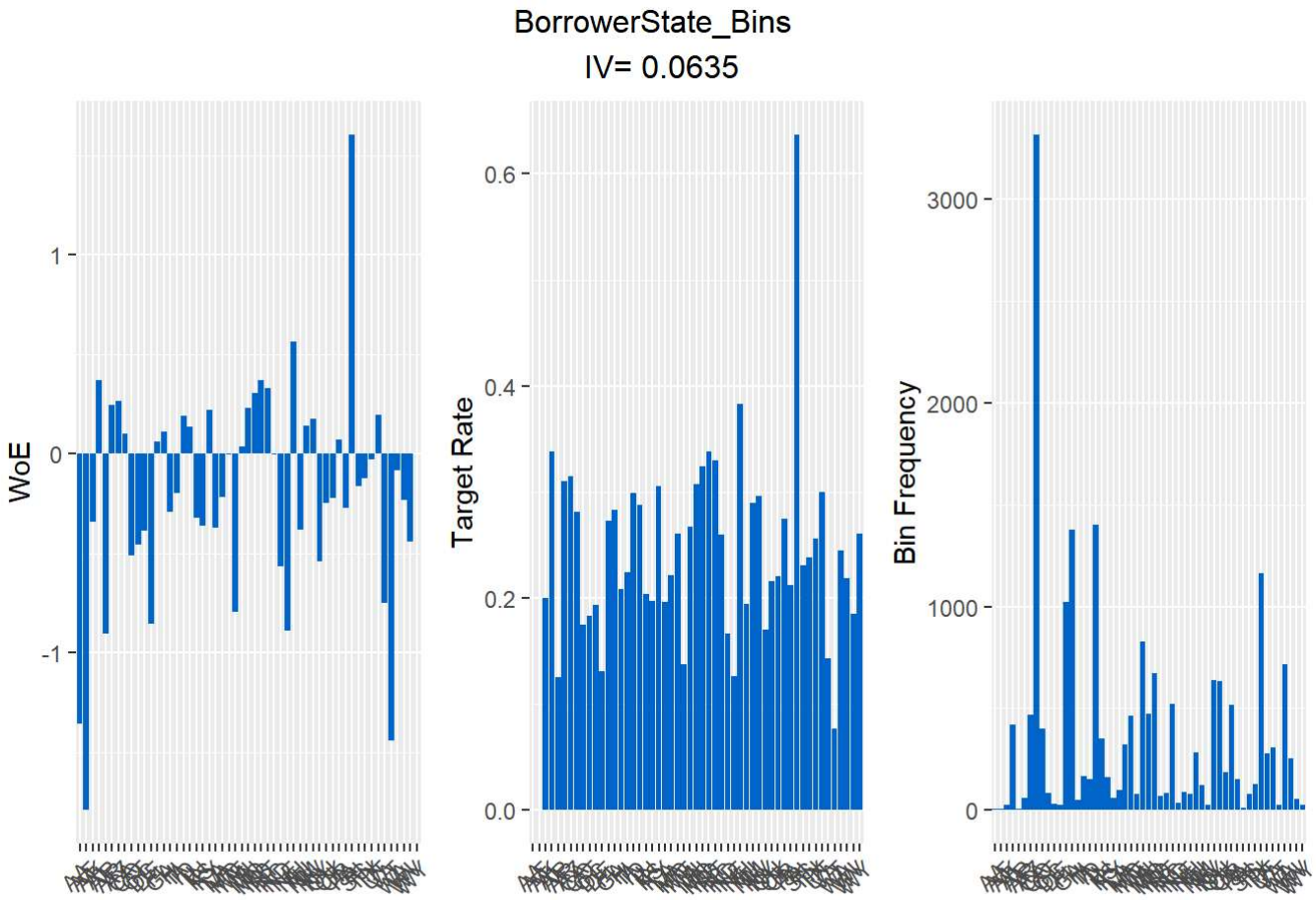
```
'data.frame':   18987 obs. of  28 variables:
 $ ID                           : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Bad                          : int  0 1 0 0 0 0 0 0 0 0 ...
 $ AmountRemaining_Bins         : Factor w/ 1 level "0": 1 1 1 1 1 1 1 1 1 1 ...
 $ BorrowerCity_Bins            : Factor w/ 1536 levels "","ABERDEEN",..: 1 1 1 1 1 1 1 1 1 1
...
 $ BorrowerState_Bins           : Factor w/ 53 levels "AA","AE","AK",..: 15 17 8 24 14 8 18 46
31 18 ...
 $ DebtToIncomeRatio_Bins       : Factor w/ 12 levels "[0,0.07)","[0.07,0.11)",..: 12 11 11 9 8
1 5 2 11 7 ...
 $ IsBorrowerHomeowner_Bins     : Factor w/ 2 levels "0","1": 2 2 1 1 1 1 1 2 1 2 ...
 $ ListingKey_Bins              : Factor w/ 18976 levels "0004337858904319728B4743",..: 14697
7238 2168 1315 16380 15311 11163 13833 13087 9505 ...
 $ ListingNumber_Bins           : Factor w/ 13 levels "[16558,43506)",..: 11 1 5 12 9 11 4 1 8
10 ...
 $ MemberKey_Bins               : Factor w/ 18030 levels "000134239614554 92C1526A",..: 16398
10095 11634 11953 15078 5697 6516 4850 2046 8186 ...
```

```
 $ AmountBorrowed_Bins           : Factor w/ 12 levels "[1000,1201)",..: 2 2 10 3 7 2 7 9 4 11
...
 $ LoanKey_Bins                  : Factor w/ 18970 levels "00013421083473792D70F75",..: 1 2 3
1166 4 5 6 7 8 9 ...
 $ CurrentDelinquencies_Bins     : Factor w/ 10 levels "[0,1)","[1,2)",..: 1 1 1 1 1 2 5 8 1 1
...
 $ DelinquenciesLast7Years_Bins  : Factor w/ 12 levels "[0,1)","[1,2)",..: 1 1 1 8 1 1 8 9 1 1
...
 $ PublicRecordsLast10Years_Bins : Factor w/ 7 levels "[0,1)","[1,2)",..: 1 2 1 2 1 2 1 1 1 1 ...
 $ FirstRecordedCreditLine_Bins  : Factor w/ 56 levels "","1947","1950",..: 46 45 36 38 54 38 42
45 53 42 ...
 $ TotalCreditLines_Bins         : Factor w/ 13 levels "[2,8)","[8,12)",..: 8 9 8 11 1 3 9 6 2 8
...
 $ InquiriesLast6Months_Bins     : Factor w/ 12 levels "[0,1)","[1,2)",..: 4 3 2 2 3 2 5 6 2 1
...
 $ AmountDelinquent_Bins         : Factor w/ 13 levels "[0,1)","[1,91)",..: 1 13 1 1 1 5 10 13 1
1 ...
 $ PublicRecordsLast12Months_Bins: Factor w/ 5 levels "[0,1)","[1,2)",..: 1 5 1 1 1 1 1 5 1 1 ...
 $ CurrentCreditLines_Bins       : Factor w/ 12 levels "[0,3)","[3,5)",..: 7 12 11 11 2 1 7 12 7
10 ...
 $ OpenCreditLines_Bins          : Factor w/ 13 levels "[0,3)","[3,4)",..: 6 13 11 11 2 1 8 13 8
10 ...
 $ RevolvingCreditBalance_Bins   : Factor w/ 13 levels "[0,1)","[1,550)",..: 5 13 11 9 3 1 5 13 7
11 ...
 $ BankcardUtilization_Bins      : Factor w/ 13 levels "[0,0.01)","[0.01,0.16)",..: 3 13 8 9 4 1
4 13 6 4 ...
 $ EmploymentStatus_Bins         : Factor w/ 7 levels "","Full-time",..: 2 3 2 2 5 2 2 3 5 2 ...
 $ LengthStatusMonths_Bins       : Factor w/ 12 levels "[0,1)","[1,6)",..: 1 1 7 1 3 1 7 1 1 1
...
 $ Income_Bins                   : Factor w/ 8 levels "[0,1)","[1,2)",..: 2 1 3 6 3 5 4 1 3 5 ...
 $ BorrowerOccupation_Bins       : Factor w/ 10 levels "Business and Finance",..: 7 6 2 6 6 6 6 6
6 3 ...
```

```r
#Plot binned data
for (i in 3:27) {
  WOEplotter(binData, "Bad", names(binData)[i])
}
```

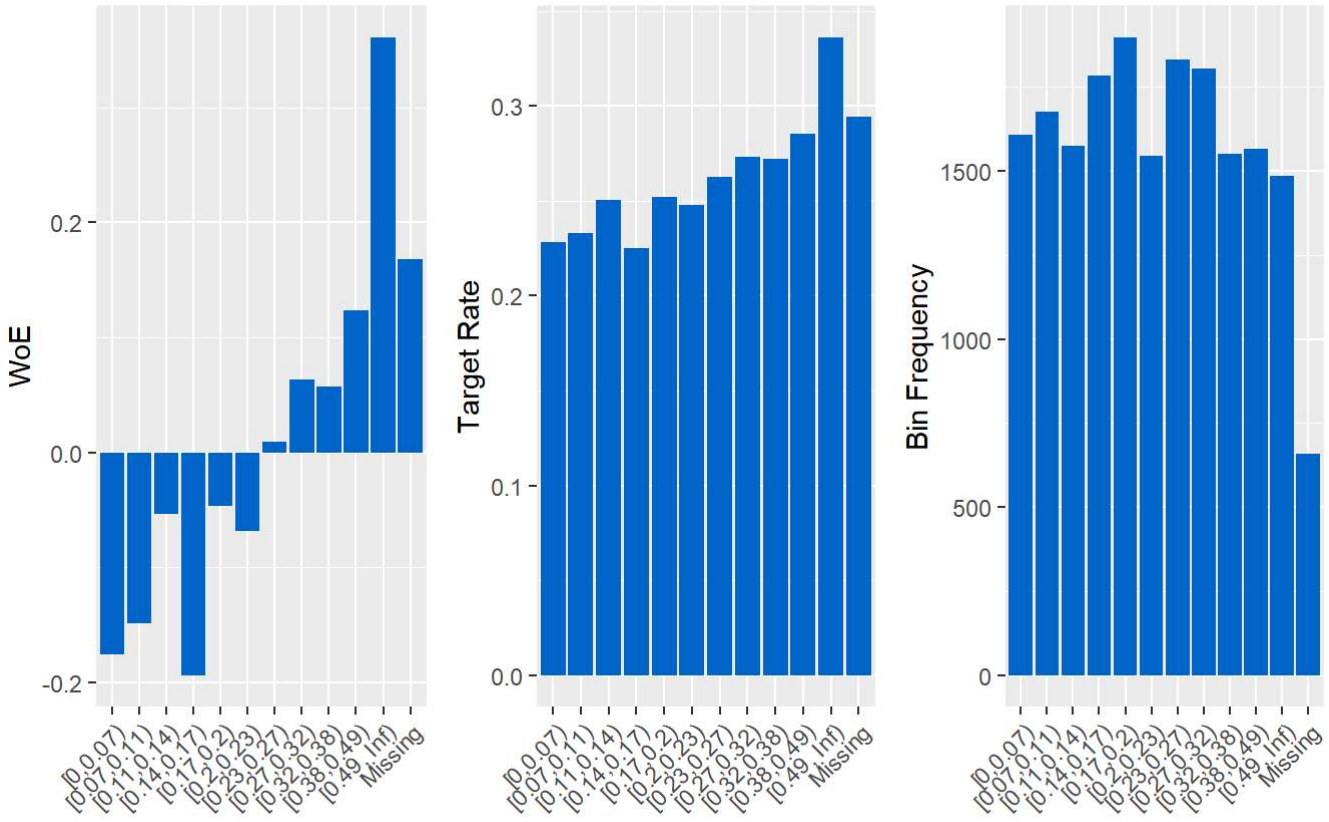

AmountRemaining_Bins
IV= 0
WoE
0.050
0.025
0.000
-0.025
-0.050
0
Target Rate
0.2
0.1
0.0
0
Bin Frequency
15000
10000
5000
0
0

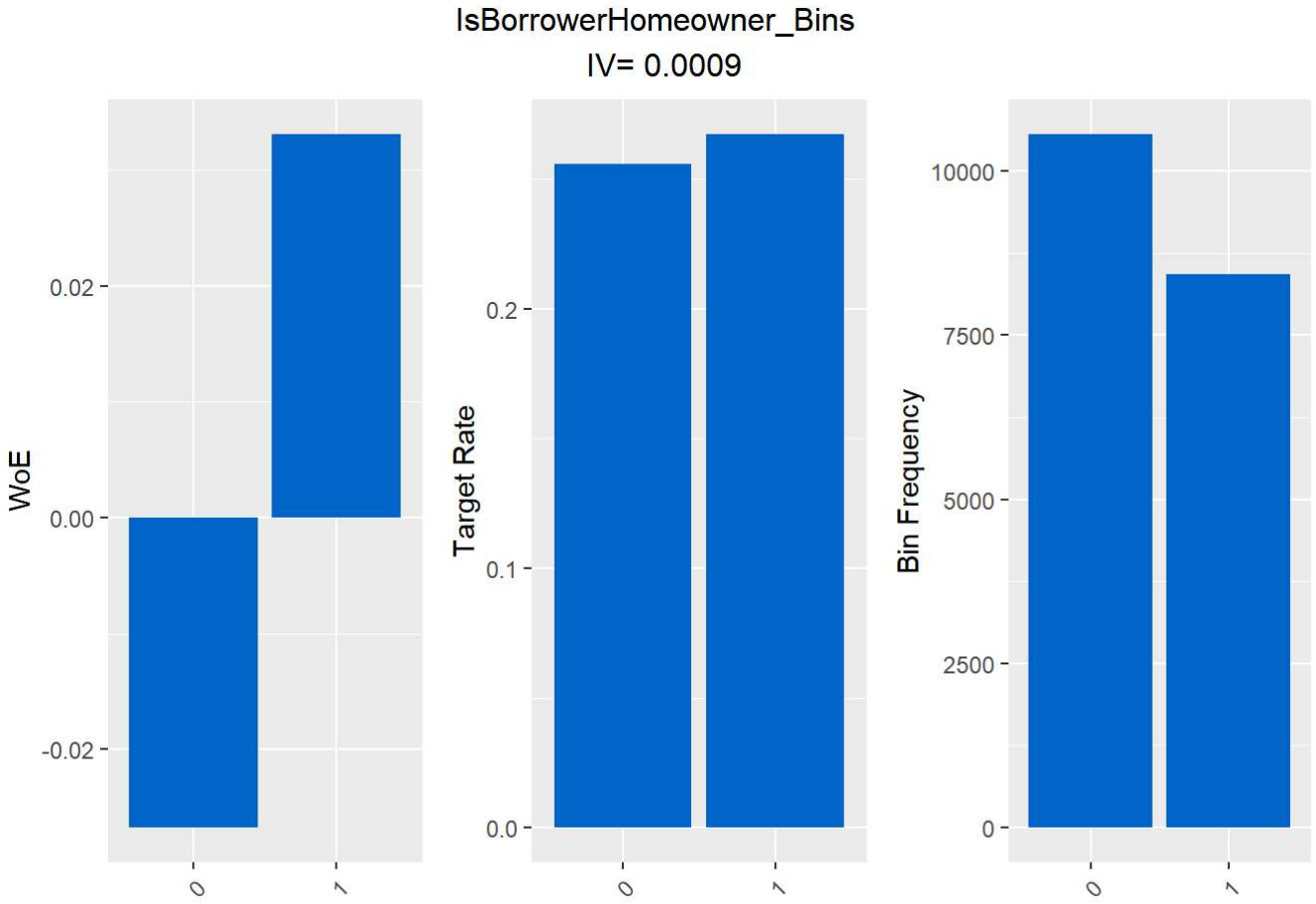

BorrowerCity_Bins
IV= 0.2725
WoE
2
0
-2
Target Rate
1.00
0.75
0.50
0.25
0.00
Bin Frequency
15000
10000
5000
0

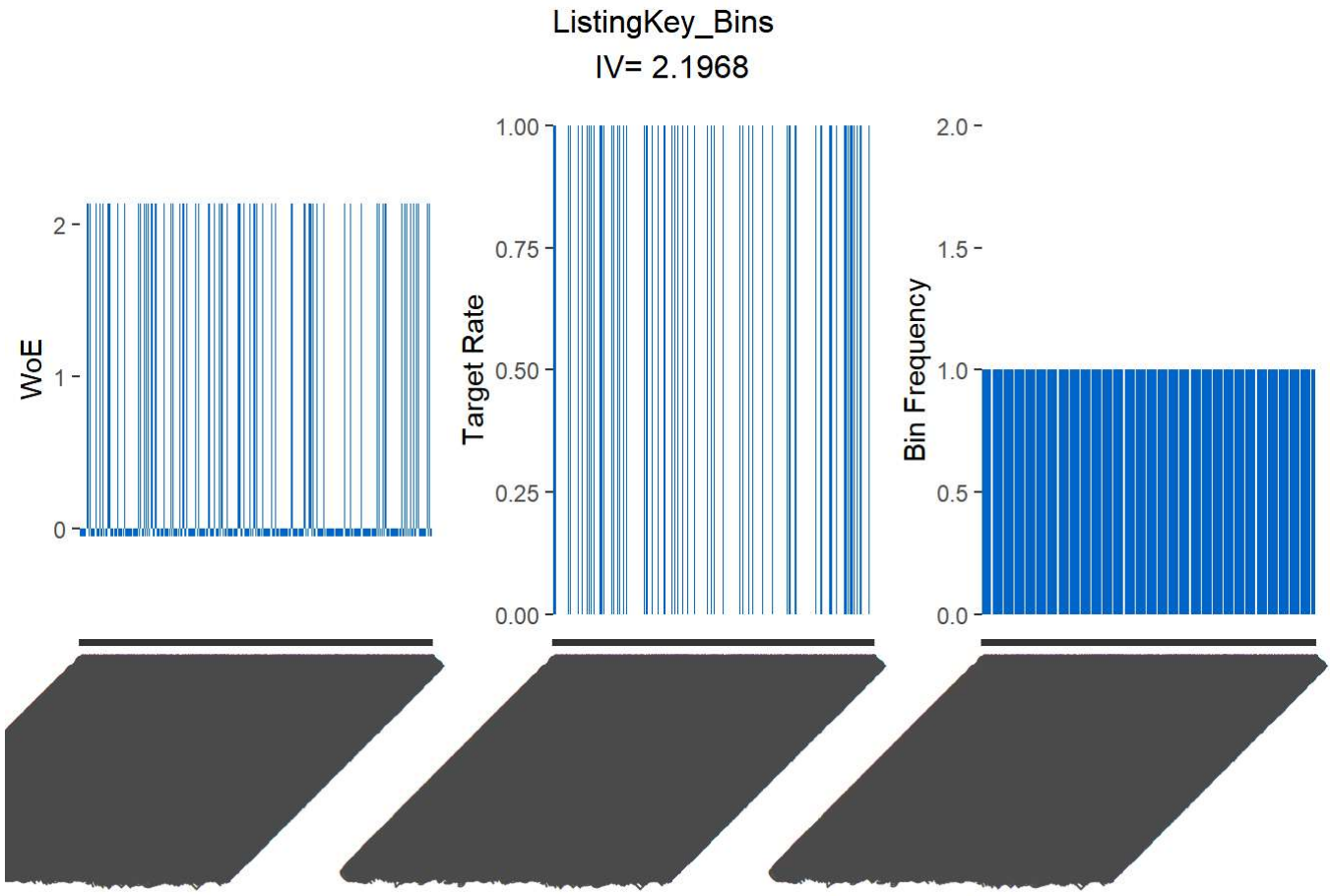

BorrowerState_Bins
IV= 0.0635
WoE
1
0
-1
Target Rate
0.6
0.4
0.2
0.0
Bin Frequency
3000
2000
1000
0

