Global Terrorism Aanalysis

Sanjeeve Raveenthiran M

rm.sanjeeve@gmail.com

```
knitr::opts_chunk$set(echo = TRUE)

library(tidyverse)
```

```
## -- Attaching packages ------------------------------ tidyverse 1.2.1 --
```

```
## v ggplot2 3.0.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.6
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## -- Conflicts --------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(readr)
library(ggmap)
library(rworldmap)
```

```
## Loading required package: sp
```

```
## ### Welcome to rworldmap ###
```

```
## For a short introduction type :   vignette('rworldmap')
```

```
library(arules)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
##
## Attaching package: 'arules'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```r
# library(arulesViz)
library(ggpubr)
```

```
## Loading required package: magrittr
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:ggmap':
##
##     inset
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
## The following object is masked from 'package:arules':
##
##     recode
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```r
library(caret)
```

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

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(forecast)
```

```
##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:ggpubr':
##
##     gghistogram
```

```r
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(ggfortify)
```

0.1 Database information

The dataset we are looking at is the Global Terrorism Database (GTD), an open-source database with information on terrorist attacks around the world from 1970 - 2016 with more than 170,000 cases. (https://www.kaggle.com/START-UMD/gtd (https://www.kaggle.com/START-UMD/gtd)) The database is maintained and updated periodically by researchers at the National Consortium for the Study of Terrorism and Responses to Terrorism (START) at the university of Maryland. More information on variable meanings can be found at http://start.umd.edu/gtd/downloads/Codebook.pdf (http://start.umd.edu/gtd/downloads/Codebook.pdf), however if a variable is used and unclear what the meaning is this report will provide a quick definition. Besides the terrorism datasets we will also use the world population dataset from the United Nations population forecasts (https://esa.un.org/unpd/wpp/Download/Standard/Population/ (https://esa.un.org/unpd/wpp/Download/Standard/Population/)) to look at terrorism trends over time compared to population growth, density, etc. It should be noted that the terrorism data has 1 year missing, we decided to just ignore this as it doesn't impact the overall analysis.

Our analysis of this database will be based on a set of research questions. All data manipulation will be done before answering each research question in the corresponding code chunk.

First some data importing, general data reduction and renaming for clarity.

0.2 Importing dataset

```
fulldf <- read_csv("globalterrorismdb_0617dist.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   eventid = col_double(),
##   iyear = col_integer(),
##   imonth = col_integer(),
##   iday = col_integer(),
##   extended = col_integer(),
##   country = col_integer(),
##   region = col_integer(),
##   latitude = col_double(),
##   longitude = col_double(),
##   specificity = col_integer(),
##   vicinity = col_integer(),
##   crit1 = col_integer(),
##   crit2 = col_integer(),
##   crit3 = col_integer(),
##   doubtterr = col_integer(),
##   alternative = col_integer(),
##   multiple = col_integer(),
##   success = col_integer(),
##   suicide = col_integer(),
##   attacktype1 = col_integer()
##   # ... with 44 more columns
## )
```

```
## See spec(...) for full column specifications.
## Warning in rbind(names(probs), probs_f): number of columns of result is not
## a multiple of vector length (arg 1)
```

```
## Warning: 246 parsing failures.
## row # A tibble: 5 x 5 col     row col       expected             actual file
expected   <int> <chr>     <chr>                <chr>  <chr>                        actual 1
7127 nhours    no trailing charact~ .25    'globalterrorismdb_0617dist~ file 2  9153 nhours
no trailing charact~ .25    'globalterrorismdb_0617dist~ row 3 10182 nhours    no trailing ch
aract~ .25    'globalterrorismdb_0617dist~ col 4 10714 nhours    no trailing charact~ .25
'globalterrorismdb_0617dist~ expected 5 11852 propvalue no trailing charact~ .78    'globalte
rrorismdb_0617dist~
## ... ................. ...
.......................................................................... ........
.......................................................................... ......
.......................................................................... ....
.......................................................................... ...
.......................................................................... ...
.......................................................................... ........
..........................................................................
## See problems(...) for more details.
```

```
glimpse(fulldf)
```

```
## Observations: 170,350
## Variables: 135
## $ eventid            <dbl> 197000000001, 197000000002, 197001000001, 1...
## $ iyear              <int> 1970, 1970, 1970, 1970, 1970, 1970, 1970, 1...
## $ imonth             <int> 7, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ iday               <int> 2, 0, 0, 0, 0, 1, 2, 2, 2, 3, 1, 6, 8, 9, 9...
## $ approxdate         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ extended           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ resolution         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ country            <int> 58, 130, 160, 78, 101, 217, 218, 217, 217, ...
## $ country_txt        <chr> "Dominican Republic", "Mexico", "Philippine...
## $ region             <int> 2, 1, 5, 8, 4, 1, 3, 1, 1, 1, 1, 1, 8, 1, 1...
## $ region_txt         <chr> "Central America & Caribbean", "North Ameri...
## $ provstate          <chr> NA, NA, "Tarlac", "Attica", NA, "Illinois",...
## $ city               <chr> "Santo Domingo", "Mexico city", "Unknown", ...
## $ latitude           <dbl> 18.45679, 19.43261, 15.47860, 37.98377, 33....
## $ longitude          <dbl> -69.95116, -99.13321, 120.59974, 23.72816, ...
## $ specificity        <int> 1, 1, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ vicinity           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ location           <chr> NA, NA, NA, NA, NA, NA, NA, "Edes Substatio...
## $ summary            <chr> NA, NA, NA, NA, NA, "1/1/1970: Unknown Afri...
## $ crit1              <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ crit2              <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ crit3              <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1...
## $ doubtterr          <int> 0, 0, 0, 0, -9, 0, 0, 1, 0, 0, 1, 1, -9, 0,...
## $ alternative        <int> NA, NA, NA, NA, NA, NA, NA, 2, NA, NA, 1, 2...
## $ alternative_txt    <chr> NA, NA, NA, NA, NA, NA, NA, "Other Crime Ty...
## $ multiple           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ success            <int> 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1...
## $ suicide            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ attacktype1        <int> 1, 6, 1, 3, 7, 2, 1, 3, 7, 7, 3, 7, 4, 7, 7...
## $ attacktype1_txt    <chr> "Assassination", "Hostage Taking (Kidnappin...
## $ attacktype2        <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ attacktype2_txt    <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ attacktype3        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ attacktype3_txt    <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ targtype1          <int> 14, 7, 10, 7, 7, 3, 3, 21, 4, 2, 4, 4, 6, 2...
## $ targtype1_txt      <chr> "Private Citizens & Property", "Government ...
## $ targsubtype1       <int> 68, 45, 54, 46, 46, 22, 25, 107, 28, 21, 27...
## $ targsubtype1_txt   <chr> "Named Civilian", "Diplomatic Personnel (ou...
## $ corp1              <chr> NA, "Belgian Ambassador Daughter", "Voice o...
## $ target1            <chr> "Julio Guzman", "Nadine Chaval, daughter", ...
## $ natlty1            <int> 58, 21, 217, 217, 217, 217, 218, 217, 217, ...
## $ natlty1_txt        <chr> "Dominican Republic", "Belgium", "United St...
## $ targtype2          <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ targtype2_txt      <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ targsubtype2       <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ targsubtype2_txt   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ corp2              <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ target2            <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ natlty2            <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ natlty2_txt        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ targtype3          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ targtype3_txt      <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ targsubtype3       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ targsubtype3_txt   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ corp3              <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ target3            <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ natlty3            <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ natlty3_txt        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ gname              <chr> "MANO-D", "23rd of September Communist Leag...
## $ gsubname           <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ gname2             <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ gsubname2          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ gname3             <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ gsubname3          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ motive             <chr> NA, NA, NA, NA, NA, "To protest the Cairo I...
## $ guncertain1        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1...
## $ guncertain2        <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ guncertain3        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ individual         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ nperps             <int> NA, 7, NA, NA, NA, -99, 3, -99, 1, 1, NA, -...
## $ nperpcap           <int> NA, NA, NA, NA, NA, -99, NA, -99, 1, 1, NA,...
## $ claimed            <int> NA, NA, NA, NA, NA, 0, NA, 0, 1, 0, NA, 0, ...
## $ claimmode          <int> NA, NA, NA, NA, NA, NA, NA, NA, 1, NA, NA, ...
## $ claimmode_txt      <chr> NA, NA, NA, NA, NA, NA, NA, NA, "Letter", N...
## $ claim2             <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ claimmode2         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ claimmode2_txt     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ claim3             <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ claimmode3         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ claimmode3_txt     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ compclaim          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ weaptype1          <int> 13, 13, 13, 6, 8, 5, 5, 6, 8, 8, 6, 8, 5, 8...
## $ weaptype1_txt      <chr> "Unknown", "Unknown", "Unknown", "Explosive...
## $ weapsubtype1       <int> NA, NA, NA, 16, NA, 5, 2, 16, 19, 20, 16, 1...
## $ weapsubtype1_txt   <chr> NA, NA, NA, "Unknown Explosive Type", NA, "...
## $ weaptype2          <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ weaptype2_txt      <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ weapsubtype2       <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ weapsubtype2_txt   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ weaptype3          <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ weaptype3_txt      <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ weapsubtype3       <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ weapsubtype3_txt   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ weaptype4          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ weaptype4_txt      <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ weapsubtype4       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ weapsubtype4_txt   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ weapdetail         <chr> NA, NA, NA, "Explosive", "Incendiary", "Sev...
## $ nkill              <int> 1, 0, 1, NA, NA, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ nkillus            <int> NA, NA, NA, NA, NA, 0, NA, 0, 0, 0, NA, 0, ...
## $ nkillter           <int> NA, NA, NA, NA, NA, 0, NA, 0, 0, 0, NA, 0, ...
## $ nwound             <int> 0, 0, 0, NA, NA, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ nwoundus           <int> NA, NA, NA, NA, NA, 0, NA, 0, 0, 0, NA, 0, ...
## $ nwoundte           <int> NA, NA, NA, NA, NA, 0, NA, 0, 0, 0, NA, 0, ...
## $ property           <int> 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1...
## $ propextent         <int> NA, NA, NA, NA, NA, 3, NA, 3, 3, 3, 3, 3, N...
## $ propextent_txt     <chr> NA, NA, NA, NA, NA, "Minor (likely < $1 mil...
## $ propvalue          <int> NA, NA, NA, NA, NA, NA, NA, 22500, 60000, N...
## $ propcomment        <chr> NA, NA, NA, NA, NA, NA, NA, "Three transfor...
## $ ishostkid          <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ nhostkid           <int> NA, 1, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ nhostkidus         <int> NA, 0, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ nhours             <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ ndays              <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ divert             <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ kidhijcountry      <chr> NA, "Mexico", NA, NA, NA, NA, NA, NA, NA, N...
## $ ransom             <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ransomamt          <int> NA, 800000, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ ransomamtus        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ ransompaid         <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ ransompaidus       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ ransomnote         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ hostkidoutcome     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ hostkidoutcome_txt <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ nreleased          <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ addnotes           <chr> NA, NA, NA, NA, NA, "The Cairo Chief of Pol...
## $ scite1             <chr> NA, NA, NA, NA, NA, "\"Police Chief Quits,\...
## $ scite2             <chr> NA, NA, NA, NA, NA, "\"Cairo Police Chief Q...
## $ scite3             <chr> NA, NA, NA, NA, NA, "Christopher Hewitt, \"...
## $ dbsource           <chr> "PGIS", "PGIS", "PGIS", "PGIS", "PGIS", "He...
## $ INT_LOG            <int> 0, 0, -9, -9, -9, -9, 0, -9, 0, 0, 0, -9, -...
## $ INT_IDEO           <int> 0, 1, -9, -9, -9, -9, 0, -9, 0, 0, 0, -9, -...
## $ INT_MISC           <int> 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0...
## $ INT_ANY            <int> 0, 1, 1, 1, 1, -9, 0, -9, 0, 0, 0, -9, 1, -...
## $ related            <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
```

```
#there are 170,350 cases and 135 variables.

#get the world population by country
fullpop <- read_csv("UNpopfile.csv")
```

```
## Parsed with column specification:
## cols(
##   LocID = col_integer(),
##   Location = col_character(),
##   VarID = col_integer(),
##   Variant = col_character(),
##   Time = col_integer(),
##   MidPeriod = col_double(),
##   PopMale = col_double(),
##   PopFemale = col_double(),
##   PopTotal = col_double()
## )
```

```
pop <- fullpop %>%
  select(-MidPeriod, -PopMale, -PopFemale, -VarID)

pop <- pop %>%
  filter(Time > 1969 & Variant == 'Medium' & Time < 2017) %>%
  select(-Variant, -LocID)

futurepop <- fullpop %>%
  filter(Time >2016 & Variant == 'Medium') %>%
  select(-Variant, -LocID, -MidPeriod, -PopMale, -PopFemale, -VarID)
```

0.3 Data Cleaning

Removing unnecessary variables & renaming some variables. We are going to focus for the sake of keeping our analysis clear on a subset of variables.

0.3.1 Variables of importance

1. iyear
2. imonth
3. iday
4. country_txt
5. region_txt
6. city
7. latitude
8. longitude
9. summary - event summary, what happened? when etc.
10. multiple - was the attack part of a multiple attack event?
11. attacktype1_txt
12. targtype1_txt
13. targsubtype1_txt
14. gname - perpetrator group name
15. weaptype1_txt
16. nkill - confirmed fatalities of event
17. nwound - number of non-fatal wounded of event
18. nkillter - fatalities of perpetrator(s)

0.3.2 selecting vars of importance and renaming

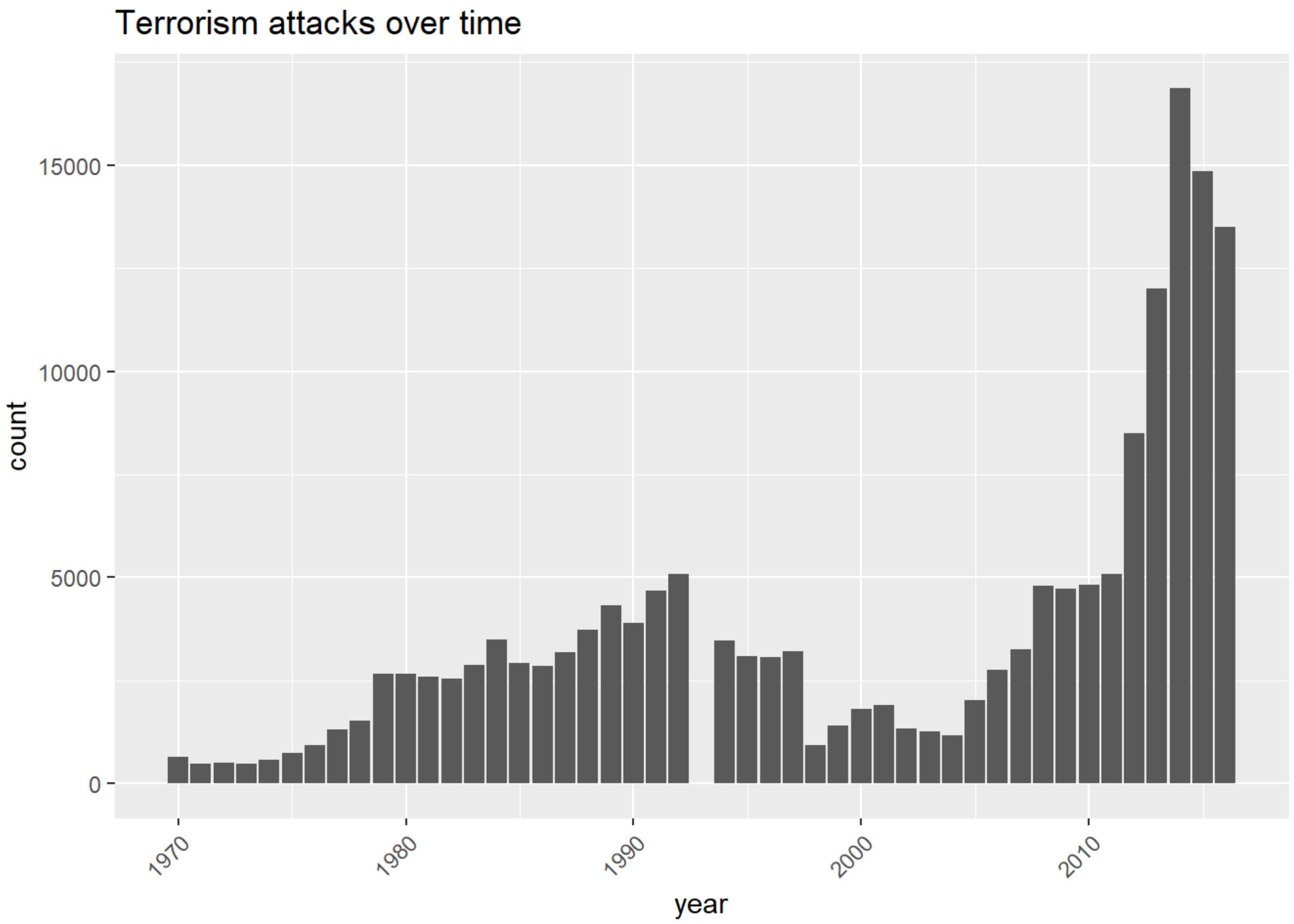
```
df <- fulldf %>%
  select(iyear, imonth, iday, country_txt, region_txt, city, latitude, longitude, summary, multiple, attacktype1_txt, targtype1_txt, targsubtype1_txt, gname, weaptype1_txt, nkill, nwound, nkillter)

df <- df %>%
  rename(year = iyear, month = imonth, day = iday, country = country_txt, region = region_txt, multiple_attack = multiple, attacktype = attacktype1_txt, target_type = targtype1_txt, target_sub_type = targsubtype1_txt, group_name = gname, weapon_type = weaptype1_txt)

df <- df %>%
  mutate(decade =
           ifelse(year<1980, '70s',
                  ifelse(year < 1990, '80s',
                         ifelse(year < 2000, '90s',
                                ifelse( year < 2010, '2000s', '2010s')))))

df$decade <- factor(df$decade, levels=c("70s", "80s", "90s", "2000s", "2010s"))
```

1. Data Overview

1.1 Number of Terrorist Attacks

```
ggplot(data=df, aes(x=year)) +
  geom_histogram(stat='count') +
  theme(axis.text.x= element_text(angle=45, hjust=1)) +
  labs(title='Terrorism attacks over time')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



```
df %>%
  summarise(nr_of_attacks = n())
```

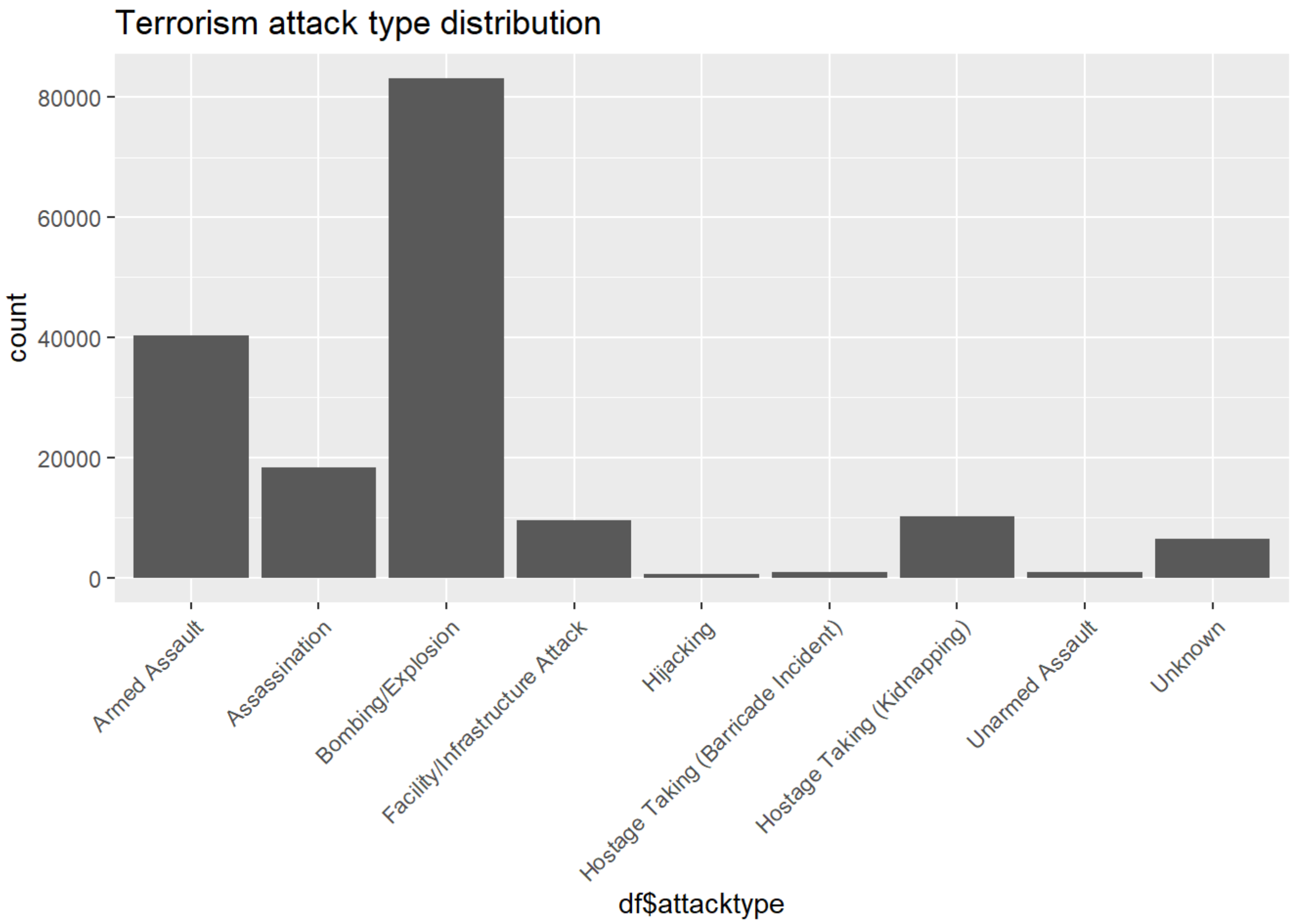
```
## # A tibble: 1 x 1
##   nr_of_attacks
##           <int>
## 1        170350
```

Over 170000 attacks happening, and they seem to have gone up!

1.2 Attack type Distribution

```
ggplot(data = df, aes(x = df$attacktype)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  geom_histogram(stat = "count") +
  labs(title='Terrorism attack type distribution')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



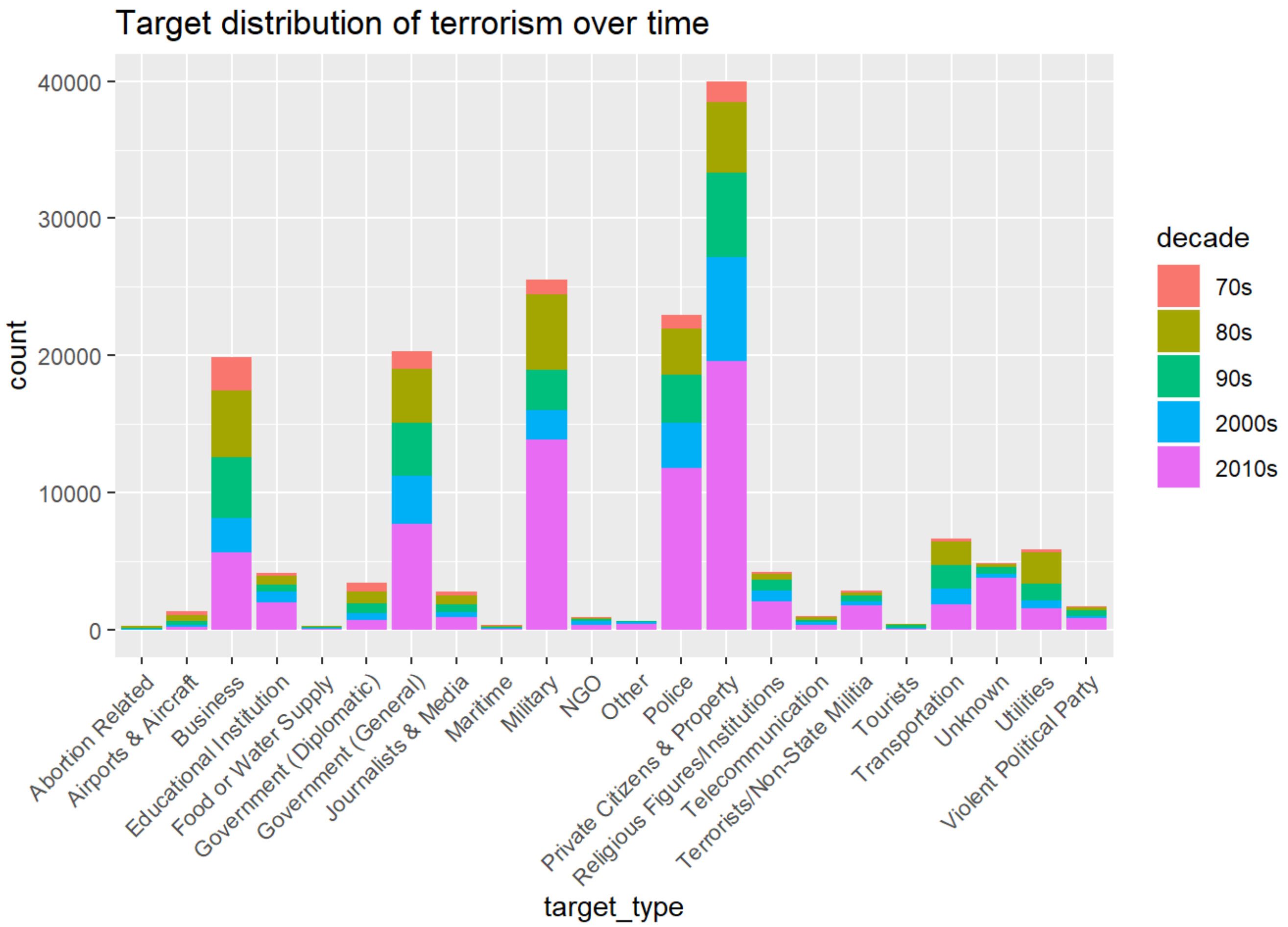
More than 80,000 Bombings, second biggest grouping is Armed assault ~40,000 attacks.

1.3 Target Distribution

Let’s get an idea of what kind of targets terrorists hit.

