raw data processing

Yunxiao Zhao and Zhenhuan Xie

11/23/2019

```r
# read raw data
data_Q1 <- read.csv('LoanStats_securev1_2018Q1.csv', skip=1)
dnames <- colnames(data_Q1)

# BrowseNote (variables available to investors)
browsenote <- read.csv('BrowseNote.csv')
browsenote <- browsenote[1:120, ]
bnames <- as.vector(browsenote[,1])
# transform to lower case
bnames <- as.vector(sapply(bnames, tolower))

# remove punctuation characters and blanks
bnames <- gsub('[[:blank:]$]', '', bnames)
bnames_pure <- gsub('[[:punct:]]', '', bnames)
dnames_pure <- gsub('[[:punct:]]', '', dnames)

# check identical variable names
var_index <- dnames_pure %in% bnames_pure
dnames <- dnames[var_index]
dnames[104] <- 'loan_status'

# subset the data
data_Q1 <- subset(data_Q1, select=dnames)
dim(data_Q1) # now we have 104 variables (1 response)
```

```
## [1] 107866    104
```

```r
# drop data with no response or multiple applicants
'%ni%' <- Negate('%in%')
data_Q1 <- data_Q1[data_Q1$loan_status %ni% c('Current', ''), ]
data_Q1 <- data_Q1[data_Q1$application_type %ni% c('Joint App', ''), ]
dim(data_Q1) # 104 variables (1 response)
```

```
## [1] 35407   104
```

```r
# remove date time values
date_var <- c('earliest_cr_line', 'sec_app_earliest_cr_line')
data_Q1 <- data_Q1[, !(names(data_Q1) %in% date_var)]
dim(data_Q1) # 102 variables
```

```
## [1] 35407   102
```

```r
# check missing values
missing <- colSums(is.na(data_Q1))
n <- dim(data_Q1)[1]
data_Q1 <- data_Q1[, missing/n<0.05]
dim(data_Q1) # 81 variables
```

```
## [1] 35407    81
```

```r
# remove 'num_tl_120dpd_2m' because it has only 1 category
drop <- c('num_tl_120dpd_2m')

# multicolinearity, keep one with better interpretation
# 'grade', 'subgrade', 'int_rate'
drop = c(drop, 'grade', 'int_rate')
# 'purpose', 'title'
drop = c(drop, 'title')
# 'acc_now_delinq', 'num_tl_30dpd'
drop = c(drop, 'num_tl_30dpd')
# 'fico_range_high', 'fico_range_low'
drop = c(drop, 'fico_range_high')
# 'funded_amnt', 'installment', 'loan_amnt'
drop = c(drop, 'funded_amnt', 'loan_amnt')
# 'mo_sin_old_il_acct', 'mths_since_rcnt_il'
drop = c(drop, 'mo_sin_old_il_acct')
# 'num_actv_rev_tl', 'num_rev_tl_bal_gt_0'
drop = c(drop, 'num_rev_tl_bal_gt_0')
# 'num_sats', 'open_acc'
drop = c(drop, 'num_sats')
# 'tot_cur_bal', 'tot_hi_cred_lim'
drop = c(drop, 'tot_hi_cred_lim')
# 'total_bal_ex_mort', 'total_bal_il', 'total_il_high_credit_limit'
drop = c(drop, 'total_bal_ex_mort')
data_Q1 <- data_Q1[, !(names(data_Q1) %in% drop)]
dim(data_Q1) # 69 variables
```

```
## [1] 35407    69
```

```r
