FE_582_bank_loan_status_prediction

Vivek, Ronald, Yatri

5/12/2020

```
###########################################
# Project : Bank Loan Status Prediction
# Course: FE 582
# Professor: Dragos Bozdog
# Semester: Spring 2020
# Team: Viveksinh , Ronald, Yatri
###########################################

rm(list=ls())
getwd()
```

```
## [1] "E:/STEVENS/study/FE-582/project/code"
```

```
# Please change directory path for data files
setwd("E:/STEVENS/study/FE-582/project/bank_loan_status")
train <- read.csv('credit_train.csv', na.strings = c("","NA",'n/a'))
#View(train)
summary(train)
```

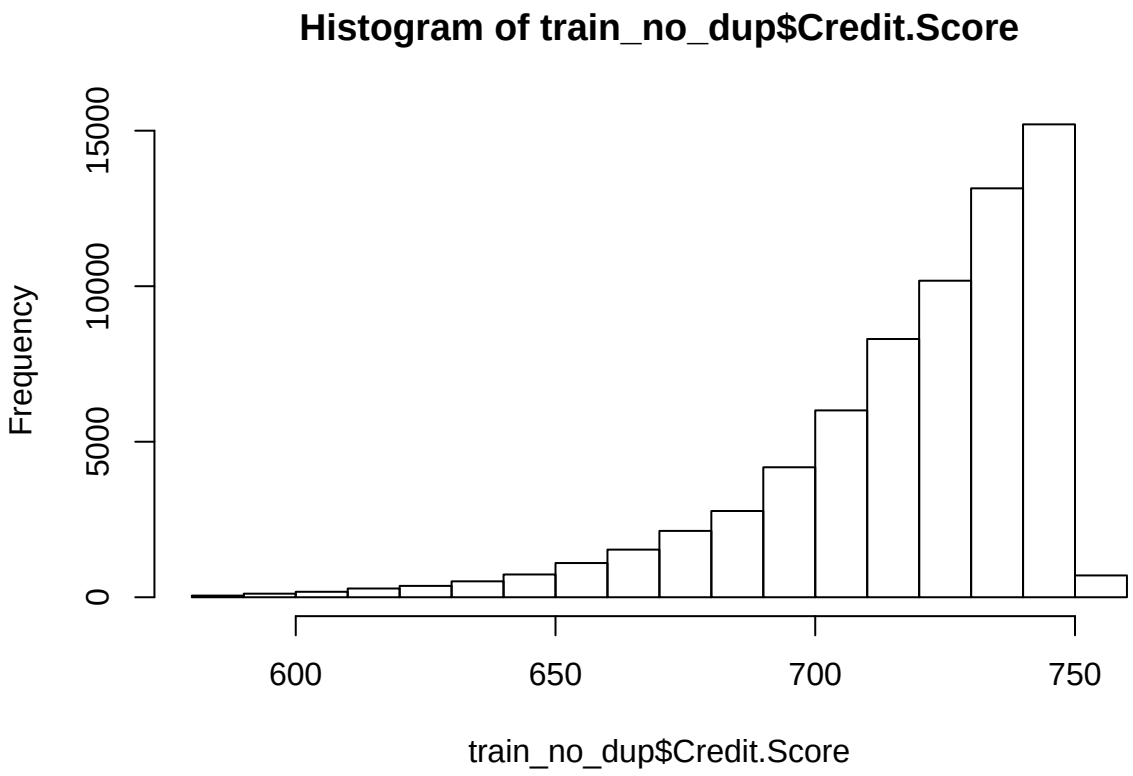
```
##                                  Loan.ID
##  00069ff1-a877-4d35-81be-7cd359b99956:    2
##  000bc65a-6a7c-4566-86f3-203b4ec35eca:    2
##  000c16df-c24f-41cf-a90e-60301d131bb9:    2
##  000ea0cb-8d0e-4284-b8c8-444ffbbe4caf:    2
##  001312a5-ed3c-4930-9525-4d09c55ba7f4:    2
##  0016d326-7878-46bb-9c18-a75af255d7fe:    2
##  (Other)                             :99988
##                                Customer.ID         Loan.Status
##  000877d4-55ed-4126-abda-968f61da7b7f:    2   Charged Off:22639
##  0008bc47-41f5-4e2b-b656-db39bc194a01:    2   Fully Paid :77361
##  000bbb5d-3a62-4712-908e-caacd7a815d5:    2
##  00127cca-7050-4867-9410-8249ef8ad4d2:    2
##  00132610-2f2f-4aeb-a371-2d66aca1248e:    2
##  001534d4-00d8-4b98-acdc-a43a92892e4f:    2
##  (Other)                             :99988
##  Current.Loan.Amount         Term        Credit.Score
##  Min.   :   10802    Long Term :27792   Min.   : 585
##  1st Qu.:  179652    Short Term:72208   1st Qu.: 705
##  Median :  312246                       Median : 724
##  Mean   :11760447                       Mean   :1076
##  3rd Qu.:  524942                       3rd Qu.: 741
##  Max.   :99999999                       Max.   :7510
##                                         NA's   :19154
```

```
##  Annual.Income       Years.in.current.job       Home.Ownership
##  Min.   :    76627   10+ years:31121      HaveMortgage :  214
##  1st Qu.:   848844   2 years  : 9134      Home Mortgage:48410
##  Median :  1174162   3 years  : 8169      Own Home     : 9182
##  Mean   :  1378277   < 1 year : 8164      Rent         :42194
##  3rd Qu.:  1650663   5 years  : 6787
##  Max.   :165557393   (Other)  :32403
##  NA's   :19154       NA's     : 4222
##                Purpose       Monthly.Debt    Years.of.Credit.History
##  Debt Consolidation:78552   Min.   :     0   Min.   : 3.6
##  other             : 6037   1st Qu.: 10214   1st Qu.:13.5
##  Home Improvements : 5839   Median : 16220   Median :16.9
##  Other             : 3250   Mean   : 18472   Mean   :18.2
##  Business Loan     : 1569   3rd Qu.: 24012   3rd Qu.:21.7
##  Buy a Car         : 1265   Max.   :435843   Max.   :70.5
##  (Other)           : 3488
##  Months.since.last.delinquent Number.of.Open.Accounts
##  Min.   :  0.0                Min.   : 0.00
##  1st Qu.: 16.0                1st Qu.: 8.00
##  Median : 32.0                Median :10.00
##  Mean   : 34.9                Mean   :11.13
##  3rd Qu.: 51.0                3rd Qu.:14.00
##  Max.   :176.0                Max.   :76.00
##  NA's   :53141
##  Number.of.Credit.Problems Current.Credit.Balance Maximum.Open.Credit
##  Min.   : 0.0000           Min.   :       0       Min.   :0.000e+00
##  1st Qu.: 0.0000           1st Qu.:  112670       1st Qu.:2.734e+05
##  Median : 0.0000           Median :  209817       Median :4.679e+05
##  Mean   : 0.1683           Mean   :  294637       Mean   :7.608e+05
##  3rd Qu.: 0.0000           3rd Qu.:  367959       3rd Qu.:7.830e+05
##  Max.   :15.0000           Max.   :32878968       Max.   :1.540e+09
##                                                   NA's   :2
##   Bankruptcies      Tax.Liens
##  Min.   :0.0000   Min.   : 0.00000
##  1st Qu.:0.0000   1st Qu.: 0.00000
##  Median :0.0000   Median : 0.00000
##  Mean   :0.1177   Mean   : 0.02931
##  3rd Qu.:0.0000   3rd Qu.: 0.00000
##  Max.   :7.0000   Max.   :15.00000
##  NA's   :204      NA's   :10
```

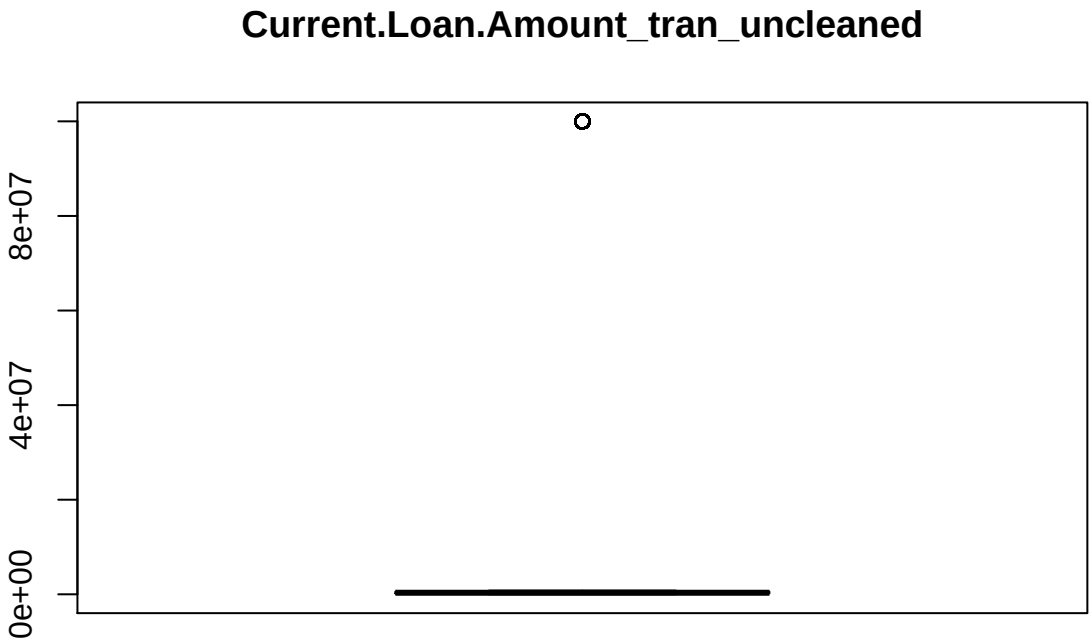
```r
# More than 50000 rows i.e more than 50% of data in months deliquent column is null
train <- train[,-13]
#nrow(train)
#ncol(train)
train <- train[complete.cases(train),]
#nrow(train)
#ncol(train)

#number of duplicate loan IDs
#nrow(train[duplicated(train$Loan.ID),])

train_no_dup <- unique(train, by='Loan.ID')
#nrow(train_no_dup)
```

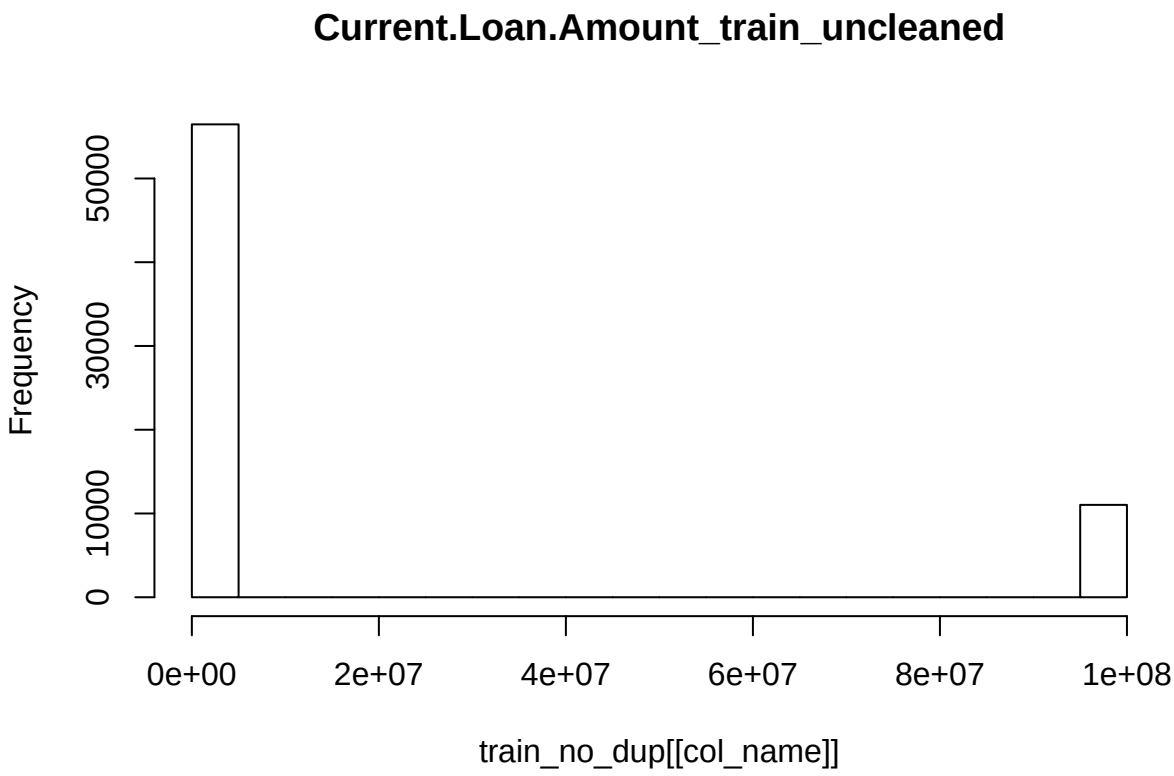
```r
#ncol(train_no_dup)
#View(train_no_dup)

hist(train_no_dup$Credit.Score)
```

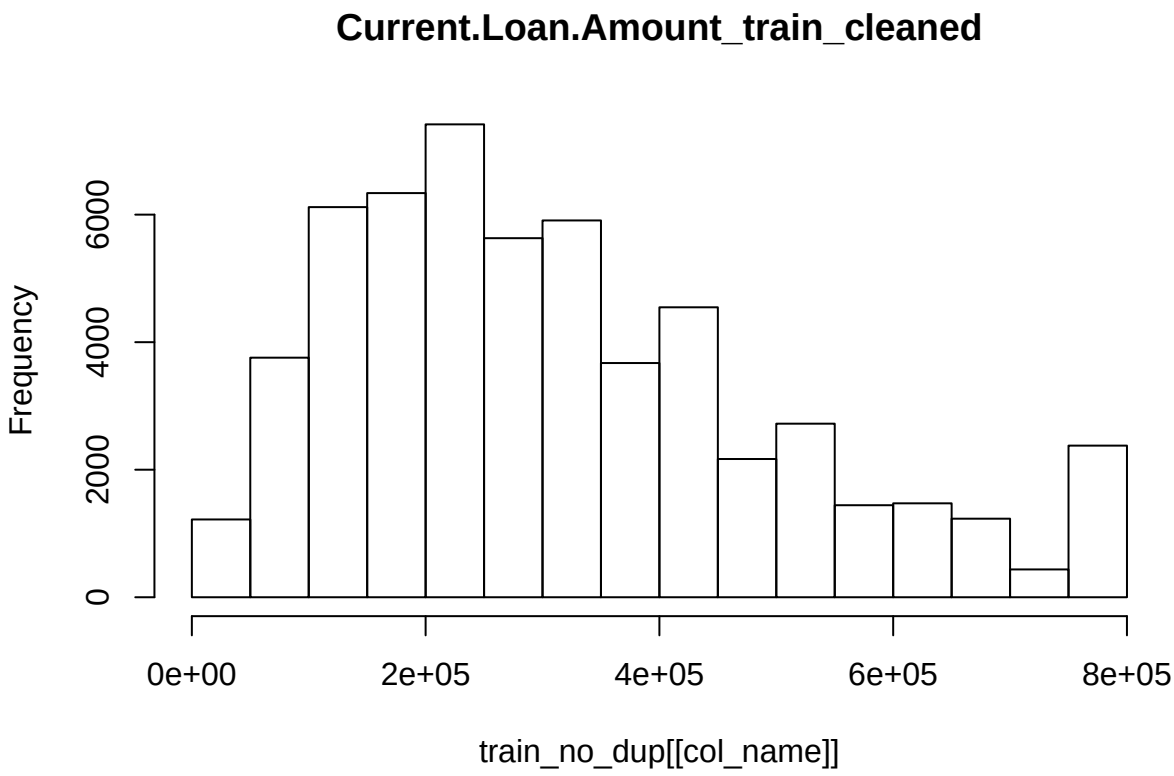

```r
train_no_dup[which(train_no_dup$Credit.Score > 1000),]$Credit.Score <- train_no_dup[which(train_no_dup$C
hist(train_no_dup$Credit.Score)
```

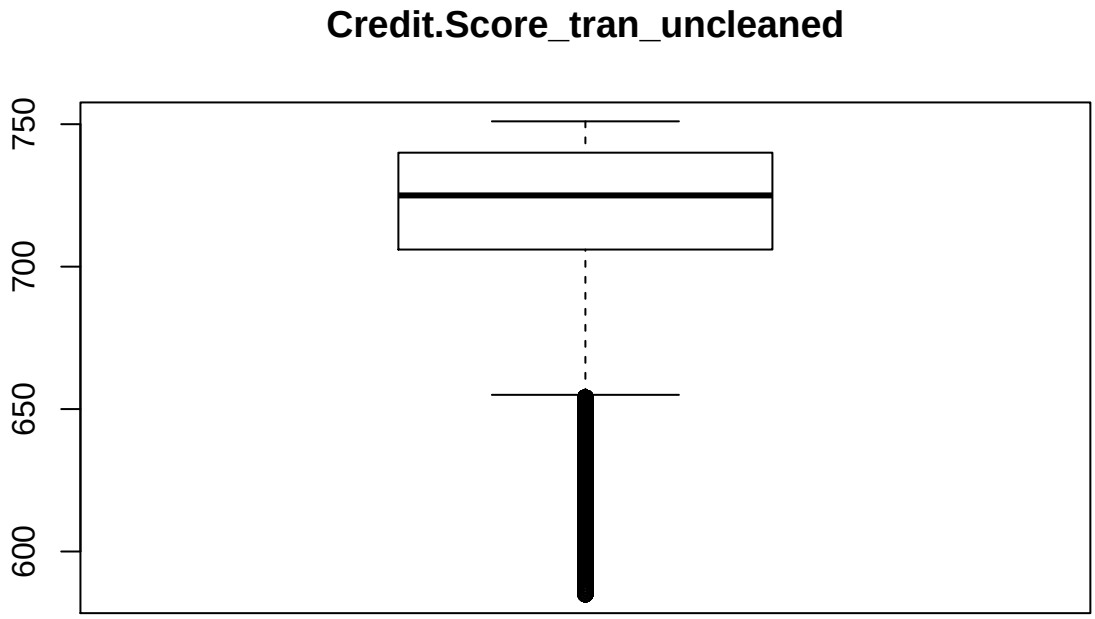


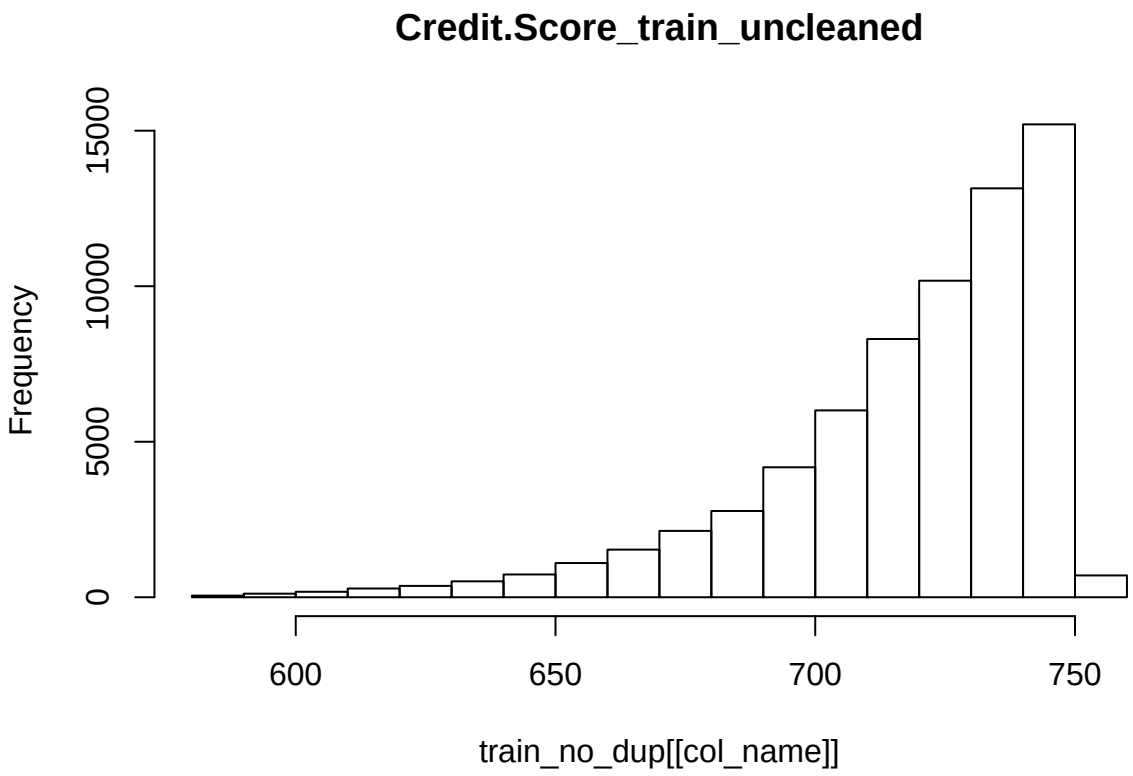

```
outlier_elim <- function(data_col){
  quant_25 <- as.numeric(quantile(data_col)[2])
  quant_75 <- as.numeric(quantile(data_col)[4])
  IQR <- quant_75-quant_25
  data_col[which((data_col < (quant_25 - 1.5*IQR)) | (data_col > (quant_75 + 1.5*IQR)))] <- NA
  return(data_col)
}


num_col_train <- colnames(train_no_dup)[c(4,6,7,11,13,15,16)]

for (col_name in num_col_train) {
  #print(col_name)
  boxplot(train_no_dup[[col_name]], main= paste(col_name,'_tran_uncleaned',sep = ''))
  hist(train_no_dup[[col_name]], main = paste(col_name,'_train_uncleaned',sep = ''))
  train_no_dup[[col_name]] <- outlier_elim(train_no_dup[[col_name]])
  sum(is.na(train_no_dup[[col_name]]))
  hist(train_no_dup[[col_name]], main = paste(col_name,'_train_cleaned',sep = ''))
}
```

