ADS - Project 1

Henrique Saboya Lopes Tavares de Melo (hs2923)

September 10, 2017

The idea behind this project/document is to argue on two main questions: whether we are able to identify different word patterns on republican speeches when compared to democratic speeches, and, if they show different word patterns, are these patterns consistent within each party?

For that, we'll analyze the 58 inaugural speeches from most US Presidents so far, using text mining, clustering and topic modeling tools.

Thus, we will break down our project into 3 steps:

1) Text Mining: We'll use "tm" package to create a corpus of documents, remove noisy data and create a Document Term Matrix;
2) Clustering: We'll cluster our documents according to the word frequencies used on the speeches;
3) Topic Modeling: We'll use topic modeling tools to verify if Political Parties have significant distinctions in topics;

Step 0

Tools

Before we start, let's first load all necessary packages:

```r
set.seed(0)

packages.used=c("tm", "wordcloud", "fpc",
                "ggplot2", "RCurl", "xlsx", "MASS","topicmodels")

# check packages that need to be installed.
packages.needed=setdiff(packages.used,
                        intersect(installed.packages()[,1],
                                  packages.used))
# install additional packages
if(length(packages.needed)>0){
  install.packages(packages.needed, dependencies = TRUE,
                   repos='http://cran.us.r-project.org')
}

library(tm)
library(wordcloud)
library(fpc)
library(ggplot2)
library(RCurl)
library(xlsx)
library(MASS)
library(topicmodels)
```

Step 1

Text Mining

Now, we need to import the information that we have been provided with. We have 2 documents, one with speech dates, and the other with detailed information about the presidents, including their political parties which is crucial to our analyses.

```
#Reading file with speech dates from local directory
#user.dir <- setwd("Directory") #User must set local directory if necessary

dir1 <- "./data/InauguationDates.txt"

dates.speeches <- read.table(dir1, header = T, skip = 1, sep = "\t")

#Reading Information file from local directory

dir2 <- "./data/InaugurationInfo.xlsx"

InaugurationInfo <- read.xlsx(dir2, sheetIndex = 1)

rm(dir1,dir2)
```

The next step is to create a corpus with all documents (speeches in our case), so that we can perform our initial text analyzes.

```
#Importing Speech data

dir3 <- "./data/InauguralSpeeches"

doc <- DirSource(directory = dir3, encoding = "UTF-8", pattern = "\\.txt$"); rm(dir3)

ovid <- VCorpus(doc, readerControl = list(language = "en"))
```

Let's get a sense of our corpus:

```
ovid
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 58
```

```
inspect(ovid[1])
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 1
##
## [[1]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 20974
```

On the following step, we'll make use of "tm" functions to: standardize all words to lower case, remove stop words, remove punctuations and clean white spaces.

```
ovid.t <- ovid   #Renaming corpus
ovid.t <- tm_map(ovid.t, content_transformer(tolower))
ovid.t <- tm_map(ovid.t, removeWords, stopwords("english"))
ovid.t <- tm_map(ovid.t, removePunctuation, preserve_intra_word_dashes = T)
ovid.t <- tm_map(ovid.t, stripWhitespace)
```

ovid.t becomes our new transformed corpus.

With our speeches data now cleaned up and organized in a corpus, it is easy to make some analyzes. Let's create a Document Term Matrix to first explore and understand which words are more commonly used by American Presidents. This will later allow us to make deeper analyzes.

```
dtm <- DocumentTermMatrix(ovid.t) #Function to create matrix
dim(dtm)
```

```
## [1]   58 9407
```

Our Document term Matrix has 58 rows (documents) and 9407 columns (vocabulary words). A brief summary is shown below:

```
inspect(dtm[1:5, 10:13])
```

```
## <<DocumentTermMatrix (documents: 5, terms: 4)>>
## Non-/sparse entries: 1/19
## Sparsity           : 95%
## Maximal term length: 10
## Weighting          : term frequency (tf)
## Sample             :
##                             Terms
## Docs                         1780 1787 1789 1789-words
##   inaugAbrahamLincoln-1.txt    0    1    0          0
##   inaugAbrahamLincoln-2.txt    0    0    0          0
##   inaugAndrewJackson-1.txt     0    0    0          0
##   inaugAndrewJackson-2.txt     0    0    0          0
##   inaugBarackObama-1.txt       0    0    0          0
```

As expected, we initially have a high level of sparsity (columns with many zeros), however, for now, let's take a look at the overall most common words used by presidents. Using a "rule of thumb", we'll select only the words that appear more than 150 times.

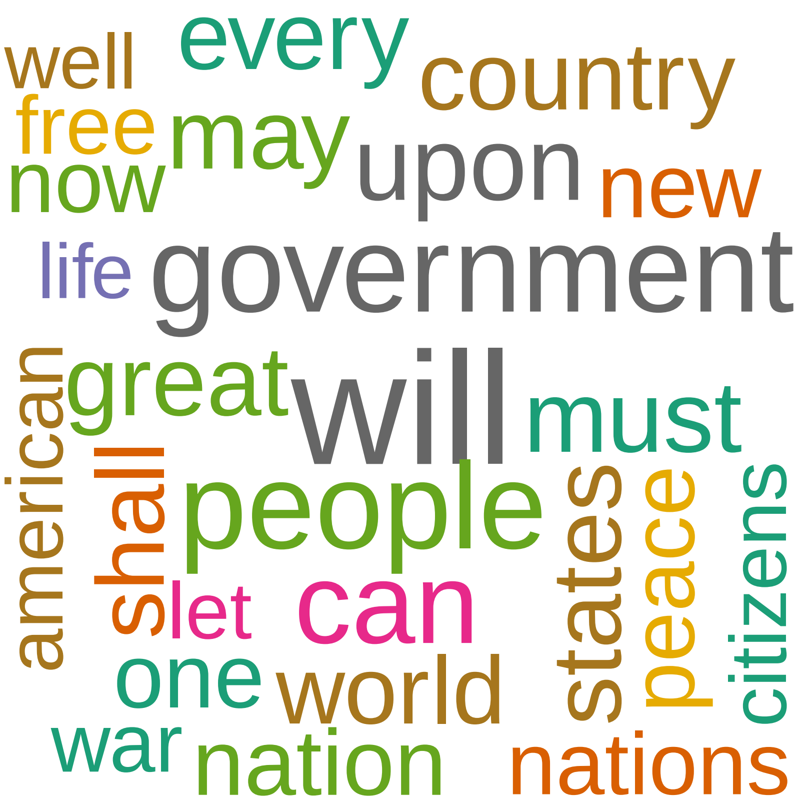
```
findFreqTerms(dtm, 150)
```

```
##  [1] "america"      "american"     "can"          "citizens"
##  [5] "constitution" "country"      "every"        "free"
##  [9] "freedom"      "government"   "great"        "made"
## [13] "may"          "must"         "nation"       "national"
## [17] "nations"      "new"          "now"          "one"
## [21] "peace"        "people"       "power"        "public"
## [25] "shall"        "states"       "time"         "union"
## [29] "united"       "upon"         "war"          "will"
## [33] "world"
```

It's now clear to see the most common words among President's speeches. First, words such as 'America', 'Country' and 'Citizens' could have been foreseen in any presidential speech, however let's focus on some specific words that may relate to American values. The words 'Free', 'Freedom', 'Peace', 'Power', 'Union'and 'War' have deep ideological meanings which we can relate to important Ideological Values for the United States.