# unrelated variables
drop2 <- c('id', 'emp_title', 'url', 'zip_code', 'addr_state', 'inital_list_status', 'collections_12_mt
data_Q1 <- data_Q1[, !(names(data_Q1) %in% drop2)]
dim(data_Q1) # 61 variables
```

```
## [1] 35407    61
```

```r
# transformation
data_Q1$dti <- data_Q1$dti / 100
data_Q1$pct_tl_nvr_dlq <- data_Q1$pct_tl_nvr_dlq / 100
data_Q1$annual_inc <- log(data_Q1$annual_inc)
data_Q1$installment <- log(data_Q1$installment)
data_Q1$revol_bal <- ifelse(data_Q1$revol_bal==0, 0, log(data_Q1$revol_bal))
data_Q1$total_bal_il <- ifelse(data_Q1$total_bal_il==0, 0, log(data_Q1$total_bal_il))
data_Q1$tot_cur_bal <- ifelse(data_Q1$tot_cur_bal==0, 0, log(data_Q1$tot_cur_bal))
data_Q1$max_bal_bc <- ifelse(data_Q1$max_bal_bc==0, 0, log(data_Q1$max_bal_bc))
data_Q1$total_rev_hi_lim <- ifelse(data_Q1$total_rev_hi_lim==0, 0, log(data_Q1$total_rev_hi_lim))
data_Q1$avg_cur_bal <- ifelse(data_Q1$avg_cur_bal==0, 0, log(data_Q1$avg_cur_bal))
data_Q1$bc_open_to_buy <- ifelse(data_Q1$bc_open_to_buy==0, 0, log(data_Q1$bc_open_to_buy))
data_Q1$total_bc_limit <- ifelse(data_Q1$total_bc_limit==0, 0, log(data_Q1$total_bc_limit))
data_Q1$total_il_high_credit_limit <- ifelse(data_Q1$total_il_high_credit_limit==0, 0, log(data_Q1$total
data_Q1$percent_bc_gt_75 <- data_Q1$percent_bc_gt_75 / 100
data_Q1$revol_util <- as.numeric(gsub('\\%','',data_Q1$revol_util)) / 100

# na values
data_Q1[is.na(data_Q1)] <- 0
```

```r
# transform response variable and write into new csv
data_Q1$loan_status <- ifelse(data_Q1$loan_status=='Fully Paid', 1, 0)
dim(data_Q1) # 60 independent variables, 1 response, 34602 observation
```

```
## [1] 35407    61
```

```r
write.csv(data_Q1,'2018Q1_processed.csv', row.names = FALSE)
```

```r
# read raw data
data_Q2 <- read.csv('LoanStats_securev1_2018Q2.csv', skip=1)
dnames <- colnames(data_Q2)

# BrowseNote (variables available to investors)
browsenote <- read.csv('BrowseNote.csv')
browsenote <- browsenote[1:120, ]
bnames <- as.vector(browsenote[,1])
# transform to lower case
bnames <- as.vector(sapply(bnames, tolower))

# remove punctuation characters and blanks
bnames <- gsub('[[:blank:]$]', '', bnames)
bnames_pure <- gsub('[[:punct:]]', '', bnames)
dnames_pure <- gsub('[[:punct:]]', '', dnames)

# check identical variable names
var_index <- dnames_pure %in% bnames_pure
dnames <- dnames[var_index]
dnames[104] <- 'loan_status'

# subset the data
data_Q2 <- subset(data_Q2, select=dnames)
dim(data_Q2) # now we have 104 variables (1 response)
```

```
## [1] 130774    104
```

```r
