User Complaints Mining

gor Baranov, Michael Parravani, Ariana Biagi, Grace Tsai

February 4, 2019

Preparing data

Load and clean the customer complaints data

The data should be downloaded from Kaggle's Consumer Complaints Database.

Loading and removing rows with no complaint narrative and unnecessary colimns:

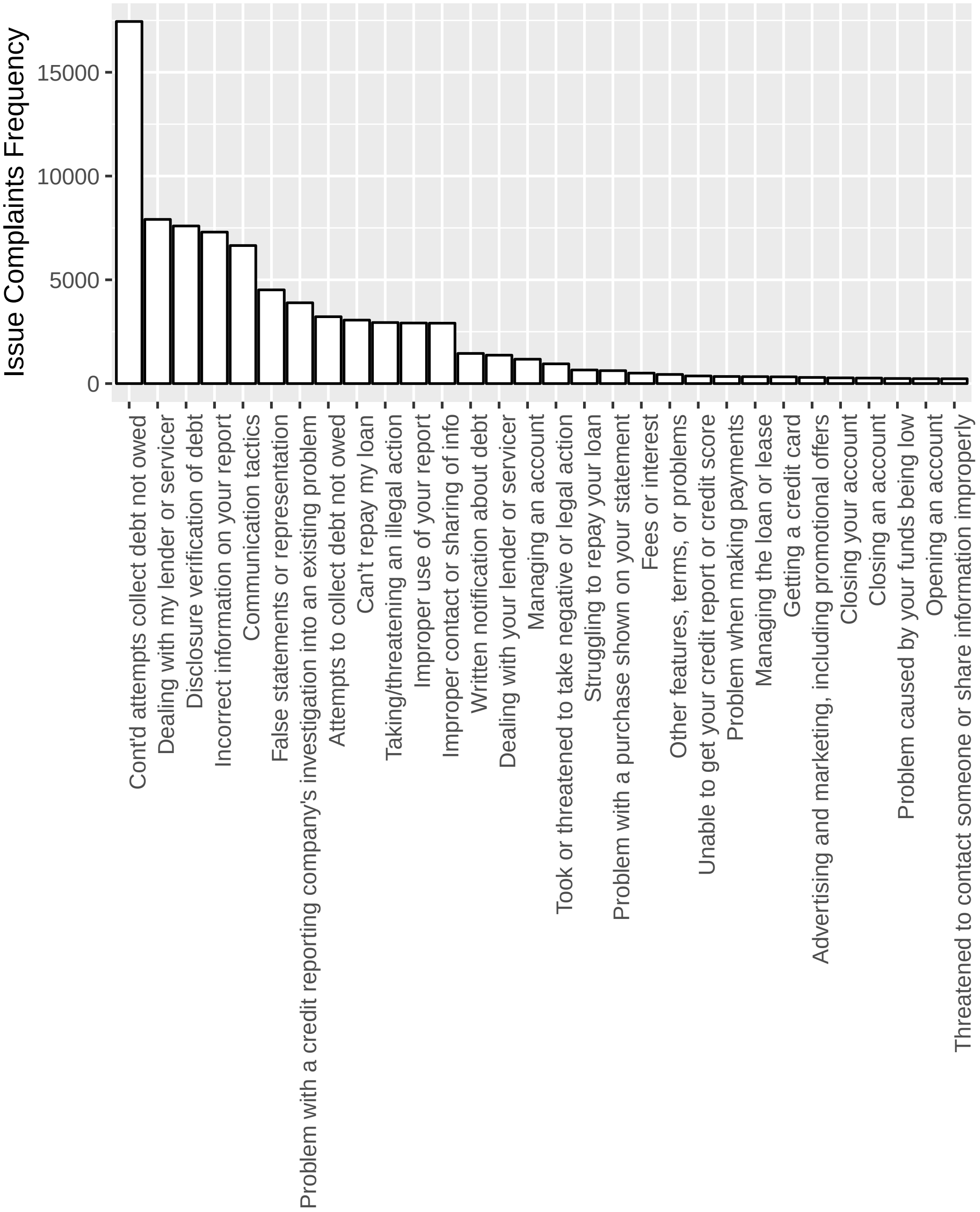
```
if(!file.exists("./user-complaints-mining/df.Rds")) {
  df <- read_csv(file="../data/Consumer_Complaints.csv.zip",col_names = TRUE)
  df <- df[,-c(1,7,9:18)]
  df <- df[!is.na(df[,"Consumer complaint narrative"]),] #199,970
  df <- df[!is.na(df[,"Company"]),] # no NA's
  df <- df[!is.na(df[,"Product"]),] # no NA's
  df <- df[!is.na(df[,"Issue"]),] # no NA's
  df <- df[!is.na(df[,"Sub-product"]),] # 147,788 total left
  df <- df[!is.na(df[,"Sub-issue"]),]   # 81,940 total left

  # Converting all but narrative columns to factors
  df$Product <- as.factor(df$Product)
  df$`Sub-product` <- as.factor(df$`Sub-product`)
  df$Issue <- as.factor(df$Issue)
  df$`Sub-issue` <- as.factor(df$`Sub-issue`)
  df$Company <- as.factor(df$Company)
  saveRDS(df, file = "./user-complaints-mining/df.Rds")
  gc()
} else {
  df <- readRDS("./user-complaints-mining/df.Rds")
}
df
```

```
## # A tibble: 81,940 x 6
##    Product  `Sub-product`   Issue  `Sub-issue`   `Consumer complai~ Company
##    <fct>    <fct>           <fct>  <fct>         <chr>              <fct>
##  1 Debt co~ Other (i.e. ph~ Discl~ Not given e~ This company refu~ The CBE~
##  2 Debt co~ Credit card     Impro~ Talked to a~ "This complaint i~ SQUARET~
##  3 Debt co~ Credit card     Takin~ Sued w/o pr~ "I am writing to ~ Selip &~
##  4 Debt co~ Other (i.e. ph~ Cont'~ Debt result~ My identity was s~ Southwe~
##  5 Student~ Federal studen~ Can't~ Can't get f~ "I was dropped fr~ AES/PHE~
##  6 Debt co~ Credit card     Discl~ Not given e~ The first communi~ Blatt, ~
##  7 Debt co~ Other (i.e. ph~ Commu~ Frequent or~ "My complaint is ~ AR Reso~
##  8 Debt co~ I do not know   False~ Attempted t~ In a clearance in~ SANTAND~
##  9 Student~ Non-federal st~ Can't~ Can't tempo~ XXXX University, ~ Navient~
## 10 Student~ Non-federal st~ Deali~ Received ba~ I had attended XX~ CITIZEN~
## # ... with 81,930 more rows
```

