Global Terrorism

STAT 4185 Project Final

Group Four; Tom Kennon, Adam Busa, Asa Brigandi, Jiapeng Guo

28 April 2018

Contents

1 Introduction 2

2 Data Wrangling 2

3 Analysis 6

4 Predictive Analytics 19

4.1 Predicting "ISIL" Terrorist Group 19

4.1.1 Naive Bayes 19

4.2 Predicting Mass Terrorist Attacks (>3 Deaths) 22

4.2.1 Decision Tree 23

4.2.2 Random Forest Model 25

4.2.3 Model Selection 27

4.3 Predicting "Successful" Terrorist Attacks 27

4.3.1 Naive Bayes 28

4.3.2 Decision Tree 28

4.3.3 Random Forest Model 29

4.3.4 Logistic Regression 30

4.3.5 Model Selection 33

5 Conclusion 34

1 Introduction

Our project is focused on analyzing global terrorism data. With many terrorist attacks prevalent in the news since the early 2000s as well as imposing threats of attacks being a controversial topic for a newly elected president, this an important topic at the forefront of people's minds. The data source we used for our analysis is The Global Terrorism Database (GTD) which is an open source database that contains information about specific global terrorism incidents spanning from 1970 to 2016. The dataset is available at: https://www.kaggle.com/START-UMD/gtd. The total dataset contains 170,350 records with 135 variables. Each record contains the date of incident, location of the incident, information about the target, information about the perpetrators etc. Some of the variables are repetitive, do not tell provide much insight for the data, and/or are riddled with missing values as this is a real-world dataset.

Some of exploratory questions we set out to accomplish in this project are:

- What are the factors (possibly weapon type, attack type, HDI, etc.) that predict the terrorist group that perpetrated an attack?
- Investigate the trends in mass terrorism (terrorist attacks with many people killed) in the dataset. Identify any possible unique characteristics of mass terrorism.
- What factors may predict the success rate of attacks? What are the significant drivers for a terrorist attack to be "successful"? Is there a bias pertaining to the recency of an attack? [Some possible solutions: logit/probit prediction, random forest model]

2 Data Wrangling

The first step to any data science project is to read in the data. Here is the global terrorism dataset that we called: "gt" provided in a tibble format to improve readability in its data structure.

```r
## Reads in data
gt <- read.csv("globalterrorismdb_0617dist.csv")

library(tidyverse)

library(knitr)

## Shows dataset structure
as_tibble(gt)
```

```
## # A tibble: 170,350 x 135
##         eventid iyear imonth  iday approxdate extended resolution country
##           <dbl> <int>  <int> <int> <fct>         <int> <fct>        <int>
##   1 197000000001  1970      7     2 ""                0 ""              58
##   2 197000000002  1970      0     0 ""                0 ""             130
##   3 197001000001  1970      1     0 ""                0 ""             160
##   4 197001000002  1970      1     0 ""                0 ""              78
##   5 197001000003  1970      1     0 ""                0 ""             101
##   6 197001010002  1970      1     1 ""                0 ""             217
##   7 197001020001  1970      1     2 ""                0 ""             218
##  8 197001020002  1970      1     2 ""                0 ""             217
##  9 197001020003  1970      1     2 ""                0 ""             217
## 10 197001030001  1970      1     3 ""                0 ""             217
## # ... with 170,340 more rows, and 127 more variables: country_txt <fct>,
## #   region <int>, region_txt <fct>, provstate <fct>, city <fct>,
## #   latitude <dbl>, longitude <dbl>, specificity <int>, vicinity <int>,
## #   location <fct>, summary <fct>, crit1 <int>, crit2 <int>, crit3 <int>,
## #   doubtterr <int>, alternative <int>, alternative_txt <fct>,
## #   multiple <int>, success <int>, suicide <int>, attacktype1 <int>,
## #   attacktype1_txt <fct>, attacktype2 <int>, attacktype2_txt <fct>,
## #   attacktype3 <int>, attacktype3_txt <fct>, targtype1 <int>,
## #   targtype1_txt <fct>, targsubtype1 <int>, targsubtype1_txt <fct>,
## #   corp1 <fct>, target1 <fct>, natlty1 <int>, natlty1_txt <fct>,
## #   targtype2 <int>, targtype2_txt <fct>, targsubtype2 <int>,
## #   targsubtype2_txt <fct>, corp2 <fct>, target2 <fct>, natlty2 <int>,
## #   natlty2_txt <fct>, targtype3 <int>, targtype3_txt <fct>,
## #   targsubtype3 <int>, targsubtype3_txt <fct>, corp3 <fct>,
## #   target3 <fct>, natlty3 <int>, natlty3_txt <fct>, gname <fct>,
## #   gsubname <fct>, gname2 <fct>, gsubname2 <fct>, gname3 <fct>,
## #   gsubname3 <fct>, motive <fct>, guncertain1 <int>, guncertain2 <int>,
## #   guncertain3 <int>, individual <int>, nperps <int>, nperpcap <dbl>,
## #   claimed <int>, claimmode <int>, claimmode_txt <fct>, claim2 <int>,
## #   claimmode2 <int>, claimmode2_txt <fct>, claim3 <int>,
## #   claimmode3 <int>, claimmode3_txt <fct>, compclaim <int>,
## #   weaptype1 <int>, weaptype1_txt <fct>, weapsubtype1 <int>,
## #   weapsubtype1_txt <fct>, weaptype2 <int>, weaptype2_txt <fct>,
## #   weapsubtype2 <int>, weapsubtype2_txt <fct>, weaptype3 <int>,
## #   weaptype3_txt <fct>, weapsubtype3 <int>, weapsubtype3_txt <fct>,
## #   weaptype4 <int>, weaptype4_txt <fct>, weapsubtype4 <int>,
## #   weapsubtype4_txt <fct>, weapdetail <fct>, nkill <int>, nkillus <int>,
## #   nkillter <int>, nwound <int>, nwoundus <int>, nwoundte <int>,
## #   property <int>, propextent <int>, propextent_txt <fct>,
## #   propvalue <dbl>, ...
```

It is often useful to include external datasets to highlight insights that may have even been unknown beforehand with just the first dataset as reference. Our first helpful dataset is a simple gross count of total world population for each year in the dataset 1960-2016. This is helpful for getting a better understanding of per capita statistics for each year rather than only total numbers for each year. The second external dataset we found helpful was a Gross Domestic Product (GDP) dataset for each country in the world. We have added the GDP of the target country from the incident year into our main dataset. The GDP data was taken from the World Bank database. We believe that GDP may be an important tool for predicting terrorist attacks. Both of these dataset's structures are shown in tibble format below.

```r
library("readxl")

## Reads in data
popworld <- read_excel("world population.xlsx")
```

```r
## Shows dataset structure
as_tibble(popworld)
```

```
## # A tibble: 57 x 2
##    population  year
##         <dbl> <dbl>
##  1 3034193297  1960
##  2 3075115342  1961
##  3 3127961482  1962
##  4 3192794384  1963
##  5 3258201476  1964
##  6 3324951621  1965
##  7 3394864530  1966
##  8 3464439525  1967
##  9 3534821115  1968
## 10 3609383725  1969
## # ... with 47 more rows
```

```r
## Reads in data
gdp<-read.csv("GDP1.csv", check.names = FALSE)

## Shows dataset structure
as_tibble(gdp)
```

```
## # A tibble: 264 x 62
##    `Country Name` `Country Code` `Indicator Name` `Indicator Code`  `1960`
##    <fct>          <fct>          <fct>            <fct>              <dbl>
##  1 Aruba          ABW            GDP (current US~ NY.GDP.MKTP.CD   NA
##  2 Afghanistan    AFG            GDP (current US~ NY.GDP.MKTP.CD    5.38e8
##  3 Angola         AGO            GDP (current US~ NY.GDP.MKTP.CD   NA
##  4 Albania        ALB            GDP (current US~ NY.GDP.MKTP.CD   NA
##  5 Andorra        AND            GDP (current US~ NY.GDP.MKTP.CD   NA
##  6 Arab World     ARB            GDP (current US~ NY.GDP.MKTP.CD   NA
##  7 United Arab E~ ARE            GDP (current US~ NY.GDP.MKTP.CD   NA
##  8 Argentina      ARG            GDP (current US~ NY.GDP.MKTP.CD   NA
##  9 Armenia        ARM            GDP (current US~ NY.GDP.MKTP.CD   NA
## 10 American Samoa ASM            GDP (current US~ NY.GDP.MKTP.CD   NA
## # ... with 254 more rows, and 57 more variables: `1961` <dbl>,
## #   `1962` <dbl>, `1963` <dbl>, `1964` <dbl>, `1965` <dbl>, `1966` <dbl>,
## #   `1967` <dbl>, `1968` <dbl>, `1969` <dbl>, `1970` <dbl>, `1971` <dbl>,
## #   `1972` <dbl>, `1973` <dbl>, `1974` <dbl>, `1975` <dbl>, `1976` <dbl>,
## #   `1977` <dbl>, `1978` <dbl>, `1979` <dbl>, `1980` <dbl>, `1981` <dbl>,
## #   `1982` <dbl>, `1983` <dbl>, `1984` <dbl>, `1985` <dbl>, `1986` <dbl>,
## #   `1987` <dbl>, `1988` <dbl>, `1989` <dbl>, `1990` <dbl>, `1991` <dbl>,
## #   `1992` <dbl>, `1993` <dbl>, `1994` <dbl>, `1995` <dbl>, `1996` <dbl>,
## #   `1997` <dbl>, `1998` <dbl>, `1999` <dbl>, `2000` <dbl>, `2001` <dbl>,
## #   `2002` <dbl>, `2003` <dbl>, `2004` <dbl>, `2005` <dbl>, `2006` <dbl>,
## #   `2007` <dbl>, `2008` <dbl>, `2009` <dbl>, `2010` <dbl>, `2011` <dbl>,
## #   `2012` <dbl>, `2013` <dbl>, `2014` <dbl>, `2015` <dbl>, `2016` <dbl>,
## #   `2017` <lgl>
```

Now the next logical step in the data wrangling process is to join these two datasets to the global terrorism dataset which we do in this next R chunk. We also provided code with output that shows the merges were successful. The sample_n function is taking a random sample of 6 of the events with their years and populations. The head function is showing the first 6 events with their country and country's respective GDPs.

```
## Joins gt data and popworld data by year as "popworld2"
popworld2 <- inner_join(popworld,gt,by=c("year"="iyear"))

## Evidence that our join was successful
set.seed(12345)
sample_n(select(popworld2, eventid, year, population),6)
```

```
## # A tibble: 6 x 3
##        eventid  year population
##          <dbl> <dbl>      <dbl>
## 1 201311030020  2013 7182860115
## 2 201506170047  2015 7355220412
## 3 201404050030  2014 7268986176
## 4 201507310083  2015 7355220412
## 5 200504090006  2005 6517020798
## 6 198606200001  1986 4928822143
```

```
## Joins gt data and gdp data by country as "master"
## a bit more complicated because the gdp dataset was less straightforward than the popworld
library("reshape2")
gdp2<-melt(gdp, id.vars = c("Country Name", "Country Code", "Indicator Name", "Indicator Code"
gdp3<-gdp2 %>% select("Country Name", "variable", "value")
names(gdp3)[1]<-"country_txt"
names(gdp3)[2]<-"iyear"
names(gdp3)[3]<-"country_gdp"
gt$iyear<-as.character(gt$iyear)
master<-gt %>% left_join(gdp3, by = c("country_txt", "iyear"))
```

```
## Warning: Column `country_txt` joining factors with different levels,
## coercing to character vector

## Warning: Column `iyear` joining character vector and factor, coercing into
## character vector
```

