

# 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 columns:

```
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 complaint n~ Company
##   <fct>   <fct>         <fct> <fct>      <chr>                <fct>
## 1 Debt co~ Other (i.e. ~ Disc~ Not given ~ This company refuses ~ The CB~
## 2 Debt co~ Credit card Impr~ Talked to ~ "This complaint is in~ SQUARE~
## 3 Debt co~ Credit card Taki~ Sued w/o p~ "I am writing to requ~ Selip ~
## 4 Debt co~ Other (i.e. ~ Cont~ Debt resul~ My identity was stole~ Southw~
## 5 Student~ Federal stud~ Can'~ Can't get ~ "I was dropped from m~ AES/PH~
## 6 Debt co~ Credit card Disc~ Not given ~ The first communicati~ Blatt,~
## 7 Debt co~ Other (i.e. ~ Comm~ Frequent o~ "My complaint is n't ~ AR Res~
## 8 Debt co~ I do not know Fals~ Attempted ~ In a clearance interv~ SANTAN~
## 9 Student~ Non-federal ~ Can'~ Can't temp~ XXXX University, XXXX~ Navien~
## 10 Student~ Non-federal ~ Deal~ Received b~ I had attended XXXX a~ CITIZE~
## # ... with 81,930 more rows
```

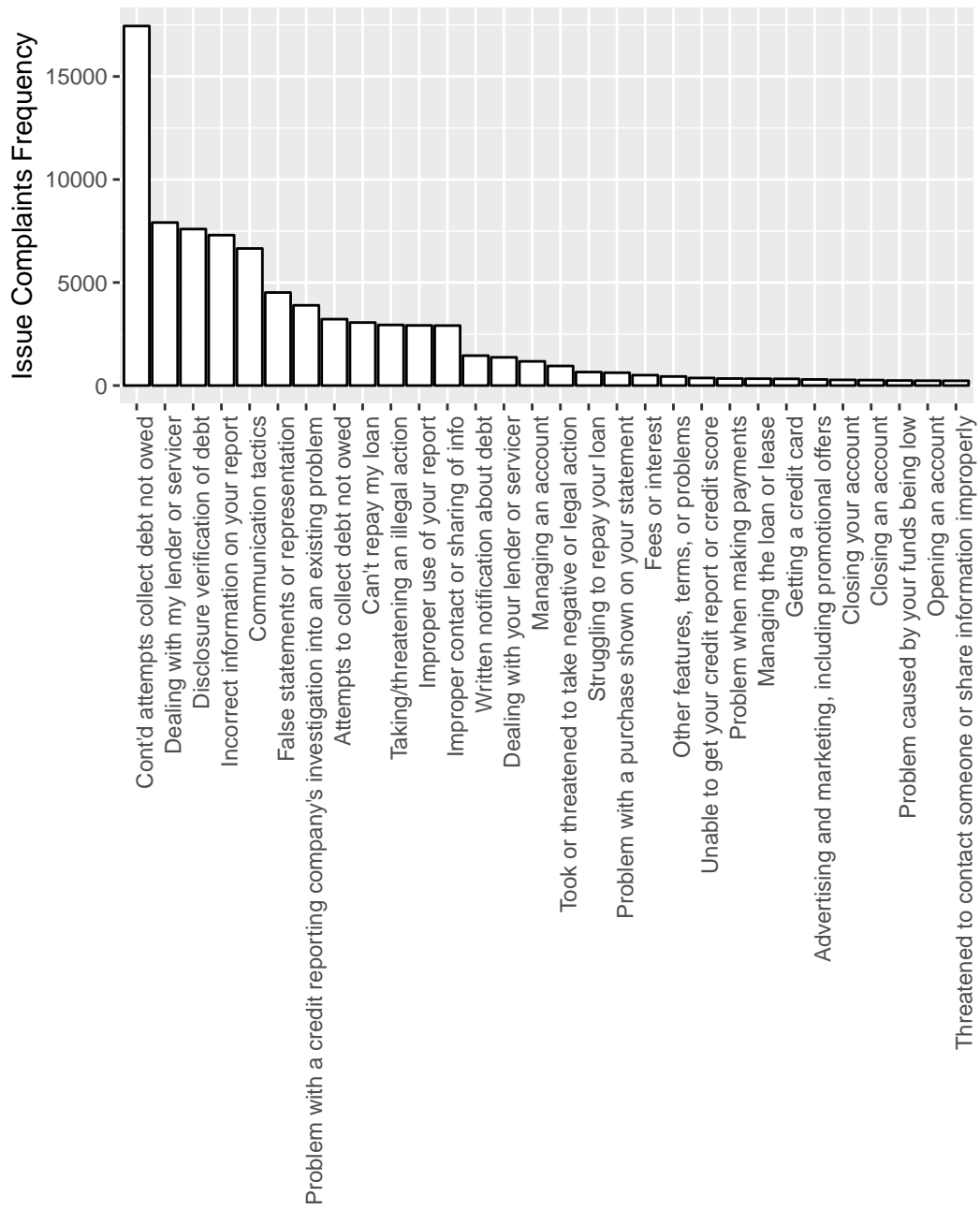
Converting all but narrative columns to factors:

```
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)
```