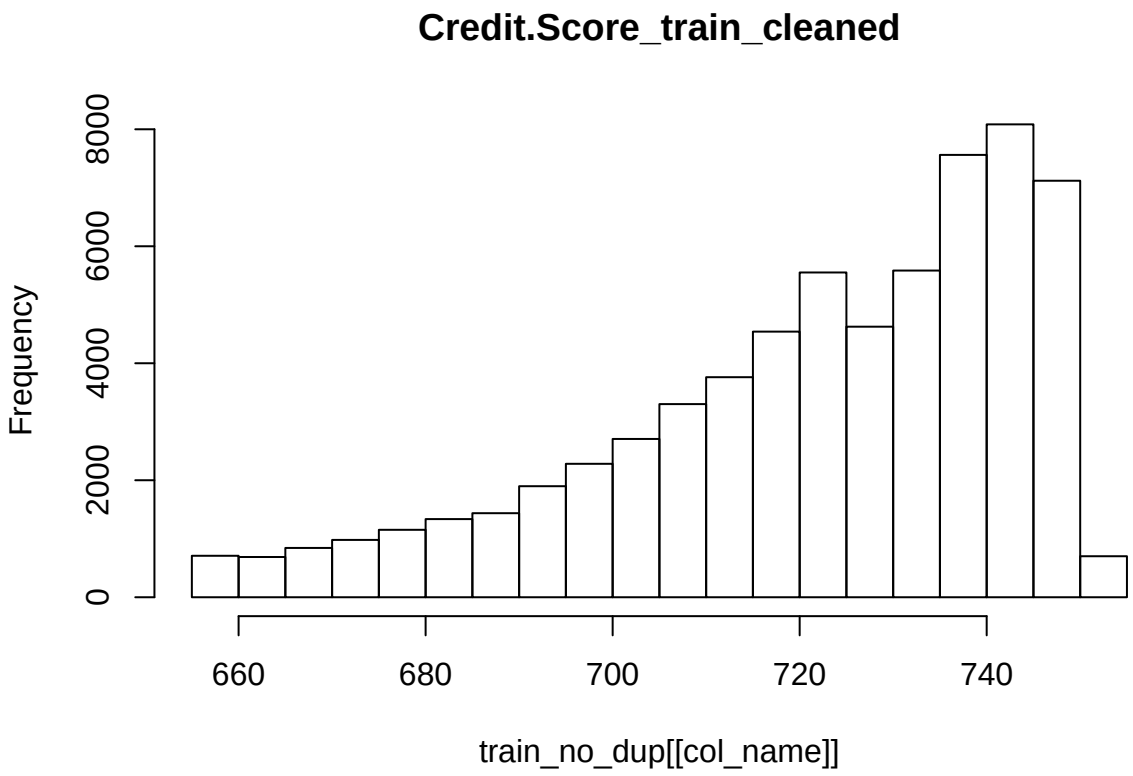

Current.Loan.Amount_tran_uncleaned
0e+00
4e+07
8e+07

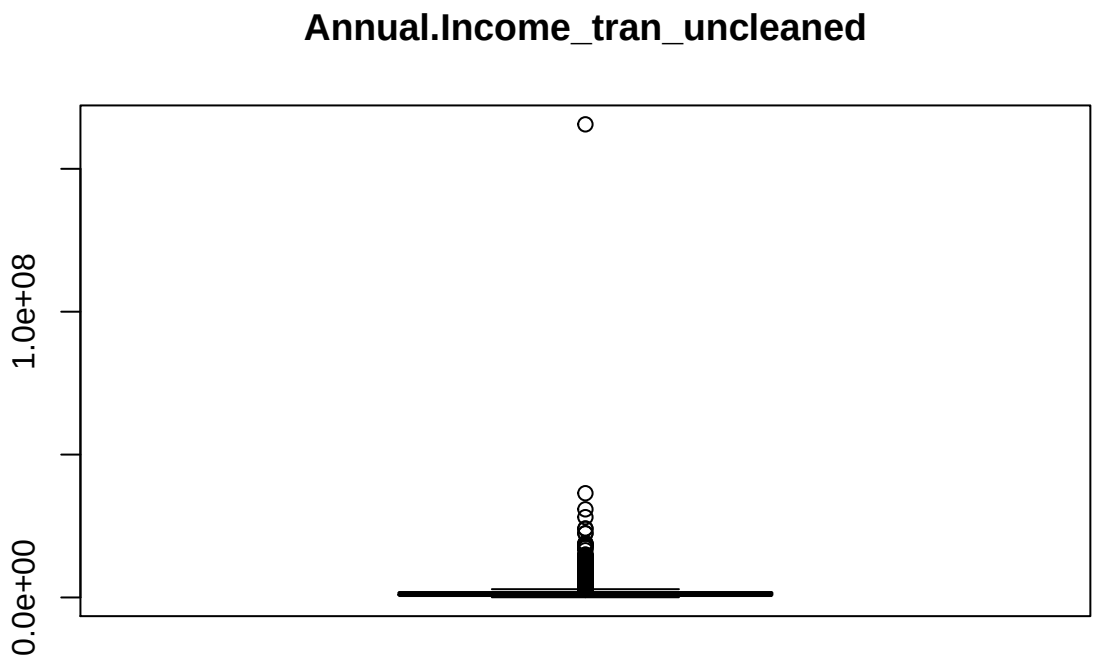

Current.Loan.Amount_train_uncleaned
Frequency
0
10000
30000
50000
0e+00
2e+07
4e+07
6e+07
8e+07
1e+08
train_no_dup[[col_name]]

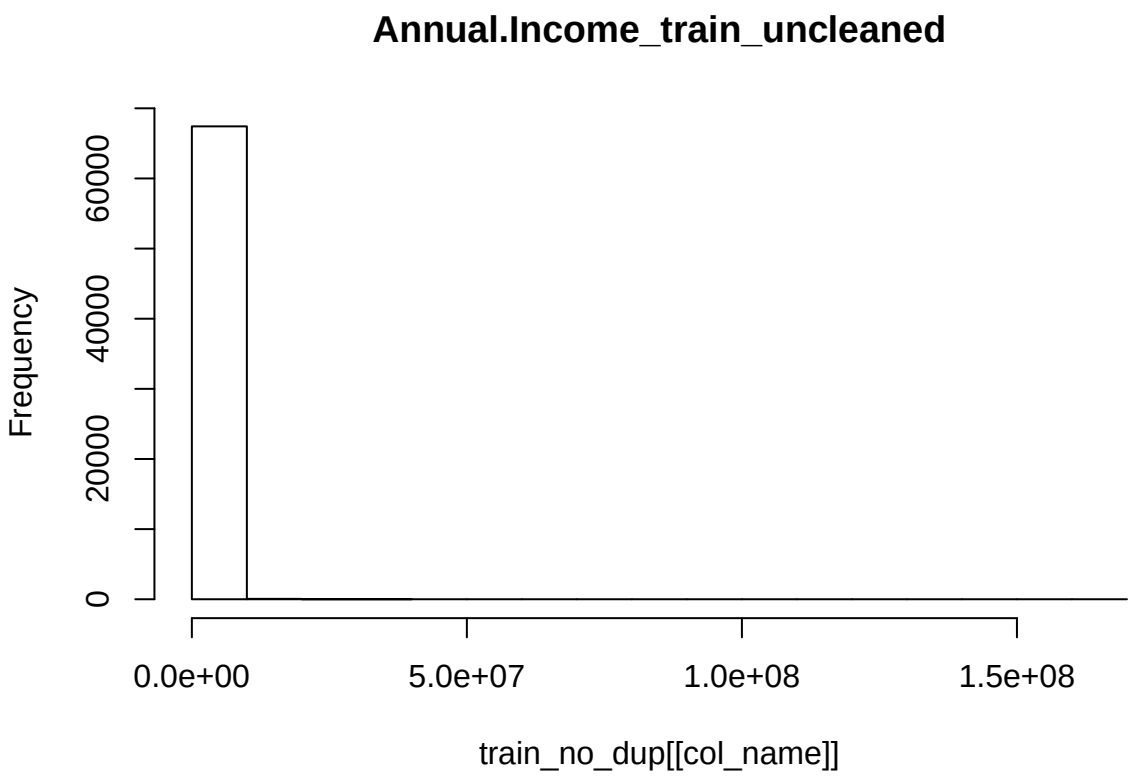

Current.Loan.Amount_train_cleaned
Frequency
0
2000
4000
6000
0e+00
2e+05
4e+05
6e+05
8e+05
train_no_dup[[col_name]]

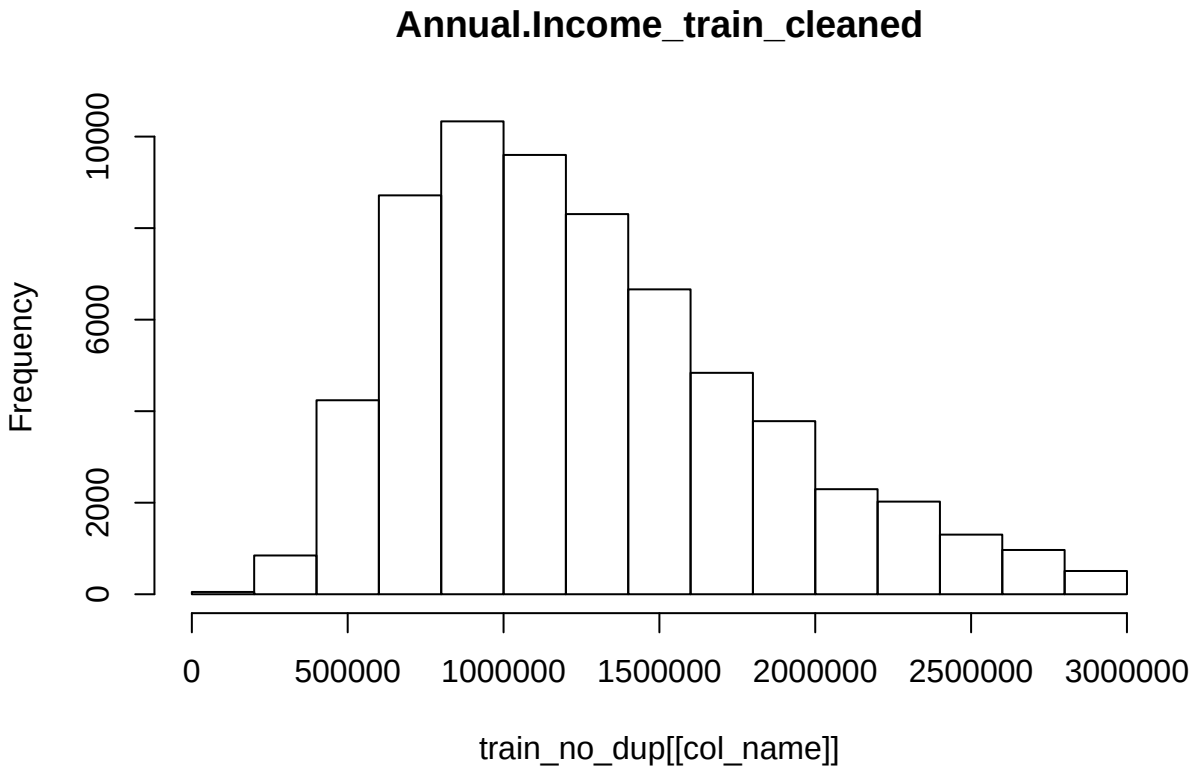

Credit.Score_tran_uncleaned
750
700
650
600

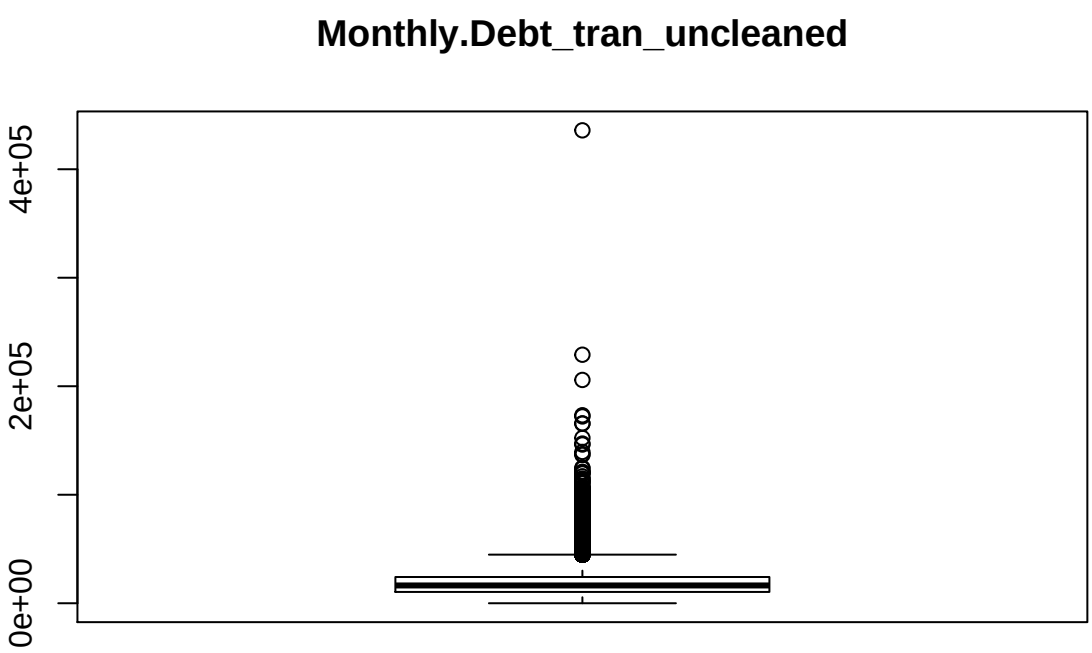

Credit.Score_train_uncleaned
Frequency
0
5000
10000
15000
600
650
700
750
train_no_dup[[col_name]]

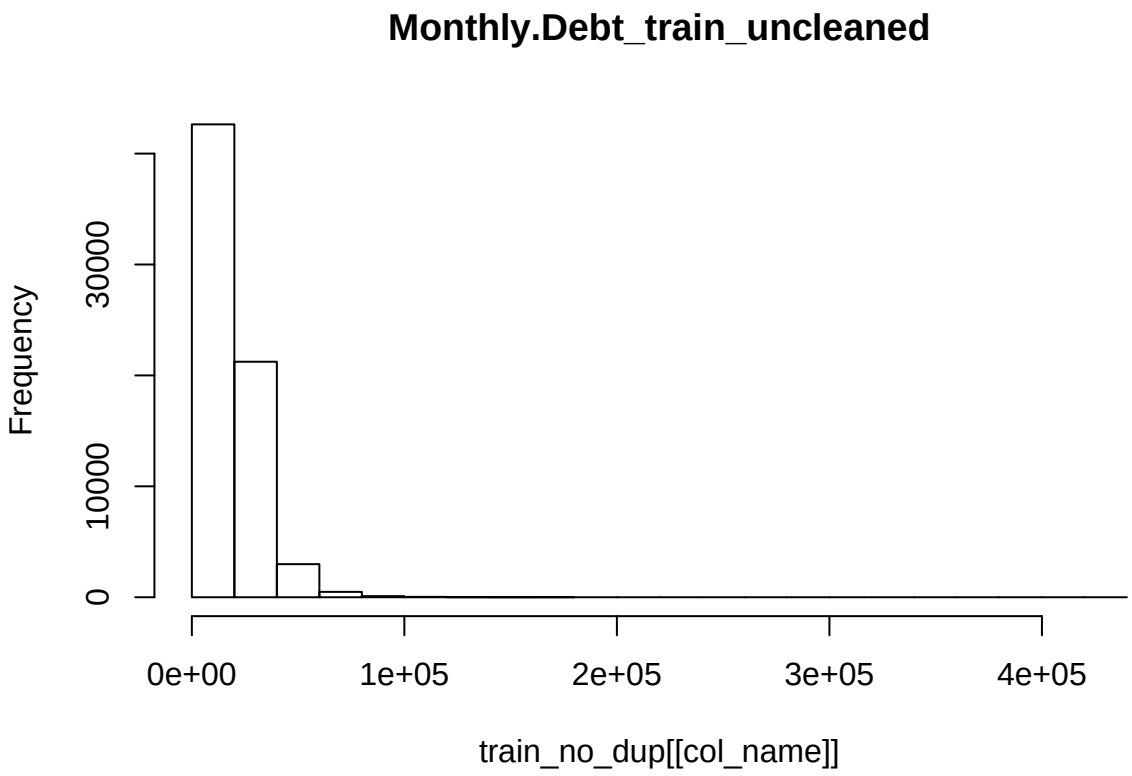

Credit.Score_train_cleaned
Frequency
0
2000
4000
6000
8000
660
680
700
720
740
train_no_dup[[col_name]]

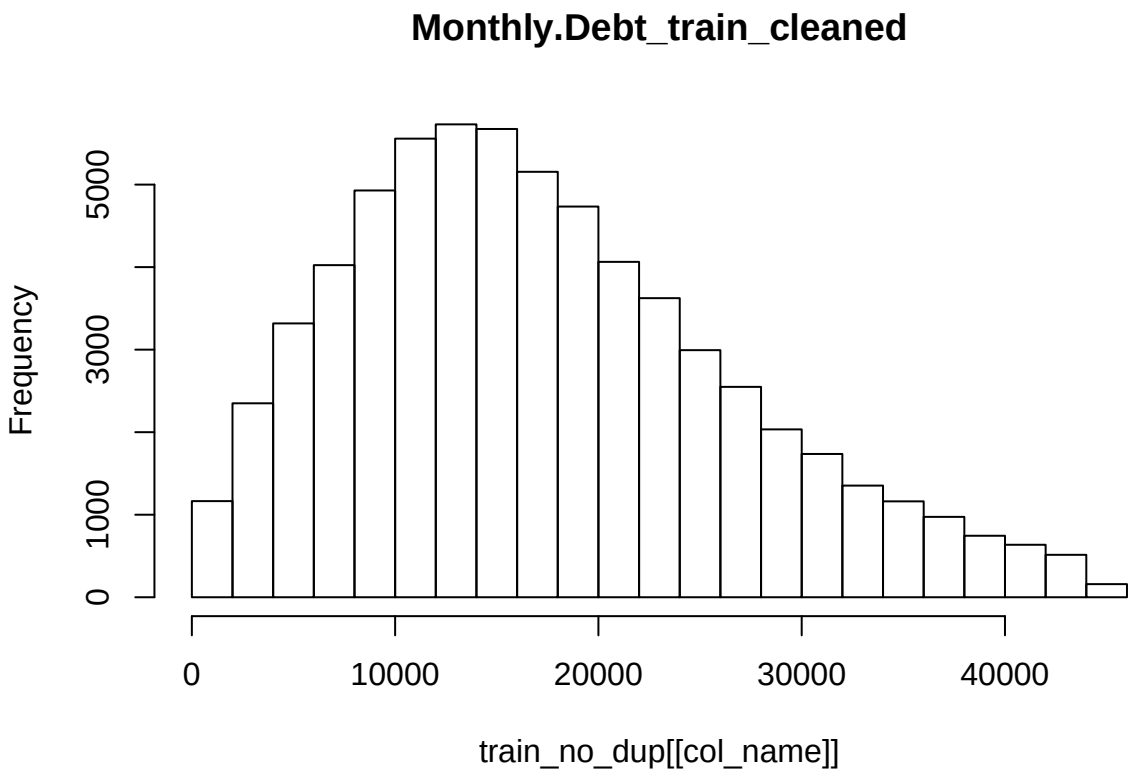

Annual.Income_tran_uncleaned
1.0e+08
0.0e+00

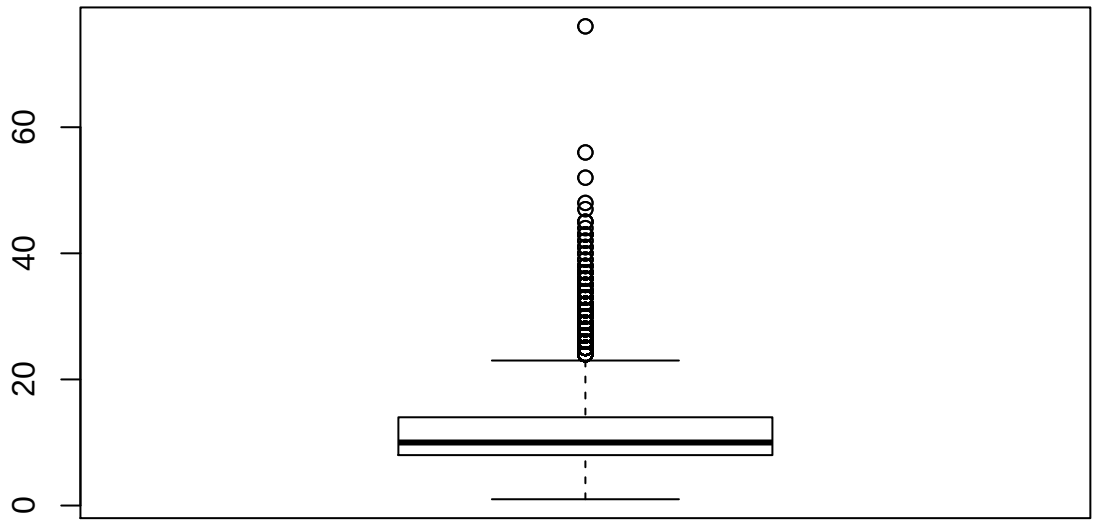

Annual.Income_train_uncleaned
Frequency
0
20000
40000
60000
0.0e+00
5.0e+07
1.0e+08
1.5e+08
train_no_dup[[col_name]]