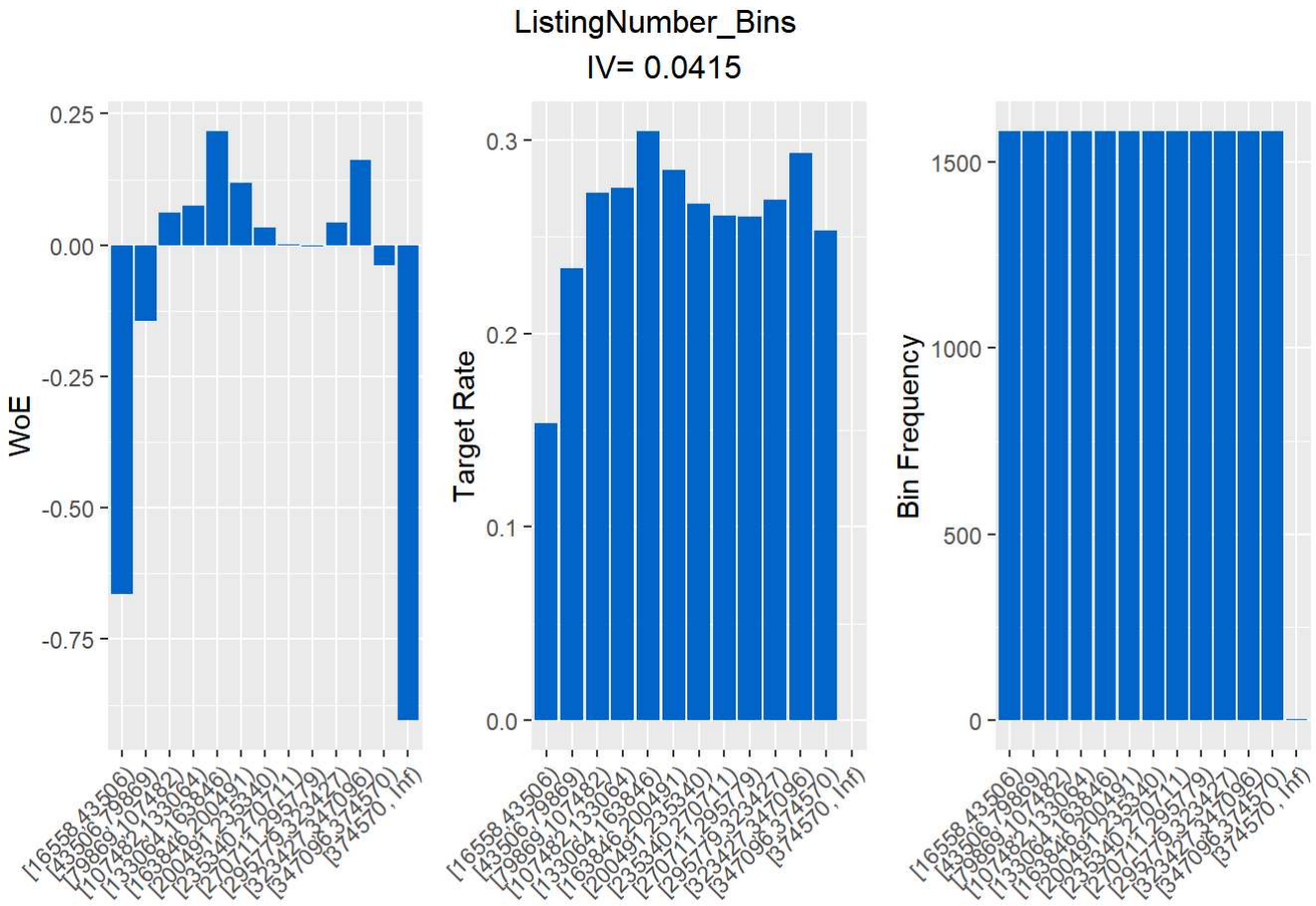
DebtToIncomeRatio_Bins

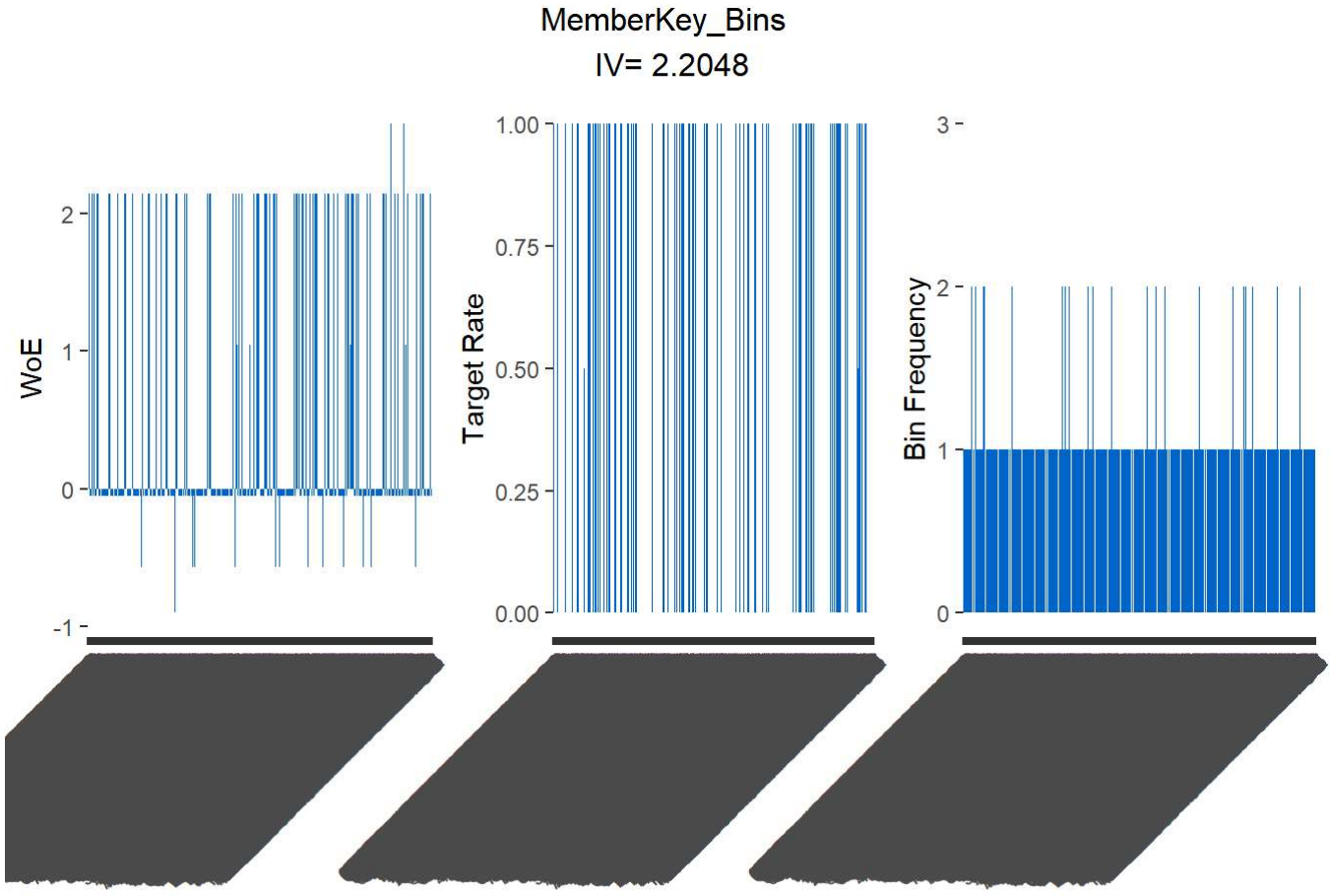
IV= 0.0225

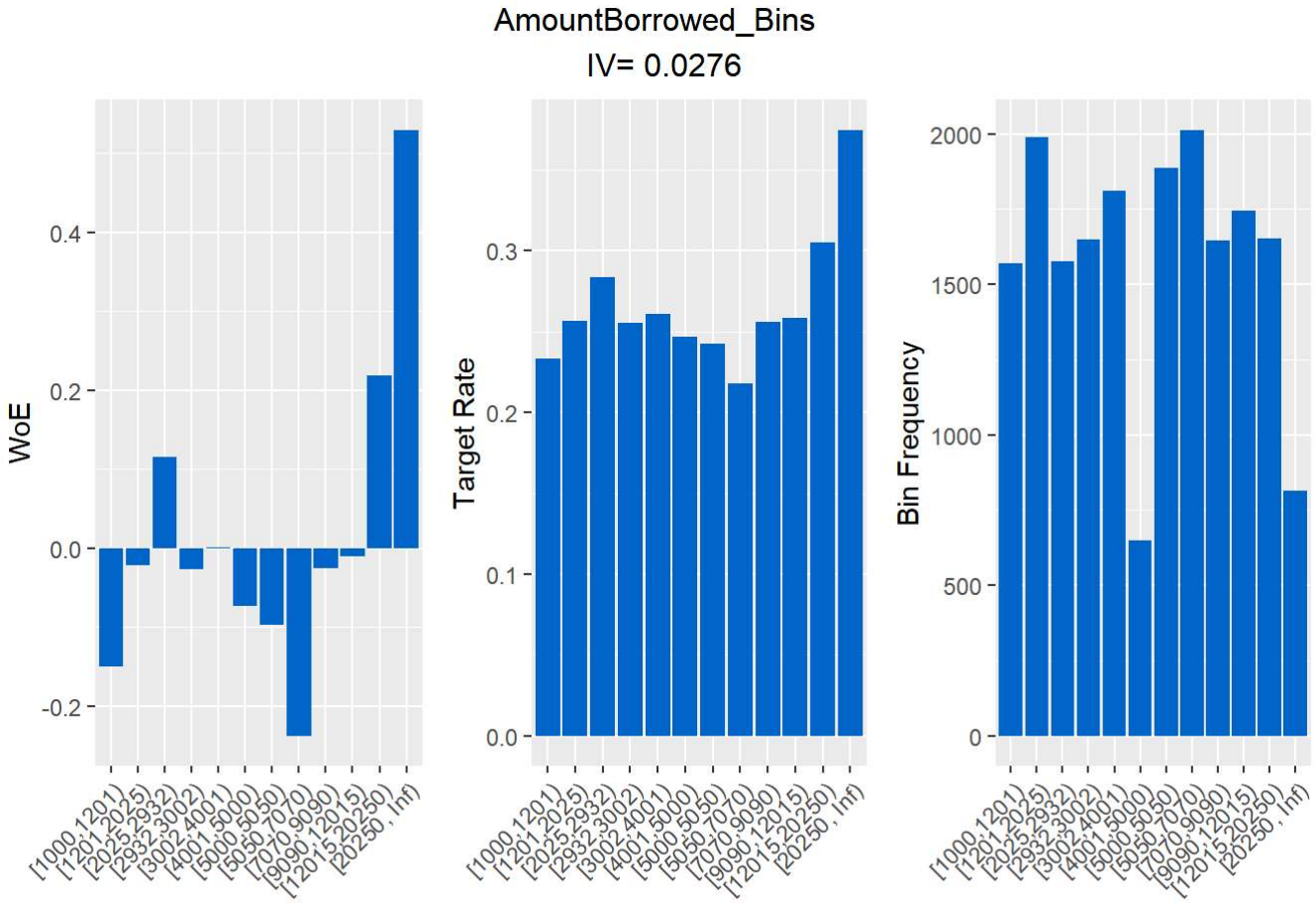


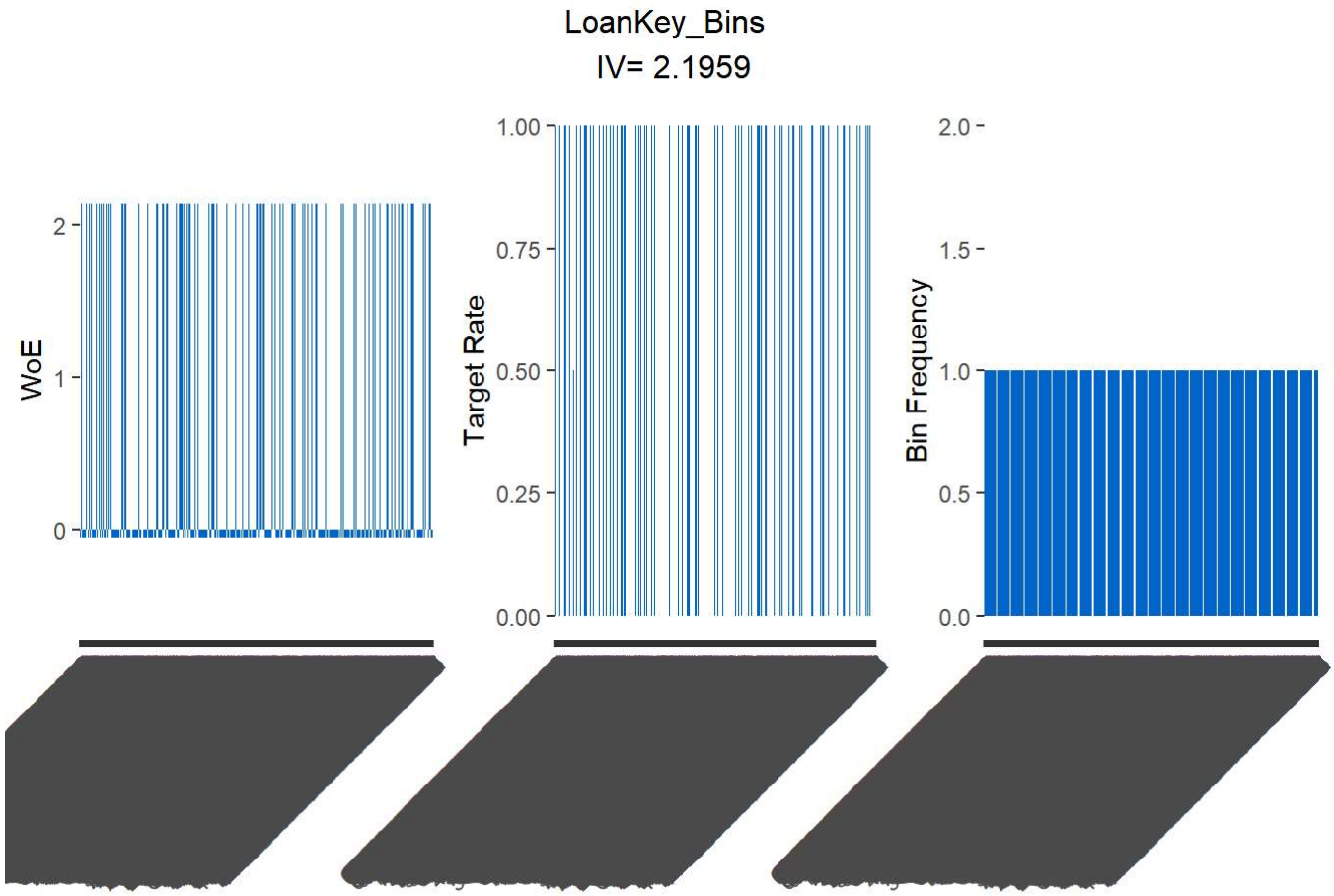

IsBorrowerHomeowner_Bins
IV= 0.0009
WoE
0.02
0.00
-0.02
0
1
Target Rate
0.2
0.1
0.0
0
1
Bin Frequency
10000
7500
5000
2500
0
0
1

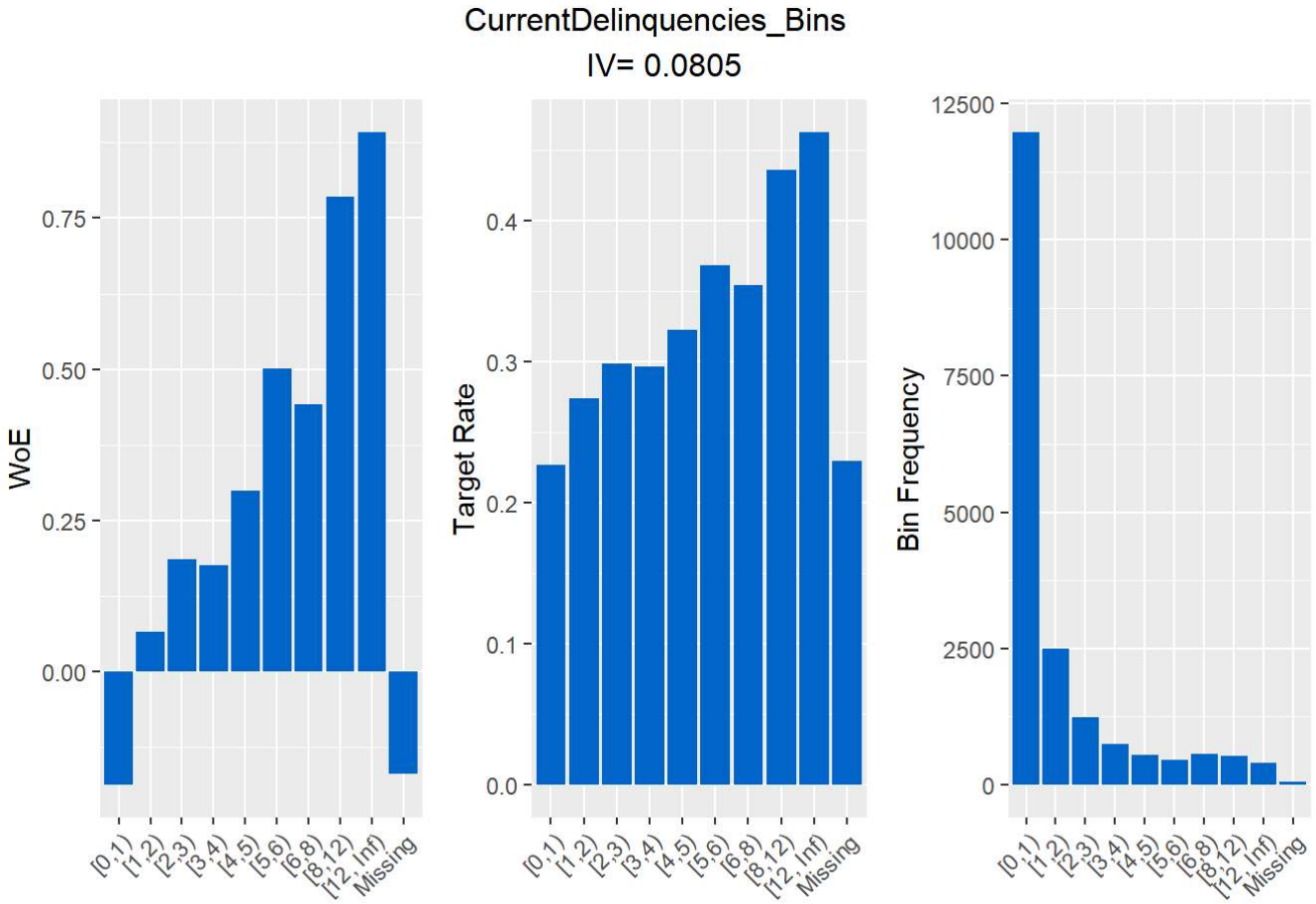

ListingKey_Bins
IV= 2.1968
WoE
2
1
0
Target Rate
1.00
0.75
0.50
0.25
0.00
Bin Frequency
2.0
1.5
1.0
0.5
0.0


ListingNumber_Bins
IV= 0.0415
WoE
Target Rate
Bin Frequency
[16558,43506)
[43506,79869)
[79869,107482)
[107482,133064)
[133064,163846)
[163846,200491)
[200491,235340)
[235340,270711)
[270711,295779)
[295779,323427)
[323427,347096)
[347096,374570)
[374570, Inf)


MemberKey_Bins
IV= 2.2048
WoE
2
1
0
-1
Target Rate
1.00
0.75
0.50
0.25
0.00
Bin Frequency
3
2
1
0

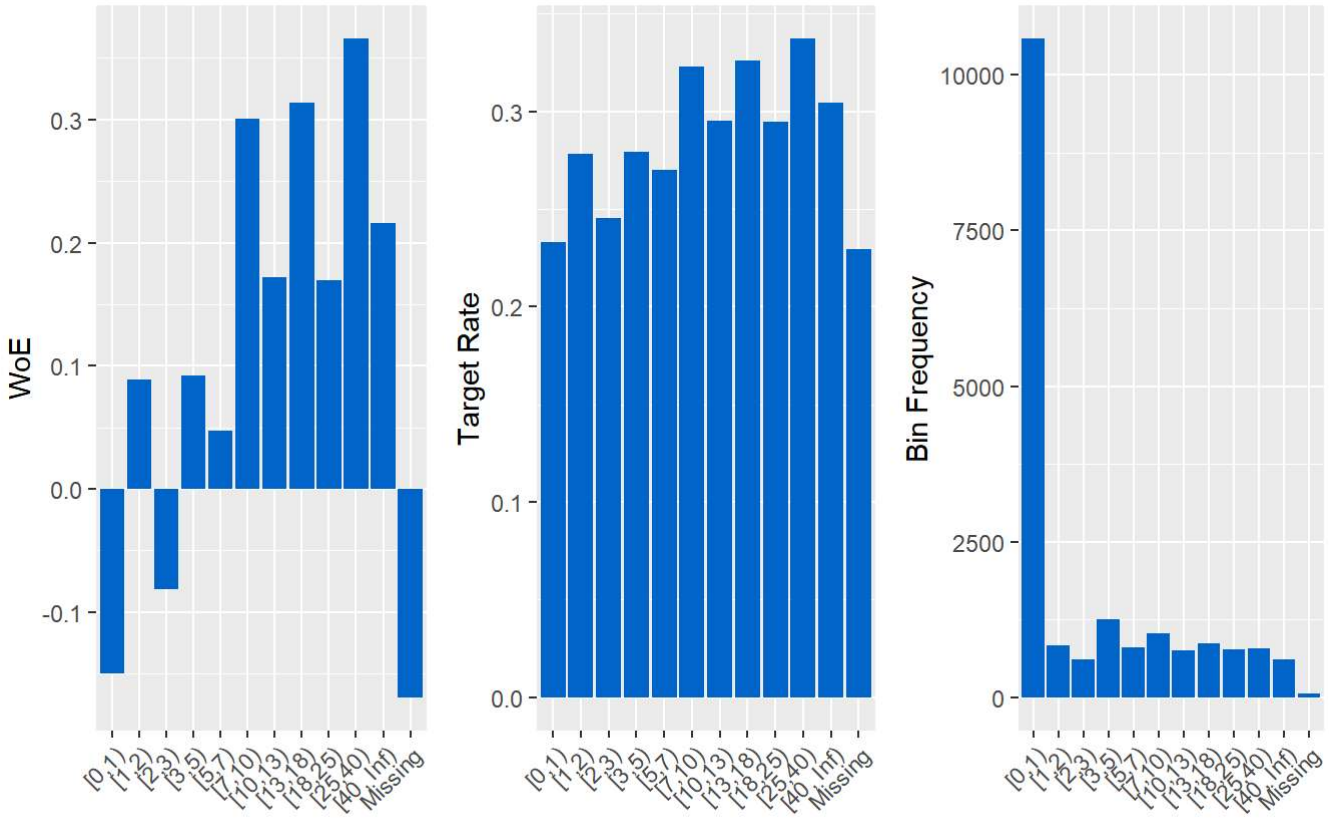

AmountBorrowed_Bins
IV= 0.0276
WoE
Target Rate
Bin Frequency
[1000,1201)
[1201,2025)
[2025,2932)
[2932,3002)
[3002,4001)
[4001,5000)
[5000,5050)
[5050,7070)
[7070,9090)
[9090,12015)
[12015,20250)
[20250, Inf)


