MDS Data Science Portfolio Part 1

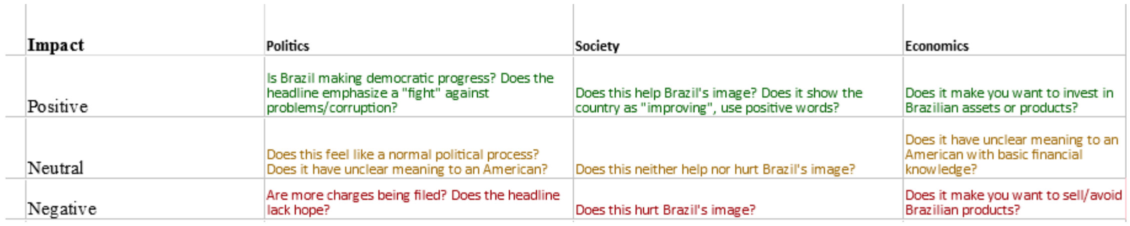
MD Shuey

Demonstration 1: Data Cleaning, Qualitative Analysis

MediaMonitor: Collection methodology

In this dataset, headlines were compiled from a specific list of sources with the keyword “Brazil” since the prior business day of collection. Sports matters were omitted, with an emphasis on collecting news regarding the state of Brazil and its economy, society, and politics, categorized as *pillars*. From there, a *topic* if applicable further delved into relevant subcategories such as agriculture, technology, energy, etc.

This data was compiled between the beginning of the 2017 calendar year and the end of November. There were 222 days of headline collection during this time period, with 2045 total headlines. Of particular interest, a subjective qualitative variable *impact* was also given with each observation: Labelling a positive, negative, or neutral impact on a potential reader:



Impact	Politics	Society	Economics
Positive	Is Brazil making democratic progress? Does the headline emphasize a "fight" against problems/corruption?	Does this help Brazil's image? Does it show the country as "improving", use positive words?	Does it make you want to invest in Brazilian assets or products?
Neutral	Does this feel like a normal political process? Does it have unclear meaning to an American?	Does this neither help nor hurt Brazil's image?	Does it have unclear meaning to an American with basic financial knowledge?
Negative	Are more charges being filed? Does the headline lack hope?	Does this hurt Brazil's image?	Does it make you want to sell/avoid Brazilian products?

```
## New names:
## * `` -> `..7`
```

```r
#Data has been read in under the variable name "medmon"
library(plyr)
library(readxl)
#tidying, establishing the correct variable types:
medmon$Sources = as.factor(medmon$Sources)
medmon$Pillar = as.factor(medmon$Pillar)
medmon$Topic = as.factor(medmon$Topic)
medmon$Impact = as.factor(medmon$Impact)
medmon$Date = as.Date(medmon$Date, origin="1900-01-01")
tail(medmon)
```

```
## # A tibble: 6 x 7
##   Title                   Sources  Date       Pillar Topic      Impact ..7
##   <chr>                   <fct>    <date>     <fct>  <fct>      <fct>
<chr>
## 1 Goldman Sees Iron Ore~ Bloombe~ 2017-11-29 Econo~ <NA>       Neutr~ <NA>
## 2 Asian groups vie for ~ Reuters  2017-11-29 Econo~ Infrastru~ Posit~ <NA>
## 3 - Brazil posts larger~ Reuters  2017-11-29 Econo~ <NA>       Posit~ <NA>
## 4 - Plan to help Brazil~ Reuters  2017-11-29 Econo~ <NA>       Neutr~ <NA>
## 5 - Brazil lower house ~ Reuters  2017-11-29 Polit~ Green Eco~ Posit~ <NA>
## 6 - Brazil Senate appro~ Reuters  2017-11-29 Econo~ Politics   Posit~ <NA>
```

```r
#From here, the goal is to get a daily count for _impact_.
library(plyr)
counter = count(medmon$Date)

mm_pos= subset(medmon, Impact == "Positive")
mm_neg= subset(medmon, Impact == "Negative")
c_pos = count(mm_pos$Date)
c_neg = count(mm_neg$Date)
counter = merge(counter, c_pos, by="x", all=TRUE)
counter = merge(counter, c_neg, by="x", all=TRUE)
colnames(counter)= c("Date","Total","Pos","Neg")

head(counter)
```

```
##         Date Total Pos Neg
## 1 2017-01-05    14   3   7
## 2 2017-01-06    11   1   7
## 3 2017-01-09     9  NA   3
## 4 2017-01-10     6   4   1
## 5 2017-01-11    12   5   2
## 6 2017-01-12    10   3   3
```