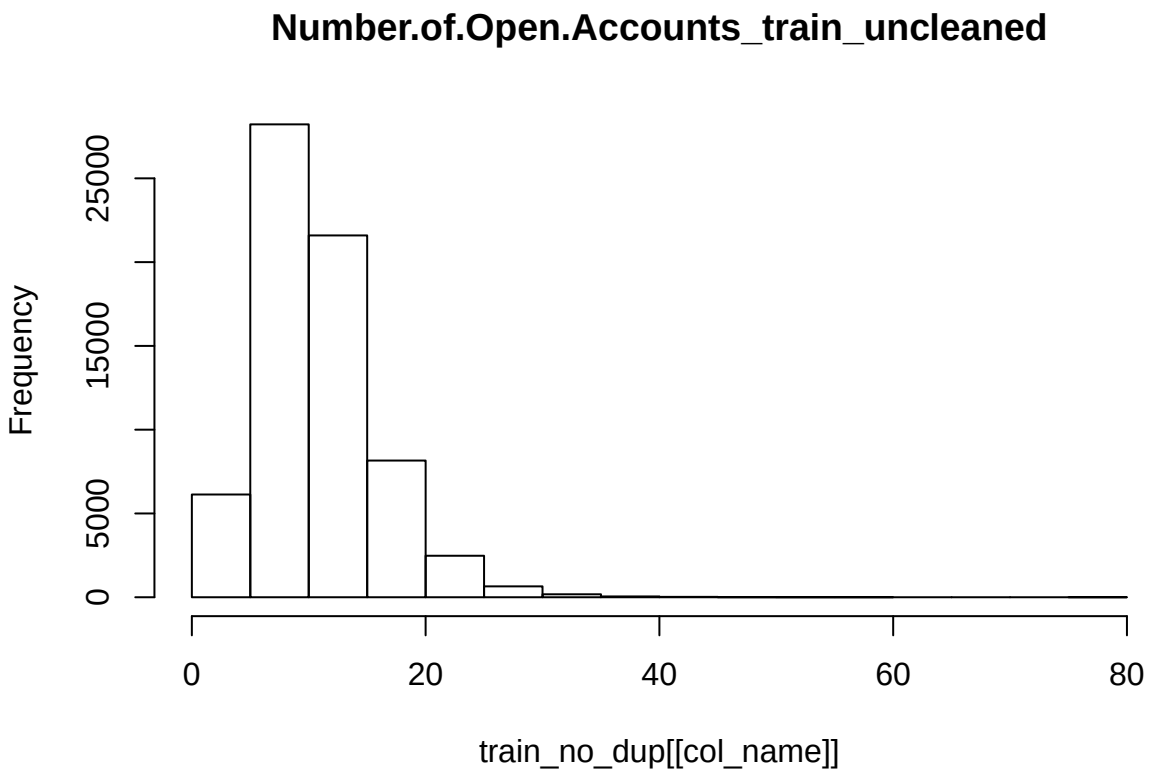

Annual.Income_train_cleaned
Frequency
0
2000
6000
10000
0
500000
1000000
1500000
2000000
2500000
3000000
train_no_dup[[col_name]]

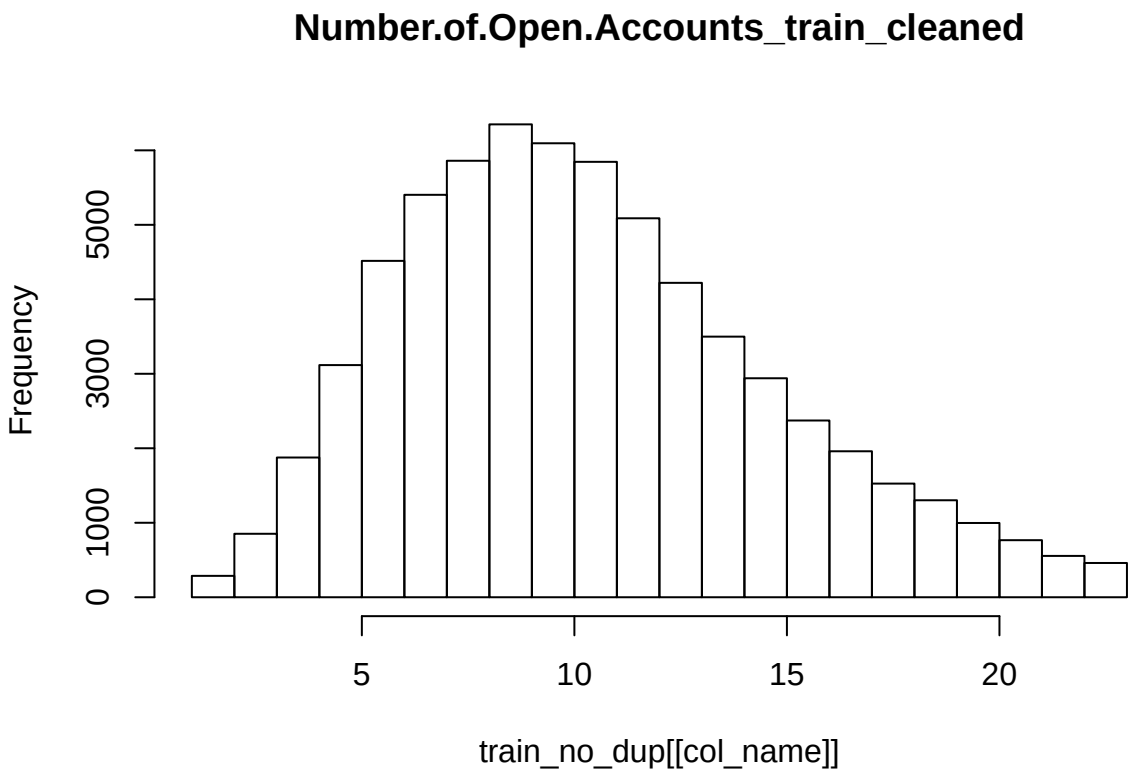

Monthly.Debt_tran_uncleaned
0e+00
2e+05
4e+05

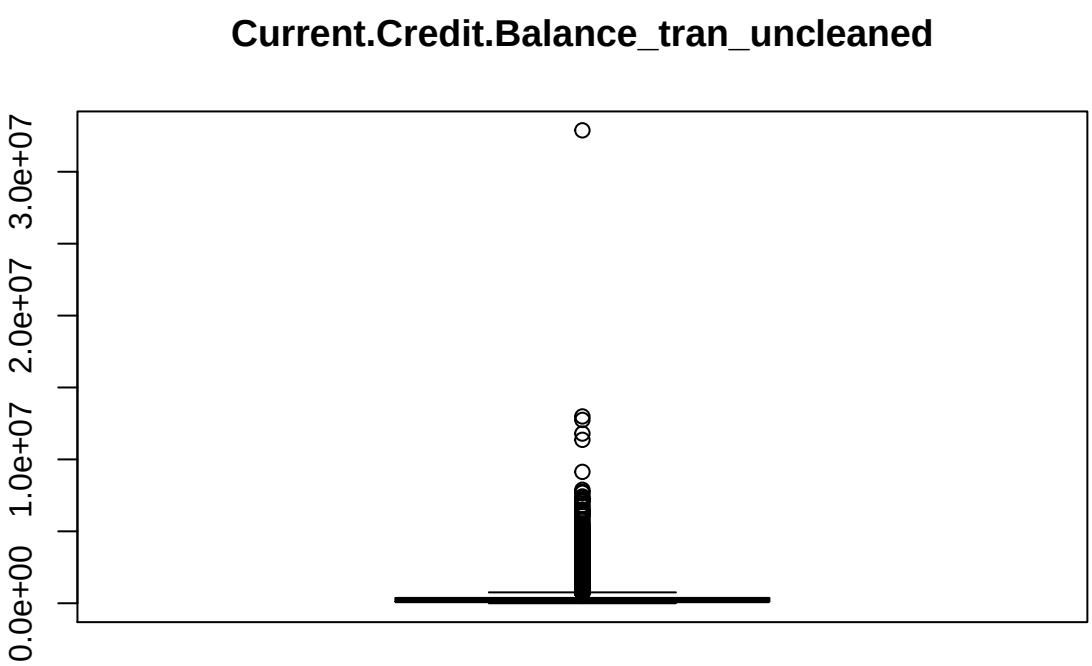

Monthly.Debt_train_uncleaned
Frequency
0
10000
30000
0e+00
1e+05
2e+05
3e+05
4e+05
train_no_dup[[col_name]]

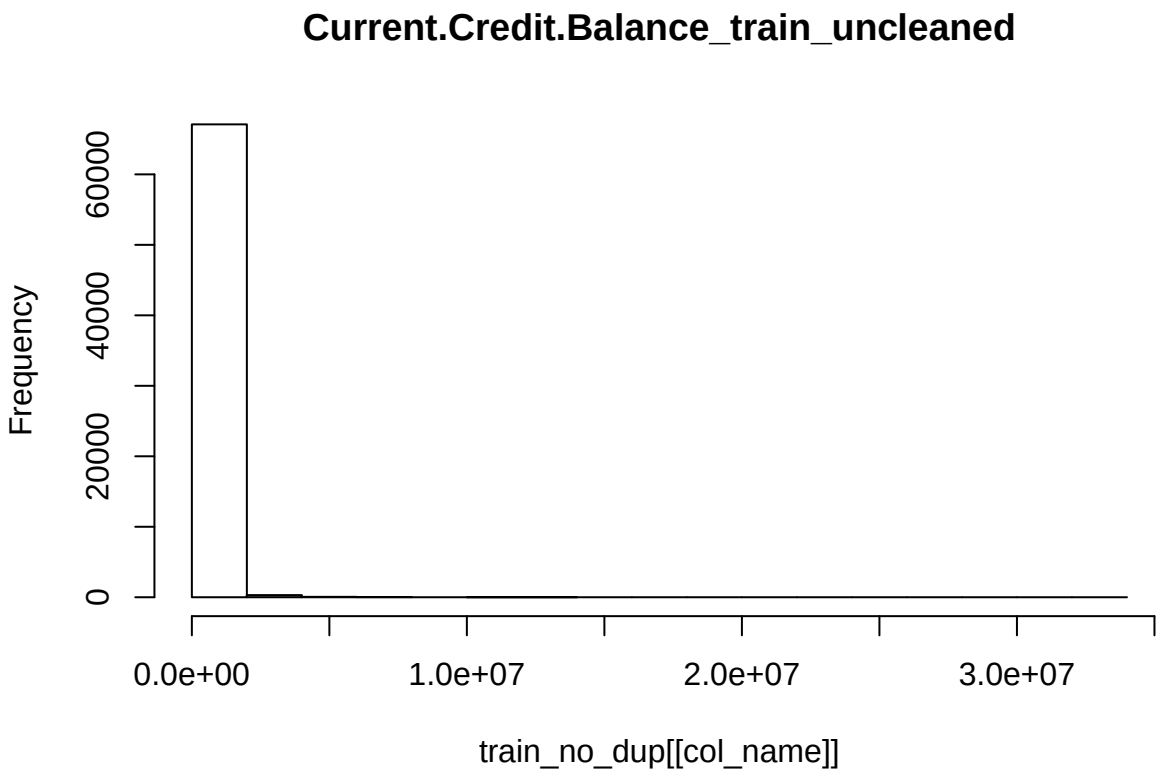

Monthly.Debt_train_cleaned
Frequency
0
1000
3000
5000
0
10000
20000
30000
40000
train_no_dup[[col_name]]


Number.of.Open.Accounts_tran_uncleaned
0
20
40
60

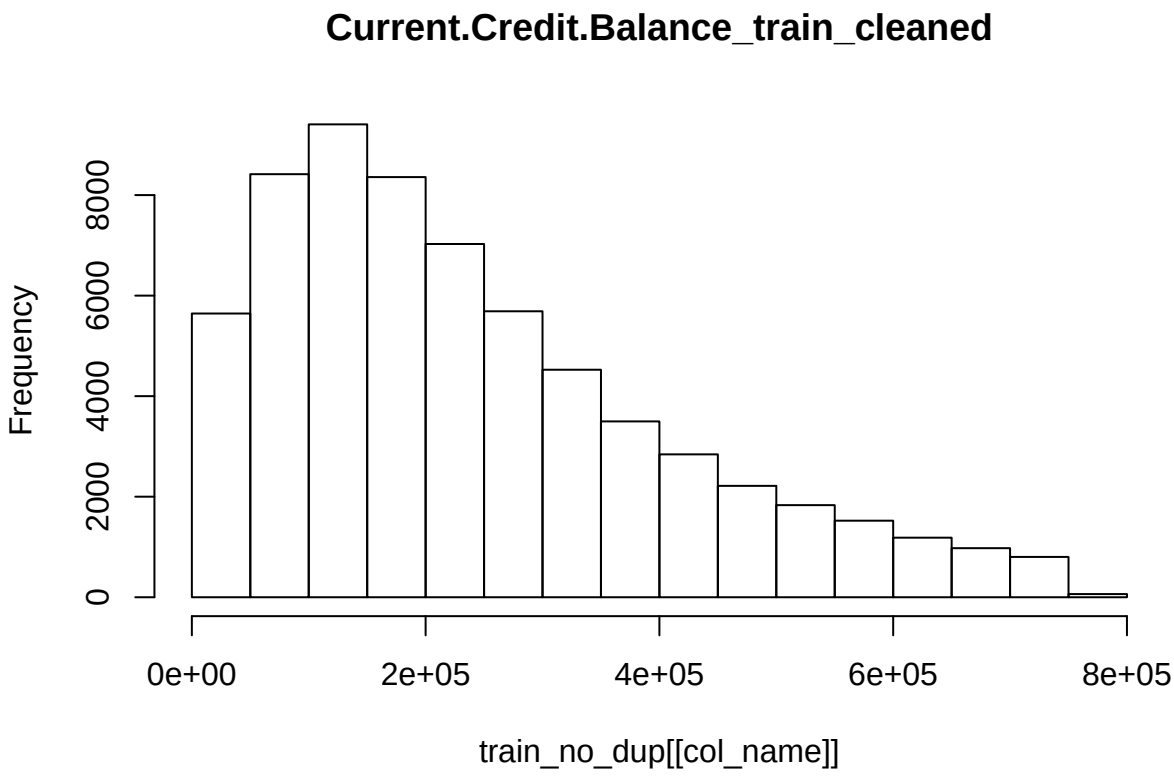

Number.of.Open.Accounts_train_uncleaned
Frequency
0
5000
15000
25000
0
20
40
60
80
train_no_dup[[col_name]]

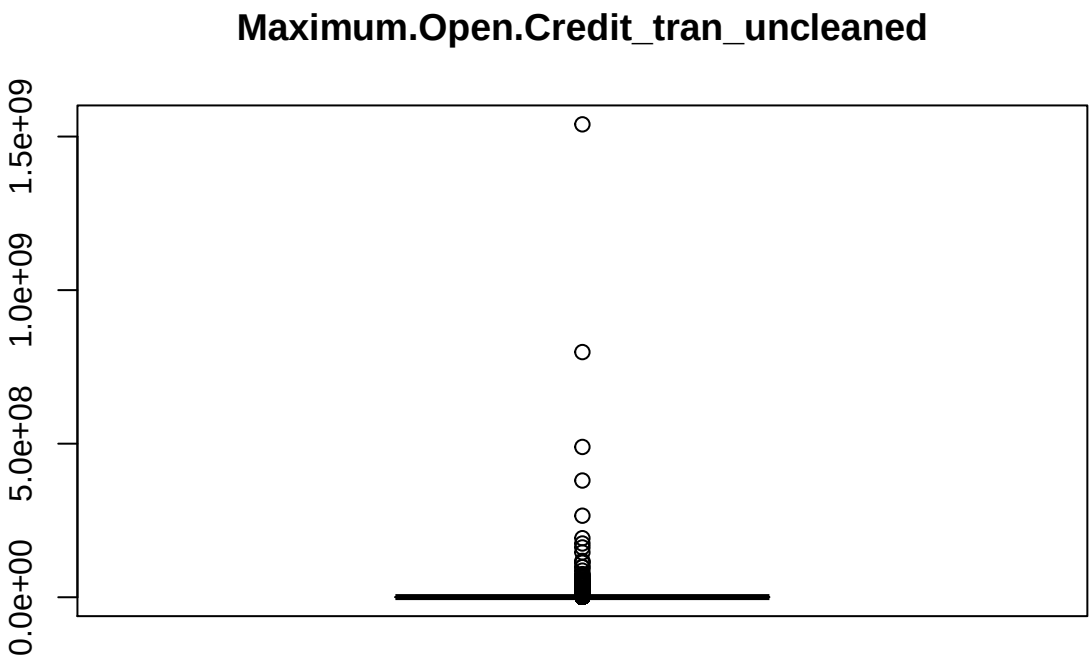

Number.of.Open.Accounts_train_cleaned
Frequency
0
1000
3000
5000
5
10
15
20
train_no_dup[[col_name]]

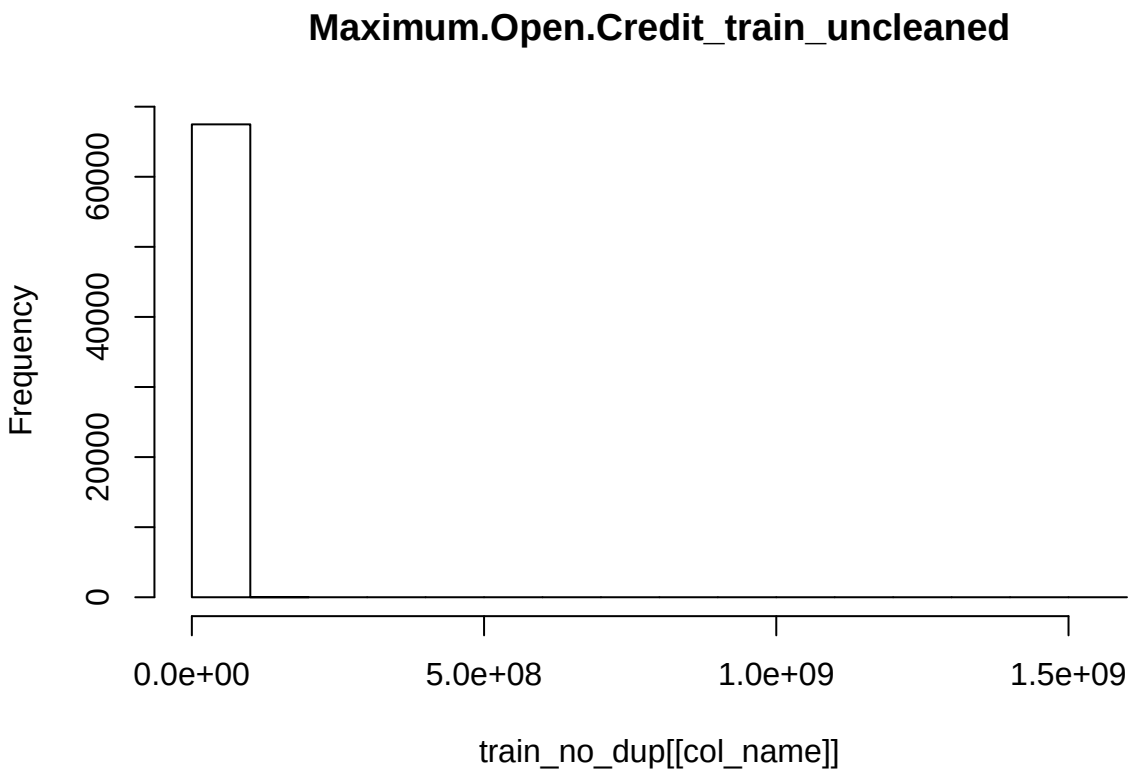

Current.Credit.Balance_tran_uncleaned
0.0e+00
1.0e+07
2.0e+07
3.0e+07




```
## Warning in x[floor(d)] + x[ceiling(d)]: NAs produced by integer overflow
```