LoanKey_Bins
IV= 2.1959
WoE
2
1
0
Target Rate
1.00
0.75
0.50
0.25
0.00
Bin Frequency
2.0
1.5
1.0
0.5
0.0


CurrentDelinquencies_Bins
IV= 0.0805
WoE
0.75
0.50
0.25
0.00
Target Rate
0.4
0.3
0.2
0.1
0.0
Bin Frequency
12500
10000
7500
5000
2500
0
[0,1)
[1,2)
[2,3)
[3,4)
[4,5)
[5,6)
[6,8)
[8,12)
[12, Inf)
Missing

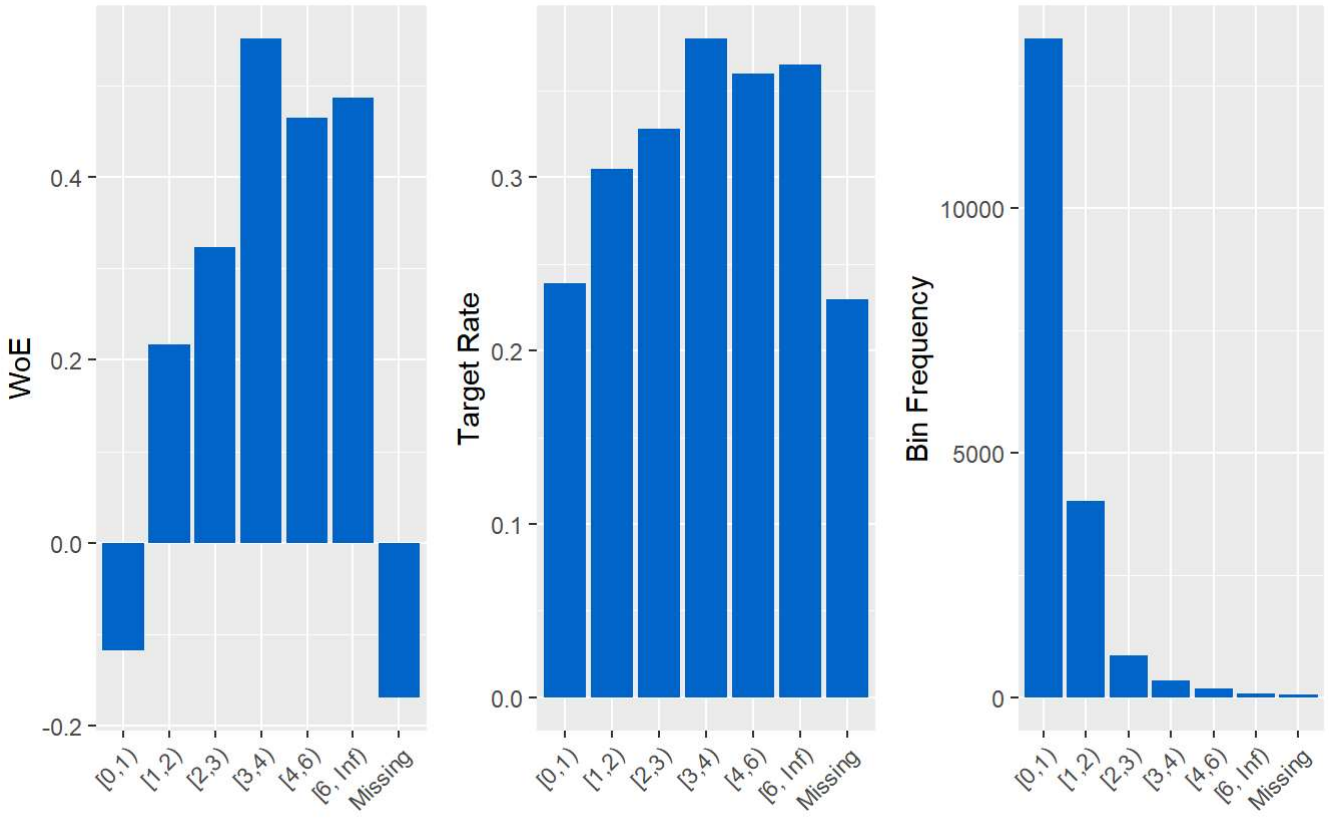
DelinquenciesLast7Years_Bins

IV= 0.0335



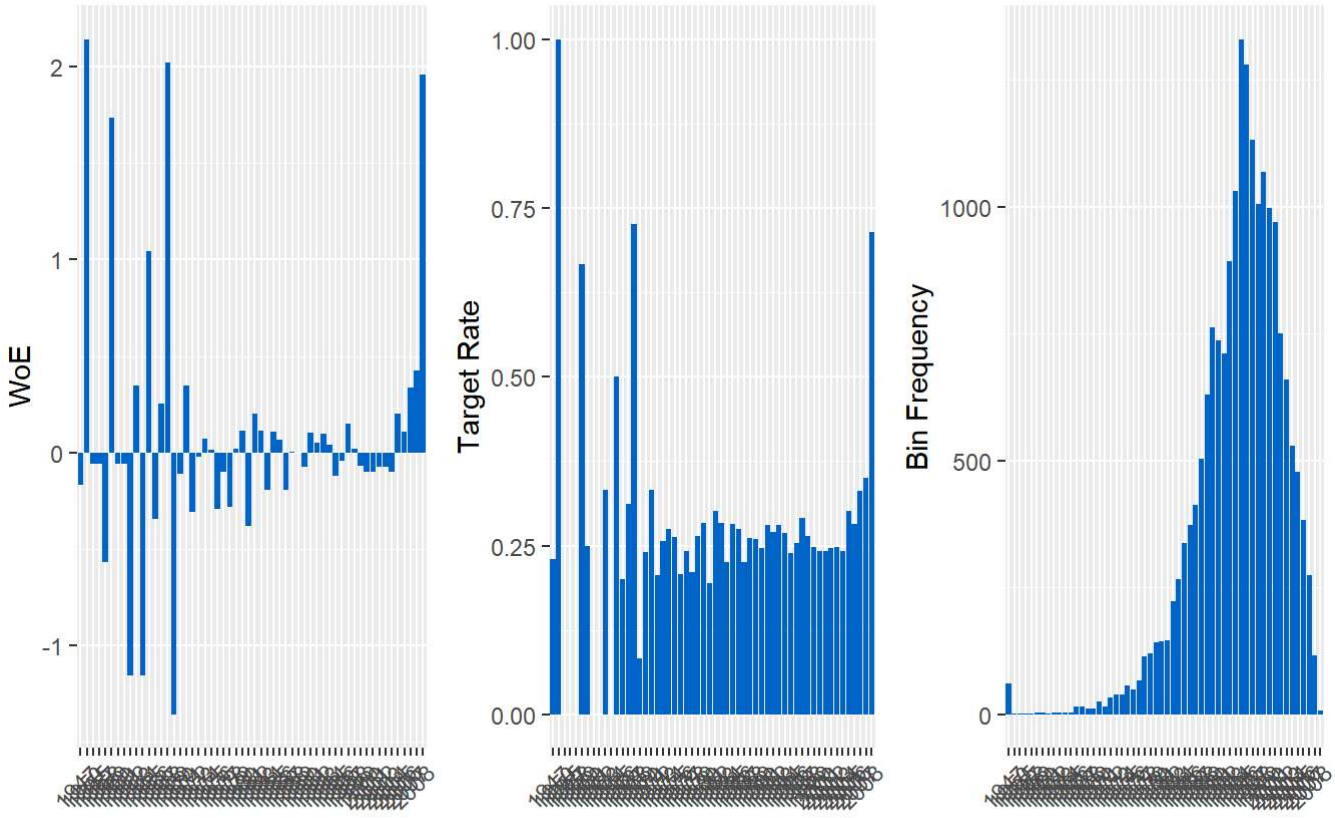
PublicRecordsLast10Years_Bins

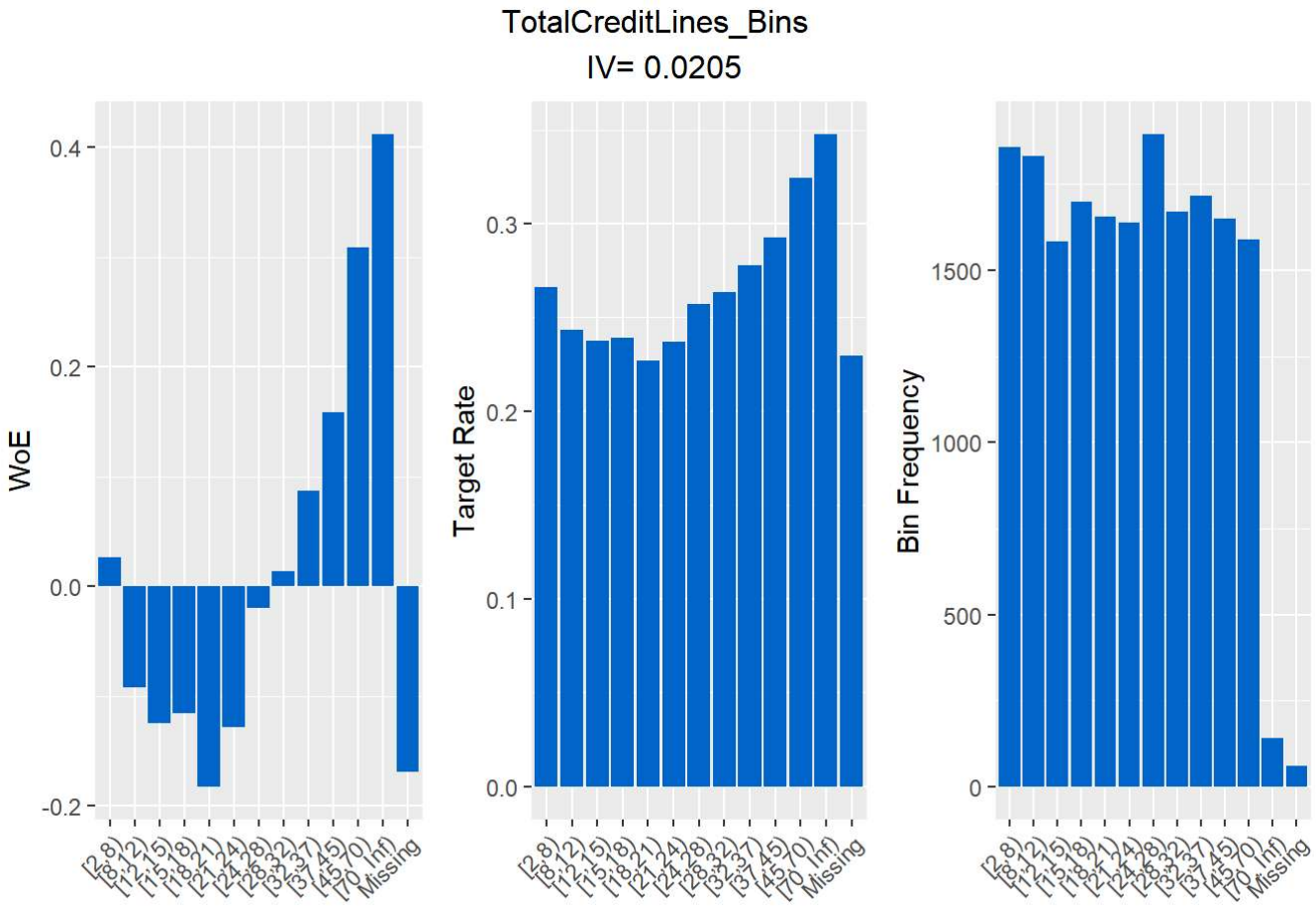
IV= 0.0346



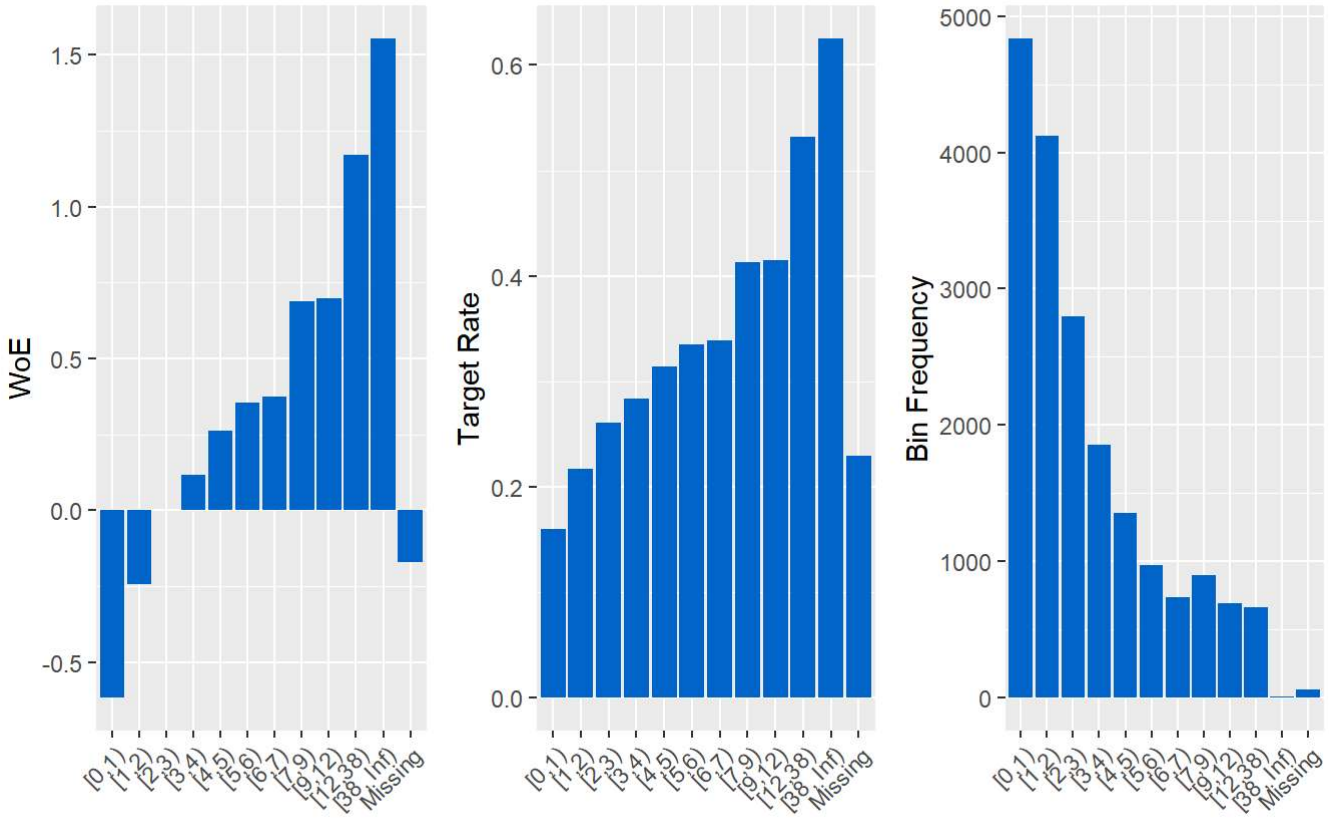
FirstRecordedCreditLine_Bins

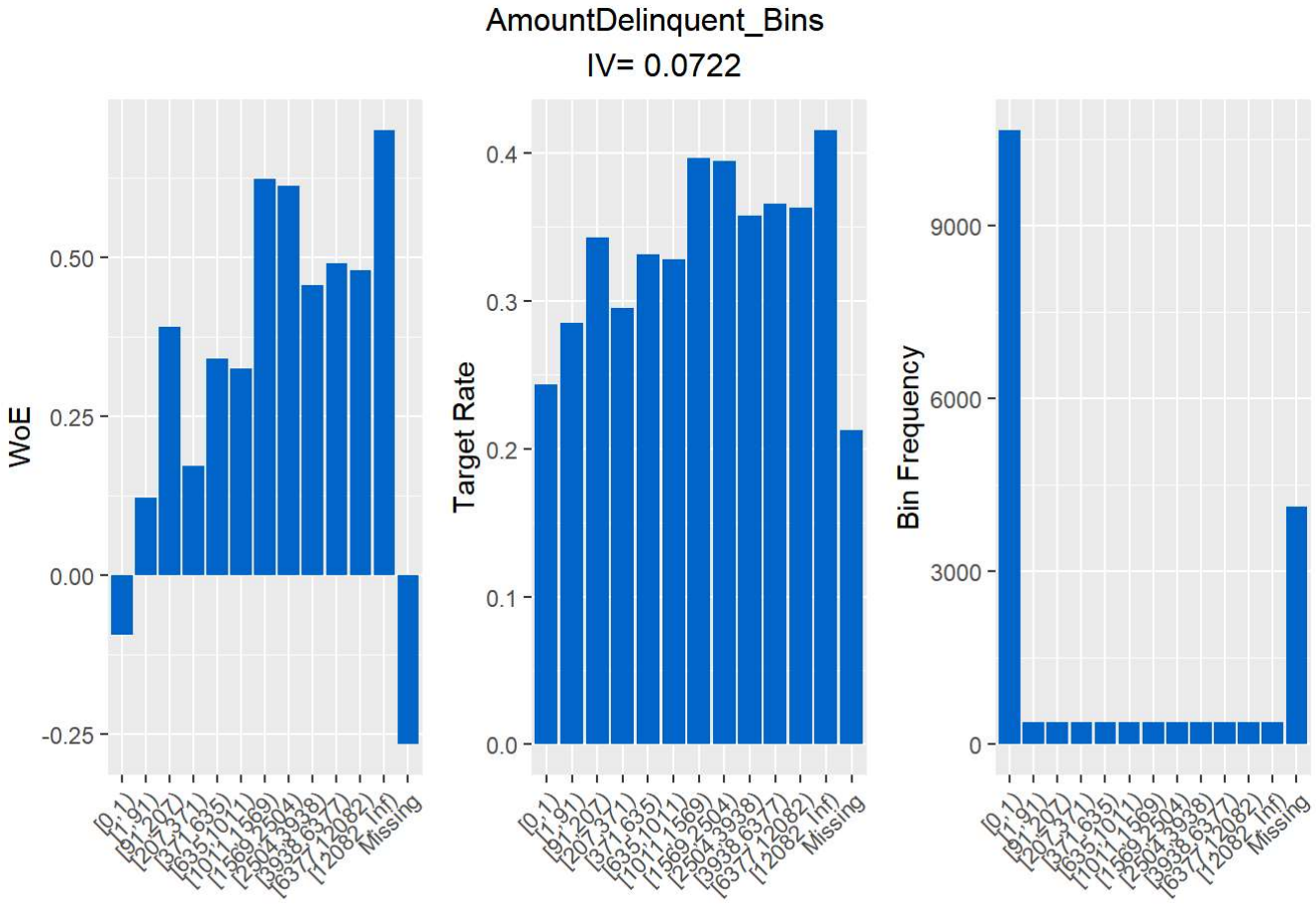
IV= 0.0217

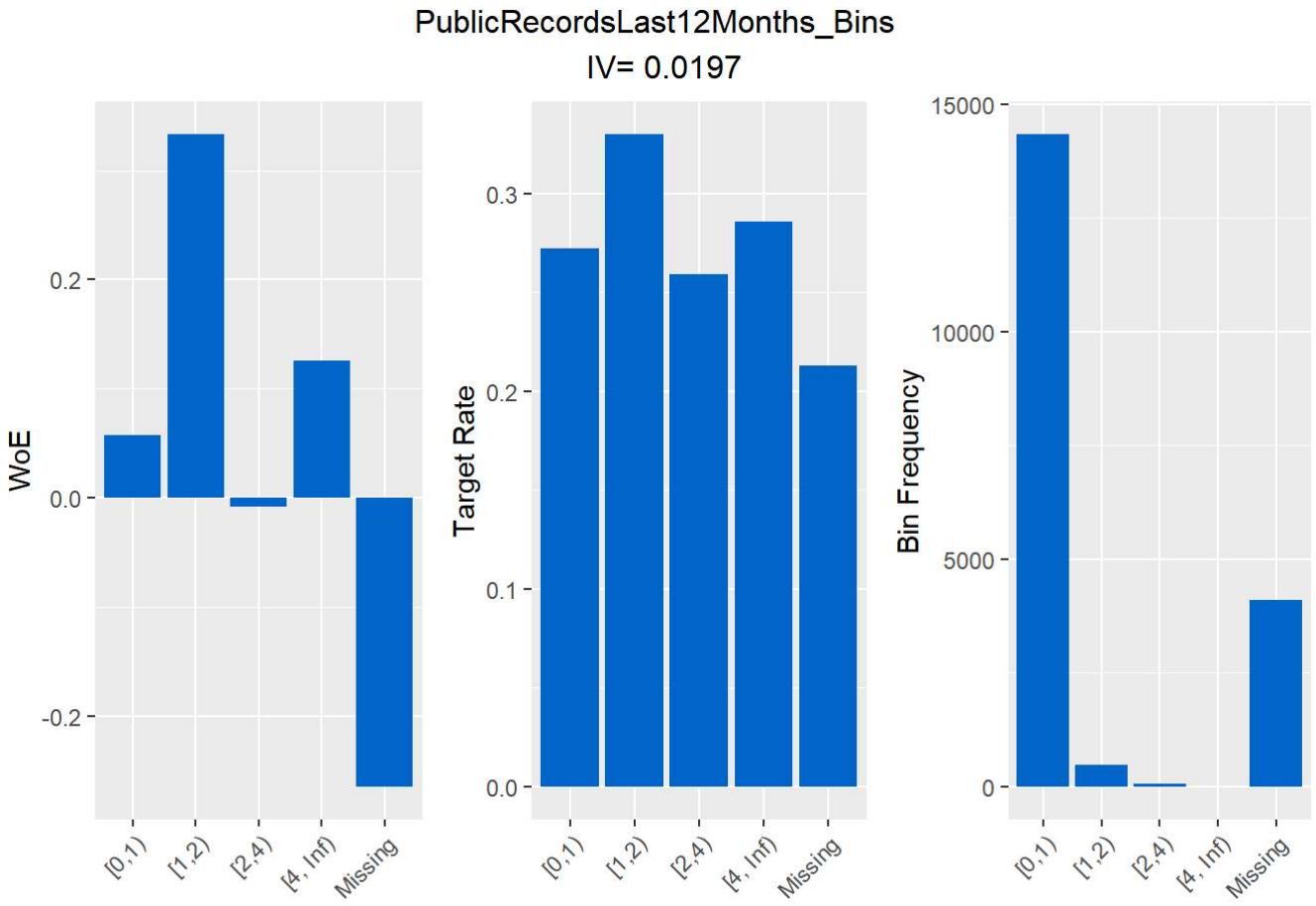


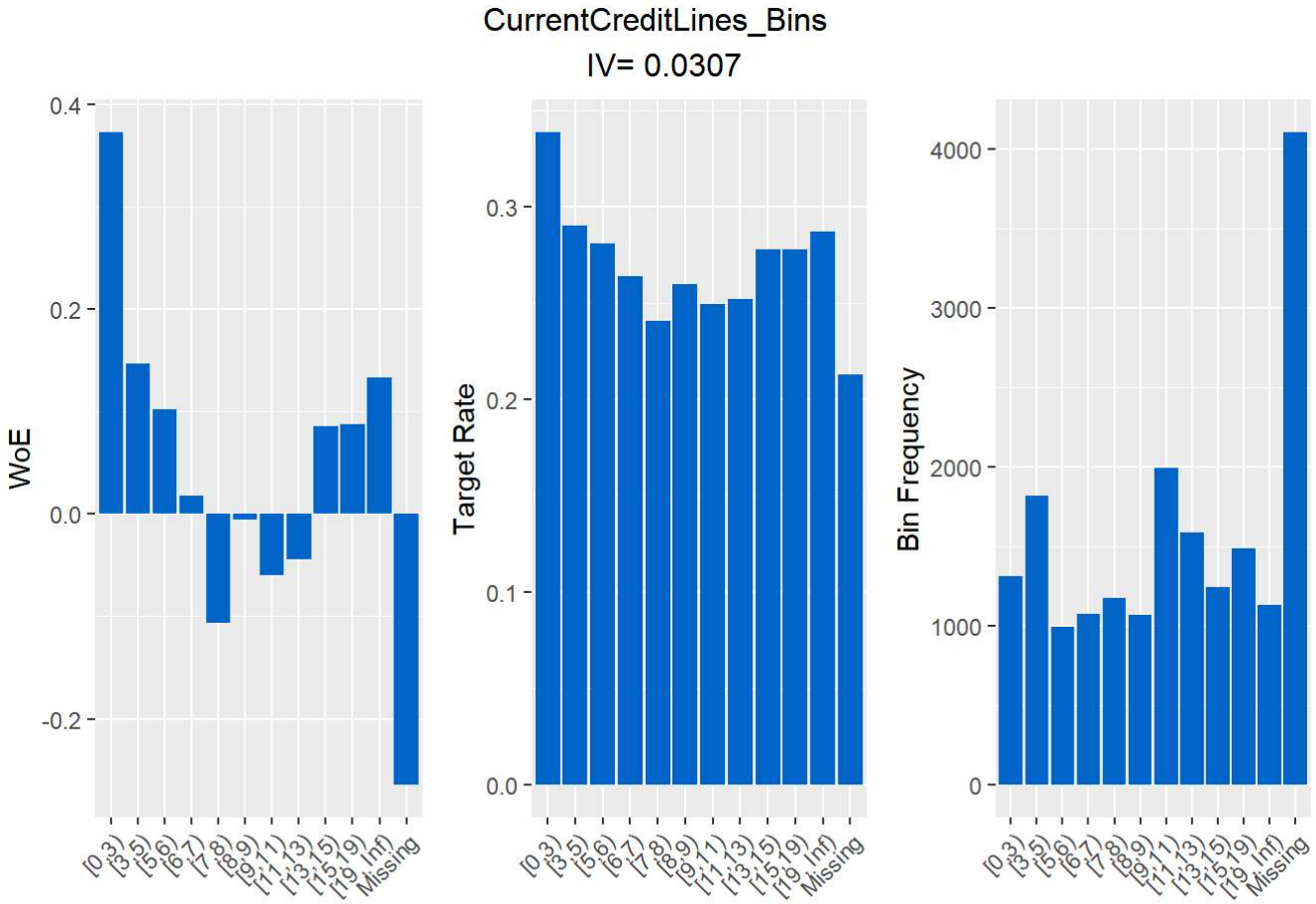

TotalCreditLines_Bins