```
#visual
ggplot(data=df, aes(x=target_type, fill=decade)) +
  geom_histogram(stat='count') +
  theme(axis.text.x= element_text(angle=45, hjust=1)) +
  labs(title='Target distribution of terrorism over time')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



```
#table
df %>%
  group_by(target_type) %>%
  summarise(nr_of_attacks = n()) %>%
  arrange(desc(nr_of_attacks)) %>%
  head(n=10)
```

```
## # A tibble: 10 x 2
##    target_type                     nr_of_attacks
##    <chr>                                   <int>
##  1 Private Citizens & Property             39994
##  2 Military                                25508
##  3 Police                                  22938
##  4 Government (General)                    20314
##  5 Business                                19873
##  6 Transportation                           6657
##  7 Utilities                                5848
##  8 Unknown                                  4873
##  9 Religious Figures/Institutions           4198
## 10 Educational Institution                  4160
```

It seems private citizens have become a bigger target, lets check this out in more depth in question 2.6

1.4 location of terrorism (region/country/city?)

We want to see where the terrorist attacks happen around the world. For this we'll use the ggmap package.

```
#For plotting clarity lets just check out attacks from the last decade and onwards.
df2000 <- df %>%
  filter(year > 2006)

world <- borders("world", colour="gray50", fill="gray50")
```

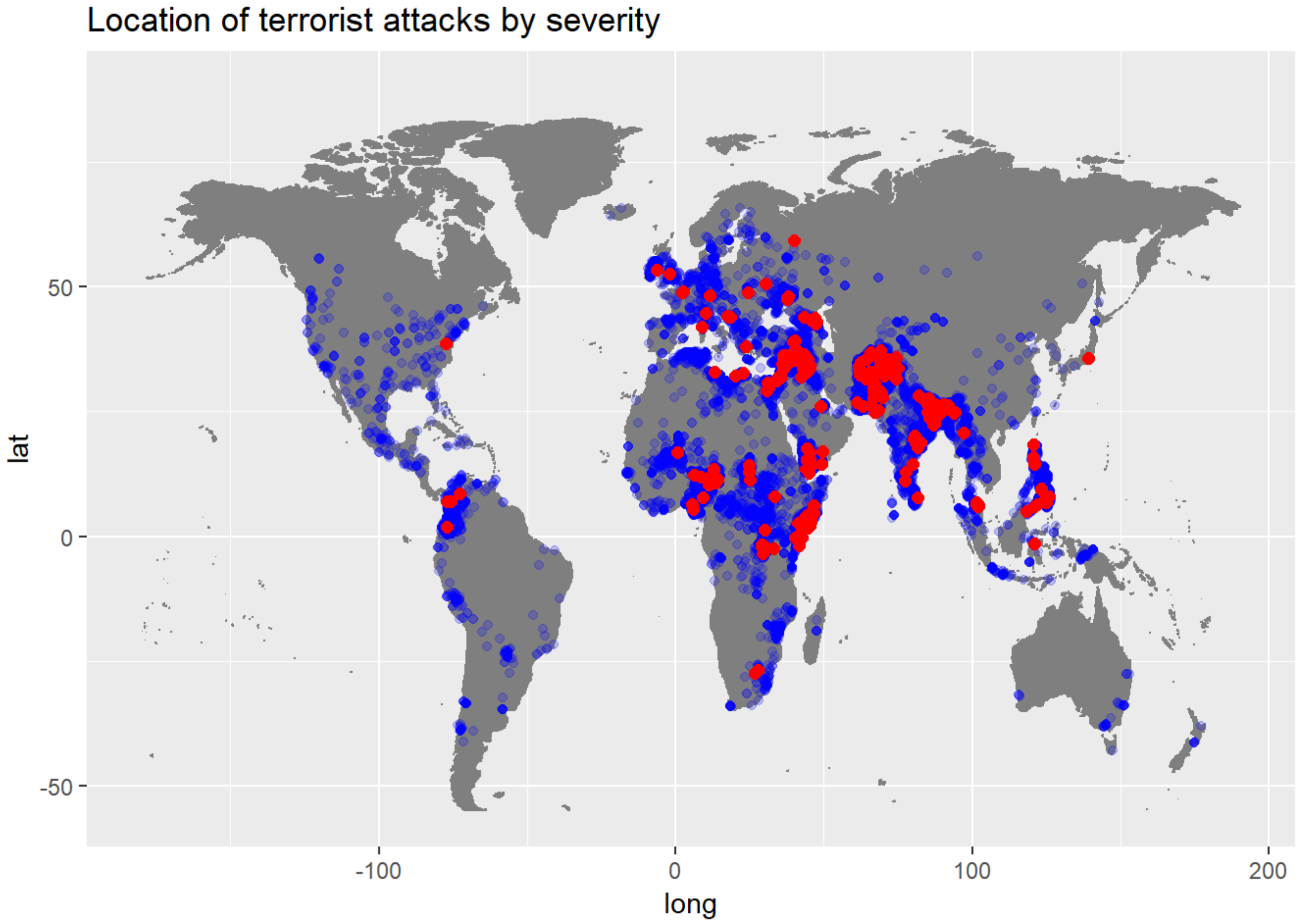
```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
worldmap <- ggplot() + world + scale_y_continuous(limits=c(-55, 90))

worldmap +
  geom_point(aes(x=df2000$longitude[df$nkill<51], y=df2000$latitude[df$nkill<51]), col='blu
e', alpha= 0.2) +
  geom_point(aes(x=df2000$longitude[df$nkill>50], y=df2000$latitude[df$nkill>50]), col='red',
size=2) +
  labs(title='Location of terrorist attacks by severity')
```

```
## Warning: Removed 88735 rows containing missing values (geom_point).
```

```
## Warning: Removed 9962 rows containing missing values (geom_point).
```



Red dots are for more than 50 deaths and blue dots for less than 51.

Let's also view the top 10 locations for terrorist attacks by region, country and city.

```
df %>%
  group_by(region) %>%
  summarise( nr_of_attacks = n()) %>%
  arrange(desc(nr_of_attacks)) %>%
  head(n=10)
```

```
## # A tibble: 10 x 2
##    region                      nr_of_attacks
##    <chr>                               <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
```

```
df %>%
  group_by(country) %>%
  summarise( nr_of_attacks = n()) %>%
  arrange(desc(nr_of_attacks)) %>%
  head(n=10)
```

```
## # A tibble: 10 x 2
##    country        nr_of_attacks
##    <chr>                  <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
```

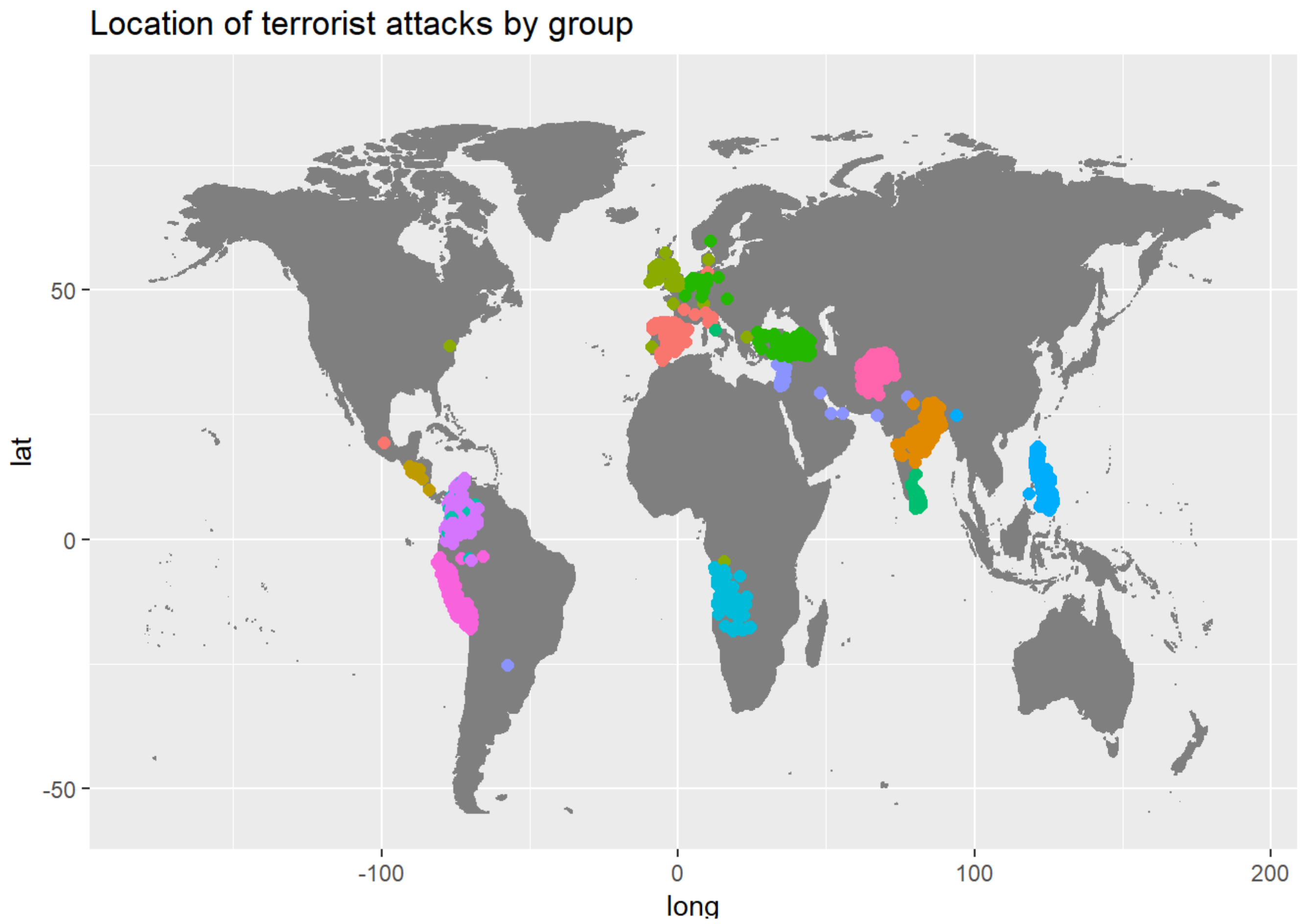
```
df %>%
  filter(city != 'Unknown') %>%
  group_by(city) %>%
  summarise( nr_of_attacks = n()) %>%
  arrange(desc(nr_of_attacks)) %>%
  head(n=10)
```

```
## # A tibble: 10 x 2
##    city         nr_of_attacks
##    <chr>                <int>
##  1 Baghdad               7206
##  2 Karachi               2609
##  3 Lima                  2358
##  4 Belfast               2140
##  5 Mosul                 1775
##  6 Santiago              1618
##  7 San Salvador          1547
##  8 Mogadishu             1351
##  9 Istanbul              1037
## 10 Athens                 987
```

Let’s also check what the distribution of terrorist groups is over the world, first we need to reduce the number of data so lets group by decade and only take the top 10000 points of each decade. And for visibility the set is further filtered based on having more than 300 attacks.

```
df500 <- df %>%
  select(decade, latitude, longitude, group_name) %>%
  group_by(decade) %>%
  slice(1:10000)

df500 <- df500 %>%
  group_by(group_name) %>%
  filter(n() >= 300 & group_name != "Unknown")

worldmap +
  geom_point(aes(x=df500$longitude, y=df500$latitude, col=df500$group_name), size=2, position
= 'jitter') +
  labs(title='Location of terrorist attacks by group') +
  theme(legend.position=c(0.5, -0.5))
```

```
## Warning: Removed 600 rows containing missing values (geom_point).
```



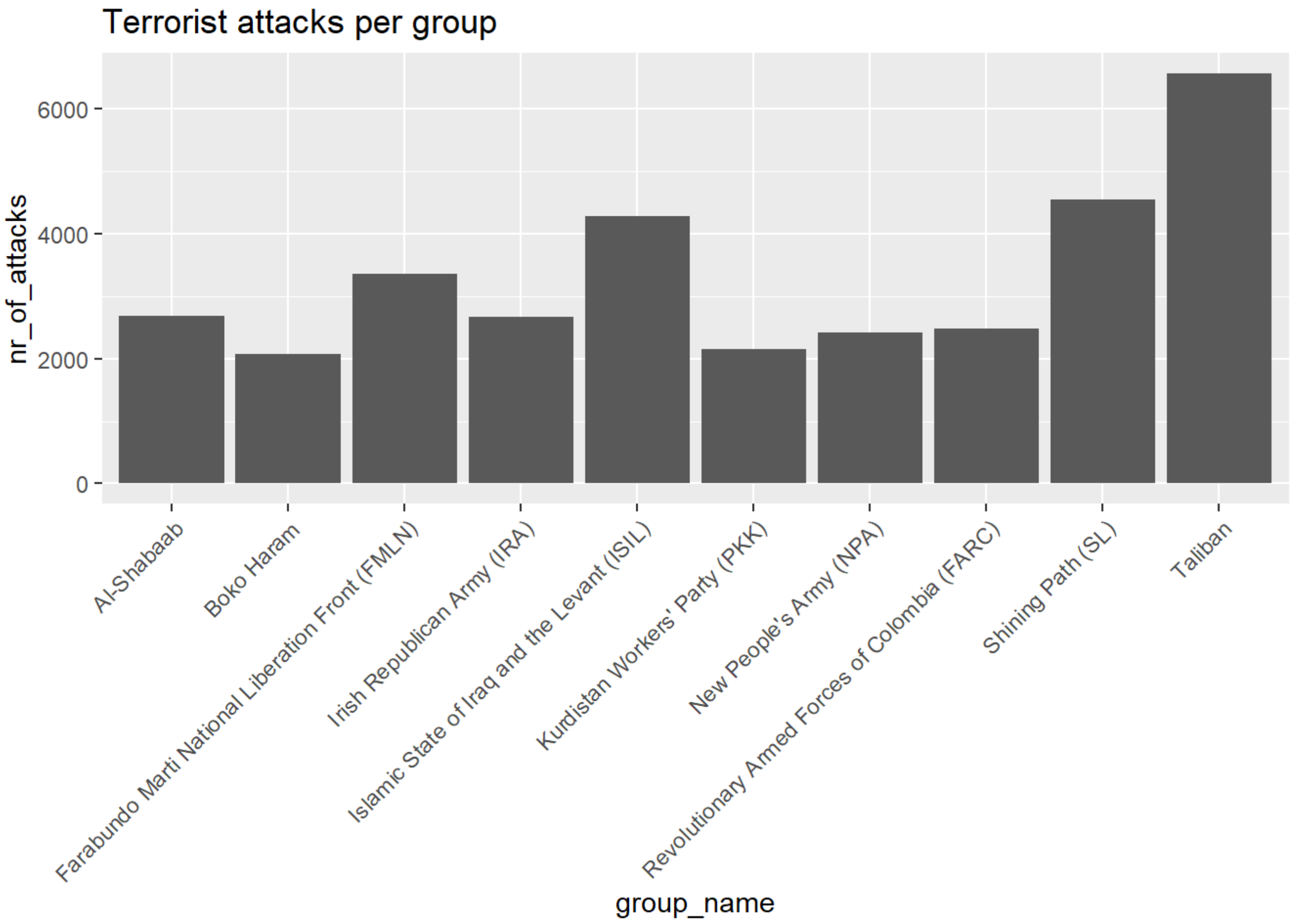
The geographical spread is very obvious with the well known groups.

1.5 distribution of terrorist groups

Who are doing these attacks?

```
#table
top10_groups <- df %>%
  filter(group_name != "Unknown") %>%
  group_by(group_name) %>%
  summarise(nr_of_attacks = n()) %>%
  arrange(desc(nr_of_attacks)) %>%
  head(n=10)

#visual
ggplot(data=top10_groups) +
  stat_summary(aes(x=group_name, y=nr_of_attacks), geom="bar") +
  theme(axis.text.x= element_text(angle=45, hjust=1)) +
  labs(title='Terrorist attacks per group')
```

```
## No summary function supplied, defaulting to `mean_se()
```



2. Trends in terorrism

2.1 Has terrorism gone up?

2.1.1 Terrorism growth

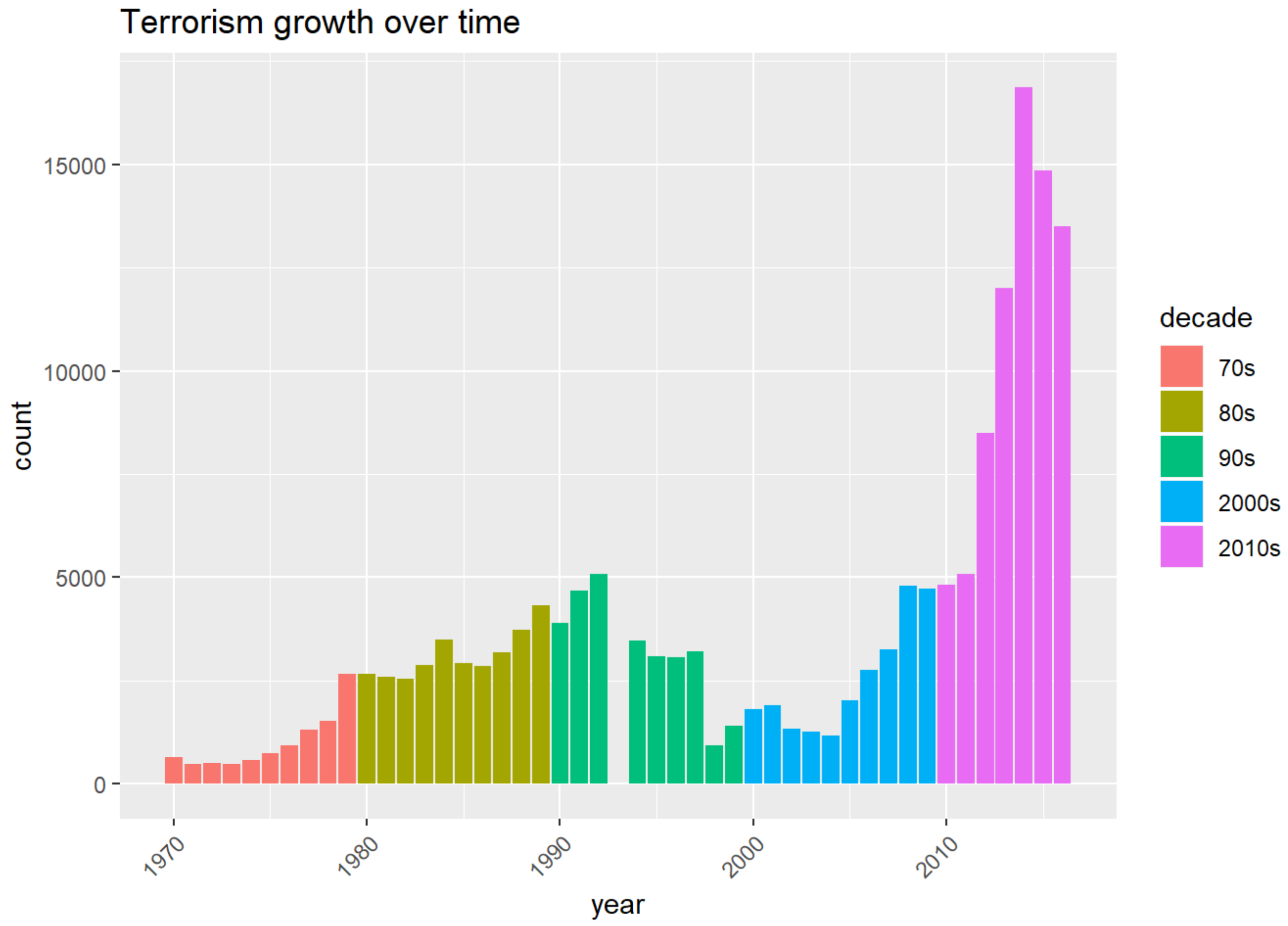
see below for decade breakdown - significant increase since 2010

```
#table
df %>%
  group_by(decade) %>%
  summarise(nr_of_attacks = n()) %>%
  arrange(desc(nr_of_attacks)) %>%head(n=10)
```

```
## # A tibble: 5 x 2
##   decade nr_of_attacks
##   <fct>          <int>
## 1 2010s          75589
## 2 80s            31159
## 3 90s            28766
## 4 2000s          24997
## 5 70s             9839
```

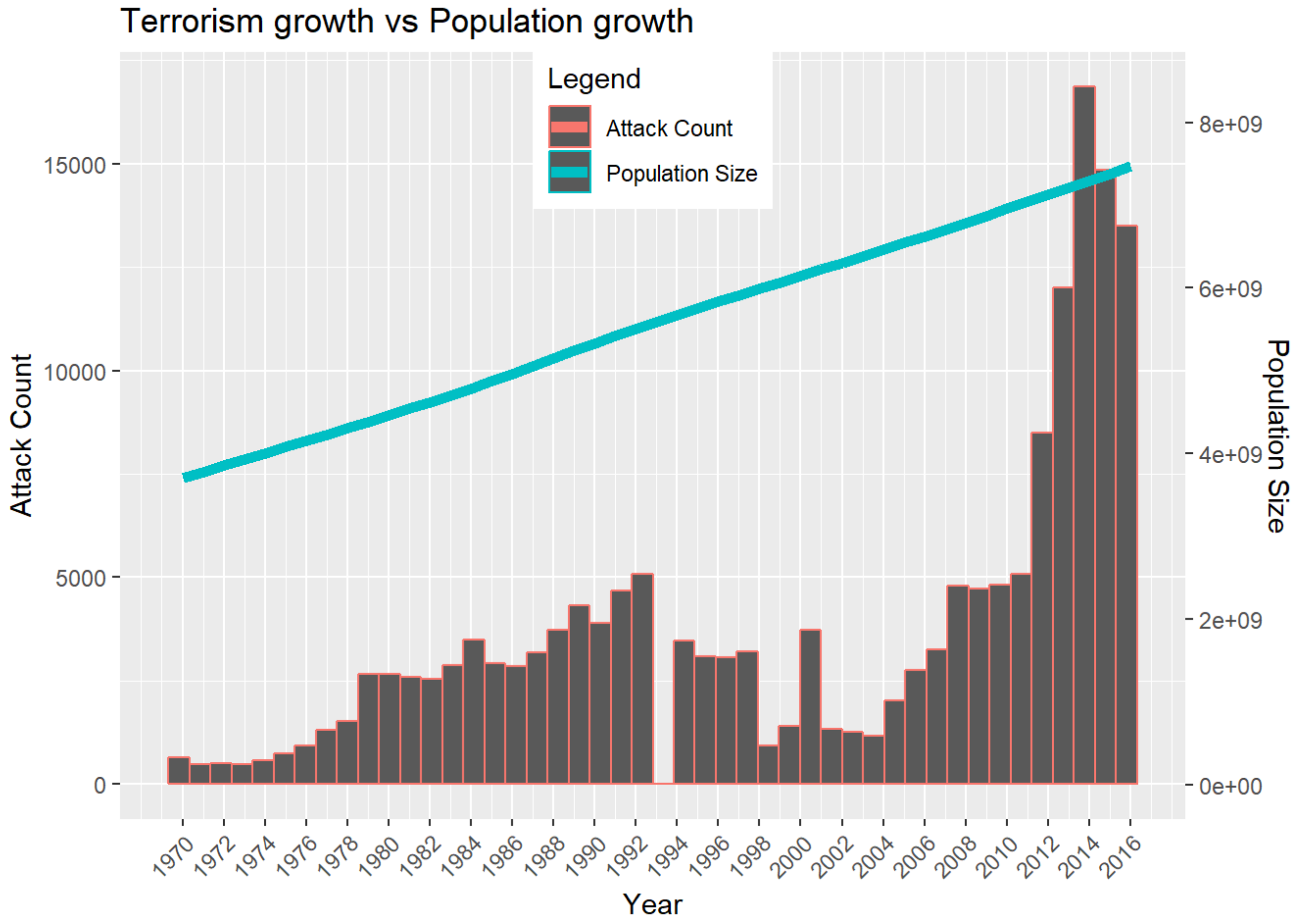
```
#visual
ggplot(data=df, aes(x=year, fill=decade)) +
  geom_histogram(stat='count') +
  theme(axis.text.x= element_text(angle=45, hjust=1)) +
  labs(title='Terrorism growth over time')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



2.1.2 Take into account world population growth

```
#get just the world population by year
popworld <- pop %>%
  filter(Location == "World") %>%
  select(-Location)

#Join to the dataframe based on year.
df2 <- inner_join(df, popworld, by= c("year" = "Time"))

#plot
p1 <- ggplot(data=df2, aes(x=year)) +
  geom_histogram(aes(col='Attack Count'), bins=46) +
  theme(axis.text.x= element_text(angle=45, hjust=1)) +
  scale_x_continuous(breaks=seq(1970, 2016, 2))

p1 +
  geom_line(aes(y=PopTotal/ 500, col='Population Size'), size=2) +
  scale_y_continuous(sec.axis = sec_axis(~ . * 500000, name = "Population Size")) +
  labs(y = "Attack Count", x = "Year", colour = "Legend") +
  theme(legend.position = c(0.5, 0.9)) +
  labs(title='Terrorism growth vs Population growth')
```



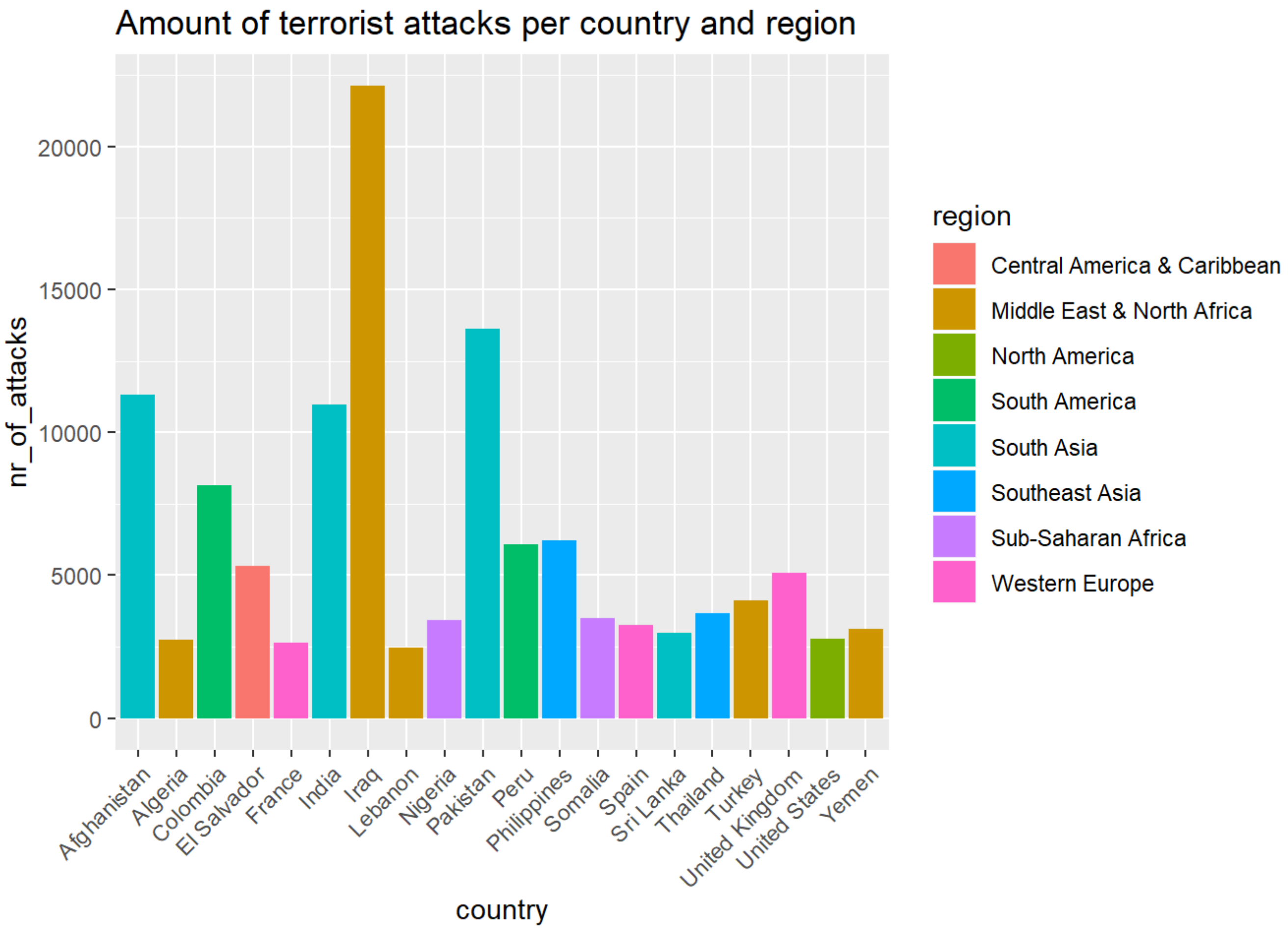
As we can see in the plot the general growth of terrorism seems to have exploded after 2010.

2.2 Locations of terrorism

Middle East and N Africa (~27% of total), South Asia (~24%) and S America (11%) are the top three regions in terms of number of attacks. Iraq (~12.9% of total), Pakistan,(~8%), Afhganistan (~6.6%), India(~6.4%) and Colombia (4.7%) are the top five countries in terms of number of attacks.