We have the initial structure now, but we need to retain counts of "0" for when we begin graphing. Then and only then we can subtract from our Total column to get the final Neutral column:

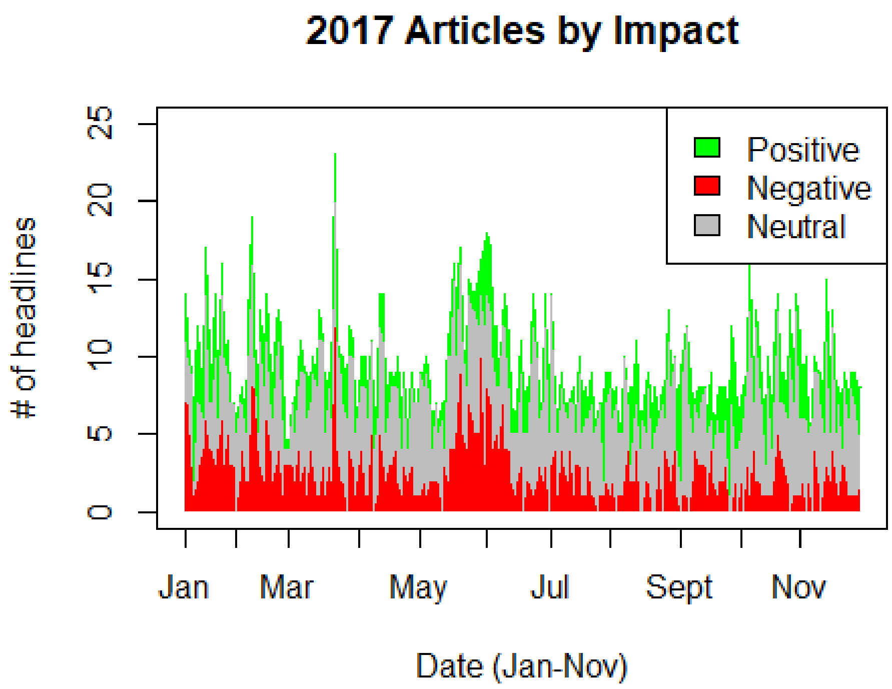
```r
counter[is.na(counter)] = 0
counter$Neu = counter$Total - counter$Pos - counter$Neg
head(counter)
```

```
##         Date Total Pos Neg Neu
## 1 2017-01-05    14   3   7   4
## 2 2017-01-06    11   1   7   3
## 3 2017-01-09     9   0   3   6
## 4 2017-01-10     6   4   1   1
## 5 2017-01-11    12   5   2   5
## 6 2017-01-12    10   3   3   4
```

Visualization

I began without using ggplot. I manually added tick marks for each month; Although there may have been a way to code this, I called upon Occam's razor and found the 11 observations where the month changed.