Making a Word Cloud will help us visualize this information.

```
wordcloud(tm_map(ovid.t, PlainTextDocument), min.freq=100, scale=c(5,2),
          random.color=T, max.word=60, random.order=F,colors=brewer.pal(8,"Dark2"))
```



Now, as described before, our Document Term Matrix has more than 9000 words. To make more meaningful analyzes, let's get rid of sparse terms, focusing on the most repeated ones. We are not looking for an optimal sparsity value to filter our data, so let's set a 50% column sparsity to be our cutting boundary.

```
dtm <- removeSparseTerms(dtm, 0.5)
dim(dtm)
```

```
## [1]  58 150
```

```
inspect(dtm[1:5, 10:13])
```

```
## <<DocumentTermMatrix (documents: 5, terms: 4)>>
## Non-/sparse entries: 12/8
## Sparsity           : 40%
## Maximal term length: 6
## Weighting          : term frequency (tf)
## Sample             :
##                            Terms
## Docs                         best better called can
##   inaugAbrahamLincoln-1.txt    2      4      0  28
##   inaugAbrahamLincoln-2.txt    0      0      1   0
##   inaugAndrewJackson-1.txt     1      0      1   5
##   inaugAndrewJackson-2.txt     1      0      1   3
##   inaugBarackObama-1.txt       0      2      0  13
```

```
dtm.df <- as.data.frame(as.matrix(dtm))
```

We now have a matrix with only 150 terms!

Also, let's merge the information from president speeches to our Document Term matrix.

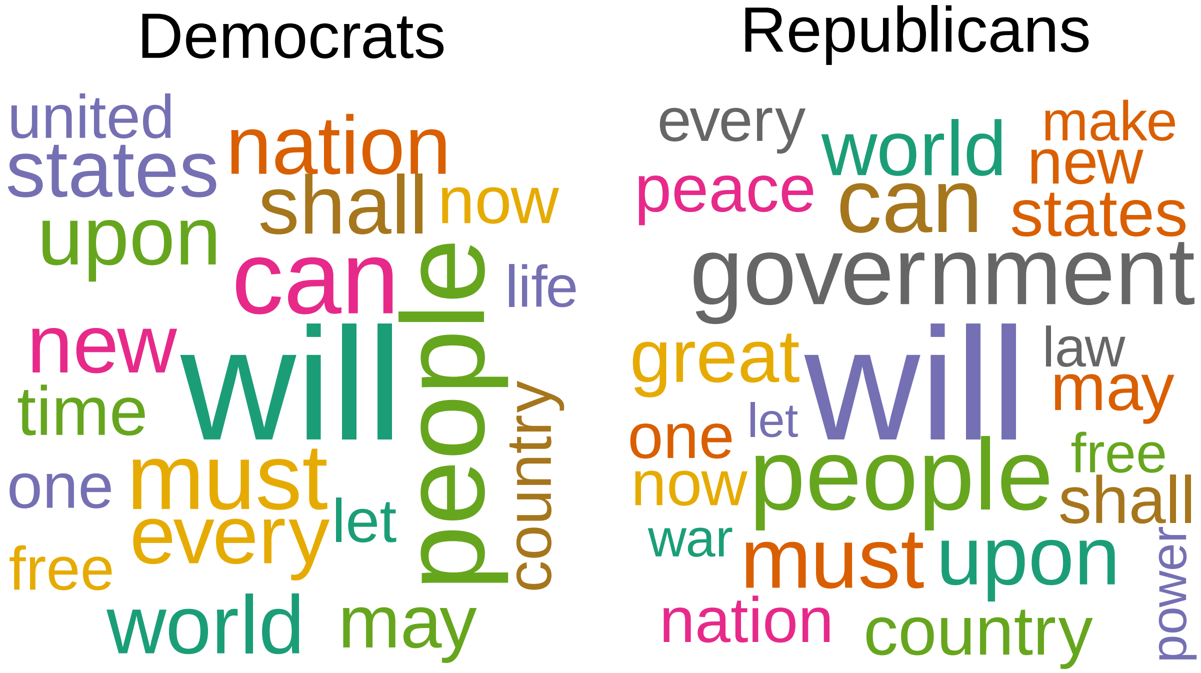
```
InaugurationInfo$Key <- paste("inaug", InaugurationInfo$File, "-",InaugurationInfo$Term, ".txt", sep =
dtm.df$Key <- rownames(dtm.df) #Creates a key which will be used to link the tables

dtm.df <- merge(x = dtm.df, y = InaugurationInfo[,c(4,6)])

#Reshape and adequate the dataframe format before merging
ds <- reshape(dates.speeches, direction = "long", varying = list(2:5), v.names = "SpeechDates")
ds <- ds[!ds$SpeechDates == "", c(1:3)]
ds$SpeechDates <- as.Date(ds$SpeechDates, "%m/%d/%Y")
names(ds) <- c("President", "Term", "SpeechDate")
ds$Key <- paste("inaug", sapply(ds$President, gsub, pattern = " ", replacement = ""), "-",ds$Term, ".tx
```

Step 2

Clustering

Before we proceed with Clustering, let's take a peek at the World Cloud for both Republicans and Democrats, to check if we can already identify some differences.

```
par(mfrow = c(1,2))

wordcloud(words = names(dtm.df[which(dtm.df$Party == "Democratic"), which(sapply(dtm.df, is.numeric))])

text(labels = "Democrats", x = 0.5, y = 1.1, cex = 2)

wordcloud(words = names(dtm.df[which(dtm.df$Party == "Republican"), which(sapply(dtm.df, is.numeric))])

text(labels = "Republicans", x = 0.5, y = 1.1, cex = 2)
```



Let us also set other parties with the same label.

```
dtm.df$Party <- ifelse(dtm.df$Party == "Republican" | dtm.df$Party == "Democratic", as.character(dtm.df$
```

At our first look, it is hard to make any conclusions. The words just tend to repeat themselves for both groups.