Current.Credit.Balance_train_cleaned
Frequency
0
2000
4000
6000
8000
0e+00
2e+05
4e+05
6e+05
8e+05
train_no_dup[[col_name]]


Maximum.Open.Credit_tran_uncleaned
0.0e+00
5.0e+08
1.0e+09
1.5e+09




```
train_no_dup <- train_no_dup[complete.cases(train_no_dup),]
#View(train_no_dup)
train_no_id <- train_no_dup[,-c(1,2)]
str(train_no_id)
```

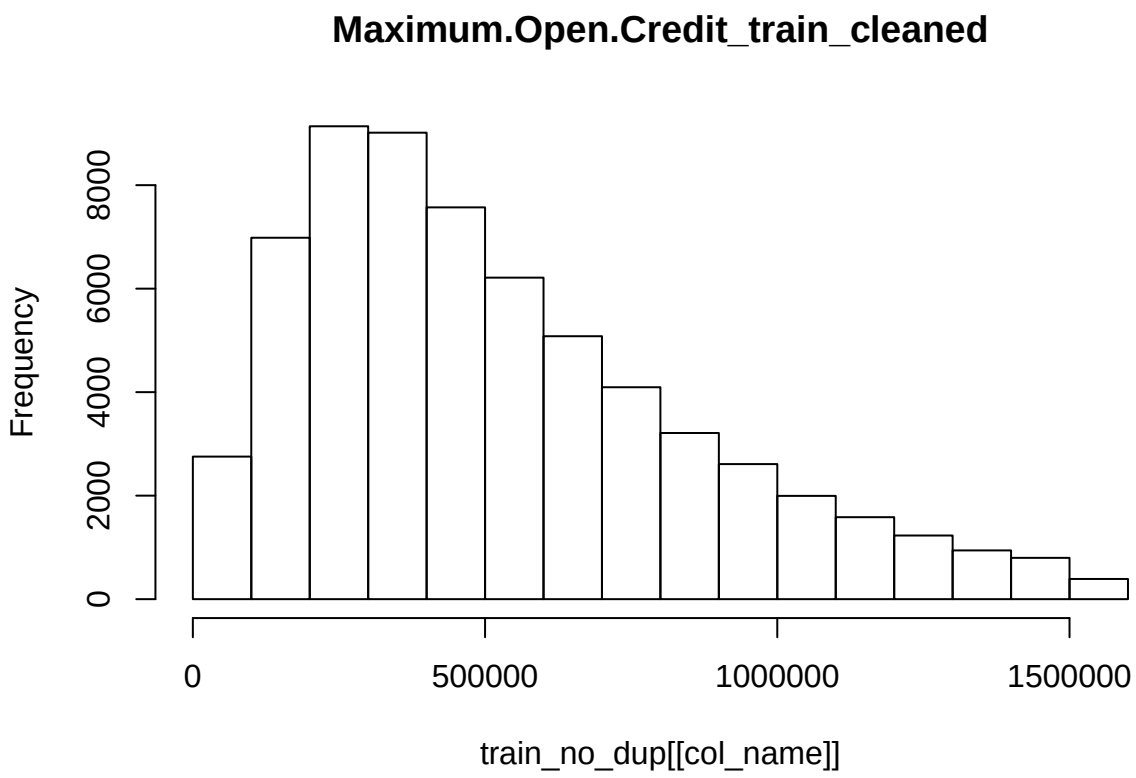
```
## 'data.frame':    46958 obs. of  16 variables:
##  $ Loan.Status               : Factor w/ 2 levels "Charged Off",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Current.Loan.Amount       : int  206602 317108 130174 688468 288948 219692 176198 78012 523908 194
##  $ Term                      : Factor w/ 2 levels "Long Term","Short Term": 2 1 2 1 2 1 2 2 1 2 ...
##  $ Credit.Score              : num  729 687 733 682 712 661 736 738 737 742 ...
##  $ Annual.Income             : int  896857 1133274 524609 1494616 537472 527839 1902090 728726 102877
##  $ Years.in.current.job      : Factor w/ 11 levels "< 1 year","1 year",..: 3 10 1 1 3 3 3 6 8 2 ...
##  $ Home.Ownership            : Factor w/ 4 levels "HaveMortgage",..: 2 4 4 4 4 4 2 4 2 4 ...
##  $ Purpose                   : Factor w/ 16 levels "Business Loan",..: 4 4 4 4 4 4 4 7 4 4 ...
##  $ Monthly.Debt              : num  16368 9633 9312 14697 5778 ...
##  $ Years.of.Credit.History   : num  17.3 17.4 15.4 16.6 14.8 17 15.4 11.4 19.3 27.4 ...
##  $ Number.of.Open.Accounts   : int  6 4 7 8 4 9 9 8 5 13 ...
##  $ Number.of.Credit.Problems: int  0 0 1 0 0 0 0 0 0 1 ...
##  $ Current.Credit.Balance    : int  215308 60287 130701 343995 132468 254277 206872 104633 474658 176
##  $ Maximum.Open.Credit       : int  272448 126940 268818 843854 164406 379918 620554 199936 742720 33
##  $ Bankruptcies              : int  0 0 1 0 0 0 0 0 0 1 ...
##  $ Tax.Liens                 : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
#View(train_no_id)
#summary(train_no_id)
cat_col_train <- colnames(train_no_id)[c(3,6,7,8)]
```

```
library(caret)
```

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

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




```
library(dplyr)
```

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

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

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

```
#creating dummy data for categorucal variables
dmy_train <- dummyVars(" ~ .", data = train_no_id[,c(3,6,7,8)])
#View(dmy_train)
trsf_train <- data.frame(predict(dmy_train, newdata = train_no_id[,c(3,6,7,8)]))
#View(trsf)

#One hot encoded data
train_one_hot <- cbind(train_no_id[,-c(3,6,7,8)],trsf_train)
#View(train_one_hot)
```

```r
#summary(train_one_hot)


##### Exploratory Data Analysis #####

# Current Loan Amount bins
summary(train_no_id$Current.Loan.Amount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15422  161282  258896  288650  391644  789184
```

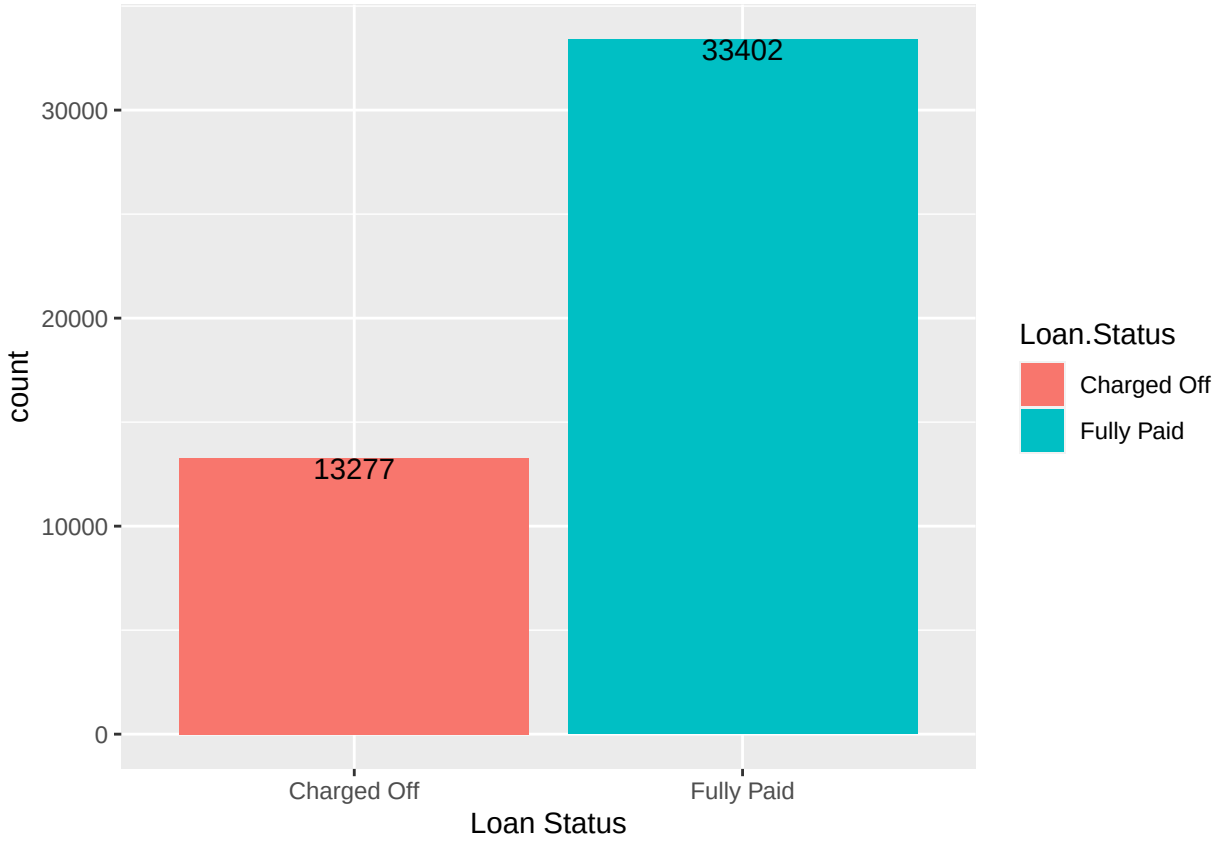
```r
train_no_id$Current.Loan.Amount.bins <- cut(train_no_id$Current.Loan.Amount,
                                            c(15000,165000,315000,465000,
                                              615000, 789184))
# Credit Score bins
summary(train_no_id$Credit.Score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   655.0   707.0   725.0   720.6   739.0   751.0
```

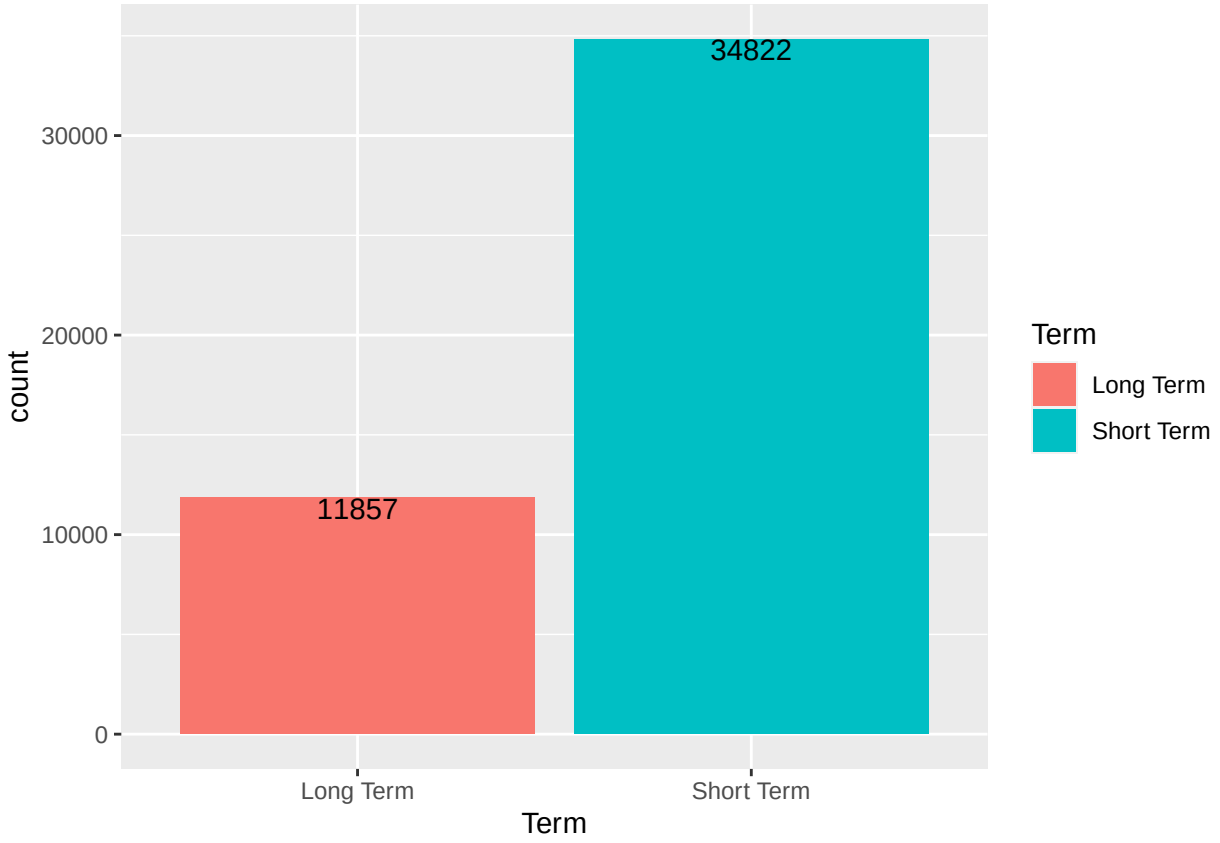
```r
train_no_id$Credit.Score.bins <- cut(train_no_id$Credit.Score,
                                     c(585, 610, 635, 660, 685,
                                       710, 735, 760))

# Annual Income Bins
train_no_id$Annual.Income.bins <- cut(train_no_id$Annual.Income, c(75000,375000,675000,975000,
                                                                   1275000, 1575000, 1875000, 2893662))

# Current credit balance bins
summary(train_no_id$Current.Credit.Balance)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0  106476  192033  228855  317528  755003
```

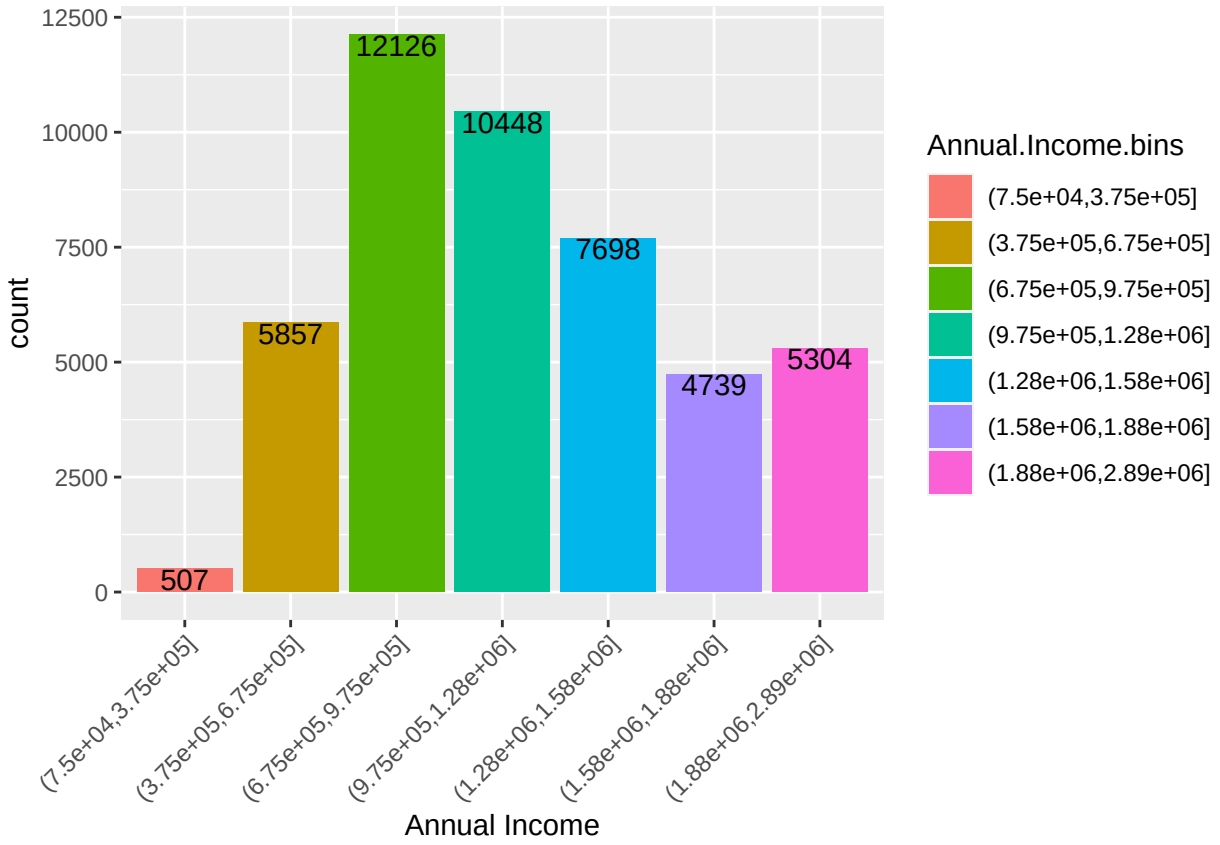
```r
train_no_id$Current.Credit.Balance.bins <- cut(train_no_id$Current.Credit.Balance,
                                               c(0,100000,200000,300000, 400000,
                                                 500000, 600000, 700000, 800000))

# Remove any NAs after binning
train_no_id <- train_no_id[complete.cases(train_no_id),]

# Subset of people whose loan got charged off
subset_charged_off <- train_no_id[train_no_id$Loan.Status == 'Charged Off',]

### Basic Analysis ###

library(ggplot2)

# Counts per Loan status
ggplot(train_no_id, aes(x=Loan.Status, y=..count..)) +
  geom_bar(aes(fill=Loan.Status)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Loan Status')
```



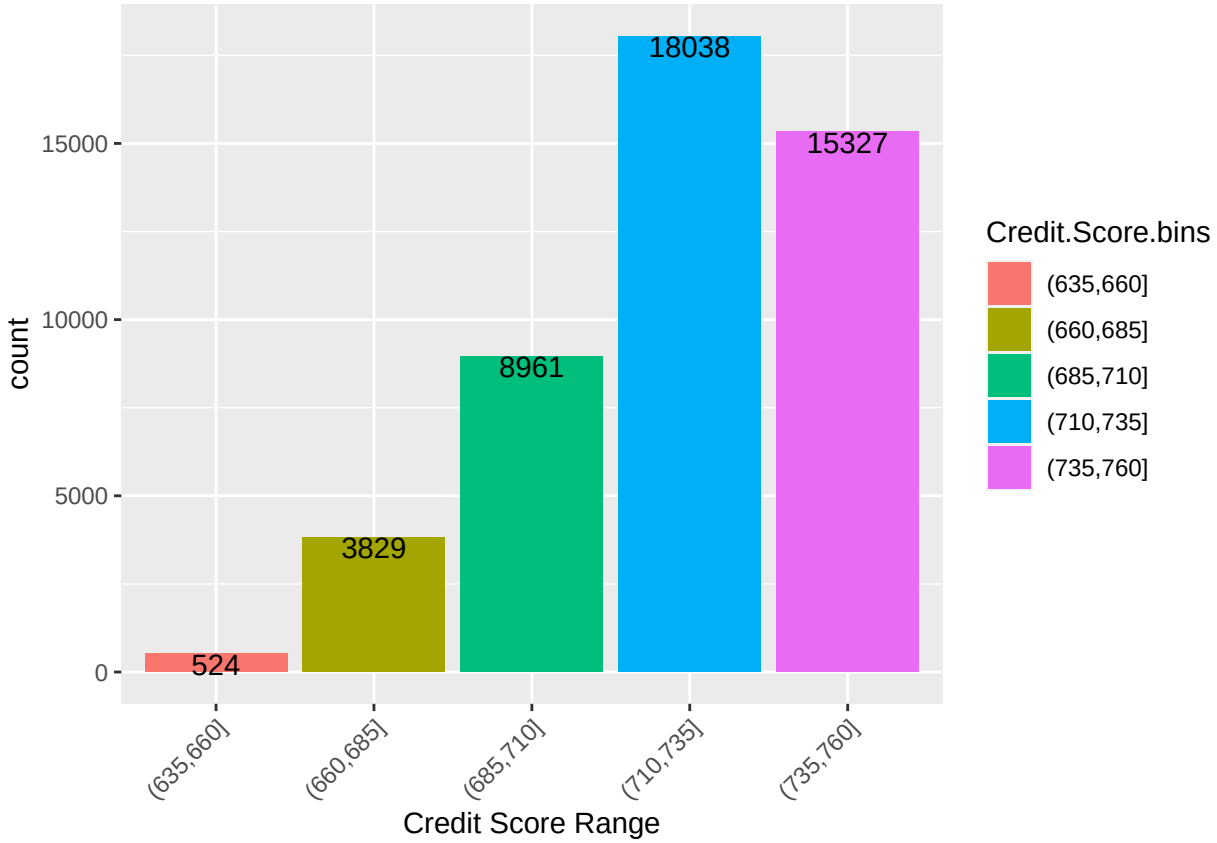

```
# Counts per Term
ggplot(train_no_id, aes(x=Term, y=..count..)) +
  geom_bar(aes(fill=Term)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Term')
```