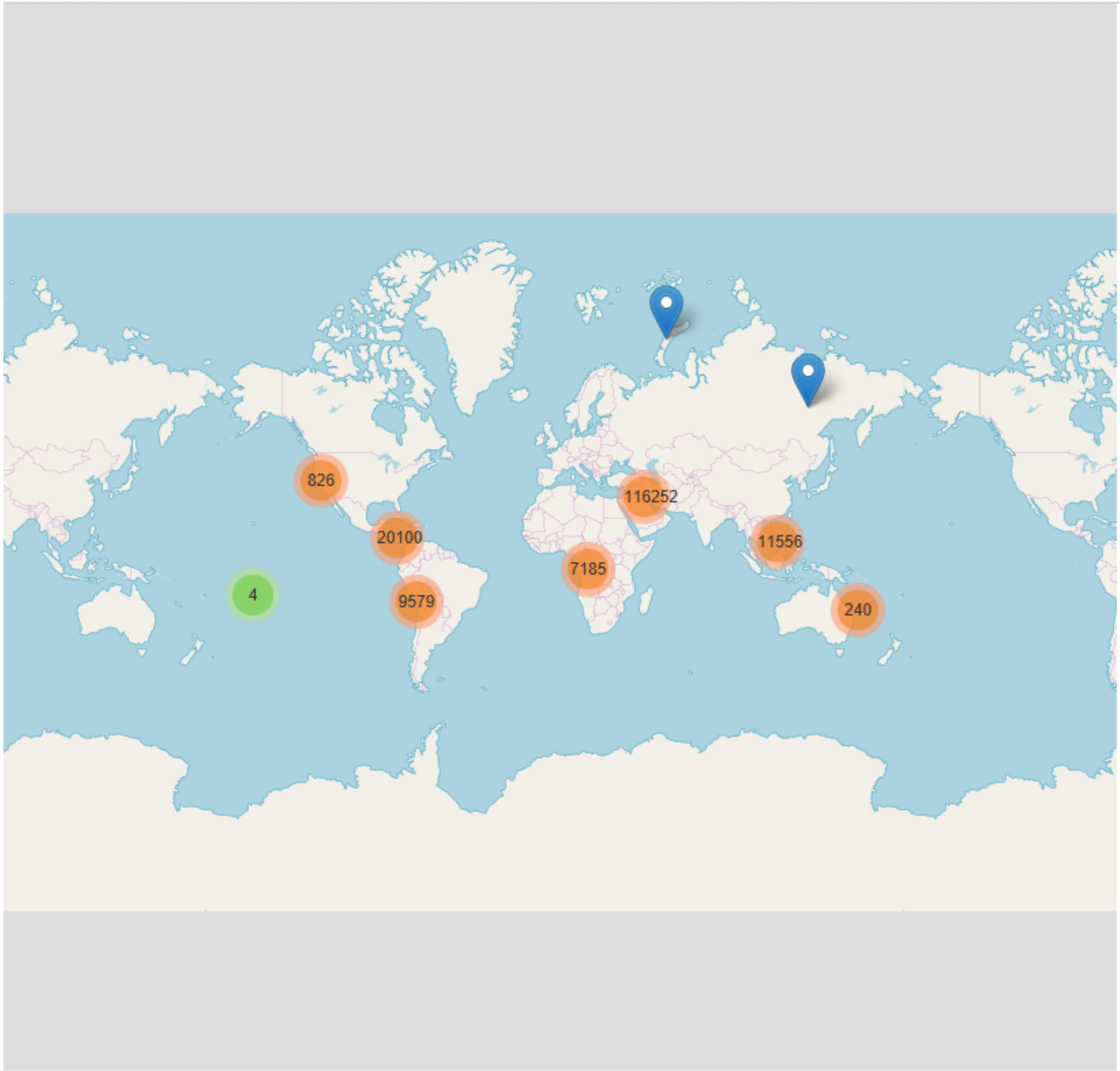
```
## Evidence that our join was successful
head(select(master, eventid, country, country_gdp))
```

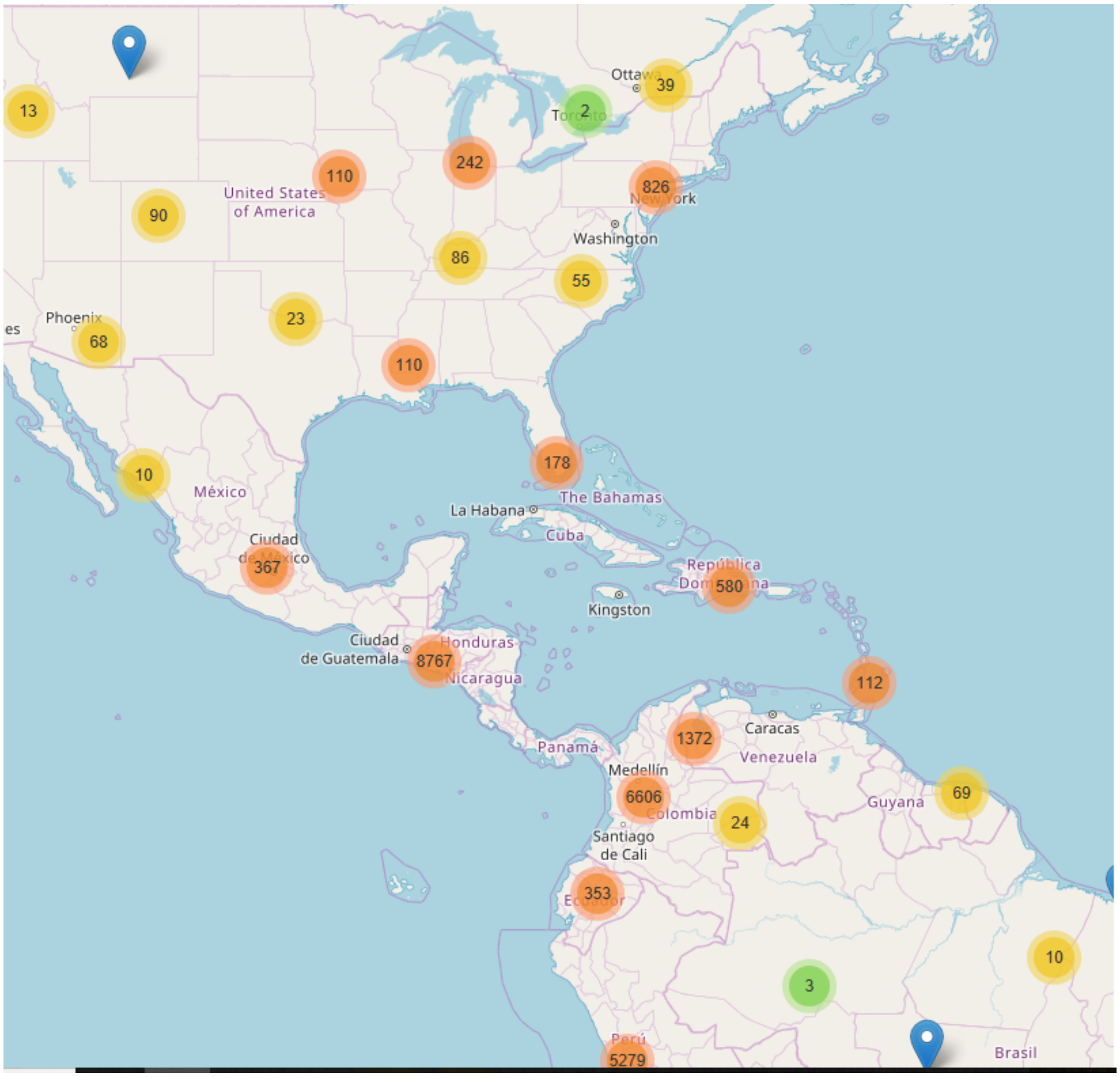
```
##       eventid country  country_gdp
## 1 1.97000e+11      58 1.485400e+09
## 2 1.97000e+11     130 3.554171e+10
## 3 1.97001e+11     160 6.687205e+09
## 4 1.97001e+11      78 1.313986e+10
## 5 1.97001e+11     101 2.115140e+11
## 6 1.97001e+11     217 1.075880e+12
```