Nonetheless, with our Document Term Matrix, we are now able to perform some Clustering analyzes. The idea is to group speeches according to the words they tend to be used. We don't need to find an optimal number os clusters, we just need to identify whether we can distinguish Democrats from Republicans by their vocabulary, hence, we'll just try different number of clusters using K-means.

```
Cluster.model2 <- kmeans(dtm, 2, iter.max = 15, nstart = 20)
Cluster.model3 <- kmeans(dtm, 3, iter.max = 15, nstart = 20)
Cluster.model4 <- kmeans(dtm, 4, iter.max = 15, nstart = 20)
Cluster.model5 <- kmeans(dtm, 5, iter.max = 15, nstart = 20)

dtm.df$Cluster2 <- Cluster.model2$cluster
dtm.df$Cluster3 <- Cluster.model3$cluster
dtm.df$Cluster4 <- Cluster.model4$cluster
dtm.df$Cluster5 <- Cluster.model5$cluster
```

Clustering our speeches into 2, 3, 4 and 5 clusters, let's understand how they differ.

```
table(dtm.df[ ,c("Party", "Cluster2")])
```

```
##             Cluster2
## Party         1  2
##   Democratic 18  4
##   Other       8  4
##   Republican 16  8
```

If we look at 2 clusters only, we could assume Republicans and Democrats do seem to have similar vocabularies.

```
table(dtm.df[ ,c("Party", "Cluster3")])
```

```
##             Cluster3
## Party         1  2  3
##   Democratic  0  4 18
##   Other       1  3  8
##   Republican  0  8 16
```

Looking at 3 clusters, we can identify what seems to be an outlier, and the rest looks similar. Let's see how it behaves further.

```
table(dtm.df[ ,c("Party", "Cluster4")])
```

```
##             Cluster4
## Party         1  2  3  4
##   Democratic 12  4  6  0
##   Other       8  3  0  1
##   Republican  5  8 11  0
```

Using 4 clusers, we are now able to raise some questions. Most Democrats fall within 1 Cluster, however republicans tend to go to 2 other Clusters mainly, in different proportion than Democrats.

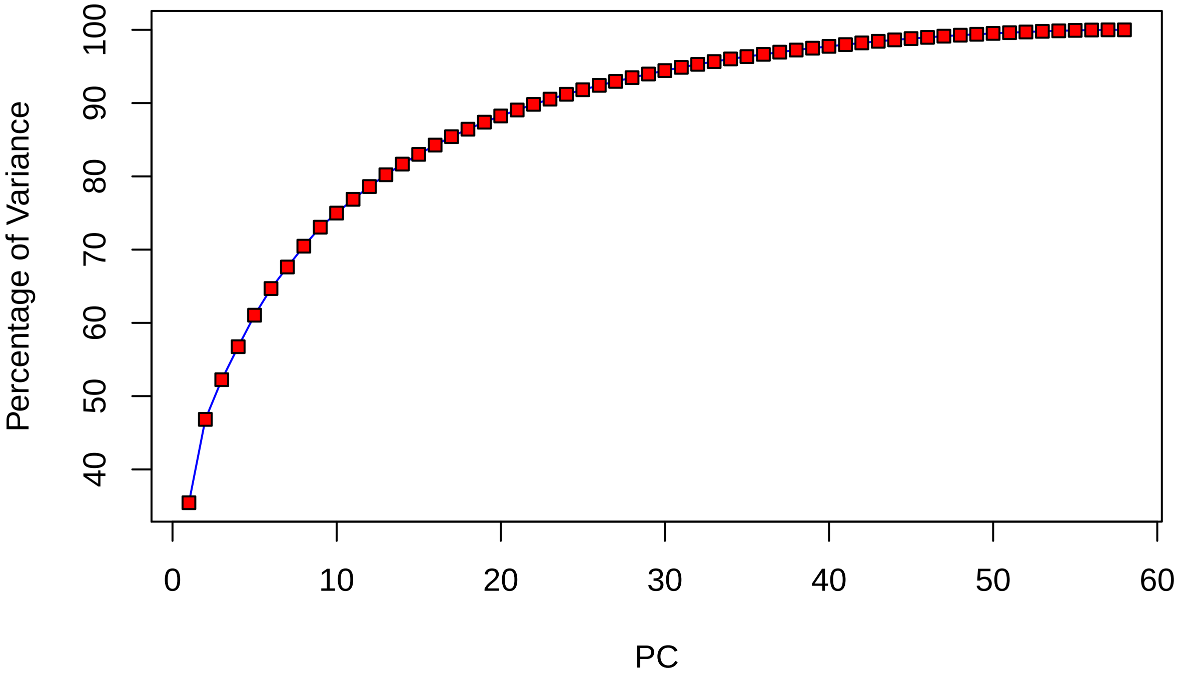
```
table(dtm.df[ ,c("Party", "Cluster5")])
```

```
##             Cluster5
## Party         1  2  3  4  5
##   Democratic  0  1  3  6 12
##   Other       1  0  4  0  7
##   Republican  0  6  3 11  4
```

Breaking down into 5 Clusters, we can see how Democrats (and also other Parties) get seppareted from Republicans, even though we can also see a significant number of exceptions.

To better visualize these Clusters, let's perform a PCA analyzes in order to reduce our problem to 2 dimensions.

