CapstoneTechReport

rdearing

December 2, 2016

R Markdown

Important Notes - Many difficulties were incurred in an effort to knit this document. In order to reduce knit time and the chance of failures, many sections of clear and simple code are placed outside of R chunks with their results and/or warnings included as text.

```
#load libraries
library(RCurl)
```

```
## Loading required package: bitops
```

```
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
```

```
library(ggplot2)
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

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

```
library(e1071)
library(SnowballC)
library(RTextTools)
```

```
## Loading required package: SparseM
```

```
##
## Attaching package: 'SparseM'
```

```
## The following object is masked from 'package:base':
##
##     backsolve
```

```
##
## Attaching package: 'RTextTools'
```

```
## The following objects are masked from 'package:SnowballC':
##
```

```
##     getStemLanguages, wordStem
```

```
library(caret)
```

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

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

load each data set

n2005 <- getURL("http://www.nserc-crsng.gc.ca/opendata/NSERC_GRT_FYR2005_AWARD.csv") n2006 <- getURL("http://www.nserc-crsng.gc.ca/opendata/NSERC_GRT_FYR2006_AWARD.csv") n2007 <- getURL("http://www.nserc-crsng.gc.ca/opendata/NSERC_GRT_FYR2007_AWARD.csv") n2008 <- getURL("http://www.nserc-crsng.gc.ca/opendata/NSERC_GRT_FYR2008_AWARD.csv") n2009 <- getURL("http://www.nserc-crsng.gc.ca/opendata/NSERC_GRT_FYR2009_AWARD.csv") n2010 <- getURL("http://www.nserc-crsng.gc.ca/opendata/NSERC_GRT_FYR2010_AWARD.csv") n2011 <- getURL("http://www.nserc-crsng.gc.ca/opendata/NSERC_GRT_FYR2011_AWARD.csv") n2012 <- getURL("http://www.nserc-crsng.gc.ca/opendata/NSERC_GRT_FYR2012_AWARD.csv") n2013 <- getURL("http://www.nserc-crsng.gc.ca/opendata/NSERC_GRT_FYR2013_AWARD.csv") n2014 <- getURL("http://www.nserc-crsng.gc.ca/opendata/NSERC_GRT_FYR2014_AWARD.csv")

read csv and set NA values

n2005NA <- read.csv(text = n2005, header = TRUE, sep = ",", na.strings = c("No summary - Aucun sommaire", "")) n2006NA <- read.csv(text = n2006, header = TRUE, sep = ",", na.strings = c("No summary - Aucun sommaire", "")) n2007NA <- read.csv(text = n2007, header = TRUE, sep = ",", na.strings = c("No summary - Aucun sommaire", "")) n2008NA <- read.csv(text = n2008, header = TRUE, sep = ",", na.strings = c("No summary - Aucun sommaire", "")) n2009NA <- read.csv(text = n2009, header = TRUE, sep = ",", na.strings = c("No summary - Aucun sommaire", "")) n2010NA <- read.csv(text = n2010, header = TRUE, sep = ",", na.strings = c("No summary - Aucun sommaire", "")) n2011NA <- read.csv(text = n2011, header = TRUE, sep = ",", na.strings = c("No summary - Aucun sommaire", "")) n2012NA <- read.csv(text = n2012, header = TRUE, sep = ",", na.strings = c("No summary - Aucun sommaire", "")) n2013NA <- read.csv(text = n2013, header = TRUE, sep = ",", na.strings = c("No summary - Aucun sommaire", "")) n2014NA <- read.csv(text = n2014, header = TRUE, sep = ",", na.strings = c("No summary - Aucun sommaire", ""))

bind data sets

totalNSERC <- bind_rows(n2005NA, n2006NA, n2007NA, n2008NA, n2009NA, n2010NA, n2011NA, n2012NA, n2013NA, n2014NA)

– throws mulitple warnings: Warning in bind_rows_(x, .id), Unequal factor levels

– due to empty cells (no type) lining up with cells containing data at binding sites

– not a concern

select fields of interest

NSERC_selected <- select(totalNSERC, Institution.Établissement, FiscalYear.Exercice.financier,AwardAmount, ApplicationSummary)

check for NAs

sum(is.na(NSERC_selected$Institution.Établissement) == TRUE) – returns 0

sum(is.na(NSERC_selected$FiscalYear.Exercice.financier) == TRUE) – returns 0

sum(is.na(NSERC_selected$AwardAmount) == TRUE) – returns 0

sum(is.na(NSERC_selected$ApplicationSummary) == TRUE) – returns 150592

filter NA values

NSERC_selected_filtered <- na.omit(NSERC_selected)

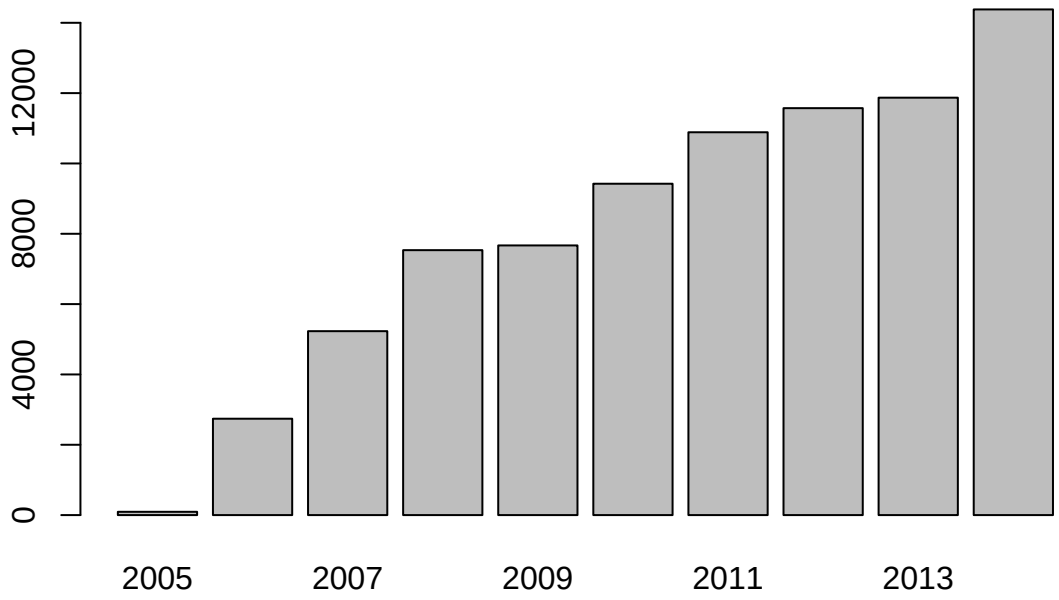
filter out applications in French.

NSERC_selected_filtered <- NSERC_selected_filtered[grep("é",NSERC_selected_filtered$ApplicationSummary, invert= TRUE),]

save this file locally to prevent reloading large data from the web

save(NSERC_selected_filtered, file = "NSERC.rdata")

```
#load selected and filtered data from saved file if neccessary
load("../Documents/NSERC.rdata")