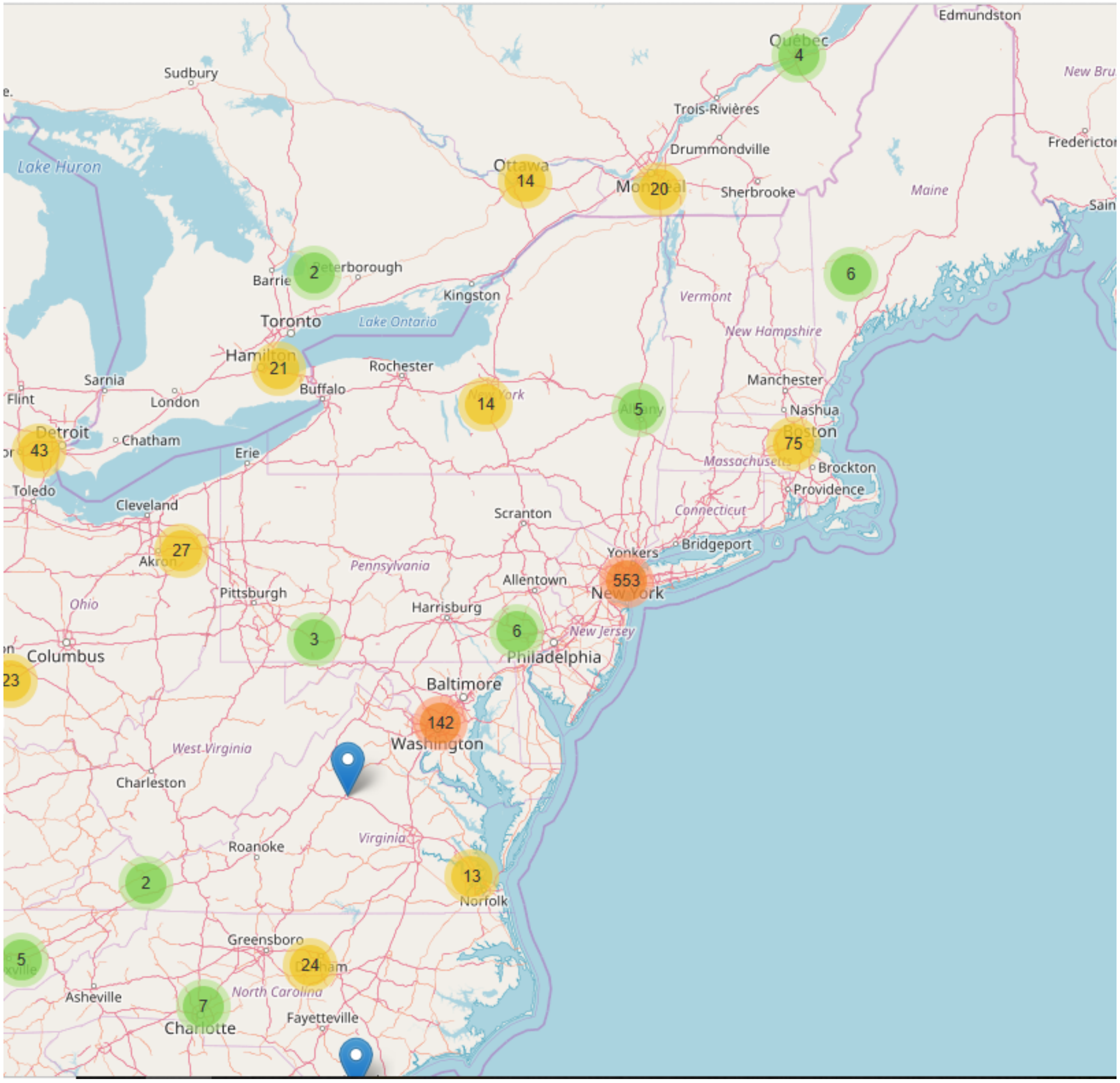
3 Analysis

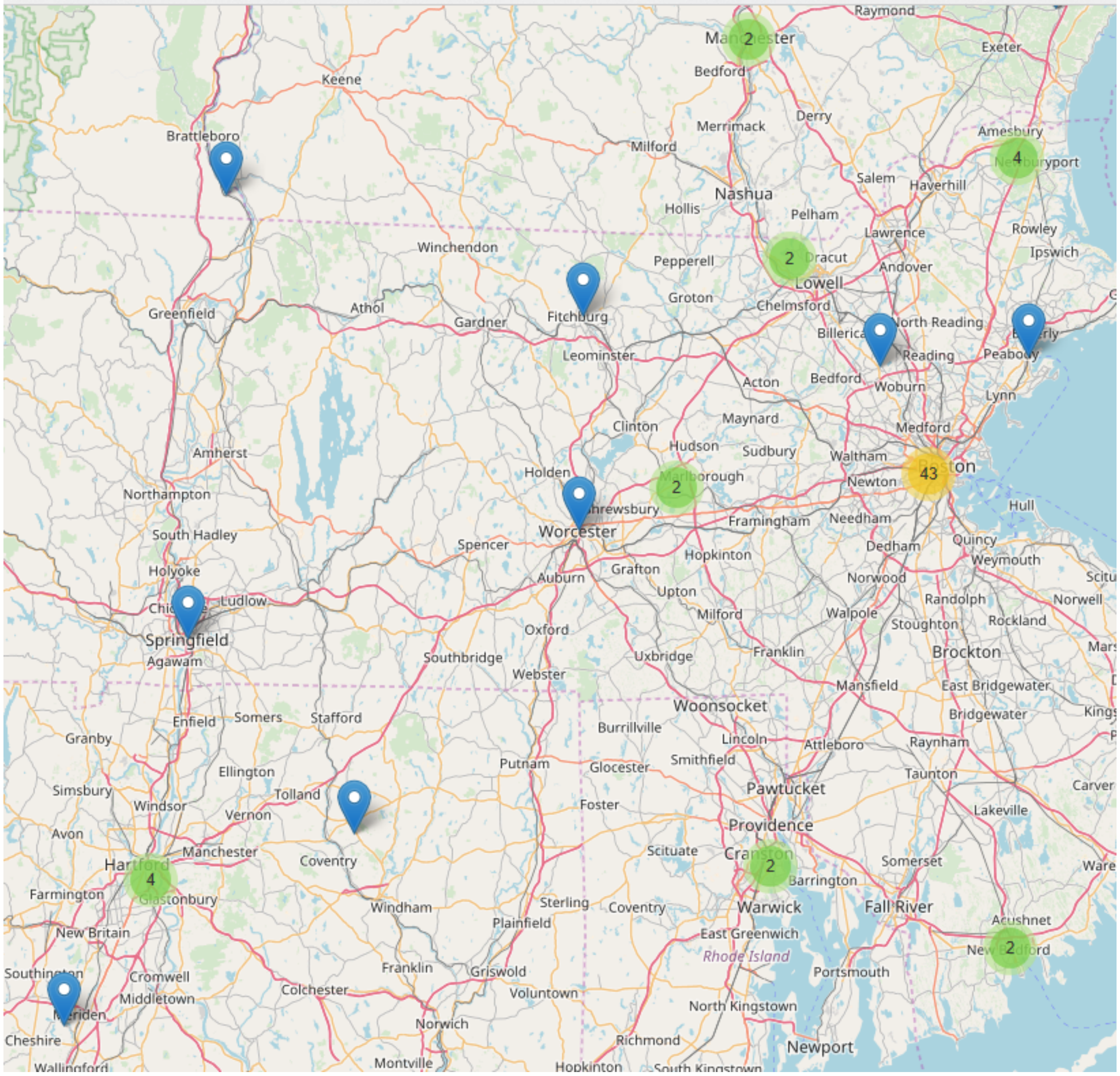
Interactive Leaflet Map of Global Terrorist Attacks with Useful Info. on Popup Markers

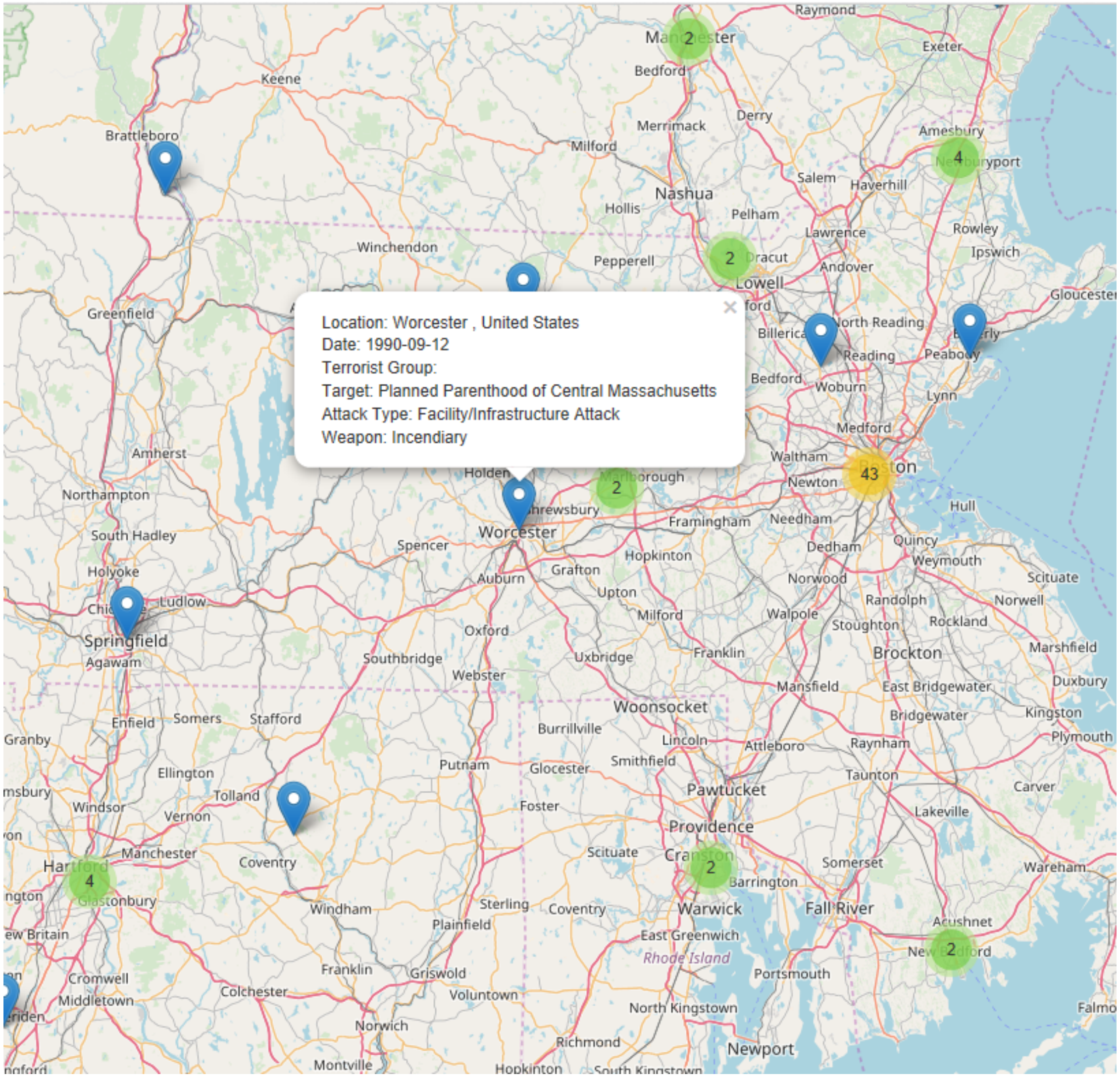
```
## Keeping columns of use to me
gt$date <- as.Date(with(gt, paste(imonth,iday,iyear,sep="-")), "%m-%d-%Y")

vars = gt %>% select(eventid,latitude,longitude,attacktype1_txt,
                            weaptype1_txt,target1,city,country_txt,date)

library(leaflet)
library(htmltools)

map <- leaflet(vars) %>%
  addTiles(urlTemplate = "https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png")
map %>% addMarkers(lat = ~latitude, lng = ~longitude,
        clusterOptions = markerClusterOptions(),
        popup = paste("Location:", vars$city, ",", vars$country_txt, "<br>",
                           "Date:", vars$date, "<br>",
                           "Terrorist Group:", vars$gname, "<br>",
                           "Target:", vars$target1, "<br>",
                           "Attack Type:", vars$attacktype1_txt, "<br>",
                           "Weapon:", vars$weaptype1_txt))
```


826
20100
9579
4
116252
7185
11556
240


Ottawa
39
13
2
Toronto
242
110
826
New York
United States
of America
90
Washington
86
55
Phoenix
es
23
68
110
178
10
The Bahamas
México
La Habana
Cuba
Ciudad
de México
367
República
Dominicana
580
Kingston
Ciudad
de Guatemala
Honduras
8767
Nicaragua
112
Caracas
1372
Panamá
Venezuela
Medellín
6606
69
Guyana
Colombia
24
Santiago
de Cali
353
Ecuador
10
3
Perú
5279
Brasil


Edmundston
Québec
4
Sudbury
New Bru
Trois-Rivières
Drummondville
Frederictor
Lake Huron
Ottawa
14
Montreal
20
Sherbrooke
Maine
Sain
Peterborough
2
Barrie
6
Kingston
Vermont
Toronto
Lake Ontario
New Hampshire
Hamilton
21
Rochester
New York
14
Sarnia
Buffalo
Manchester
Flint
London
5
Albany
Nashua
Detroit
43
Chatham
Boston
75
Erie
Massachusetts
Brockton
Toledo
Providence
Cleveland
Scranton
Connecticut
Yonkers
Bridgeport
27
Akron
Pennsylvania
Allentown
553
New York
Pittsburgh
Ohio
Harrisburg
6
New Jersey
3
Columbus
Philadelphia
23
Baltimore
142
Washington
West Virginia
Charleston
Virginia
Roanoke
13
Norfolk
2
Greensboro
5
24
Durham
North Carolina
Asheville
7
Charlotte
Fayetteville


2
4
2
43
2
4
2
2
Manchester
Nashua
Lowell
Worcester
Springfield
Hartford
Boston
Providence
Pawtucket
Woonsocket
Brockton
Fall River
Newport
Rhode Island



Location: Worcester , United States
Date: 1990-09-12
Terrorist Group:
Target: Planned Parenthood of Central Massachusetts
Attack Type: Facility/Infrastructure Attack
Weapon: Incendiary

Above is an interactive geospatial visualization showing all of the terrorist attacks in this database. Because this is an html document, you can use the interactive map to find whatever terrorist attacks that interest you. When a popup is chosen, the location, date, terrorist group, target name, attack type, and weapon type are displayed. This map highlights where many terrorist attacks are. The Middle East area has a lot of attacks. To see a more detailed number for each country or region, see the following two outputs.