# drop data with no response or multiple applicants
'%ni%' <- Negate('%in%')
data_Q2 <- data_Q2[data_Q2$loan_status %ni% c('Current', ''), ]
data_Q2 <- data_Q2[data_Q2$application_type %ni% c('Joint App', ''), ]
dim(data_Q2) # 104 variables (1 response)
```

```
## [1] 36176   104
```

```r
# remove date time values
date_var <- c('earliest_cr_line', 'sec_app_earliest_cr_line')
data_Q2 <- data_Q2[, !(names(data_Q2) %in% date_var)]
dim(data_Q2) # 102 variables
```

```
## [1] 36176   102
```

```r
# check missing values
missing <- colSums(is.na(data_Q2))
n <- dim(data_Q2)[1]
data_Q2 <- data_Q2[, missing/n<0.05]
dim(data_Q2) # 81 variables
```

```
## [1] 36176    81
```

```r
# remove 'num_tl_120dpd_2m' because it has only 1 category
drop <- c('num_tl_120dpd_2m')

# multicolinearity, keep one with better interpretation
# 'grade', 'subgrade', 'int_rate'
drop = c(drop, 'grade', 'int_rate')
# 'purpose', 'title'
drop = c(drop, 'title')
# 'acc_now_delinq', 'num_tl_30dpd'
drop = c(drop, 'num_tl_30dpd')
# 'fico_range_high', 'fico_range_low'
drop = c(drop, 'fico_range_high')
# 'funded_amnt', 'installment', 'loan_amnt'
drop = c(drop, 'funded_amnt', 'loan_amnt')
# 'mo_sin_old_il_acct', 'mths_since_rcnt_il'
drop = c(drop, 'mo_sin_old_il_acct')
# 'num_actv_rev_tl', 'num_rev_tl_bal_gt_0'
drop = c(drop, 'num_rev_tl_bal_gt_0')
# 'num_sats', 'open_acc'
drop = c(drop, 'num_sats')
# 'tot_cur_bal', 'tot_hi_cred_lim'
drop = c(drop, 'tot_hi_cred_lim')
# 'total_bal_ex_mort', 'total_bal_il', 'total_il_high_credit_limit'
drop = c(drop, 'total_bal_ex_mort')
data_Q2 <- data_Q2[, !(names(data_Q2) %in% drop)]
dim(data_Q2) # 69 variables
```

```
## [1] 36176    69
```

```r
# unrelated variables
drop2 <- c('id', 'emp_title', 'url', 'zip_code', 'addr_state', 'inital_list_status', 'collections_12_mth
data_Q2 <- data_Q2[, !(names(data_Q2) %in% drop2)]
dim(data_Q2) # 61 variables
```

```
## [1] 36176    61
```

```r
# transformation
data_Q2$dti <- data_Q2$dti / 100
data_Q2$pct_tl_nvr_dlq <- data_Q2$pct_tl_nvr_dlq / 100
data_Q2$annual_inc <- log(data_Q2$annual_inc)
data_Q2$installment <- log(data_Q2$installment)
data_Q2$revol_bal <- ifelse(data_Q2$revol_bal==0, 0, log(data_Q2$revol_bal))
data_Q2$total_bal_il <- ifelse(data_Q2$total_bal_il==0, 0, log(data_Q2$total_bal_il))
data_Q2$tot_cur_bal <- ifelse(data_Q2$tot_cur_bal==0, 0, log(data_Q2$tot_cur_bal))
data_Q2$max_bal_bc <- ifelse(data_Q2$max_bal_bc==0, 0, log(data_Q2$max_bal_bc))
data_Q2$total_rev_hi_lim <- ifelse(data_Q2$total_rev_hi_lim==0, 0, log(data_Q2$total_rev_hi_lim))