```
#table
top20_countries <- df %>%
  group_by(region, country) %>%
  summarise(nr_of_attacks = n()) %>%
  mutate(percent = nr_of_attacks/sum(nr_of_attacks))%>%
  arrange(desc(nr_of_attacks)) %>%
  head(n=20)

#visual by country
ggplot(data=top20_countries) +
  stat_summary(aes(x=country, y=nr_of_attacks, fill=region), geom="bar") +
  theme(axis.text.x= element_text(angle=45, hjust=1)) +
  labs(title='Amount of terrorist attacks per country and region')
```

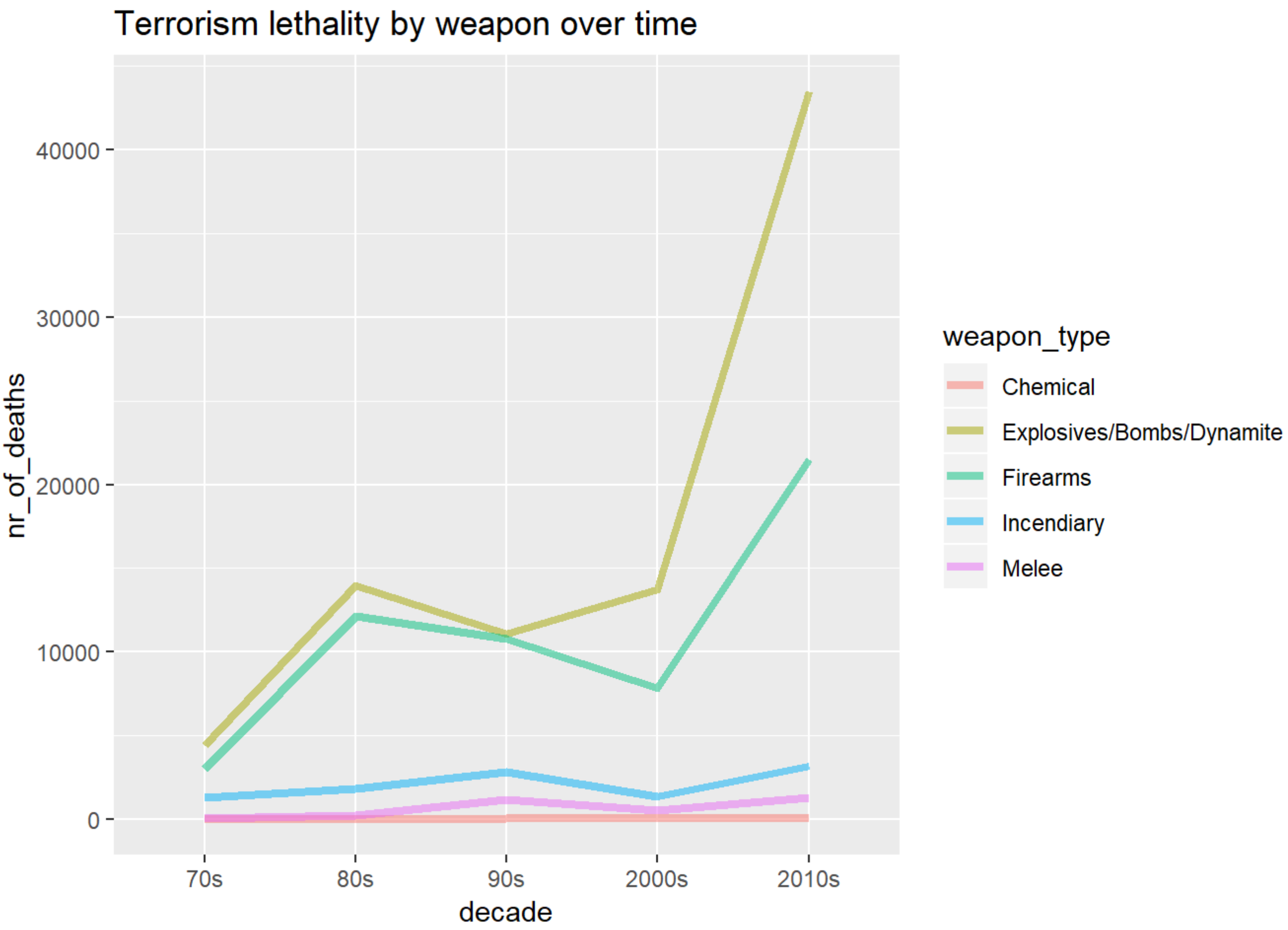
```
## No summary function supplied, defaulting to `mean_se()
```



2.3 Has terrorism become deadlier?

Over half of all deaths by terrorist attack have occurred during bomb attacks. The next deadliest weapon grouping is “Firearms” responsibile for ~32% of all Terror attack deaths.

```
#table
weapon_lethality <- df %>%
  filter(weapon_type != "Unknown") %>%
  select(decade, weapon_type, nkill)%>%
  group_by(decade,weapon_type)%>%
  summarise(nr_of_deaths = n())%>%
  top_n(n=5, wt=nr_of_deaths) %>%
  mutate(percent_deaths = (nr_of_deaths/sum(nr_of_deaths)*100))

#Visual by decade / weapon type
ggplot(data=weapon_lethality, aes(x=decade, y=nr_of_deaths, col=weapon_type, group= weapon_ty
pe)) +
  geom_line(size=1.5, alpha=0.5) +
  labs(title='Terrorism lethality by weapon over time')
```



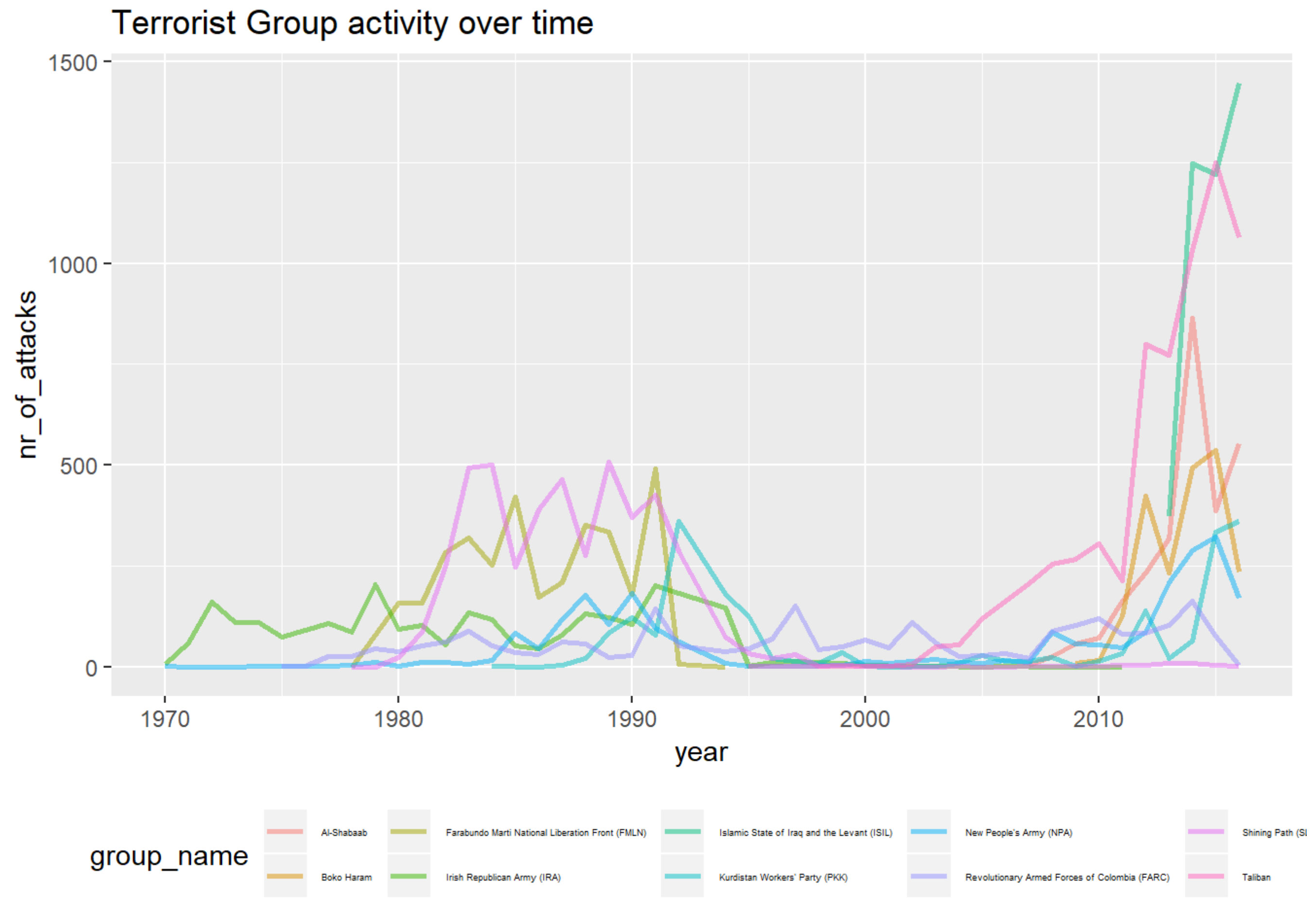
2.4 Activity of groups over time

First we identify the top ten Terror Groups in terms of number of attacks

```
top10_groups <- df %>%
  filter(group_name != "Unknown") %>%
  group_by(group_name) %>%
  summarise(nr_of_attacks = n()) %>%
  arrange(desc(nr_of_attacks)) %>%
  head(n=10)

top10_groups
```

```
## # A tibble: 10 x 2
##    group_name                                       nr_of_attacks
##    <chr>                                                    <int>
##  1 Taliban                                                   6575
##  2 Shining Path (SL)                                         4551
##  3 Islamic State of Iraq and the Levant (ISIL)               4287
##  4 Farabundo Marti National Liberation Front (FMLN)          3351
##  5 Al-Shabaab                                                2683
##  6 Irish Republican Army (IRA)                               2669
##  7 Revolutionary Armed Forces of Colombia (FARC)             2481
##  8 New People's Army (NPA)                                   2414
##  9 Kurdistan Workers' Party (PKK)                            2152
## 10 Boko Haram                                                2077
```

```
#table
top10_groups_activity <- df %>%
filter(df$group_name %in% c("Taliban", "Shining Path (SL)", "Islamic State of Iraq and the Le
vant (ISIL)", "Farabundo Marti National Liberation Front (FMLN)", "Al-Shabaab", "Irish Republ
ican Army (IRA)", "Revolutionary Armed Forces of Colombia (FARC)", "New People's Army (NPA)",
"Kurdistan Workers' Party (PKK)", "Boko Haram"))%>%
select(year, group_name)%>%
group_by(year, group_name) %>%
  summarise(nr_of_attacks = n())%>%
  arrange(desc(nr_of_attacks))%>%
   top_n(n=10, wt=nr_of_attacks)

#Visual by Top 10 Terror Group Activity  / decade since 1970
ggplot(data=top10_groups_activity, aes(x=year, y=nr_of_attacks, col=group_name, group= group_
name)) +
  geom_line(size=1, alpha=0.5) +
  theme(legend.position="right")+
  labs(title='Terrorist Group activity over time') +
  theme(legend.position="bottom", legend.text=element_text(size=3.5))
```

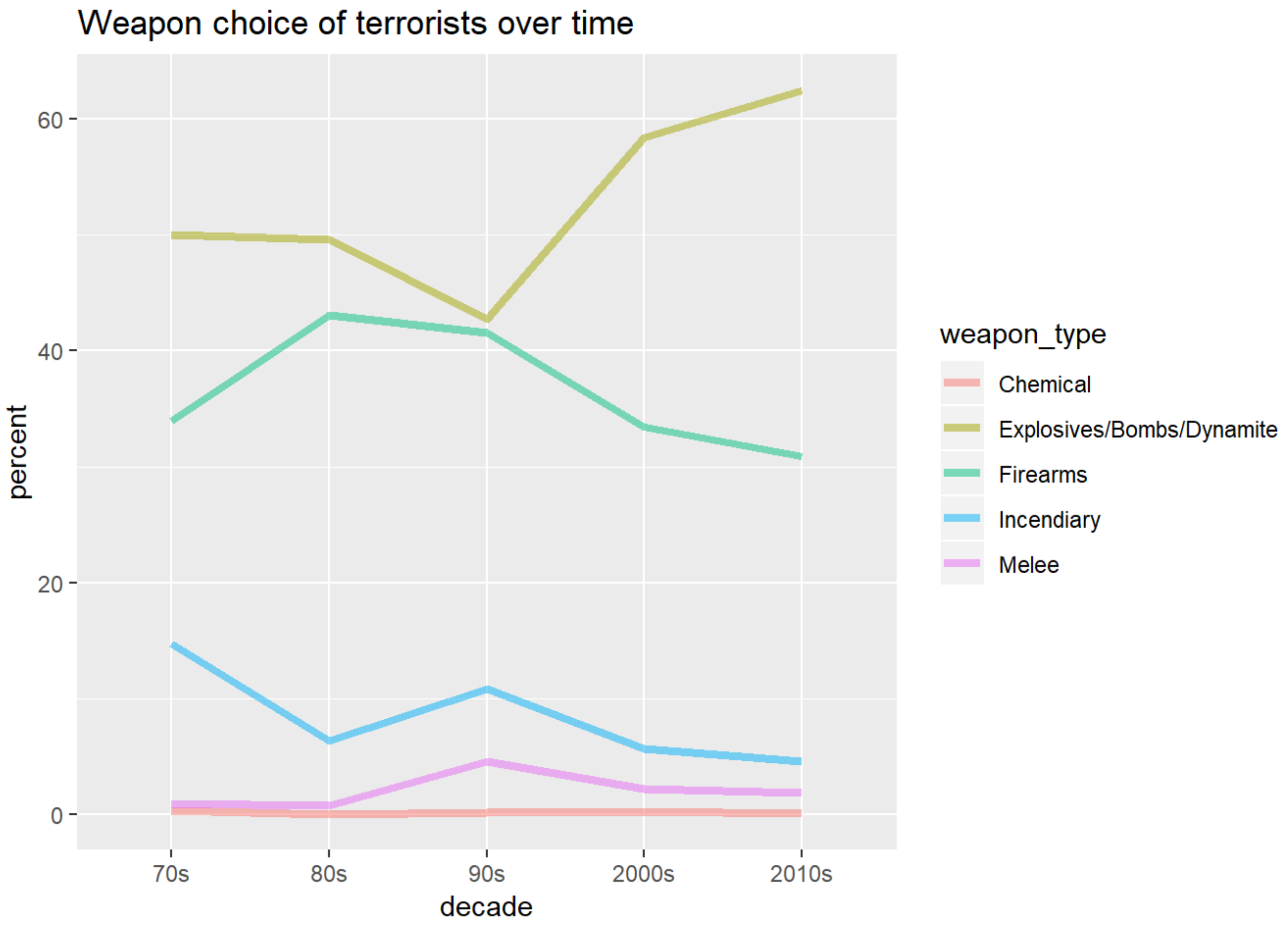


The current spike in Terror Activity (since 2000) has been maintained primarily by 4 x Main Groups - Taliban - Boko Haram - NPA - ISIL FARC have shown a small spike in activity since 2000 and IRA and Shining Path have shown a decrease in Activity since 2000

2.5 Weapon choice over time

Did technology growth change what weapons terrorist use?

```
dfweapons <- df %>%
  select(year, weapon_type, decade) %>%
  filter(weapon_type != "Unknown")

#table
top15_weapons <- dfweapons %>%
  group_by(decade, weapon_type) %>%
  summarise(nr_of_attacks = n()) %>%
  top_n(n=5, wt=nr_of_attacks) %>%
  mutate(percent = nr_of_attacks/sum(nr_of_attacks)*100) %>%
  arrange(decade, desc(nr_of_attacks))

#visual
ggplot(data=top15_weapons, aes(x=decade, y=percent, col=weapon_type, group= weapon_type)) +
  geom_line(size=1.5, alpha=0.5) +
  labs(title='Weapon choice of terrorists over time')
```

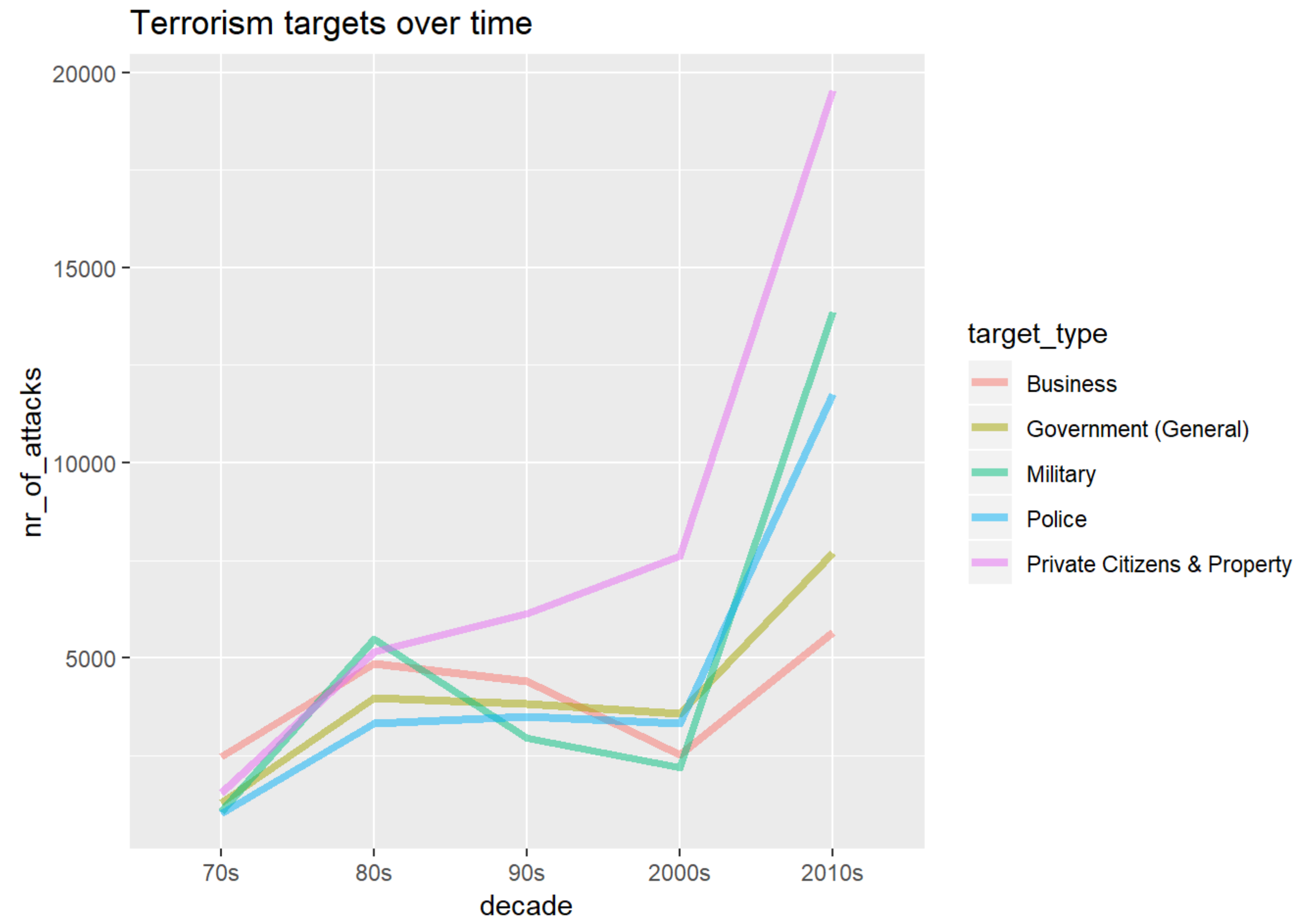


It seems that explosives and firearmas have been consistently the most popular. The whole top3 of weapon choices seems to have stayed completely consistent over the years.

2.6 Target choice over time

Have the targets changed? Have terrorists changed what targets they use?

```
dftargets <- df %>%
  select(year, target_type, target_sub_type, decade) %>%
  filter(target_type != "Unknown")

#table
dftargetstop <- dftargets %>%
  group_by(decade, target_type) %>%
  summarise(nr_of_attacks = n()) %>%
  top_n(n=5, wt=nr_of_attacks) %>%
  arrange(decade, desc(nr_of_attacks))

#visual
ggplot(data=dftargetstop, aes(x=decade, y=nr_of_attacks, col=target_type, group= target_typ
e)) +
  geom_line(size=1.5, alpha=0.5)+
  labs(title='Terrorism targets over time')
```

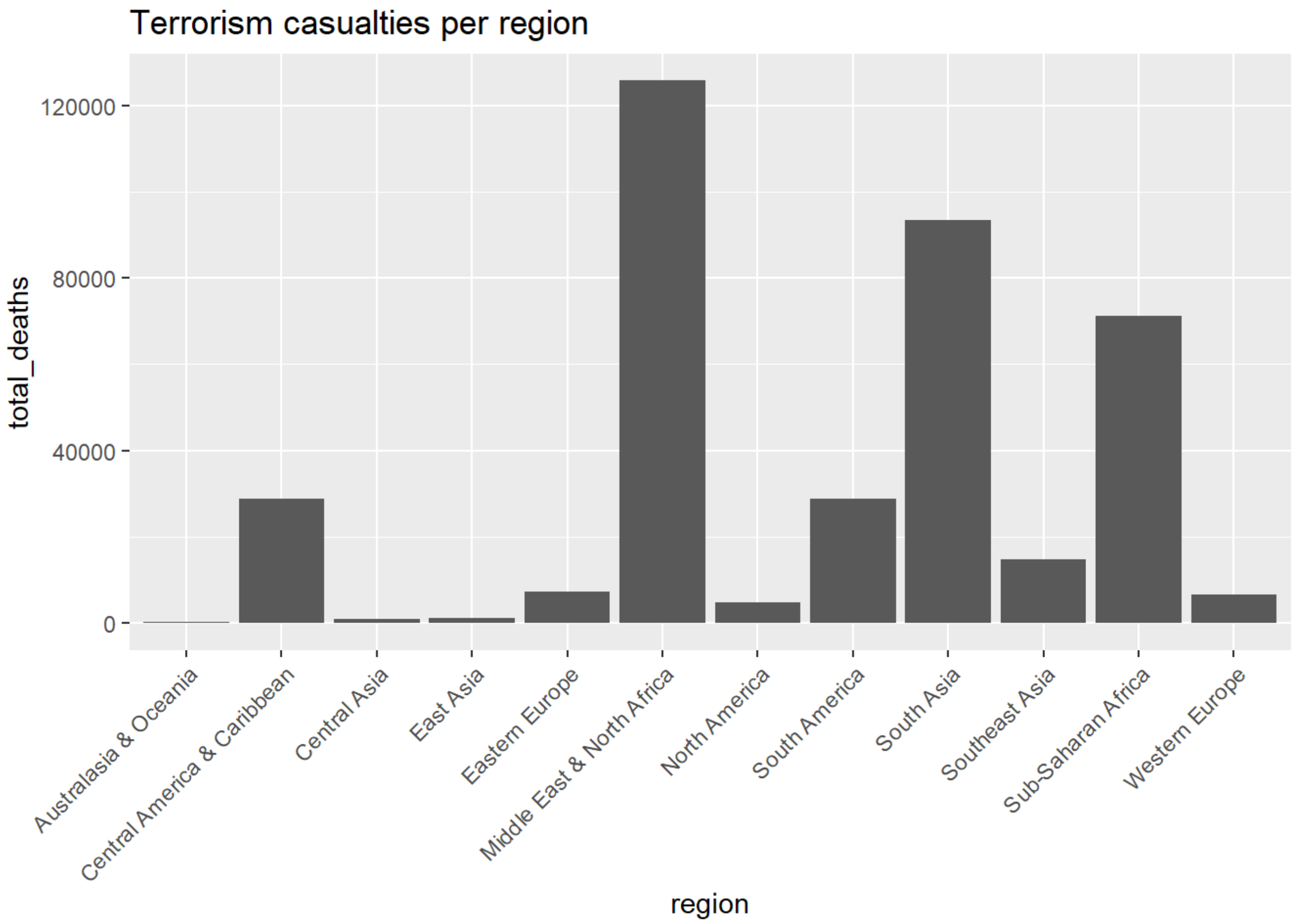


From the data we an see that private citizens have become a way bigger target group than before. It seems that violence has escelated to this innocent group. Besides that the Military has become a bigger target over the decades.

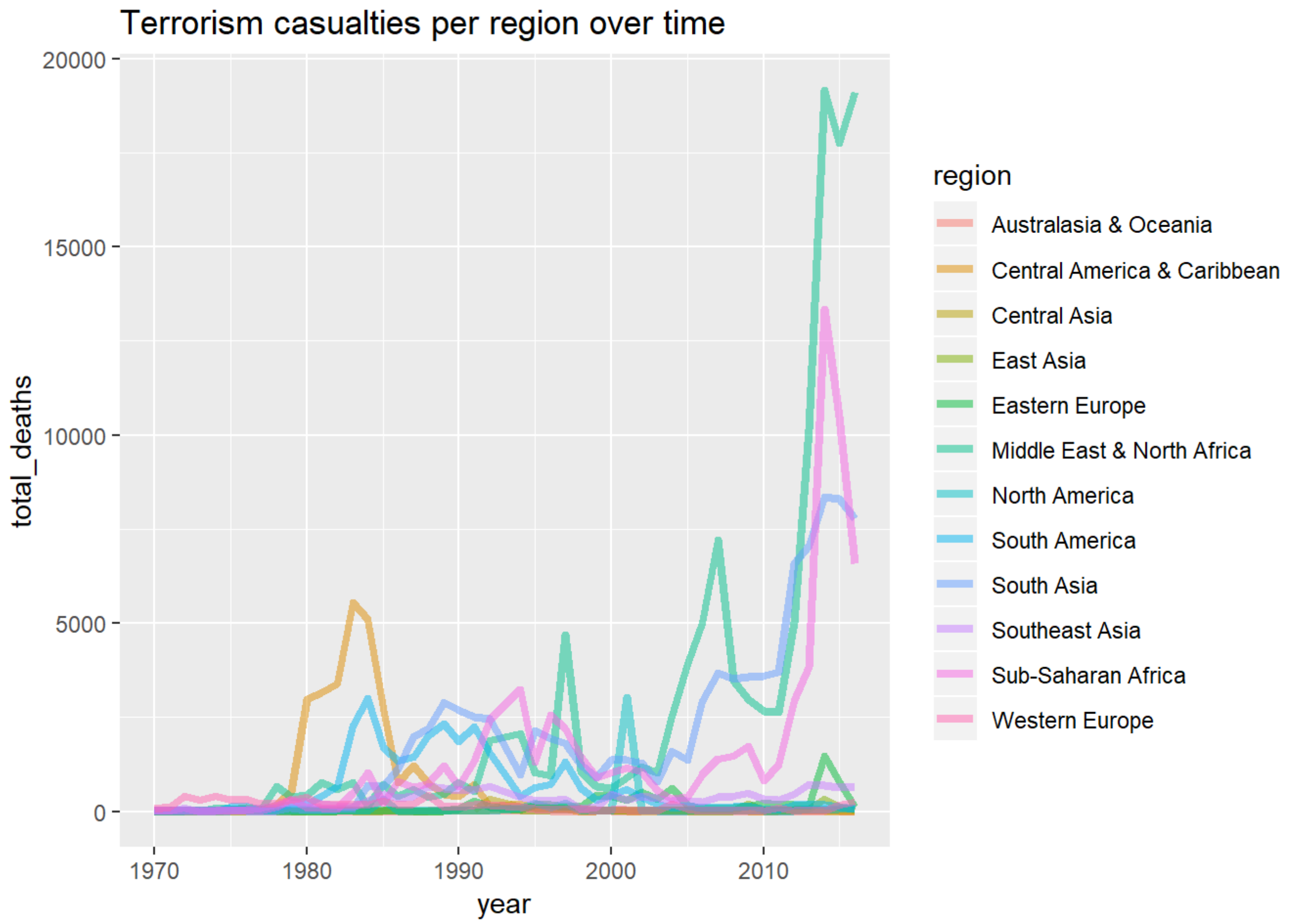
2.7 Location vs Mortality