Feature engineering

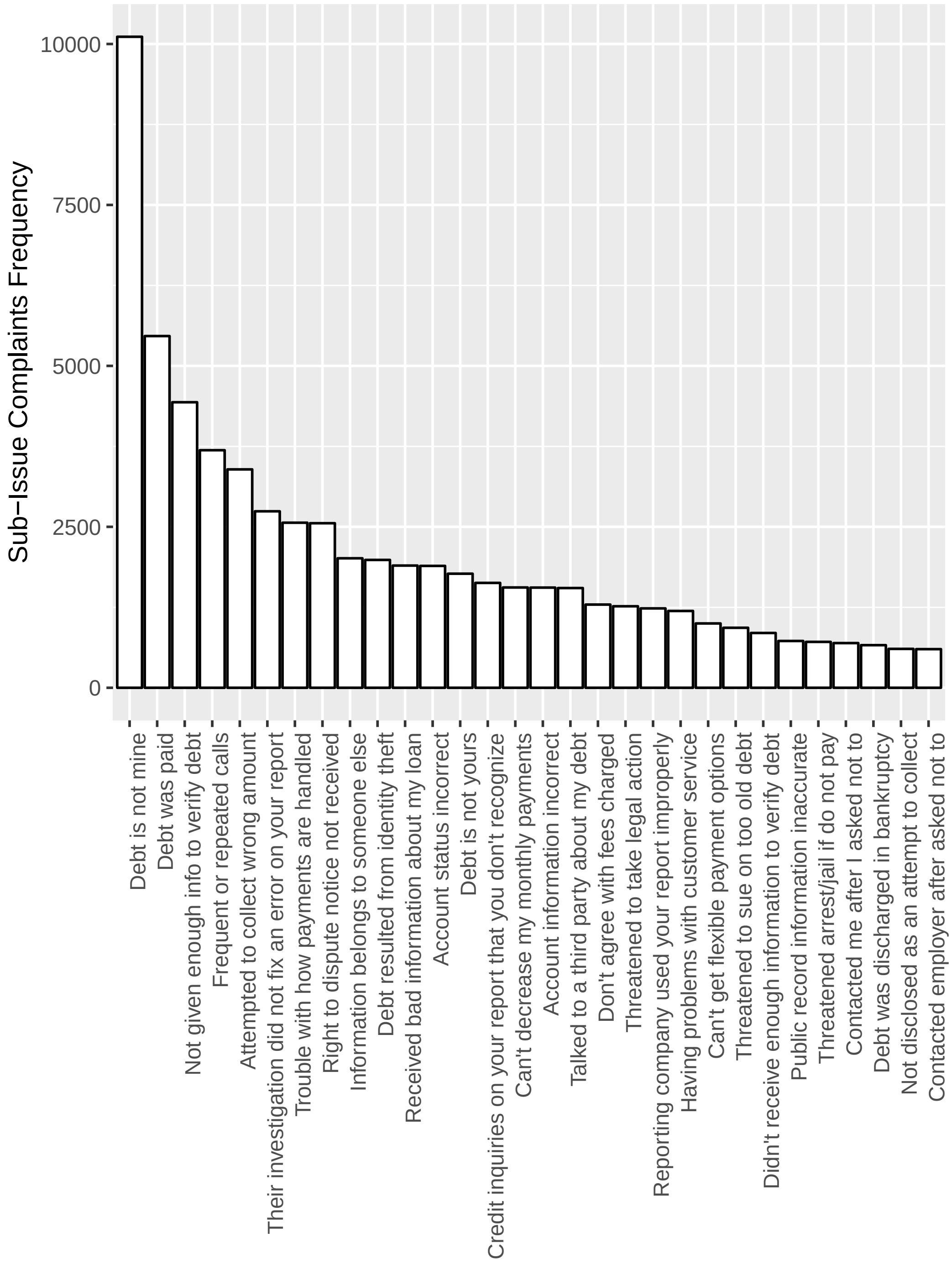
Distribution of the most frequent "Issue" complaints

```r
full_sorted_issues_list <- levels(fct_infreq(df$Issue))
saveRDS(full_sorted_issues_list, file = "./user-complaints-mining/full_sorted_issues_list.Rds")

ggplot() + aes(fct_infreq(df[df$Issue %in% full_sorted_issues_list[1:30],]$Issue))+
  geom_histogram(colour="black", fill="white", stat = "count")+
  ylab("Issue Complaints Frequency") + xlab("")+
  theme(axis.text.x = element_text(angle =90, hjust = 1))
```


Issue Complaints Frequency
15000
10000
5000
0
Cont'd attempts collect debt not owed
Dealing with my lender or servicer
Disclosure verification of debt
Incorrect information on your report
Communication tactics
False statements or representation
Problem with a credit reporting company's investigation into an existing problem
Attempts to collect debt not owed
Can't repay my loan
Taking/threatening an illegal action
Improper use of your report
Improper contact or sharing of info
Written notification about debt
Dealing with your lender or servicer
Managing an account
Took or threatened to take negative or legal action
Struggling to repay your loan
Problem with a purchase shown on your statement
Fees or interest
Other features, terms, or problems
Unable to get your credit report or credit score
Problem when making payments
Managing the loan or lease
Getting a credit card
Advertising and marketing, including promotional offers
Closing your account
Closing an account
Problem caused by your funds being low
Opening an account
Threatened to contact someone or share information improperly

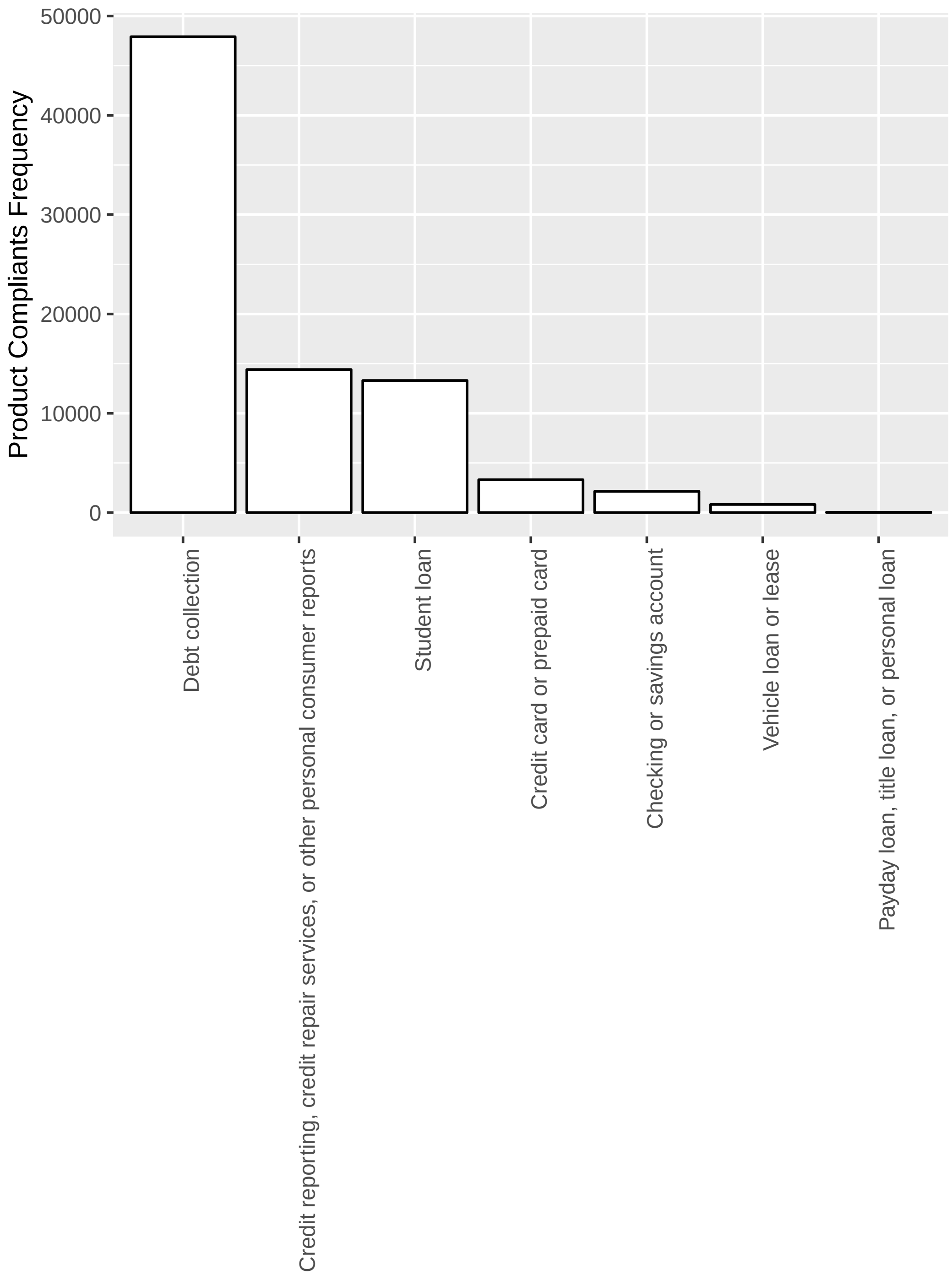
Distribution of the most frequent "Sub-issue" complaints

```
most_freq_subissues_list <- levels(fct_infreq(df$`Sub-issue`))[1:30]
ggplot() + aes(fct_infreq(df[df$`Sub-issue` %in% most_freq_subissues_list,]$`Sub-issue`))+
  geom_histogram(colour="black", fill="white", stat = "count")+
  ylab("Sub-Issue Complaints Frequency") + xlab("")+
  theme(axis.text.x = element_text(angle =90, hjust = 1))
```