#explore the data
barplot(table(NSERC_selected_filtered$FiscalYear.Exercice.financier)) #summaries increasing
```




```
sum(as.numeric(NSERC_selected_filtered$AwardAmount)) #over 3.7 billion in funds
```

```
## [1] 3767917412
```

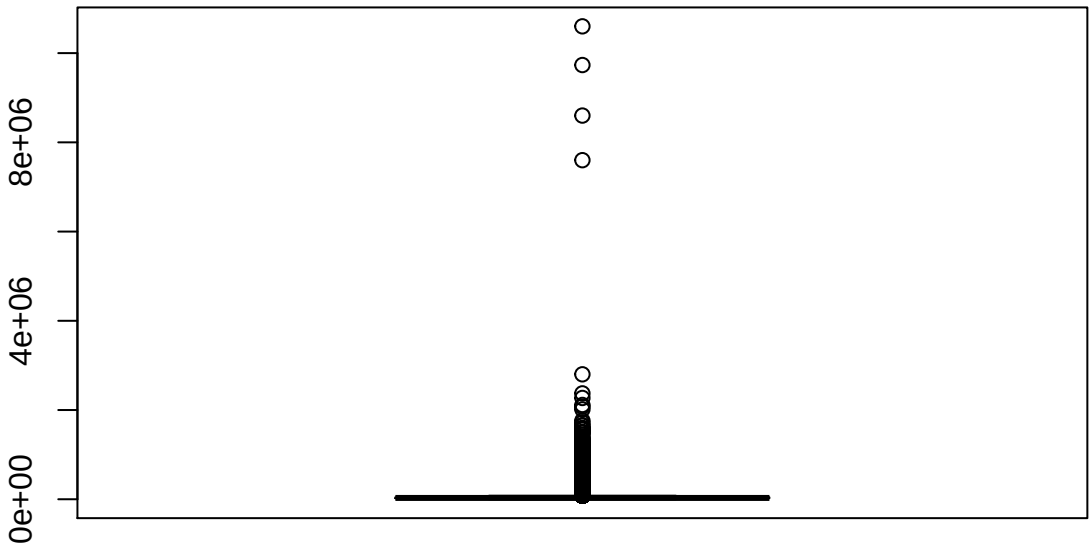
```
summary(NSERC_selected_filtered$AwardAmount)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##        7    22000    29000    46290    45000 10600000
```

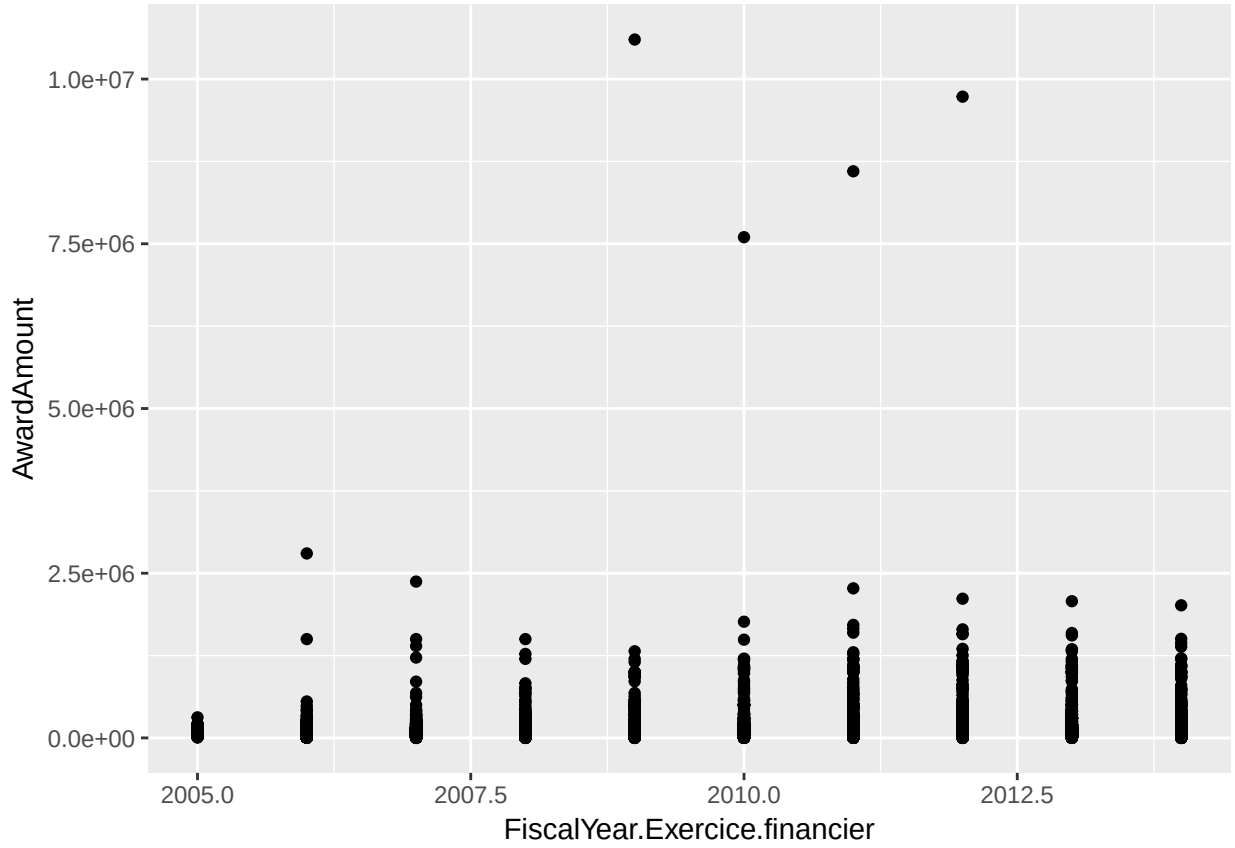
```
sd(NSERC_selected_filtered$AwardAmount) #seems too high, check for outliers
```