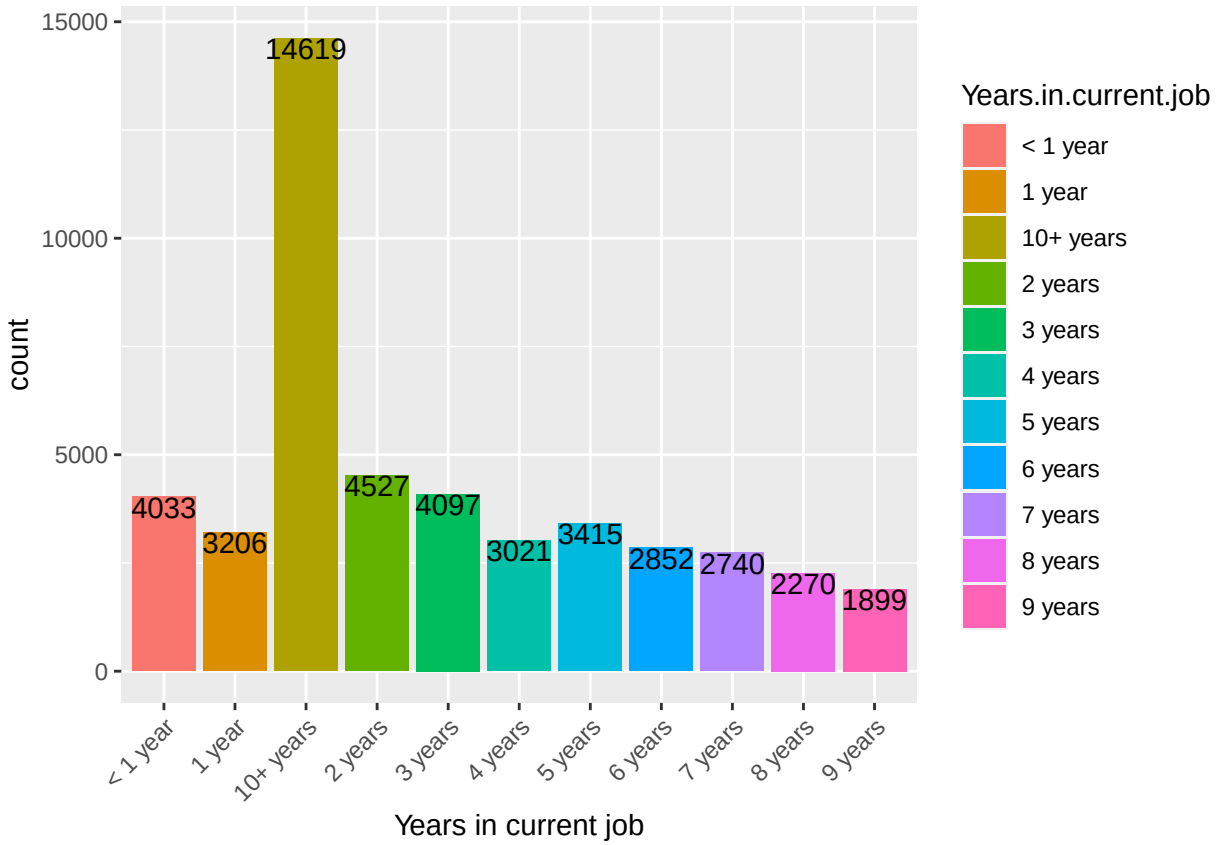

```
# Counts per Annual Income range
ggplot(train_no_id, aes(x=Annual.Income.bins, y=..count..)) +
  geom_bar(aes(fill=Annual.Income.bins)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Annual Income')+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



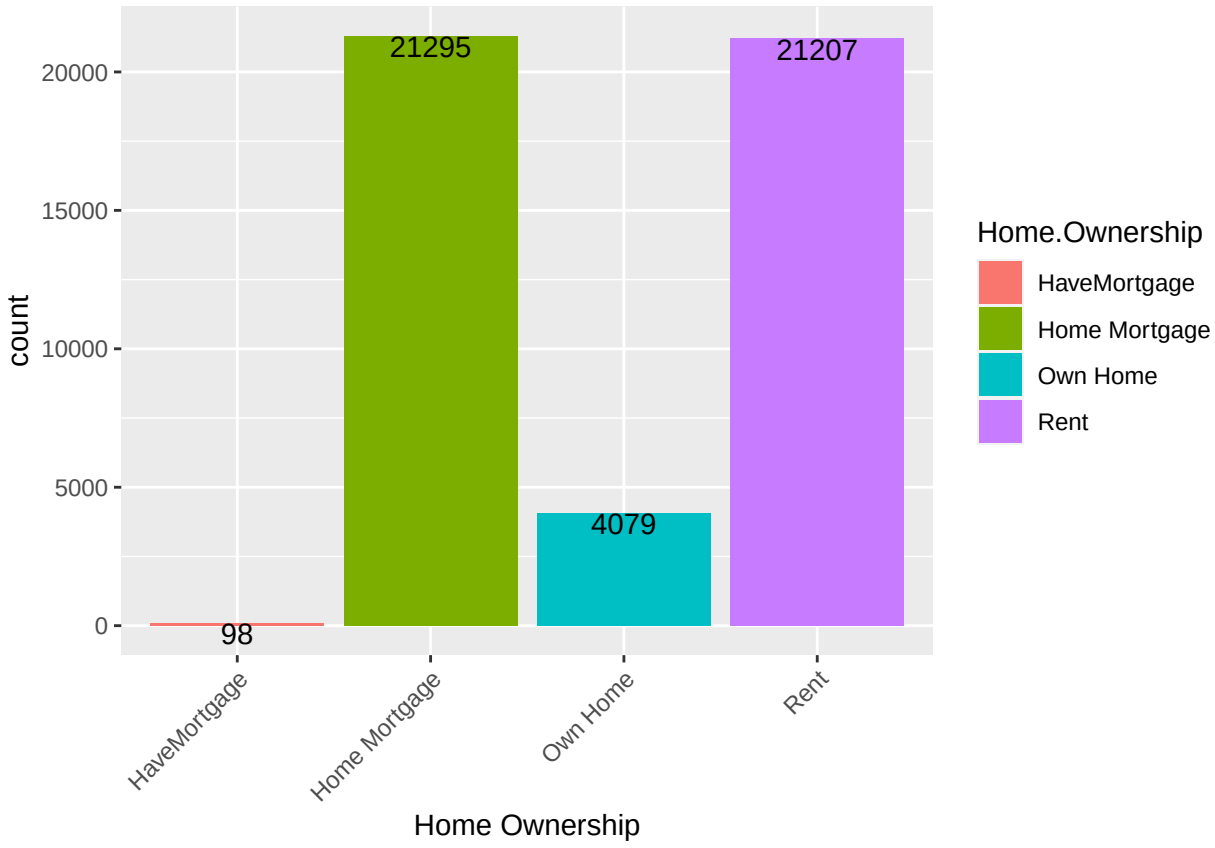

```
# Counts per Credit Score range
# Obvious observation: The higher the credit score,
# more the chances of getting a loan
ggplot(train_no_id, aes(x=Credit.Score.bins, y=..count..)) +
  geom_bar(aes(fill=Credit.Score.bins)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Credit Score Range')+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



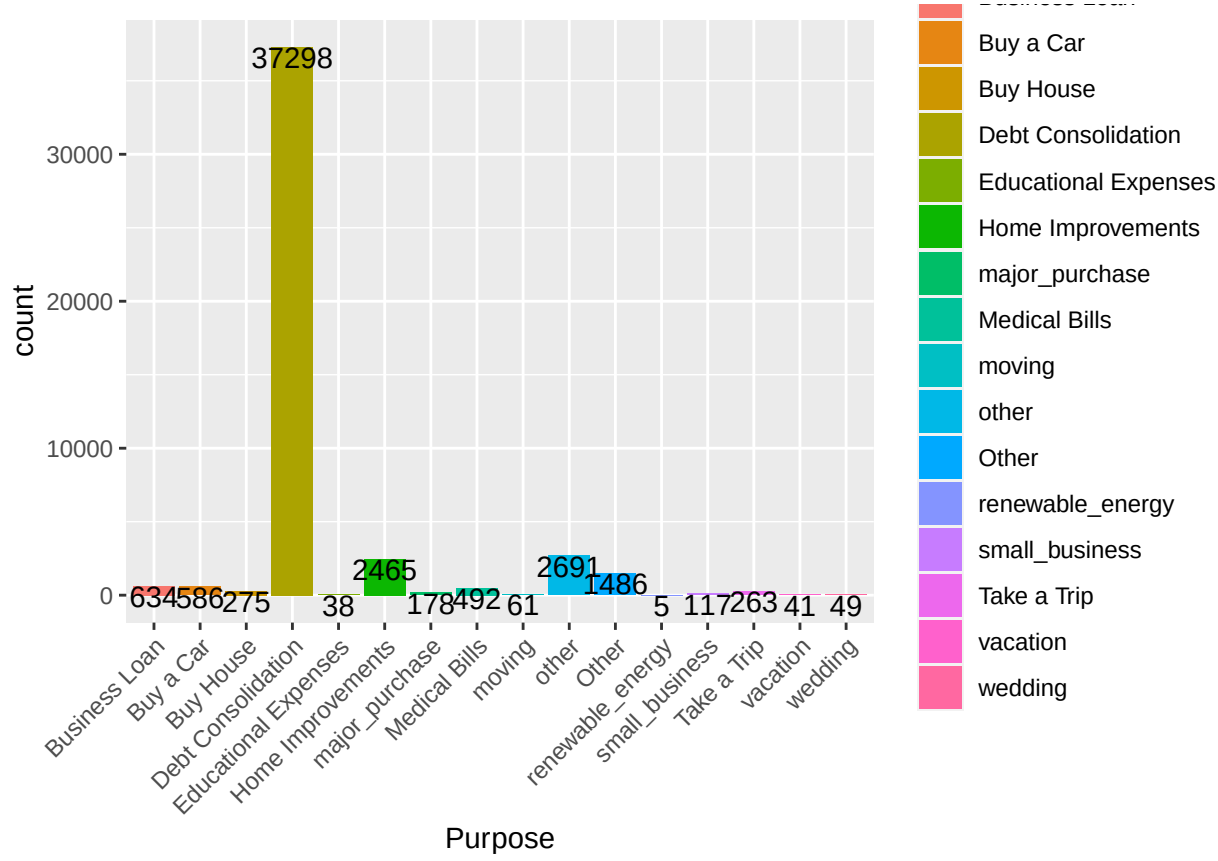

```
# Counts per Years in current job
ggplot(train_no_id, aes(x=Years.in.current.job, y=..count..)) +
  geom_bar(aes(fill=Years.in.current.job)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Years in current job')+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



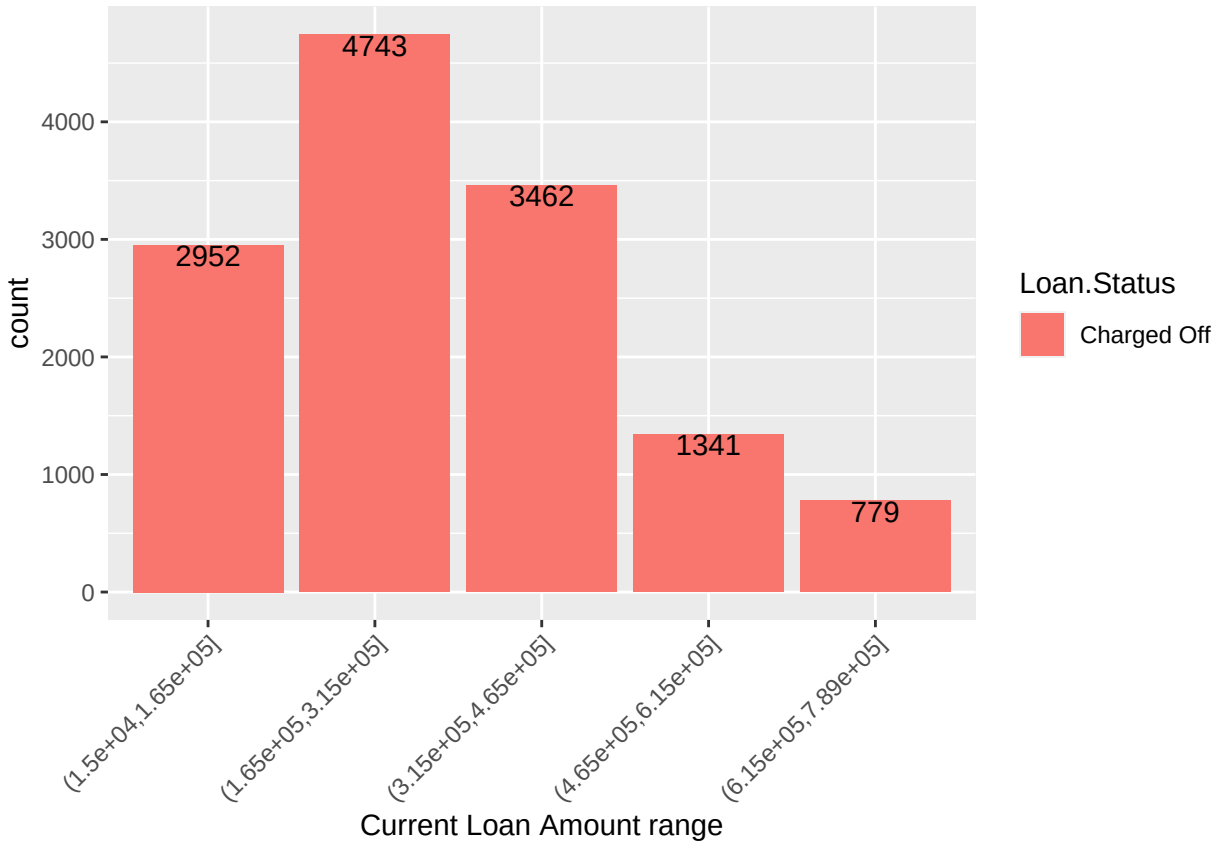

```
# Counts per Home ownership
ggplot(train_no_id, aes(x=Home.Ownership, y=..count..)) +
  geom_bar(aes(fill=Home.Ownership)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Home Ownership')+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



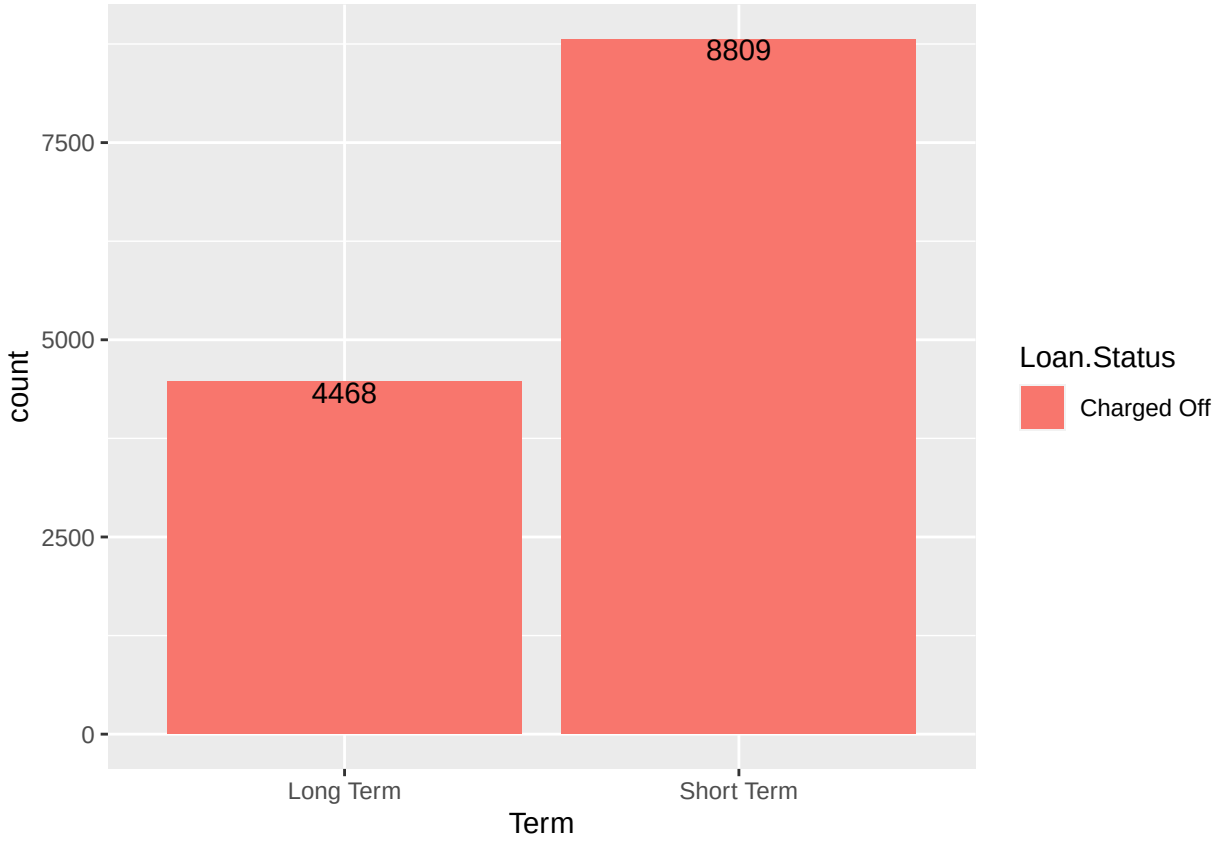

```
# Counts per purpose
ggplot(train_no_id, aes(x=Purpose, y=..count..)) +
  geom_bar(aes(fill=Purpose)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Purpose')+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



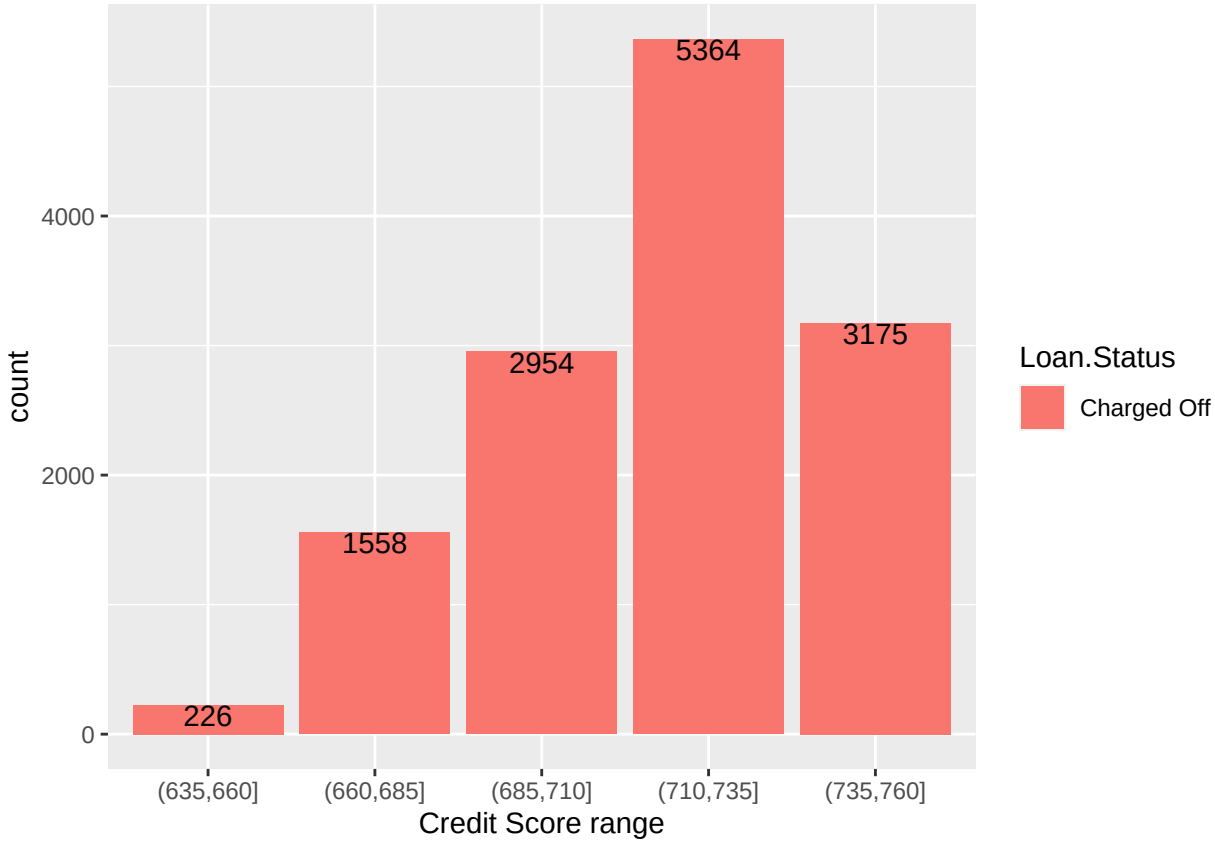