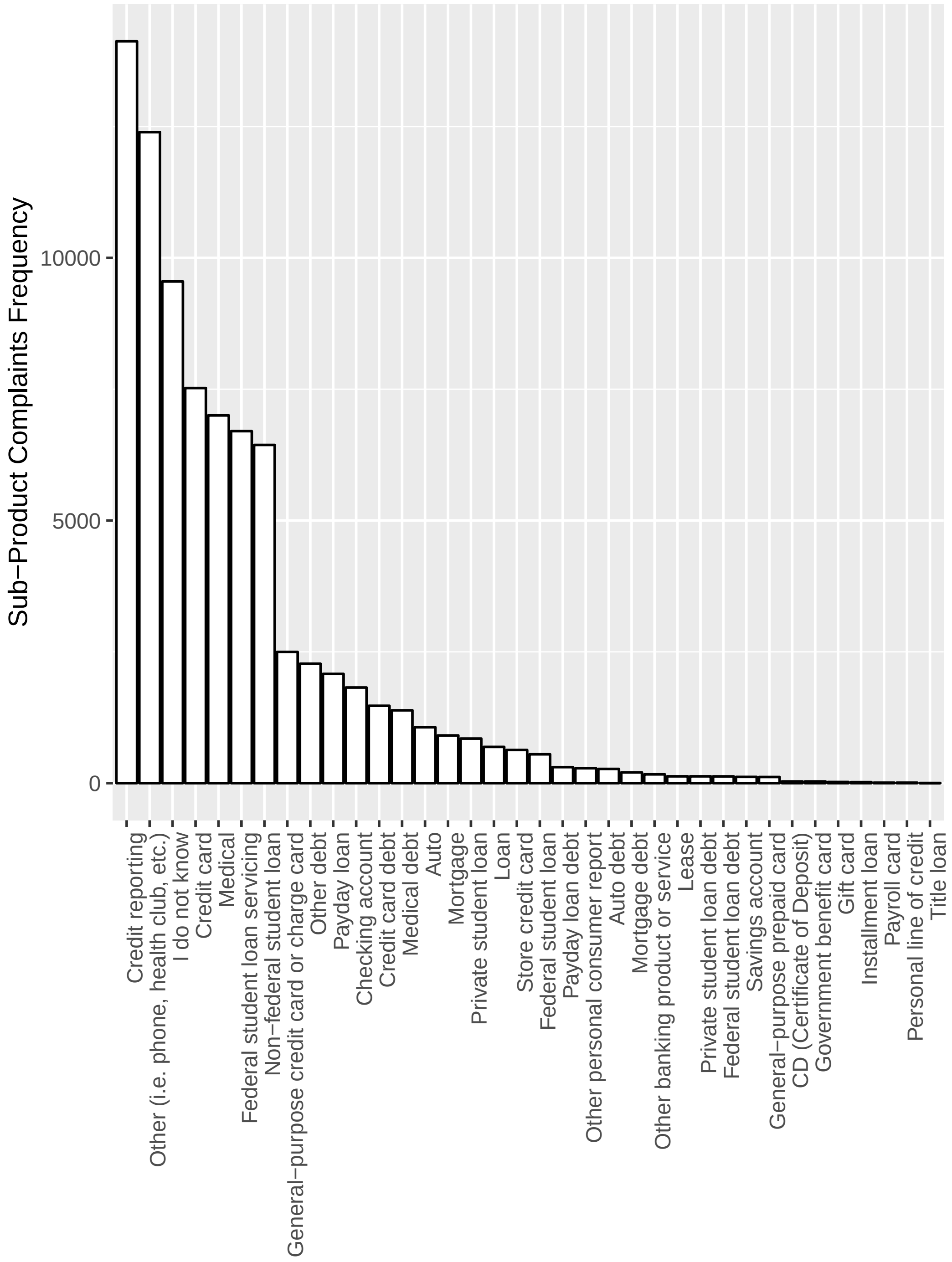
Distribution of the most frequent "Product" complaints

```
most_freq_product_list <- levels(fct_infreq(df$Product))[1:30]
ggplot() + aes(fct_infreq(df[df$Product %in% most_freq_product_list,]$Product))+
  geom_histogram(colour="black", fill="white", stat = "count")+
  ylab("Product Compliants Frequency") + xlab("")+
  theme(axis.text.x = element_text(angle =90, hjust = 1))
```