IV= 0.0205
WoE
Target Rate
Bin Frequency
[2,8)
[8,12)
[12,15)
[15,18)
[18,21)
[21,24)
[24,28)
[28,32)
[32,37)
[37,45)
[45,70)
[70,Inf)
Missing

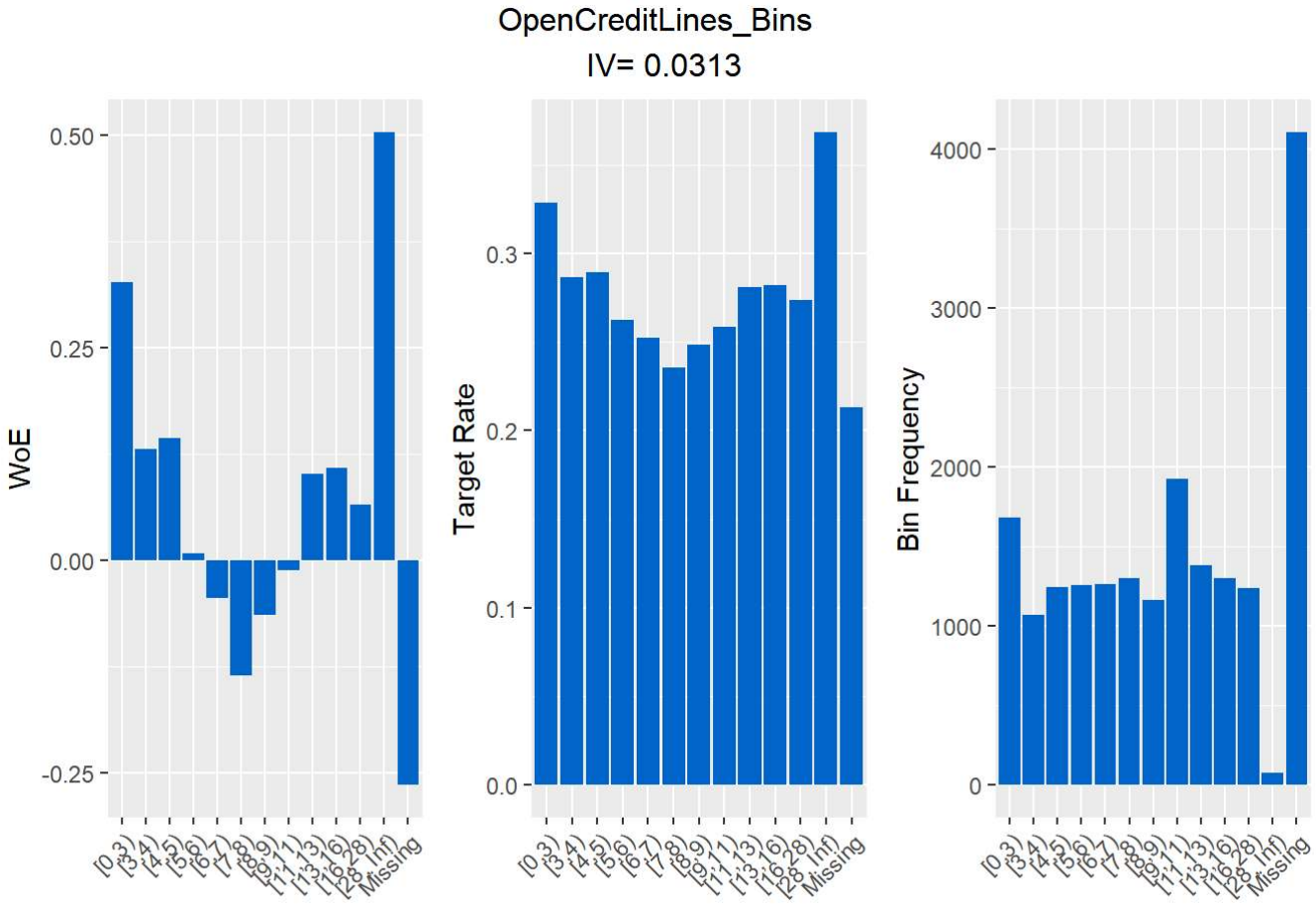
InquiriesLast6Months_Bins
IV= 0.2186

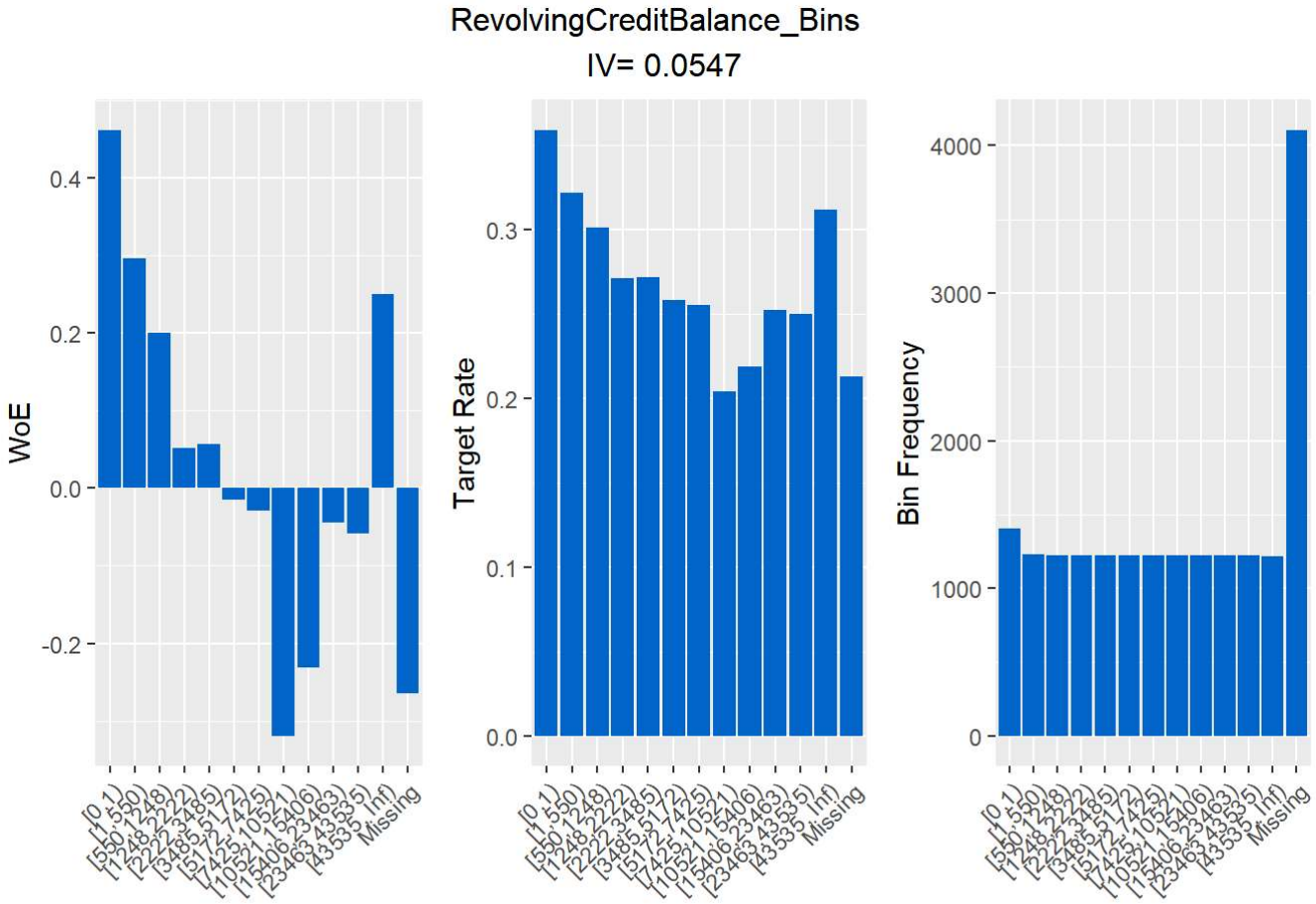


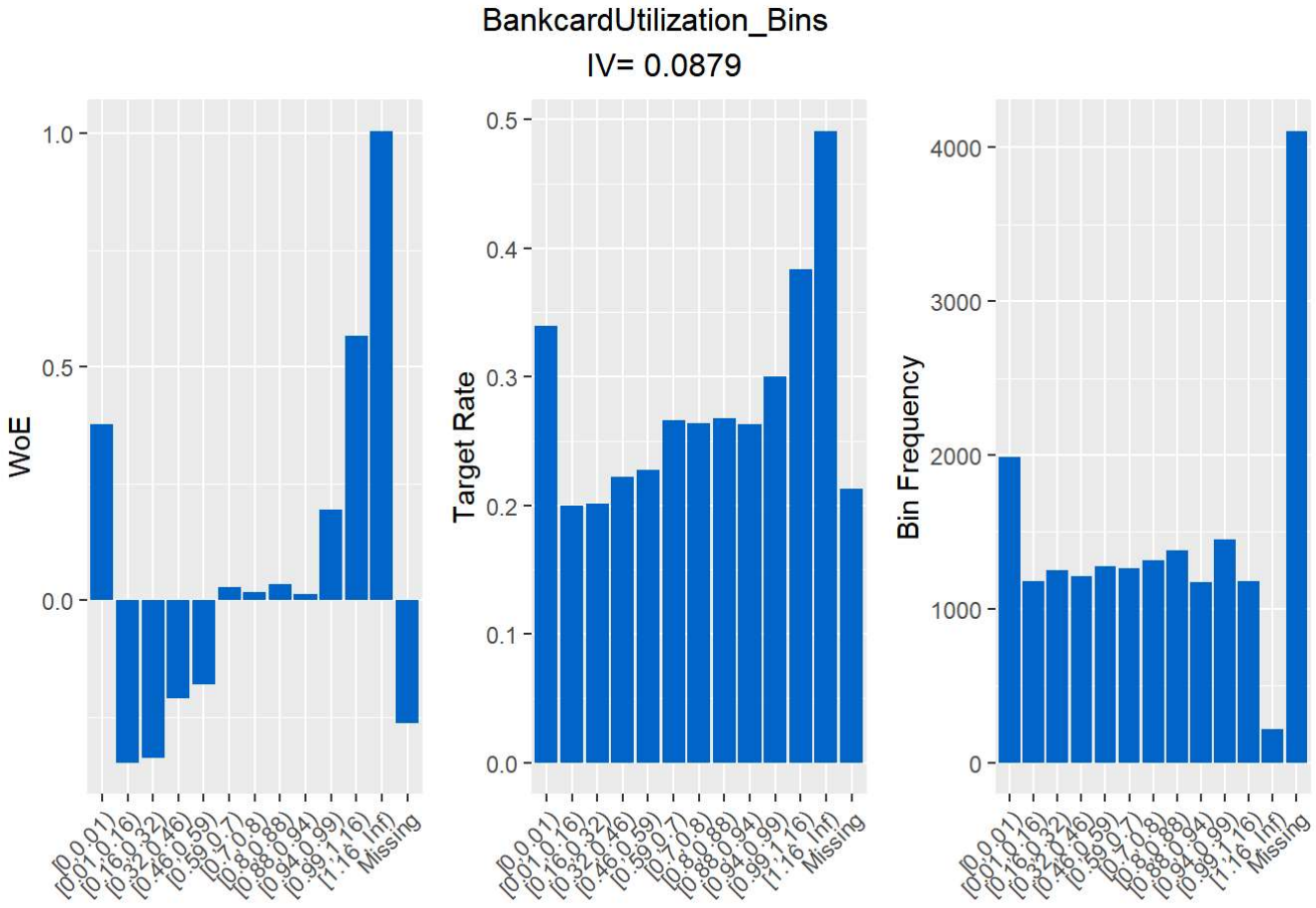

AmountDelinquent_Bins
IV= 0.0722
WoE
0.50
0.25
0.00
-0.25
Target Rate
0.4
0.3
0.2
0.1
0.0
Bin Frequency
9000
6000
3000
0
[0;1)
[1;91)
[91;207)
[207;371)
[371;635)
[635;1011)
[1011;1569)
[1569;2504)
[2504;3938)
[3938;6377)
[6377;12082)
[12082;Inf)
Missing

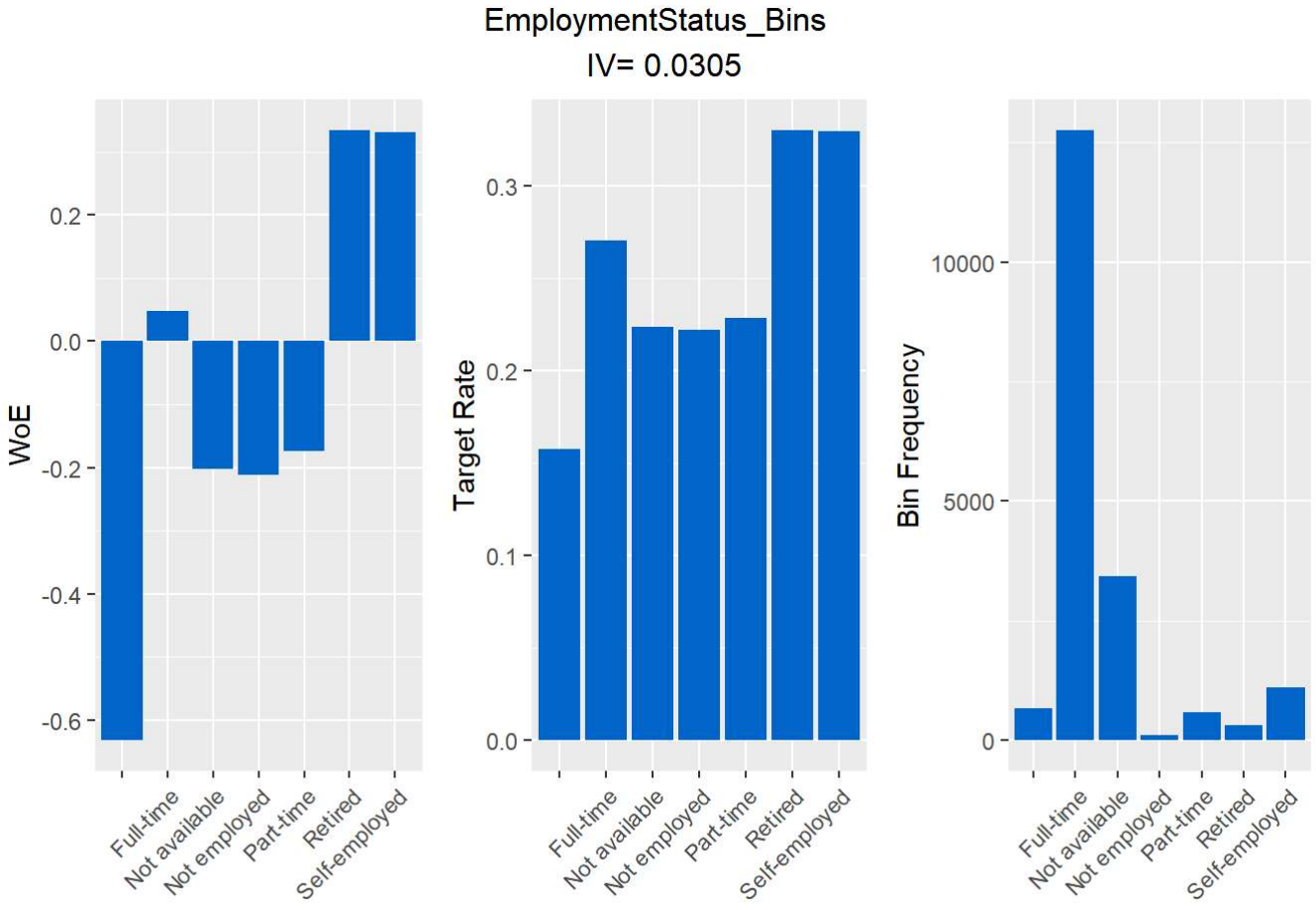

PublicRecordsLast12Months_Bins
IV= 0.0197
WoE
0.2
0.0
-0.2
Target Rate
0.3
0.2
0.1
0.0
Bin Frequency
15000
10000
5000
0
[0,1)
[1,2)
[2,4)
[4, Inf)
Missing