```
#Mortality by Region
regionmort <- df %>%
  filter(nkill != 'Unknown') %>%
  select(region, nkill, nwound, year, group_name, decade) %>%
  group_by(region, year) %>%
  summarise(total_deaths = sum(nkill))

#Raw region amounts
ggplot(data=regionmort, aes(x=region, y=total_deaths)) +
  geom_histogram(stat='identity') +
  theme(axis.text.x= element_text(angle=45, hjust=1))+
  labs(title='Terrorism casualties per region')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



```
#and over time
ggplot(data=regionmort, aes(x=year, y=total_deaths, col=region, group= region)) +
  geom_line(size=1.5, alpha=0.5) +
  labs(title='Terrorism casualties per region over time')
```



The Middle East and North Africa have had an immense increase of deaths from 2003 upwards with links in well with incraesed instability in the region during and after the Iraq war.

3. Has terrorism gone up over the past few decades - taking into account population growth?

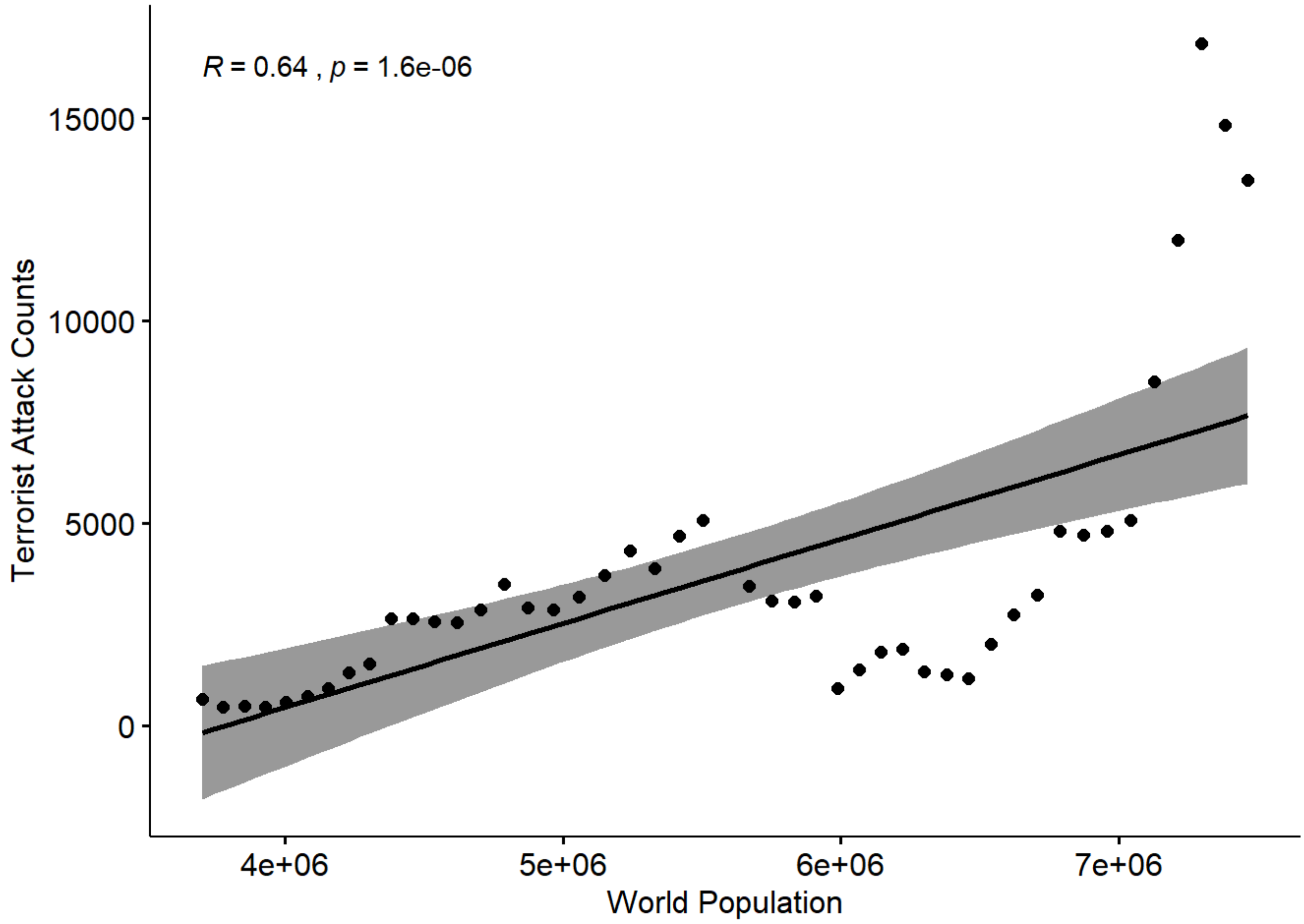
We have to take into account popoulation growth first. To do this we'll first check if population growth and amount of attacks per year are correlated.

3.1 Correlation analysis.

```
#first reframe the data to get it grouped per year
df3 <- df2 %>%
  group_by(year) %>%
  summarise(terrorist_attacks_count = n())

df3 <- inner_join(df3, popworld, by = c("year" = "Time"))

df3 <- df3 %>%
  mutate(decade =
           ifelse(year<1980, '70s',
                  ifelse(year < 1990, '80s',
                         ifelse(year < 2000, '90s',
                                ifelse( year < 2010, '2000s', '2010s')))))



df3$decade <- factor(df3$decade, levels=c("70s", "80s", "90s", "2000s", "2010s"))

cor.test(df3$PopTotal, df3$terrorist_attacks_count, method="pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  df3$PopTotal and df3$terrorist_attacks_count
## t = 5.5338, df = 44, p-value = 1.626e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4303294 0.7849295
## sample estimates:
##       cor
## 0.6406011
```

```
ggscatter(df3, y = "terrorist_attacks_count", x = "PopTotal",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          ylab = "Terrorist Attack Counts", xlab = "World Population")
```



It seems that there is a medium correlation (r=0.64, p<0.05) for world population and terrorist attack counts. However, they seem to disconnect when the population reaches more than 5.8 billion. Our linear model doesn’t explain the variance too well so perhaps we could try seeing if a polynomial model works better.

3.2 Are the amounts of attacks different over time?

Lets do linear regression to see if the variance in amount of attacks can be explained merely by population growth.

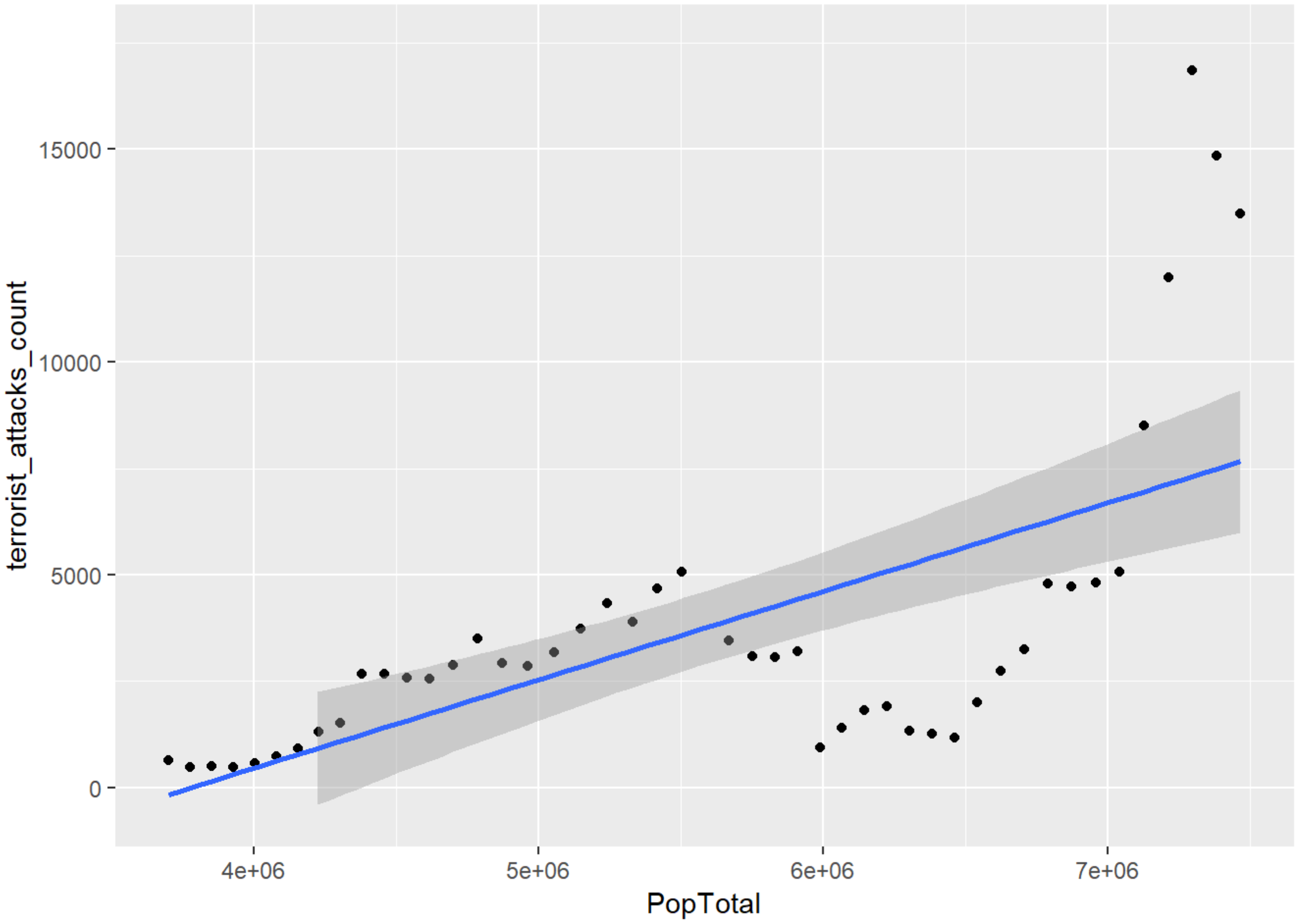
$$\begin{cases} H_0: & \text{variance-terrorist attacks properly explained by population growth} \\ H_a: & \text{variance-terrorist attacks not properly explained by populationgrowth} \end{cases}$$

3.2.1 Linear Model

```
m1 <- lm(data=df3, terrorist_attacks_count ~ PopTotal)
summary(m1)
```

```
##
## Call:
## lm(formula = terrorist_attacks_count ~ PopTotal, data = df3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4413.0 -1713.2   365.4   994.9  9544.8
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.857e+03  2.132e+03  -3.686 0.000623 ***
## PopTotal     2.079e-03  3.757e-04   5.534 1.63e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2884 on 44 degrees of freedom
## Multiple R-squared:  0.4104, Adjusted R-squared:  0.397
## F-statistic: 30.62 on 1 and 44 DF,  p-value: 1.626e-06
```

```
#Adjusted R-squared: 0.397 with p<0.05.

ggplot(data=df3, aes(x=PopTotal, y=terrorist_attacks_count)) +
  geom_point() +
  geom_smooth(method="lm",formula= y ~ x)+
  scale_y_continuous(limits = c(-500, 17500))
```



The Linear Model is significant but explains the variance in terrorist attacks count quite poorly with an adjusted R-squared of 0.397 (p<0.05). This means we REJECT H0 as the variance is not properly explained by population growth, meaning there are other factors in play.

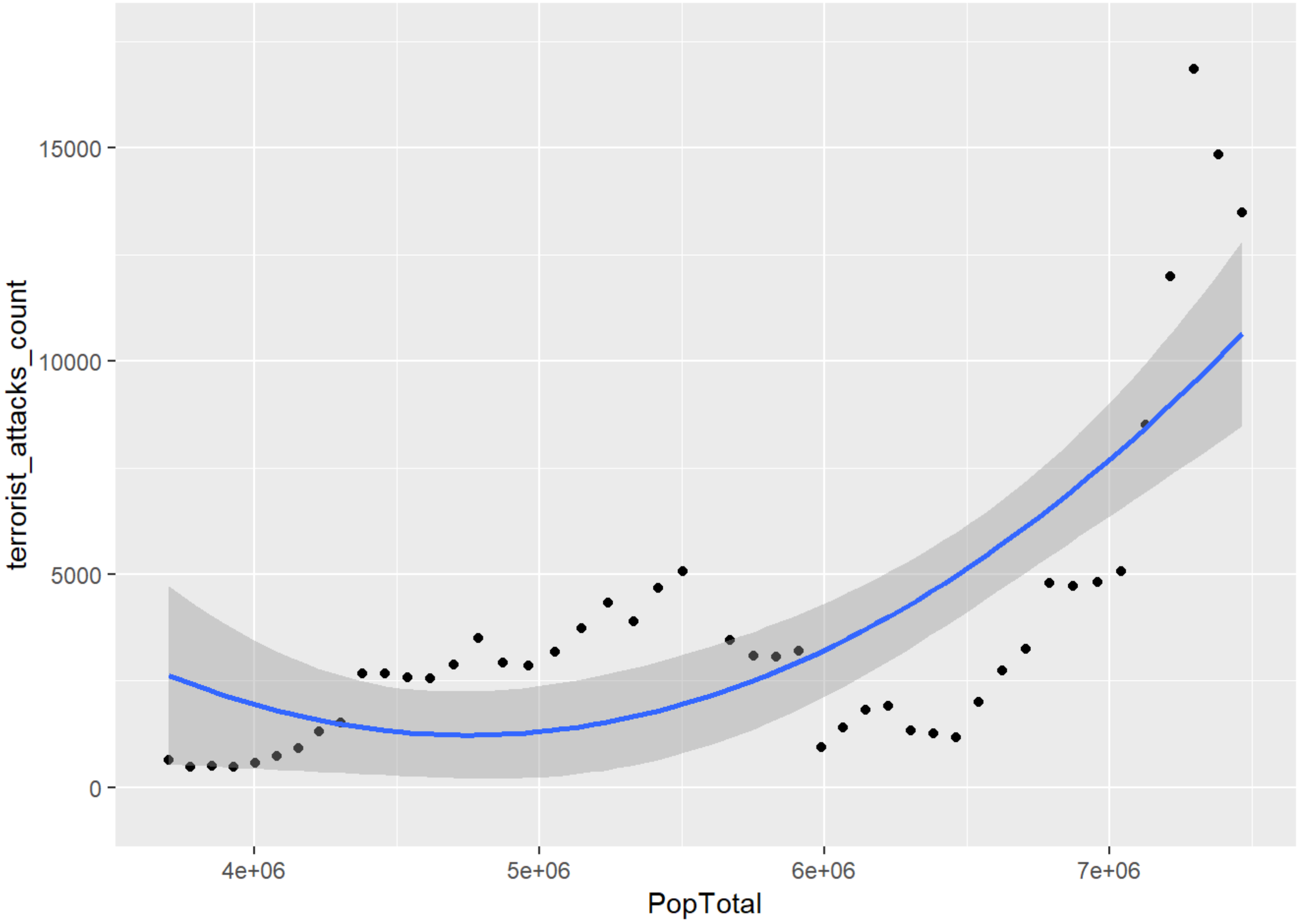
3.2.2 Polynomial regression

What seemed to be the case in our linear was that after a while there was no linear relationship anymore as most of the points fell outside of the 95% conf interval of the fitted line. Let's see if we can better explain the data with a quadratic term of PopTotal. It should ofcourse be kept in mind that we are heavily overfitting the model by adding these polynomial terms. However, because we suspect a non-linear relationship based on our initial analysis its an interesting thing to see on what kind of order the terrorist attack count has diverted from normal growth through population gains. Beside, that we can try and making some predictions about future amount of terrorist attacks based on our better fitted model since the dataset lacks other interesting variables to predict terrorist attacks.

```
m2 <- lm(data=df3, terrorist_attacks_count ~ PopTotal + I(PopTotal^2))
summary(m2)
```

```
##
## Call:
## lm(formula = terrorist_attacks_count ~ PopTotal + I(PopTotal^2),
##     data = df3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3804.6 -2030.3    47.9  1661.4  7339.5
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.005e+04  1.024e+04   2.935 0.005330 **
## PopTotal      -1.213e-02  3.788e-03  -3.203 0.002560 **
## I(PopTotal^2)  1.277e-09  3.391e-10   3.766 0.000499 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2530 on 43 degrees of freedom
## Multiple R-squared:  0.5566, Adjusted R-squared:  0.536
## F-statistic: 26.99 on 2 and 43 DF,  p-value: 2.544e-08
```

```
#Adjusted R-squared: 0.536 with p<0.05.

ggplot(data=df3, aes(x=PopTotal, y=terrorist_attacks_count)) +
  geom_point() +
  geom_smooth(method="lm",formula= y ~ x + I(x^2)) +
  scale_y_continuous(limits = c(-500, 17500))
```



This already explains the variance in terrorist attack count much better with an adjusted R-squared of 0.536 (p<0.05) but still most points fall outside of the confidence interval.

3.2.3 Third order Polynomial regression

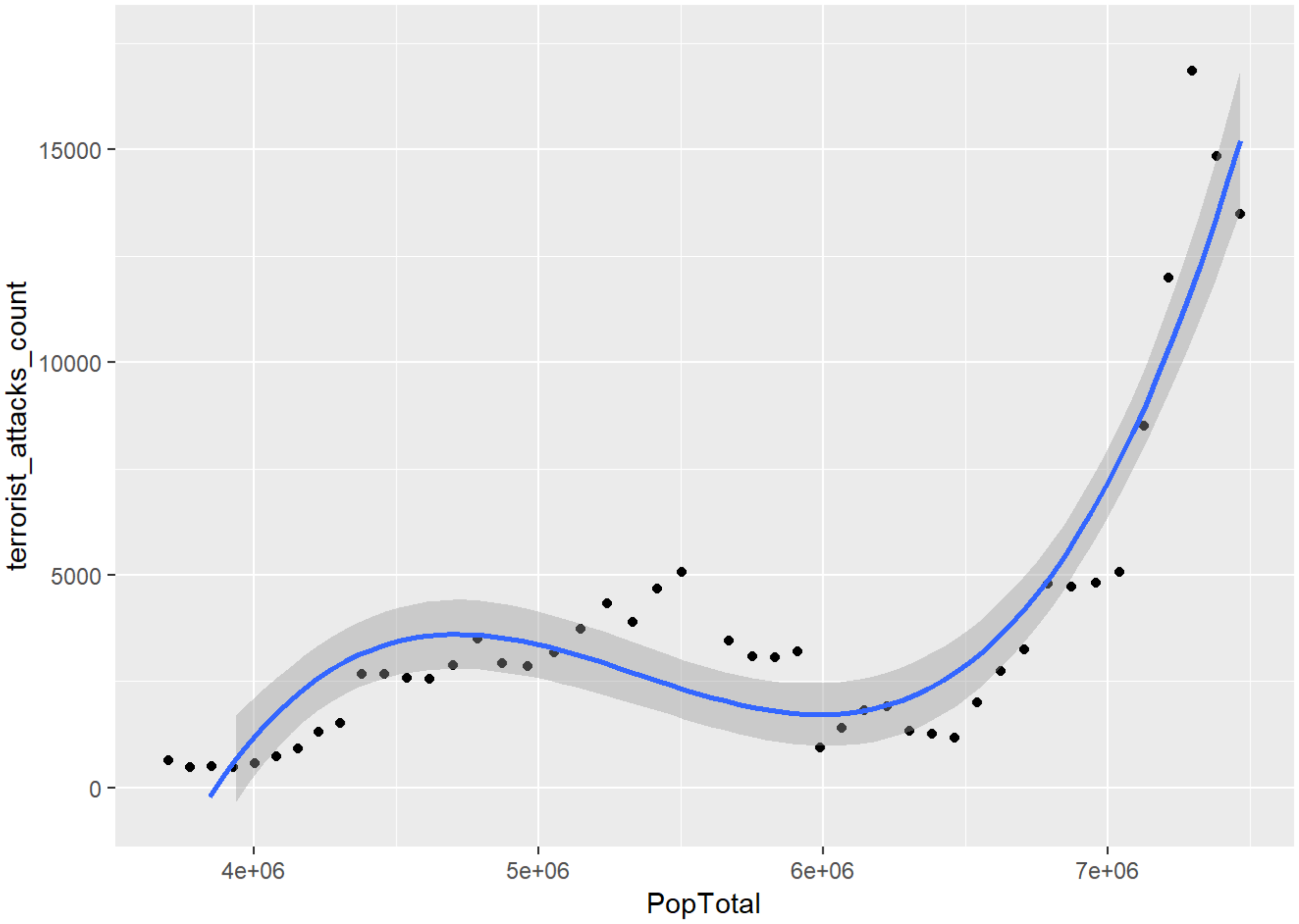
Lets see if we can make a better fit with higher order versions of the population variable.

```
m3 <- lm(data=df3, terrorist_attacks_count ~ PopTotal + I(PopTotal^2) + I(PopTotal^3))
summary(m3)
```

```
## 
## Call:
## lm(formula = terrorist_attacks_count ~ PopTotal + I(PopTotal^2) + 
##     I(PopTotal^3), data = df3)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -2663.9  -987.3  -472.8  1188.6  5061.7 
## 
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   -2.641e+05  3.358e+04  -7.865 8.65e-10 ***
## PopTotal       1.540e-01  1.879e-02   8.197 2.97e-10 ***
## I(PopTotal^2) -2.920e-08  3.428e-09  -8.519 1.07e-10 ***
## I(PopTotal^3)  1.820e-15  2.043e-16   8.907 3.15e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1506 on 42 degrees of freedom
## Multiple R-squared:  0.8465, Adjusted R-squared:  0.8356 
## F-statistic: 77.23 on 3 and 42 DF,  p-value: < 2.2e-16
```

```
#Adjusted R-squared: 0.8356 with p<0.05.

ggplot(data=df3, aes(x=PopTotal, y=terrorist_attacks_count)) +
  geom_point() +
  geom_smooth(method="lm",formula= y ~ x + I(x^2) + I(x^3)) +
  scale_y_continuous(limits = c(-500, 17500))
```

```
## Warning: Removed 3 rows containing missing values (geom_smooth).
```



Again, even a better fit is achieved and again, overfitting is incredibly high right now. Let’s see if we can see which of these models is better or worse through an ANOVA.