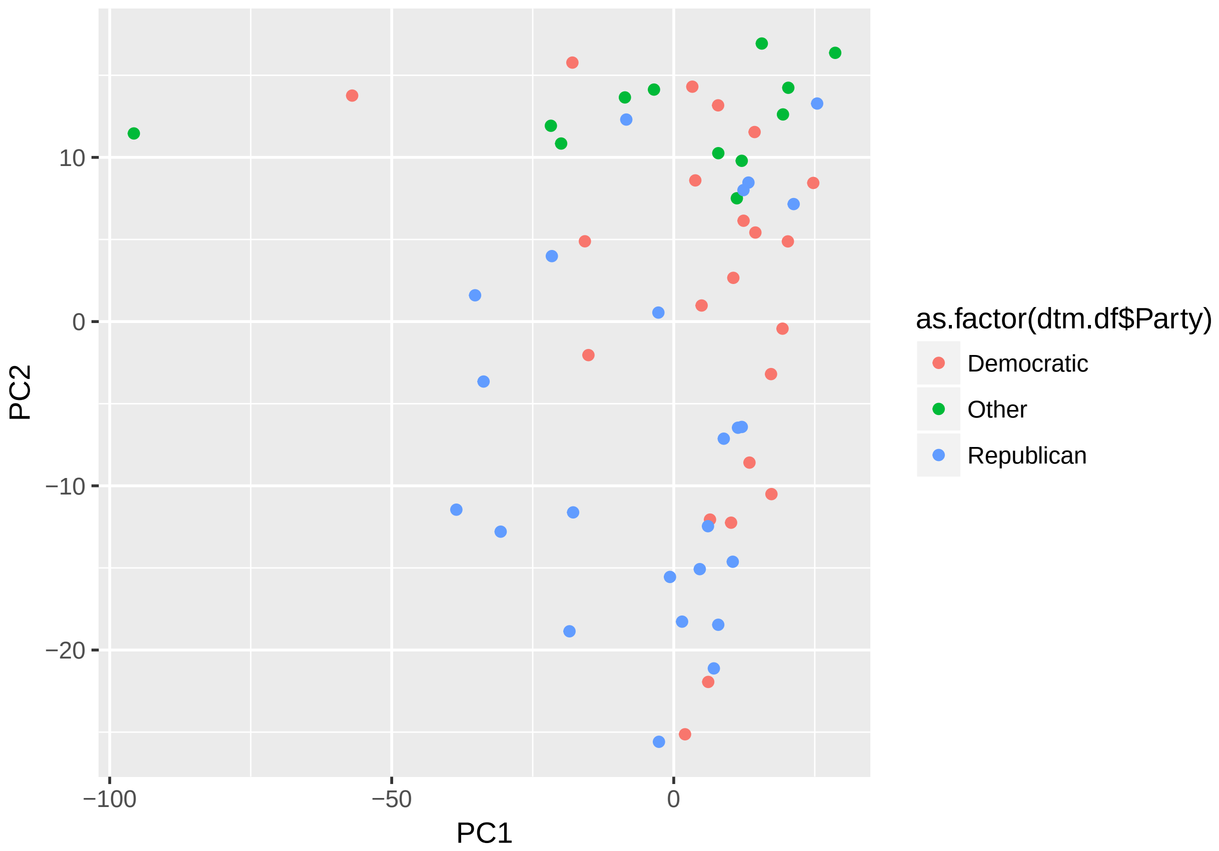
```
PCA.model <- prcomp(dtm)

plot(cumsum((PCA.model$sdev^2)/sum(PCA.model$sdev^2)*100), type = "l", col = "blue", ylab = "Percentage
points(cumsum((PCA.model$sdev^2)/sum(PCA.model$sdev^2)*100), pch = 22, bg = "red")
```



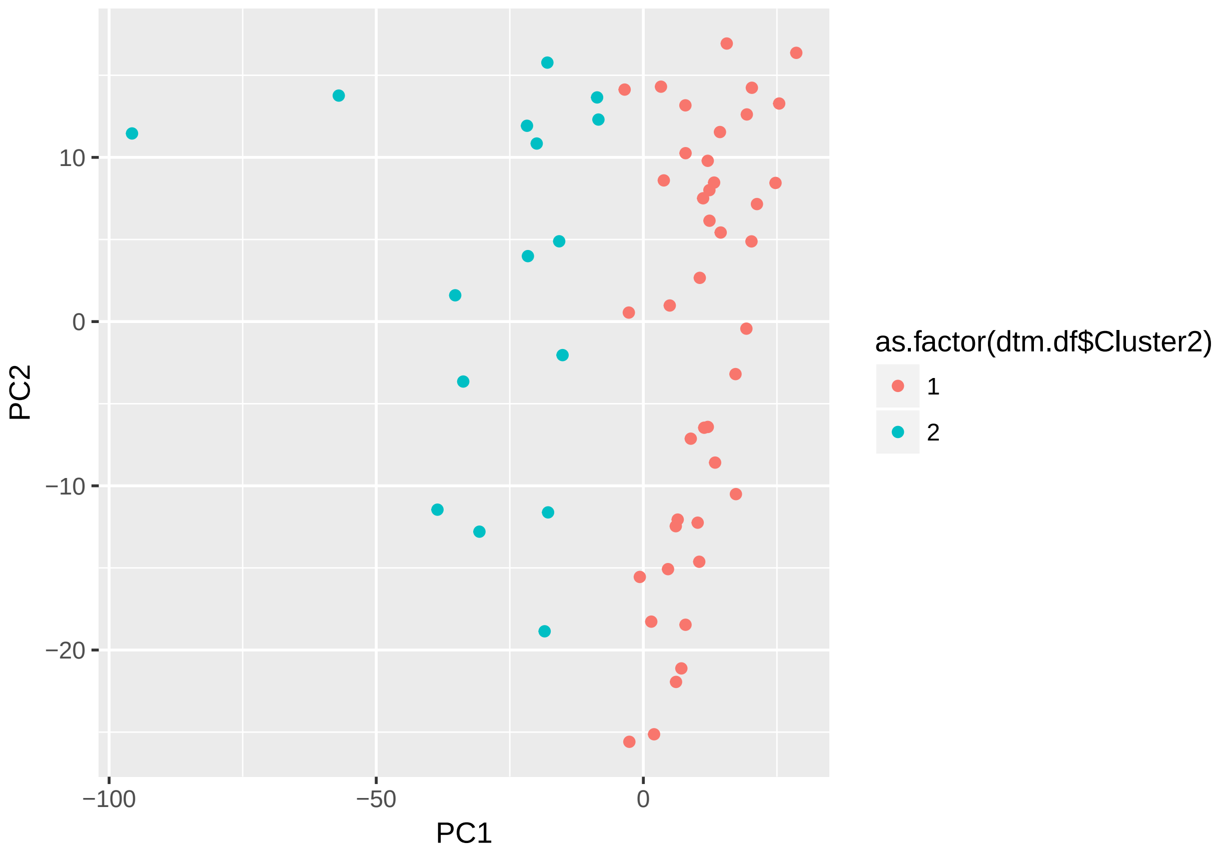
Our first 2 Principal Components are capable of explaining approximately 50% of our Variance.