data_Q2$avg_cur_bal <- ifelse(data_Q2$avg_cur_bal==0, 0, log(data_Q2$avg_cur_bal))
data_Q2$bc_open_to_buy <- ifelse(data_Q2$bc_open_to_buy==0, 0, log(data_Q2$bc_open_to_buy))
data_Q2$total_bc_limit <- ifelse(data_Q2$total_bc_limit==0, 0, log(data_Q2$total_bc_limit))
data_Q2$total_il_high_credit_limit <- ifelse(data_Q2$total_il_high_credit_limit==0, 0, log(data_Q2$total
data_Q2$percent_bc_gt_75 <- data_Q2$percent_bc_gt_75 / 100
data_Q2$revol_util <- as.numeric(gsub('\\%','',data_Q2$revol_util)) / 100

# na values
data_Q2[is.na(data_Q2)] <- 0
```

```r
# transform response variable and write into new csv
data_Q2$loan_status <- ifelse(data_Q2$loan_status=='Fully Paid', 1, 0)
dim(data_Q2) # 60 independent variables, 1 response, 34602 observation
```

```
## [1] 36176    61
```

```r
write.csv(data_Q2,'2018Q2_processed.csv', row.names = FALSE)
```

```r
# read raw data
data_Q3 <- read.csv('LoanStats_securev1_2018Q3.csv', skip=1)
dnames <- colnames(data_Q3)

# BrowseNote (variables available to investors)
browsenote <- read.csv('BrowseNote.csv')
browsenote <- browsenote[1:120, ]
bnames <- as.vector(browsenote[,1])
# transform to lower case
bnames <- as.vector(sapply(bnames, tolower))

# remove punctuation characters and blanks
bnames <- gsub('[[:blank:]$]', '', bnames)
bnames_pure <- gsub('[[:punct:]]', '', bnames)
dnames_pure <- gsub('[[:punct:]]', '', dnames)

# check identical variable names
var_index <- dnames_pure %in% bnames_pure
dnames <- dnames[var_index]
dnames[104] <- 'loan_status'

# subset the data
data_Q3 <- subset(data_Q3, select=dnames)
dim(data_Q3) # now we have 104 variables (1 response)
```

```
## [1] 128196    104
```

```r
# drop data with no response or multiple applicants
'%ni%' <- Negate('%in%')
data_Q3 <- data_Q3[data_Q3$loan_status %ni% c('Current', ''), ]
data_Q3 <- data_Q3[data_Q3$application_type %ni% c('Joint App', ''), ]
dim(data_Q3) # 104 variables (1 response)
```

```
## [1] 27593   104
```

```r
# remove date time values
date_var <- c('earliest_cr_line', 'sec_app_earliest_cr_line')
data_Q3 <- data_Q3[, !(names(data_Q3) %in% date_var)]
dim(data_Q3) # 102 variables
```

```
## [1] 27593   102
```

```r
# check missing values
missing <- colSums(is.na(data_Q3))
n <- dim(data_Q3)[1]
data_Q3 <- data_Q3[, missing/n<0.05]
dim(data_Q3) # 81 variables
```

```
## [1] 27593    81
```

```r
# remove 'num_tl_120dpd_2m' because it has only 1 category
drop <- c('num_tl_120dpd_2m')

# multicolinearity, keep one with better interpretation
# 'grade', 'subgrade', 'int_rate'
drop = c(drop, 'grade', 'int_rate')
# 'purpose', 'title'
drop = c(drop, 'title')