```
## [1] 94074.74
```

```
boxplot(NSERC_selected_filtered$AwardAmount)
```




```
ggplot(NSERC_selected_filtered, aes(x=FiscalYear.Exercice.financier, y=AwardAmount )) + geom_point() #f
```



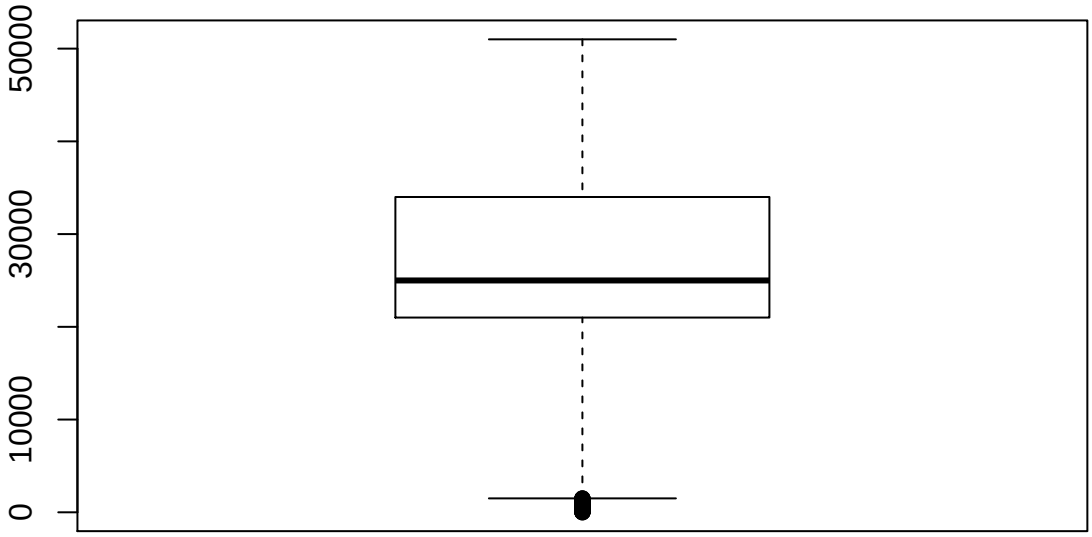

```
#Guidance from Tamer: explore the distribution to see where the values really start to take off,
#then cut off the top x percentile of the data
q <- quantile(NSERC_selected_filtered$AwardAmount, +
                c(0,1/10,2/10,3/10,4/10,5/10,6/10,7/10,8/10,9/10,1))
q #somewhere between 80 and 90
```

```
##         0%        10%        20%        30%        40%        50%
##        7.0    17000.0    21000.0    24000.0    25000.0    29000.0
##        60%        70%        80%        90%       100%
##    34000.0    40000.0    51060.6    85233.4 10600000.0
```

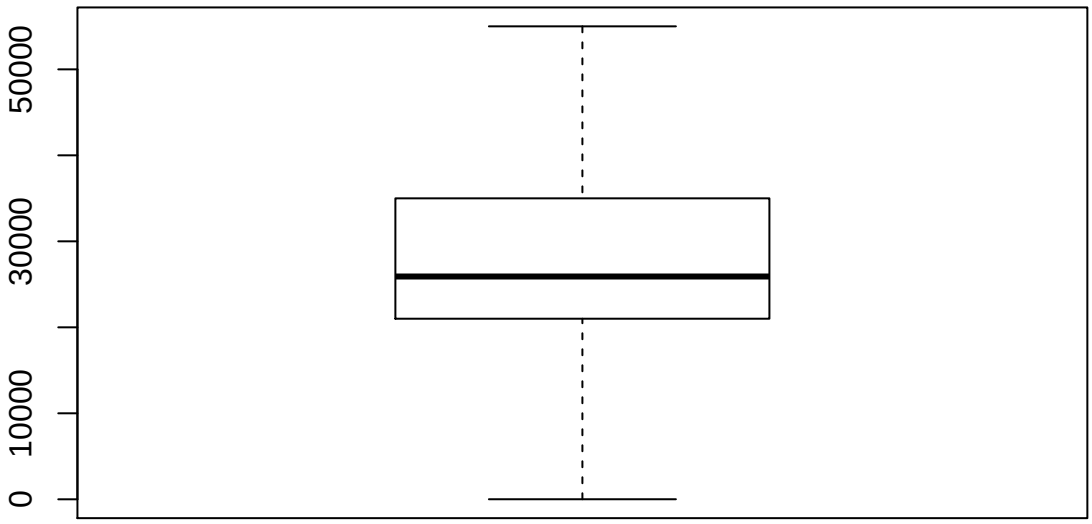
```
q <- quantile(NSERC_selected_filtered$AwardAmount, +
                c(8/10,81/100,82/100,83/100,84/100,85/100,86/100,87/100,88/100,89/100))
q #realtively large jump from 80 to 81, again from 85 to 86. Test from 80-85 (51,000-63,000)
```

```
##     80%     81%     82%     83%     84%     85%     86%     87%     88%
## 51060.6 54000.0 55550.0 57790.0 60000.0 62100.0 67000.0 70000.0 75000.0
##     89%
## 80000.0
```

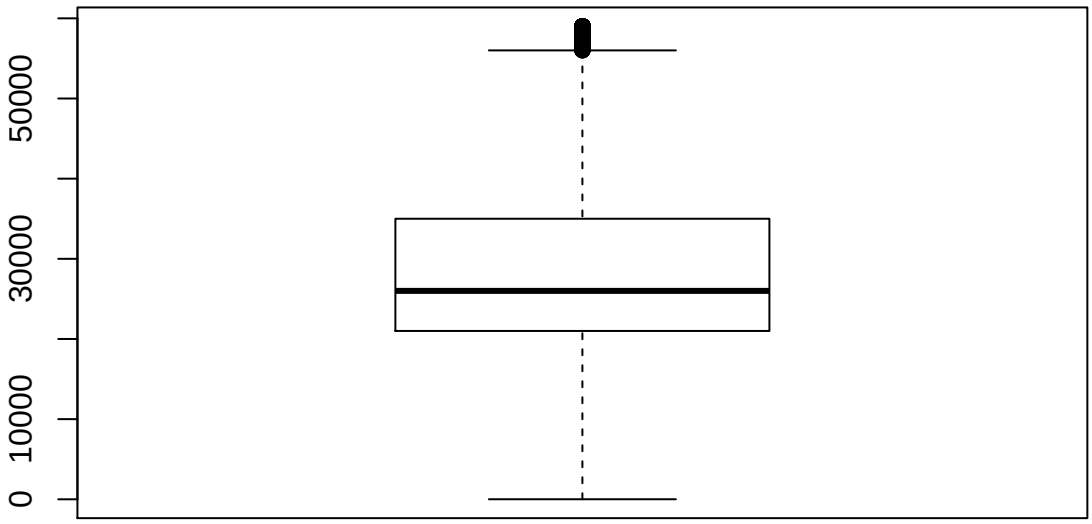
```
NSERC_trimmed1 <- subset(NSERC_selected_filtered, AwardAmount <= 51000)
NSERC_trimmed2 <- subset(NSERC_selected_filtered, AwardAmount <= 55000)
NSERC_trimmed3 <- subset(NSERC_selected_filtered, AwardAmount <= 59000)
NSERC_trimmed4 <- subset(NSERC_selected_filtered, AwardAmount <= 63000)