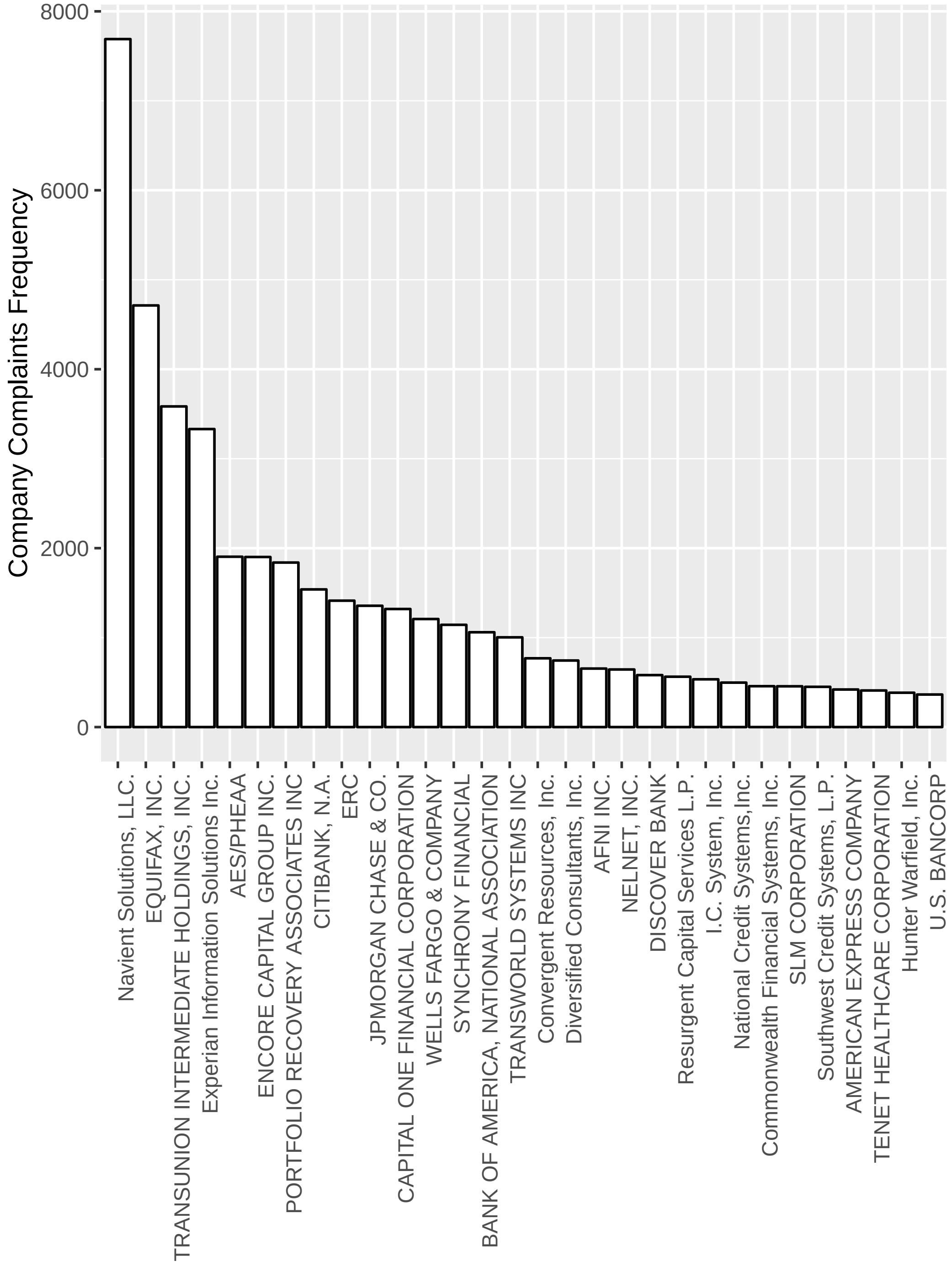
Distribution of the most frequent "Sub-product" complaints

```
most_freq_subproduct_list <- levels(fct_infreq(df$`Sub-product`))
ggplot() + aes(fct_infreq(df[df$`Sub-product` %in% most_freq_subproduct_list,]$`Sub-product`))+
  geom_histogram(colour="black", fill="white", stat = "count")+
  ylab("Sub-Product Complaints Frequency") + xlab("")+
  theme(axis.text.x = element_text(angle =90, hjust = 1))
```



Distribution of the most frequent "Company" complaints

```r
most_freq_company_list <- levels(fct_infreq(df$Company))[1:30]
ggplot() + aes(fct_infreq(df[df$Company %in% most_freq_company_list,]$Company))+
  geom_histogram(colour="black", fill="white", stat = "count")+
  ylab("Company Complaints Frequency") + xlab("")+
  theme(axis.text.x = element_text(angle =90, hjust = 1))
```



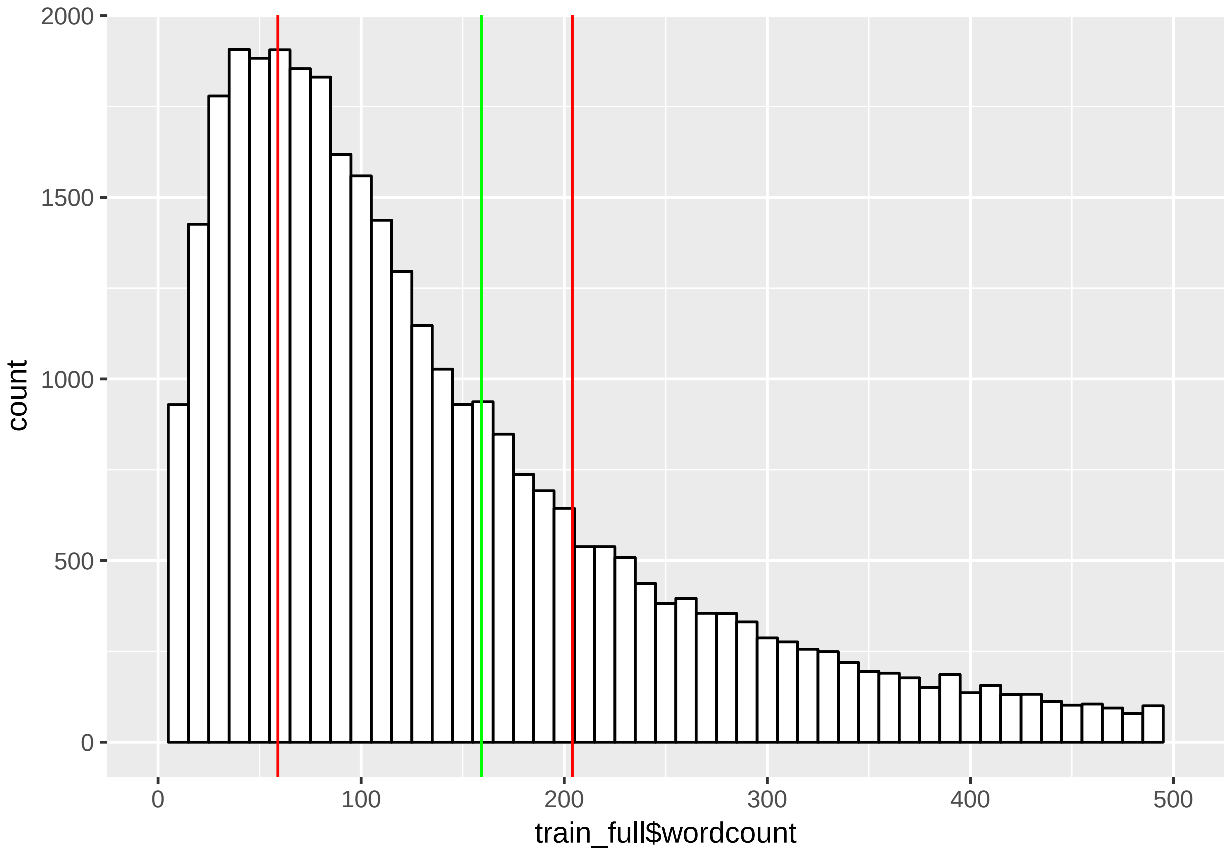
Text Minig

Split data into test and train sets for Issue prediction

```r
issuesToPredict <- 12

df$issueId <- match(df$Issue, full_sorted_issues_list)
df_issues <- df[df$issueId <= issuesToPredict,]

set.seed(123)
sample = sample.split(df_issues$`Consumer complaint narrative`, SplitRatio = .5)
train_full = subset(df_issues, sample == TRUE)
train_full$oid <- c(1:nrow(train_full))
test  = subset(df_issues, sample == FALSE)
```