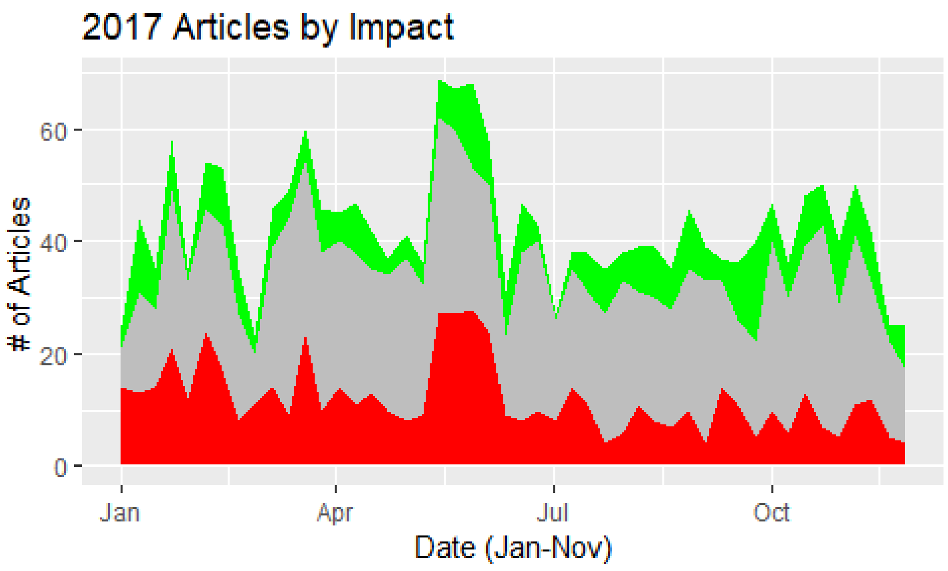
```
c_cols = c("Black", "Black", "Green", "Red", "Gray")
plot.ts(counter$Total, ylim= c(0, 25), col="Green",
        xaxt= "n", ylab="# of headlines", xlab= "Date (Jan-Nov)")
    title(main= "2017 Articles by Impact")
    axis(1, at=c(1, 18, 35, 58, 78, 100, 121, 140, 163, 183, 202), labels =
c("Jan","Feb","Mar","Apr", "May", "Jun", "Jul","Aug","Sept","Oct", "Nov"))
#We begin plotting by stacking each layer with an appropriate color label
    polygon(x=c(1:222, 222:1), border = NA, y=c(counter$Total,
rev(counter$Neg+counter$Neu)), col= "Green")
    polygon(x=c(1:222, 222:1), border = NA, y=c((counter$Neg + counter$Neu),
rev(counter$Neg)), col= "Grey")
    polygon(x=c(1:222, 222:1), border = NA, y=c((counter$Neg), rep(0,
times=222)), col= "Red")
    legend("topright", c("Positive", "Negative", "Neutral"), fill =
c("Green", "Red", "Gray"))
```



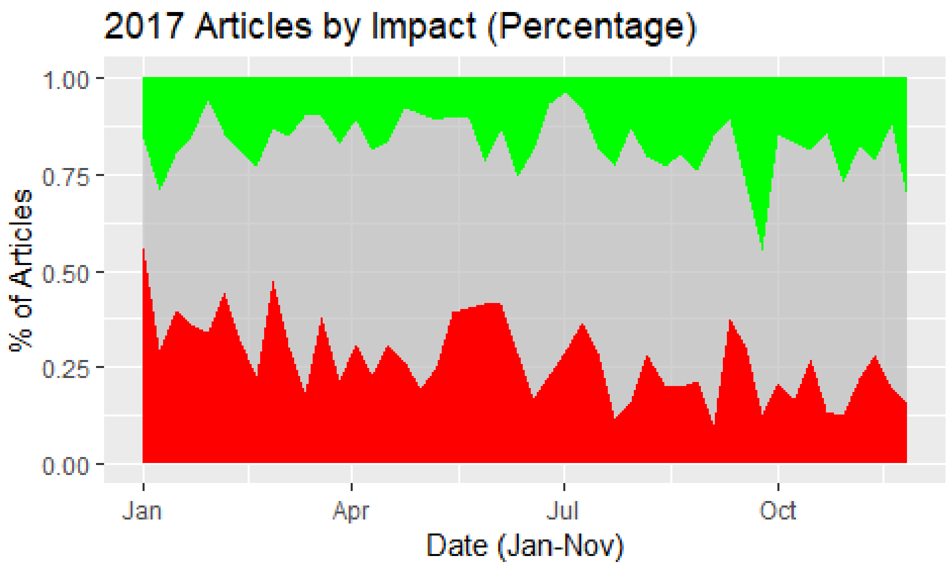
Already we can see some interesting trends we will want to explore. From here we will switch to the more efficient code of ggplot. To make trends more visible to humans, we're going to bin data by the week:

```
#Preparing the weekly sums
counter$DateW=as.Date(cut.Date(counter$Date, breaks="week", start.on.monday =
F))
counterW=aggregate(cbind(counter$Total, counter$Pos, counter$Neg,
counter$Neu)~DateW,data=counter,FUN = sum)
colnames(counterW)=c("DateW", "Total", "Pos", "Neg", "Neu")
```