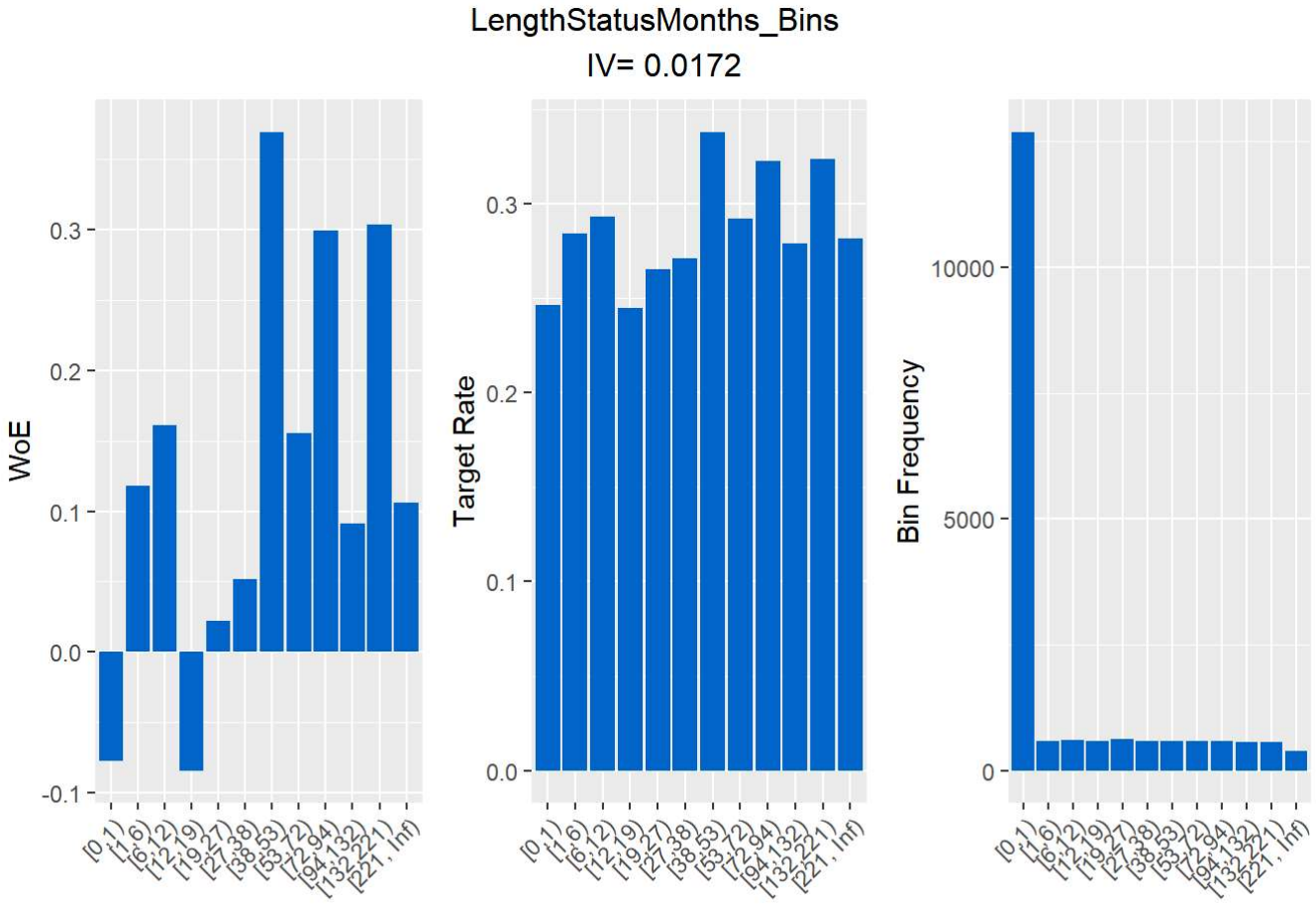

CurrentCreditLines_Bins
IV= 0.0307
WoE
0.4
0.2
0.0
-0.2
Target Rate
0.3
0.2
0.1
0.0
Bin Frequency
4000
3000
2000
1000
0
[0,3)
[3,5)
[5,6)
[6,7)
[7,8)
[8,9)
[9,11)
[11,13)
[13,15)
[15,19)
[19,Inf)
Missing

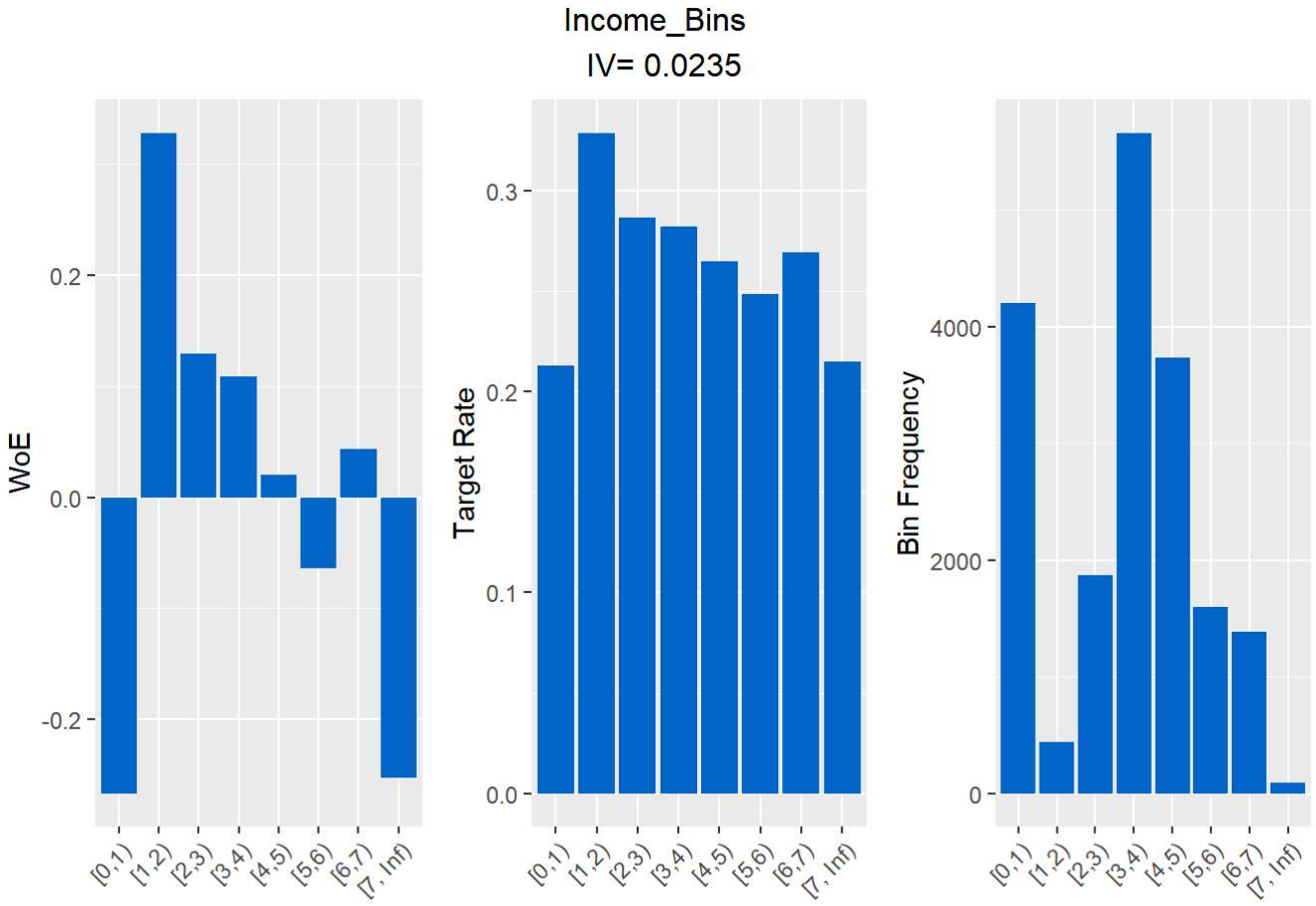

OpenCreditLines_Bins
IV= 0.0313
WoE
0.50
0.25
0.00
-0.25
Target Rate
0.3
0.2
0.1
0.0
Bin Frequency
4000
3000
2000
1000
0
[0,3)
[3,4)
[4,5)
[5,6)
[6,7)
[7,8)
[8,9)
[9,11)
[11,13)
[13,16)
[16,28)
[28,Inf)
Missing


RevolvingCreditBalance_Bins
IV= 0.0547
WoE
Target Rate
Bin Frequency
[0,1)
[1,550)
[550,1248)
[1248,2222)
[2222,3485)
[3485,5172)
[5172,7425)
[7425,10521)
[10521,15406)
[15406,23463)
[23463,43535)
[43535,Inf)
Missing


BankcardUtilization_Bins
IV= 0.0879
WoE
Target Rate
Bin Frequency
[0,0.01)
[0.01,0.16)
[0.16,0.32)
[0.32,0.46)
[0.46,0.59)
[0.59,0.7)
[0.7,0.8)
[0.8,0.88)
[0.88,0.94)
[0.94,0.99)
[0.99,1.16)
[1.16,Inf)
Missing


EmploymentStatus_Bins
IV= 0.0305
WoE
0.2
0.0
-0.2
-0.4
-0.6
Target Rate
0.3
0.2
0.1
0.0
Bin Frequency
10000
5000
0
Full-time
Not available
Not employed
Part-time
Retired
Self-employed


LengthStatusMonths_Bins
IV= 0.0172
WoE
Target Rate
Bin Frequency
[0,1)
[1,6)
[6,12)
[12,19)
[19,27)
[27,38)
[38,53)
[53,72)
[72,94)
[94,132)
[132,221)
[221, Inf)




```
#How to manually change bin cuts
#data$UtilityBins=cut(data$Utility,breaks=c(-Inf,0,.01,.25,.5,.90,Inf),right = F)


WOEdata = WOEProfet(data = binData, id = "ID", target = "Bad")
```

```
Warning: package 'RSQLite' was built under R version 4.4.2
```

```
names(WOEdata)
```

```
[1] "Bin"  "WOE"  "IV"   "vars"
```

```
#Get information values for each variable
head(WOEdata$IV[order(-WOEdata$IV$IV),],20)
```

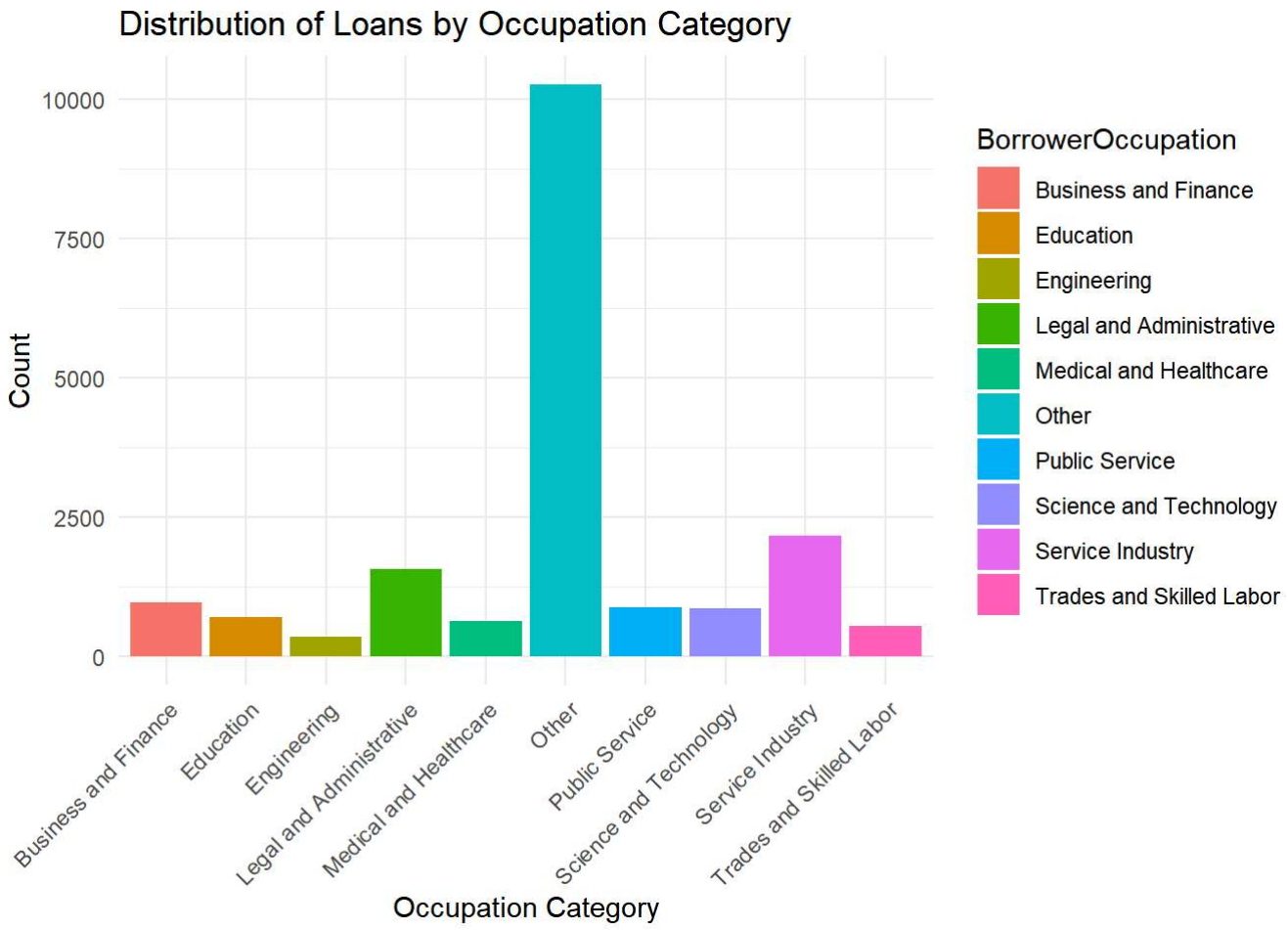
```
                      Variable         IV
8                 MemberKey_Bins 2.20482541
6                ListingKey_Bins 2.19680068
10                  LoanKey_Bins 2.19587931
2              BorrowerCity_Bins 0.27254881
16     InquiriesLast6Months_Bins 0.21855117
22      BankcardUtilization_Bins 0.08790324
```

```
11     CurrentDelinquencies_Bins 0.08054334
17         AmountDelinquent_Bins 0.07218131
3             BorrowerState_Bins 0.06353524
21   RevolvingCreditBalance_Bins 0.05468491
7             ListingNumber_Bins 0.04147971
26       BorrowerOccupation_Bins 0.03637143
13 PublicRecordsLast10Years_Bins 0.03463367
12  DelinquenciesLast7Years_Bins 0.03352331
20          OpenCreditLines_Bins 0.03129187
19       CurrentCreditLines_Bins 0.03066352
23         EmploymentStatus_Bins 0.03049373
9            AmountBorrowed_Bins 0.02762367
25                  Income_Bins 0.02354100
4         DebtToIncomeRatio_Bins 0.02253717
```

```r
view(WOEdata$WOE)
```

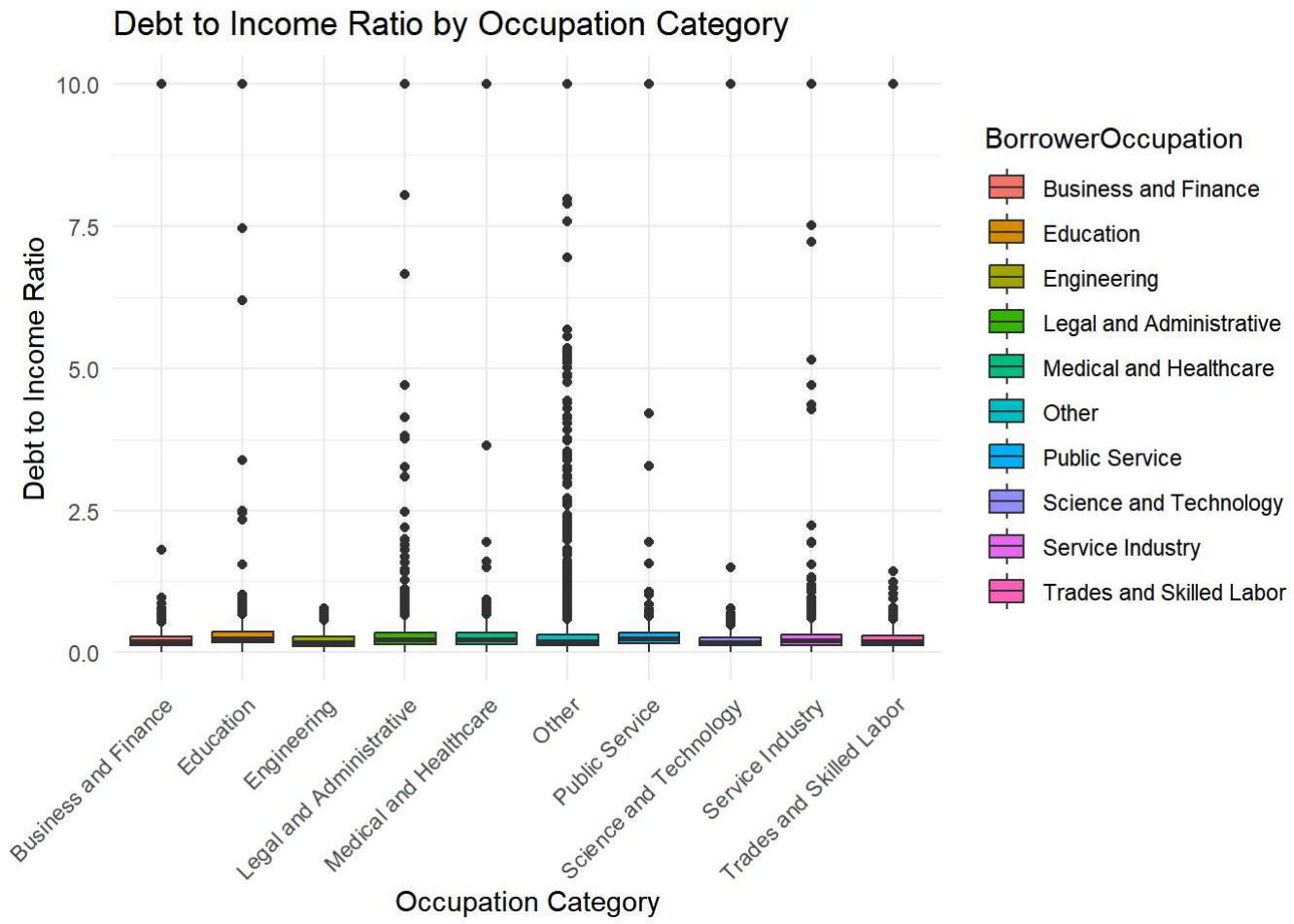
EDA Plots

```r
# Bar plot for Distribution of Loans by Occupation Category
ggplot(input_vars, aes(x = BorrowerOccupation, fill = BorrowerOccupation)) +
  geom_bar() +
  theme_minimal() +
  labs(title = "Distribution of Loans by Occupation Category",
       x = "Occupation Category", y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

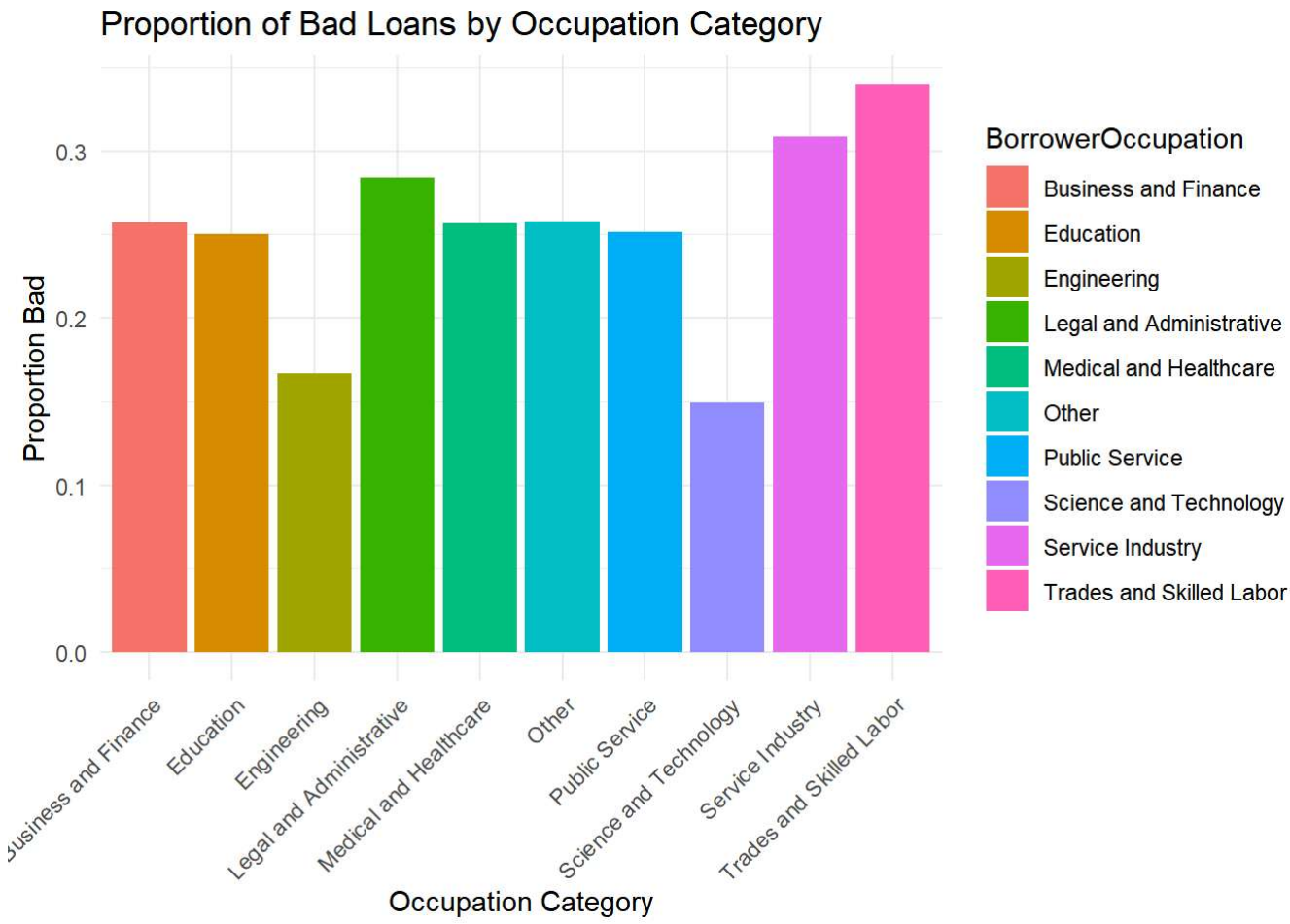