Distribution of the text word count

```r
library(ngram)

train_full$wordcount <- sapply(train_full$"Consumer complaint narrative", wordcount)
stats <- summary(train_full$wordcount)

ggplot() + aes(train_full$wordcount)+ geom_histogram(binwidth=10, colour="black", fill="white")+
  geom_vline(xintercept=stats[["Mean"]], color="green")+
  geom_vline(xintercept=c(stats[["1st Qu."]],stats[["3rd Qu."]]), color="red")+
  scale_x_continuous(limits = c(0, 500))
```



```r
train <- train_full[(train_full$wordcount >= stats[["1st Qu."]] & train_full$wordcount <= stats[["3rd Qu
train <- sample_n(train, 15000)
train$id <- c(1:nrow(train))
```

Word analysis

Building a corpus, which is a collection of text documents VectorSource specifies that the source is character vectors.After that, the corpus needs a couple of transformations, including changing letters to lower case, removing punctuations/numbers and removing stop words. The general English stop-word list is tailored by adding some words specific to the documents in question.

```r
if(!file.exists("./user-complaints-mining/myCorpus.Rds")) {
  myCorpus <- Corpus(VectorSource(train$`Consumer complaint narrative`))
  myCorpus <- tm_map(myCorpus, removePunctuation)
  myCorpus <- tm_map(myCorpus, removeNumbers)
  myCorpus <- tm_map(myCorpus, tolower)
  myStopwords <- c(stopwords(language="en", source="smart"),
                  "xx", "xxxx", "xxxxxxxxxxxx", "xxxxxxxx",
                  "told", "well", "month", "year"
                   )
  myCorpus <- tm_map(myCorpus, removeWords, myStopwords)
  myCorpus <- tm_map(myCorpus, stripWhitespace)
  saveRDS(myCorpus, file = "./user-complaints-mining/myCorpus.Rds")
  gc()
} else {
  myCorpus <- readRDS("./user-complaints-mining/myCorpus.Rds")
}
print(myCorpus[1:5]$content)
```

```
## [1] " account credit report listed federal bond collection stating opened amount
current balance owed fraudulent collection open accounts loans time amounts documentation
produced prove collection valid belongs "
## [2] " private tuition answer student loans school loans serviced navient charged nt
pay monthly payments servicer gave options default loan loans doubled interest fees
period navient continues report credit report erroneous information closing date payment
made false illegally extending statue limitation account company predatory lender
collection tactics abusive illegal ca nt loan statement describes fees interest fees
interest added account exorbitant plead cfpb legal action consumer predator navient"
## [3] " filed voluntary chapter bankruptcy petition unable sell house time petition owed
consolidated student loans forbearance instructed make payments transferred loans navient
completion chapter plan bankruptcy discharge contacted navient resume payments find
navient added interest loans total amount owed lost job received incomebased repayment
plan interest added loans end owing owed forbearance began"
## [4] "national credit system attempting collect debt establised contract balance
balance negative item credit report file false witness violating fcra law status
limitation collect debt"
## [5] " representative total card nt identify repeatedly called reference debt owed
credit card account fraudulent belong representative repeatedly harassed oppressed
verbally abused threaten violence death arrest seizure property wage garnishment total
embarrassment pay debt owed total card violation fair debt collection practices act
minnesota debt collection laws representative countless times victim identity theft
account belong answers requests disregarded ssn dob "
```

Steming

```r
dictCorpus <- myCorpus
myCorpus <- tm_map(myCorpus, stemDocument)
```

```r
print(myCorpus[1:5]$content)
```

```
## [1] "account credit report list feder bond collect state open amount current balanc
owe fraudul collect open account loan time amount document produc prove collect valid
belong"
## [2] "privat tuition answer student loan school loan servic navient charg nt pay month
payment servic gave option default loan loan doubl interest fee period navient continu
report credit report erron inform close date payment made fals illeg extend statu limit
account compani predatori lender collect tactic abus illeg ca nt loan statement describ
fee interest fee interest ad account exorbit plead cfpb legal action consum predat
navient"
## [3] "file voluntari chapter bankruptci petit unabl sell hous time petit owe consolid
student loan forbear instruct make payment transfer loan navient complet chapter plan
bankruptci discharg contact navient resum payment find navient ad interest loan total
amount owe lost job receiv incomebas repay plan interest ad loan end owe owe forbear
began"
## [4] "nation credit system attempt collect debt establis contract balanc balanc negat
item credit report file fals wit violat fcra law status limit collect debt"
## [5] "repres total card nt identifi repeat call refer debt owe credit card account
fraudul belong repres repeat harass oppress verbal abus threaten violenc death arrest
seizur properti wage garnish total embarrass pay debt owe total card violat fair debt
collect practic act minnesota debt collect law repres countless time victim ident theft
account belong answer request disregard ssn dob"
```

Building a Document-Term Matrix

This operation is resource and time consuming. To avoid calculation, the pre-build myDtm object will be loaded from the file system. To recalculate it needs to be removed from the file system first.

```r
if(!file.exists("./user-complaints-mining/myDtm.Rds")) {
  myDtm <- TermDocumentMatrix(myCorpus, control = list(minWordLength = 1))
  rowTotals <- apply(myDtm , 1, sum) #Find the sum of words in each Document
  myDtm <- myDtm[rowTotals > 0, ] #remove all docs without words
  #myDtm <- removeSparseTerms(myDtm, sparse = 0.99)
  saveRDS(myDtm, file = "./user-complaints-mining/myDtm.Rds")
  gc()
} else {
  myDtm <- readRDS("./user-complaints-mining/myDtm.Rds")
}
inspect(myDtm)
```

```
## <<TermDocumentMatrix (terms: 12064, documents: 15000)>>
## Non-/sparse entries: 452086/180507914
## Sparsity           : 100%
## Maximal term length: 85
## Weighting          : term frequency (tf)
## Sample             :
##          Docs
## Terms     10418 12603 12838 1872 2847 4816 5725 5892 9315 9960
##   account     6     0     0    3    5    0    0    7    0    1
##   call        0     0     0    1    0    0    0    0    1    0
##   collect     0     2     8    2    4    8    4    2    0    0
##   compani     0     0     0    0    0    0    1    0    0    0
##   credit      2     1     0    4    3    0    2    2    3    4
##   debt        0     0     5    1    7    5    8    1    2    0
##   inform      0     0     1    1    2    1    1    1    2    0
##   loan        0     6     0    0    0    0    2    0    1    5
##   payment     9     2     0    0    2    0    1    0    0    0
##   report      3     0     0    9    2    0    3    4    3    4
```