Thus, Let's first look where do our speeches fall according to their Political parties, and then according to their clusters:

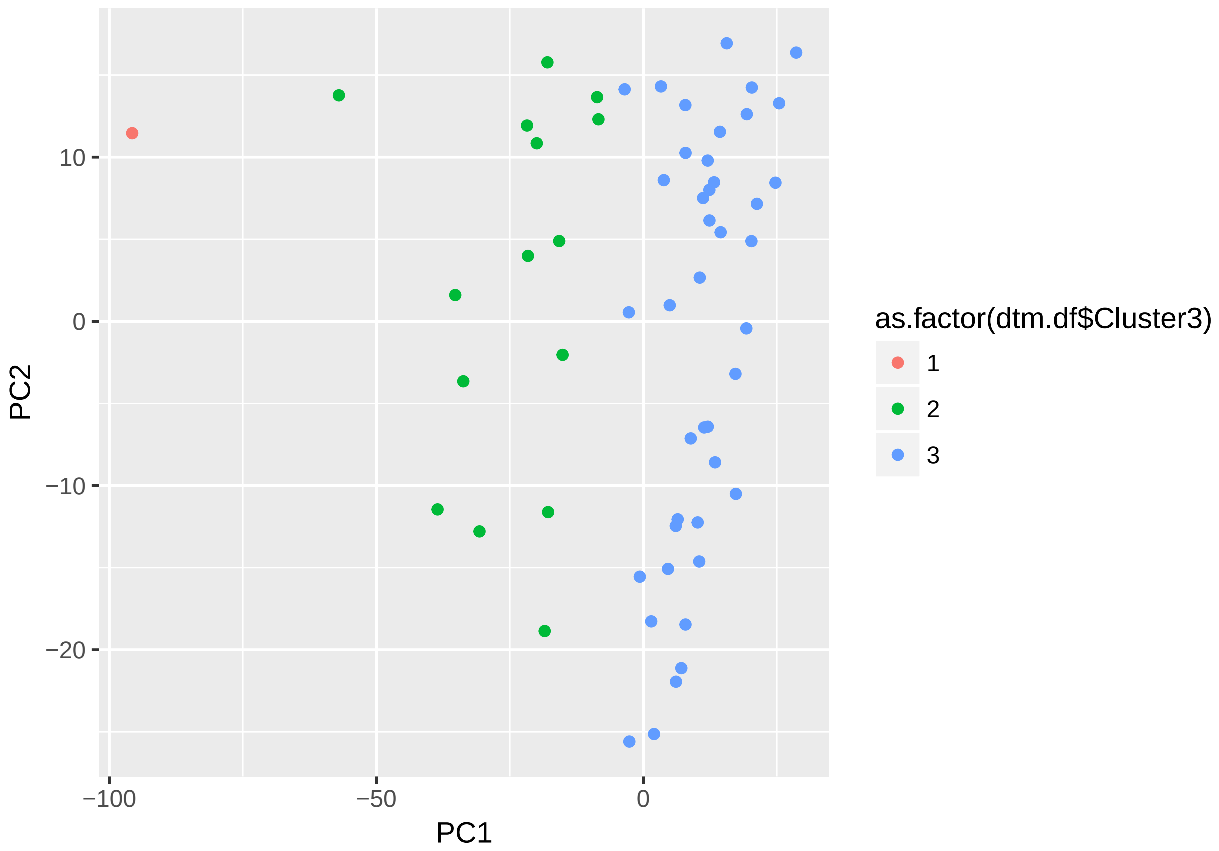
```
ggplot(data = as.data.frame(PCA.model$x[,1:2]), aes(PC1, PC2)) +
  geom_point(aes(col = as.factor(dtm.df$Party)))
```



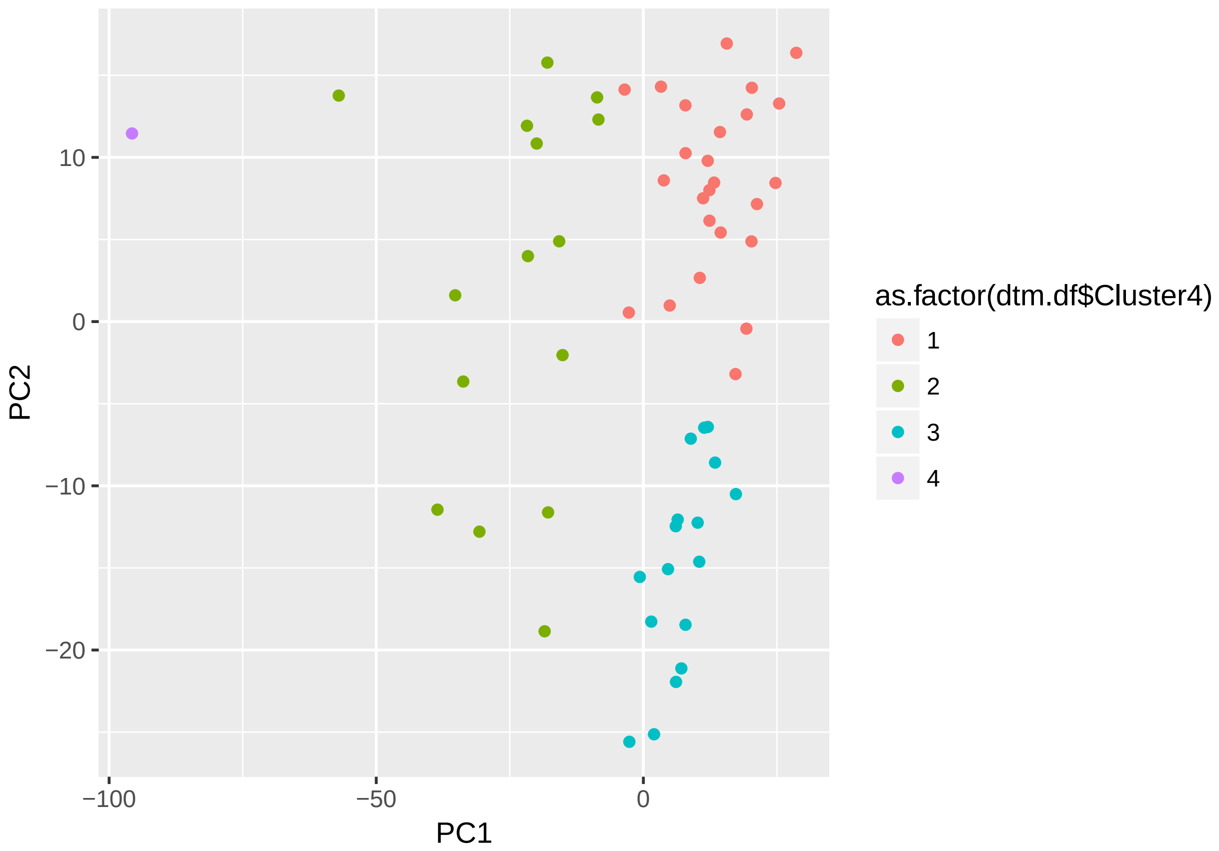
```
#2 Clusters

ggplot(data = as.data.frame(PCA.model$x[,1:2]), aes(PC1, PC2)) +
  geom_point(aes(col = as.factor(dtm.df$Cluster2)))
```