## Feature engineering

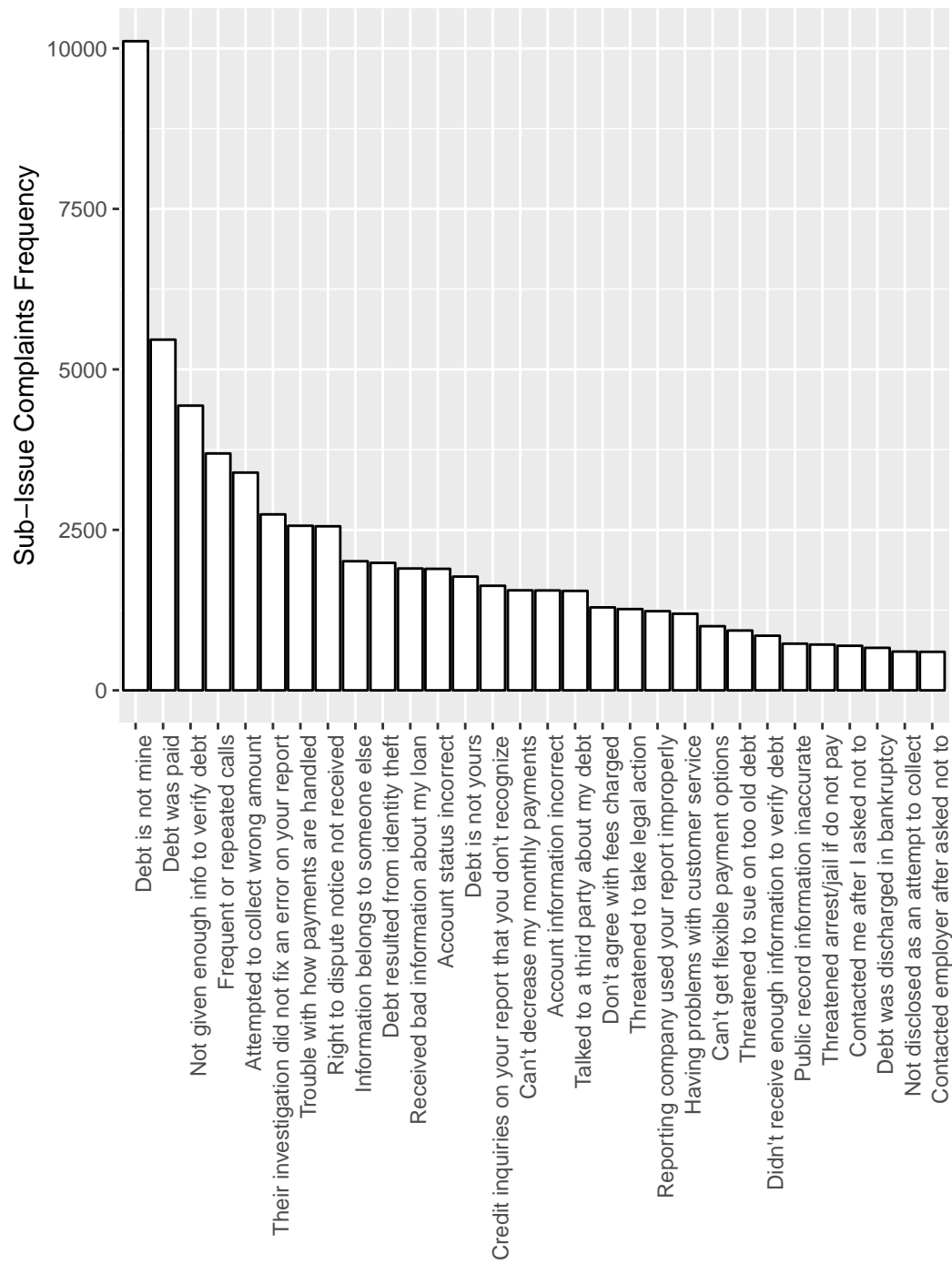
### Distribution of the most frequent “Issue” complaints