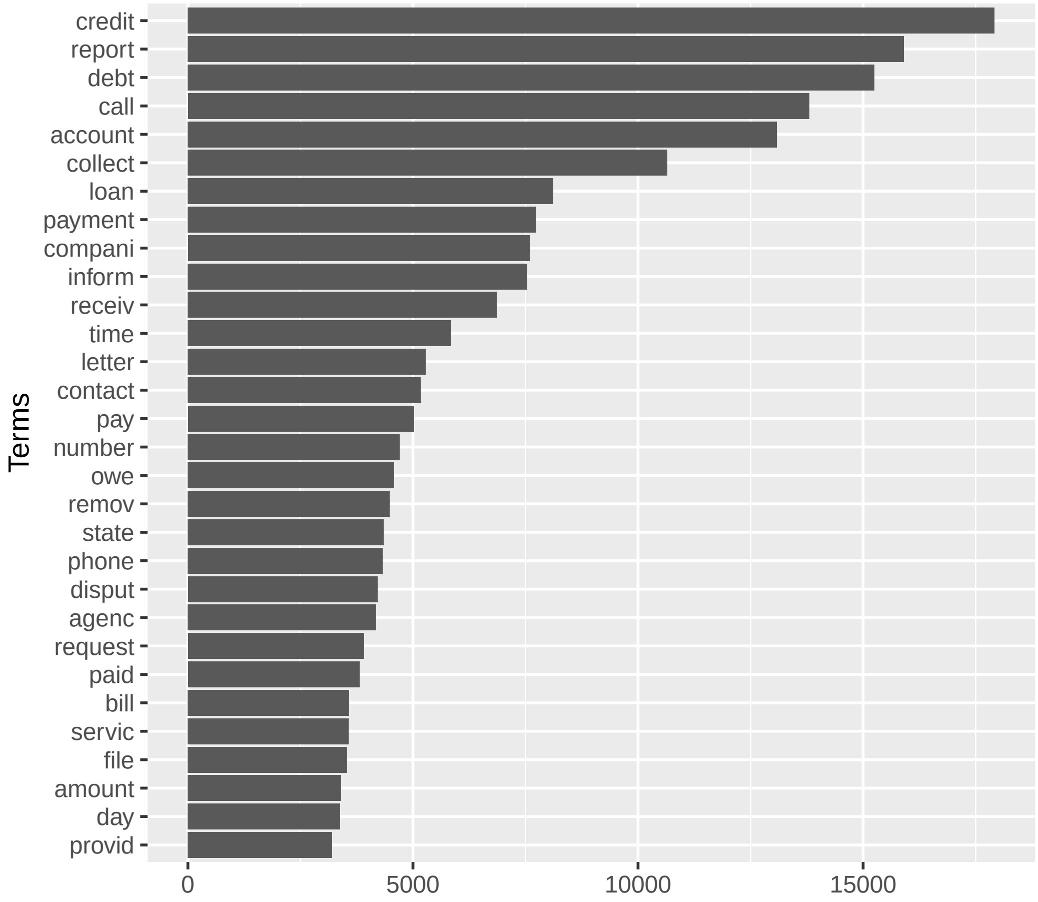


Figure 1: 30 Most Frequent Terms

Frequent Terms and Association

```r
freq.terms <- findFreqTerms(myDtm, lowfreq=5)
term.freq <- rowSums(as.matrix(myDtm))
term.freq <- subset(term.freq, term.freq >= 5)
```

```r
dfTerms <- data.frame(term = names(term.freq), freq = term.freq)
ggplot(dfTerms[order(-dfTerms$freq),][1:30,], aes(x = reorder(term, freq), y = freq)) +
  geom_bar(stat = "identity") + xlab("Terms") + ylab("") + coord_flip()
```

Which words are associated with term "loan"?

```r
findAssocs(myDtm, c('loan'), 0.3)
```

```
## $loan
##   student  navient   privat  payment interest
##      0.55     0.35     0.34     0.31     0.30
```

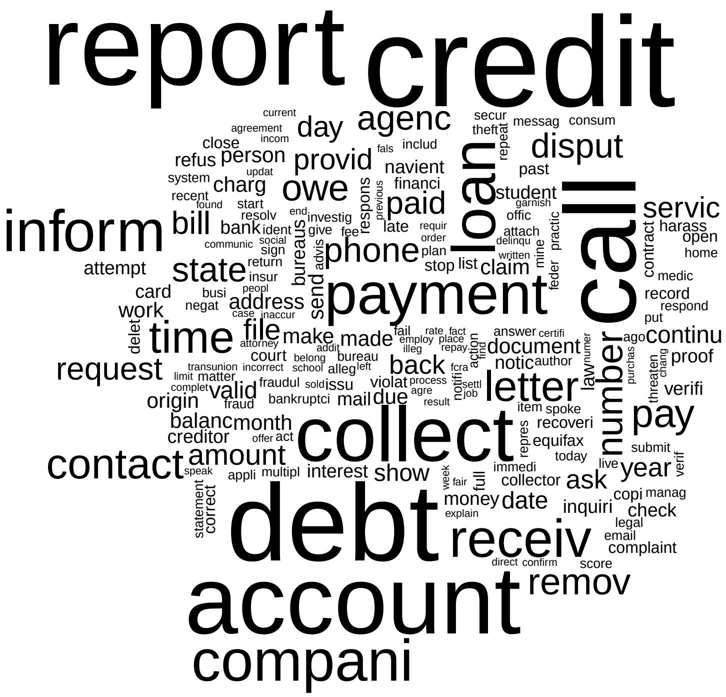


Figure 2: Words Cloud of Complaints

Building word cloud:

```
library(wordcloud)
m <- as.matrix(myDtm)
v <- sort(rowSums(m), decreasing=TRUE)
myNames <- names(v)
d <- data.frame(word=myNames, freq=v)
wordcloud(d$word, d$freq, min.freq=20, scale=c(4,.2), max.words = 200)
```

Building LDA model for the ‘train’ set of complaints

```
library(topicmodels)
dtm <- as.DocumentTermMatrix(myDtm)
numTopics <- 12 # 7, 10 or 30
topicsFileName <- paste("./user-complaints-mining/lda",numTopics,".Rds", sep = "")
ui = unique(dtm$i) #array of unique row ids in 'train' set
if(!file.exists(topicsFileName)) {
  dtm.new = dtm[ui,]
  train.lda <- LDA(dtm.new, k = numTopics) # identify topics
  saveRDS(train.lda, file = topicsFileName)
  gc()
} else {
  train.lda <- readRDS(topicsFileName)
}
(train.terms <- terms(train.lda, 10)) # first terms of every topic
```

```
##       Topic 1   Topic 2   Topic 3   Topic 4   Topic 5   Topic 6
##  [1,] "debt"    "report"  "account" "call"    "account" "report"
##  [2,] "call"    "credit"  "credit"  "collect" "call"    "credit"
##  [3,] "collect" "debt"    "collect" "loan"    "number"  "call"
##  [4,] "credit"  "contact" "payment" "receiv"  "compani" "collect"
##  [5,] "account" "day"     "letter"  "credit"  "credit"  "debt"
##  [6,] "pay"     "pay"     "receiv"  "payment" "debt"    "letter"
##  [7,] "compani" "account" "report"  "time"    "loan"    "account"
##  [8,] "inform"  "state"   "call"    "compani" "payment" "loan"
##  [9,] "receiv"  "collect" "time"    "debt"    "phone"   "time"
## [10,] "remov"   "time"    "request" "contact" "back"    "disput"
##       Topic 7   Topic 8   Topic 9   Topic 10  Topic 11  Topic 12
##  [1,] "account" "credit"  "debt"    "report"  "report"  "debt"
##  [2,] "credit"  "loan"    "collect" "credit"  "credit"  "report"
##  [3,] "payment" "call"    "account" "call"    "debt"    "paid"
##  [4,] "time"    "inform"  "letter"  "payment" "account" "account"
##  [5,] "debt"    "report"  "receiv"  "receiv"  "inform"  "loan"
##  [6,] "receiv"  "debt"    "number"  "remov"   "remov"   "contact"
##  [7,] "compani" "compani" "report"  "amount"  "collect" "continu"
##  [8,] "inform"  "payment" "contact" "account" "compani" "year"
##  [9,] "call"    "letter"  "inform"  "contact" "call"    "pay"
## [10,] "amount"  "pay"     "state"   "loan"    "agenc"   "owe"
```