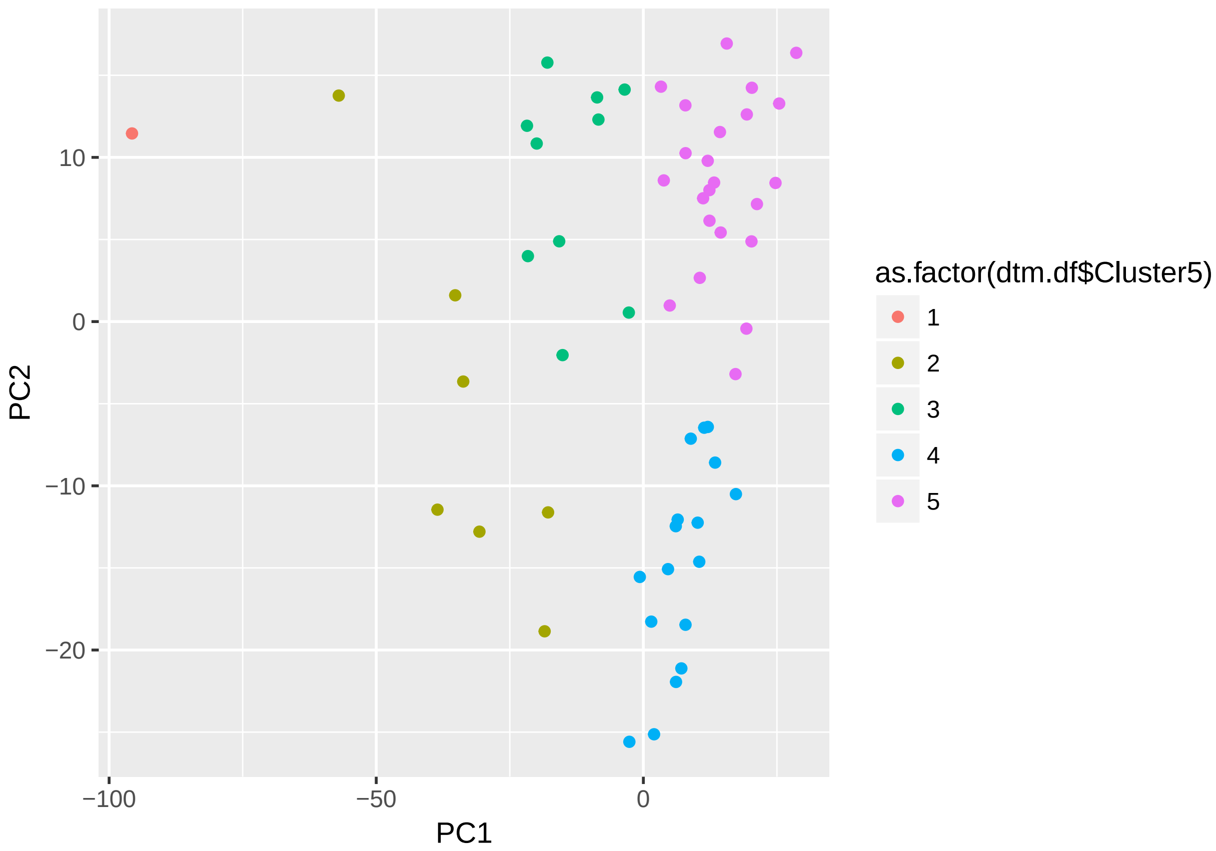
```
#3 Clusters

ggplot(data = as.data.frame(PCA.model$x[,1:2]), aes(PC1, PC2)) +
  geom_point(aes(col = as.factor(dtm.df$Cluster3)))
```



```
#4 Clusters

ggplot(data = as.data.frame(PCA.model$x[,1:2]), aes(PC1, PC2)) +
  geom_point(aes(col = as.factor(dtm.df$Cluster4)))
```



```
#5 Clusters

ggplot(data = as.data.frame(PCA.model$x[,1:2]), aes(PC1, PC2)) +
  geom_point(aes(col = as.factor(dtm.df$Cluster5)))
```



Now that we have seen our contigency tables and had a visual input of our data, what can we conclude so far:

1. There are some patterns on Democratic speeches indeed that are different from Republican speeches, with some exceptions apparently;
2. Most democratic speeches seem to be consistent with each other, however republican speeches tend to diverge from their selves and from other parties, also with exceptions;
3. Looking at the PC1 and PC2 scatterplot, we can argue that many President's inaugural speeches tend to use the same group of words, however, there are some "innovative" speeches that fall a bit outisde of the group of regular speeches, where their majority are within the Republican Party. In summary, the Republicans usually fall outside of the common.

The choice of K-Means with PCA was good to have a sense of the Words used and differences among speeches, however when we look at the data, using a non-linear model may seem a good idea if we are to better segregate Democrats from Republicans.

Step 3

Topic Modeling

Now, let's use topic modeling tools to verify if we can distinguish republican speeches from democratic speeches based on topic assignments. If most Democratic and Republican speeches differ in topic assignemnts, we can argue they might have different speeches indeed.

For that, we'll perform LDA, setting k to be 4 topics first:

```
#Set parameters for Gibbs sampling
burnin <- 4000
iter <- 2000
thin <- 500
seed <-list(2003,5,63,100001,765)
nstart <- 5
best <- TRUE

k <- 4

LDA.model <-LDA(dtm, k, method="Gibbs", control = list(nstart = nstart,
                                                 seed = seed, best = best,
                                                 burnin = burnin, iter = iter,
                                                 thin = thin))
```

After running LDA, let's look at our output:.

```
LDA.model.topics <- as.matrix(topics(LDA.model))

table(c(1:k, LDA.model.topics))
```

```
## 
##  1  2  3  4 
## 21 25 11  5
```

```
#write.csv(LDA.model.topics, file = paste("./output/LDAGibbs", k, "DocsToTopics.csv"))
```

Most of the speeches fall within topics 1 and 2

```
#top 20 terms in each topic
LDA.model.terms <- as.matrix(terms(LDA.model, 20))
#write.csv(LDA.model.terms, file = paste("./output/LDAGibbs", k, "TopicsToTerms.csv"))

#probabilities associated with each topic assignment
LDA.model.prob <- as.data.frame(LDA.model@gamma)
#write.csv(LDA.model.prob, file = paste("./output/LDAGibbs", k, "TopicProbabilities.csv"))

head(LDA.model.terms)
```

```
##      Topic 1        Topic 2   Topic 3   Topic 4 
## [1,] "government"   "world"   "great"   "will"  
## [2,] "upon"         "new"     "country" "can"   
## [3,] "states"       "nation"  "every"   "people"
## [4,] "shall"        "freedom" "may"     "one"   
## [5,] "public"       "peace"   "united"  "must"  
## [6,] "constitution" "nations" "war"     "now"
```