boxplot(NSERC_trimmed1$AwardAmount) #small outliers
```



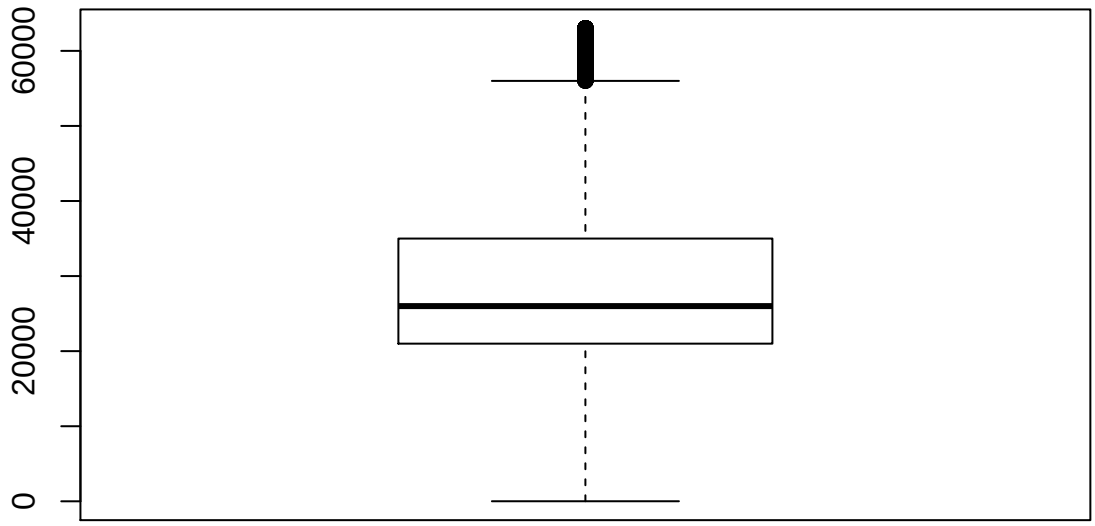

```
boxplot(NSERC_trimmed2$AwardAmount) #no outliers present
```



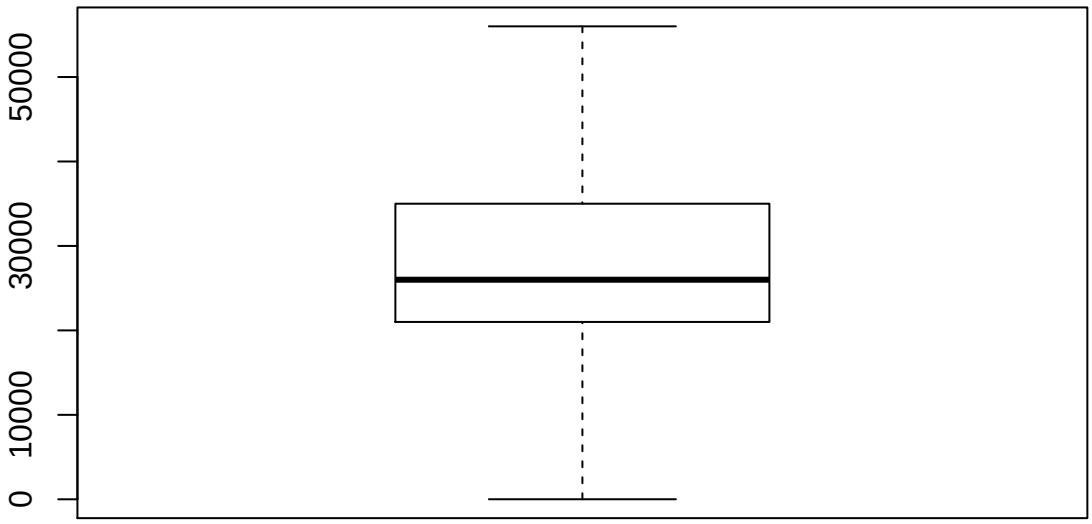

```
boxplot(NSERC_trimmed3$AwardAmount) #large outliers
```



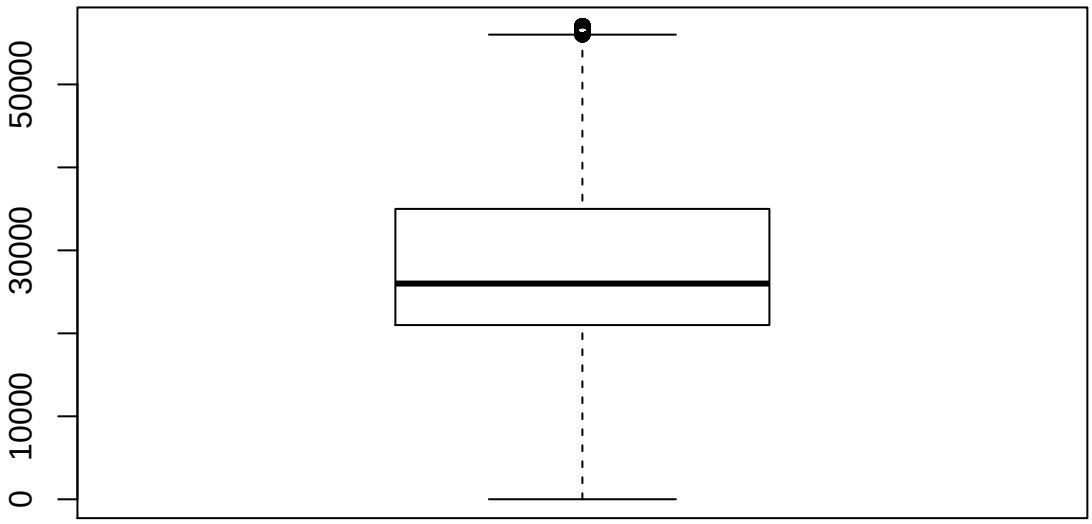

```
boxplot(NSERC_trimmed4$AwardAmount) #large outliers
```



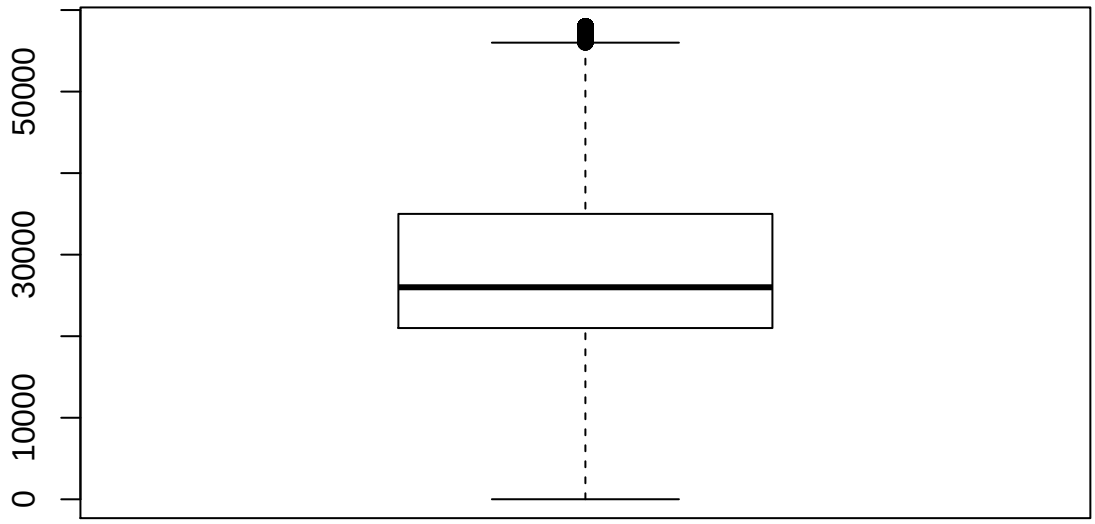