# 'acc_now_delinq', 'num_tl_30dpd'
drop = c(drop, 'num_tl_30dpd')
# 'fico_range_high', 'fico_range_low'
drop = c(drop, 'fico_range_high')
# 'funded_amnt', 'installment', 'loan_amnt'
drop = c(drop, 'funded_amnt', 'loan_amnt')
# 'mo_sin_old_il_acct', 'mths_since_rcnt_il'
drop = c(drop, 'mo_sin_old_il_acct')
# 'num_actv_rev_tl', 'num_rev_tl_bal_gt_0'
drop = c(drop, 'num_rev_tl_bal_gt_0')
# 'num_sats', 'open_acc'
drop = c(drop, 'num_sats')
# 'tot_cur_bal', 'tot_hi_cred_lim'
drop = c(drop, 'tot_hi_cred_lim')
# 'total_bal_ex_mort', 'total_bal_il', 'total_il_high_credit_limit'
drop = c(drop, 'total_bal_ex_mort')
data_Q3 <- data_Q3[, !(names(data_Q3) %in% drop)]
dim(data_Q3) # 69 variables
```

```
## [1] 27593    69
```

```r
# unrelated variables
drop2 <- c('id', 'emp_title', 'url', 'zip_code', 'addr_state', 'inital_list_status', 'collections_12_mth
data_Q3 <- data_Q3[, !(names(data_Q3) %in% drop2)]
dim(data_Q3) # 61 variables
```

```
## [1] 27593    61
```

```r
# transformation
data_Q3$dti <- data_Q3$dti / 100
data_Q3$pct_tl_nvr_dlq <- data_Q3$pct_tl_nvr_dlq / 100
data_Q3$annual_inc <- log(data_Q3$annual_inc)
data_Q3$installment <- log(data_Q3$installment)
data_Q3$revol_bal <- ifelse(data_Q3$revol_bal==0, 0, log(data_Q3$revol_bal))
data_Q3$total_bal_il <- ifelse(data_Q3$total_bal_il==0, 0, log(data_Q3$total_bal_il))
data_Q3$tot_cur_bal <- ifelse(data_Q3$tot_cur_bal==0, 0, log(data_Q3$tot_cur_bal))
data_Q3$max_bal_bc <- ifelse(data_Q3$max_bal_bc==0, 0, log(data_Q3$max_bal_bc))
data_Q3$total_rev_hi_lim <- ifelse(data_Q3$total_rev_hi_lim==0, 0, log(data_Q3$total_rev_hi_lim))
data_Q3$avg_cur_bal <- ifelse(data_Q3$avg_cur_bal==0, 0, log(data_Q3$avg_cur_bal))
data_Q3$bc_open_to_buy <- ifelse(data_Q3$bc_open_to_buy==0, 0, log(data_Q3$bc_open_to_buy))
data_Q3$total_bc_limit <- ifelse(data_Q3$total_bc_limit==0, 0, log(data_Q3$total_bc_limit))
data_Q3$total_il_high_credit_limit <- ifelse(data_Q3$total_il_high_credit_limit==0, 0, log(data_Q3$total
data_Q3$percent_bc_gt_75 <- data_Q3$percent_bc_gt_75 / 100
data_Q3$revol_util <- as.numeric(gsub('\\%','',data_Q3$revol_util)) / 100

# na values
data_Q3[is.na(data_Q3)] <- 0
```

```
# transform response variable and write into new csv
data_Q3$loan_status <- ifelse(data_Q3$loan_status=='Fully Paid', 1, 0)
dim(data_Q3) # 60 independent variables, 1 response, 34602 observation
```

```
## [1] 27593    61
```

```
write.csv(data_Q3,'2018Q3_processed.csv', row.names = FALSE)
```

```
# read raw data
data_Q4 <- read.csv('LoanStats_securev1_2018Q4.csv', skip=1)