Show complaints correlation to topics in 'train' set

```
train.topics <- topics(train.lda, 5, threshold=.005)
for (i in c(1,55,500,333)) {
  print (paste(" "))
  print (paste("Complaint# ", i))
  print(paste("Topic(s) found: ", train.topics[i]))
  print (train$`Consumer complaint narrative`[ui[i]])
}
```

```
## [1] " "
## [1] "Complaint# 1"
## [1] "Topic(s) found: 2"
## [1] "There is an account on my credit report listed as Federal Bond and Collection
stating it was opened in XXXX 2015 in the amount of {$2500.00} but the current balance
owed is {$4600.00}. This is a fraudulent collection as I did not open any accounts or
loans around that time and in those amounts. There needs to be documentation produced to
prove this collection is valid and belongs to me."
## [1] " "
## [1] "Complaint# 55"
## [1] "Topic(s) found: 11"
## [1] "I have a settlement that I am paying off through Attorney XXXX at XXXX, FL XXXX.
\n\nThe settlement is for {$4700.00} and payments began in XX/XX/2015. I have sent
written communication to this debt collector over the past 6 months requesting the
account balance. I have yet to receive any communication from this debt collector
regarding my account balance, however did receive a letter that my payment was late with
them one month."
## [1] " "
## [1] "Complaint# 500"
## [1] "Topic(s) found: 12"
## [1] "When I called the bank, someone said they were XXXX XXXX and this was a auto loan
inquiry. I said I had bought a car in over four years and was not looking for a car. I
got rerouted to a number that was not available. XXXX/XXXX/2016 XXXX has requested a copy
of your Credit Report Hide Details Business : Phone : Inquiry Date : XXXX XXXX XXXX XXXX
XXXX XXXX, TX XXXX XXXX XXXX/XXXX/2016Reported By : Experian"
## [1] " "
## [1] "Complaint# 333"
## [1] "Topic(s) found: 5"
## [1] "A company called \" Credit Collection Services '' and \" XXXX '' have sent
another letter concerning a debt for XXXX XXXX XXXX of {$25.00} to XXXX XXXX XXXX who
does not live at this address and also, is not even the correct middle name. The middle
name is that of my mother who died over XXXX years ago. I have filed a complaint in the
past for this same company and this is another continued attempt at fraud. I checked
multiple website and every single site says the same thing \" fraudulent. '' I contacted
the company and spoke with a \" XXXX XXXX ( not sure on spelling ) '' or \" XXXX XXXX or
XXXX '' she gave multiple names which made the call even more suspicious. I was intrigued
by this and asked if she had any more aliases she went by. She stated no. I informed them
I was filing a complaint. I also contacted XXXX and informed them of the situation
providing them with the File Number and PIN number on the document."
```

Topics related to most frequent issues

Function definitions

```r
prepareCorpus <- function(textArr) {
  myCorpus <- Corpus(VectorSource(textArr))
  myCorpus <- tm_map(myCorpus, removePunctuation)
  myCorpus <- tm_map(myCorpus, removeNumbers)
  myCorpus <- tm_map(myCorpus, tolower)
  myStopwords <- c(stopwords(language="en", source="smart"),
                  "xx", "xxxx", "xxxxxxxxxxxx", "xxxxxxxx")
  myCorpus <- tm_map(myCorpus, removeWords, myStopwords)
  myCorpus <- tm_map(myCorpus, stripWhitespace)

  return (myCorpus)
}

# Steming and reverse - long operation!
doSteming <- function(myCorpus) {
  #dictCorpus <- myCorpus
  myCorpus <- tm_map(myCorpus, stemDocument)
  #myCorpus <- tm_map(myCorpus, stemCompletion, dictionary=dictCorpus)
  return (myCorpus)
}

getBestTopicNums <- function(arr, threshold) {
  topNums <- sort(arr, index.return=TRUE, decreasing = TRUE)$ix
  arr <- arr[arr>threshold]
  return (topNums[1:length(arr)])
}
```

Form a matrix of topic probability vectors for each issue

```
issueTopicsProbMat <- matrix(NA,nrow = issuesToPredict, ncol = numTopics)

# for each issue
for (i in 1:issuesToPredict) {
  issueName <- full_sorted_issues_list[i]
  print(issueName)

  # matrix of issue topics probabilities
  issueids <- subset(train[,c("id")], train$Issue == issueName)
  uiIds <- match(issueids$id, ui)
  topicsProbabMatr <- train.lda@gamma[uiIds,]
  meanVec <- colMeans(topicsProbabMatr, na.rm = T)
  issueTopicsProbMat[i,] <- meanVec
}
```

```
## [1] "Cont'd attempts collect debt not owed"
## [1] "Dealing with my lender or servicer"
## [1] "Disclosure verification of debt"
## [1] "Incorrect information on your report"
## [1] "Communication tactics"
## [1] "False statements or representation"
## [1] "Problem with a credit reporting company's investigation into an existing problem"
## [1] "Attempts to collect debt not owed"
## [1] "Can't repay my loan"
## [1] "Taking/threatening an illegal action"
## [1] "Improper use of your report"
## [1] "Improper contact or sharing of info"
```

```
saveRDS(issueTopicsProbMat, file = "./user-complaints-mining/issueTopicsProbMat.Rds")
```