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



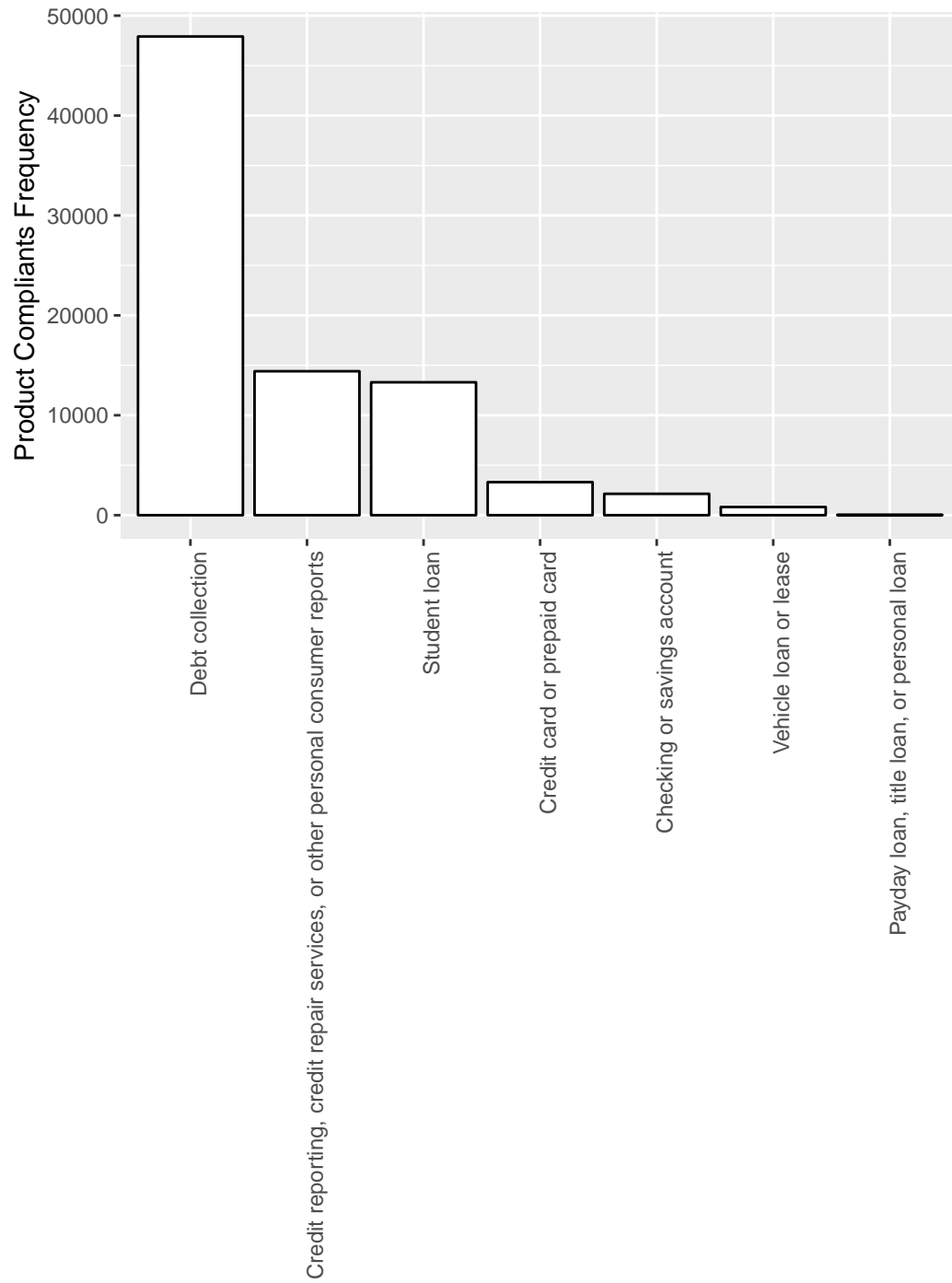
## Distribution of the most frequent “Sub-issue” complaints

```
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))
```