```
# Box plot for Debt to Income Ratio by Occupation Category
ggplot(input_vars, aes(x = BorrowerOccupation, y = DebtToIncomeRatio, fill = BorrowerOccupation))
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Debt to Income Ratio by Occupation Category",
       x = "Occupation Category", y = "Debt to Income Ratio") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
Warning: Removed 659 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```




```
bad_loans_by_occupation <- input_vars %>%
  group_by(BorrowerOccupation) %>%
  summarise(ProportionBad = mean(Bad, na.rm = TRUE))

# Bar plot for Proportion of Bad Loans by Occupation Category
ggplot(bad_loans_by_occupation, aes(x = BorrowerOccupation, y = ProportionBad, fill = BorrowerOcc
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Proportion of Bad Loans by Occupation Category",
       x = "Occupation Category", y = "Proportion Bad") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

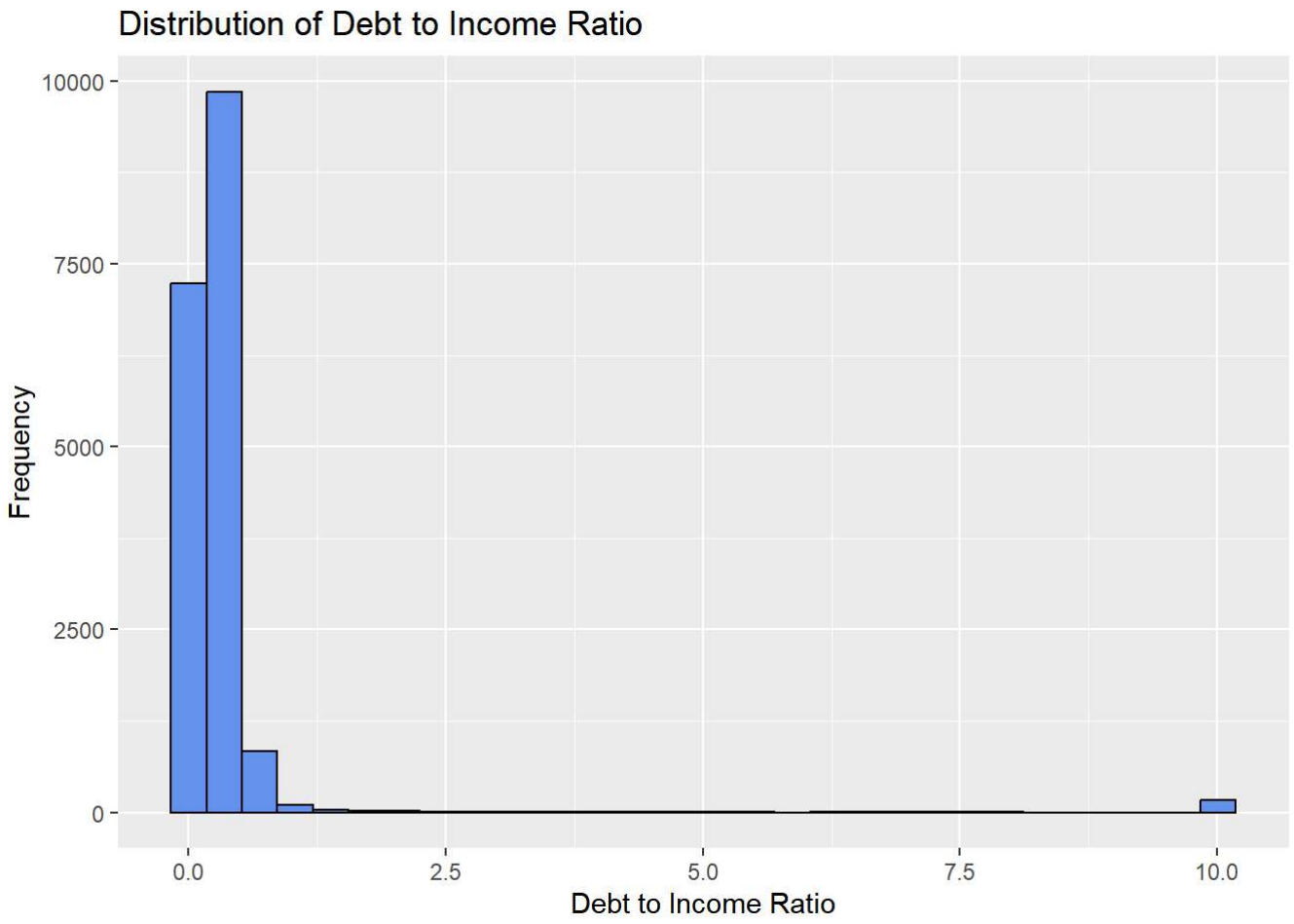


Financial Health Metrics Analysis

1. Debt-to-Income Ratio

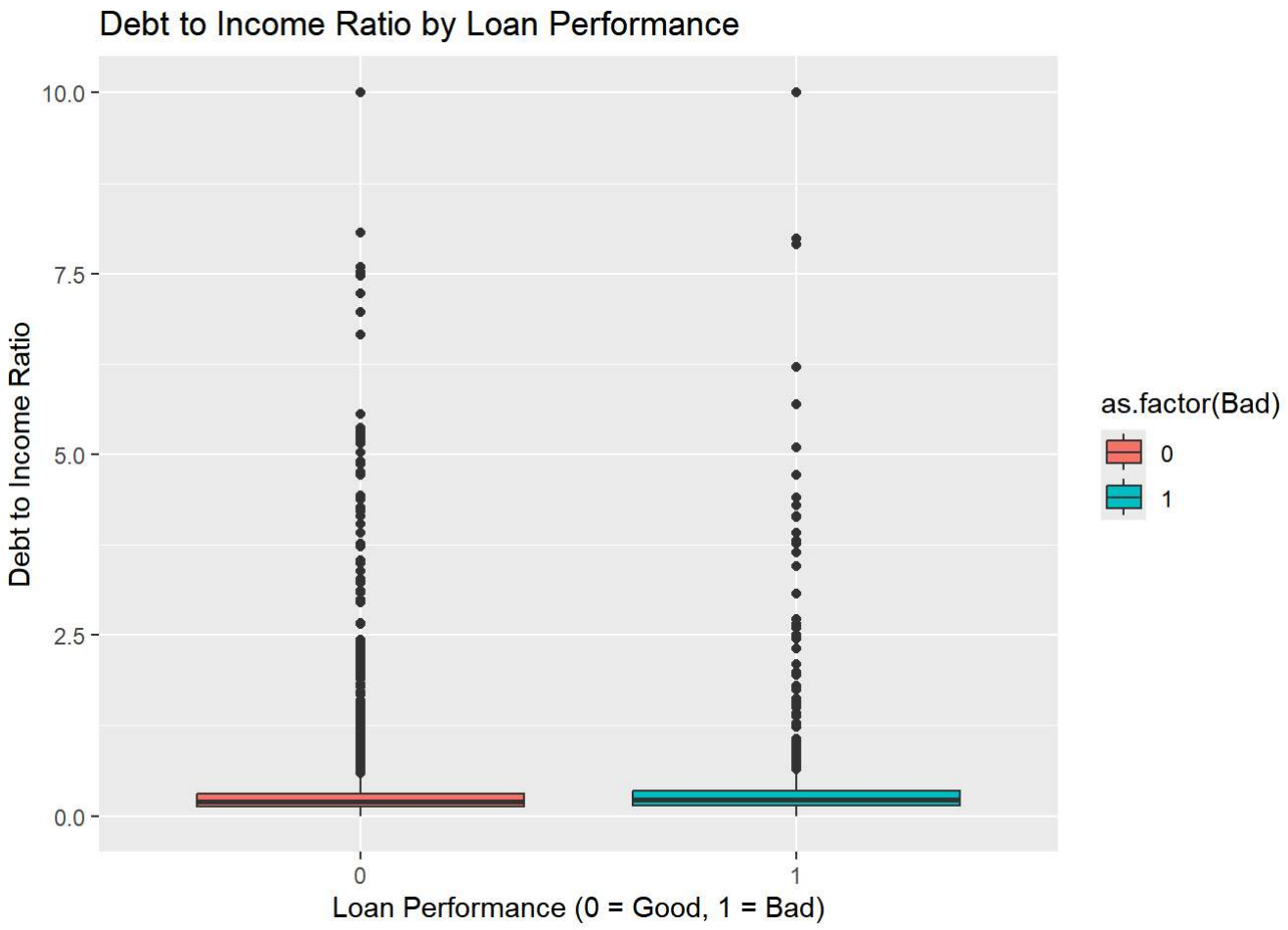
```
# Histogram of Debt to Income Ratio
ggplot(input_vars, aes(x = DebtToIncomeRatio)) +
  geom_histogram(bins = 30, fill = "cornflowerblue", color = "black") +
  ggtitle("Distribution of Debt to Income Ratio") +
  xlab("Debt to Income Ratio") + ylab("Frequency")
```

```
Warning: Removed 659 rows containing non-finite outside the scale range
(`stat_bin()`).
```




```
# Boxplot of Debt to Income Ratio by Bad status
ggplot(input_vars, aes(x = as.factor(Bad), y = DebtToIncomeRatio, fill = as.factor(Bad))) +
  geom_boxplot() +
  ggtitle("Debt to Income Ratio by Loan Performance") +
  xlab("Loan Performance (0 = Good, 1 = Bad)") + ylab("Debt to Income Ratio")
```

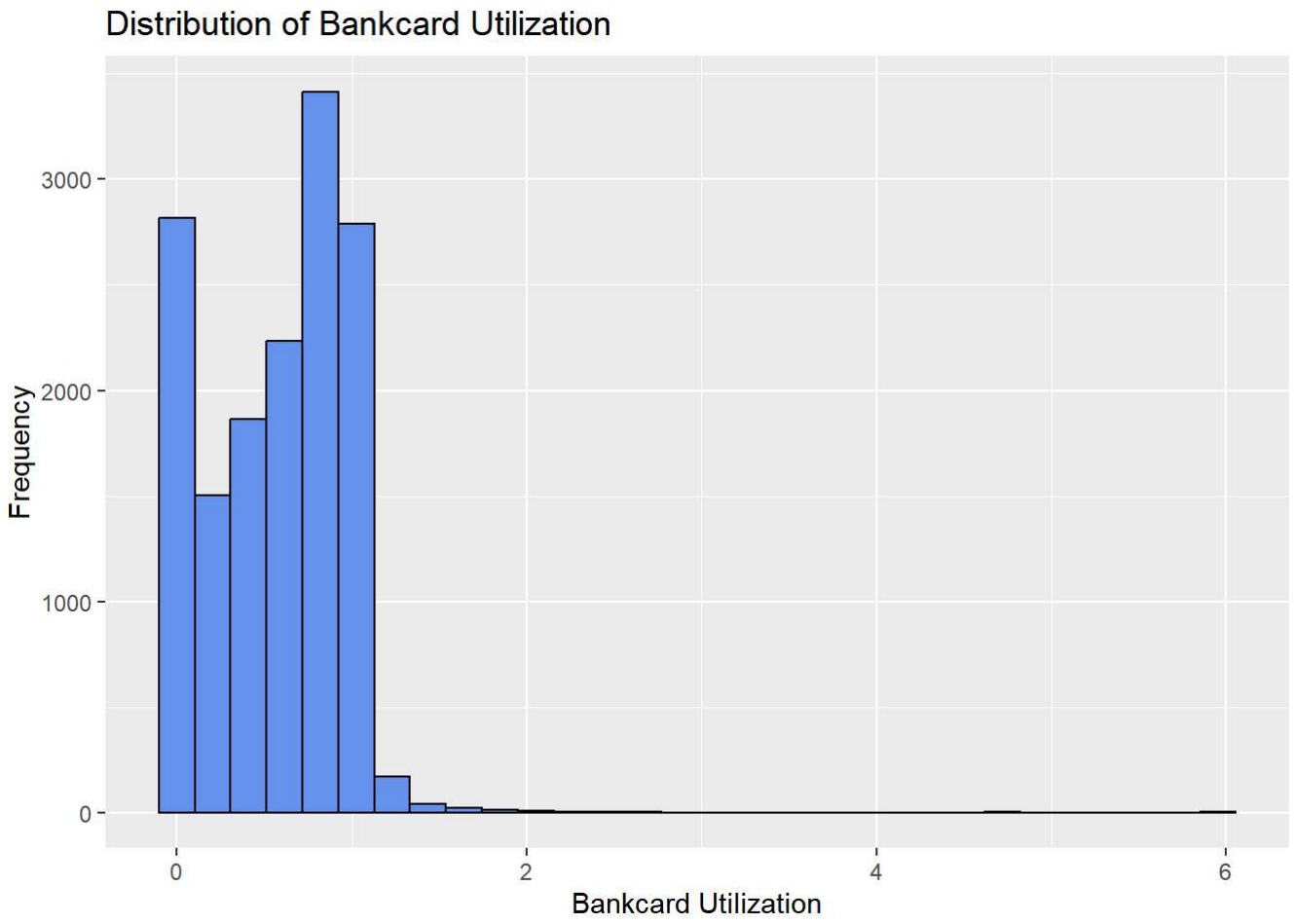
```
Warning: Removed 659 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```



2. Credit Utilization

```
# Histogram of Bankcard Utilization
ggplot(input_vars, aes(x = BankcardUtilization)) +
  geom_histogram(bins = 30, fill = "cornflowerblue", color = "black") +
  ggtitle("Distribution of Bankcard Utilization") +
  xlab("Bankcard Utilization") + ylab("Frequency")
```

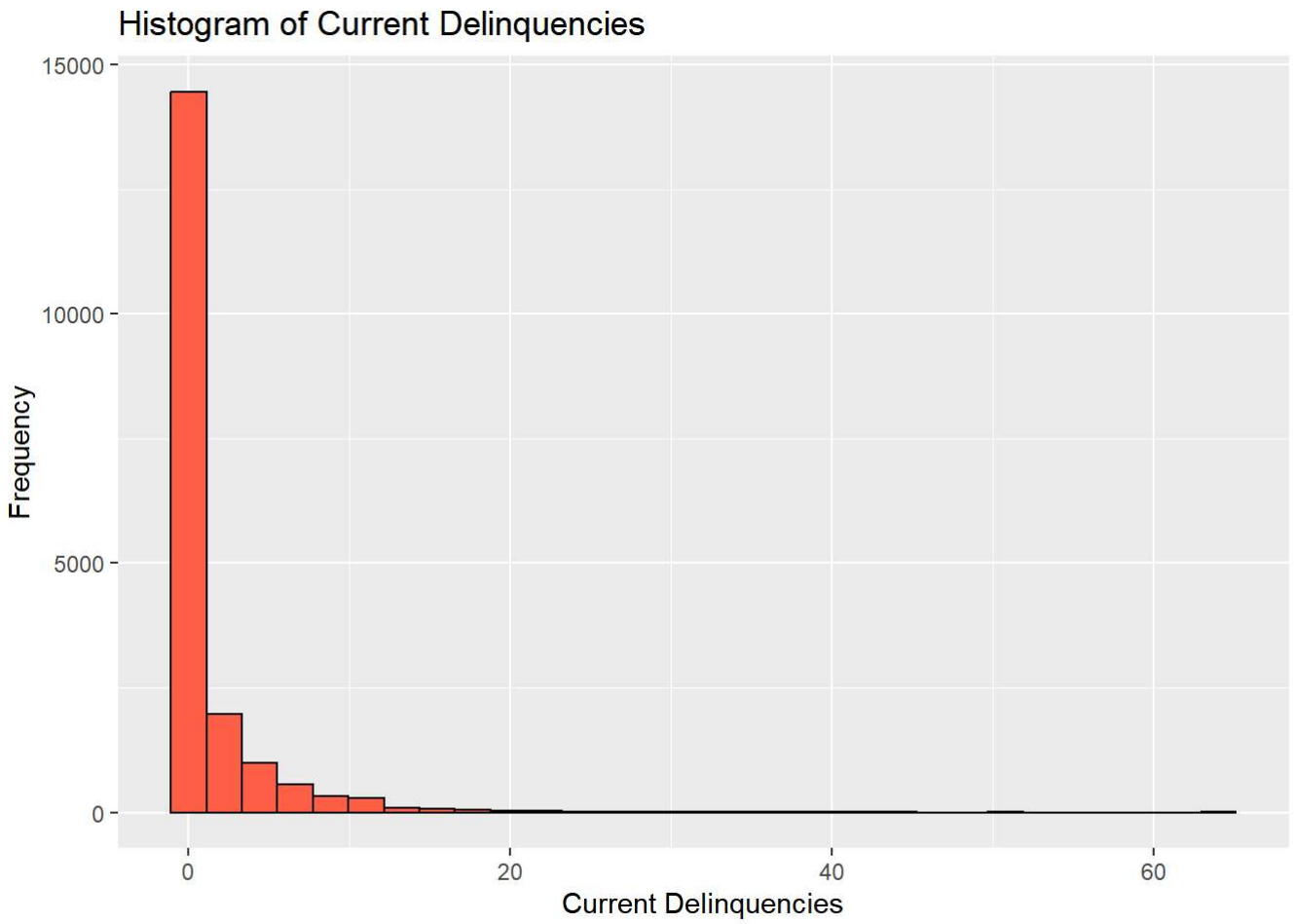
```
Warning: Removed 4105 rows containing non-finite outside the scale range
(`stat_bin()`).
```



3. Delinquency Rates

```
# Histogram for Current Delinquencies
ggplot(input_vars, aes(x = CurrentDelinquencies)) +
  geom_histogram(bins = 30, fill = "tomato", color = "black") +
  ggtitle("Histogram of Current Delinquencies") +
  xlab("Current Delinquencies") + ylab("Frequency")
```