Identify issues of random 'test' complaints

```r
distInfo <- function(vec, mat) {
  retVec <- vector(length = nrow(mat))
  for (i in 1:nrow(mat)) {
    retVec[i] <- JSD(rbind(vec,mat[i,]), est.prob = "empirical")
  }
  retVec <- sort(retVec, index.return = TRUE)
  return (retVec)
}
# prepare test data
numOfTests <- 10000
numOfComplaintsToPrint <- 5
testThreshold =0.1
set.seed(123)
randomIds <- sample(nrow(test),numOfTests)
test.corpus <- prepareCorpus(test$`Consumer complaint narrative`[randomIds])
test.corpus <- doSteming(test.corpus)
test.dtm <- DocumentTermMatrix(test.corpus, control = list(minWordLength = 1))

#remove all docs with low number of words and correct supporting info
rowTotals  <- apply(test.dtm , 1, sum) #num words in each Document
test.dtm   <- test.dtm[rowTotals > 10,]
numOfTests <- test.dtm$nrow
randomIds  <- randomIds[rowTotals > 10]
test.topics <- posterior(train.lda,test.dtm)

origCls <- replicate(numOfTests, NA)
infrCls <- replicate(numOfTests, NA)
for (i in 1:numOfTests) {
  issuesFound <- distInfo (test.topics$topics[i,],issueTopicsProbMat)
  tops <- getBestTopicNums(test.topics$topics[i,], testThreshold)

  # orig issue id
  origIssueTxt <- test[[randomIds[i],"Issue"]][1]
  origIssueId <- match(origIssueTxt, full_sorted_issues_list)
  origCls[i] <-  origIssueId

  # inferred issue id
  if (origIssueId %in% issuesFound$ix[1:3]) {
    infrCls[i] <- origIssueId
  } else {
    infrCls[i] <- issuesFound$ix[1]
  }
  if (i <= numOfComplaintsToPrint) {
    print("----------")
    print (paste("Original Issue  : ", origIssueTxt))
    print (paste("Inferred Issue 1: ", full_sorted_issues_list[issuesFound$ix[1]]))
    print (paste("Inferred Issue 2: ", full_sorted_issues_list[issuesFound$ix[2]]))
    print (paste("Inferred Issue 3: ", full_sorted_issues_list[issuesFound$ix[3]]))
    print(paste("Complaint narrative:",test[[randomIds[i],"Consumer complaint narrative"]]))
  }
}
```

```
## [1] "----------"
## [1] "Original Issue : Incorrect information on your report"
## [1] "Inferred Issue 1: Dealing with my lender or servicer"
## [1] "Inferred Issue 2: Can't repay my loan"
## [1] "Inferred Issue 3: Incorrect information on your report"
## [1] "Complaint narrative: My issue is with FedLoan Servicing.  I missed payments
because at the time I was very young and did not have a general understanding of how all
student loans work. I applied for deferments and forbearance 's, but was never told or
educated in regards to how long each would normally last and when to reapply. Over the
past years, I have experienced job loss, financial hardship, and transitioning into XXXX
XXXX XXXX including deployments to XXXX XXXX and XXXX . I have taken steps to ensure my
financial responsibility moving forward. After finding out that my student loans were in
collections a few years ago while I was in XXXX , I quickly paid to get them out, but my
credit report still shows the closed student loans with late payment statuses. These
loans were transferred to XXXX afterwards but the old loans from FedLoan Servicing are
still showing on my credit report. I reached out to them through disputing through the
credit bureau 's and have also sent them a letter directly and received nothing back."
## [1] "----------"
## [1] "Original Issue : Incorrect information on your report"
## [1] "Inferred Issue 1: Problem with a credit reporting company's investigation into an
existing problem"
## [1] "Inferred Issue 2: Incorrect information on your report"
## [1] "Inferred Issue 3: Improper use of your report"
## [1] "Complaint narrative: I have disputed this account many times with all XXXX credit
bureus. This account has never been verifiable and is reporting different account
statuses and amounts on each one of my reports.  XXXX shows {$6500.00} and closed and
Transunion shows {$6600.00} and open. By law the FCRA states all information has to be
100 % verifiable and accurate. This account is not following the law. The account is with
XXXX XXXX XXXX XXXX"
## [1] "----------"
## [1] "Original Issue : Dealing with my lender or servicer"
## [1] "Inferred Issue 1: Improper use of your report"
## [1] "Inferred Issue 2: Dealing with my lender or servicer"
## [1] "Inferred Issue 3: Incorrect information on your report"
## [1] "Complaint narrative: I am a Victim of ID Theft and I am trying to improve my
credit. I went through a lot and I owe {$50000.00} without certification and I am XXXX
years of age. I have asked for consolidation of my school loans, I applied for a credit
card with XXXX XXXX. I was turned down due to negative items on my credit file, when I
checked I have XXXX negative accounts without any explanation that everything was
consolidated. Now the negative items are there from XXXX plus negative items of school
loans still from XXXX XXXX."
## [1] "----------"
## [1] "Original Issue : Improper use of your report"
## [1] "Inferred Issue 1: Dealing with my lender or servicer"
## [1] "Inferred Issue 2: Can't repay my loan"
## [1] "Inferred Issue 3: False statements or representation"
## [1] "Complaint narrative: Re : XXXX XXXX Account ending in XXXX This credit card had a
credit protection service on it, in the event of job loss. When I called in to report my
job loss, the creditor said they no longer honor that program. Subsequently, this is a
violation of basic contracts law as there was no meeting of the minds at inception,
violation of the UCC section 2-201, as I had made regular payments and had a reasonable
expectation of the minimum payments to be made during the time of my job loss.
Additionally, I find a violation of section 1-304 of the UCC as the creditor had no good
faith, and entered the contract with unclean hands."
```

```
## [1] "----------"
## [1] "Original Issue : False statements or representation"
## [1] "Inferred Issue 1: Disclosure verification of debt"
## [1] "Inferred Issue 2: Attempts to collect debt not owed"
## [1] "Inferred Issue 3: Cont'd attempts collect debt not owed"
## [1] "Complaint narrative: XXXX XXXX, XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX, NC
XXXX XXXX XXXX EOSCCA XXXX XXXX XXXX XXXX, MA XXXX Re : Acct # XXXX Re : Acct # XXXX To
Whom It May Concern : This letter is in response to your recent claim regarding account #
XXXX, XXXX which you claim I owe {$2400.00}. I am requesting validation, made pursuant to
the Fair Debt Collection Practices Act. Please note that I am requesting \" validation ''
; that is competent evidence bearing my signature, showing that I have ( or ever had )
some contractual obligation to pay you. My reason for disputing is : my contract was
canceled. Please also be aware that any negative mark found on my credit reports (
including XXXX, XXXX and XXXX ) from your company or any company that you represent, for
a debt that I do n't owe, is a violation of the Fair Debt Collection Practices Act ;
therefore if you can not validate the debt, you must request that all credit reporting
agencies delete the entry. Pending the outcome of my investigation of any evidence that
you submit, you are instructed to take no action that could be detrimental to any of my
credit reports. Failure to respond within 30 days of receipt of this certified letter
will result in small claims legal action against your company at my local venue. I will
be seeking a minimum of {$5000.00} in damages for : 1 ) Defamation 2 ) Negligent
Enablement of Identity Fraud 3 ) Violation of the Fair Debt Collection Practices Act (
including but not limited to Section 807-8 ) You will be required to appear in a court
venue local to me, in order to formally defend yourself. \nFor the purposes of 15 USC
1692 et seq., this Notice has the same effect as a dispute to the validity of the alleged
debt and a dispute to the validity of your claims. Please Note : This notice is an
attempt to correct your records, and any information received from you will be collected
as evidence should any further action be necessary. This is a request for information
only, and is not a statement, election, or waiver of status. P.S. Please be aware that
dependent upon your response, I may be detailing any potential issues with your company
via an online public press release, including documentation of any potential small claims
action."
```

Issue prediction accuracy

```
(tbl <- table(origCls,infrCls))
```

```
##         infrCls
## origCls    1    2    3    4    5    6    7    8    9   10   11   12
##      1  1088   52   55  107  263   53  168   55   53  178   50  100
##      2    11  893    7   50   39   24   13    7    0   28    7   13
##      3    27   19  470   51   86   81   93   33   12   57   24   11
##      4    26   35   24  718    9   41    8   60   17   11   24    6
##      5    30   33   26    4  617   94    6   15   20   44    5   28
##      6     5   16   49   26   74  254   28   39   38   53   17   31
##      7    11   21   19   11    3   11  356   32    9    5   14    4
##      8     5   10   14   15   35   40   31  205   10   38    7   22
##      9     3    7    0    2    8   19    0    0  342    8    2    5
##      10   16   13   27    7    5   19    5   15    8  231   10    3
##      11   17    4    5    6    5   12    7   14    1    9  246    9
##      12   12    5   23    9    0   21   11   14    8    9    6  240
```

```
(sum(diag(tbl)))/numOfTests
```

```
## [1] 0.6162221
```

Prepare data for use in Shiny App

Save random test complaints from the set of 'issuesToPredict'

```r
complaintsToSave <- 2000 # will be about half of that after cleaning
set.seed(1234)
randomIds <- sample(nrow(test),complaintsToSave)
sampleComplaints <- test[randomIds, c("issueId","Issue","Consumer complaint narrative")]
sampleComplaints$wordcount <- sapply(sampleComplaints$"Consumer complaint narrative", wordcount)
stats <- summary(sampleComplaints$wordcount)
sampleComplaints <- sampleComplaints[
  (sampleComplaints$wordcount >= stats[["1st Qu."]] &
     sampleComplaints$wordcount <= stats[["3rd Qu."]]),]
saveRDS(df, file = "./user-complaints-mining/sampleComplaints.Rds")
```