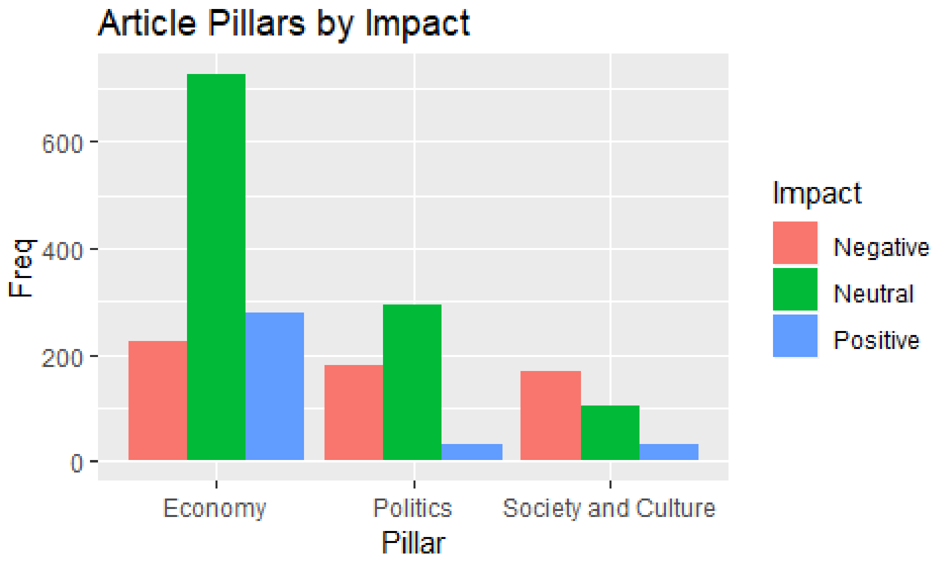
```
#Plotting begins
g1=ggplot(data=counterW, aes(x=DateW, y=Total)) +
  geom_ribbon(aes(ymin=0, ymax=counterW$Neg),fill="red")+
  geom_ribbon(aes(ymin=counterW$Neg, ymax=(counterW$Neg+counterW$Neu)),
fill="gray") +
  geom_ribbon(aes(ymin=(counterW$Total-counterW$Pos), ymax=counterW$Total),
fill = "green") +
  labs(title="2017 Articles by Impact", x="Date (Jan-Nov)", y="# of
Articles")

#Percentage of totals:

counterW$posp=counterW$Pos/counterW$Total
counterW$negp=counterW$Neg/counterW$Total
counterW$neup=counterW$Neu/counterW$Total
#Plotting continues
g2=ggplot(data=counterW, aes(x=DateW, y=1)) +
  geom_ribbon(aes(ymin=0, ymax=counterW$negp),fill="red")+
  geom_ribbon(aes(ymin=counterW$negp, ymax=(counterW$neg+counterW$neup)),
fill="gray", alpha=0.7) +
  geom_ribbon(aes(ymin=(1-counterW$posp), ymax=1), fill = "green") +
  labs(title="2017 Articles by Impact (Percentage)", x="Date (Jan-Nov)", y="%
of Articles")
g1
```