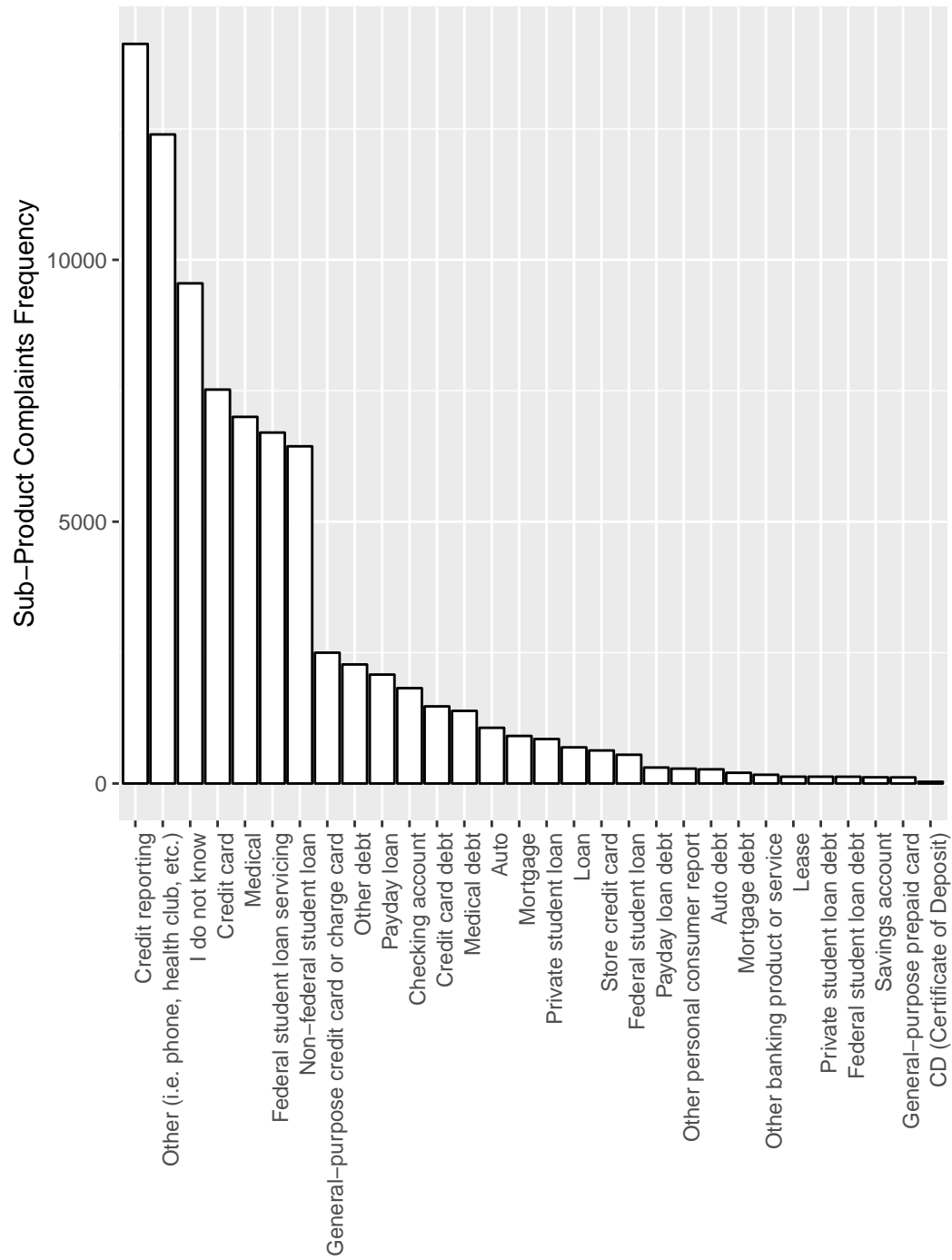
## Distribution of the most frequent “Product” complaints

```
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))
```