Rank of Number of Attacks for each Country with Descending Order.

```
## Rank of number of attacks for each country with descending order.
gt %>%
  group_by(country_txt) %>%
  summarise( nr_of_attacks = n()) %>%
  arrange(desc(nr_of_attacks)) %>%
  head(n=10)
```

```
## # A tibble: 10 x 2
##    country_txt    nr_of_attacks
##    <fct>                  <int>
##  1 Iraq                   22130
##  2 Pakistan               13634
##  3 Afghanistan            11306
##  4 India                  10978
##  5 Colombia                8163
##  6 Philippines             6212
##  7 Peru                    6088
##  8 El Salvador             5320
##  9 United Kingdom          5098
## 10 Turkey                  4106
```

As evidenced by the output above, Iraq has the most number of attacks, followed by Pakistan, Afghanistan, etc. These countries have been known to have had a lot of terrorist attacks and this is backed in our findings as well.

Rank of Number of Attacks for each Region with Descending Order.

```
## Rank by region
gt %>%
  group_by(region_txt) %>%
  summarise(attacks = n()) %>%
  arrange(desc(attacks)) %>%
  head(n=10)
```

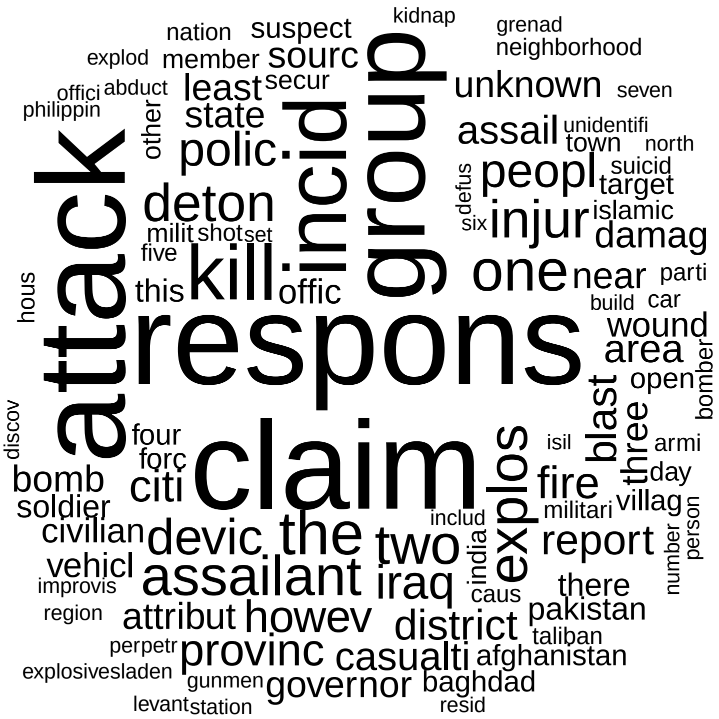
```
## # A tibble: 10 x 2
##    region_txt                  attacks
##    <fct>                         <int>
##  1 Middle East & North Africa    46511
##  2 South Asia                    41497
##  3 South America                 18762
##  4 Western Europe                16307
##  5 Sub-Saharan Africa            15491
##  6 Southeast Asia                11453
##  7 Central America & Caribbean   10340
##  8 Eastern Europe                 5031
##  9 North America                  3346
## 10 East Asia                       794
```

As evidenced by the output above, the Middle East and North Africa and South Asia have the most number of attacks which has been echoed in world news lately due to shifting political regimes.

Wordcloud of Summaries of Global Terrorist Attacks Since 1970

```
library(tm)
library(SnowballC)
library(wordcloud)

corpus <- Corpus(VectorSource(gt$summary))
corpus <- tm_map(corpus,removePunctuation)
corpus <- tm_map(corpus,removeWords,c("the","this",stopwords("english")))
```

```r
corpus <- tm_map(corpus,stemDocument)

wordcloud(corpus,max.words=100,random.order=FALSE)
```



The above plot shows a wordcloud visualization of detailed summaries of these terrorist attacks. The most frequent word used in the descriptive summaries for these attacks is “response”, followed by “attack”, “claim”, and “group”. This is a good way to visualize what is important to know while investigating terrorist attacks so this aids us in our analysis.

```r
################
### Figure 1 ###
################



## Bar chart
ggplot(gt,aes(attacktype1)) +
  geom_bar(aes(fill=attacktype1_txt)) +
  labs(title="Attack Types", y="Count", x="") +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank()) +
  guides(fill=guide_legend(title="Attack Type"))
```

In Figure 1, we can see that the most popular attack type in terrorist attacks since 1970 are bombings/explosions followed by armed assaults and assassinations respectively. But how do these numbers change over time?

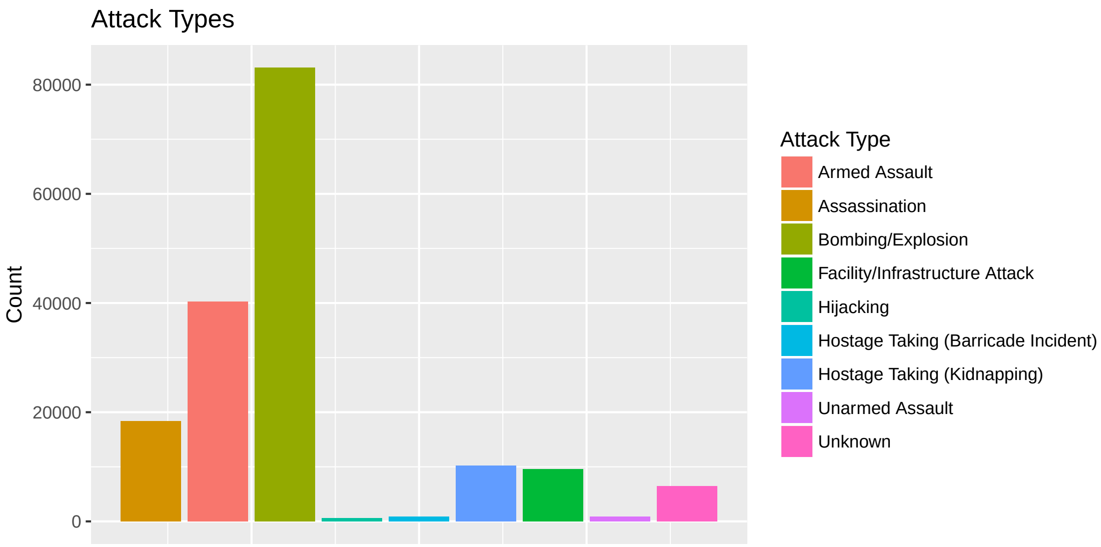


Figure 1: Breakdown of Attack Type.

```
################
### Figure 2 ###
################

cbbPalette <- c("#000000", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#

y<-as.data.frame(table(gt$iyear, gt$attacktype1_txt))
colnames(y) <- c("iyear", "attacktype1_txt", "freq")
ggplot(data=y) +
  geom_point(mapping = aes(x=iyear, y=freq, color=attacktype1_txt))+
  scale_color_manual(values=cbbPalette) +
  scale_x_discrete(breaks=c(1970,1980,1990,2000,2010))
```

```
################
### Figure 3 ###
################

ggplot(data=gt) +
  geom_count(mapping = aes(x=iyear, y=attacktype1_txt)) +
  scale_x_discrete(breaks=c(1970,1980,1990,2000,2010))
```

In Figure 3 and Figure 2, we have created 2 different graphs that illustrate trends in attack types since 1970. In it we can clearly see there has been a large spike in bombings in recent years, as well as a mild spike in armed assault. While most other methods have seen an increase in recent years, these two are by far the greatest increase. This may imply that these are simply the preferred methods by groups that started up since the late 1990's/early 2000's. It could also mean that terrorist groups have just experienced more success with these methods, and thus have been utilizing them more. We have the necessary data to test both of these possibilities. Another interesting variable to look into is the terrorist group.

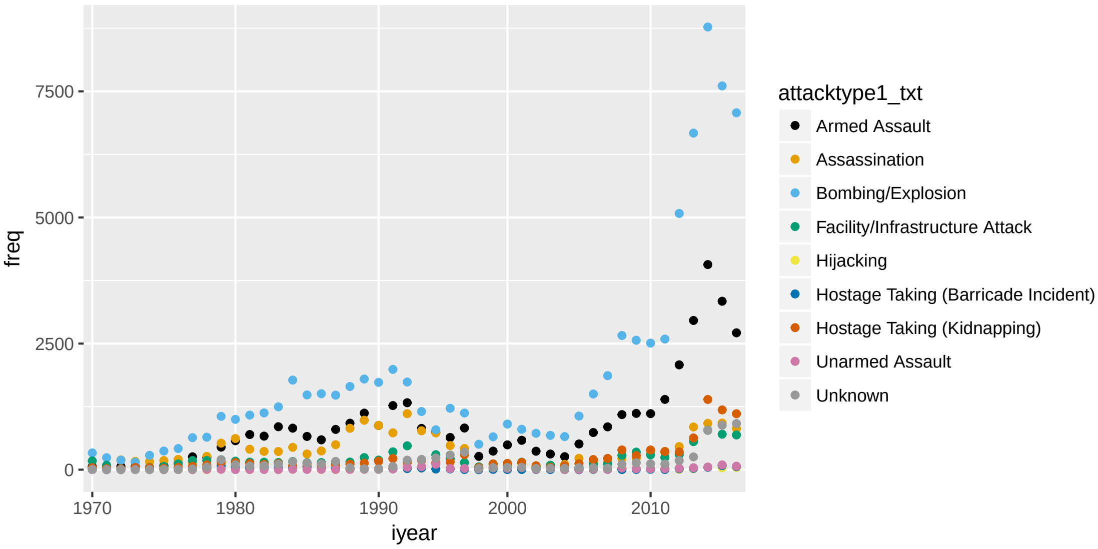


Figure 2: Scatterplot: Trends in Attack Types Since 1970.

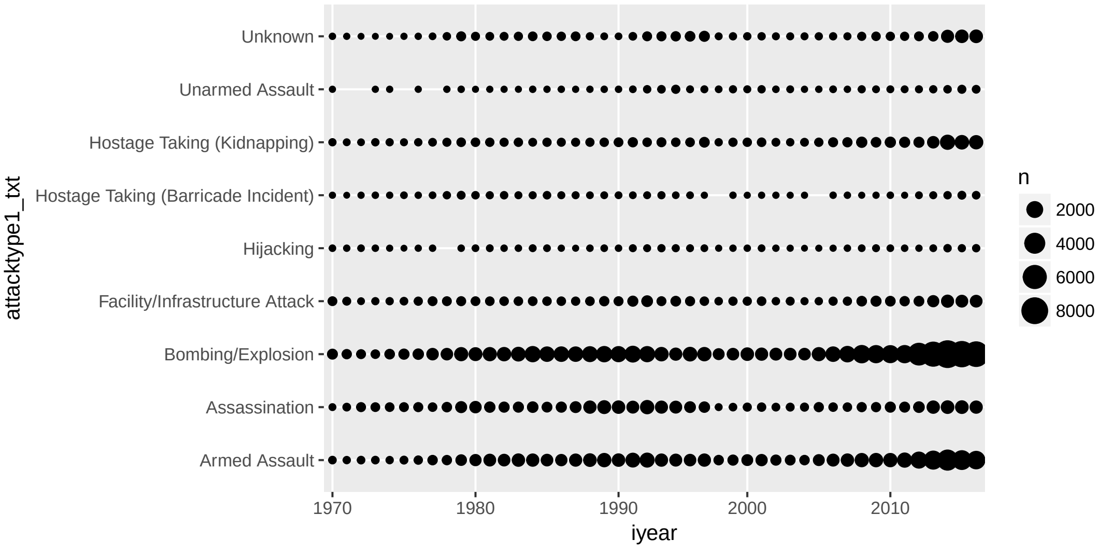


Figure 3: Bubblechart: Trends in Attack Types Since 1970.