```
#somewhere between 55 and 59 thousand

NSERC_trimmed5 <- subset(NSERC_selected_filtered, AwardAmount <= 56000)
NSERC_trimmed6 <- subset(NSERC_selected_filtered, AwardAmount <= 57000)
NSERC_trimmed7 <- subset(NSERC_selected_filtered, AwardAmount <= 58000)

boxplot(NSERC_trimmed5$AwardAmount) #no outliers
```




```
boxplot(NSERC_trimmed6$AwardAmount) #large outliers
```




```
boxplot(NSERC_trimmed7$AwardAmount) #large outliers
```




```
#go with 56000 cut off (about 82% of data)
NSERC_trimmed <- subset(NSERC_selected_filtered, AwardAmount <= 56000)

#explore the new distribution
sum(as.numeric(NSERC_trimmed$AwardAmount)) #over 1.9 billion in funds
```

```
## [1] 1902836821
```

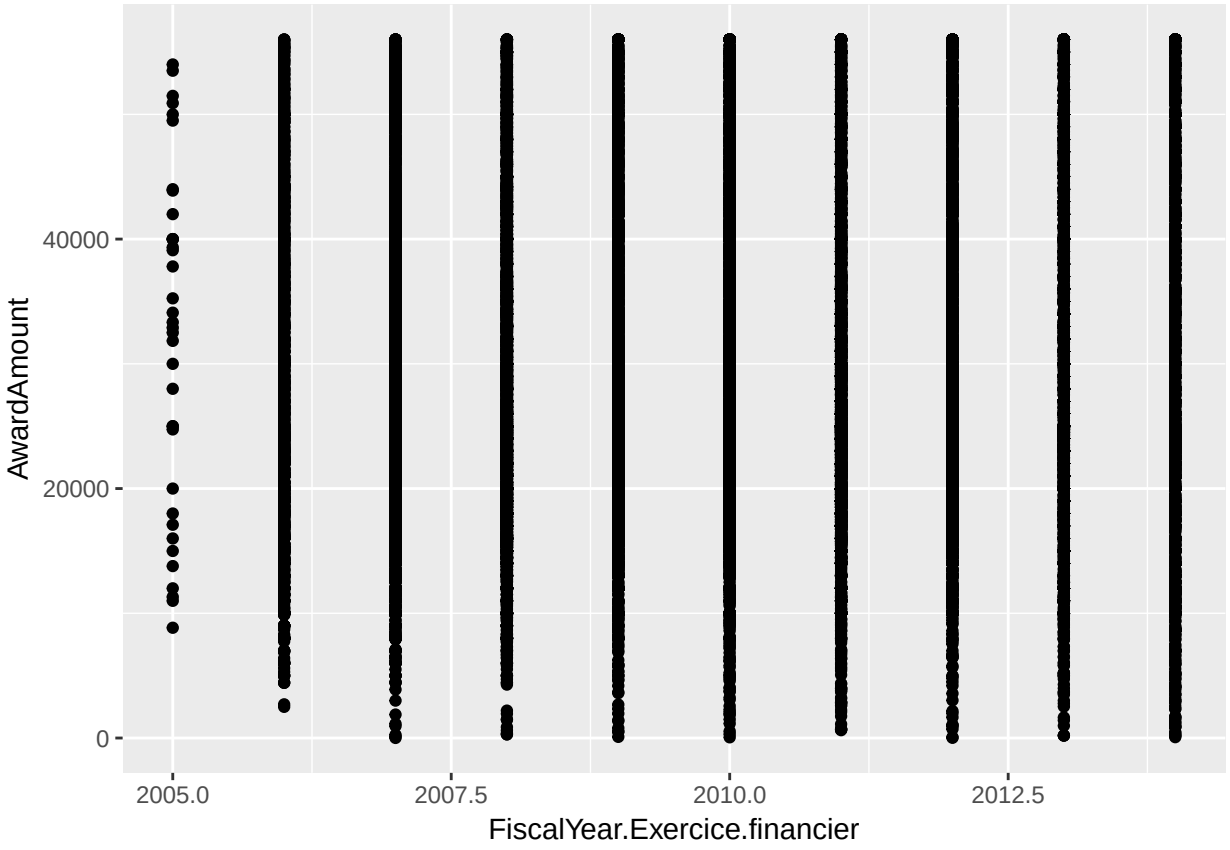
```
summary(NSERC_trimmed$AwardAmount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       7   21000   26000   28330   35000   56000
```