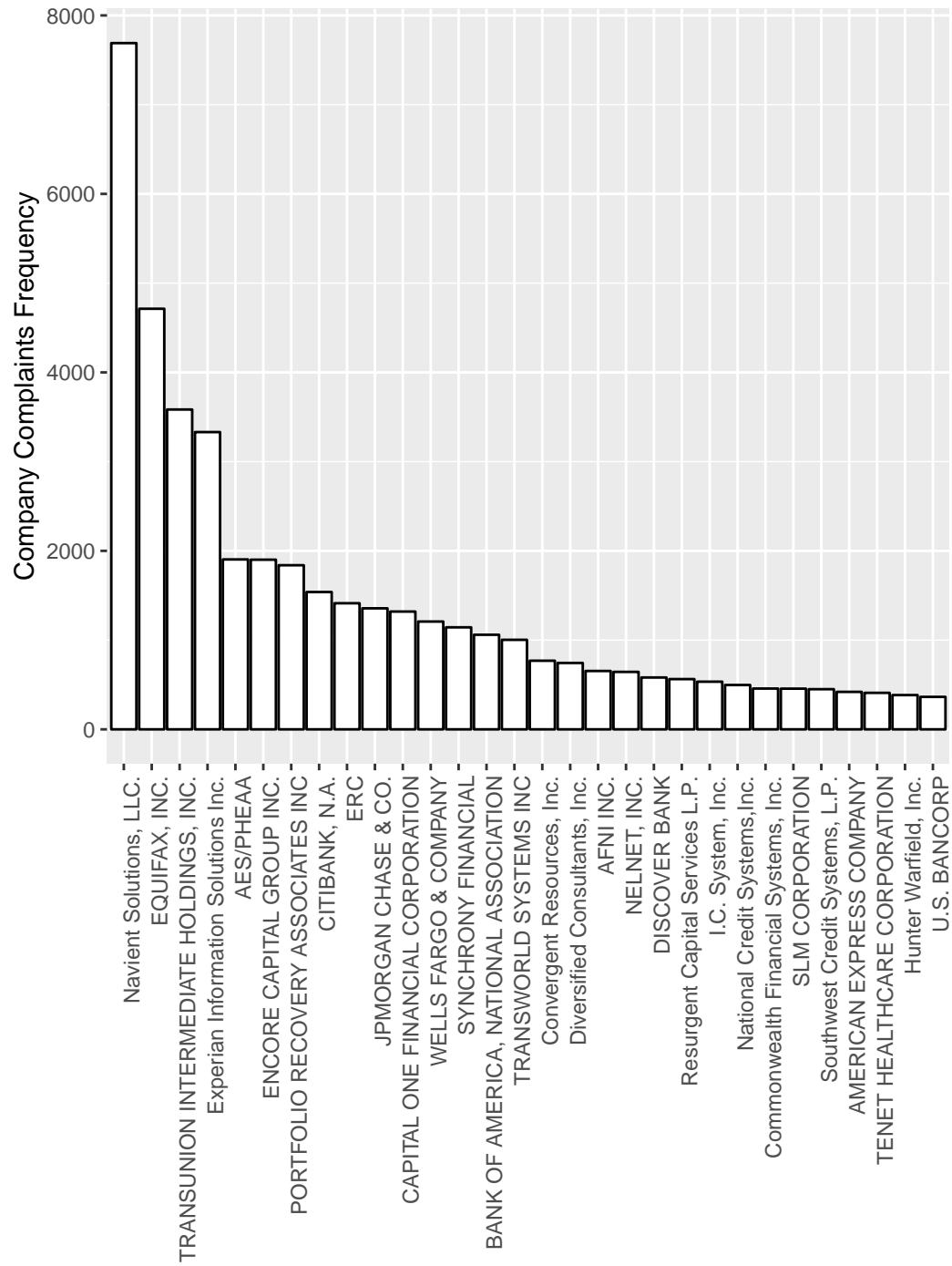
## Distribution of the most frequent “Sub-product” complaints

```
most_freq_subproduct_list <- levels(fct_infreq(df$`Sub-product`))[1:30]
ggplot() + aes(fct_infreq(df[df$`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

```
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

```
set.seed(123)
sample = sample.split(df$`Consumer complaint narrative`, SplitRatio = .5)
train = subset(df, sample == TRUE)
test = subset(df, sample == FALSE)
```

### 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.

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

```
## [1] " company refuses provide verification validation debt fdcpa debt mine"
## [2] " writing request assistance deceptive practices collection lawfirm appears
tactics violating consumer protection law debt collection practices depriving consumers
rights dispute received notice company day contacted offices instructed memo dated
instruct contact plaintiff attorney court instructions provided contacted plaintiff
attorney phone faxed letter disputing debt letter company responded letter dated sending
bill due date requested bill showing balance back made payment back disputing amount owed
disputing charges wrote back company faxed dispute letter continue dispute amount owed
company response furnished information disputing owe disputing balance inaccurate needed
proof charges activity account paragraph letter disputing amount send letter letter
company disputing balance requesting documents company responded letter communication
received letter copy default judgment filed court clerk office indicating failed respond
judgment facts judgment firm served original judgment attaching dispute letters showing
responded instructed office occasions plaintiff failed respond dispute furnish
information providedand unable obtain proof original debt credible legal procedure settle
debts utilized unfaithful dirty tactics violated rights court committed perjury law
filing false documents court defaulted judgment failed respond fact responded failed
furnish proof court house clerk office told company notify offices contact company told
court clerk office respond summons clerks office granted default judgement based false
information respond summons filed false affirmation clerk office"
## [3] " communication received debt collector court summons delivered mother laws home
```