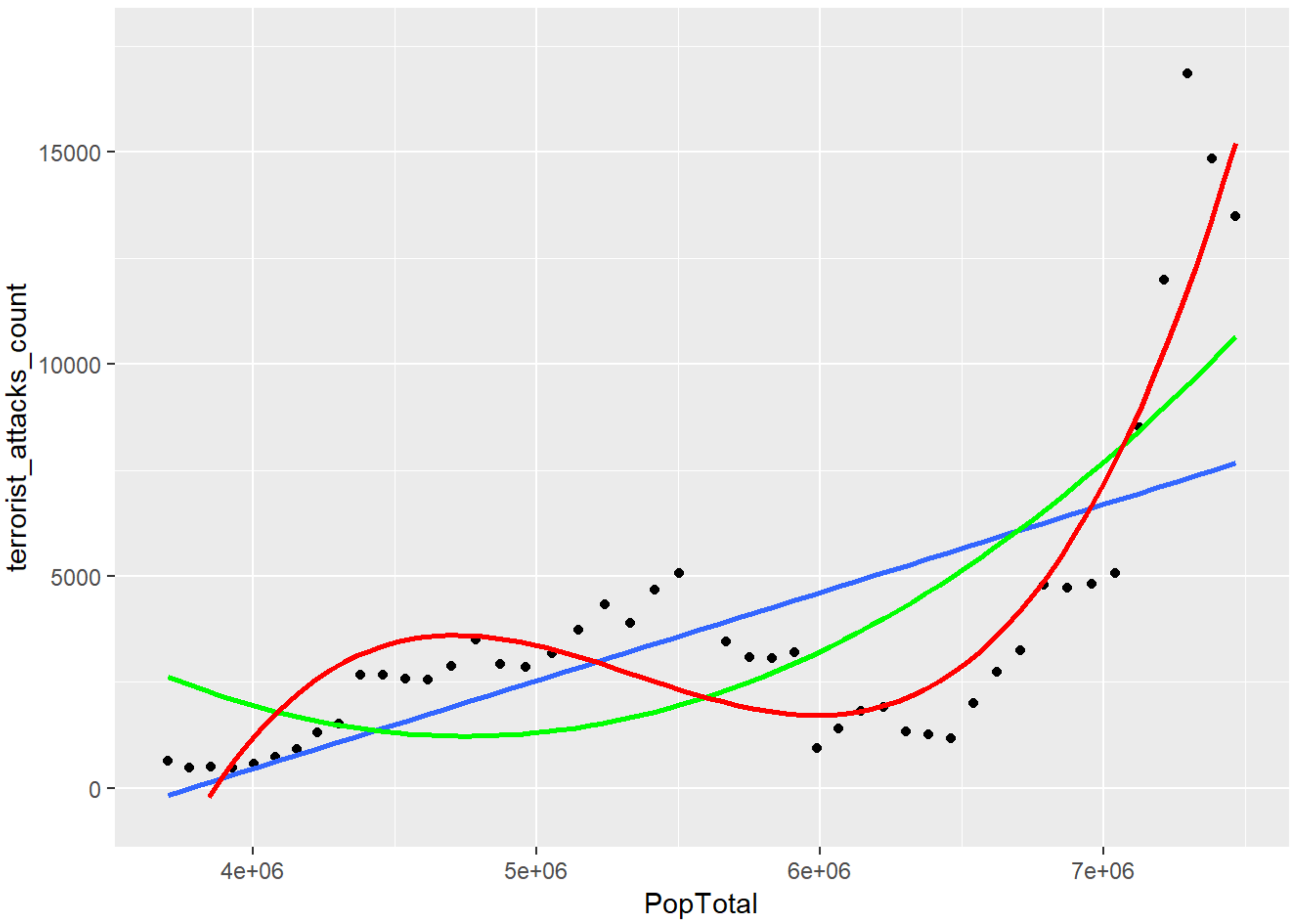
3.2.4 Model tests

Because the above models are all nested we can use ANOVA to see which one is better.

$$\begin{cases} H_0 : \text{SSE}m1 = \text{SSE}m2 = \text{SSE}m3 \\ H_a : \text{atleast two SSE's are different} \end{cases}$$

```
#visual, without confidence interval for visiblity
ggplot(data=df3, aes(x=PopTotal, y=terrorist_attacks_count)) +
  geom_point() +
  geom_smooth(method="lm",formula= y ~ x, se=F) +
  geom_smooth(method="lm",formula= y ~ x + I(x^2), se=F, col='green') +
  geom_smooth(method="lm",formula= y ~ x + I(x^2) + I(x^3), se=F, col='red') +
  scale_y_continuous(limits = c(-500, 17500))
```

```
## Warning: Removed 3 rows containing missing values (geom_smooth).
```



Which of these models is better? First we compare model 1 and model 2.

```
anova(m1, m2, test="F")
```

```
## Analysis of Variance Table
##
## Model 1: terrorist_attacks_count ~ PopTotal
## Model 2: terrorist_attacks_count ~ PopTotal + I(PopTotal^2)
##   Res.Df       RSS Df Sum of Sq      F    Pr(>F)
## 1     44 366026348
## 2     43 275234163  1  90792185 14.184 0.0004988 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#p<0.05 for model 2, meaning it's the better model. Is it also better than model 3?

anova(m2, m3, test="F")
```

```
## Analysis of Variance Table
##
## Model 1: terrorist_attacks_count ~ PopTotal + I(PopTotal^2)
## Model 2: terrorist_attacks_count ~ PopTotal + I(PopTotal^2) + I(PopTotal^3)
##   Res.Df       RSS Df Sum of Sq      F    Pr(>F)
## 1     43 275234163
## 2     42  95265353  1 179968810 79.344 3.153e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#p<0.05 for model 3, meaning it's the best model.
```

The ANOVA at the end to compare the models doesn’t explain too much as, yes, for this dataset the variance is better explained by the second and third model but we are overfitting the data massively at this point, especially since we only have an n=46 because we are looking at aggregates per year.

Not too much can be inferred from this except that clearly populating growth has decoupled from the occurance of terrorist attacks and some other variable(s) have entered the fold in the past decade or so that have pushed up the amount of terrorist attacks.

Before we start doing some predictions with our model, let’s check some assumptions with the help of the ‘car’ package.

3.3 Regression Diagnostics

3.3.1 Outlier Tests

One of the main reasons we suspect that population growth has decoupled as a linear predictor for terrorism attacks is because of the data from the last ~10 years. Let’s see first what is deemed an outlier with a Bonferroni Outlier Test.

```
outlierTest(m1)
```

```
##    rstudent unadjusted p-value Bonferonni p
## 44 3.972684         0.00026652      0.01226
```

```
outlierTest(m2)
```

```
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##    rstudent unadjusted p-value Bonferonni p
## 44 3.482553          0.0011736     0.053988
```

```
outlierTest(m3)
```

```
##    rstudent unadjusted p-value Bonferonni p
## 44 4.376442         8.1125e-05    0.0037318
```

Model 1 and Model 3 return a p<0.05 outlier which is observation. Let’s see that in a bit more detail.

```
df3[44, ]
```

```
## # A tibble: 1 x 4
##    year terrorist_attacks_count PopTotal decade
##   <int>                   <int>    <dbl> <fct>
## 1  2014                   16860 7298453. 2010s
```

So the year 2014 is a definite outlier, this is also very clear in the plots but we shouldn’t remove this as, since these our yearly aggregates, this is an important datapoint. Let’s also view some influence plots to see what years are the most distortionairy in regards to our linear model. Grid arrange seems to not accept influence plots so they are not in 1 plot.

```
infl1 <- influencePlot(m1, id.method = "noteworthy")
```

```
## Warning in plot.window(...): "id.method" n'est pas un paramètre graphique
```

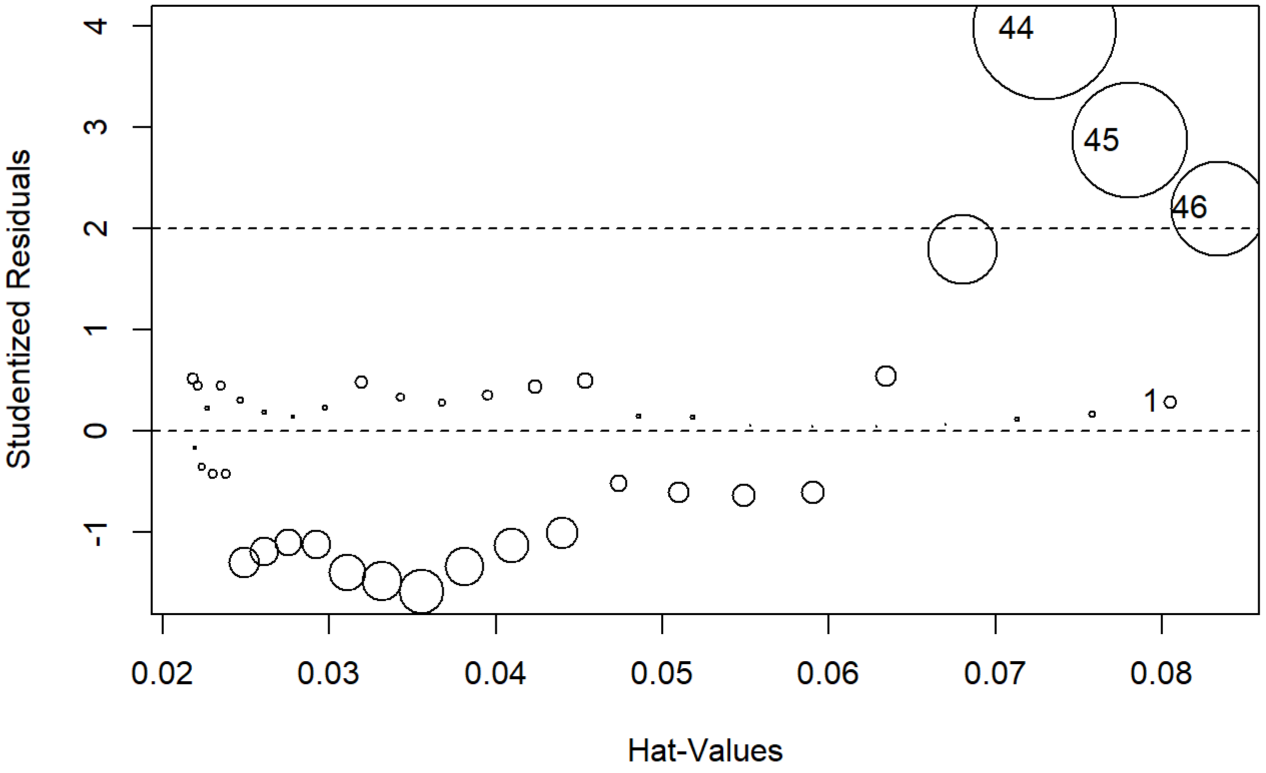
```
## Warning in plot.xy(xy, type, ...): "id.method" n'est pas un paramètre
## graphique
```

```
## Warning in axis(side = side, at = at, labels = labels, ...): "id.method"
## n'est pas un paramètre graphique

## Warning in axis(side = side, at = at, labels = labels, ...): "id.method"
## n'est pas un paramètre graphique
```

```
## Warning in box(...): "id.method" n'est pas un paramètre graphique
```

```
## Warning in title(...): "id.method" n'est pas un paramètre graphique
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "id.method" n'est
## pas un paramètre graphique
```



```
infl2 <- influencePlot(m2, id.method = "noteworthy")
```

```
## Warning in plot.window(...): "id.method" n'est pas un paramètre graphique
```

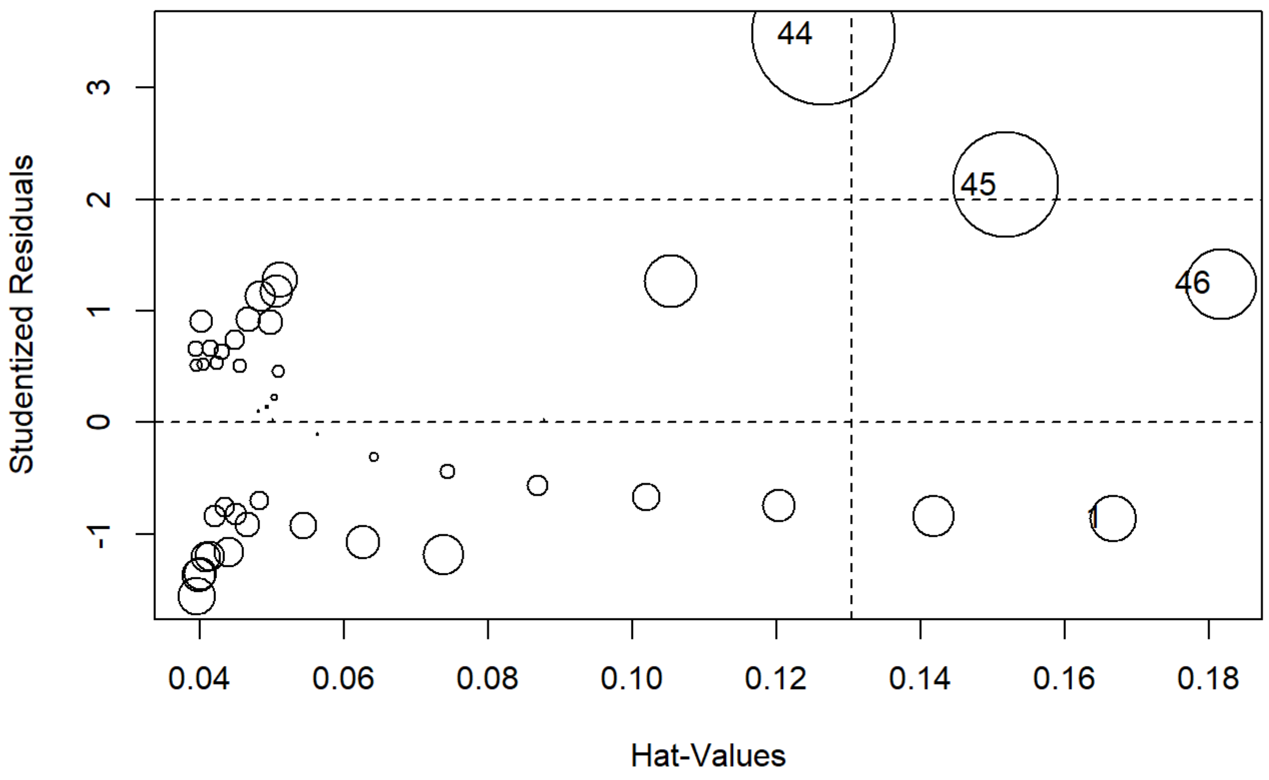
```
## Warning in plot.xy(xy, type, ...): "id.method" n'est pas un paramètre
## graphique
```

```
## Warning in axis(side = side, at = at, labels = labels, ...): "id.method"
## n'est pas un paramètre graphique

## Warning in axis(side = side, at = at, labels = labels, ...): "id.method"
## n'est pas un paramètre graphique
```

```
## Warning in box(...): "id.method" n'est pas un paramètre graphique
```

```
## Warning in title(...): "id.method" n'est pas un paramètre graphique
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "id.method" n'est
## pas un paramètre graphique
```



```
infl3 <- influencePlot(m3, id.method = "noteworthy")
```

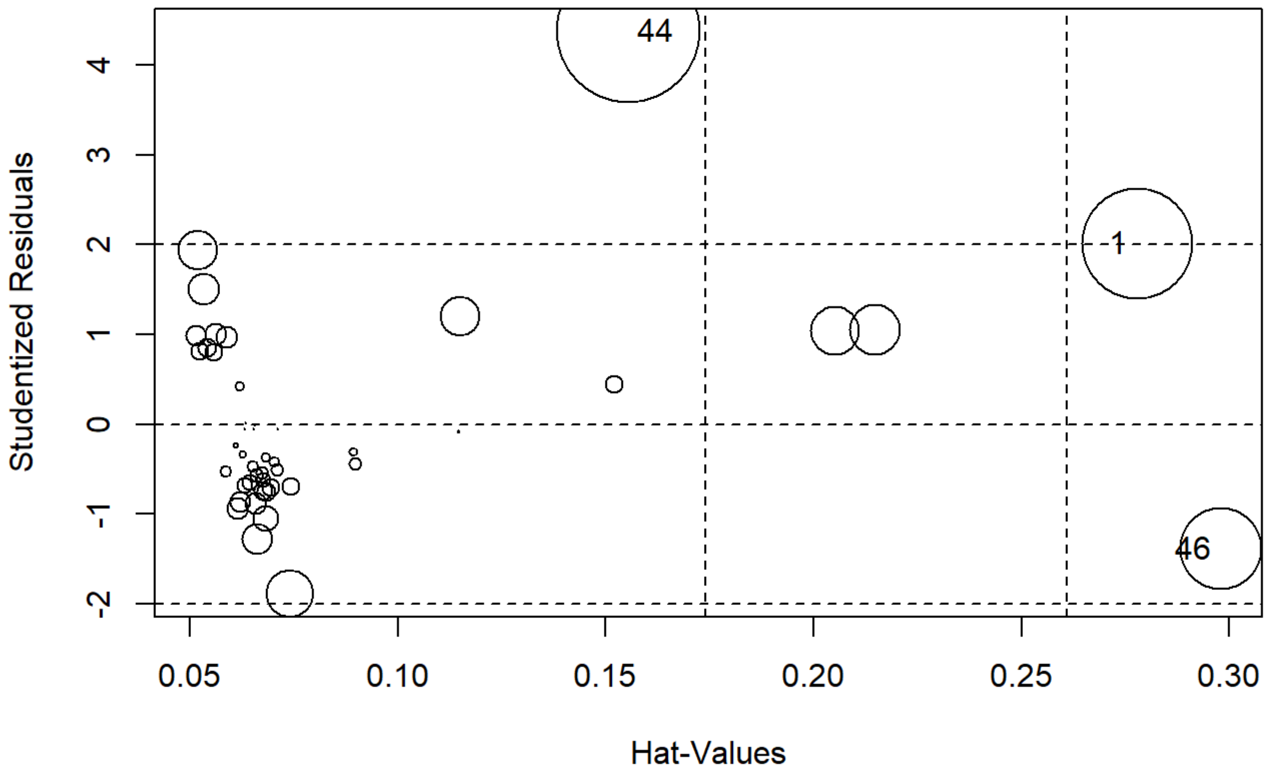
```
## Warning in plot.window(...): "id.method" n'est pas un paramètre graphique
## Warning in plot.xy(xy, type, ...): "id.method" n'est pas un paramètre
## graphique
```

```
## Warning in axis(side = side, at = at, labels = labels, ...): "id.method"
## n'est pas un paramètre graphique

## Warning in axis(side = side, at = at, labels = labels, ...): "id.method"
## n'est pas un paramètre graphique
```

```
## Warning in box(...): "id.method" n'est pas un paramètre graphique
```

```
## Warning in title(...): "id.method" n'est pas un paramètre graphique
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "id.method" n'est
## pas un paramètre graphique
```



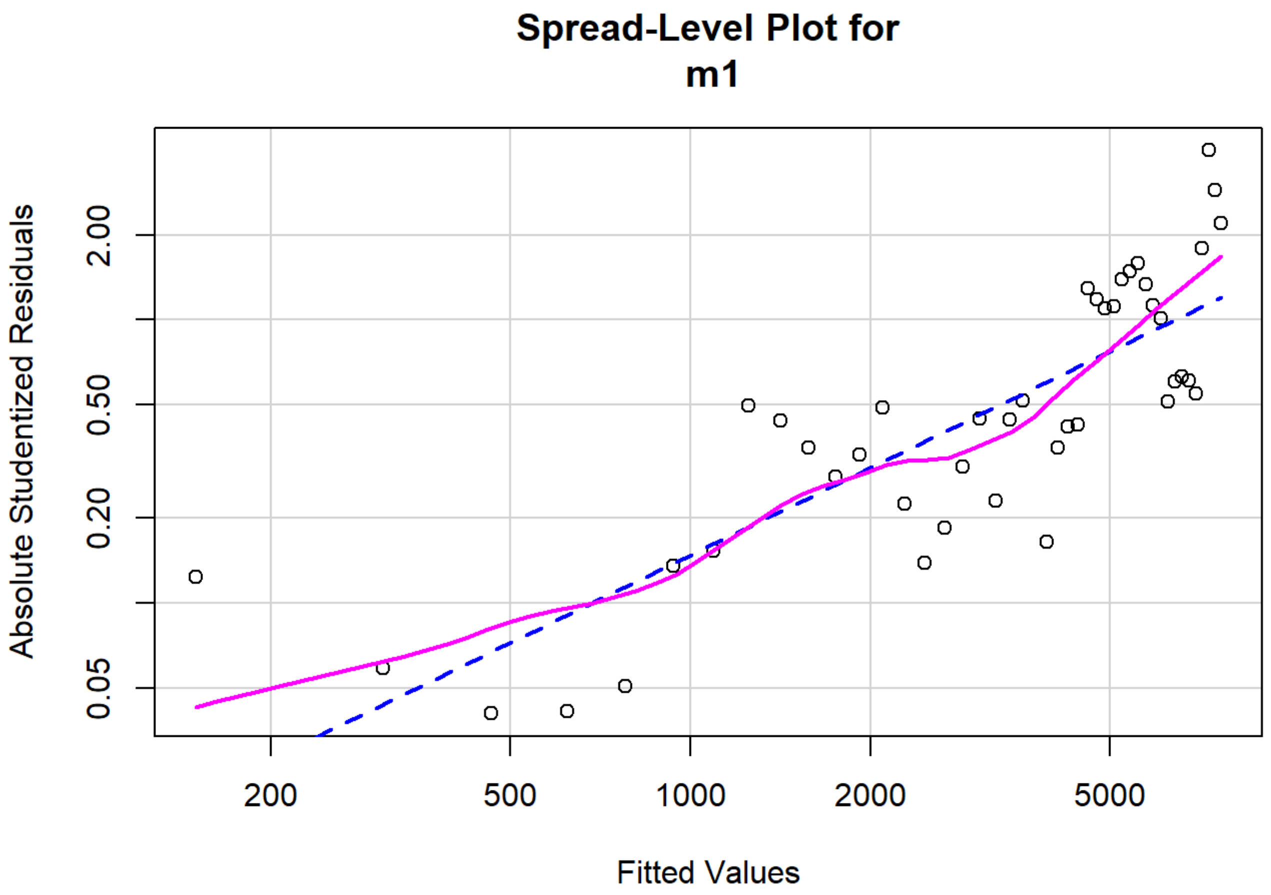
Constant recurring datapoints are 46 and 44, year 2016 and 2014 respectively.

3.3.2 Homoscedasticity

We expect none of the models to fare well under this assumption due to the fact that terrorist attacks seem to have completely decoupled from population levels about ~10 years ago. Next to that our second and third model employ higher order polynomials to model linearly a non-linear relationship between the two variables. Below are a few spread level plots but it’s quite obvious this assumption won’t be met. Again, grid arrange can’t handle plots with a numerical output.

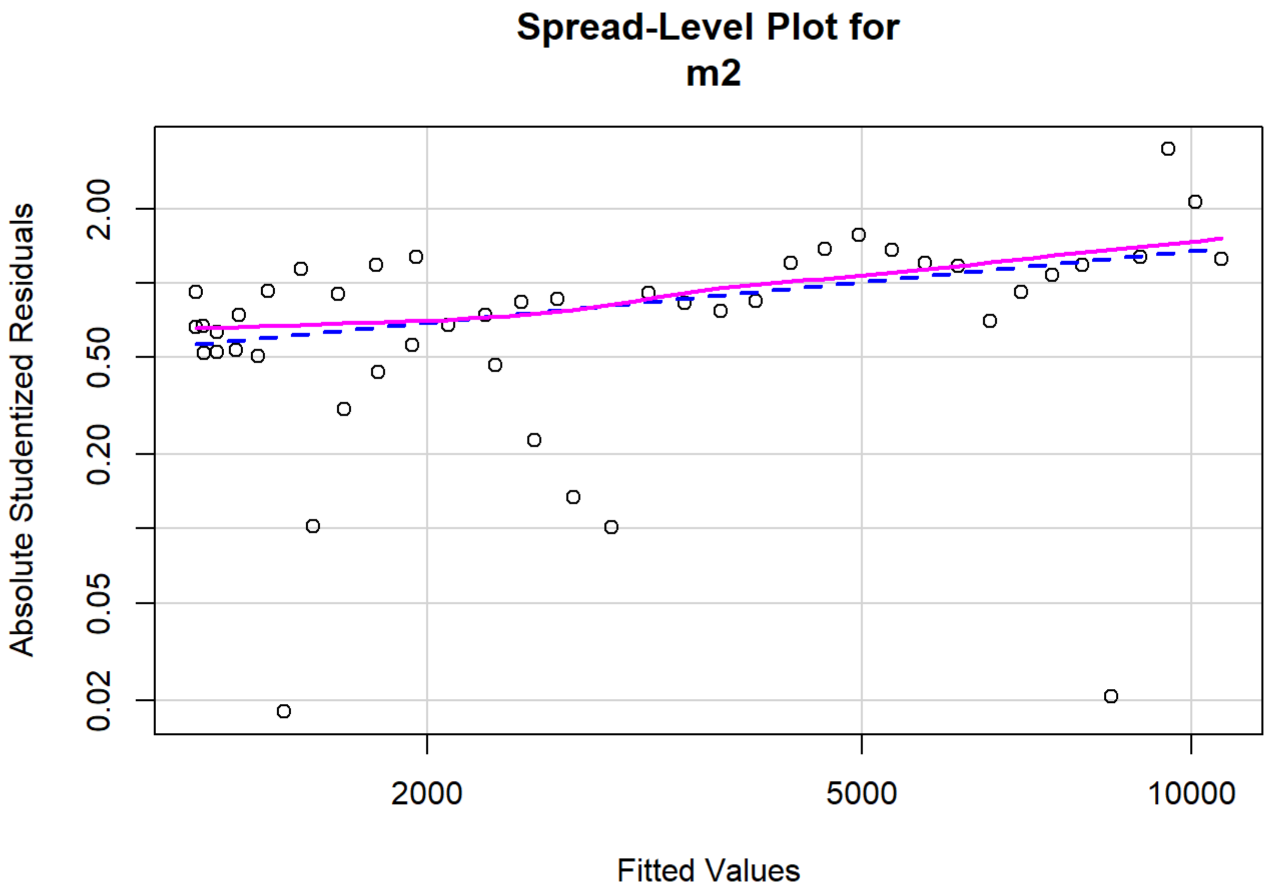
```
spreadLevelPlot(m1)
```

```
## Warning in spreadLevelPlot.lm(m1):
## 2 negative fitted values removed
```



```
##
## Suggested power transformation:  -0.02958061
```