```
sd(NSERC_trimmed$AwardAmount) #looks much better
```

```
## [1] 10444.89
```

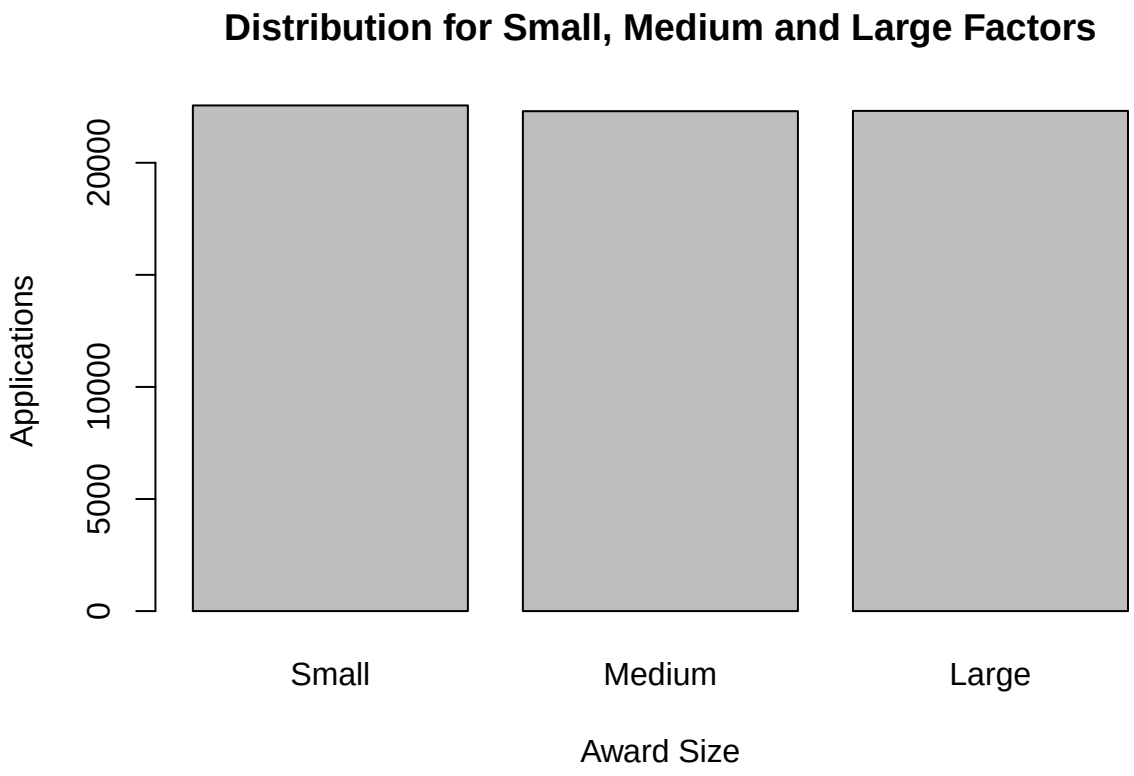
```
ggplot(NSERC_trimmed, aes(x = FiscalYear.Exercice.financier, y = AwardAmount )) + geom_point()
```



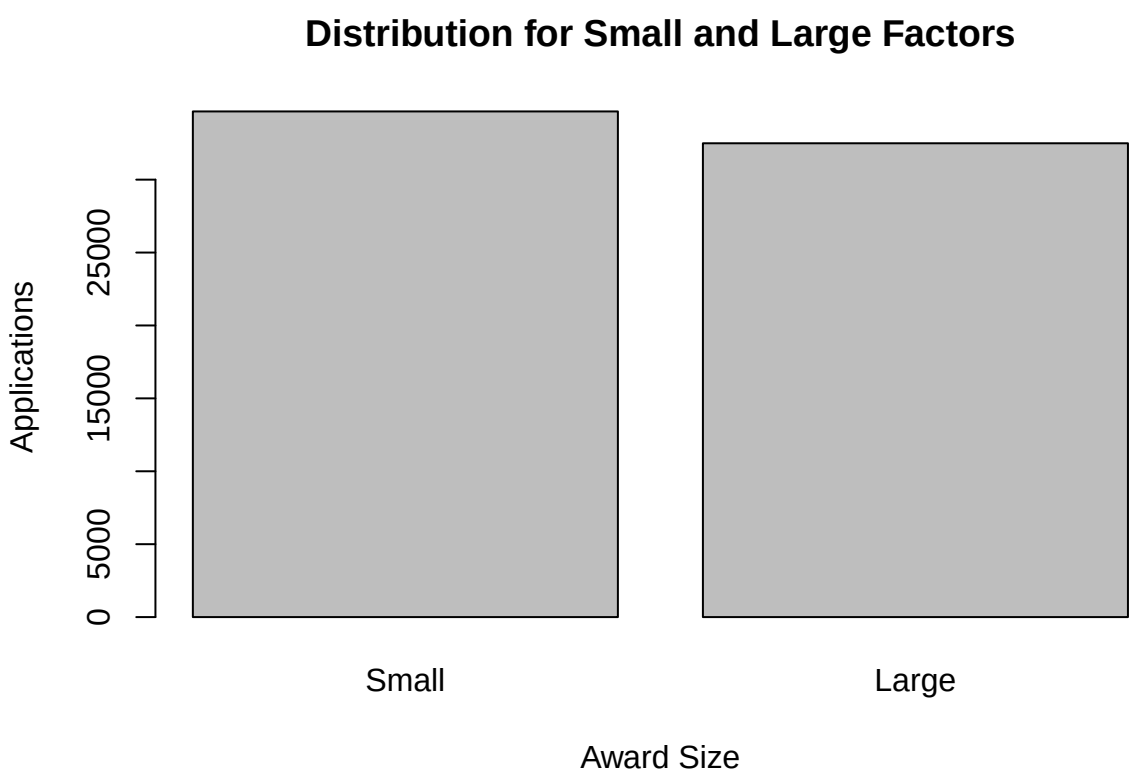

```
#bin the data, S, M, L
q <- quantile(NSERC_trimmed$AwardAmount, c(0,1/3,2/3,1))
q #returns 3 bins, consider small as < 23000, medium as between 23000 and 30999, and large as >= 31000
```

```
##         0% 33.33333% 66.66667%      100%
##          7     23000     31000     56000
```

```
NSERC_ready <- data.frame(NSERC_trimmed, bin=cut(NSERC_trimmed$AwardAmount, q, include.lowest = TRUE))
barplot(table(NSERC_ready$bin), main="Distribution for Small, Medium and Large Factors", xlab="Award Siz
        ylab="Applications", names.arg = c("Small","Medium","Large"))
```




```r
#create a 2nd bin, S, L
q <- quantile(NSERC_ready$AwardAmount, c(0,1/2,1))
NSERC_ready <- data.frame(NSERC_ready, bin2=cut(NSERC_ready$AwardAmount, q, include.lowest = TRUE))
barplot(table(NSERC_ready$bin2), main="Distribution for Small and Large Factors", xlab="Award Size",
        ylab="Applications", names.arg = c("Small","Large"))
```




```
#save data for next step
save(NSERC_ready, file = "NSERC_ready.rdata")
```

_____move forward with clean data_____

```
#load cleaned data if neccessary
load("NSERC_ready.rdata")
```

split into test and train

smp_size <- floor(0.65 * nrow(NSERC_ready))

set.seed(123)

train_ind <- sample(seq_len(nrow(NSERC_ready)), size = smp_size)

traindata <- NSERC_ready[train_ind,]

testdata <- NSERC_ready[-train_ind,]

create a text vector which will be used to create a source

trainvector <- as.vector(traindata$ApplicationSummary)$testvector <- as.vector(testdata$ApplicationSummary)

create source

trainsource <- VectorSource(trainvector) testsource <- VectorSource(testvector)

create corpus

traincorpus <- Corpus(trainsource) testcorpus <- Corpus(testsource)

Perform transformations: remove whitespace, change to lower case, remove stop words,

remove punctuation, stem, remove numbers

traincorpus <- tm_map(traincorpus, tolower)

traincorpus <- tm_map(traincorpus, removeWords,stopwords("english"))

traincorpus <- tm_map(traincorpus, removeNumbers)

traincorpus <- tm_map(traincorpus, removePunctuation)

traincorpus <- tm_map(traincorpus, stemDocument)

traincorpus <- tm_map(traincorpus, stripWhitespace) #this needs to happen after removals

traincorpus <- tm_map(traincorpus, PlainTextDocument)

testcorpus <- tm_map(testcorpus, tolower)

testcorpus <- tm_map(testcorpus, removeWords,stopwords("english"))

testcorpus <- tm_map(testcorpus, removeNumbers)

testcorpus <- tm_map(testcorpus, removePunctuation)

testcorpus <- tm_map(testcorpus, stemDocument)

testcorpus <- tm_map(testcorpus, stripWhitespace) #this needs to happen after removals

testcorpus <- tm_map(testcorpus, PlainTextDocument)

create term document matrix

trainmatrix <- DocumentTermMatrix(traincorpus, control = list(bounds = list(global = c(436,Inf))))

testmatrix <- DocumentTermMatrix(testcorpus, control = list(bounds = list(global = c(235,Inf))))

– the numbers 436 and 235 coincide with 99% of their respective corpi (setting sparsity rate)

save matrices

save(trainmatrix, file = "train_matr.rdata") save(testmatrix, file = "test_matr.rdata")

```
#load matrices as neccessary
load("../Documents/train_matr.rdata")

load("../Documents/test_matr.rdata")