received summons hand week summons stated alledgedly owed money debt collector advisement dispute debt days demand debt collector validate debt attempted demand validation debt online research certified letter days ago post office told today approximately certified letter attempted delivered days mailed accepted debt collector today withoyut knowledge information sufficient form opinion truth accuracy claim based deny generally specifically claim debt collector"

## [4] " attended forced loan attend school loan interest rate make payments attended school enrolled told classmates received reduction balance curious criteria receive reduction"

## [5] "years ago harassed issue asked send application form signed understood charged extra pet damages paid additional month pet covered damages wanted photos damages property manager told walk left apartment thing scam money contacted requested information"

## Steming

```
dictCorpus <- myCorpus
myCorpus <- tm_map(myCorpus, stemDocument)
print(myCorpus[1:5]$content)
```

## [1] "compani refus provid verif valid debt fdcpa debt mine"

## [2] "write request assist decept practic collect lawfirm appear tactic violat consum protect law debt collect practic depriv consum right disput receiv notic compani day contact offic instruct memo date instruct contact plaintiff attorney court instruct provid contact plaintiff attorney phone fax letter disput debt letter compani respond letter date send bill due date request bill show balanc back made payment back disput amount owe disput charg wrote back compani fax disput letter continu disput amount owe compani respons furnish inform disput owe disput balanc inaccur need proof charg activ account paragraph letter disput amount send letter letter compani disput balanc request document compani respond letter communic receiv letter copi default judgment file court clerk offic indic fail respond judgment fact judgment firm serv origin judgment attach disput letter show respond instruct offic occas plaintiff fail respond disput furnish inform providedand unabl obtain proof origin debt credibl legal procedur settl debt util unfaith dirti tactic violat right court commit perjuri law file fals document court default judgment fail respond fact respond fail furnish proof court hous clerk offic told compani notifi offic contact compani told court clerk offic respond summon clerk offic grant default judgement base fals inform respond summon file fals affirm clerk offic"

## [3] "communic receiv debt collector court summon deliv mother law home receiv summon hand week summon state alledg owe money debt collector advis disput debt day demand debt collector valid debt attempt demand valid debt onlin research certifi letter day ago post offic told today approxim certifi letter attempt deliv day mail accept debt collector today withoyut knowledg inform suffici form opinion truth accuraci claim base deni general specif claim debt collector"

## [4] "attend forc loan attend school loan interest rate make payment attend school enrol told classmat receiv reduct balanc curious criteria receiv reduct"

## [5] "year ago harass issu ask send applic form sign understood charg extra pet damag paid addit month pet cover damag want photo damag properti manag told walk left apart thing scam money contact request inform"

## Building a Document-Term Matrix

This operation requires 64GB of RAM. 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.

```
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  
  saveRDS(myDtm, file = "./user-complaints-mining/myDtm.Rds")  
  gc()  
} else {  
  myDtm <- readRDS("./user-complaints-mining/myDtm.Rds")  
}  
inspect(myDtm)
```

```
## <<TermDocumentMatrix (terms: 27440, documents: 40970)>>  
## Non-/sparse entries: 1585496/1122631304  
## Sparsity          : 100%  
## Maximal term length: 124  
## Weighting          : term frequency (tf)  
## Sample            :  
##  
##      Docs  
## Terms      11425 11904 31399 31930 33039 34137 35502 35599 36932 9698  
## account    26    21    60    62    18    64     5     8     0     6  
## call        3    10     2     0    18     0     3     0     2     7  
## collect     0     0     5     0     0     4     0     6     0    21  
## credit      3     1     0     0     0     2    13     2     2     0  
## debt        0     0     1     1     0     1     0     0     3    68  
## inform     11     5     8     8     6     8    10    73     5     9  
## loan        5     0     2     0    11     0     2     0     3     0  
## payment     2    12    27    27    24    27     7     0     0     1  
## receiv      4     2     0     0    25     0     1     0     8    10  
## report      1     0    25    27     1    28    26     8     0     0
```