```
### Current Loan Amount

## Current loan amount Range vs count for loan defaulters
ggplot(subset_charged_off, aes(x=Current.Loan.Amount.bins, y=..count..)) +
  geom_bar(aes(fill=Loan.Status)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Current Loan Amount range')+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



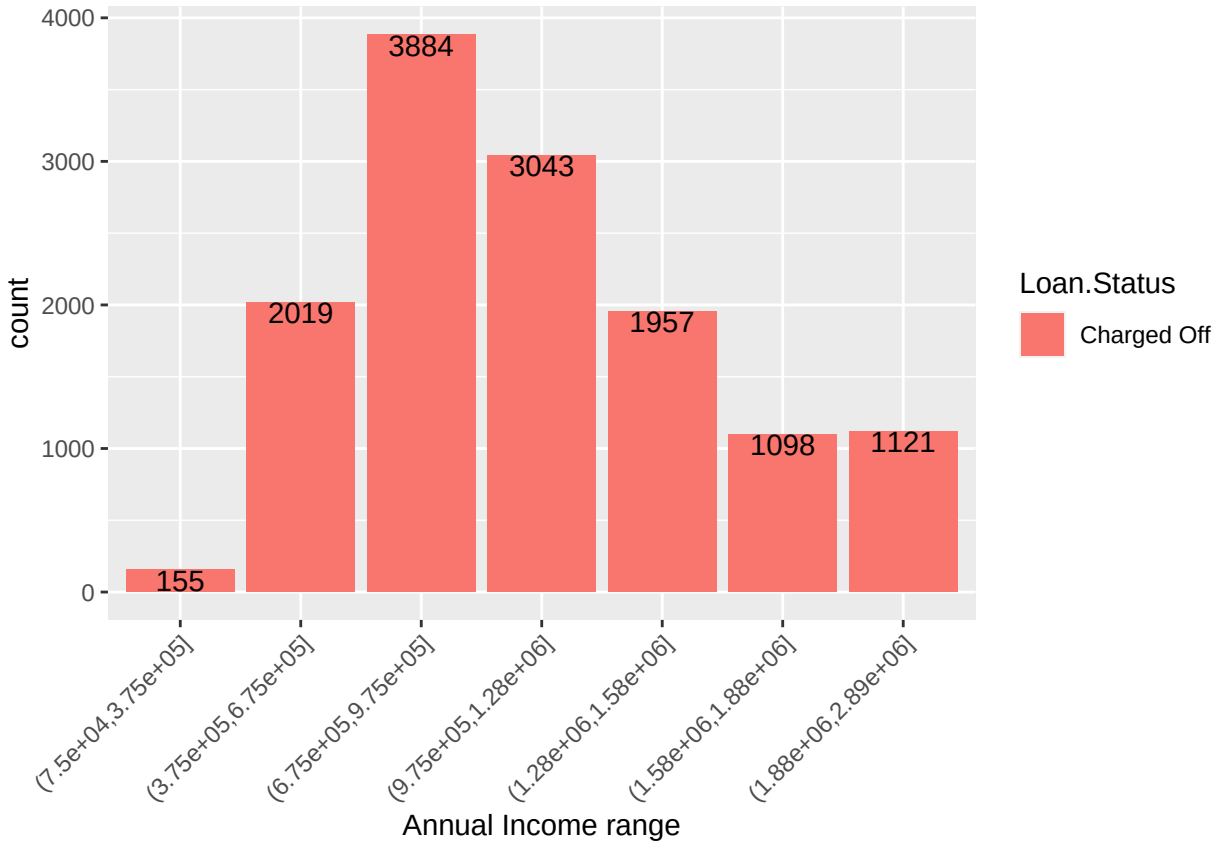

```
# ~63%(~8200) loan defaulters are in current-loan-amount range of 165k - 465k
# Higher chance of loan default in this range


### Term Analysis
ggplot(subset_charged_off, aes(x=Term, y=..count..))+
  geom_bar(aes(fill=Loan.Status))+
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Term')
```




```
# This plot denotes that short term loan borrowers are more prone to default loan
# then long term loan borrowers

### Credit Score

## Credit Score Range vs count for loan defaulters
ggplot(subset_charged_off, aes(x=Credit.Score.bins, y=..count..)) +
  geom_bar(aes(fill=Loan.Status)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Credit Score range')
```



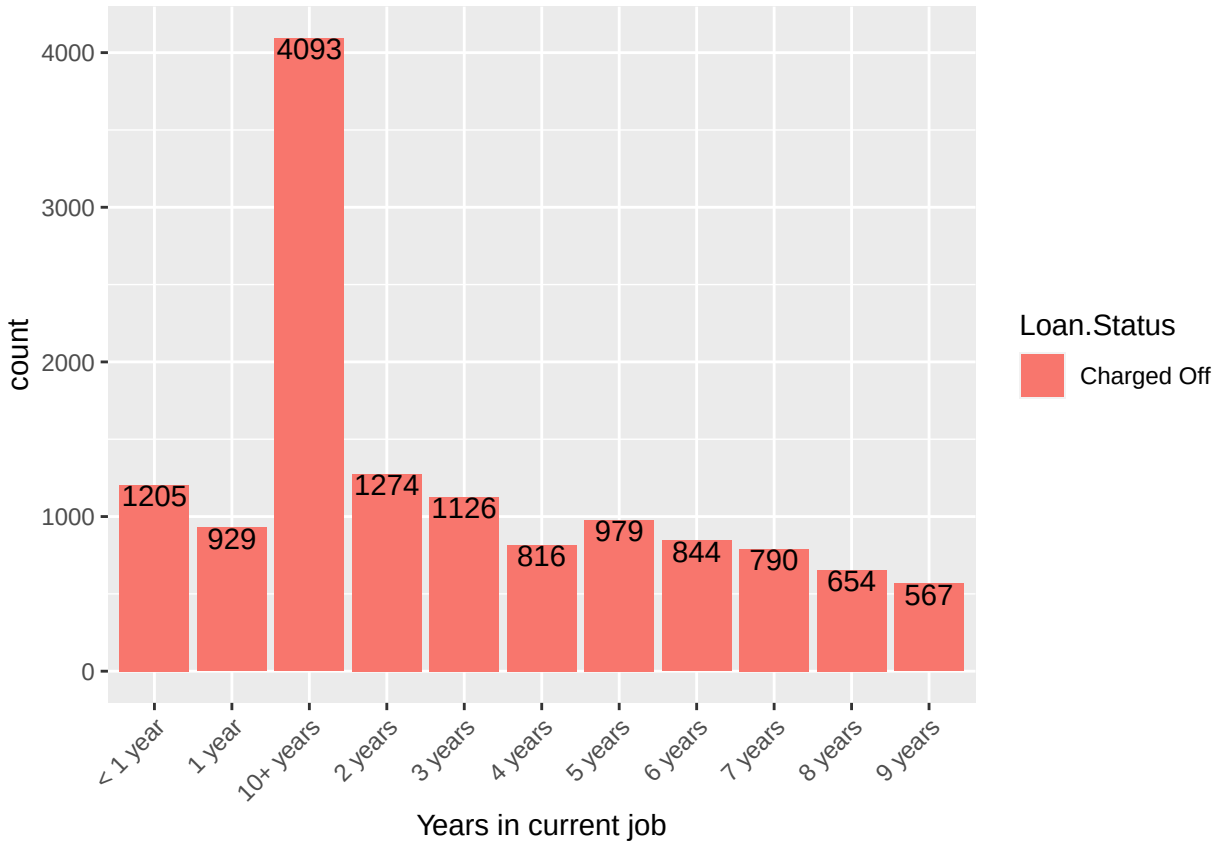

```
# Surprisingly enough ~40% (5391) loan defaulters have credit score
# between 710-735

### Annual Income analysis

## Annual Income Range vs count for loan defaulters
ggplot(subset_charged_off, aes(x=Annual.Income.bins, y=..count..)) +
  geom_bar(aes(fill=Loan.Status)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Annual Income range')+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



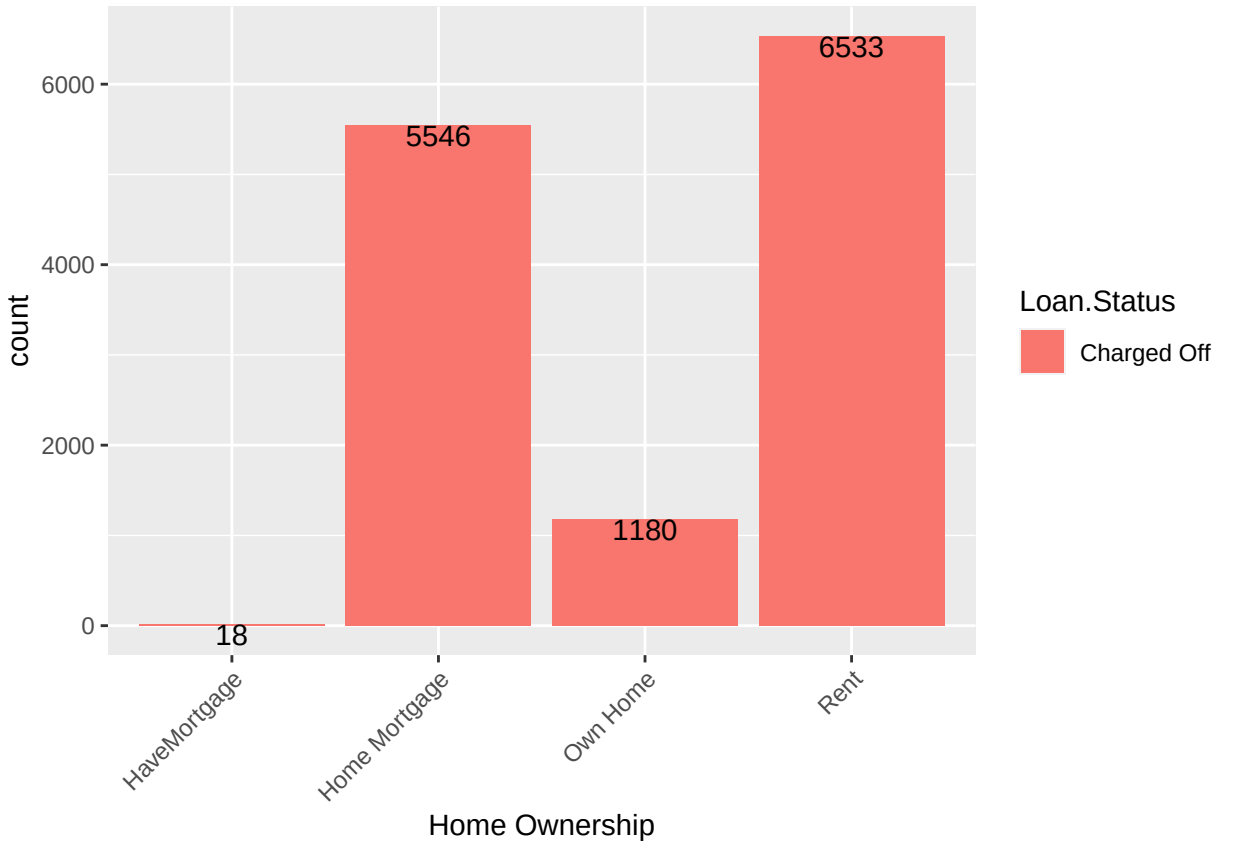

```
# As we can see, there ~4000 defaulters whose income is between 675k - 975k
# ~3000 defaulters have income between 975k - 1275k
# Hence, almost ~54% (~7000) loan defaulters come in income range of 675k - 1275k

income_lower <- as.numeric(quantile(subset_charged_off$Annual.Income)[2])
income_upper <- as.numeric(quantile(subset_charged_off$Annual.Income)[4])
print(paste0("Income of most of the people, who defaulted, lies between: ",
             income_lower, "-", round(income_upper)))
```

```
## [1] "Income of most of the people, who defaulted, lies between: 766688-1390154"
```

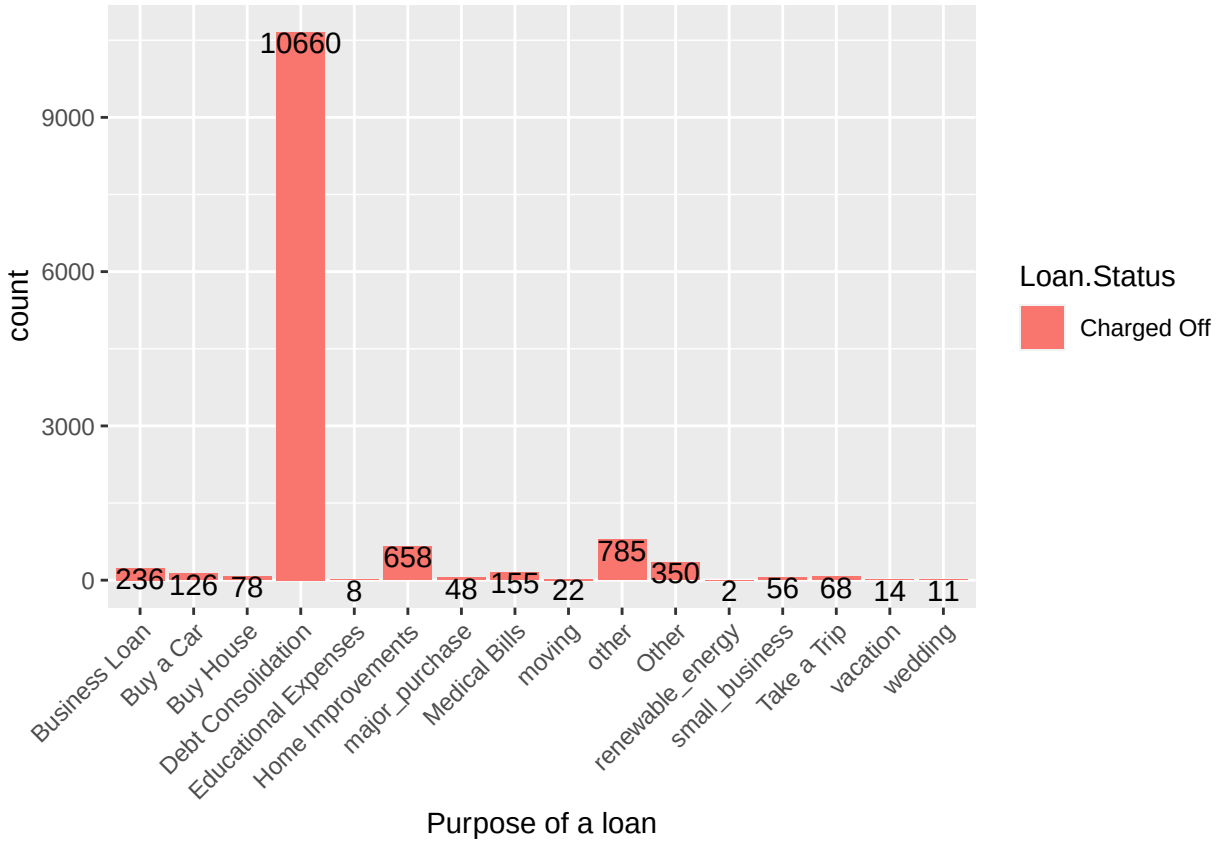
```
## Number of Years in current job
ggplot(subset_charged_off, aes(x=Years.in.current.job, y=..count..)) +
  geom_bar(aes(fill=Loan.Status)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Years in current job')+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



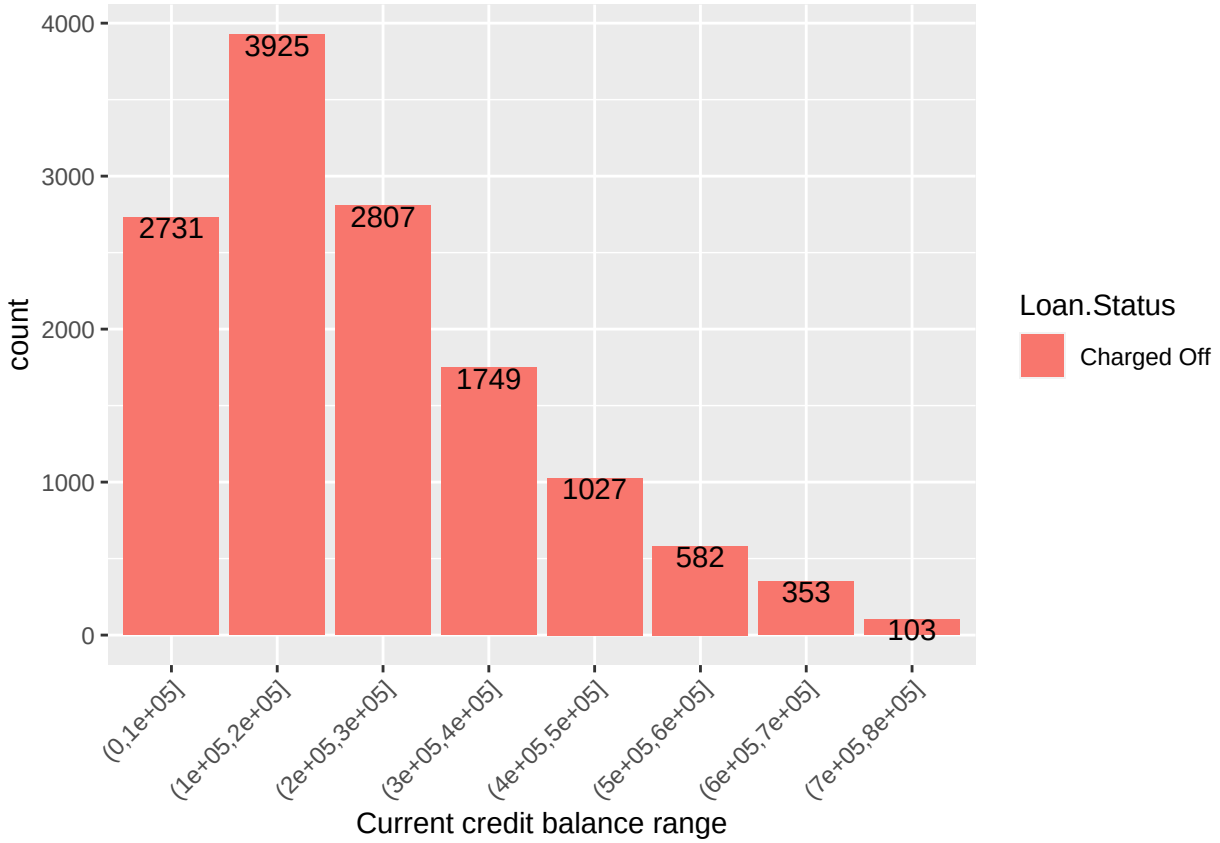

```
## Home Ownership
ggplot(subset_charged_off, aes(x=Home.Ownership, y=..count..)) +
  geom_bar(aes(fill=Loan.Status)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Home Ownership')+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



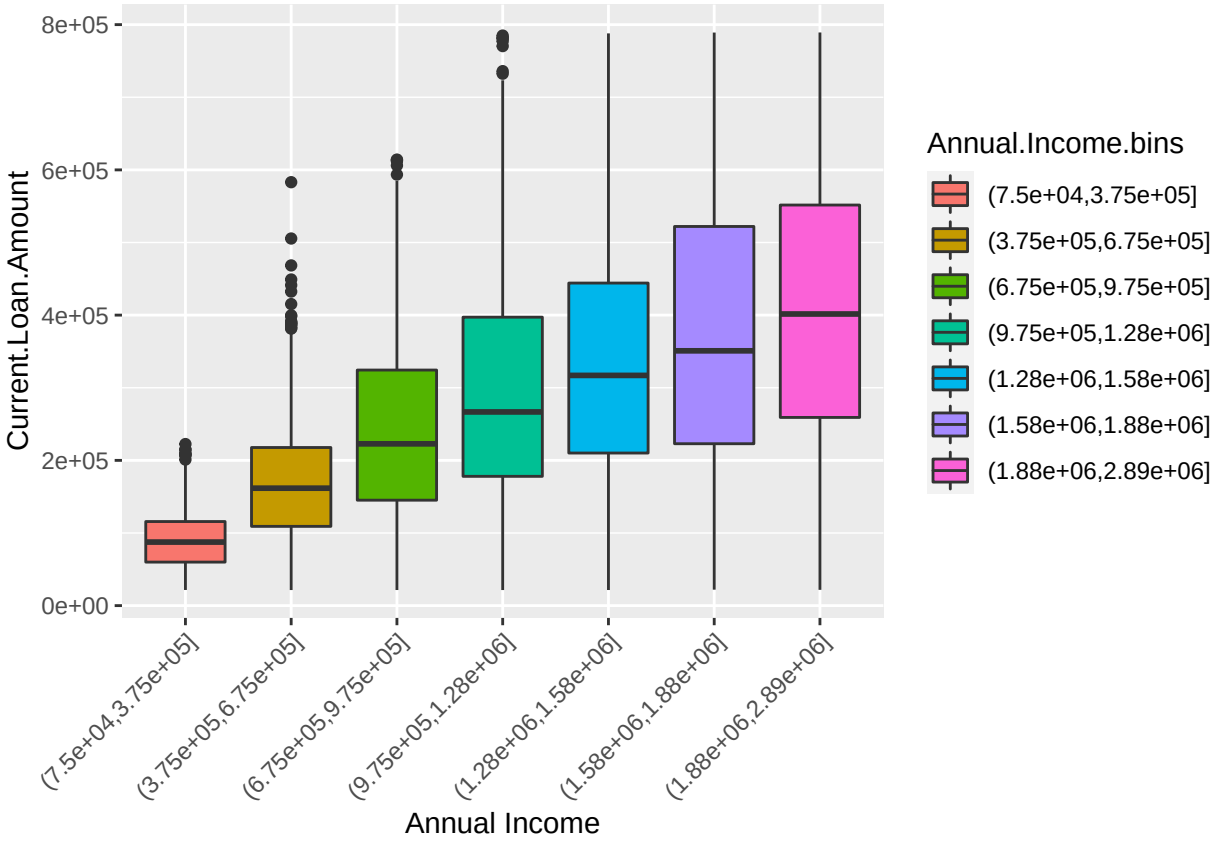