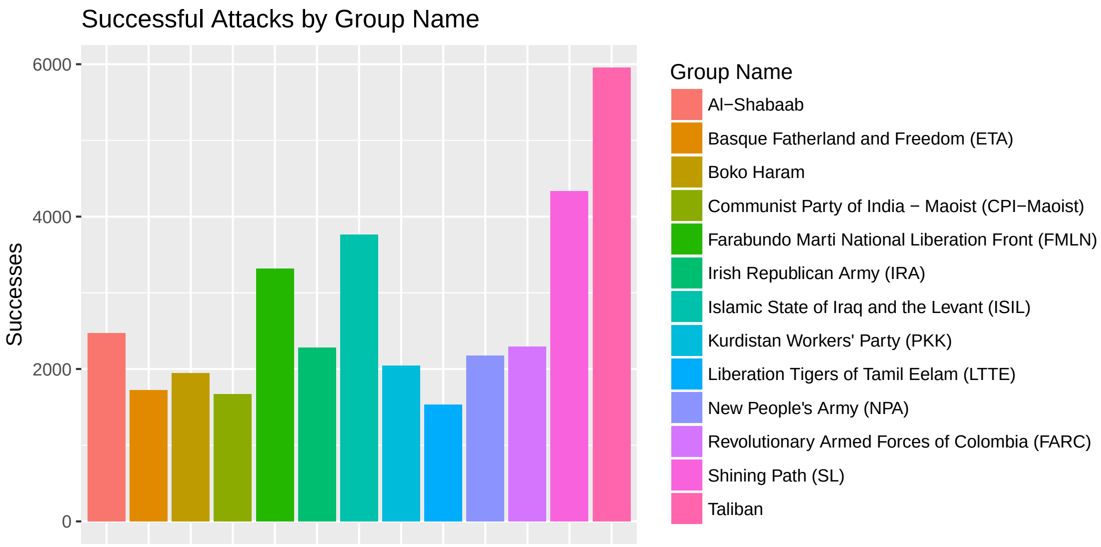


Figure 4: Successful Terrorist Groups

```
################
### Figure 4 ###
################


## Identify major terrorist organizations
terr.groups <- gt %>%
  select(gname, success) %>%
  filter(!gname %in% "Unknown") %>%
  group_by(gname) %>%
  summarise(n.success=sum(success)) %>%
  arrange(desc(n.success)) %>%
  filter(n.success >= 1500) #this will filter out any terrorist groups with less than 1500 suc

terr.groups$n.success <- as.numeric(terr.groups$n.success)

## Bar chart
ggplot(terr.groups, aes(gname, n.success)) +
  geom_bar(stat="identity",aes(fill=gname)) +
  labs(title="Successful Attacks by Group Name", y="Successes", x="") +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank()) +
  guides(fill=guide_legend(title="Group Name"))
```

In Figure 4 we can see the which terrorist groups have the most successful attacks. The Taliban has the most successful attacks at approximately 6000 and the Shining Path (SL) and Islamic State of Iraq and the Levant (ISIL) have the second and third most respectively.

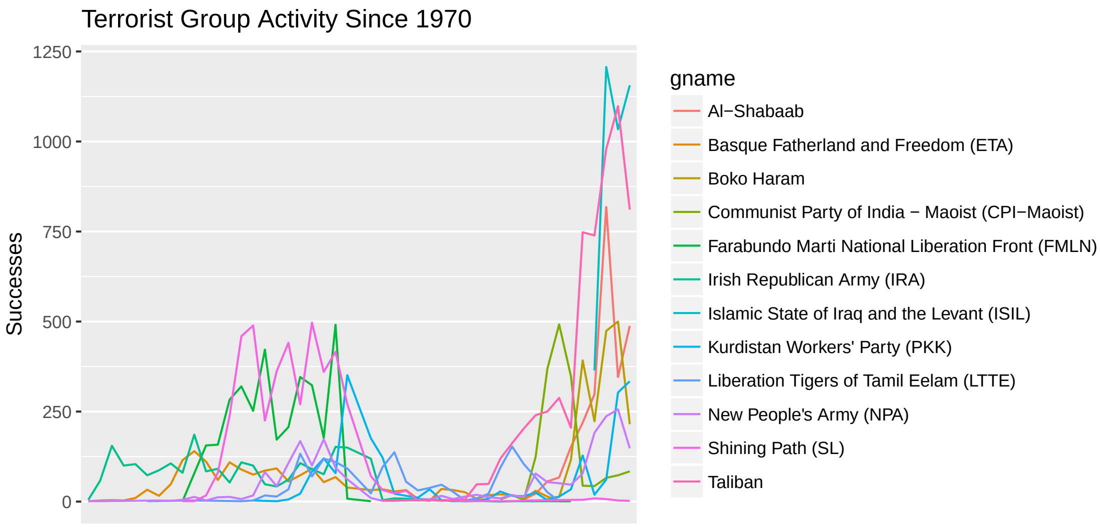


Figure 5: Successful Terrorist Groups Over Time

```
################
### Figure 5 ###
################

## Over time
gname.line <- gt %>%
  select(gname,iyear,success) %>%
  group_by(gname,iyear) %>%
  summarise(n.success = sum(success)) %>%
  filter(gname %in% c("Al-Shabaab","Basque Fatherland and Freedom (ETA)", "Boko Haram", "Commun

gname.line$iyear <- as.numeric(gname.line$iyear)

ggplot(gname.line,aes(iyear, n.success)) +
  geom_line(aes(color=gname)) +
  labs(title="Terrorist Group Activity Since 1970", y="Successes", x="") +
  guides(fill=guide_legend(title="Group Name")) +
  scale_x_discrete(breaks=c(1970,1980,1990,2000,2010))
```

In Figure 5 we can see the which terrorist groups have the most successful attacks over time. This is a useful insight that adds to Figure 4 because now it is evident which groups have had an influx in attacks as of late compared to groups that have died out. Al-Shabaab has only had attacks since 2007, while the Irish Republican Army used to have a lot of terrorist attacks but not as many as of recently. The Islamic State of Iraq and the Levant (ISIL)'s terrorist attacks have only had attacks as of 2013 yet have the third most successful terrorist attacks in data back to 1970 so they are a scary threat.

```
################
### Figure 6 ###
```

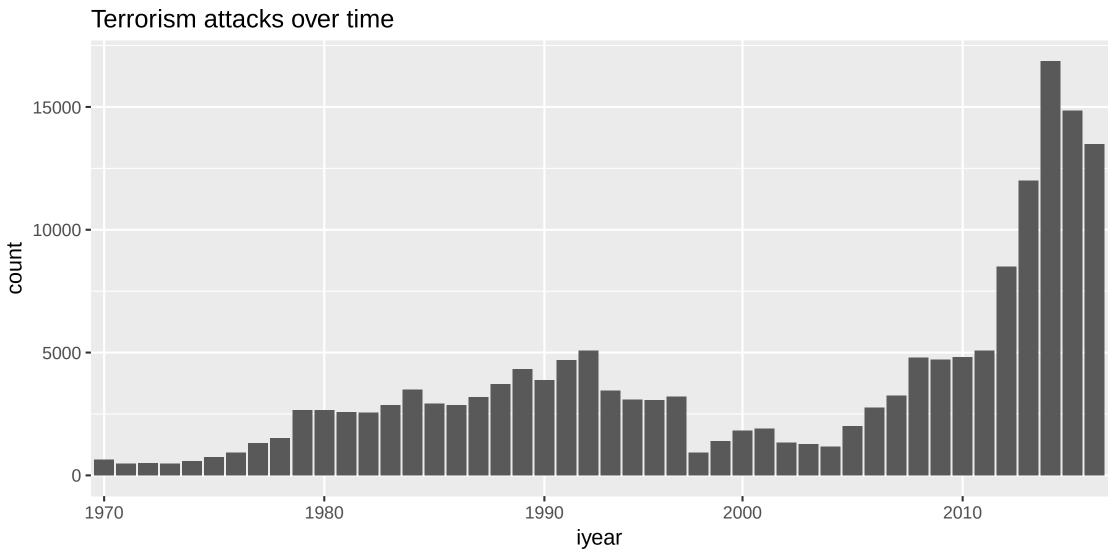


Figure 6: Plot of Terrorist Attacks Over Time.

```
################



## number of terrorist attacks over time
ggplot(gt, aes(x=iyear)) +
  geom_histogram(stat='count') +
  labs(title='Terrorism attacks over time') +
  scale_x_discrete(breaks=c(1970,1980,1990,2000,2010))
```

```
################
### Figure 7 ###
################

## plot terrorism growth versus population growth
p1 <- ggplot(data=popworld2, aes(x=year)) +
  geom_histogram(aes(col='Attack Count'),bins=45) +
  scale_x_continuous(breaks=seq(1970, 2016, 10))

p1+ geom_line(aes(y=population/500000, col='Population')) +
    scale_y_continuous(sec.axis = sec_axis(~ . * 500000, name = "Population Size"))+
    labs(y = "Attack Count", x = "Year", colour = "Blue") +
    labs(title="Terrorism growth vs Population growth")
```

As evidenced by Figure 6, there is an increase in terrorist attacks as the year increases. However, there is also an increase in the general world population as we know from our merge with the world population dataset. One can see from viewing Figure 7, that there is still a rapid increase in terrorist attacks even accounting for world population size aka there is still an increasing amount of terrorist attacks per capita as the years increase which relates to one of our main questions.

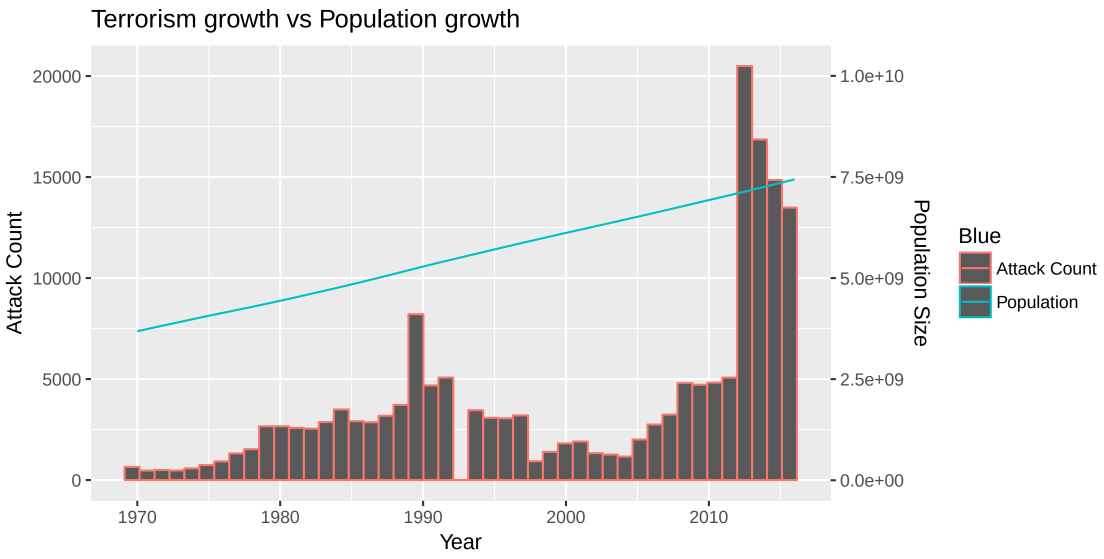


Figure 7: Plot of Terrorism Growth vs. World Population Growth Over Time

```
################
### Figure 8 ###
################

## Worldwide killings grouped by year
datakill <- gt %>% filter(nkill>0)
data_1<-datakill %>% group_by(iyear) %>% summarise(n=sum(nkill))
colnames(data_1)<-c("Year","Killed")
data_1$Year <- as.numeric(data_1$Year)
data_1$Killed <- as.numeric(data_1$Killed)
p2<-ggplot(data_1,aes(x=Year,y=Killed))+
  geom_line()+theme_bw()
p2
```

As evidenced by Figure 8, there is an increasing trend of number of people killed after 2000 and a very high increasing trend after 2010. This relates to one of main questions if there is an increase in terrorist attacks as of recently indicating that there is not just a recency bias! More people are being killed in terrorist attacks as the years go on.