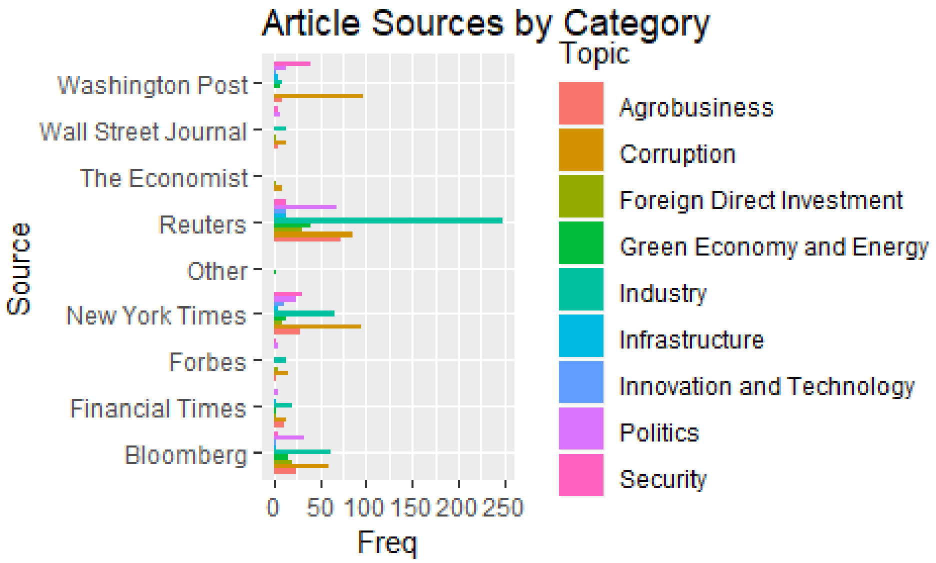
```
g2
```



```r
#Qualitative factor breakdown
pill.imp = table(medmon$Pillar, by=medmon$Impact)
pill.imp=as.data.frame(pill.imp)
colnames(pill.imp)= c( "Pillar", "Impact", "Freq")
ggplot(pill.imp, aes(Pillar, Freq, fill=Impact))+geom_bar(stat="identity",
position = position_dodge()) +labs(title ="Article Pillars by Impact")
```



```r
#A list of sources by categorized topic

source.top = table(medmon$Sources, by=medmon$Topic)
source.top = as.data.frame(source.top[])
colnames(source.top) = c("Source", "Topic", "Freq")
ggplot(source.top, aes(Source, Freq, fill=Topic))+geom_bar(stat="identity",
position = position_dodge())+ coord_flip()+labs(title="Article Sources by
Category")
```



As we can see, there are some stark differences in the number of articles involving our subject country within each media source.

Text Mining

Let's say we were now interested in finding the most frequent words within headlines of the year.

```
library(tm)

## Loading required package: NLP

##
## Attaching package: 'NLP'

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

library(SnowballC)
library(data.table)

corp=medmon$Title
corpstring=paste(corp,collapse = "", sep = " ")
corpstring= VectorSource(corpstring)
corp=VCorpus(corpstring )

#Transformations to isolate the key root words of each headline
corp=tm_map(corp, removePunctuation)
corp=tm_map(corp, content_transformer(tolower))
corp=tm_map(corp, removeWords, stopwords("english"))
corp=tm_map(corp, stemDocument)
matrix=DocumentTermMatrix(corp)
#Determining most common word stems
```

```
wordfreq = findMostFreqTerms(matrix, 100)
wordfreq=wordfreq$`1`
wordfreq=as.data.frame(wordfreq)
wordfreq=setDT(wordfreq, keep.rownames=T)
head(wordfreq, 25)
```

```
##             rn wordfreq
##  1:     brazil      456
##  2:        say      152
##  3:      temer      122
##  4:     presid       76
##  5:        new       73
##  6:  brazilian       71
##  7:      court       66
##  8:    pension       65
##  9:    corrupt       63
## 10:     reform       63
## 11:       bank       62
## 12:        see       62
## 13:      probe       53
## 14:   petrobra       52
## 15:        cut       51
## 16:      polic       51
## 17:       rate       51
## 18:    billion       50
## 19:        may       49
## 20:       sale       48
## 21:      graft       47
## 22:    brazil⍰       46
## 23:      polit       46
## 24:        rio       44
## 25:    central       43
##             rn wordfreq
```

From here we can filter keywords and find the key terms readers have seen with their news over the year. These are simple examples of the insight that can be gained from such data; Much more robust collection methods can lead to an even fuller picture of a country’s portrayal in the media.