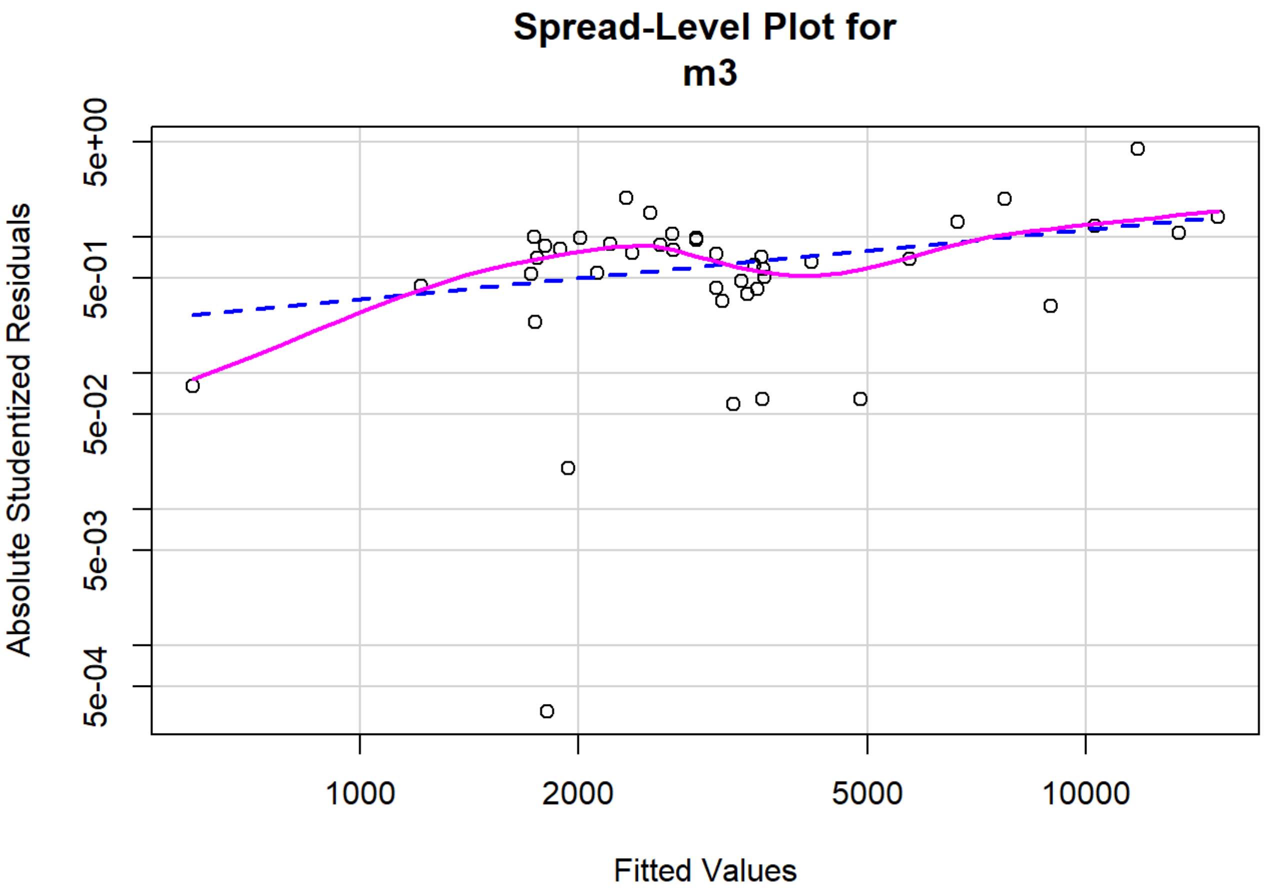
```
spreadLevelPlot(m2)
```



```
##
## Suggested power transformation:  0.584423
```

```
spreadLevelPlot(m3)
```

```
## Warning in spreadLevelPlot.lm(m3):
## 3 negative fitted values removed
```

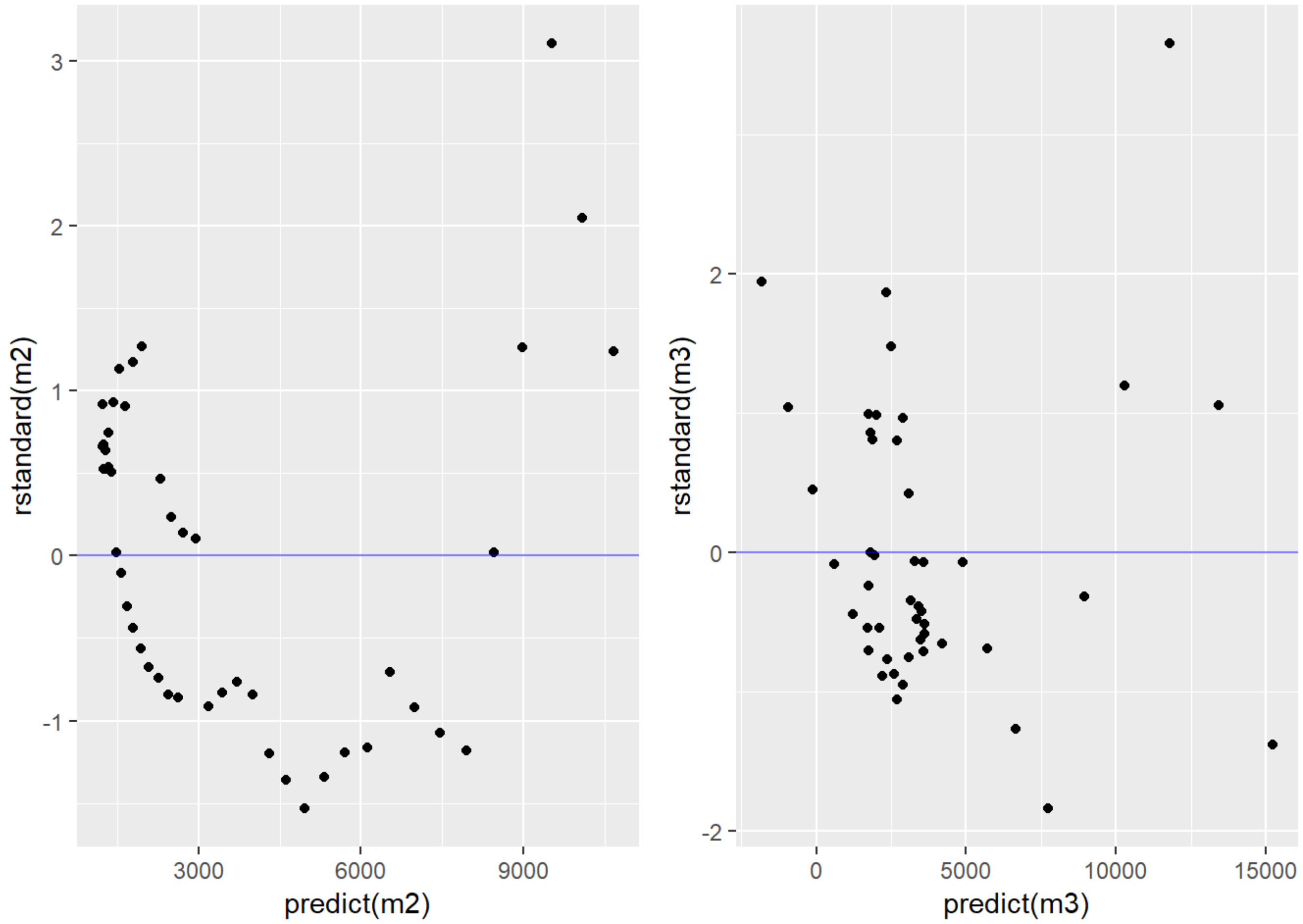


```
##
## Suggested power transformation:  0.4948599
```

3.3.3 Linearity

It was already quite clear from our original correlation plot that linearity is absent, this is why we added in nested polynomials, however it might still be interesting to see how the points are distributed in the two models with more than 1 independent variable.

```
lin2 <- qplot(predict(m2), rstandard(m2), geom="point") + geom_hline(yintercept=0, colour=I
("blue"), alpha=I(0.5))
lin3 <- qplot(predict(m3), rstandard(m3), geom="point") + geom_hline(yintercept=0, colour=I
("blue"), alpha=I(0.5))

grid.arrange(lin2, lin3, nrow=1)
```



As we can see they both lack linearity which is exactly as expected as we added in non-linear elements through the nested polynomials purely because we spotted a partly non-linear relationship in the first model.

3.3.4 Independence of residuals

Let’s check the independence of residuals with the help of a Durbin Watson test. Unfortunately since our data is essentially ‘panel data’, specifically Time Series, since it’s measurements over time from the same group (the world) the standard car packaged DWtest is not going to be useful. It will always tell us that the errors are autocorrelated.

```
durbinWatsonTest(m1)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.8601814     0.1851997       0
##  Alternative hypothesis: rho != 0
```

```
durbinWatsonTest(m2)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.8609013      0.234719       0
##  Alternative hypothesis: rho != 0
```

```
durbinWatsonTest(m3)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1        0.623576     0.6561569       0
##  Alternative hypothesis: rho != 0
```

As expected the we reject the H0 three times with p=0.

3.3.5 multicollinearity

Since we are working with nested polynomial models this diagnostic is also going to be less relevant as multicollinearity will be a thing.

```
vif(m2)
```

```
##      PopTotal I(PopTotal^2)
##      132.1659      132.1659
```

```
vif(m3)
```

```
##      PopTotal I(PopTotal^2) I(PopTotal^3)
##      9174.501     38105.664     10096.862
```

As expected the variance inflation factors are really high and normally this would mean dropping the extra variables. But since this is the rule only fro *linear combinations* of variables our model is exempt.

3.4 Model validation and forecasting

3.4.1 k-fold cross-validation

We can validate our models to see whether adding the polynomials of world population improved our predictive power. We'll be using k-fold validation to do so. First we'll set a seed for reproducibility, we won't divy up the data in a test and train set for now since this is all the *real data* we have.

```
set.seed(42)
```

Now let's use Caret for training the model with cross validation.

```
trainmethod <- trainControl(method="cv", number=5, returnData=TRUE, returnResamp='all')

model1 <- train(data=df3, terrorist_attacks_count ~ PopTotal, method='lm', trControl=trainmet
hod)
model2 <- train(data=df3, terrorist_attacks_count ~ PopTotal + I(PopTotal^2), method='lm', tr
Control=trainmethod)
model3 <- train(data=df3, terrorist_attacks_count ~ PopTotal + I(PopTotal^2) + I(PopTotal^3),
method='lm', trControl=trainmethod)
```

Let's analyse our obtained models. We don't use summarise() on the cross validated models because it seems that base-r function can't read the cross-validation information and just outputs the original r-squared values.

```
model1$resample
```

```
##       RMSE   Rsquared      MAE intercept Resample
## 1 3109.181 0.07229873 2706.775      TRUE    Fold1
## 2 2857.503 0.71447392 1671.191      TRUE    Fold2
## 3 2627.368 0.50363929 1866.278      TRUE    Fold3
## 4 3599.133 0.47382163 2115.796      TRUE    Fold4
## 5 2564.777 0.39918869 2179.964      TRUE    Fold5
```

```
model1
```

```
## Linear Regression
##
## 46 samples
##  1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 38, 36, 37, 37, 36
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   2951.592  0.4326845  2108.001
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
model2$resample
```

```
##       RMSE   Rsquared      MAE intercept Resample
## 1 2142.131 0.02507241 1995.081      TRUE    Fold1
## 2 1735.354 0.70300784 1523.539      TRUE    Fold2
## 3 2595.254 0.67675345 2293.707      TRUE    Fold3
## 4 2326.537 0.53421211 2079.168      TRUE    Fold4
## 5 4011.504 0.70570846 3392.330      TRUE    Fold5
```

```
model2
```

```
## Linear Regression
##
## 46 samples
##  1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 37, 37, 37, 36, 37
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   2562.156  0.5289509  2256.765
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
model3$resample
```

```
##       RMSE  Rsquared      MAE intercept Resample
## 1 2640.656 0.9048036 1878.053      TRUE    Fold1
## 2 1754.100 0.8636692 1568.825      TRUE    Fold2
## 3 1340.562 0.5582318 1067.235      TRUE    Fold3
## 4 1548.630 0.8006498 1191.823      TRUE    Fold4
## 5 1212.688 0.5872373 1012.291      TRUE    Fold5
```

```
model3
```

```
## Linear Regression
##
## 46 samples
##  1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 37, 38, 35, 37, 37
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   1699.327  0.7429184  1343.646
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

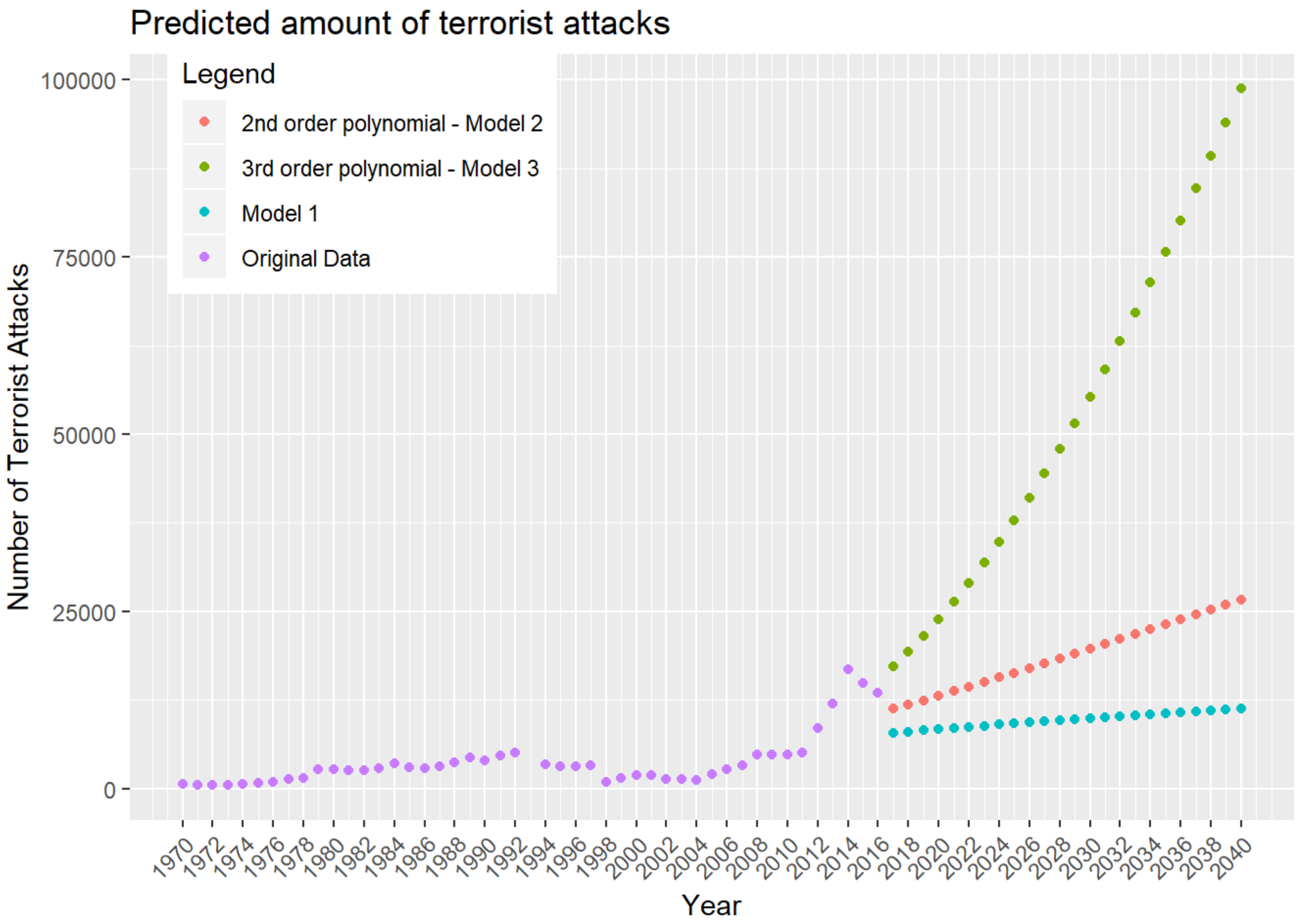
model1: Ajusted R-squared of 0.57 model2: Ajusted R-squared of 0.50 *model3: Ajusted R-squared of 0.79

The adjust r-squared has shifted up for model 1 (0.4 -> 0.49) while model 2 and model 3 shifted down (0.54 -> 0.50 and 0.84 -> 0.79 respectively)

3.4.2 standard forecasting

We can try using our slightly overfitted models to predict the amount of future terrorism attacks based on future PopTotal given by the United Nations predicted population dataset.

```r
#First make a dataframe with empty terrorist values with the project UN populations
futurepopworld <- futurepop %>%
  filter(Location == "World") %>%
  select(-Location)

futurepopworld$terrorist_attacks_count <- NA

#duplicate for three models
fpopworld1 <- futurepopworld
fpopworld2 <- futurepopworld
fpopworld3 <- futurepopworld

#predict new values of terrorist_attacks_count based on the three models.
fpopworld1$terrorist_attacks_count <- predict(object=m1, newdata=fpopworld1)

fpopworld2$terrorist_attacks_count <- predict(object=m2, newdata=fpopworld2)

fpopworld3$terrorist_attacks_count <- predict(object=m3, newdata=fpopworld3)

#filter until 2040
fpopworld1 <- filter(fpopworld1, Time < 2041)
fpopworld2 <- filter(fpopworld2, Time < 2041)
fpopworld3 <- filter(fpopworld3, Time < 2041)

#visual
ggplot() +
  geom_point(data=df3, aes(x=year, y=terrorist_attacks_count, col='Original Data')) +
  geom_point(data=fpopworld1, aes(x=Time, y=terrorist_attacks_count, col='Model 1')) +
  geom_point(data=fpopworld2, aes(x=Time, y=terrorist_attacks_count, col='2nd order polynomia
l - Model 2')) +
  geom_point(data=fpopworld3, aes(x=Time, y=terrorist_attacks_count, col='3rd order polynomia
l - Model 3')) +
  labs(title='Predicted amount of terrorist attacks') +
  theme(legend.position = c(0.2, 0.85)) +
  labs(x= 'Year', y= 'Number of Terrorist Attacks', colour = 'Legend') +
  scale_x_continuous(breaks = seq(1970, 2040, 2)) +
  theme(axis.text.x= element_text(angle=45, hjust=1))
```



As can be confirmed in the graph the 2nd order polynomial seems to be giving the most accurate prediction based on the upswing from the past decades but assuming that's just a blip in the data the first model gives the best prediction. To test this correctly we would have to compare it with new data coming in for the coming years. However, with ISIL currently being 'defeated' we suspect that the Model 1 prediction will be most correct.

Besides all that it should also be noted that due to the fact that the three models are all descriptive regression models made with the goal of obtaining a best fit, *all* predictions based on data outside of the range of the original data are pure extrapolation. This can be useful for short term prediction but in actuallity this falls apart quite fast with extremely wide confidence intervals. Due to this we decided to use the 'forecast' package to forecast the data as pure time-series data to forecast percentage changes instead of raw numbers.

3.4.3 Time-Series Forecasting

To do this we first have to transform the dataset into a time-serie frame. Let's also impute the missing year (1993) with an average value of the preceding and following year so that we can convert to time-series.

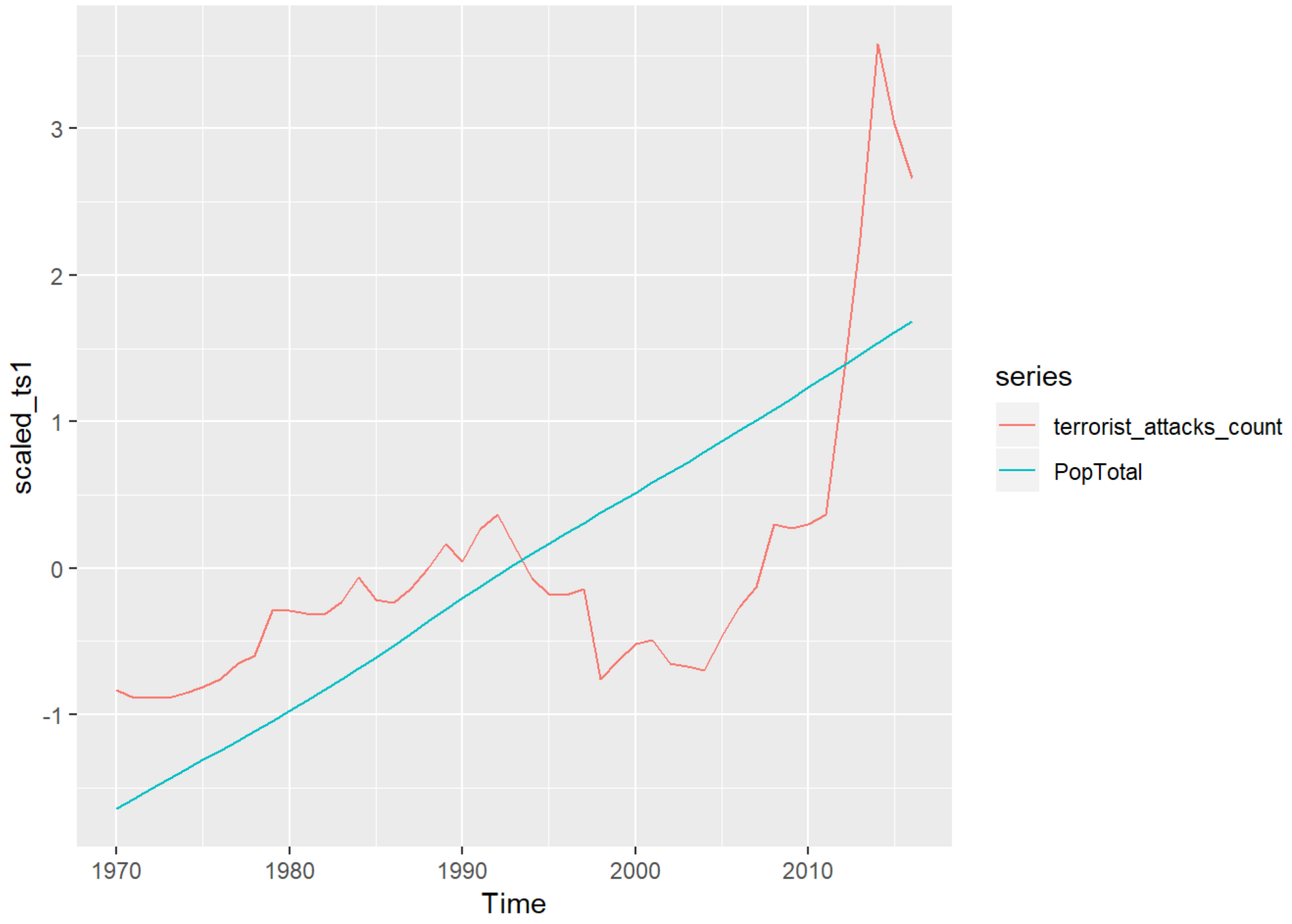
```
#for the missing value
df3[df3$year==1992 | df3$year == 1994, ]
```

```
## # A tibble: 2 x 4
##    year terrorist_attacks_count PopTotal decade
##   <int>                   <int>    <dbl> <fct>
## 1  1992                    5073 5504401. 90s
## 2  1994                    3458 5670320. 90s
```

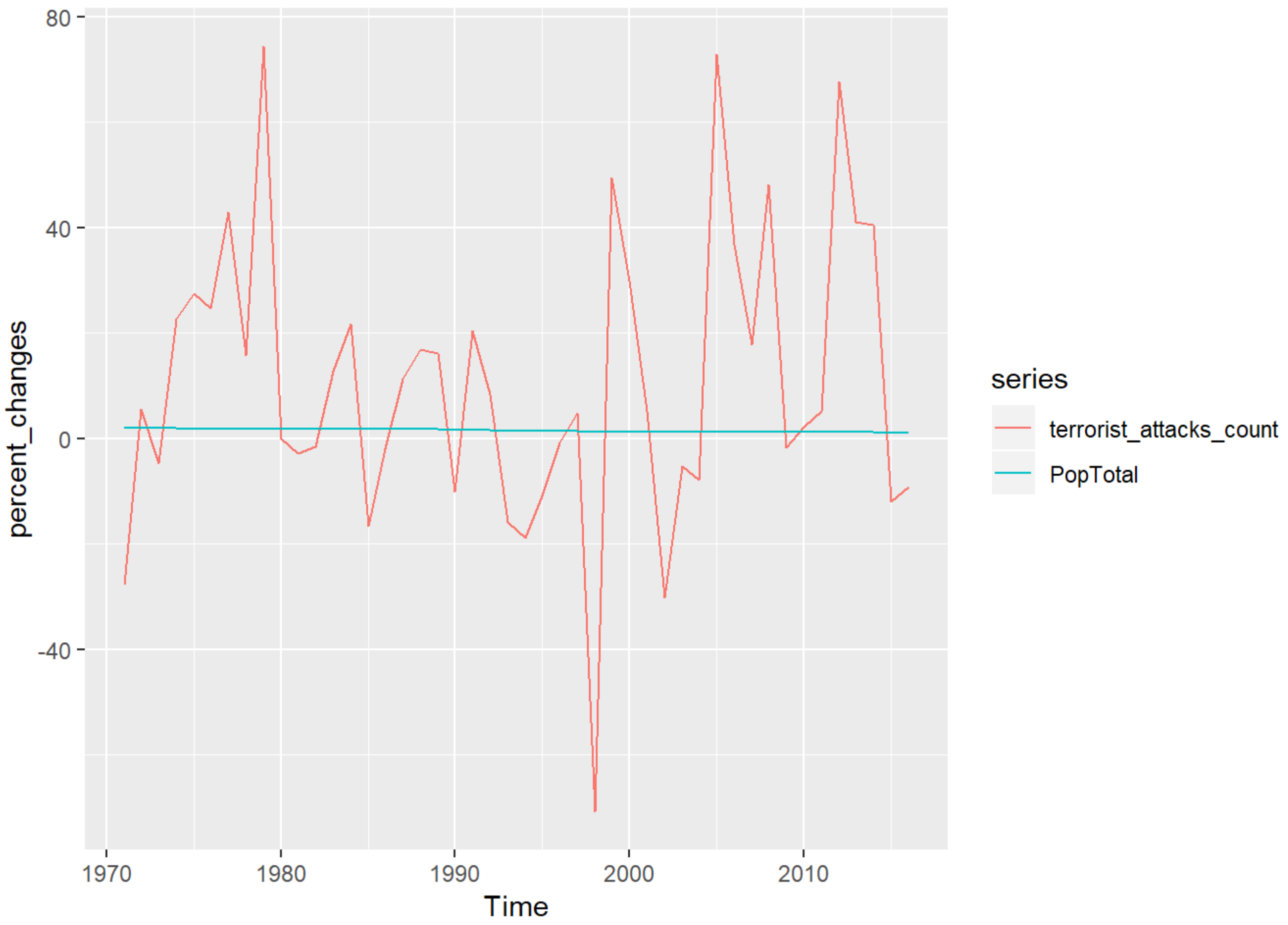
```
newrow <- data_frame(year=1993, terrorist_attacks_count=(5073+3458)/2-0.5, PopTotal=(5504401+
5670320)/2)

df4 <- df3 %>%
  select(-decade) %>%
  bind_rows(newrow)

df4 <- df4 %>%
  arrange(year)

values <- df4[, 2:3]

ts1 <- ts(values, start=1970,end=2016,frequency=1)

scaled_ts1 <- scale(ts1)

percent_changes <- diff(ts1)/ts1[-nrow(ts1),] * 100
```

So now we have a time-series data frame with percent changes. Let’s visualize how that looks. Because Ggplot can’t out of the box handle ts objects we’re using ggfortify.