4 Predictive Analytics

4.1 Predicting "ISIL" Terrorist Group

4.1.1 Naive Bayes

Here we lay out our methodology for the Naive Bayes Model.

```
## # A tibble: 170,350 x 135
```

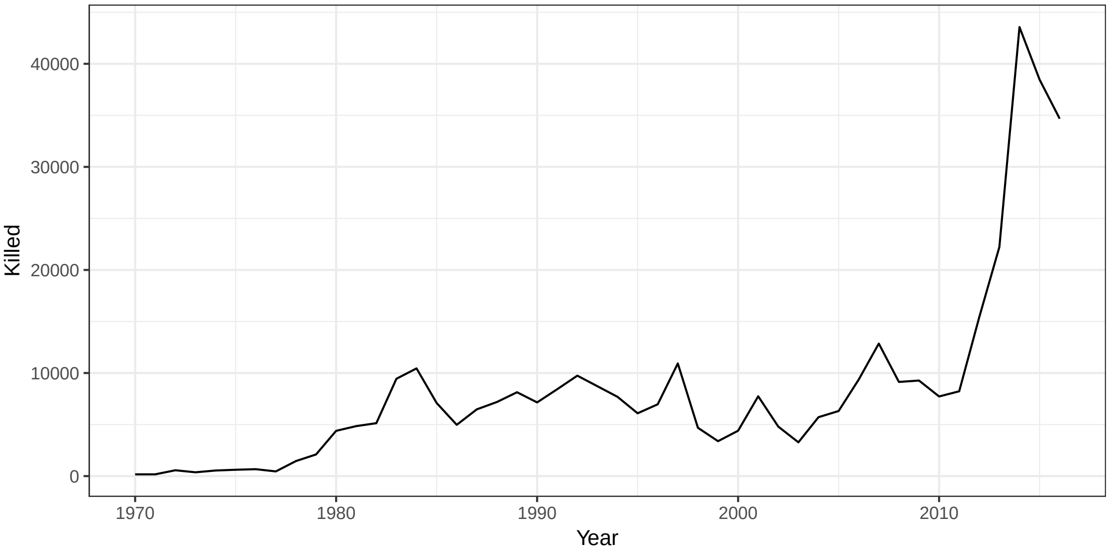


Figure 8: Worldwide Killings Grouped by Year.

```
##          eventid iyear imonth  iday approxdate extended resolution country
##            <dbl> <int>  <int> <int> <fct>         <int> <fct>        <int>
##  1 197000000001  1970      7     2 ""                0 ""              58
##  2 197000000002  1970      0     0 ""                0 ""             130
##  3 197001000001  1970      1     0 ""                0 ""             160
##  4 197001000002  1970      1     0 ""                0 ""              78
##  5 197001000003  1970      1     0 ""                0 ""             101
##  6 197001010002  1970      1     1 ""                0 ""             217
##  7 197001020001  1970      1     2 ""                0 ""             218
##  8 197001020002  1970      1     2 ""                0 ""             217
##  9 197001020003  1970      1     2 ""                0 ""             217
## 10 197001030001  1970      1     3 ""                0 ""             217
## # ... with 170,340 more rows, and 127 more variables: country_txt <fct>,
## #   region <int>, region_txt <fct>, provstate <fct>, city <fct>,
## #   latitude <dbl>, longitude <dbl>, specificity <int>, vicinity <int>,
## #   location <fct>, summary <fct>, crit1 <int>, crit2 <int>, crit3 <int>,
## #   doubtterr <int>, alternative <int>, alternative_txt <fct>,
## #   multiple <int>, success <int>, suicide <int>, attacktype1 <int>,
## #   attacktype1_txt <fct>, attacktype2 <int>, attacktype2_txt <fct>,
## #   attacktype3 <int>, attacktype3_txt <fct>, targtype1 <int>,
## #   targtype1_txt <fct>, targsubtype1 <int>, targsubtype1_txt <fct>,
## #   corp1 <fct>, target1 <fct>, natlty1 <int>, natlty1_txt <fct>,
## #   targtype2 <int>, targtype2_txt <fct>, targsubtype2 <int>,
## #   targsubtype2_txt <fct>, corp2 <fct>, target2 <fct>, natlty2 <int>,
## #   natlty2_txt <fct>, targtype3 <int>, targtype3_txt <fct>,
## #   targsubtype3 <int>, targsubtype3_txt <fct>, corp3 <fct>,
## #   target3 <fct>, natlty3 <int>, natlty3_txt <fct>, gname <fct>,
## #   gsubname <fct>, gname2 <fct>, gsubname2 <fct>, gname3 <fct>,
## #   gsubname3 <fct>, motive <fct>, guncertain1 <int>, guncertain2 <int>,
## #   guncertain3 <int>, individual <int>, nperps <int>, nperpcap <dbl>,
## #   claimed <int>, claimmode <int>, claimmode_txt <fct>, claim2 <int>,
## #   claimmode2 <int>, claimmode2_txt <fct>, claim3 <int>,
## #   claimmode3 <int>, claimmode3_txt <fct>, compclaim <int>,
## #   weaptype1 <int>, weaptype1_txt <fct>, weapsubtype1 <int>,
## #   weapsubtype1_txt <fct>, weaptype2 <int>, weaptype2_txt <fct>,
## #   weapsubtype2 <int>, weapsubtype2_txt <fct>, weaptype3 <int>,
## #   weaptype3_txt <fct>, weapsubtype3 <int>, weapsubtype3_txt <fct>,
## #   weaptype4 <int>, weaptype4_txt <fct>, weapsubtype4 <int>,
## #   weapsubtype4_txt <fct>, weapdetail <fct>, nkill <int>, nkillus <int>,
## #   nkillter <int>, nwound <int>, nwoundus <int>, nwoundte <int>,
## #   property <int>, propextent <int>, propextent_txt <fct>,
## #   propvalue <dbl>, ...
```

```
tally(~Isil, data=train, format="percent")
```

```
## Isil
##         0         1
## 92.501694  7.498306
```

```
## Creating a null model, shows 7.5 % of attacks are known to be carried out by ISIL

form<-as.formula("Isil~iyear+region_txt+attacktype1_txt+targtype1_txt+weaptype1_txt")
## Creating a model using the country, attack type, target type,
##  and weapon type as predictors
```

Our null model shows that 7.5% of terrorist attacks are to be carried out by ISIL, so this is the model to beat. We also write our formula to use for the Naive Bayes modelling. We use ISIL (coded 0 for non-ISIL terrorist group and 1 for ISIL terrorist group) as the dependent variable. The predictors we threw into the model include year the attack happened, region of the world where the attack occurred, the attack type, the target type, and the weapon type.

Here is our Naive Bayes model fitting and our corresponding predictions for the training and testing set.

```
mod_NB_train<-naiveBayes(form, data=train)
## Naive Bayes model for Training Data

pred_NB_train<-predict(mod_NB_train, newdata=train)
## Prediction using the Naive Bayes model for Training Data

mod_NB_test<-naiveBayes(form, data=test)
## Naive Bayes model for Testing Data

pred_NB_test<-predict(mod_NB_test, newdata=test)
## Prediction using the Naive Bayes model for Testing Data
```

```
confusion_train <- table(pred_NB_train, train$success)
confusion_train
```

```
##
```

```
## pred_NB_train     0     1
##             0  6372 37973
##             1   253  1159
```

```
## Confusion Matrix for Training Set

Accuracy_train <-sum(diag(confusion_train))   / nrow(train)
Accuracy_train
```

```
## [1] 0.1645868
```

```
## Our training set appears to have been predicted with about 16.46 percent accuracy



confusion_test <- table(pred_NB_test, test$success)
confusion_test
```

```
##
## pred_NB_test    0    1
##            0 1578 9624
##            1   61  176
```

```
## Confusion Matrix for Test set



Accuracy_test <-sum(diag(confusion_test))   / nrow(test)
Accuracy_test
```

```
## [1] 0.1533351
```

```
## The test set was predicted with about 15.33 percent accuracy
## Overall, the model does not appear to be overfitted, and this is an improvement from the nul
```

Here we are trying to predict whether a terrorist attack was carried out by ISIL, utilizing weapon type, target type, attack type, and country as predictors. To do this effectively, we limit the data to only the years 2013-2016, as these are the years when ISIL became more active in their attacks (as seen in our descriptive analysis seen in Figure 5). Our null model showed that if we assume all attacks are committed by ISIL, we are only right about 7.5 percent of the time. However, we were able to correctly determine if an attack was carried out by ISIL with 17.67 percent accuracy in our training set, and with about 17.21 percent accuracy in our testing set. These accuracies are a decent improvement over the null model, and are close enough that we are not concerned with overfitting. In this example we have laid out the general framework for a Naive Bayes classification so in our future analysis discussing Naive Bayes we will only provide and interpret the output rather than show all of our code.

4.2 Predicting Mass Terrorist Attacks (>3 Deaths)

In this section, we show which predictors could lead to mass terrorist attacks. We want to investigate mass terrorism in our dataset. We define a "mass terrorist attack" as an attack that killed more than three people (3 kills is the 85th percentile of the *nkill* variable in our dataset). We drop any

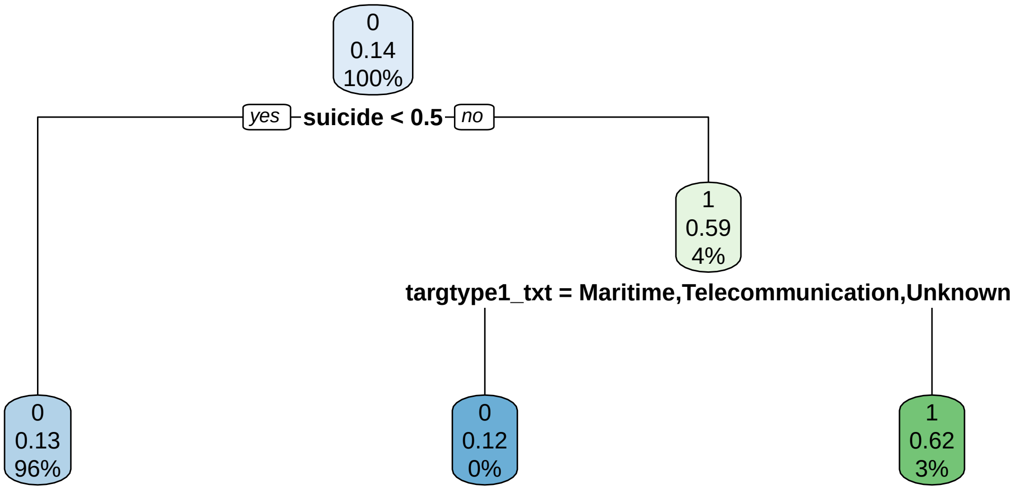


Figure 9: Worldwide Killings Grouped by Year.

missing values in the *nkill* variable.

```
##   0%  25%  50%  75%  85%  90% 100%
##    0    0    0    2    3    5 1500
```

4.2.1 Decision Tree

```
tally(~massTerr, data=train,format="percent") #null model has 85.62% accuracy
```

```
## massTerr
##        0        1
## 85.61937 14.38063
```

```
## Null model


mod_tree <- rpart(massTerr ~ ., data = train, method = "class",control = rpart.control(cp = 0.
#mod_tree
## Builds Decision Tree
```

```
rpart.plot(mod_tree)
```

```
## Plots Decision Tree
```

Our null model shows that 85.62% of terrorist attacks have 3 or fewer kills. We also build our decision tree here. We use massTerr (coded 0 for 3 or fewer kills in the terrorist attack and 1 for greater than 3 kills in the terrorist attack) as the dependent variable. The predictors we threw into the model include the attack type, the year the attack happened, the target type, the weapon type, extended, if there was a suicide attempt, and region of the world where the attack occurred.