dnames <- colnames(data_Q4)

# BrowseNote (variables available to investors)
browsenote <- read.csv('BrowseNote.csv')
browsenote <- browsenote[1:120, ]
bnames <- as.vector(browsenote[,1])
# transform to lower case
bnames <- as.vector(sapply(bnames, tolower))

# remove punctuation characters and blanks
bnames <- gsub('[[:blank:]$]', '', bnames)
bnames_pure <- gsub('[[:punct:]]', '', bnames)
dnames_pure <- gsub('[[:punct:]]', '', dnames)

# check identical variable names
var_index <- dnames_pure %in% bnames_pure
dnames <- dnames[var_index]
dnames[104] <- 'loan_status'

# subset the data
data_Q4 <- subset(data_Q4, select=dnames)
dim(data_Q4) # now we have 104 variables (1 response)
```

```
## [1] 128414    104
```

```
# drop data with no response or multiple applicants
'%ni%' <- Negate('%in%')
data_Q4 <- data_Q4[data_Q4$loan_status %ni% c('Current', ''), ]
data_Q4 <- data_Q4[data_Q4$application_type %ni% c('Joint App', ''), ]
dim(data_Q4) # 104 variables (1 response)
```

```
## [1] 20720   104
```

```
# remove date time values
date_var <- c('earliest_cr_line', 'sec_app_earliest_cr_line')
data_Q4 <- data_Q4[, !(names(data_Q4) %in% date_var)]
dim(data_Q4) # 102 variables
```

```
## [1] 20720   102
```

```
# check missing values
missing <- colSums(is.na(data_Q4))
n <- dim(data_Q4)[1]
data_Q4 <- data_Q4[, missing/n<0.05]
dim(data_Q4) # 81 variables
```

```
## [1] 20720    81
```

```
# remove 'num_tl_120dpd_2m' because it has only 1 category
drop <- c('num_tl_120dpd_2m')

# multicolinearity, keep one with better interpretation
# 'grade', 'subgrade', 'int_rate'
drop = c(drop, 'grade', 'int_rate')
# 'purpose', 'title'
drop = c(drop, 'title')
# 'acc_now_delinq', 'num_tl_30dpd'
drop = c(drop, 'num_tl_30dpd')
# 'fico_range_high', 'fico_range_low'
drop = c(drop, 'fico_range_high')
# 'funded_amnt', 'installment', 'loan_amnt'
drop = c(drop, 'funded_amnt', 'loan_amnt')
# 'mo_sin_old_il_acct', 'mths_since_rcnt_il'
drop = c(drop, 'mo_sin_old_il_acct')
# 'num_actv_rev_tl', 'num_rev_tl_bal_gt_0'
drop = c(drop, 'num_rev_tl_bal_gt_0')
# 'num_sats', 'open_acc'
drop = c(drop, 'num_sats')
# 'tot_cur_bal', 'tot_hi_cred_lim'
drop = c(drop, 'tot_hi_cred_lim')
# 'total_bal_ex_mort', 'total_bal_il', 'total_il_high_credit_limit'
drop = c(drop, 'total_bal_ex_mort')
data_Q4 <- data_Q4[, !(names(data_Q4) %in% drop)]
dim(data_Q4) # 69 variables
```

```
## [1] 20720    69
```

```
# unrelated variables
drop2 <- c('id', 'emp_title', 'url', 'zip_code', 'addr_state', 'inital_list_status', 'collections_12_mth
data_Q4 <- data_Q4[, !(names(data_Q4) %in% drop2)]