The table above shows the top 6 words within each topic. We can use these to label our topics names. For instance we can say our topics are: Government, Liberalism, Patriotism and Belief.

```
topics.names <- c("Government", "Liberalism", "Patriotism", "Belief")

#Find relative importance of top 2 topics
topic1ToTopic2 <- lapply(1:nrow(dtm), function(x)
sort(LDA.model.prob[x,])[k]/sort(LDA.model.prob[x,])[k-1])

#write to file
```

```
#write.csv(topic1ToTopic2, file = paste("./output/LDAGibbs", k, "Topic1ToTopic2.csv"))
```

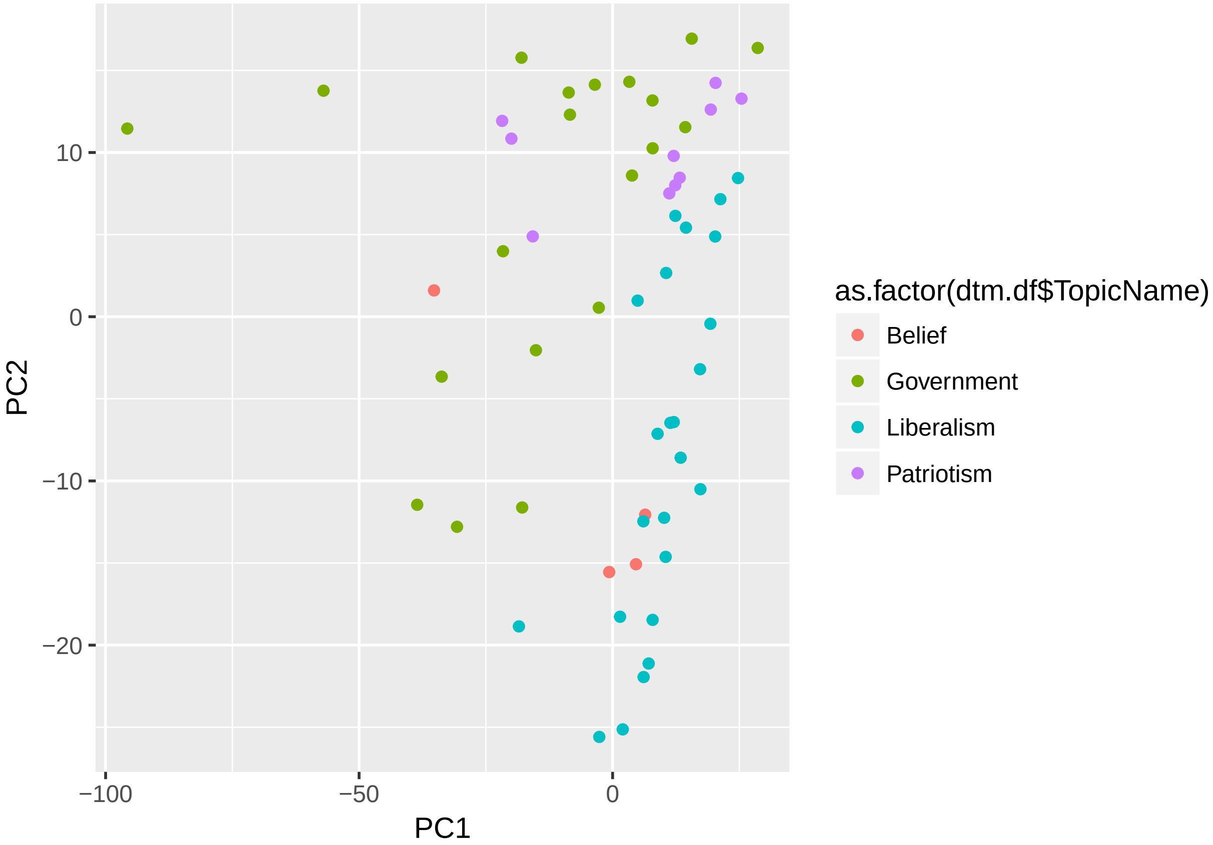
Now, let's understand how our Topics relate to Political Parties.

```
#Merge the topic information with previous dataframe
LDA.model.topics.df <- as.data.frame(as.matrix(topics(LDA.model)))
LDA.model.topics.df$Key <- rownames(as.matrix(topics(LDA.model)))
names(LDA.model.topics.df) <- c(paste("Topic ",k), "Key")

LDA.model.topics.df$TopicName <- topics.names[LDA.model.topics.df$Topic]

dtm.df <- merge(x = dtm.df, y = LDA.model.topics.df)

table(dtm.df$Party, dtm.df$TopicName)
```

```
##
##               Belief Government Liberalism Patriotism
##   Democratic       1          7         13          1
##   Other            0          6          0          6
##   Republican       3          7         11          3
```

Apparently, from the table above, we cannot see any difference on speeche topics from Republicans to Democrats. If we look at our previous PCA Plots:

```
ggplot(data = as.data.frame(PCA.model$x[,1:2]), aes(PC1, PC2)) +
  geom_point(aes(col = as.factor(dtm.df$TopicName)))
```



Now, these similarities maybe due to the fact that we are taking a low number of topics, only 4. Thus, these topics become to broad. Let's try to increase the number of topics:

```
k <- 10

LDA.model <-LDA(dtm, k, method="Gibbs", control = list(nstart = nstart,
                                                  seed = seed, best = best,
                                                  burnin = burnin, iter = iter,
                                                  thin = thin))


LDA.model.topics <- as.matrix(topics(LDA.model))

table(c(1:k, LDA.model.topics))
```

```
## 
##  1  2  3  4  5  6  7  8  9 10 
##  6  8  6  4 11  9  6  3  4 11
```

```
#write.csv(LDA.model.topics, file = paste("./output/LDAGibbs", k, "DocsToTopics.csv"))

#top 20 terms in each topic
LDA.model.terms <- as.matrix(terms(LDA.model, 20))
#write.csv(LDA.model.terms, file = paste("./output/LDAGibbs", k, "TopicsToTerms.csv"))

#probabilities associated with each topic assignment
LDA.model.prob <- as.data.frame(LDA.model@gamma)
#write.csv(LDA.model.prob, file = paste("./output/LDAGibbs", k, "TopicProbabilities.csv"))

head(LDA.model.terms)
```

```
##      Topic 1   Topic 2  Topic 3    Topic 4      Topic 5  Topic 6  
## [1,] "nations" "upon"   "power"    "must"       "new"    "nation" 
## [2,] "freedom" "shall"  "may"      "can"        "world"  "life"   
## [3,] "peace"   "people" "people"   "government" "let"    "men"    
## [4,] "world"   "laws"   "can"      "people"     "time"   "justice"
## [5,] "free"    "law"    "one"      "progress"   "work"   "know"   
## [6,] "faith"   "may"    "citizens" "better"     "people" "day"    
##      Topic 7        Topic 8    Topic 9   Topic 10    
## [1,] "government"   "will"     "great"   "country"   
## [2,] "states"       "now"      "war"     "public"    
## [3,] "constitution" "american" "made"    "every"     
## [4,] "union"        "every"    "years"   "just"      
## [5,] "united"       "right"    "without" "principles"
## [6,] "powers"       "make"     "force"   "confidence"
```