Based on the plot of this decision tree in Figure 9, the most important factor used in predicting a "mass terrorist attack" was whether or not it was a suicide attack. If the attack was not a suicide attack (representing about 96% of the data), the model decides that the attack was not a mass terrorist attack. If the attack *was* a suicide (representing about 4% of the data), the model then looks to the "target type" variable to determine whether the attack killed more than three people. If the suicide attack was directed toward some maritime, telecommunication, or some unknown target, the model decides that the attack was not a mass terrorist attacks. If the suicide attack was not directed at one of the aforementioned targets, the model decides that the attack was a mass terrorist attack.

```
train.dt <- train %>%
  mutate(massTerr_dtree   = predict(mod_tree,type="class",data=train))
confusion.mass.terr <- tally(massTerr_dtree   ~ massTerr,data=train.dt,format="count")
confusion.mass.terr
```

```
##                 massTerr
## massTerr_dtree      0      1
##              0 108416  15824
##              1   1634   2660
```

```
## Confusion Matrix Training



dtAcc.mt <- sum(diag(confusion.mass.terr))  / nrow(train)
dtAcc.mt
```

```
## [1] 0.864176
```

```
## Accuracy for Training Data


test.dt <- test %>%
  mutate(massTerr_dtree   = predict(mod_tree, newdata=test, type="class"))
confusion.mass.terr_test <- tally(massTerr_dtree   ~ massTerr, data = test.dt, format = "count"
confusion.mass.terr_test
```

```
##                 massTerr
## massTerr_dtree     0     1
##              0 27132  3924
##              1   416   662
```

```
## Confusion Matrix Testing


dtAcc.mt_test <- sum(diag(confusion.mass.terr_test))  / nrow(test)
dtAcc.mt_test
```

```
## [1] 0.8649406
```

```
## Accuracy for Testing Data
```

```
DT.mt <- c(dtAcc.mt,dtAcc.mt_test)
dtAcc.mt ## Accuracy of Training
```

```
## [1] 0.864176
```

```
dtAcc.mt_test ## Accuracy of Testing
```

```
## [1] 0.8649406
```

Our null model shows that 85.62% of terrorist attacks have 3 or fewer kills. However, we were able to correctly determine if an attack was a mass terrorist attack with 86.42% accuracy in our training set, and with about 86.49% percent accuracy in our testing set. These accuracies are a slight improvement over the null model, and are close enough that we are not concerned with overfitting. Honestly, the testing set had a higher accuracy than the training, so we may have been able to include even more and still be okay! In this example we have laid out the general framework for a decision tree classification so in our future analysis discussing decision trees we will only provide and interpret the output rather than show all of our code.

4.2.2 Random Forest Model

```
##################### To make Random Forest
mod_forest.mt <- randomForest(as.factor(massTerr) ~ . ,data=train,ntree=200,mtry=3)
mod_forest.mt
```

```
##
## Call:
##  randomForest(formula = as.factor(massTerr) ~ ., data = train,      ntree = 200, mtry = 3)
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 12.66%
## Confusion matrix:
##        0    1 class.error
## 0 106345 3705  0.03366652
## 1  12570 5914  0.68004761
```

```
## Modeling Training data







forestAcc.mt <- sum(diag(mod_forest.mt$confusion)) / nrow(train)
forestAcc.mt ## 0.8733331
```

```
## [1] 0.8733798
## Accuracy of Training


imp.mt <- importance(mod_forest.mt)    %>%
     as.data.frame()   %>%
     rownames_to_column()    %>%
     arrange(desc(MeanDecreaseGini))

imp.mt
```

```
##           rowname MeanDecreaseGini
## 1           iyear        2116.9001
## 2   targtype1_txt        2085.1791
## 3         suicide        1959.7063
## 4      region_txt        1743.4835
## 5 attacktype1_txt        1589.6736
## 6   weaptype1_txt         705.4621
## 7        extended         161.7085
```

```
## Importance Scores



mod_forest.mt_test <- randomForest(as.factor(massTerr) ~ ., data = test, ntree = 200, mtry = 3
mod_forest.mt_test
```

```
##
## Call:
##  randomForest(formula = as.factor(massTerr) ~ ., data = test,      ntree = 200, mtry = 3)
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 12.89%
## Confusion matrix:
##       0    1 class.error
## 0 26547 1001  0.03633658
## 1  3140 1446  0.68469254
```

```
## Modeling Test data


forestAcc.mt_test <- sum(diag(mod_forest.mt_test$confusion)) / nrow(test)
forestAcc.mt_test ## 0.8716002
```

```
## [1] 0.8711334
```

```
## Accuracy of Testing

Forest.mt <- c(forestAcc.mt,forestAcc.mt_test)
```

Table 1: Mass Terrorism Model Accuracy Comparison

Data	Null	Decision.Tree	Random.Forest
Train	0.8561937	0.8641760	0.8733798
Test	0.8572851	0.8649406	0.8711334

```
## the training accuracy is 0.8733331 0.8716002
```

Based on the random forest model the most three important predictors are which year, target type and whether is a suicide attack or not.
The training accuracy is 0.873 and the testing accuracy is 0.872. So we can find that this two number is close which means our model is not overfitting.

4.2.3 Model Selection

Based on Table 1, we can conclude that the Random Forest model is the best predictor of mass terrorism attacks. This said, it is important to note that neither the Decision Tree nor the Random Forest model improved on the null by more than 2% in accuracy. It is possible that the accuracies of these models could be improved through more intensive model specifications, but optimizing the models was not the goal of this investigation. In this example we have laid out the general framework for a random forest classification so in our future analysis discussing random forest we will only provide and interpret the output rather than show all of our code.

4.3 Predicting "Successful" Terrorist Attacks

In the dataset, success is a logical variable describing the "success" of an attack. "Success" is defined differently for different types of attacks (i.e. the success of a kidnapping is defined differently than the success of a bombing). Our goal is to build a model that can predict the success of an attack given information provided in the data. We will approach this question using four models:

- Naive Bayes
- Decision Tree
- Random Forest
- Logistic Regression

Since only about 10.25% of the incidents in the training data were not successful, the accuracy of the null model is already 89.75% just by assuming all attacks are successful.

```
#null model
tally(~success, data = train, format = "percent")
```

```
## success
##        0        1
## 10.24875 89.75125
```

4.3.1 Naive Bayes

```
## [1] "form2"
```

```
## as.factor(success) ~ iyear + region_txt + attacktype1_txt + targtype1_txt +
##     weaptype1_txt
```

```
## [1] "confusionNB"
```

```
##
## pred_NB2      0      1
##        0   2395   1356
##        1  11572 120957
```

```
## [1] "NBAcc"
```

```
## [1] 0.9051365
```

```
## [1] "confusionNB_test"
```

```
##
## pred_NB2_test     0     1
##             0   604   336
##             1  3078 30052
```

```
## [1] "NBAcc_test"
```

```
## [1] 0.8997945
```

Here, we try to predict the success of a terrorist attack, utilizing year, region, weapon type, target type, and attack type as predictors. Our null model showed that if we assume all attacks are successful, we are only right 89.75% of the time. However, we were able to correctly determine successful attack with 90.51% accuracy in our training set, and with about 89.97% accuracy in our testing set. These accuracies are an improvement over the null model, and they are close enough that we are not concerned with overfitting. However, these are still very inaccurate. Thus, while this is an improvement from the null model, we cannot recommend utilizing this model to determine if an attack was successful.

4.3.2 Decision Tree

```
## [1] "form"
```

```
## as.factor(success) ~ iyear + attacktype1_txt + targtype1_txt +
##     weaptype1_txt
```

Figure 10 is a decision tree that identifies target type as the most important predictor in the model. If the target type is unknown for a particular attack (representing 3% of the data), the model uses the year of the attack to further classify the attack as either a success or a failure. If the target type is known (representing 97% of the data), the model then uses attacktype to further classify the observation. If the target type is known and the attack type was not an assassination, the model classifies the attack as a success. If the target type is known and the attack type was and assassination, the model then looks to the weapon type to determine whether the attack was successful.

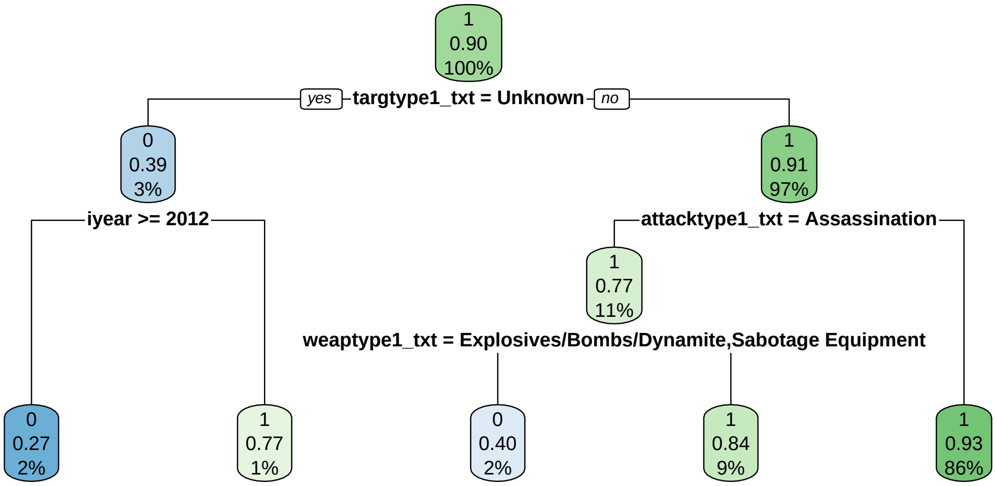


Figure 10: Worldwide Killings Grouped by Year.

```
## [1] "Confusion matrix training"
```

```
##               success
## success_dtree      0      1
##             0   3528   1716
##             1  10439 120597
```

```
## [1] "Confusion matrix testing"
```

```
##               success
## success_dtree     0     1
##             0   936   412
##             1  2746 29976
```

```
## [1] "Accuracy Training"
```

```
## [1] 0.9108086
```

```
## [1] "Accuracy Testing"
```

```
## [1] 0.9073085
```

Despite this attempt to intuitively choose predictors for a decision tree, its rules did not sufficiently improve on the accuracy of the null model. Choosing predictors based on intuition can lead to dead ends, and perhaps the importance of predictors should be left to a random forest model.