dim(data_Q4) # 61 variables
```

```
## [1] 20720    61
```

```
# transformation
data_Q4$dti <- data_Q4$dti / 100
data_Q4$pct_tl_nvr_dlq <- data_Q4$pct_tl_nvr_dlq / 100
data_Q4$annual_inc <- log(data_Q4$annual_inc)
data_Q4$installment <- log(data_Q4$installment)
data_Q4$revol_bal <- ifelse(data_Q4$revol_bal==0, 0, log(data_Q4$revol_bal))
data_Q4$total_bal_il <- ifelse(data_Q4$total_bal_il==0, 0, log(data_Q4$total_bal_il))
data_Q4$tot_cur_bal <- ifelse(data_Q4$tot_cur_bal==0, 0, log(data_Q4$tot_cur_bal))
data_Q4$max_bal_bc <- ifelse(data_Q4$max_bal_bc==0, 0, log(data_Q4$max_bal_bc))
data_Q4$total_rev_hi_lim <- ifelse(data_Q4$total_rev_hi_lim==0, 0, log(data_Q4$total_rev_hi_lim))
data_Q4$avg_cur_bal <- ifelse(data_Q4$avg_cur_bal==0, 0, log(data_Q4$avg_cur_bal))
data_Q4$bc_open_to_buy <- ifelse(data_Q4$bc_open_to_buy==0, 0, log(data_Q4$bc_open_to_buy))
data_Q4$total_bc_limit <- ifelse(data_Q4$total_bc_limit==0, 0, log(data_Q4$total_bc_limit))
data_Q4$total_il_high_credit_limit <- ifelse(data_Q4$total_il_high_credit_limit==0, 0, log(data_Q4$total
data_Q4$percent_bc_gt_75 <- data_Q4$percent_bc_gt_75 / 100
data_Q4$revol_util <- as.numeric(gsub('\\%','',data_Q4$revol_util)) / 100

# na values
data_Q4[is.na(data_Q4)] <- 0
```

```r
# transform response variable and write into new csv
data_Q4$loan_status <- ifelse(data_Q4$loan_status=='Fully Paid', 1, 0)
dim(data_Q4) # 60 independent variables, 1 response, 34602 observation
```

```
## [1] 20720    61
```

```r
write.csv(data_Q4,'2018Q4_processed.csv', row.names = FALSE)
```

```r
# Comparison between seasons
Q1=table(data_Q1$loan_status)
Q1
```

```
##
##     0     1
##  9677 25730
```

```r
Q2=table(data_Q2$loan_status)
Q2
```

```
##
##     0     1
## 10949 25227
```

```r
Q3=table(data_Q3$loan_status)
Q3
```

```
##
##     0     1
##  7836 19757
```

```r
Q4=table(data_Q4$loan_status)
Q4
```

```
##
##     0     1
##  6062 14658
```

```r
table.2018=cbind(Q1,Q2,Q3,Q4)
table.2018
```

```
##      Q1    Q2    Q3    Q4
## 0  9677 10949  7836  6062
## 1 25730 25227 19757 14658
```

```r
chisq.test(table.2018)
```

```
##
##  Pearson's Chi-squared test
##
## data:  table.2018
## X-squared = 79.469, df = 3, p-value < 2.2e-16
```

```r
data_full_18=rbind(data_Q1,data_Q2,data_Q3,data_Q4)
dim(data_full_18)
```

```
## [1] 119896     61
```

```r
write.csv(data_full_18,'2018full_processed.csv', row.names = FALSE)
```

```
data_Q1$season=as.factor("Q1")
data_Q2$season=as.factor("Q2")
data_Q3$season=as.factor("Q3")
data_Q4$season=as.factor("Q4")