Given the output above, let's name our topics:

```
topics.names <- c("Liberalism", "Law", "Belief", "Progress", "Union", "Nation", "Government", "America"

#Find relative importance of top 2 topics
topic1ToTopic2 <- lapply(1:nrow(dtm), function(x)
sort(LDA.model.prob[x,])[k]/sort(LDA.model.prob[x,])[k-1])

#write to file
#write.csv(topic1ToTopic2, file = paste("./output/LDAGibbs", k, "Topic1ToTopic2.csv"))
```

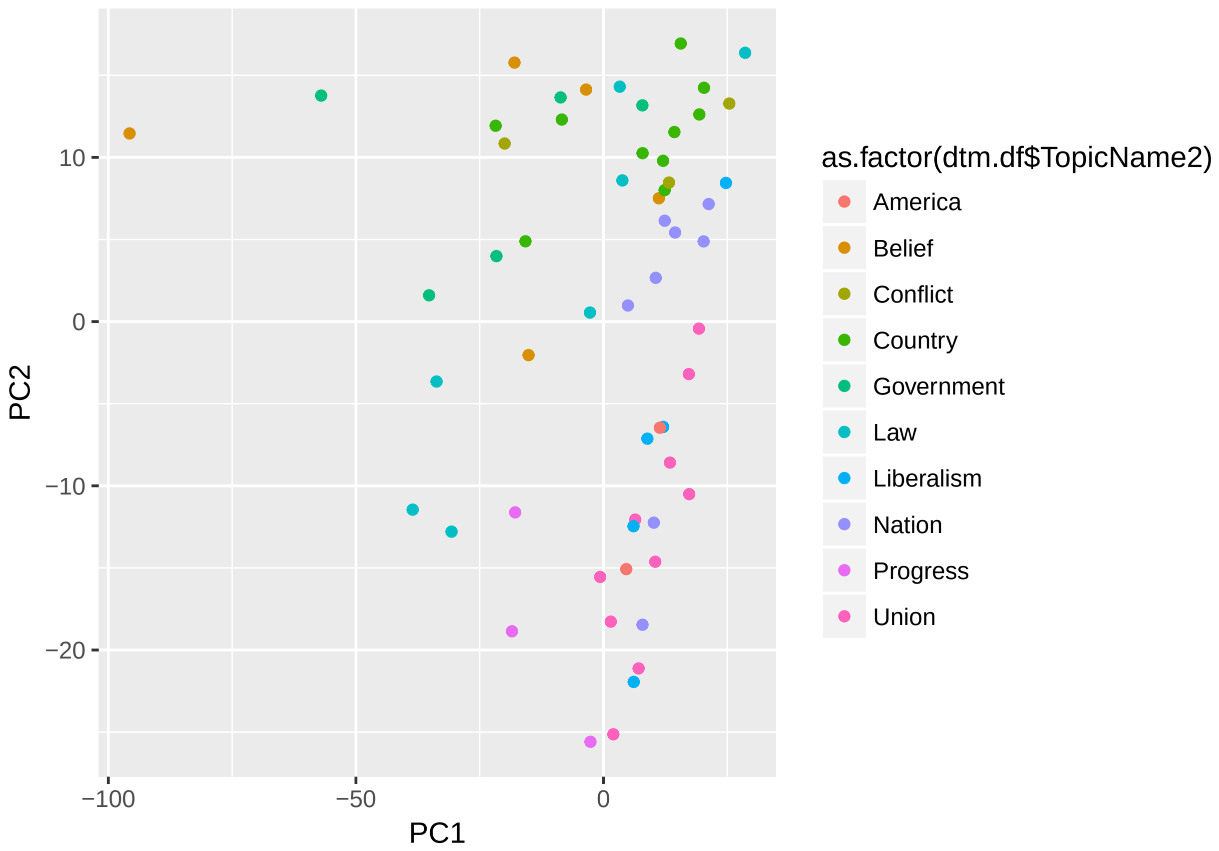
```r
#Merge the topic information with previous dataframe
LDA.model.topics.df <- as.data.frame(as.matrix(topics(LDA.model)))
LDA.model.topics.df$Key <- rownames(as.matrix(topics(LDA.model)))
names(LDA.model.topics.df) <- c(paste("Topic ",k), "Key")

LDA.model.topics.df$TopicName2 <- topics.names[LDA.model.topics.df$Topic]

dtm.df <- merge(x = dtm.df, y = LDA.model.topics.df)

table(dtm.df$Party, dtm.df$TopicName2)
```

```
##
##              America Belief Conflict Country Government Law Liberalism
##   Democratic       0      2        0       2          2   2          2
##   Other            0      3        1       6          1   1          0
##   Republican       2      0        2       2          2   4          3
##
##              Nation Progress Union
##   Democratic      6        0     6
##   Other           0        0     0
##   Republican      2        3     4
```

```r
table(dtm.df$TopicName2, dtm.df$Party)
```

```
##
##              Democratic Other Republican
##   America             0     0          2
##   Belief              2     3          0
##   Conflict            0     1          2
##   Country             2     6          2
##   Government          2     1          2
##   Law                 2     1          4
##   Liberalism          2     0          3
##   Nation              6     0          2
##   Progress            0     0          3
##   Union               6     0          4
```

```r
ggplot(data = as.data.frame(PCA.model$x[,1:2]), aes(PC1, PC2)) +
  geom_point(aes(col = as.factor(dtm.df$TopicName2)))
```



Now, there's not much we can interpret out of the graph, however, we can make some interesting comments on the table. Though it is hard to say, in general, which political Party does one speech belongs to, based only on the topic, we can see that:

1. The Topics "America" and "Conflict" are mostly republican topics. We don't see any democrat speech classified as such.
2. Republicans fall spreadout within all Topics, which corroborates with what we said above, that Republicans tend to have innovative speeches.
3. Most Democrats fall within either "Union" or "Nation", which shows some repetition in words and topcis used.

Final Notes

In conclusion, these analyzes may have not provided us with a generalized "yes" or "no" answer, on whether Republicans and Democrats have different speeches, by just looking at words. However, we can clearly see significant distinctions on word choices and topics, whithin these 2 political parties. If we would continue this study in a qualitative level, analizing political and economic aspects, these differents might make total sense.

As next step, I would suggest looking at sentence patterns, not just words, and compare both results.