4.3.3 Random Forest Model

```
#Call:
# randomForest(formula = form, data = train, ntree = 200, mtry = 3)
#                Type of random forest: classification
#                      Number of trees: 200
```

Table 2: Attack Success: Random Forest Predictor Importance

rowname	MeanDecreaseGini
targtype1_txt	2895.576
iyear	1791.396
weaptype1_txt	1128.004
attacktype1_txt	1018.026

```
#No. of variables tried at each split: 3
#
#        OOB estimate of  error rate: 8.93
#Confusion matrix:
#     0      1 class.error
#0 3513  10454  0.74847856
#1 1720 120593  0.01406228
```

```
## [1] "Accuracy of Training"
```

```
## [1] 0.9106692
```

```
## [1] "Accuracy of Testing"
```

```
## [1] 0.9039331
```

Table 2 according to the Random Forest model identifies “target type,” “country,” and “attack type” as the most important variables for predicting the success of a terrorist attack. The Random Forest model was able to improve on the accuracy of the decision tree model, and it is safe to assume that the model was not overfitted to the train data based on the similarity in accuracy between the train and test data. This said, neither of the tree models were really able to noticeably (>2% better) improve on the 89.75% null model accuracy.

4.3.4 Logistic Regression

```
logit1 <- glm(success ~ iyear + extended + suicide + targtype1_txt, family = binomial(logit),
summary(logit1)
```

```
##
## Call:
## glm(formula = success ~ iyear + extended + suicide + targtype1_txt,
##     family = binomial(logit), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4037   0.3454   0.3984   0.4589   1.4561
##
## Coefficients:
##                                                Estimate Std. Error z value
## (Intercept)                                 32.7323503 1.5837437  20.668
## iyear                                       -0.0155395 0.0007883 -19.712
## extended                                     3.0936172 0.1805461  17.135
## suicide                                     -0.0994006 0.0469862  -2.116
## targtype1_txtAirports & Aircraft            -0.0440610 0.2166087  -0.203
## targtype1_txtBusiness                        0.8635537 0.2003353   4.311
## targtype1_txtEducational Institution         0.6668754 0.2072492   3.218
## targtype1_txtFood or Water Supply            0.4098680 0.2903462   1.412
## targtype1_txtGovernment (Diplomatic)        -0.1587797 0.2047364  -0.776
## targtype1_txtGovernment (General)            0.0669939 0.1991731   0.336
## targtype1_txtJournalists & Media             0.3270071 0.2100814   1.557
## targtype1_txtMaritime                        0.2353235 0.2802643   0.840
## targtype1_txtMilitary                        0.7463153 0.1996099   3.739
## targtype1_txtNGO                             0.9595879 0.2613759   3.671
## targtype1_txtOther                           1.0894923 0.2701364   4.033
## targtype1_txtPolice                          0.7691343 0.1998145   3.849
## targtype1_txtPrivate Citizens & Property     1.0579186 0.1993995   5.306
## targtype1_txtReligious Figures/Institutions  0.8515694 0.2081937   4.090
## targtype1_txtTelecommunication               1.1743688 0.2516574   4.667
## targtype1_txtTerrorists/Non-State Militia    0.8113485 0.2122893   3.822
## targtype1_txtTourists                        0.3271829 0.2729211   1.199
## targtype1_txtTransportation                  0.5458982 0.2032644   2.686
## targtype1_txtUnknown                        -1.9402307 0.2009590  -9.655
## targtype1_txtUtilities                       1.1704067 0.2081676   5.622
## targtype1_txtViolent Political Party         0.2252857 0.2139045   1.053
##                                             Pr(>|z|)    
## (Intercept)                                  < 2e-16 ***
## iyear                                        < 2e-16 ***
## extended                                     < 2e-16 ***
## suicide                                     0.034385 *  
## targtype1_txtAirports & Aircraft            0.838812    
## targtype1_txtBusiness                       1.63e-05 ***
## targtype1_txtEducational Institution        0.001292 ** 
## targtype1_txtFood or Water Supply           0.158052    
## targtype1_txtGovernment (Diplomatic)        0.438025    
## targtype1_txtGovernment (General)           0.736599    
## targtype1_txtJournalists & Media            0.119572    
## targtype1_txtMaritime                       0.401105    
## targtype1_txtMilitary                       0.000185 ***
## targtype1_txtNGO                            0.000241 ***
## targtype1_txtOther                          5.50e-05 ***
## targtype1_txtPolice                         0.000118 ***
## targtype1_txtPrivate Citizens & Property    1.12e-07 ***
## targtype1_txtReligious Figures/Institutions 4.31e-05 ***
## targtype1_txtTelecommunication              3.06e-06 ***
## targtype1_txtTerrorists/Non-State Militia   0.000132 ***
## targtype1_txtTourists                       0.230599    
## targtype1_txtTransportation                 0.007239 ** 
## targtype1_txtUnknown                             < 2e-16 ***
## targtype1_txtUtilities                          1.88e-08 ***
## targtype1_txtViolent Political Party            0.292246
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 90085  on 136279  degrees of freedom
## Residual deviance: 81202  on 136255  degrees of freedom
## AIC: 81252
##
## Number of Fisher Scoring iterations: 7
```

```r
predLogit <- predict(logit1, newdata=train, type=c("response"))

#confusion matrix
confusion.Logit <- table(predLogit > 0.5,train$success)
confusion.Logit
```

```
##
##                0      1
##   FALSE   2360   1406
##   TRUE   11607 120907
```

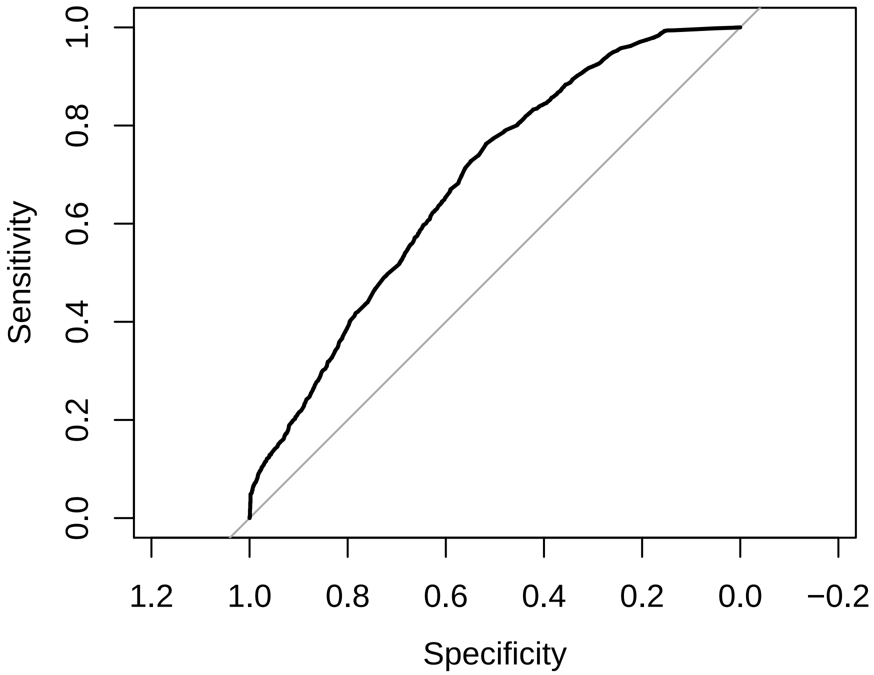
```r
#accuracy
logitAcc <- sum(diag(confusion.Logit)) / nrow(train)

#overfitting?
predLogit_test <- predict(logit1, newdata=test)
confusion.Logit_test <- table(predLogit_test > 0.5,test$success)
logitAcc_test <- sum(diag(confusion.Logit_test)) / nrow(test)

Logit <- c(logitAcc,logitAcc_test)
Logit
```

```
## [1] 0.9045128 0.8983857
```

```r
#ROC Curve
roccurve <- roc(test$success,predLogit_test)
plot(roccurve)
```

Table 3: Attack Success: Model Accuracy Comparison Table

Data	Null	Naive.Bayes	Decision.Tree	Random.Forest	Logit
Train	0.8975125	0.9051365	0.9108086	0.9106692	0.9045128
Test	0.8919284	0.8997945	0.9073085	0.9039331	0.8983857



The model does not seem to be overfit, seeing that the accuracy of the model on both the train and test data are very similar. This model yielded an accuracy of 90.45% for the training data set and an accuracy of 89.84% for the test data set. The accuracy of this model is still unable to predict attack success effectively better than just assuming all attacks are successful. The ROC curve shown is an indication that the model is predicting success better than random chance (since it lies well above the 45 degree line). While this is not of much use for a single model other than to show its general suitability, it is helpful when comparing logistic regression models with different predictors.

4.3.5 Model Selection

```
Label <- c("Train","Test")
nullAcc <- table(train$success)[[2]]/nrow(train)
nullAcc_test <- table(test$success)[[2]]/nrow(test)
Null <- c(nullAcc,nullAcc_test)

modelCompare <- data.frame("Data"=Label,Null,"Naive Bayes"=NB,"Decision Tree"=DT,"Random Fores
knitr::kable(modelCompare,caption = 'Attack Success: Model Accuracy Comparison Table', booktab
```

Table 3 shows the accuracies of these 4 different modeling techniques on the predicting of the success rate. So according to accuracy, the random forest, although computationally expensive appeared to perform the best. The null model appeared to perform the worst (which means that our models were at least somewhat helpful!).

5 Conclusion

To recap, some of our exploratory questions we set out to accomplish in this project are:

- What are the factors (possibly weapon type, attack type, HDI, etc.) that predict the terrorist group that perpetrated an attack?
- Investigate the trends in mass terrorism (terrorist attacks with many people killed) in the dataset. Identify any possible unique characteristics of mass terrorism.
- What factors may predict the success rate of attacks? What are the significant drivers for a terrorist attack to be "successful"? Is there a bias pertaining to the recency of an attack? [Some possible solutions: logit/probit prediction, random forest model]

We know from our descriptive analysis, that terrorist attacks, as well as number of people killed in terrorist attacks, are increasing worldwide. Even accounting for the world's growing population, the number of attacks are still rising per capita at an alarming rate. So thus, there is not just a recency bias as some may have speculated. We will seek to model and interpret why this could be later in our project. There have also been a lot of terrorist attacks in the Middle East and South Asia. This fact is highly known in current events as there is a lot of political turmoil in many of those nations. We have also seen that there has been a large increase in bombings in recent years, as well as a mild spike in armed assault both of which also have the highest gross total of attacks since 1970. Since the late 1990's/early 2000's, this increase has been seen. Why is this so? This is a question we wish to further explore in our predictive analysis in the future; is this increase because terrorist attacks of this type are more successful? We also observed that terrorist groups like Islamic State of Iraq and the Levant and Al-Shabaab are on the rise recently, while the Taliban has the most successful attacks since 1970.

For predicting if the terrorist group was "ISIL" or not using a Naive Bayes model, our null model showed that if we assume all attacks are committed by ISIL, we are only right about 7.5 percent of the time. However, we were able to correctly determine if an attack was carried out by ISIL with 17.67 percent accuracy in our training set, and with about 17.21 percent accuracy in our testing set. So we are decently better at predicting by accuracy than the null model.

For predicting if the terrorist attack was a "mass attack" or not using a Decision tree and Random Forest, we saw in Table 1, we could conclude that the Random Forest model was the best predictor of mass terrorism attacks. This said, it is important to note that neither the Decision Tree nor the Random Forest model improved on the null by more than 2% in accuracy.

For predicting if a terrorist attack was a success, we produced 4 different prediction models, including Naive Bayes, Decision Tree, Random Forest, and even a Logistic Regression, with the Random Forest performing the best and all of our models beating the null. Table 3 can show more detail. We feel we have accomplished the goals and exploratory questions we originally set out to accomplish in this analysis.