data_full_18=rbind(data_Q1,data_Q2,data_Q3,data_Q4)

# size=round(c(50000*nrow(data_Q1)/nrow(data_full_18),50000*nrow(data_Q2)/nrow(data_full_18),50000*nrow
# set.seed(0)
# data_sample_18=Strata(data_full_18,stratanames = "season", size = size,method = "srswr")
# data_sample_18=data_sample_18[,1:62]  #remove the unnecessary columns created during sampling process
data_sample_18=data_full_18
dim(data_sample_18)
```

```
## [1] 119896     62
```

```
library(fastDummies)
```

```
## Warning: package 'fastDummies' was built under R version 3.5.2
```

```
data_sample_18_dummy=dummy_cols(data_sample_18,remove_most_frequent_dummy=T)
data_sample_18_dummy=subset(data_sample_18_dummy, select=-c(season,term,sub_grade,emp_length,home_owner
data_sample_18_dummy$loan_status=as.factor(data_sample_18_dummy$loan_status)

library(caret)
```

```
## Warning: package 'caret' was built under R version 3.5.2

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
set.seed(0)
splitIndex <- createDataPartition(data_sample_18_dummy$loan_status, p = 0.5, list =FALSE, times = 1)
test <- data_sample_18_dummy[-splitIndex,]
train <- data_sample_18_dummy[splitIndex,]
table(train$loan_status)
```

```
##
##     0     1
## 17262 42686
```

```
library(DMwR)
```

```
## Loading required package: grid
```

```
trainBalance=SMOTE(loan_status~ ., train, perc.over = 100, perc.under=200)
dim(trainBalance)
```

```
## [1] 69048   120
```

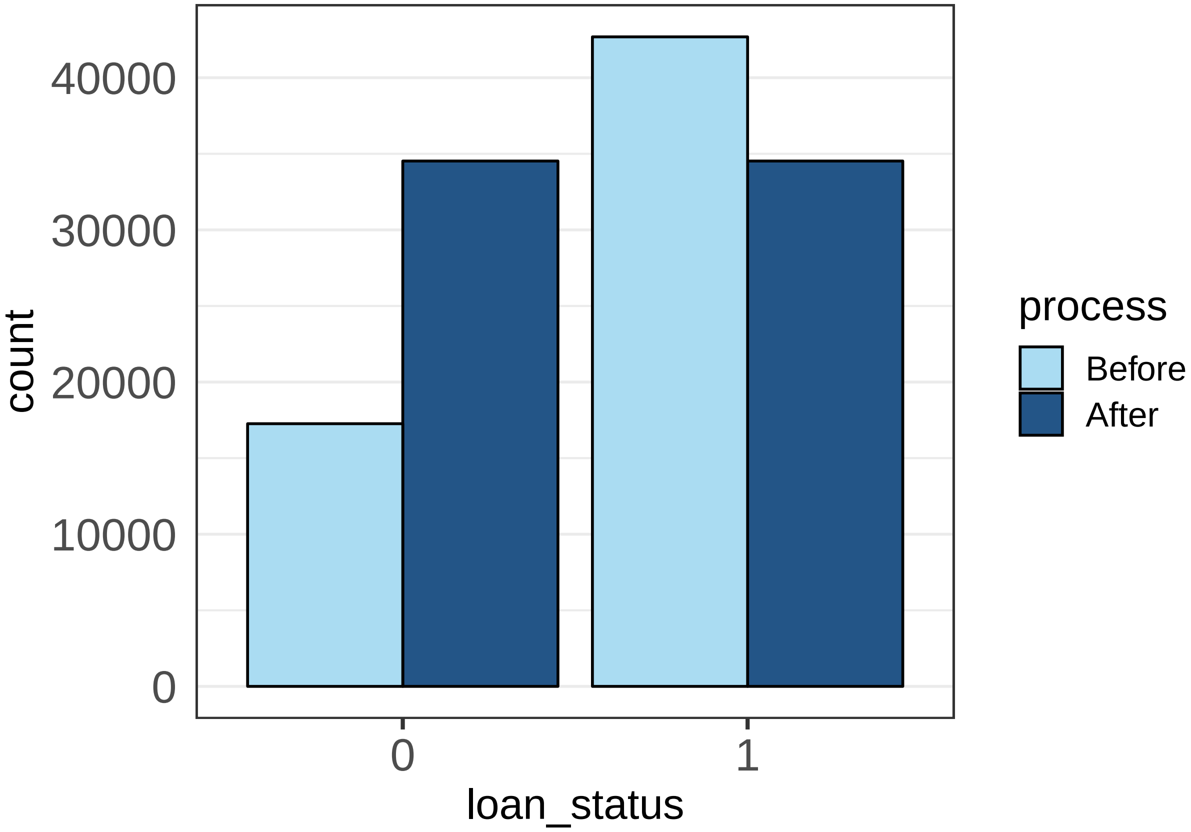
```
table(trainBalance$loan_status)
```

```
##
##     0     1
## 34524 34524
```

```
# compare balanced and unbalanced data
t <- data.frame(count = c(table(train$loan_status)[1], table(trainBalance$loan_status)[1], table(train$
```

```r
t$loan_status <- c("0", "0", "1", "1")
t$process <- factor(c("Before", "After", "Before", "After"), levels = c("Before", "After"))
theme_plot <- theme_bw(16) +
    theme(axis.text.y = element_text(size = rel(1.3)),
          axis.ticks.y = element_blank(),
          axis.text.x = element_text(size = rel(1.3)),
          panel.grid.major.x = element_blank(),
          panel.grid.major.y = element_line(size = 0.5),
          panel.grid.minor.x = element_blank())

ggplot(data = t, aes(x=loan_status, y=count, fill=process)) +
  geom_bar(stat="identity", color="black", position=position_dodge()) +
  theme_plot +
  scale_fill_manual(values=c('#aadcf2','#235587'))
```



```r
# Now the data is balanced. We save this data for future use.

write.csv(trainBalance,'trainBalance.csv', row.names = FALSE)
write.csv(test,'test_18.csv',row.names = FALSE)
```