#run garbage collector to speed up processing
gc()
```

```
##            used  (Mb) gc trigger  (Mb) max used  (Mb)
## Ncells  1981232 105.9    3205452 171.2  2699610 144.2
## Vcells 25577123 195.2   37258832 284.3 25751967 196.5
```

SVM model

sprs_trainmatrix <- removeSparseTerms(trainmatrix, 0.70)

sprs_testmatrix <- removeSparseTerms(testmatrix, 0.70)

– iteractions at sparsity og .99 failed, .95 failed, .90 failed, .80 failed, 0.70 success

– model is likely to be weak ay 70% sparsity

format matrix and test data for svm

dtm_svm <- as.matrix.csr(as.matrix(sprs_trainmatrix))

traindata_svm <- factor(traindata$bin)

dtmtest_svm <- as.matrix.csr(as.matrix(sprs_testmatrix))

build model

svm_model <- svm(dtm_svm,traindata_svm, kernel = “linear”)

evaluate results

svm_results <- predict(svm_model,newdata = dtm_svm)

save(svm_model, file = “svm_model.rdata”)

save(svm_results, file = “svm_results.rdata”)

```
#load matrices as neccessary
#load data if neccessary
load("../Documents/svm_results.rdata")
load("../Documents/svm_model.rdata")
load("../Documents/train_data.rdata")

#view accuracy
```

```
svm_conf_mat <- table(pred=svm_results, true=traindata$bin)
svm_AC <- (svm_conf_mat[1,1] + svm_conf_mat[2,2] + svm_conf_mat[3,3]) / sum(svm_conf_mat)
#39.1% accuracy is inferior to 48.4% (nb), not surprising given the compromises made to obtain a model
#nb is superior in this instance, abandon sum
```

---

nb model

train nb model (s,m,l)

nb_model <- naiveBayes(as.matrix(trainmatrix),as.factor(traindata$bin))

train nb2 model (s,l)

nb2_model <- naiveBayes(as.matrix(trainmatrix),as.factor(traindata$bin2))

get nb predictions

nb_results <- predict(nb_model,as.matrix(testmatrix))

get nb2 predictions

nb2_results <- predict(nb2_model,as.matrix(testmatrix))

save models and predictions

save(nb_model, file = “nb_model.rdata”) save(nb_results, file = “nb_results.rdata”) save(nb2_model, file = “nb2_model.rdata”) save(nb2_results, file = “nb2_results.rdata”)

```
#load models and results
load("../Documents/nb_results.rdata")
load("../Documents/nb_model.rdata")
load("../Documents/nb2_results.rdata")
load("../Documents/nb2_model.rdata")
load("../Documents/test_data.rdata")

#create confusion matrix for nb
nb_conf_mat <- table(pred=nb_results, true=testdata$bin)
nb_AC <- (nb_conf_mat[1,1] + nb_conf_mat[2,2] + nb_conf_mat[3,3]) / sum(nb_conf_mat) #48.4% accuracy
```

```r
#create confusion matrix for nb2
nb2_conf_mat <- table(pred=nb2_results, true=testdata$bin2)
nb2_AC <- (nb2_conf_mat[1,1] + nb2_conf_mat[2,2]) / sum(nb2_conf_mat)

save(nb_conf_mat, file = "cm.rdata")
load("../Documents/cm.rdata")

save(nb2_conf_mat, file = "cm2.rdata")
load("../Documents/cm2.rdata")

#more detailed measures, precision, recall, accuracy, kappa, F1
measures <- confusionMatrix(nb_conf_mat, mode = "prec_recall")
measures
```

```
## Confusion Matrix and Statistics
##
##                    true
## pred                [7,2.3e+04] (2.3e+04,3.1e+04] (3.1e+04,5.6e+04]
##   [7,2.3e+04]              4985              2910              2250
##   (2.3e+04,3.1e+04]        1205              2001              1164
##   (3.1e+04,5.6e+04]        1727              2882              4387
##
## Overall Statistics
##
##                Accuracy : 0.4837
##                  95% CI : (0.4773, 0.4901)
##     No Information Rate : 0.3367
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.225
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: [7,2.3e+04] Class: (2.3e+04,3.1e+04]
## Precision                         0.4914                  0.45789
## Recall                            0.6297                  0.25677
## F1                                0.5520                  0.32903
## Prevalence                        0.3367                  0.33146
## Detection Rate                    0.2120                  0.08511
## Detection Prevalence              0.4315                  0.18587
## Balanced Accuracy                 0.6494                  0.55302
##                      Class: (3.1e+04,5.6e+04]
## Precision                               0.4877
## Recall                                  0.5624
## F1                                      0.5224
## Prevalence                              0.3318
## Detection Rate                          0.1866
## Detection Prevalence                    0.3826
## Balanced Accuracy                       0.6345
```

```r
#
measures2 <- confusionMatrix(nb2_conf_mat, mode = "prec_recall")
measures2
```

```
## Confusion Matrix and Statistics
##
##                    true
## pred                [7,2.6e+04] (2.6e+04,5.6e+04]
##   [7,2.6e+04]              8498              4364
##   (2.6e+04,5.6e+04]        3621              7028
##
##                Accuracy : 0.6604
##                  95% CI : (0.6543, 0.6664)
##     No Information Rate : 0.5155
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3188
##  Mcnemar's Test P-Value : < 2.2e-16
##
##               Precision : 0.6607
##                  Recall : 0.7012
##                      F1 : 0.6804
##              Prevalence : 0.5155
##          Detection Rate : 0.3614
##    Detection Prevalence : 0.5471
##       Balanced Accuracy : 0.6591
##
##        'Positive' Class : [7,2.6e+04]
##
```

Technical Analysis - 2-Factor

After selecting Naïve Bayes as the preferred classification method, a model was built using 65% of the data. The model was then tested using the subsequent 35% of the data. Figure 1 shows the overall accuracy of the model to be 48.4%. This is significantly better ($p < 0.001$) than the baseline chance or "No Information Rate" of 33.7%. Further analysis, however, shows that not all classification levels are performing equally. While precision rates are similar, the recall rate for "Medium" ($23,000 - $30,999) awards is much less than that of "Small" (< $23,000) and "Large" Awards (> $31,000). This led me to believe that re-binning the data into 2 factors, Small and Large, might result in a stronger model.

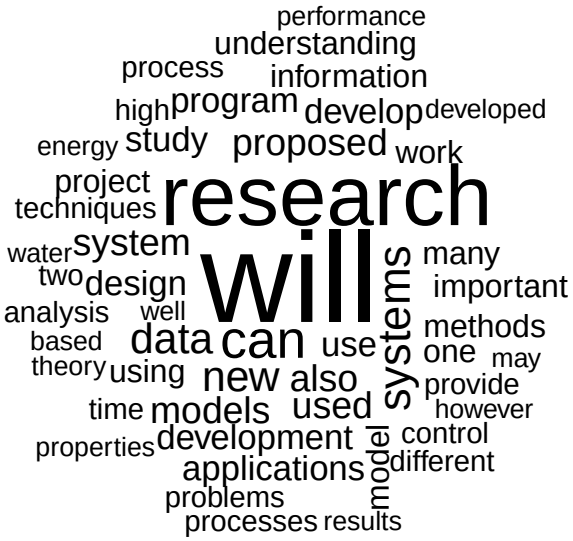
Technical Analysis - 3-Factor

The new 2 Factor Naïve Bayes model was constructed using identical test and training data, with the only difference being the new award amount classification split. This model is also significant at $p < 0.001$ with an overall accuracy of 66.0%, albeit compared to a No Information Rate of 51.6%. Figure 4 shows us that the recall rate across factors is much more balanced as compared to the previous model. Since these models have different baseline chance values, (No Information Rates), they must be compared using balanced accuracy rates. Using this statistic as a comparison, the 2 Factor model outperforms the 3 Factor model with a balanced accuracy rate of 65.9% compared to 61.2%.

Extra