```
## Purpose
ggplot(subset_charged_off, aes(x=Purpose, y=..count..)) +
  geom_bar(aes(fill=Loan.Status)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Purpose of a loan') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



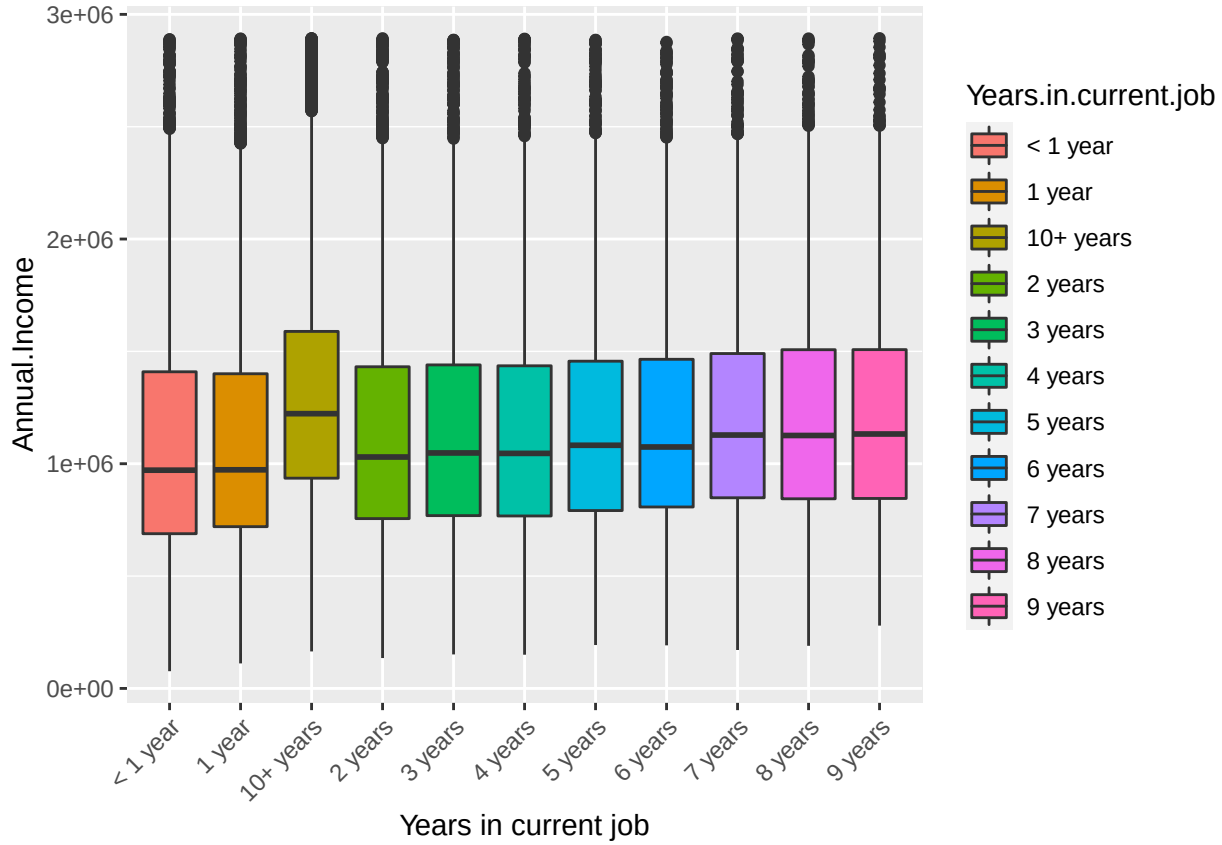

```
# most of the loan defaulters have "debt consolidation" as common purpose

## Current credit balance

ggplot(subset_charged_off, aes(x=Current.Credit.Balance.bins, y=..count..)) +
  geom_bar(aes(fill=Loan.Status)) +
  geom_text(stat='count', aes(label=..count..), vjust=1)+
  xlab('Current credit balance range') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



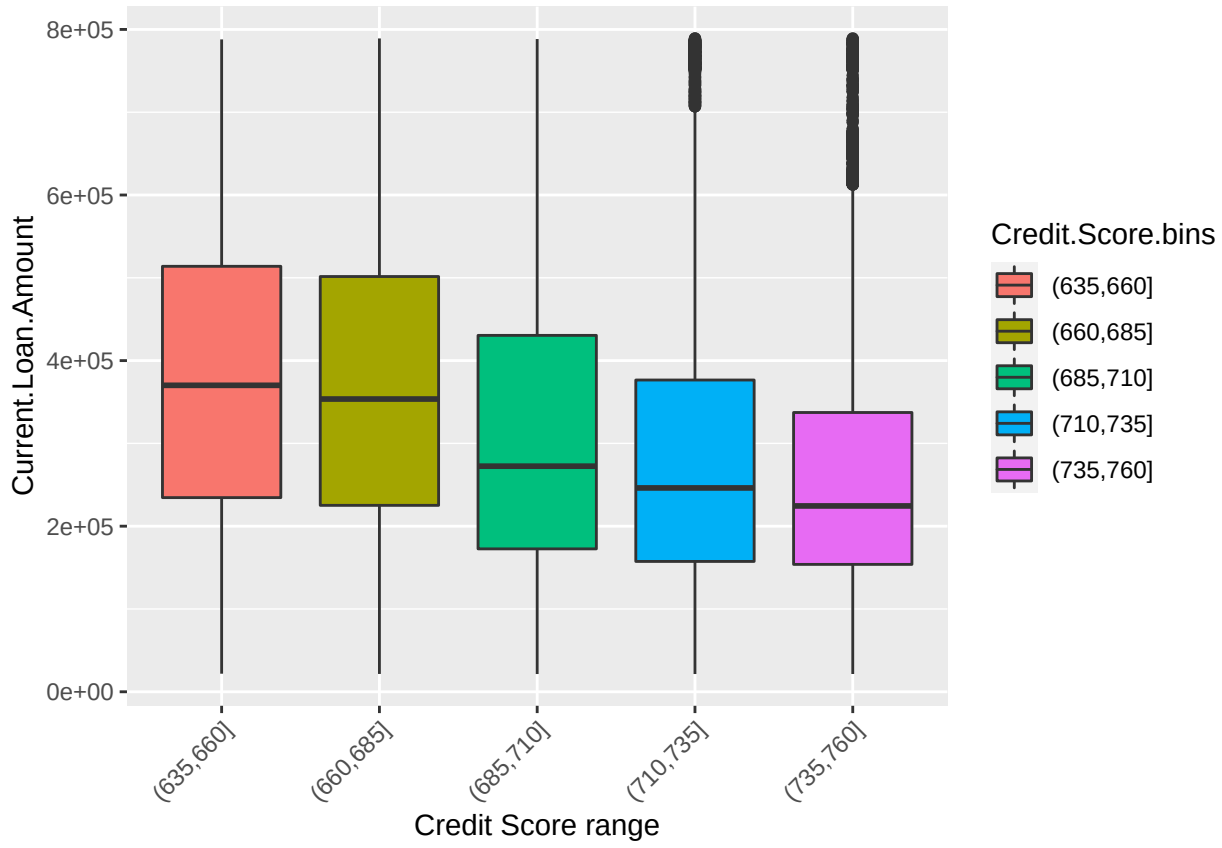

```
### Other Analysis ###

# 1) Annual Income correlated with Current Loan Amount
ggplot(train_no_id, aes(x=Annual.Income.bins, y=Current.Loan.Amount,
                        fill=Annual.Income.bins)) +
  geom_boxplot() + xlab("Annual Income")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