```
autoplot(scaled_ts1)
```

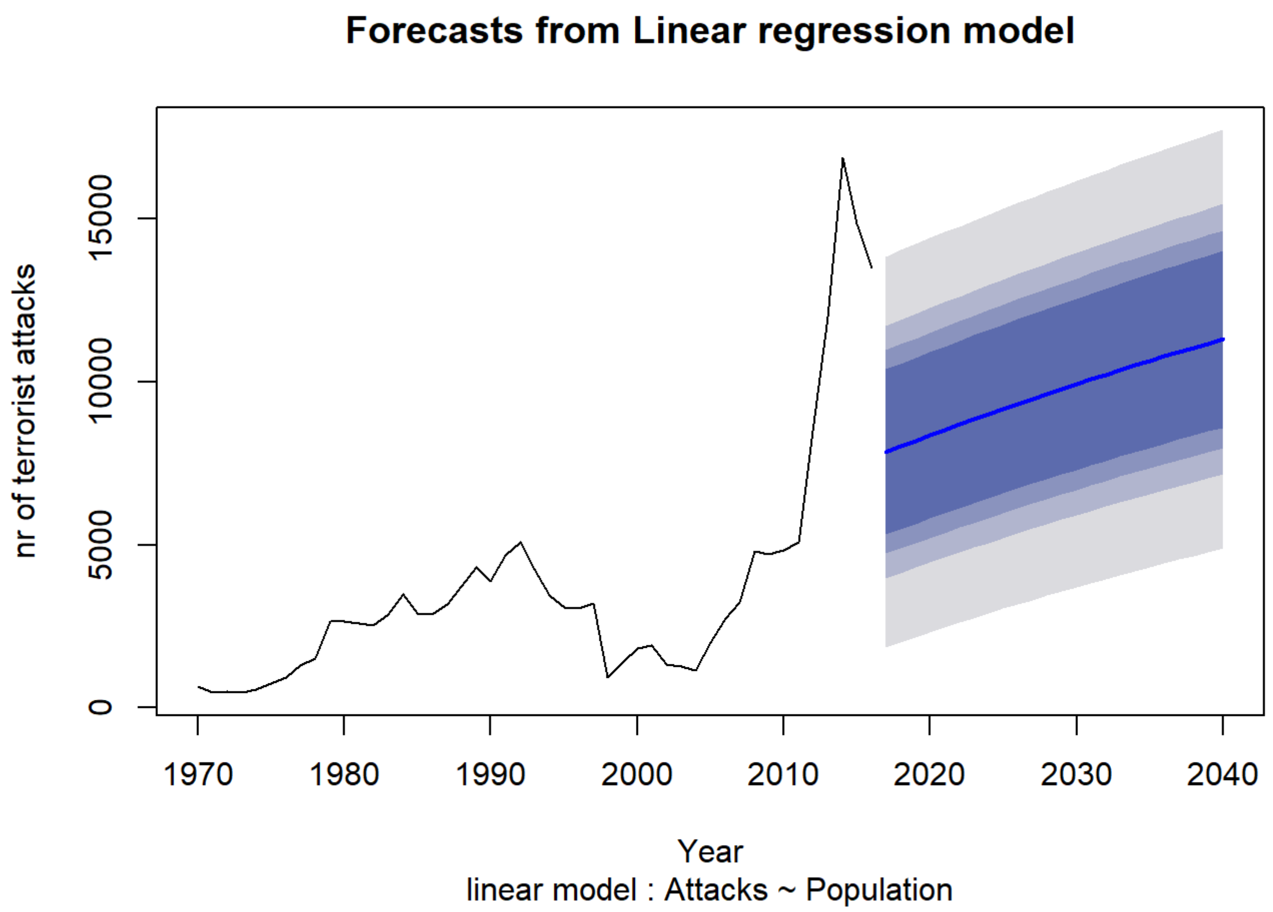


```
autoplot(percent_changes)
```

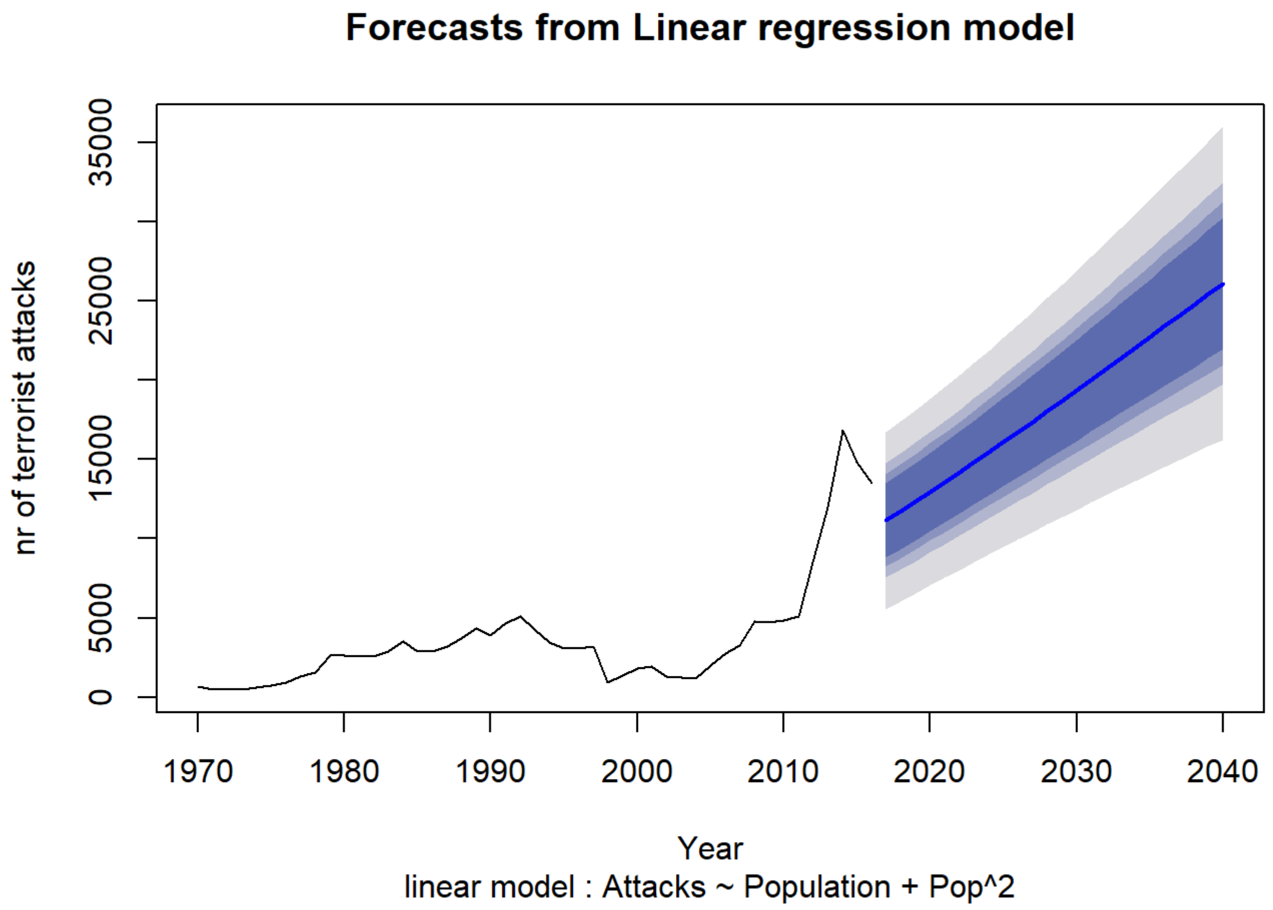


Let’s now use the forecast package to forecast this time serie. First we need to build a time series linear model.

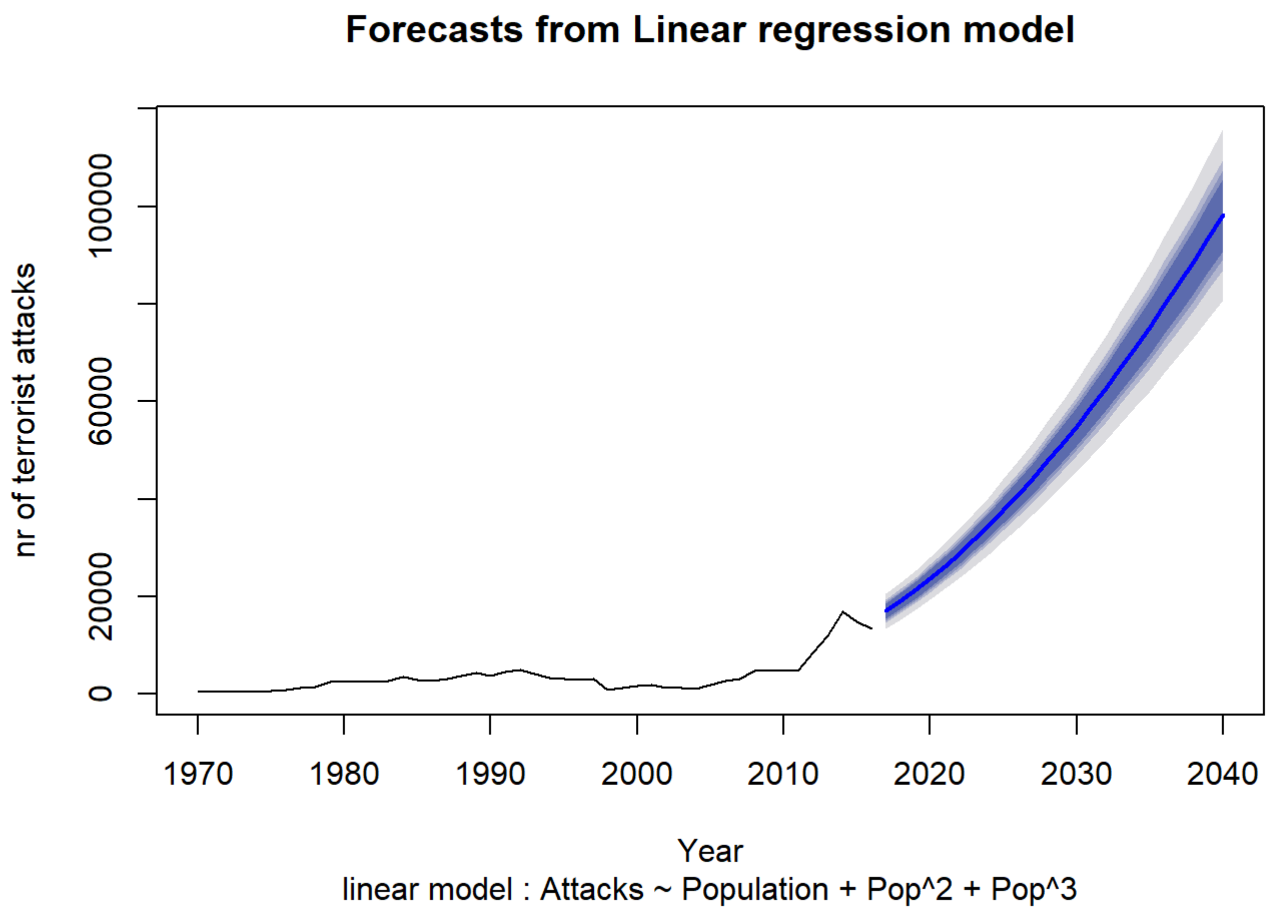
```
tm1 <- tslm(terrorist_attacks_count ~ PopTotal, data=ts1)

tm2 <- tslm(terrorist_attacks_count ~ PopTotal + I(PopTotal^2), data=ts1)

tm3 <- tslm(terrorist_attacks_count ~ PopTotal + I(PopTotal^2) + I(PopTotal^3), data=ts1)

futurepopworld4 <- futurepopworld %>%
  filter(Time < 2041)

f1 <- forecast(tm1, newdata=futurepopworld4, level=c(60, 70, 80, 95))
f2 <- forecast(tm2, newdata=futurepopworld4, level=c(60, 70, 80, 95))
f3 <- forecast(tm3, newdata=futurepopworld4, level=c(60, 70, 80, 95))

plot(f1, ylab = 'nr of terrorist attacks', xlab = 'Year', sub = 'linear model : Attacks ~ Pop
ulation')
```



```
plot(f2, ylab = 'nr of terrorist attacks', xlab = 'Year', sub = 'linear model : Attacks ~ Pop
ulation + Pop^2')
```



```
plot(f3, ylab = 'nr of terrorist attacks', xlab = 'Year', sub = 'linear model : Attacks ~ Pop
ulation + Pop^2 + Pop^3')
```



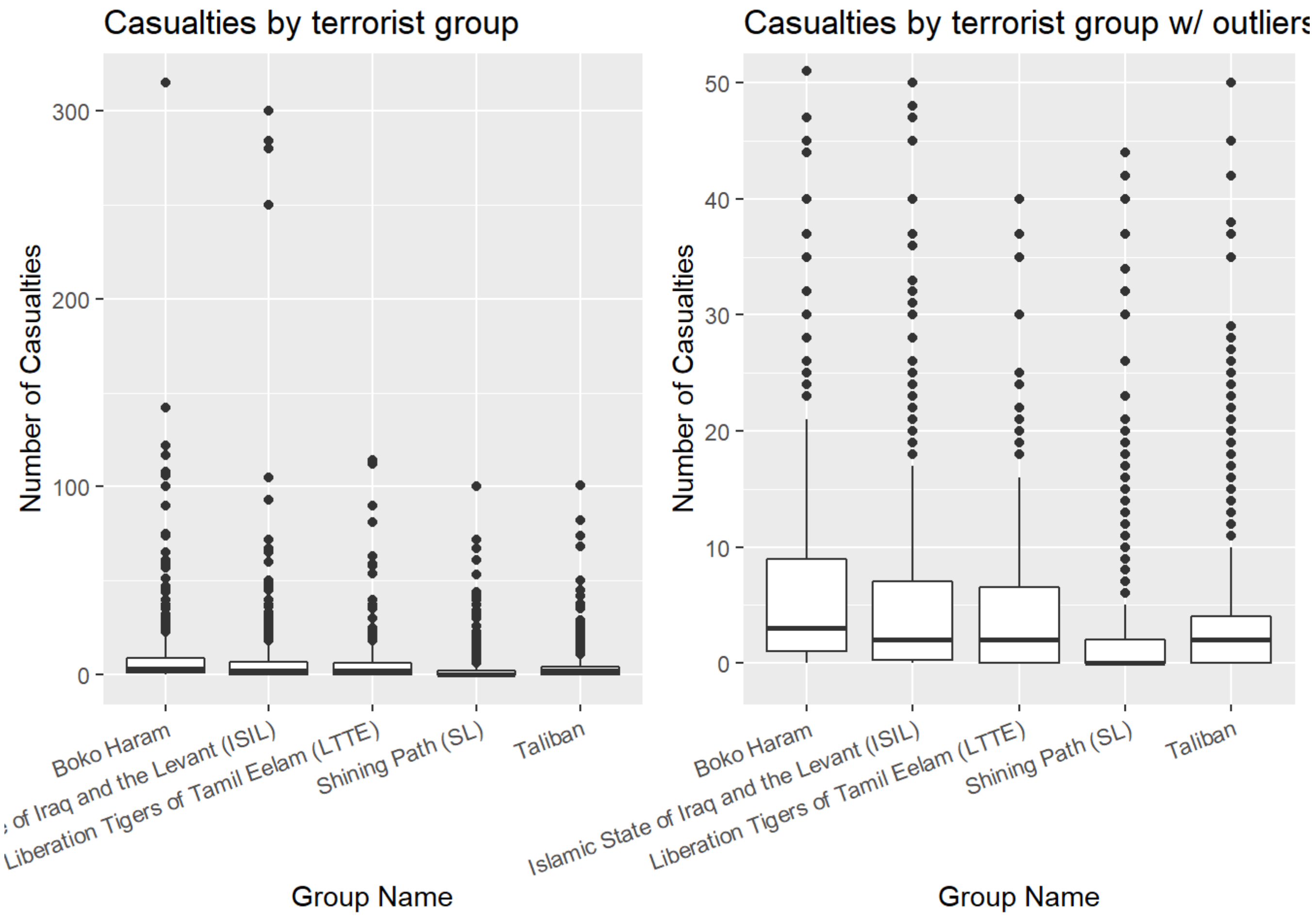
These three forecasts show the difficulty and issues that arise when forecasting outside of the fitted model. The 4 different colored confidence intervals show how unsure the forecast is. From the outside in the confidence intervals for the colored bands are: 95%, 80%, 70% and 60%.

4. Is there a difference in average casulaties per terrorist group?

4.1 Data Preparation

```
top5_groups <- df %>%
 group_by(group_name) %>%
 filter(group_name != "Unknown") %>%
 summarise(nr_of_attacks = n(), avgkilled = sum(nkill, na.rm = TRUE)/n(), nkill =   sum(nkill, na.rm=T)) %>%
 top_n(n=5, wt=nkill) %>%
 arrange(desc(nkill))

dftop5 <- df%>%
  filter(group_name == c("Islamic State of Iraq and the Levant (ISIL)", "Taliban", "Boko Haram", "Shining Path (SL)", "Liberation Tigers of Tamil Eelam (LTTE)"))

b1 <- ggplot(data=dftop5) +
  geom_boxplot(aes(x=group_name, y=nkill)) +
  theme(axis.text.x= element_text(angle=20, hjust=1)) +
  labs(title='Casualties by terrorist group') +
  labs(x='Group Name', y='Number of Casualties')

b2 <- ggplot(data=dftop5) +
  geom_boxplot(aes(x=group_name, y=nkill)) +
  theme(axis.text.x= element_text(angle=20, hjust=1)) +
  labs(title='Casualties by terrorist group w/ outliers') +
  labs(x='Group Name', y='Number of Casualties') +
  coord_cartesian(ylim = c(-1, 50))

grid.arrange(b1, b2, nrow=1)
```

```
## Warning: Removed 247 rows containing non-finite values (stat_boxplot).

## Warning: Removed 247 rows containing non-finite values (stat_boxplot).
```



There are a few big outliers that make the visualization somewhat useless. Let's see if the Anova can give us better information.

4.2 ANOVA

$$\begin{cases} H_0 : \mu_{ISIL} = \mu_{Taliban} = \mu_{BokoHaram} = \mu_{SL} = \mu_{LTTE} \\ H_a : \qquad \text{at least two means are different} \end{cases}$$

4.2.1 Model

```
mm <- aov(data=dftop5, nkill ~ group_name)
summary(mm)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)    
## group_name      4  23142    5785    26.4 <2e-16 ***
## Residuals    3597 788172     219                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 247 observations deleted due to missingness
```

ANOVA gives p<0.05, meaning we REJECT the null-hypothesis, atleast two means are different. Let's do a post-hoc test to see which ones are different.

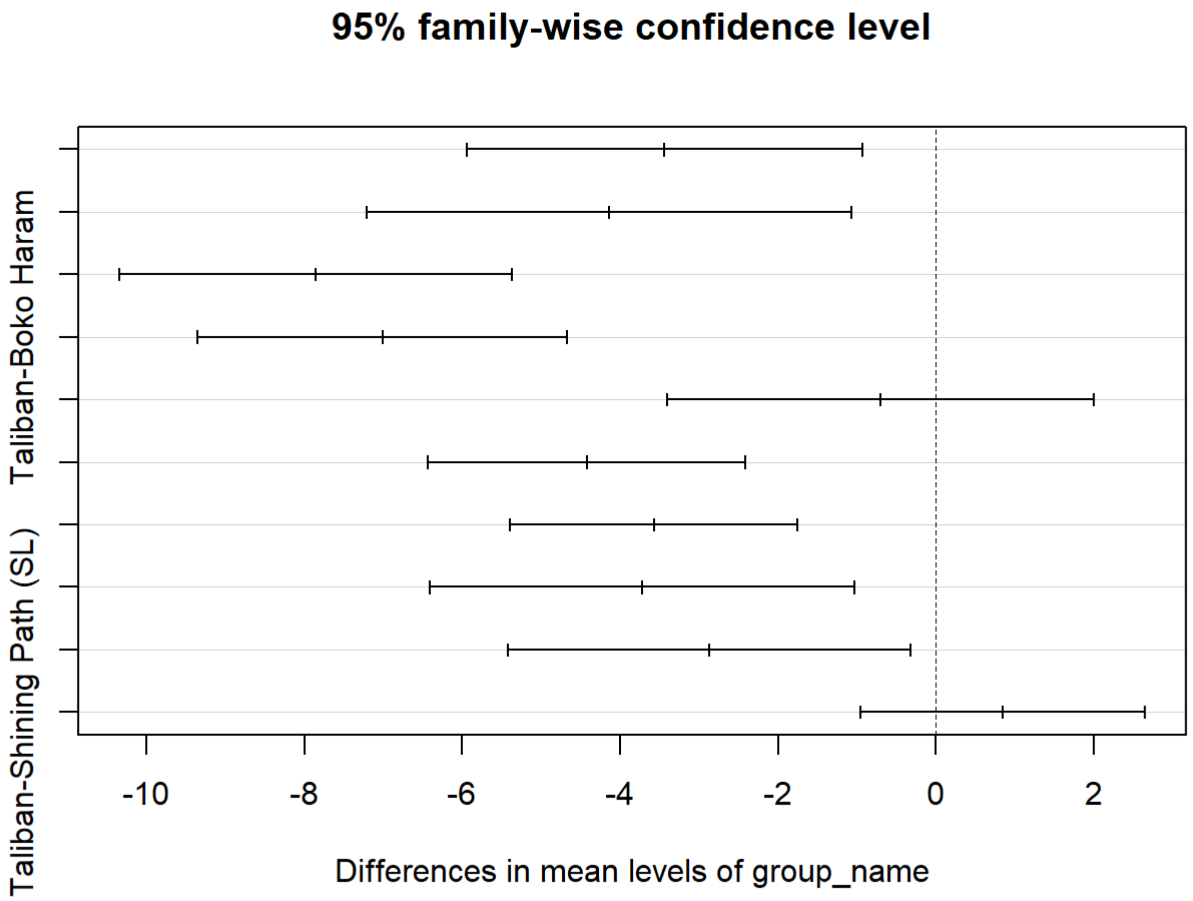
4.2.2 Post-Hoc test

```
TukeyHSD(mm)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = nkill ~ group_name, data = dftop5)
##
## $group_name
##
diff
## Islamic State of Iraq and the Levant (ISIL)-Boko Haram                              -3.435
4823
## Liberation Tigers of Tamil Eelam (LTTE)-Boko Haram                                  -4.141
0055
## Shining Path (SL)-Boko Haram                                                        -7.857
7774
## Taliban-Boko Haram                                                                  -7.010
4519
## Liberation Tigers of Tamil Eelam (LTTE)-Islamic State of Iraq and the Levant (ISIL) -0.705
5232
## Shining Path (SL)-Islamic State of Iraq and the Levant (ISIL)                       -4.422
2951
## Taliban-Islamic State of Iraq and the Levant (ISIL)                                 -3.574
9696
## Shining Path (SL)-Liberation Tigers of Tamil Eelam (LTTE)                           -3.716
7719
## Taliban-Liberation Tigers of Tamil Eelam (LTTE)                                     -2.869
4464
## Taliban-Shining Path (SL)                                                            0.847
3255
##
lwr
## Islamic State of Iraq and the Levant (ISIL)-Boko Haram                               -5.94
21023
## Liberation Tigers of Tamil Eelam (LTTE)-Boko Haram                                   -7.21
75368
## Shining Path (SL)-Boko Haram                                                        -10.34
83033
## Taliban-Boko Haram                                                                   -9.35
14870
## Liberation Tigers of Tamil Eelam (LTTE)-Islamic State of Iraq and the Levant (ISIL)  -3.40
99126
## Shining Path (SL)-Islamic State of Iraq and the Levant (ISIL)                        -6.43
51032
## Taliban-Islamic State of Iraq and the Levant (ISIL)                                  -5.39
95579
## Shining Path (SL)-Liberation Tigers of Tamil Eelam (LTTE)                            -6.40
62509
## Taliban-Liberation Tigers of Tamil Eelam (LTTE)                                      -5.42
11171
## Taliban-Shining Path (SL)                                                            -0.95
50889
##
upr
## Islamic State of Iraq and the Levant (ISIL)-Boko Haram                              -0.928
8622
## Liberation Tigers of Tamil Eelam (LTTE)-Boko Haram                                  -1.064
```

```
4742
## Shining Path (SL)-Boko Haram                                               -5.367
2515
## Taliban-Boko Haram                                                         -4.669
4169
## Liberation Tigers of Tamil Eelam (LTTE)-Islamic State of Iraq and the Levant (ISIL)  1.998
8662
## Shining Path (SL)-Islamic State of Iraq and the Levant (ISIL)              -2.409
4870
## Taliban-Islamic State of Iraq and the Levant (ISIL)                        -1.750
3814
## Shining Path (SL)-Liberation Tigers of Tamil Eelam (LTTE)                  -1.027
2929
## Taliban-Liberation Tigers of Tamil Eelam (LTTE)                            -0.317
7758
## Taliban-Shining Path (SL)                                                   2.649
7398
##                                                                               p
adj
## Islamic State of Iraq and the Levant (ISIL)-Boko Haram                     0.0017
459
## Liberation Tigers of Tamil Eelam (LTTE)-Boko Haram                         0.0022
577
## Shining Path (SL)-Boko Haram                                               0.0000
000
## Taliban-Boko Haram                                                         0.0000
000
## Liberation Tigers of Tamil Eelam (LTTE)-Islamic State of Iraq and the Levant (ISIL) 0.9538
279
## Shining Path (SL)-Islamic State of Iraq and the Levant (ISIL)              0.0000
000
## Taliban-Islamic State of Iraq and the Levant (ISIL)                        0.0000
009
## Shining Path (SL)-Liberation Tigers of Tamil Eelam (LTTE)                  0.0015
471
## Taliban-Liberation Tigers of Tamil Eelam (LTTE)                            0.0184
032
## Taliban-Shining Path (SL)                                                  0.7017
643
```