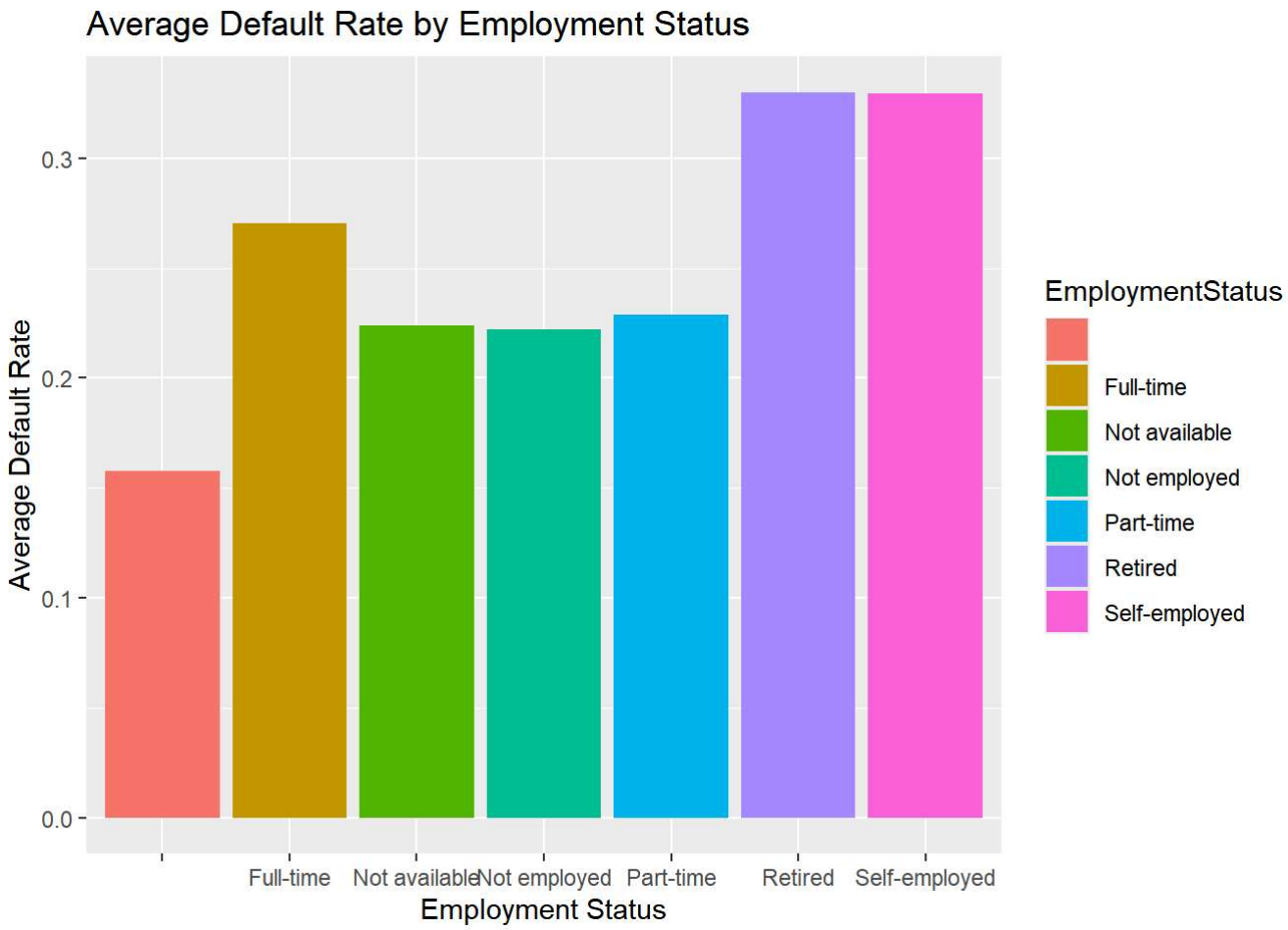
```
Warning: Removed 61 rows containing non-finite outside the scale range
(`stat_bin()`).
```



Borrower Demographics Analysis

1. Employment Status

```
# Bar plot of Loan Performance by Employment Status
employment_status_summary <- input_vars %>%
  group_by(EmploymentStatus) %>%
  summarise(AverageDefault = mean(Bad, na.rm = TRUE))

ggplot(employment_status_summary, aes(x = EmploymentStatus, y = AverageDefault, fill = Employment
  geom_bar(stat = "identity") +
  ggtitle("Average Default Rate by Employment Status") +
  xlab("Employment Status") + ylab("Average Default Rate")
```



Split Data Into Train/Test Sets

Split the dataset into training (60%) and testing (40%) sets

train_data: 11,393
test_data: 7,594

```
#Split into training and testing datasets
set.seed(123)

# # Split the dataset into training (60%) and testing (40%) sets
# split_index <- createDataPartition(input_vars$Bad, p = 0.60, list = FALSE)
# # Create training set
# train_data <- input_vars[split_index, ]
# # Create testing set
# test_data <- input_vars[-split_index, ]

# Split the data into training (60%) and testing (40%) sets
split_index <- createDataPartition(WOEdata$WOE$Bad, p = 0.60, list = FALSE)
# Create training set
train_data <- WOEdata$WOE[split_index, ]
# Create testing set
```

```
test_data <- WOEdata$WOE[-split_index, ]
#Sizes of each data subset
dim(train_data)
```

```
[1] 11393    28
```

```
dim(test_data)
```

```
[1] 7594   28
```

Model Performance Functions

```
#Input:
# X: A data frame with 2+n columns (ID, Bad, model1.pred, ..., modeln.pred)
# n: An integer n <=6 which indicates how many models scored.
# names: A vecor of the model(s) names
# plot: A boolean (True/False) which control the display of the ROC plot

roc <- function(X,n=1,names=c("Logistic Model", "WOE Model"),plot=T){
  library(ROCR)
  color <- c('navy','cyan4', 'gold2', 'yellowgreen','coral','chocolate','red')
  auc <- c()

  if(n>1){
    roc.preds <- prediction(X[,3],X[,2],label.ordering=NULL)
    roc <-  performance(roc.preds,measure='tpr',x.measure='fpr')

    if(plot==T){
      plot(roc@x.values[[1]],roc@y.values[[1]],main='ROC Chart',col=color[1],type='l',xaxt='n',ya
           xlab="False Positive Rate",ylab='True Positive Rate')
      abline(0,1,lty=2,col='black')
      axis(1,seq(0,1,.1),lwd=2)
      axis(2,seq(0,1,.1),lwd=2)
      legend('bottomright',names,lty=1,col=color[1:n],box.lwd=1,lwd=3)
    }
    t <- performance(roc.preds,measure='auc')
    auc <- c(auc, t@y.values)
    for(i in 2:n){
      roc.preds <- prediction(X[,2+i],X[,2],label.ordering=NULL)
      roc <- performance(roc.preds,measure='tpr',x.measure='fpr')
      if(plot==T){
        points(roc@x.values[[1]],roc@y.values[[1]],type='l',col=color[i])
      }
      t <- performance(roc.preds,measure='auc')
      auc <- c(auc, t@y.values)
    }
    names(auc) <- names
    return(auc)
```

```
  }else{
    roc.preds <- prediction(X[,3],X[,2],label.ordering=NULL)
    roc <- performance(roc.preds,measure='tpr',x.measure='fpr')
    if(plot==T){
      plot(roc,main='ROC Chart',col='navy',xlab='False Positive Rate',
           ylab='True Positive Rate')
    }
    auc <- performance(roc.preds,measure='auc')
    return(auc@y.values)
  }
}



#Input:
# X: is data frame with 3 columns (ID, Bad, model1.pred)
# names: A vecor of the model name
# plot: A boolean (True/False) which control the display of the KS plot

ks <- function(X,names=c("Logistic Model"),plot=T){

  A <- X[order(-X[,3]),]
  rank <- rank(-A[,3],ties.method='average')
  A$rank <- rank
  cum_1 <- cumsum(A[,2])
  A$cum_1 <- cum_1
  cum_0 <- cumsum(A[,2]==0)
  A$cum_0 <- cum_0
  percentile <- round(rank/length(A[,1]),4)
  A$percentile <- percentile
  cum_perc_of_1 <- round(cum_1/sum(A[,2]),4)
  A$cum_perc_of_1 <- cum_perc_of_1
  cum_perc_of_0 <- round(cum_0/sum(A[,2]==0),4)
  A$cum_perc_of_0 <- cum_perc_of_0
  diff <- A$cum_perc_of_1 - A$cum_perc_of_0
  ks.stat <- max(diff)

  if(plot==T){
    plot(A$percentile,A$cum_perc_of_1,main=paste("KS Chart\n",names),xlab='%of Sorted Population',
         ylab='True Positive Rate',col='navy',type='l',xaxt='n',yaxt='n')
    lines(A$percentile,A$cum_perc_of_0,col='gold2',type='l')
    abline(0,1,lty=2,col='black')
    abline(v=ks.stat,lty=5,col='red')
    axis(1,seq(0,1,.1),lwd=2)
    axis(2,seq(0,1,.1),lwd=2)
  }
  #return(A)
  return(ks.stat)
}
```

Logistic Regression Model

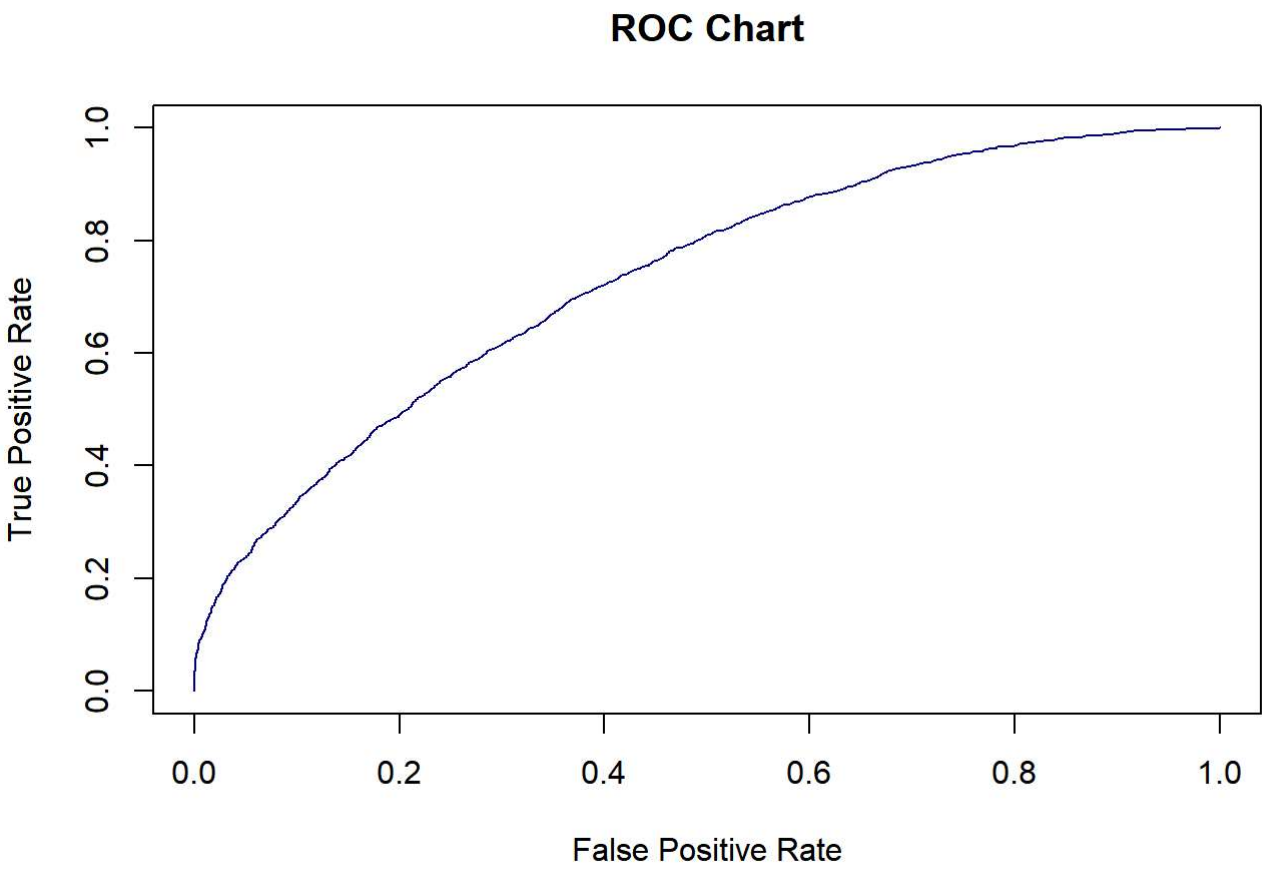
```
log_model <- glm(Bad ~ BorrowerCity_WOE + InquiriesLast6Months_WOE + BankcardUtilization_WOE + Cu
                 data = train_data,
                 family = binomial)

# Print summary of the logistic model
summary(log_model)
```

```
Call:
glm(formula = Bad ~ BorrowerCity_WOE + InquiriesLast6Months_WOE +
    BankcardUtilization_WOE + CurrentDelinquencies_WOE + BorrowerState_WOE +
    RevolvingCreditBalance_WOE + BorrowerOccupation_WOE + PublicRecordsLast10Years_WOE +
    EmploymentStatus_WOE + DebtToIncomeRatio_WOE + FirstRecordedCreditLine_WOE,
    family = binomial, data = train_data)

Coefficients:
                            Estimate Std. Error z value             Pr(>|z|)
(Intercept)                -1.16705    0.02414 -48.346 < 0.0000000000000002
BorrowerCity_WOE            1.58903    0.07572  20.984 < 0.0000000000000002
InquiriesLast6Months_WOE    0.90106    0.05049  17.845 < 0.0000000000000002
BankcardUtilization_WOE     0.61894    0.08873   6.976     0.00000000000304
CurrentDelinquencies_WOE    0.82447    0.08934   9.229 < 0.0000000000000002
BorrowerState_WOE           0.68620    0.09593   7.153     0.00000000000085
RevolvingCreditBalance_WOE  0.46438    0.11970   3.879             0.000105
BorrowerOccupation_WOE      0.72250    0.12987   5.563     0.00000002646772
PublicRecordsLast10Years_WOE 0.44873   0.13119   3.420             0.000625
EmploymentStatus_WOE        0.89106    0.16169   5.511     0.00000003567446
DebtToIncomeRatio_WOE       1.36984    0.16122   8.497 < 0.0000000000000002
FirstRecordedCreditLine_WOE 0.90781    0.16758   5.417     0.00000006052100

(Intercept)                 ***
BorrowerCity_WOE            ***
InquiriesLast6Months_WOE    ***
BankcardUtilization_WOE     ***
CurrentDelinquencies_WOE    ***
BorrowerState_WOE           ***
RevolvingCreditBalance_WOE  ***
BorrowerOccupation_WOE      ***
PublicRecordsLast10Years_WOE ***
EmploymentStatus_WOE        ***
DebtToIncomeRatio_WOE       ***
FirstRecordedCreditLine_WOE ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 12960  on 11392  degrees of freedom
Residual deviance: 11149  on 11381  degrees of freedom
AIC: 11173

Number of Fisher Scoring iterations: 5
```

```r
# Make predictions on the test set
test_data$predictions1 <- predict(log_model, newdata = test_data, type = "response")

# Convert predicted probabilities to binary (0 or 1)
test_data$predicted_bad1 <- ifelse(test_data$predictions1 > 0.5, 1, 0)

# Create the input data frame for the custom ROC and KS functions
roc_ks_data <- data.frame(
  ID = 1:nrow(test_data),  # Creating a unique identifier
  Bad = test_data$Bad,     # Actual values (1 = bad, 0 = good)
  Predicted = test_data$predictions1  # Model-predicted probabilities
)

# Compute ROC Curve and AUC using custom function
auc_value <- roc(roc_ks_data, n = 1, names = "Logistic Model", plot = TRUE)
```

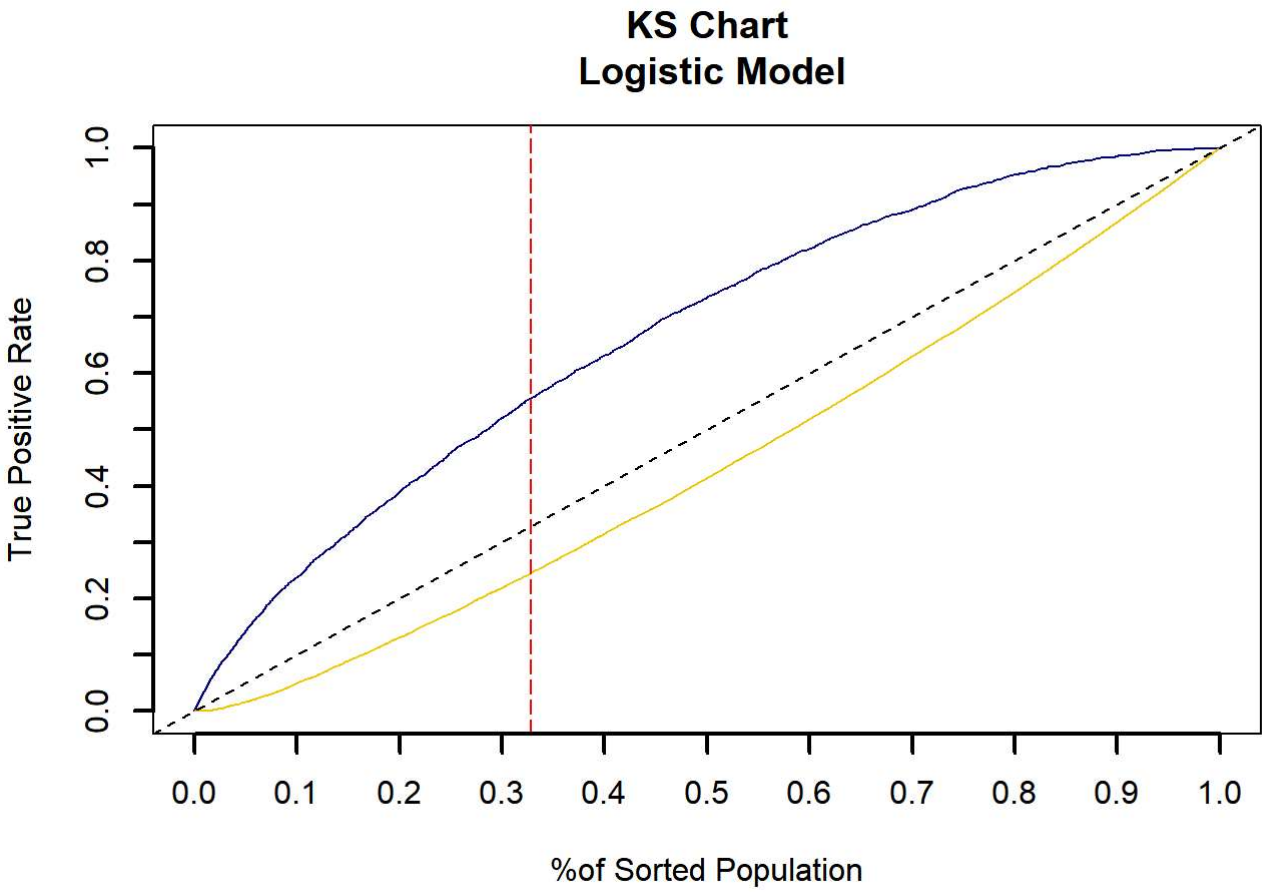
```
Warning: package 'ROCR' was built under R version 4.4.2
```



```
print(paste("AUC:", auc_value))
```

```
[1] "AUC: 0.732003453843767"
```

```
# Compute KS Statistic using custom function
ks_value <- ks(roc_ks_data, names = "Logistic Model", plot = TRUE)
```




```
print(paste("KS Statistic:", ks_value))
```

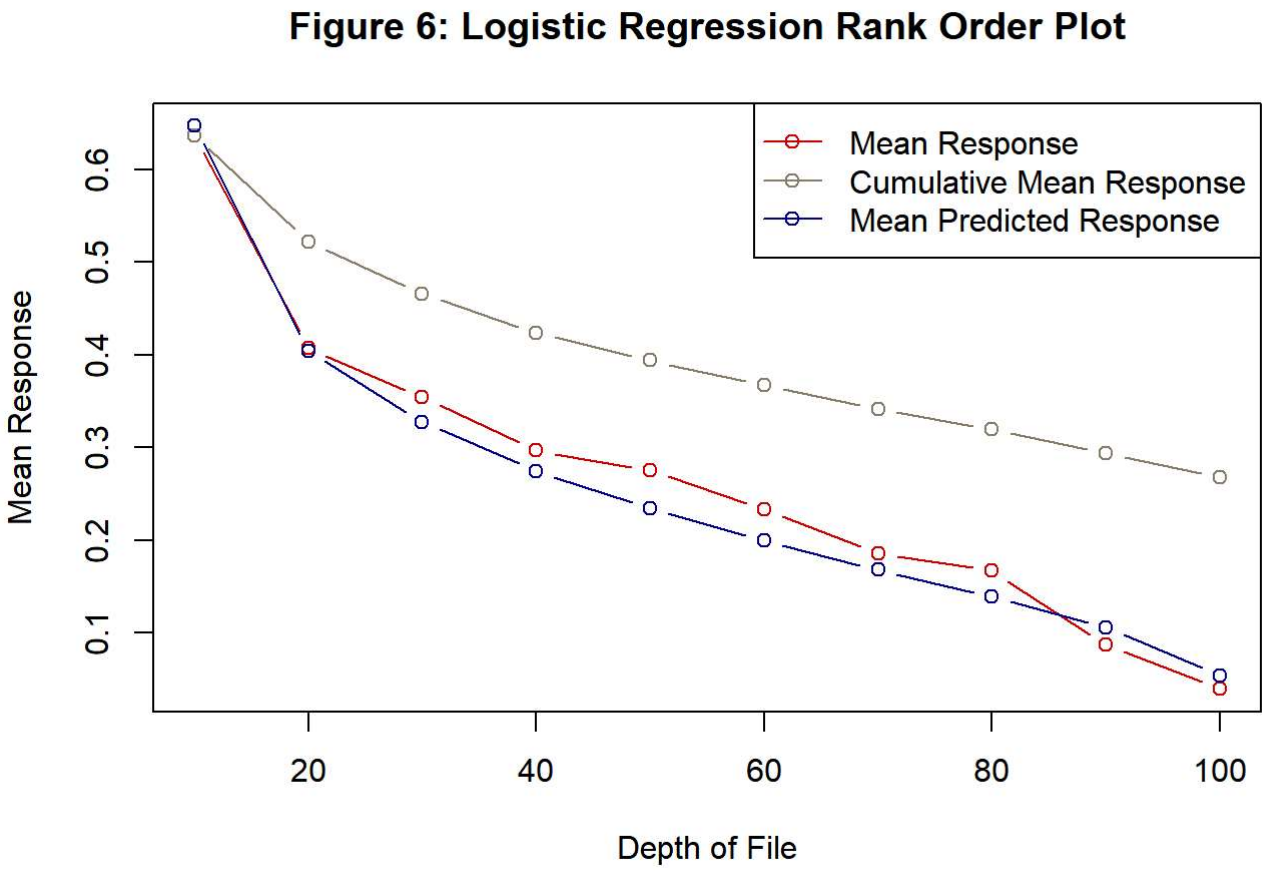
```
[1] "KS Statistic: 0.3287"
```

```
# Create gains table
gains_table1 <- gains(actual = test_data$Bad, predicted = test_data$predictions1)

#Print Gains Table
gains_table1
```

```
Depth                          Cume   Cume Pct                    Mean
 of          Cume     Mean      Mean   of Total    Lift   Cume     Model
File     N     N      Resp      Resp      Resp    Index   Lift     Score
-------------------------------------------------------------------------
```