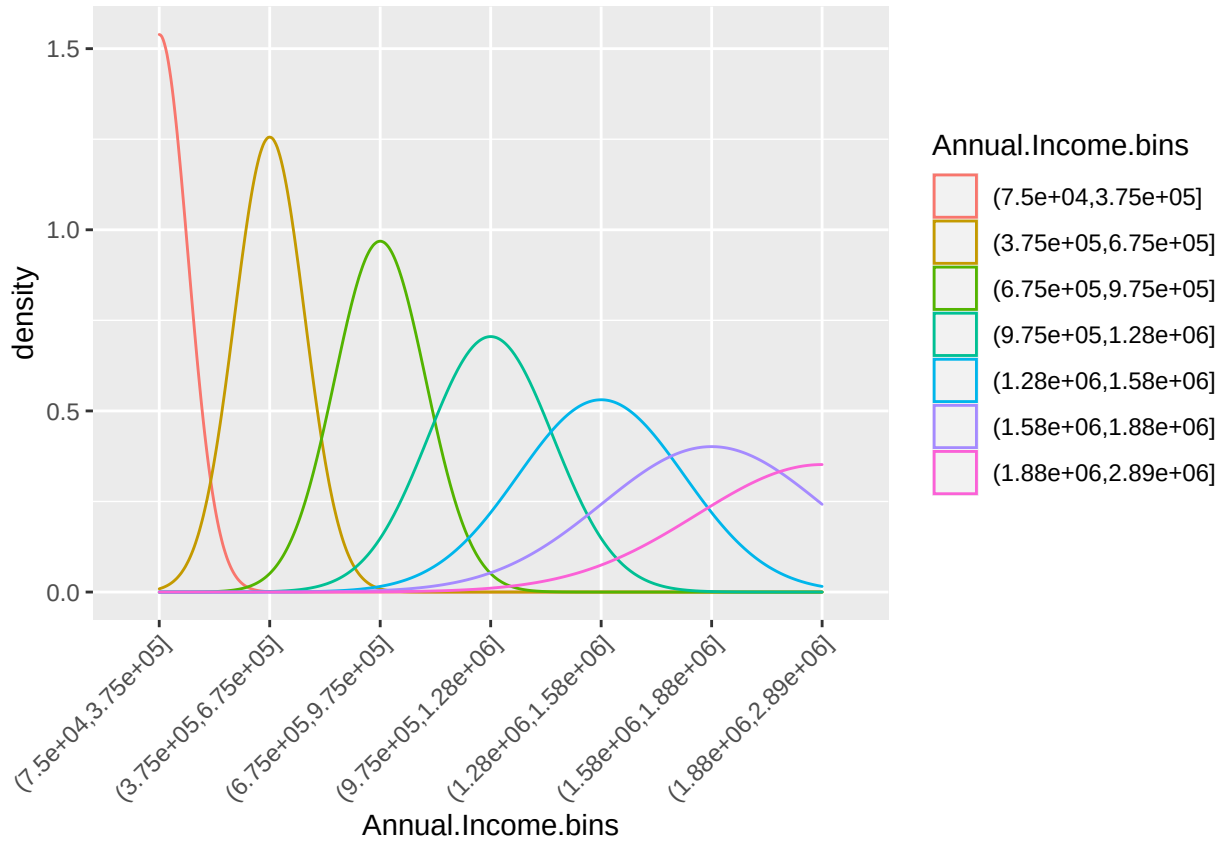

```
# 2) find the range in which most of the income falls

ggplot(train_no_id, aes(x=Years.in.current.job, y=Annual.Income,
                        fill=Years.in.current.job)) +
  geom_boxplot() + xlab("Years in current job")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
# 3) Current loan amount vs Credit score
ggplot(train_no_id, aes(x=Credit.Score.bins, y=Current.Loan.Amount,
                        fill=Credit.Score.bins)) +
  geom_boxplot() + xlab("Credit Score range")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
# 4) More people with less income
ggplot(train_no_id, aes(x=Annual.Income.bins, color=Annual.Income.bins)) +
  geom_density() + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
#Removig co-linear features
train_y = train_one_hot$Loan.Status
train_x = train_one_hot[,-c(1)]
reduced_Data = cor(train_x)
hc = findCorrelation(reduced_Data, cutoff=0.8) # putt any value as a "cutoff"
hc = sort(hc)
train_one_hot_reduced = train_x[,-c(hc)]
train_one_hot_reduced$Loan.Status = train_y
#View(train_one_hot_reduced)

library(dplyr)
#removing perfectly correlated columns
df <- select(train_one_hot_reduced, -c(Years.in.current.job.9.years, Purpose.wedding))

levels(df$Loan.Status)[levels(df$Loan.Status)=="Fully Paid"] <- 1
levels(df$Loan.Status)[levels(df$Loan.Status)=="Charged Off"] <- 0
#View(df)

train_df <- df[1:37567,]
test_df <- df[37568:46958,]

#View(test_df)

#Logistic Regression
model <- glm(Loan.Status ~.,family=binomial(link='logit'),data=train_df)
print(model)
```

```
##
## Call:  glm(formula = Loan.Status ~ ., family = binomial(link = "logit"),
##     data = train_df)
##
## Coefficients:
##                     (Intercept)             Current.Loan.Amount
##                       -1.903e+00                       -9.178e-07
##                    Credit.Score                    Annual.Income
##                       2.889e-03                        7.435e-07
##                    Monthly.Debt          Years.of.Credit.History
##                       -1.787e-05                        1.434e-03
##         Number.of.Open.Accounts        Number.of.Credit.Problems
##                       -1.981e-02                       -6.369e-02
##          Current.Credit.Balance             Maximum.Open.Credit
##                       -7.331e-07                        6.283e-07
##                    Bankruptcies                         Tax.Liens
##                       1.292e-01                       -3.418e-02
##                 Term.Short.Term   Years.in.current.job...1.year
##                       4.102e-01                       -3.205e-03
##     Years.in.current.job.1.year  Years.in.current.job.10..years
##                       3.585e-02                        4.959e-02
##    Years.in.current.job.2.years    Years.in.current.job.3.years
##                       5.650e-02                        1.284e-01
##    Years.in.current.job.4.years    Years.in.current.job.5.years
##                       1.187e-01                        3.494e-02
##    Years.in.current.job.6.years    Years.in.current.job.7.years
##                       2.124e-02                        4.136e-02
##    Years.in.current.job.8.years     Home.Ownership.HaveMortgage
##                       3.968e-03                        1.246e-01
##         Home.Ownership.Own.Home             Home.Ownership.Rent
##                       -1.015e-01                       -2.187e-01
##           Purpose.Business.Loan               Purpose.Buy.a.Car
##                       -5.765e-01                        1.947e-01
##               Purpose.Buy.House      Purpose.Debt.Consolidation
##                       -2.032e-01                       -5.306e-02
##    Purpose.Educational.Expenses       Purpose.Home.Improvements
##                       3.466e-01                       -1.926e-01
##          Purpose.major_purchase           Purpose.Medical.Bills
##                       -2.296e-01                       -3.026e-01
##                  Purpose.moving                   Purpose.other
##                       -3.513e-01                       -1.634e-01
##                    Purpose.Other        Purpose.renewable_energy
##                       3.904e-02                       -8.301e-01
##          Purpose.small_business             Purpose.Take.a.Trip
##                       -1.129e+00                       -7.852e-02
##                Purpose.vacation
##                       -4.234e-01
##
## Degrees of Freedom: 37566 Total (i.e. Null);  37526 Residual
## Null Deviance:       48890
## Residual Deviance: 47180     AIC: 47260
```

```
summary(model)
```

```
##
```

```
## Call:
## glm(formula = Loan.Status ~ ., family = binomial(link = "logit"),
##     data = train_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3074  -1.2953   0.7846   0.9536   1.6743
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -1.903e+00  5.530e-01  -3.441  0.00058 ***
## Current.Loan.Amount            -9.178e-07  8.972e-08 -10.229  < 2e-16 ***
## Credit.Score                    2.889e-03  5.766e-04   5.010 5.44e-07 ***
## Annual.Income                   7.435e-07  3.009e-08  24.710  < 2e-16 ***
## Monthly.Debt                   -1.787e-05  1.719e-06 -10.391  < 2e-16 ***
## Years.of.Credit.History         1.434e-03  1.822e-03   0.787  0.43131
## Number.of.Open.Accounts        -1.981e-02  3.141e-03  -6.306 2.86e-10 ***
## Number.of.Credit.Problems      -6.369e-02  6.171e-02  -1.032  0.30205
## Current.Credit.Balance         -7.331e-07  1.141e-07  -6.425 1.32e-10 ***
## Maximum.Open.Credit             6.283e-07  6.157e-08  10.205  < 2e-16 ***
## Bankruptcies                    1.292e-01  6.909e-02   1.871  0.06139 .
## Tax.Liens                      -3.418e-02  7.567e-02  -0.452  0.65148
## Term.Short.Term                 4.102e-01  2.956e-02  13.876  < 2e-16 ***
## Years.in.current.job...1.year  -3.205e-03  6.553e-02  -0.049  0.96099
## Years.in.current.job.1.year     3.585e-02  6.822e-02   0.526  0.59922
## Years.in.current.job.10..years  4.959e-02  5.729e-02   0.866  0.38676
## Years.in.current.job.2.years    5.650e-02  6.457e-02   0.875  0.38150
## Years.in.current.job.3.years    1.284e-01  6.558e-02   1.957  0.05030 .
## Years.in.current.job.4.years    1.187e-01  6.944e-02   1.709  0.08741 .
## Years.in.current.job.5.years    3.494e-02  6.741e-02   0.518  0.60424
## Years.in.current.job.6.years    2.124e-02  6.948e-02   0.306  0.75986
## Years.in.current.job.7.years    4.136e-02  7.025e-02   0.589  0.55606
## Years.in.current.job.8.years    3.968e-03  7.328e-02   0.054  0.95681
## Home.Ownership.HaveMortgage     1.246e-01  3.217e-01   0.387  0.69847
## Home.Ownership.Own.Home        -1.015e-01  4.119e-02  -2.464  0.01374 *
## Home.Ownership.Rent            -2.187e-01  2.511e-02  -8.710  < 2e-16 ***
## Purpose.Business.Loan          -5.765e-01  3.829e-01  -1.506  0.13213
## Purpose.Buy.a.Car               1.947e-01  3.873e-01   0.503  0.61510
## Purpose.Buy.House              -2.032e-01  3.993e-01  -0.509  0.61079
## Purpose.Debt.Consolidation     -5.306e-02  3.729e-01  -0.142  0.88687
## Purpose.Educational.Expenses    3.466e-01  5.581e-01   0.621  0.53457
## Purpose.Home.Improvements      -1.926e-01  3.760e-01  -0.512  0.60853
## Purpose.major_purchase         -2.296e-01  4.161e-01  -0.552  0.58120
## Purpose.Medical.Bills          -3.026e-01  3.869e-01  -0.782  0.43419
## Purpose.moving                 -3.513e-01  4.636e-01  -0.758  0.44867
## Purpose.other                  -1.634e-01  3.754e-01  -0.435  0.66330
## Purpose.Other                   3.904e-02  3.782e-01   0.103  0.91780
## Purpose.renewable_energy       -8.301e-01  1.003e+00  -0.828  0.40769
## Purpose.small_business         -1.129e+00  4.263e-01  -2.649  0.00807 **
## Purpose.Take.a.Trip            -7.852e-02  4.126e-01  -0.190  0.84908
## Purpose.vacation               -4.234e-01  5.093e-01  -0.831  0.40585
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 48890  on 37566  degrees of freedom
## Residual deviance: 47183  on 37526  degrees of freedom
## AIC: 47265
##
## Number of Fisher Scoring iterations: 4
```

```r
anova(model, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Loan.Status
##
## Terms added sequentially (first to last)
##
##
##                               Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                          37566      48890
## Current.Loan.Amount            1    89.92     37565      48800 < 2.2e-16
## Credit.Score                   1   401.94     37564      48398 < 2.2e-16
## Annual.Income                  1   609.92     37563      47788 < 2.2e-16
## Monthly.Debt                   1   176.63     37562      47612 < 2.2e-16
## Years.of.Credit.History        1     3.71     37561      47608 0.0541188
## Number.of.Open.Accounts        1     3.56     37560      47604 0.0591627
## Number.of.Credit.Problems      1     0.01     37559      47604 0.9091764
## Current.Credit.Balance         1     1.06     37558      47603 0.3028283
## Maximum.Open.Credit            1    76.30     37557      47527 < 2.2e-16
## Bankruptcies                   1    10.87     37556      47516 0.0009756
## Tax.Liens                      1     0.32     37555      47516 0.5741886
## Term.Short.Term                1   161.59     37554      47354 < 2.2e-16
## Years.in.current.job...1.year  1     4.29     37553      47350 0.0383214
## Years.in.current.job.1.year    1     1.07     37552      47349 0.3009053
## Years.in.current.job.10..years 1     0.48     37551      47348 0.4906086
## Years.in.current.job.2.years   1     0.27     37550      47348 0.6007722
## Years.in.current.job.3.years   1     2.83     37549      47345 0.0923182
## Years.in.current.job.4.years   1     2.88     37548      47342 0.0895067
## Years.in.current.job.5.years   1     0.00     37547      47342 0.9783200
## Years.in.current.job.6.years   1     0.00     37546      47342 0.9657740
## Years.in.current.job.7.years   1     0.34     37545      47342 0.5600529
## Years.in.current.job.8.years   1     0.00     37544      47342 0.9936858
## Home.Ownership.HaveMortgage    1     0.94     37543      47341 0.3330744
## Home.Ownership.Own.Home        1     0.14     37542      47341 0.7044470
## Home.Ownership.Rent            1    69.92     37541      47271 < 2.2e-16
## Purpose.Business.Loan          1    30.55     37540      47240 3.253e-08
## Purpose.Buy.a.Car              1     6.84     37539      47234 0.0089296
## Purpose.Buy.House              1     0.73     37538      47233 0.3929358
## Purpose.Debt.Consolidation     1    12.20     37537      47221 0.0004790
## Purpose.Educational.Expenses   1     1.59     37536      47219 0.2070535
## Purpose.Home.Improvements      1     0.44     37535      47219 0.5093422
## Purpose.major_purchase         1     0.18     37534      47219 0.6680454
## Purpose.Medical.Bills          1     2.37     37533      47216 0.1240789
## Purpose.moving                 1     0.63     37532      47216 0.4264034
```

```
## Purpose.other                   1     1.45     37531      47214 0.2280644
## Purpose.Other                   1    13.93     37530      47200 0.0001899
## Purpose.renewable_energy        1     0.16     37529      47200 0.6910970
## Purpose.small_business          1    16.09     37528      47184 6.041e-05
## Purpose.Take.a.Trip             1     0.21     37527      47184 0.6433969
## Purpose.vacation                1     0.70     37526      47183 0.4041486
##
## NULL
## Current.Loan.Amount            ***
## Credit.Score                   ***
## Annual.Income                  ***
## Monthly.Debt                   ***
## Years.of.Credit.History        .
## Number.of.Open.Accounts        .
## Number.of.Credit.Problems
## Current.Credit.Balance
## Maximum.Open.Credit            ***
## Bankruptcies                   ***
## Tax.Liens
## Term.Short.Term                ***
## Years.in.current.job...1.year  *
## Years.in.current.job.1.year
## Years.in.current.job.10..years
## Years.in.current.job.2.years
## Years.in.current.job.3.years   .
## Years.in.current.job.4.years   .
## Years.in.current.job.5.years
## Years.in.current.job.6.years
## Years.in.current.job.7.years
## Years.in.current.job.8.years
## Home.Ownership.HaveMortgage
## Home.Ownership.Own.Home
## Home.Ownership.Rent            ***
## Purpose.Business.Loan          ***
## Purpose.Buy.a.Car              **
## Purpose.Buy.House
## Purpose.Debt.Consolidation     ***
## Purpose.Educational.Expenses
## Purpose.Home.Improvements
## Purpose.major_purchase
## Purpose.Medical.Bills
## Purpose.moving
## Purpose.other
## Purpose.Other                  ***
## Purpose.renewable_energy
## Purpose.small_business         ***
## Purpose.Take.a.Trip
## Purpose.vacation
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fitted.results <- predict(model,newdata=test_df[,-c(41)],type='response')
fitted.results <- ifelse(fitted.results > 0.5,1,0)
```

```
#confusionMatrix(fitted.results != test_df$Loan.Status)

misClasificError <- mean(fitted.results != test_df$Loan.Status)
print(paste('Accuracy',1-misClasificError))
```

```
## [1] "Accuracy 0.952720690022362"
```

```
#"ACCURACY 0.672239378127995"

#kNN Algorithm.............................
#install.packages('class')
library(class)

data_test_pred <- knn(train = train_df, test = test_df,cl = train_df$Loan.Status,k = 9)

table(data_test_pred,test_df$Loan.Status)
```

```
##
## data_test_pred    0    1
##              0    0 1688
##              1    0 7703
```

```
confusionMatrix(data_test_pred,test_df$Loan.Status)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0    0 1688
##          1    0 7703
##
##                Accuracy : 0.8203
##                  95% CI : (0.8123, 0.828)
##     No Information Rate : 1
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity :     NA
##             Specificity : 0.8203
##          Pos Pred Value :     NA
##          Neg Pred Value :     NA
##              Prevalence : 0.0000
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.1797
##       Balanced Accuracy :     NA
##
##        'Positive' Class : 0
##
```

```
# ACCURACY: 0.6561

#Naive Bayes Algorithm...................
```

```r
library(e1071)
#Fitting the Naive Bayes model
Naive_Bayes_Model=naiveBayes(Loan.Status ~., data=train_df)
#What does the model say? Print the model summary
Naive_Bayes_Model
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         0         1
## 0.3553651 0.6446349
##
## Conditional probabilities:
##    Current.Loan.Amount
## Y       [,1]     [,2]
##   0 302222.2 168714.1
##   1 284924.1 168324.6
##
##    Credit.Score
## Y       [,1]     [,2]
##   0 715.8972 23.33703
##   1 721.1635 21.88779
##
##    Annual.Income
## Y      [,1]     [,2]
##   0 1125486 484989.7
##   1 1230944 523464.0
##
##    Monthly.Debt
## Y       [,1]     [,2]
##   0 16685.68 8869.747
##   1 16160.55 8921.606
##
##    Years.of.Credit.History
## Y       [,1]     [,2]
##   0 17.18396 6.559153
##   1 17.54455 6.572023
##
##    Number.of.Open.Accounts
## Y       [,1]     [,2]
##   0 10.66247 4.243515
##   1 10.41062 4.188247
##
##    Number.of.Credit.Problems
## Y        [,1]      [,2]
##   0 0.1672659 0.4909470
##   1 0.1723170 0.4831964
##
##    Current.Credit.Balance
```

```
## Y       [,1]     [,2]
##   0 233220.3 158838.9
##   1 227772.8 162846.1
##
##    Maximum.Open.Credit
## Y       [,1]     [,2]
##   0 479063.2 301530.4
##   1 498847.3 317399.7
##
##    Bankruptcies
## Y        [,1]      [,2]
##   0 0.1141573 0.3446915
##   1 0.1250361 0.3620017
##
##    Tax.Liens
## Y         [,1]      [,2]
##   0 0.02966292 0.2701541
##   1 0.02547797 0.2512999
##
##    Term.Short.Term
## Y        [,1]      [,2]
##   0 0.6639700 0.4723670
##   1 0.7675187 0.4224229
##
##    Years.in.current.job...1.year
## Y         [,1]      [,2]
##   0 0.09078652 0.2873160
##   1 0.08324731 0.2762614
##
##    Years.in.current.job.1.year
## Y         [,1]      [,2]
##   0 0.07003745 0.2552197
##   1 0.06809266 0.2519100
##
##    Years.in.current.job.10..years
## Y        [,1]      [,2]
##   0 0.3080899 0.4617212
##   1 0.3182888 0.4658219
##
##    Years.in.current.job.2.years
## Y         [,1]      [,2]
##   0 0.09617978 0.2948487
##   1 0.09472684 0.2928433
##
##    Years.in.current.job.3.years
## Y         [,1]      [,2]
##   0 0.08464419 0.2783619
##   1 0.08952389 0.2855043
##
##    Years.in.current.job.4.years
## Y         [,1]      [,2]
##   0 0.06179775 0.2407969
##   1 0.06565636 0.2476856
##
```

```
##    Years.in.current.job.5.years
## Y         [,1]      [,2]
##   0 0.07385768 0.2615489
##   1 0.07251105 0.2593376
##
##    Years.in.current.job.6.years
## Y         [,1]      [,2]
##   0 0.06337079 0.2436378
##   1 0.06057728 0.2385582
##
##    Years.in.current.job.7.years
## Y         [,1]      [,2]
##   0 0.05917603 0.2359627
##   1 0.05822356 0.2341705
##
##    Years.in.current.job.8.years
## Y         [,1]      [,2]
##   0 0.04921348 0.2163216
##   1 0.04785894 0.2134721
##
##    Home.Ownership.HaveMortgage
## Y          [,1]       [,2]
##   0 0.001423221 0.03770016
##   1 0.002436305 0.04929980
##
##    Home.Ownership.Own.Home
## Y         [,1]      [,2]
##   0 0.08883895 0.2845218
##   1 0.08741793 0.2824523
##
##    Home.Ownership.Rent
## Y        [,1]      [,2]
##   0 0.4922846 0.4999592
##   1 0.4399802 0.4963948
##
##    Purpose.Business.Loan
## Y         [,1]      [,2]
##   0 0.01820225 0.1336872
##   1 0.01263575 0.1116987
##
##    Purpose.Buy.a.Car
## Y          [,1]       [,2]
##   0 0.009588015 0.09745151
##   1 0.014411364 0.11918164
##
##    Purpose.Buy.House
## Y          [,1]       [,2]
##   0 0.005992509 0.07718190
##   1 0.005822356 0.07608348
##
##    Purpose.Debt.Consolidation
## Y        [,1]      [,2]
##   0 0.8008989 0.3993392
##   1 0.7920882 0.4058217
```

```
##
##    Purpose.Educational.Expenses
## Y           [,1]       [,2]
##   0 0.0005992509 0.02447318
##   1 0.0009497460 0.03080395
##
##    Purpose.Home.Improvements
## Y         [,1]      [,2]
##   0 0.04966292 0.2172557
##   1 0.05578726 0.2295152
##
##    Purpose.major_purchase
## Y          [,1]       [,2]
##   0 0.003595506 0.05985688
##   1 0.003551224 0.05948747
##
##    Purpose.Medical.Bills
## Y         [,1]      [,2]
##   0 0.01168539 0.1074696
##   1 0.01073626 0.1030603
##
##    Purpose.moving
## Y          [,1]       [,2]
##   0 0.001722846 0.04147297
##   1 0.001362679 0.03689009
##
##    Purpose.other
## Y         [,1]      [,2]
##   0 0.05985019 0.2372180
##   1 0.05896684 0.2355675
##
##    Purpose.Other
## Y         [,1]      [,2]
##   0 0.02644195 0.1604515
##   1 0.03377792 0.1806608
##
##    Purpose.renewable_energy
## Y           [,1]       [,2]
##   0 0.0001498127 0.01223934
##   1 0.0001238799 0.01112968
##
##    Purpose.small_business
## Y          [,1]       [,2]
##   0 0.004419476 0.06633456
##   1 0.001734319 0.04160988
##
##    Purpose.Take.a.Trip
## Y          [,1]       [,2]
##   0 0.005243446 0.07222425
##   1 0.006276583 0.07897749
##
##    Purpose.vacation
## Y           [,1]       [,2]
##   0 0.0011235955 0.03350249
```

```
##   1 0.0008258661 0.02872661
```

```
#Prediction on the dataset
NB_Predictions=predict(Naive_Bayes_Model,test_df)
#Confusion matrix to check accuracy
table(NB_Predictions, test_df$Loan.Status)
```

```
##
## NB_Predictions    0    1
##              0    0 1116
##              1    0 8275
```

```
confusionMatrix(NB_Predictions, test_df$Loan.Status)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0    0 1116
##          1    0 8275
##
##                Accuracy : 0.8812
##                  95% CI : (0.8744, 0.8876)
##     No Information Rate : 1
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity :     NA
##             Specificity : 0.8812
##          Pos Pred Value :     NA
##          Neg Pred Value :     NA
##              Prevalence : 0.0000
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.1188
##       Balanced Accuracy :     NA
##
##        'Positive' Class : 0
##
```

```
# ACCURACY : 0.6674

#Random Forest Algorithm
#install.packages("randomForest")
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

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

```
model1 <- randomForest(Loan.Status ~ ., data = train_df, importance = TRUE)
model1
```

```
##
## Call:
##  randomForest(formula = Loan.Status ~ ., data = train_df, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 6
##
##         OOB estimate of  error rate: 35.26%
## Confusion matrix:
##      0     1 class.error
## 0 1973 11377   0.8522097
## 1 1868 22349   0.0771359
```

```
model2 <- randomForest(Loan.Status ~ ., data = train_df, ntree = 500, mtry = 6, importance = TRUE)
model2
```

```
##
## Call:
##  randomForest(formula = Loan.Status ~ ., data = train_df, ntree = 500,      mtry = 6, importance = TR
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 6
##
##         OOB estimate of  error rate: 35.07%
## Confusion matrix:
##      0     1 class.error
## 0 1971 11379  0.85235955
## 1 1795 22422  0.07412148
```

```
# Predicting on test set
predTest <- predict(model1, test_df, type = "class")
# Checking classification accuracy
confusionMatrix(predTest, test_df$Loan.Status)
```

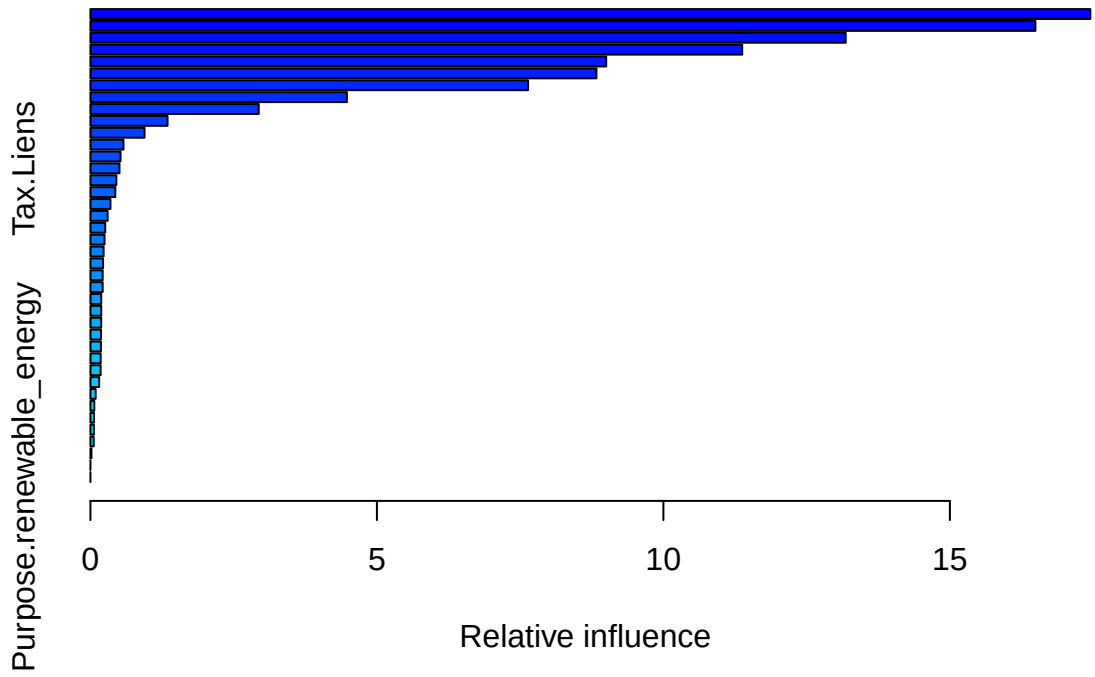
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0    0  572
##          1    0 8819
##
##                Accuracy : 0.9391
##                  95% CI : (0.9341, 0.9438)
##     No Information Rate : 1
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
```

```
##             Sensitivity :      NA
##             Specificity : 0.93909
##          Pos Pred Value :      NA
##          Neg Pred Value :      NA
##              Prevalence : 0.00000
##          Detection Rate : 0.00000
##    Detection Prevalence : 0.06091
##       Balanced Accuracy :      NA
##
##        'Positive' Class : 0
##
```

```
#install.packages("gbm")
library(gbm)
```

```
## Loaded gbm 2.1.5
```

```
set.seed(1)
boost.loan=gbm(Loan.Status~.,data=train_df,distribution="gaussian",n.trees=500,interaction.depth=4,cv.f
summary(boost.loan)
```




```
##                                                    var     rel.inf
## Credit.Score                                Credit.Score 17.45192020
## Annual.Income                              Annual.Income 16.49279146
## Current.Loan.Amount                  Current.Loan.Amount 13.18173996
## Monthly.Debt                                Monthly.Debt 11.37533944
## Maximum.Open.Credit                  Maximum.Open.Credit  9.00178006
## Current.Credit.Balance            Current.Credit.Balance  8.83147630
```

```
## Years.of.Credit.History              Years.of.Credit.History  7.63765382
## Term.Short.Term                              Term.Short.Term  4.47548089
## Number.of.Open.Accounts            Number.of.Open.Accounts  2.93738830
## Home.Ownership.Rent                   Home.Ownership.Rent  1.34575632
## Purpose.Debt.Consolidation       Purpose.Debt.Consolidation  0.94404247
## Purpose.Business.Loan                 Purpose.Business.Loan  0.57665921
## Purpose.small_business               Purpose.small_business  0.52541047
## Tax.Liens                                         Tax.Liens  0.50673894
## Bankruptcies                                   Bankruptcies  0.45227235
## Number.of.Credit.Problems         Number.of.Credit.Problems  0.43434121
## Years.in.current.job...1.year   Years.in.current.job...1.year  0.34947319
## Years.in.current.job.10..years Years.in.current.job.10..years  0.30002535
## Years.in.current.job.3.years     Years.in.current.job.3.years  0.25770357
## Years.in.current.job.5.years     Years.in.current.job.5.years  0.24617126
## Purpose.Home.Improvements          Purpose.Home.Improvements  0.22965936
## Years.in.current.job.8.years     Years.in.current.job.8.years  0.22066236
## Purpose.Buy.a.Car                           Purpose.Buy.a.Car  0.21364891
## Years.in.current.job.4.years     Years.in.current.job.4.years  0.21343869
## Purpose.Medical.Bills                 Purpose.Medical.Bills  0.18906596
## Years.in.current.job.1.year       Years.in.current.job.1.year  0.18906359
## Years.in.current.job.2.years     Years.in.current.job.2.years  0.18805823
## Home.Ownership.Own.Home             Home.Ownership.Own.Home  0.18469093
## Years.in.current.job.6.years     Years.in.current.job.6.years  0.18307433
## Purpose.Other                                 Purpose.Other  0.17904887
## Purpose.moving                               Purpose.moving  0.17847437
## Purpose.other                                 Purpose.other  0.15245782
## Purpose.major_purchase               Purpose.major_purchase  0.09036764
## Years.in.current.job.7.years     Years.in.current.job.7.years  0.06734562
## Purpose.Educational.Expenses     Purpose.Educational.Expenses  0.06204908
## Purpose.Take.a.Trip                   Purpose.Take.a.Trip  0.06041012
## Purpose.vacation                         Purpose.vacation  0.05700729
## Purpose.Buy.House                       Purpose.Buy.House  0.01731205
## Home.Ownership.HaveMortgage       Home.Ownership.HaveMortgage  0.00000000
## Purpose.renewable_energy             Purpose.renewable_energy  0.00000000
```