```
plot(TukeyHSD(mm))
```



The pairs of groups with non-significantly different means in number of kills are: * Taliban - Shining Path, * LTTE - ISIL

5. Is there a difference in average casualties per weapon type?

Let's see if the different weapons have a statistically different mortality rate. For the sake of interesting analysis we'll do the ANOVA on just the top 5 weapons.

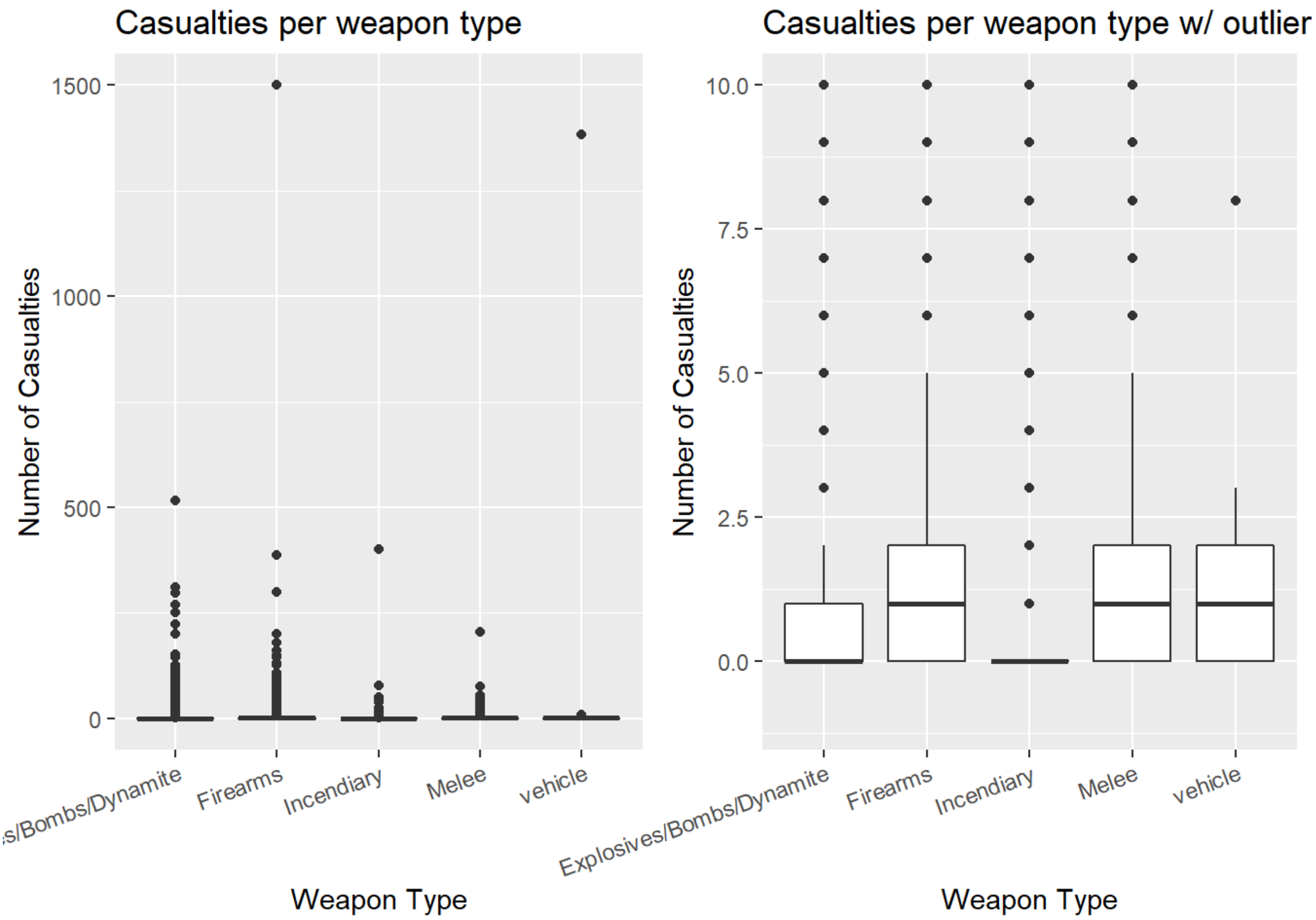
5.1 Data Preperation

```
#table
df %>%
  select(weapon_type, nkill, nwound) %>%
  group_by(weapon_type) %>%
  filter(weapon_type != "Unknown") %>%
  summarise(total_kills = sum(nkill, na.rm=T),
            total_wounded = sum(nwound, na.rm=T),
            average_kills = mean(nkill, na.rm=T),
            average_wounded = mean(nwound, na.rm = T)) %>%
  arrange(desc(total_kills)) %>%
  head(n=5)
```

```
## # A tibble: 5 x 5
##   weapon_type      total_kills total_wounded average_kills average_wounded
##   <chr>                  <int>         <int>         <dbl>           <dbl>
## 1 Firearms              166224         72159         3.18             1.47
## 2 Explosives/Bomb~      160534        367817         1.94             4.54
## 3 Melee                  10143          5183         3.10             1.65
## 4 Incendiary              5282          5420         0.536            0.560
## 5 Vehicle (not to~        3124         15217        26.9            133.
```

```
#renaming an unwieldy name
df$weapon_type <- as.factor(df$weapon_type)
levels(df$weapon_type)[levels(df$weapon_type) == "Vehicle (not to include vehicle-borne explo
sives, i.e., car or truck bombs)"] <- "vehicle"

#visualizing
df6 <- df %>%
  filter(weapon_type == c("vehicle", "Firearms", "Melee", "Incendiary", "Explosives/Bombs/Dyn
amite"))

b3 <- ggplot(data=df6) +
  geom_boxplot(aes(x=weapon_type, y=nkill)) +
  labs(title='Casualties per weapon type') +
  labs(x='Weapon Type', y='Number of Casualties') +
  theme(axis.text.x = element_text(angle=20, hjust=1))

b4 <- ggplot(data=df6) +
  geom_boxplot(aes(x=weapon_type, y=nkill)) +
  theme(axis.text.x= element_text(angle=20, hjust=1)) +
  labs(title='Casualties per weapon type w/ outliers') +
  labs(x='Weapon Type', y='Number of Casualties') +
  coord_cartesian(ylim = c(-1, 10))

grid.arrange(b3, b4, nrow=1)
```

```
## Warning: Removed 1511 rows containing non-finite values (stat_boxplot).

## Warning: Removed 1511 rows containing non-finite values (stat_boxplot).
```



There are a few big outliers that make the visualization somewhat useless. Let’s see if the Anova can give us better information.

5.2 ANOVA

$$\begin{cases} H_0 : \mu_{Firearms} = \mu_{Bombs} = \mu_{Melee} = \mu_{Incendiary} = \mu_{vehicle} \\ H_a : \quad \text{at least two means are different} \end{cases}$$

5.2.1 Model

```
a1 <- aov(data=df6, nkill~weapon_type)
summary(a1)
```

```
##                Df  Sum Sq Mean Sq F value Pr(>F)    
## weapon_type     4  114477   28619   129.9 <2e-16 ***
## Residuals   29751 6554198     220                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 1511 observations deleted due to missingness
```

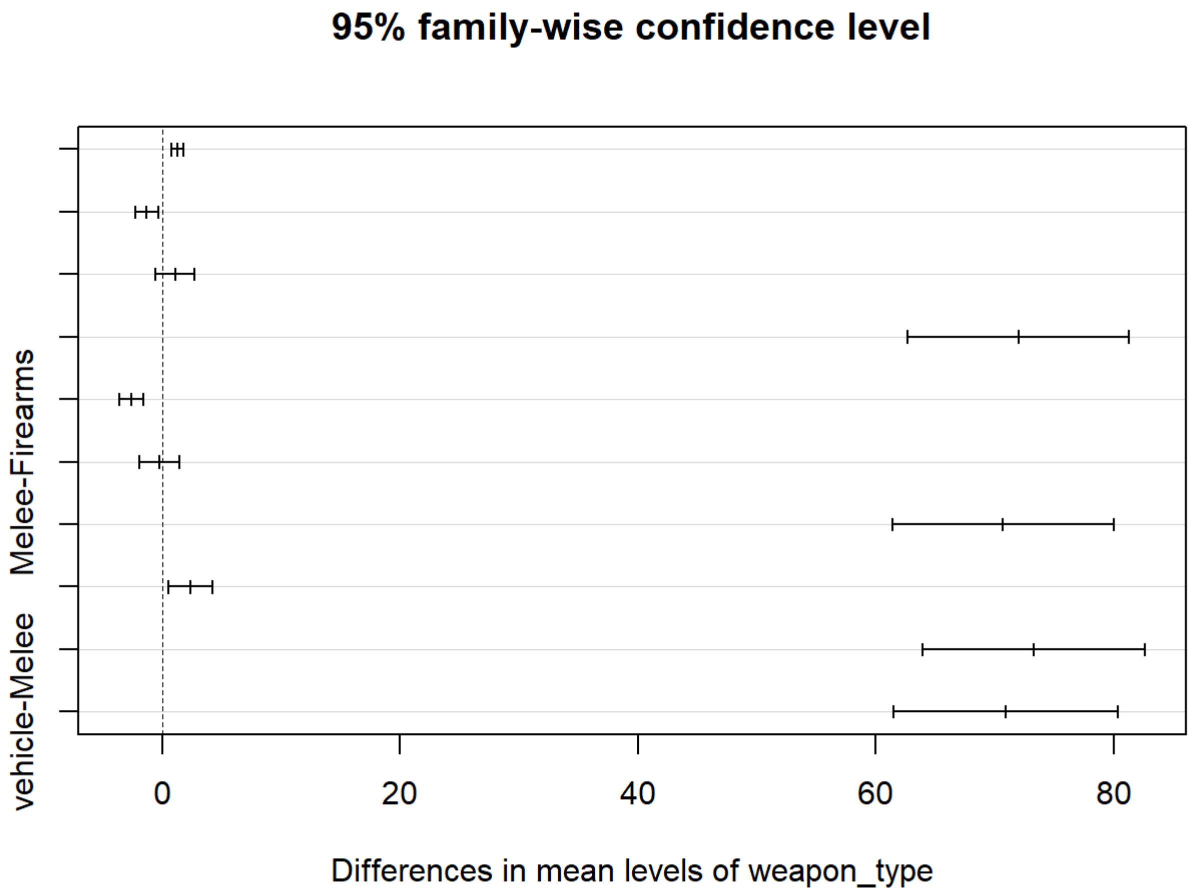
The ANOVA gives p<0.05 meaning we REJECT the null-hypothesis, atleast two means are different. Let’s do a post-hoc test to see which ones are different.

5.2.2 Post-Hoc test

```
TukeyHSD(a1)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = nkill ~ weapon_type, data = df6)
##
## $weapon_type
##                                            diff        lwr        upr
## Firearms-Explosives/Bombs/Dynamite    1.3067410  0.8029701  1.8105119
## Incendiary-Explosives/Bombs/Dynamite -1.2838559 -2.2546273 -0.3130844
## Melee-Explosives/Bombs/Dynamite       1.0958995 -0.5589053  2.7507044
## vehicle-Explosives/Bombs/Dynamite    71.9949588 62.7012501 81.2886674
## Incendiary-Firearms                  -2.5905969 -3.5899402 -1.5912536
## Melee-Firearms                       -0.2108415 -1.8825678  1.4608849
## vehicle-Firearms                     70.6882178 61.3914812 79.9849543
## Melee-Incendiary                      2.3797554  0.5134004  4.2461104
## vehicle-Incendiary                   73.2788146 63.9451164 82.6125128
## vehicle-Melee                        70.8990592 61.4696423 80.3284762
##                                          p adj
## Firearms-Explosives/Bombs/Dynamite   0.0000000
## Incendiary-Explosives/Bombs/Dynamite 0.0028573
## Melee-Explosives/Bombs/Dynamite      0.3697015
## vehicle-Explosives/Bombs/Dynamite    0.0000000
## Incendiary-Firearms                  0.0000000
## Melee-Firearms                       0.9969960
## vehicle-Firearms                     0.0000000
## Melee-Incendiary                     0.0045954
## vehicle-Incendiary                   0.0000000
## vehicle-Melee                        0.0000000
```

```
plot(TukeyHSD(a1))
```



Tukey's post-hoc test gives us some additional in-depth information. It seems that a few are not different while most of them are for p<0.05.

All of the weapon_type comparisson's are one-on-one significantly different EXCEPT for: * Melee-Explosives/Bombs/Dynamite * Melee-Firearms

6. What are some common links in weapon type, target type and group?

In terrorism every group has their type of target, weapon and style that make them famous. Their signature style. Or atleast, thats the idea. Is that actually true? Let's try to uncover some commmon links through applying an Apriori Association algorithm to the dataset with as consequents of interest the group names of the organisations to see how often they 'appear together'. We'll use the library arules for this.

6.1 Data preparation for first association analysis

```
#get relevant variables in new dataset
trules <- df %>%
  select(country, region, attacktype, target_type, group_name, target_sub_type, weapon_type,
nkill, nwound)

#change nkill and nwound to factors
trules <- trules %>%
  mutate(nkill = ifelse(nkill==0, 0,
                      ifelse(nkill<2, 1,
                             ifelse(nkill <6, 2,
                                    ifelse(nkill < 16, 3, 4))))) %>%
  mutate(nwound = ifelse(nwound==0, 0,
                      ifelse(nwound<2, 1,
                             ifelse(nwound <6, 2,
                                    ifelse(nwound < 16, 3, 4)))))

#change everything to factors
trules$country <- as.factor(trules$country)
trules$region <- as.factor(trules$region)
trules$attacktype <- as.factor(trules$attacktype)
trules$target_type <- as.factor(trules$target_type)
trules$group_name <- as.factor(trules$group_name)
trules$target_sub_type <- as.factor(trules$target_sub_type)
trules$weapon_type <- as.factor(trules$weapon_type)
trules$nkill <- as.factor(trules$nkill)
trules$nwound <- as.factor(trules$nwound)


terror_rules <- apriori(trules, parameter=list(support =0.01, confidence =0.5, minlen=2, maxl
en=5))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5    0.01      2
##  maxlen target   ext
##       5  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 1703
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[3834 item(s), 170350 transaction(s)] done [0.43s].
## sorting and recoding items ... [115 item(s)] done [0.02s].
## creating transaction tree ... done [0.15s].
## checking subsets of size 1 2 3 4 5
## Warning in apriori(trules, parameter = list(support = 0.01, confidence =
## 0.5, : Mining stopped (maxlen reached). Only patterns up to a length of 5
## returned!
```

```
##  done [0.09s].
## writing ... [4544 rule(s)] done [0.00s].
## creating S4 object  ... done [0.03s].
```

```
inspect(head(sort(terror_rules, by="lift"),3))
```

```
##     lhs                                              rhs                                      
support confidence     lift count
## [1] {target_sub_type=Non-State Militia}           => {target_type=Terrorists/Non-State Militi
a} 0.01100088  1.0000000 59.58377  1874
## [2] {target_type=Terrorists/Non-State Militia} => {target_sub_type=Non-State Militia}        
0.01100088  0.6554739 59.58377  1874
## [3] {region=North America,                                                                  
##      nkill=0,                                                                               
##      nwound=0}                                   => {country=United States}                  
0.01271500  0.9006237 55.62772  2166
```

too much correlation between factors as evidenced by the presence of rules with confidence of 1.00. Let’s cut correlates from the set like target_sub_type (correlated with target_type) and locations (country and region, because both are correlated too much with group_names). Let’s also get rid of ‘Unknowns’

```
trules <- trules %>% filter(group_name != 'Unknown')
trules <- trules %>% filter(attacktype != 'Unknown')
trules <- trules %>% filter(target_type != 'Unknown')
trules <- trules %>% filter(nkill != 'Unknown')
trules <- trules %>% filter(nwound != 'Unknown')
trules <- trules %>% filter(weapon_type != 'Unknown')

trules <- trules %>%
  select(-target_sub_type, -region, -country)
```

Let’s also specify that we are only looking for consequents that are groupnames of the top10 most busy terrorist groups (see question 1.7), so that we can find what they are linked with which are the antecedents Classic example for this ofcourse is: If someone gets bread, and milk what else will they most likely buy? In this case: if an attack happens on a police station with a grenade, what group does this?

```
trules %>%
  group_by(group_name) %>%
  summarise(nr_of_attacks = n()) %>%
  arrange(desc(nr_of_attacks)) %>%
  head(n=10)
```

```
## # A tibble: 10 x 2
##    group_name                                       nr_of_attacks
##    <fct>                                                    <int>
##  1 Taliban                                                   5041
##  2 Shining Path (SL)                                         3737
##  3 Islamic State of Iraq and the Levant (ISIL)               3023
##  4 Farabundo Marti National Liberation Front (FMLN)          2478
##  5 New People's Army (NPA)                                   2022
##  6 Revolutionary Armed Forces of Colombia (FARC)             1894
##  7 Kurdistan Workers' Party (PKK)                            1882
##  8 Basque Fatherland and Freedom (ETA)                       1754
##  9 Al-Shabaab                                                1656
## 10 Irish Republican Army (IRA)                               1608
```

```
#We also exclude nwound because it's making the results too noisy.

trules <- trules %>%
  select( -nwound)
```

6.2 lets start with Taliban - Association rules

```
taliban_rules <- apriori(trules, parameter=list(support=0.01, confidence=0.1, minlen=1, maxlen=5), appearance = list(rhs='group_name=Taliban', default="lhs"))
```

```
## Apriori
## 
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.1    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target   ext
##       5  rules FALSE
## 
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
## 
## Absolute minimum support count: 714 
## 
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[3033 item(s), 71421 transaction(s)] done [0.12s].
## sorting and recoding items ... [45 item(s)] done [0.00s].
## creating transaction tree ... done [0.03s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [12 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(head(sort(taliban_rules, by='lift'), 10))
```

```
##      lhs                                         rhs                     support confidence
lift count
## [1]  {target_type=Police,
##       nkill=2}                                => {group_name=Taliban} 0.01003906  0.2501745
3.544477   717
## [2]  {attacktype=Armed Assault,
##       target_type=Police,
##       weapon_type=Firearms}                   => {group_name=Taliban} 0.01085115  0.1858068
2.632514   775
## [3]  {weapon_type=Explosives/Bombs/Dynamite,
##       nkill=2}                                => {group_name=Taliban} 0.01362344  0.1843851
2.612372   973
## [4]  {attacktype=Bombing/Explosion,
##       weapon_type=Explosives/Bombs/Dynamite,
##       nkill=2}                                => {group_name=Taliban} 0.01229330  0.1839514
2.606227   878
## [5]  {attacktype=Bombing/Explosion,
##       nkill=2}                                => {group_name=Taliban} 0.01229330  0.1796603
2.545431   878
## [6]  {attacktype=Armed Assault,
##       target_type=Police}                     => {group_name=Taliban} 0.01131320  0.1756140
2.488103   808
## [7]  {target_type=Police,
##       weapon_type=Explosives/Bombs/Dynamite} => {group_name=Taliban} 0.01009507  0.1555220
2.203439   721
## [8]  {target_type=Police}                     => {group_name=Taliban} 0.02314445  0.1515402
2.147024  1653
## [9]  {target_type=Police,
##       weapon_type=Firearms}                   => {group_name=Taliban} 0.01218129  0.1510154
2.139590   870
## [10] {nkill=2}                                => {group_name=Taliban} 0.02496465  0.1383780
1.960542  1783
```

This yields some incredibly interesting results; The Taliban tends to attack the Police extremely frequently with various types of weapons, firearms and explosives most commonly. They also seem to kill a small amount per attack, less than 6 but more than 1.

Lets see if we can get rules for each terror group in the top10. If the group doesnt show any interesting rules we have excluded them.

6.3 Islamic State of Iraq and the Levant (ISIL) - Association rules

```
isil_rules <- apriori(trules, parameter=list(support=0.01, confidence=0.1, minlen=1, maxlen=
5), appearance = list(rhs='group_name=Islamic State of Iraq and the Levant (ISIL)', default
="lhs"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.1    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target   ext
##       5  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 714
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[3033 item(s), 71421 transaction(s)] done [0.12s].
## sorting and recoding items ... [45 item(s)] done [0.00s].
## creating transaction tree ... done [0.03s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [6 rule(s)] done [0.00s].
## creating S4 object  ... done [0.01s].
```

```
inspect(head(sort(isil_rules, by='lift'), 10))
```

```
##     lhs                                             rhs
support confidence     lift count
## [1] {attacktype=Bombing/Explosion,
##      target_type=Private Citizens & Property,
##      weapon_type=Explosives/Bombs/Dynamite}   => {group_name=Islamic State of Iraq and the
Levant (ISIL)} 0.01520561  0.1773061 4.189011  1086
## [2] {attacktype=Bombing/Explosion,
##      target_type=Private Citizens & Property} => {group_name=Islamic State of Iraq and the
Levant (ISIL)} 0.01534563  0.1760360 4.159003  1096
## [3] {target_type=Private Citizens & Property,
##      weapon_type=Explosives/Bombs/Dynamite}   => {group_name=Islamic State of Iraq and the
Levant (ISIL)} 0.01545764  0.1729052 4.085037  1104
## [4] {attacktype=Bombing/Explosion,
##      weapon_type=Explosives/Bombs/Dynamite,
##      nkill=2}                                 => {group_name=Islamic State of Iraq and the
Levant (ISIL)} 0.01069713  0.1600670 3.781723   764
## [5] {attacktype=Bombing/Explosion,
##      nkill=2}                                 => {group_name=Islamic State of Iraq and the
Levant (ISIL)} 0.01073914  0.1569470 3.708009   767
## [6] {weapon_type=Explosives/Bombs/Dynamite,
##      nkill=2}                                 => {group_name=Islamic State of Iraq and the
Levant (ISIL)} 0.01099117  0.1487588 3.514555   785
```

ISIL uses bombs and explosives almost exclusively on citizens. They ofcourse are famous for their terror campaigns targetting innocent citizens. ISIL also kills a small amount per attack, similar to Taliban.

6.4 Farabundo Marti National Liberation Front (FMLN) - Association Rules

```
FMLN_rules <- apriori(trules, parameter=list(support=0.01, confidence=0.1, minlen=1, maxlen=
5), appearance = list(rhs='group_name=Farabundo Marti National Liberation Front (FMLN)', defa
ult="lhs"))
```

```
## Apriori
## 
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.1    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target   ext
##       5  rules FALSE
## 
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
## 
## Absolute minimum support count: 714 
## 
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[3033 item(s), 71421 transaction(s)] done [0.12s].
## sorting and recoding items ... [45 item(s)] done [0.00s].
## creating transaction tree ... done [0.04s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [11 rule(s)] done [0.00s].
## creating S4 object  ... done [0.01s].
```

```
inspect(head(sort(FMLN_rules, by='lift'), 10))
```

```
##      lhs                                        rhs                                        
support confidence     lift count
## [1]  {attacktype=Bombing/Explosion,                                                                     
##       target_type=Utilities,                                                                           
##       nkill=0}                              => {group_name=Farabundo Marti National Liber
ation Front (FMLN)} 0.01023508  0.2378783 6.856136   731
## [2]  {attacktype=Bombing/Explosion,                                                                     
##       target_type=Utilities,                                                                           
##       weapon_type=Explosives/Bombs/Dynamite,                                                           
##       nkill=0}                              => {group_name=Farabundo Marti National Liber
ation Front (FMLN)} 0.01017908  0.2375041 6.845351   727
## [3]  {target_type=Utilities,                                                                           
##       weapon_type=Explosives/Bombs/Dynamite,                                                           
##       nkill=0}                              => {group_name=Farabundo Marti National Liber
ation Front (FMLN)} 0.01017908  0.2366536 6.820839   727
## [4]  {attacktype=Bombing/Explosion,                                                                     
##       target_type=Utilities}                => {group_name=Farabundo Marti National Liber
ation Front (FMLN)} 0.01029109  0.2329635 6.714483   735
## [5]  {attacktype=Bombing/Explosion,                                                                     
##       target_type=Utilities,                                                                           
##       weapon_type=Explosives/Bombs/Dynamite} => {group_name=Farabundo Marti National Liber
ation Front (FMLN)} 0.01020708  0.2323876 6.697884   729
## [6]  {target_type=Utilities,                                                                           
##       weapon_type=Explosives/Bombs/Dynamite} => {group_name=Farabundo Marti National Liber
ation Front (FMLN)} 0.01020708  0.2313551 6.668125   729
## [7]  {target_type=Utilities,                                                                           
##       nkill=0}                              => {group_name=Farabundo Marti National Liber
ation Front (FMLN)} 0.01033310  0.2309859 6.657484   738
## [8]  {target_type=Utilities}                => {group_name=Farabundo Marti National Liber
ation Front (FMLN)} 0.01045911  0.2250678 6.486912   747
## [9]  {attacktype=Armed Assault,                                                                        
##       target_type=Military,                                                                            
##       weapon_type=Firearms}                 => {group_name=Farabundo Marti National Liber
ation Front (FMLN)} 0.01106117  0.1426250 4.110743   790
## [10] {target_type=Military,                                                                            
##       weapon_type=Firearms}                 => {group_name=Farabundo Marti National Liber
ation Front (FMLN)} 0.01171924  0.1315004 3.790109   837
```

The Farabundo Marti National LIberation Front (FMLN) seems to favor non-citizen targets and mostly did sabotage with bombs and explosives. Utilities were a frequent target. Next to that the FMLN doesn't seem to kill people with their attacks. Not a single rule shows up where nkill > 0.

6.5 All

The other groups in the top10 don't seem to have any other interesting rules. But let's do all the groups together to get a nice overview.

```
terror_rules <- apriori(trules, parameter=list(support=0.01, confidence=0.1, minlen=1, maxlen=5), appearance = list(rhs=c('group_name=Taliban', 'group_name=Shining Path (SL)', 'group_name=Islamic State of Iraq and the Levant (ISIL)', 'group_name=Farabundo Marti National Liberation Front (FMLN)', 'group_name=New People\'s Army (NPA)', 'group_name=Revolutionary Armed Forces of Colombia (FARC)', 'group_name=Kurdistan Workers\' Party (PKK)', 'group_name=Basque Fatherland and Freedom (ETA)', 'group_name=Al-Shabaab', 'group_name=Irish Republican Army (IRA)'), default="lhs"))
```

```
## Apriori
## 
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.1    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target   ext
##       5  rules FALSE
## 
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
## 
## Absolute minimum support count: 714 
## 
## set item appearances ...[10 item(s)] done [0.00s].
## set transactions ...[3033 item(s), 71421 transaction(s)] done [0.11s].
## sorting and recoding items ... [45 item(s)] done [0.00s].
## creating transaction tree ... done [0.03s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [29 rule(s)] done [0.00s].
## creating S4 object  ... done [0.01s].
```