10	759	759	0.64	0.64	23.7%	237	237	0.65
20	759	1518	0.41	0.52	38.9%	152	195	0.40
30	760	2278	0.35	0.47	52.1%	132	174	0.33
40	759	3037	0.30	0.42	63.2%	111	158	0.27
50	760	3797	0.28	0.39	73.4%	103	147	0.23
60	759	4556	0.23	0.37	82.1%	87	137	0.20
70	759	5315	0.19	0.34	89.0%	69	127	0.17
80	760	6075	0.17	0.32	95.3%	62	119	0.14
90	759	6834	0.09	0.29	98.5%	32	109	0.11
100	760	7594	0.04	0.27	100.0%	15	100	0.05

```
# Plot rank-order plot
plot(gains_table1, main = "Figure 6: Logistic Regression Rank Order Plot")
```




```
#Significant Variables
# Extract the p-values from the summary of the logistic regression model
p_values <- summary(log_model)$coefficients[, "Pr(>|z|)"]

# Select significant variables based on a predetermined significance level (e.g., p-value less tha
significant_vars <- names(p_values[p_values < 0.05])
```

```
# Print the significant variables
print(significant_vars)
```

```
 [1] "(Intercept)"                  "BorrowerCity_WOE"
 [3] "InquiriesLast6Months_WOE"     "BankcardUtilization_WOE"
 [5] "CurrentDelinquencies_WOE"     "BorrowerState_WOE"
 [7] "RevolvingCreditBalance_WOE"   "BorrowerOccupation_WOE"
 [9] "PublicRecordsLast10Years_WOE" "EmploymentStatus_WOE"
[11] "DebtToIncomeRatio_WOE"        "FirstRecordedCreditLine_WOE"
```

Random Forest Model

```
# Split the data into training (60%) and testing (40%) sets
split_index <- createDataPartition(WOEdata$WOE$Bad, p = 0.60, list = FALSE)
# Create training set
train_data <- WOEdata$WOE[split_index, ]
# Create testing set
test_data <- WOEdata$WOE[-split_index, ]

#Sizes of each data subset
dim(train_data)
```

```
[1] 11393    28
```

```
dim(test_data)
```

```
[1] 7594   28
```

```
###################################
#New Type of Model - Random Forest
library(randomForest)
```

```
randomForest 4.7-1.1

Type rfNews() to see new features/changes/bug fixes.


Attaching package: 'randomForest'

The following object is masked from 'package:outliers':

    outlier

The following object is masked from 'package:dplyr':

    combine
```

```
The following object is masked from 'package:ggplot2':

    margin
```

```
# Fit the random forest model
rf_model <- randomForest(Bad ~ InquiriesLast6Months_WOE + BankcardUtilization_WOE + CurrentDelinqu
```

```
Warning in randomForest.default(m, y, ...): The response has five or fewer
unique values.  Are you sure you want to do regression?
```

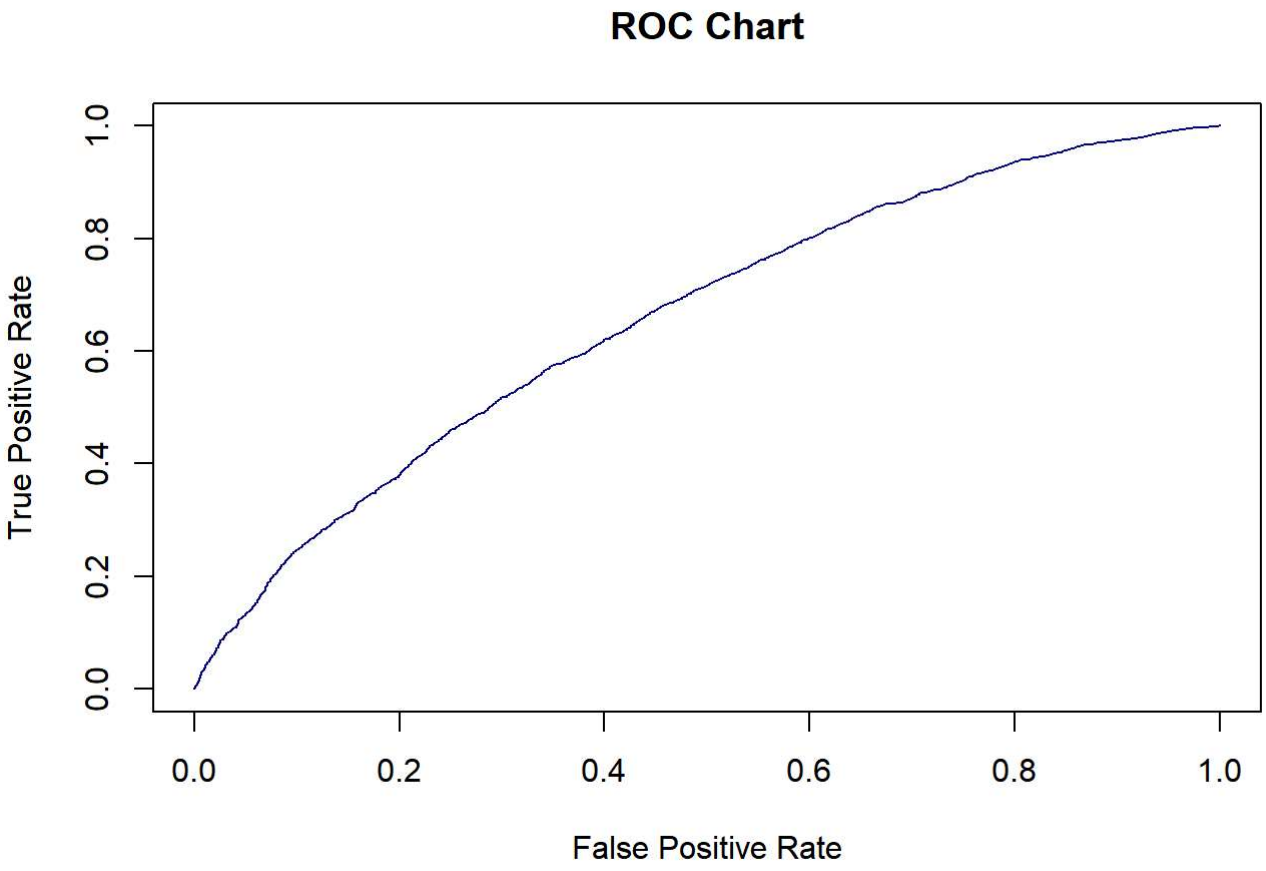
```
# Print the summary of the model
print(rf_model)
```

```

Call:
 randomForest(formula = Bad ~ InquiriesLast6Months_WOE + BankcardUtilization_WOE +
CurrentDelinquencies_WOE + AmountDelinquent_WOE + RevolvingCreditBalance_WOE +
PublicRecordsLast10Years_WOE + DelinquenciesLast7Years_WOE +      OpenCreditLines_WOE +
CurrentCreditLines_WOE + AmountBorrowed_WOE +      Income_WOE + DebtToIncomeRatio_WOE +
TotalCreditLines_WOE +      PublicRecordsLast12Months_WOE, data = train_data)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 0.1791517
                    % Var explained: 6.76
```

```
summary(rf_model)
```

```
                Length Class  Mode
call               3  -none- call
type               1  -none- character
predicted      11393  -none- numeric
mse              500  -none- numeric
rsq              500  -none- numeric
oob.times      11393  -none- numeric
importance        14  -none- numeric
importanceSD       0  -none- NULL
localImportance    0  -none- NULL
proximity          0  -none- NULL
ntree              1  -none- numeric
mtry               1  -none- numeric
forest            11  -none- list
coefs              0  -none- NULL
y              11393  -none- numeric
test               0  -none- NULL
inbag              0  -none- NULL
terms              3  terms  call
```

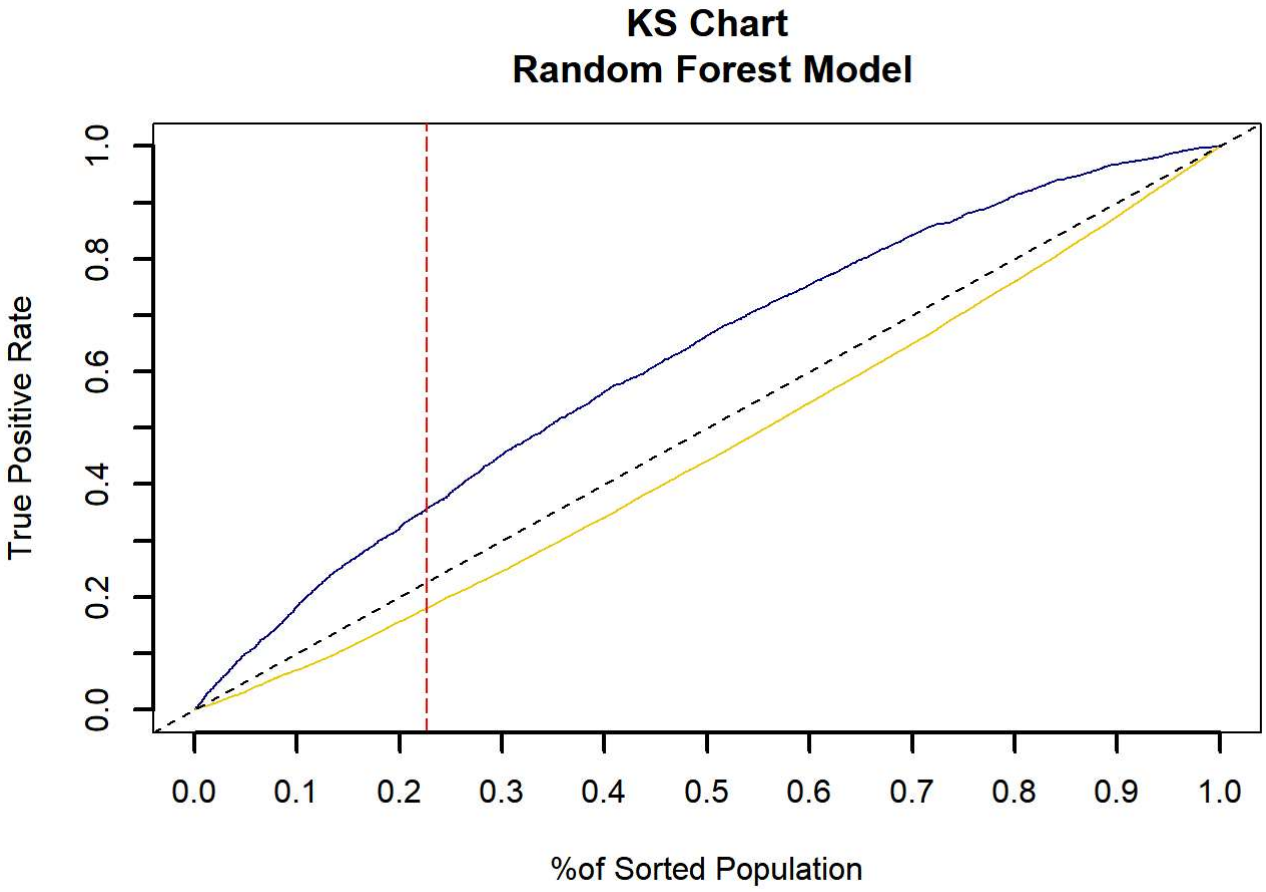
```
# Make predictions on the test set
test_data$rf_predictions <- predict(rf_model, newdata = test_data, type = "response")

# Convert predicted probabilities to binary (0 or 1)
test_data$rf_predicted_bad <- ifelse(test_data$rf_predictions > 0.5, 1, 0)

# Prepare data for ROC and KS functions
rf_roc_ks_data <- data.frame(
  ID = 1:nrow(test_data),  # Unique identifier
  Bad = test_data$Bad,     # Actual values
  Predicted = test_data$rf_predictions  # Model-predicted probabilities
)

# Compute ROC Curve and AUC for Random Forest
rf_auc_value <- roc(rf_roc_ks_data, n = 1, names = "Random Forest Model", plot = TRUE)
```




```
print(paste("Random Forest AUC:", rf_auc_value))
```

```
[1] "Random Forest AUC: 0.659012558879823"
```

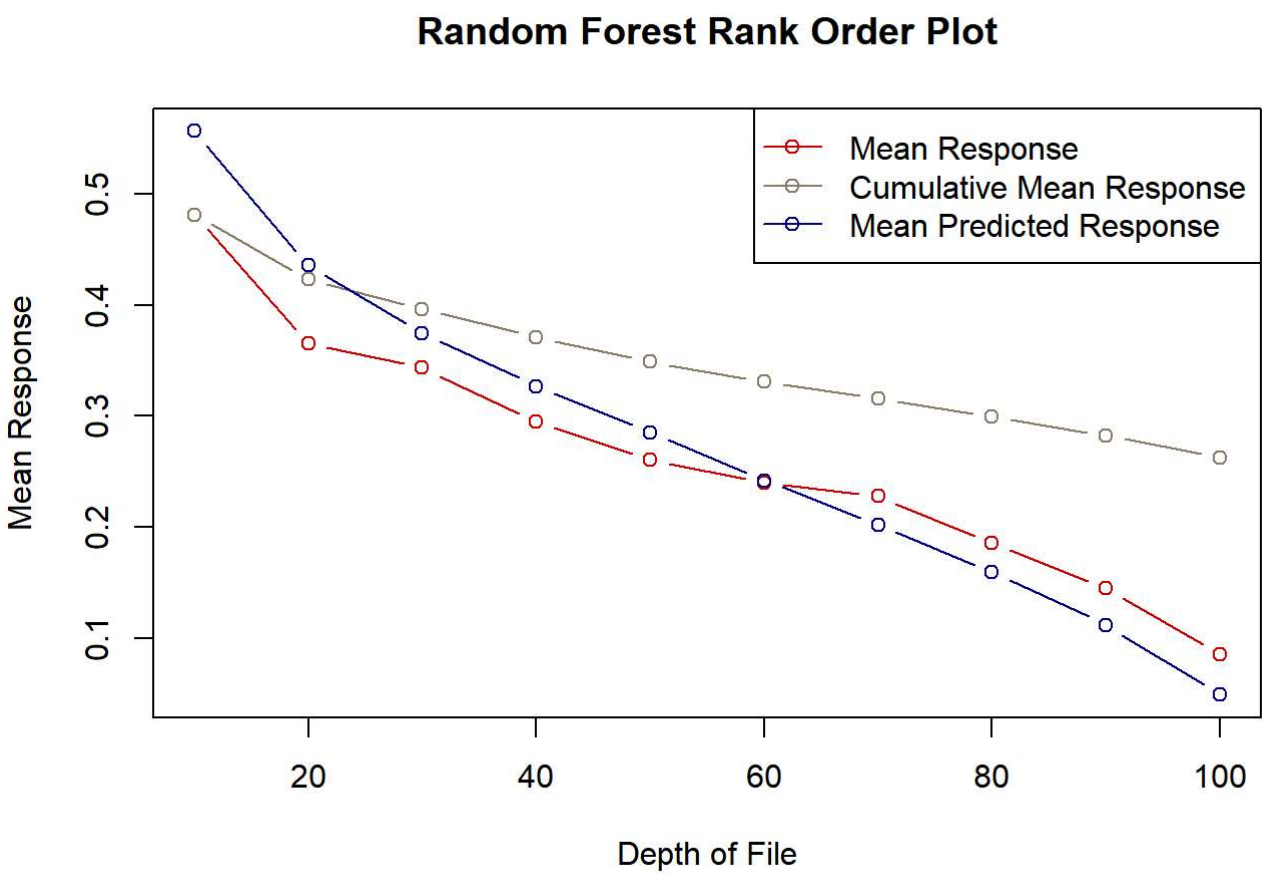
```
# Compute KS Statistic for Random Forest
rf_ks_value <- ks(rf_roc_ks_data, names = "Random Forest Model", plot = TRUE)
```



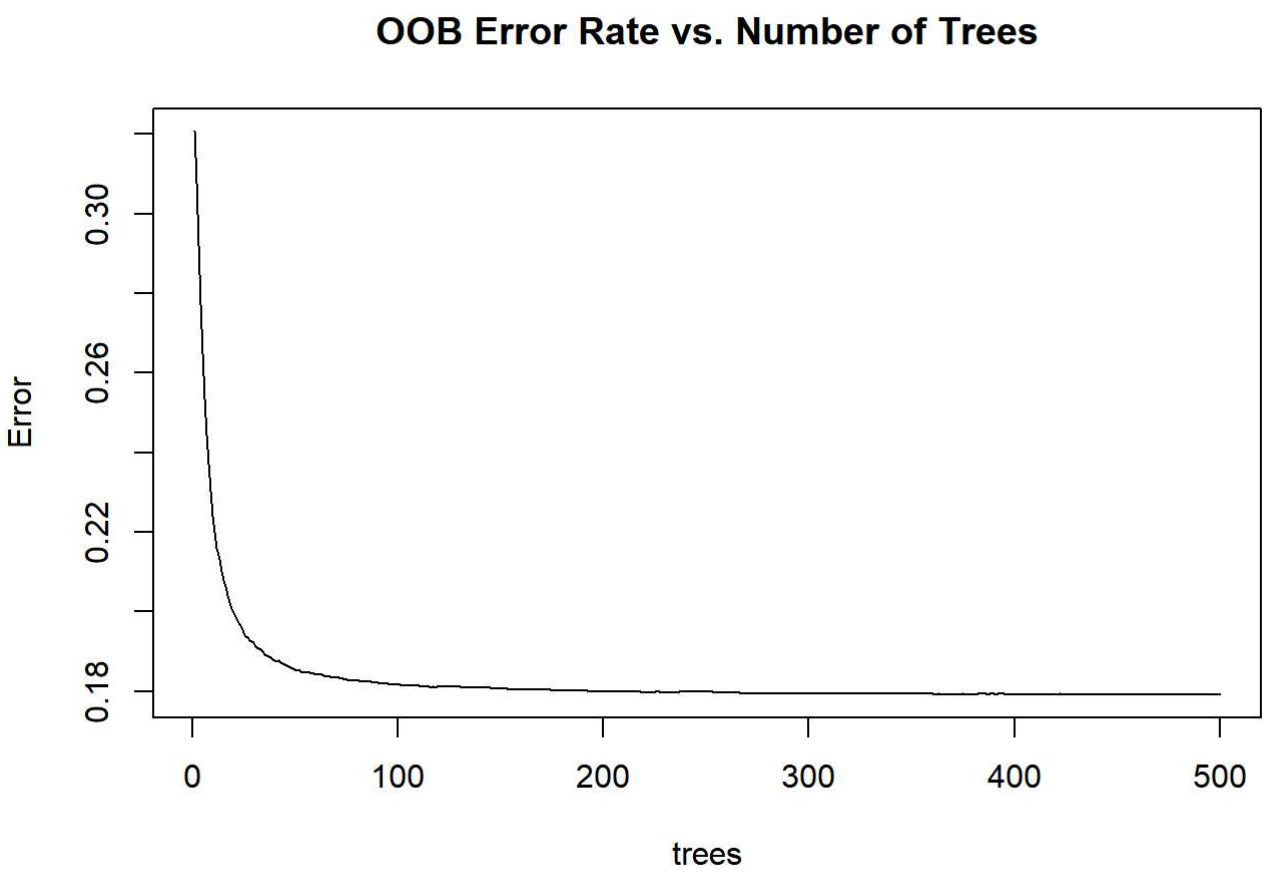

```
print(paste("Random Forest KS Statistic:", rf_ks_value))
```

```
[1] "Random Forest KS Statistic: 0.2265"
```

```
# Create gains table
gains_table_rf <- gains(actual = test_data$Bad, predicted = test_data$rf_predictions)

# Plot rank-order plot
plot(gains_table_rf, main = "Random Forest Rank Order Plot")
```




```
# Plot the OOB error rate with a title
plot(rf_model, main = "OOB Error Rate vs. Number of Trees")
```


OOB Error Rate vs. Number of Trees
Error
0.18
0.22
0.26
0.30
0
100
200
300
400
500
trees