```
#Extra - Play with word clouds
load("../Documents/NSERC_ready.rdata")

Small <- subset(NSERC_ready$ApplicationSummary, NSERC_ready$bin2 == "[7,2.6e+04]")
Large <- subset(NSERC_ready$ApplicationSummary, NSERC_ready$bin2 == "(2.6e+04,5.6e+04]")
```

```
Smallvector <- as.vector(Small)
Largevector <- as.vector(Large)
Smallsource <- VectorSource(Smallvector)
Largesource <- VectorSource(Largevector)
Smallcorpus <- Corpus(Smallsource)
Largecorpus <- Corpus(Largesource)

# Perform transformations
Smallcorpus <- tm_map(Smallcorpus, tolower)
Smallcorpus <- tm_map(Smallcorpus, removeWords,stopwords("english"))
Smallcorpus <- tm_map(Smallcorpus, removeNumbers)
Smallcorpus <- tm_map(Smallcorpus, removePunctuation)
Smallcorpus <- tm_map(Smallcorpus, stemDocument)
Smallcorpus <- tm_map(Smallcorpus, stripWhitespace)
Smallcorpus <- tm_map(Smallcorpus, PlainTextDocument)

Largecorpus <- tm_map(Largecorpus, tolower)
Largecorpus <- tm_map(Largecorpus, removeWords,stopwords("english"))
Largecorpus <- tm_map(Largecorpus, removeNumbers)
Largecorpus <- tm_map(Largecorpus, removePunctuation)
Largecorpus <- tm_map(Largecorpus, stemDocument)
Largecorpus <- tm_map(Largecorpus, stripWhitespace)
Largecorpus <- tm_map(Largecorpus, PlainTextDocument)


wordcloud(Smallcorpus, max.words = 50, random.order = FALSE)
```




```
wordcloud(Largecorpus, max.words = 50, random.order = FALSE)
```


```
#remove shared words
SmallCorpus_Rmv <- tm_map(Smallcorpus, removeWords, c("will","research","can","new","also","system","sy
wordcloud(SmallCorpus_Rmv, max.words = 50, random.order = FALSE)
```

```
## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): methods could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): understanding could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): important could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): project could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): provide could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): techniques could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): analysis could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): problems could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): process could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): different could not be fit on page. It will not be plotted.

## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): high could not be fit on page. It will not be plotted.

## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): processes could not be fit on page. It will not be plotted.

## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): based could not be fit on page. It will not be plotted.

## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): theory could not be fit on page. It will not be plotted.

## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): properties could not be fit on page. It will not be plotted.

## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): may could not be fit on page. It will not be plotted.

## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): developed could not be fit on page. It will not be plotted.

## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): energy could not be fit on page. It will not be plotted.

## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): performance could not be fit on page. It will not be plotted.

## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): technology could not be fit on page. It will not be plotted.

## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): proposal could not be fit on page. It will not be plotted.

## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): canada could not be fit on page. It will not be plotted.

## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): studies could not be fit on page. It will not be plotted.

## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): large could not be fit on page. It will not be plotted.

## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): problem could not be fit on page. It will not be plotted.

## Warning in wordcloud(SmallCorpus_Rmv, max.words = 50, random.order =
## FALSE): materials could not be fit on page. It will not be plotted.
```


```
LargeCorpus_Rmv <- tm_map(Largecorpus, removeWords, c("will","research","can","new","also","system","sy
wordcloud(LargeCorpus_Rmv, max.words = 50, random.order = FALSE)
```

```
## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): understanding could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): processes could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): develop could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): information could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): applications could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): control could not be fit on page. It will not be plotted.
```

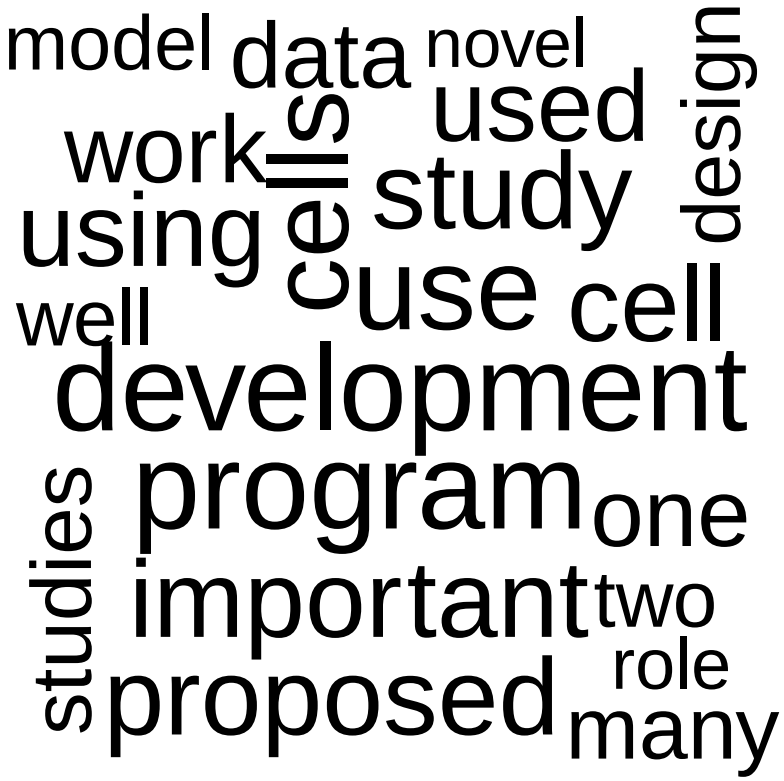
```
## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): proteins could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): different could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): provide could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
```

```
## FALSE): materials could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): properties could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): methods could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): mechanisms could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): models could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): techniques could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): energy could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): high could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): molecular could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): process could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): may could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): species could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): function could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): understand could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): protein could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): proposal could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): project could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): changes could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): specific could not be fit on page. It will not be plotted.

## Warning in wordcloud(LargeCorpus_Rmv, max.words = 50, random.order =
## FALSE): brain could not be fit on page. It will not be plotted.
```


model
data
novel
design
used
work
cells
study
using
use
cell
well
development
program
one
studies
important
two
role
proposed
many