

Data-607 Project-3

Students Name : Sachid Deshmukh, Mohammad Younus, Robert Lauto, Santosh Manjrekar, Olga Shiligin, Hantz Angrand

Date : 10/18/2018

- GitHub Location for rmd file
 - GitHub Location for pdf file
 - RPub's location of published file
 - Data File
-

Context

Below example demonstrate what it takes from Data Scientist to analyze a given data and derive a meaningful information from the data. Data scientist frequently work with ambiguous data and are tasked to infer meaningful information from data. Let's take example of The Global Terrorism Database (GTD).

This is an open-source database including information on terrorist attacks around the world from 1970 through 2017

The GTD includes systematic data on domestic as well as international terrorist incidents that have occurred during this time period and now includes more than 180,000 attacks. The database is maintained by researchers at the National Consortium for the Study of Terrorism and Responses to Terrorism (START), headquartered at the University of Maryland. More Information

Let's see how a Data Scientist will approach this data set and what kind of insights can be obtained

1] Library Initialization

```
library(ggplot2) # Data visualization
library(readr) # CSV file I/O, e.g. the read_csv function
library(tidyverse)
```

```
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Conflicts with tidy packages -----
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
library(RCurl)
```

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

```
##
## Attaching package: 'RCurl'

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

2] Load Data

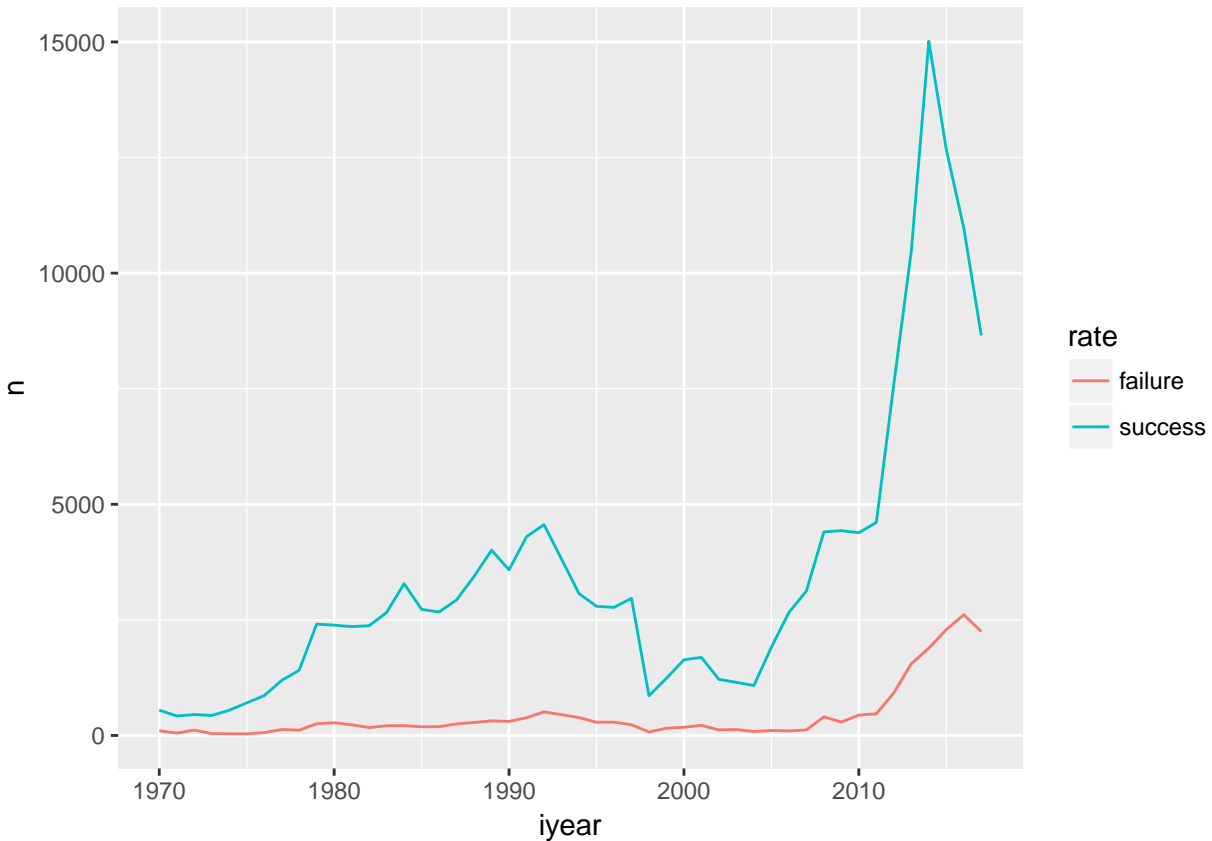
```
data.url = getURL("https://media.githubusercontent.com/media/mlforsachid/Data607-Project3/master/Misc/g")
attack <- read_csv(data.url,
  col_types = cols(
    nkill = col_double(),
    nhours = col_double(),
    propvalue = col_double(),
    nwound = col_double(),
    nperpcap= col_double(),
    nhostkid = col_double(),
    nreleased= col_double(),
    nkilllter= col_double(),
    nkillus = col_double(),
    nwoundus = col_double(),
    ransomamt = col_double(),
    ransompaid= col_double(),
    nwoundte= col_double()
  )
)
```

3] Analyze success and failure rate

```
attack %>%
  mutate(total = 1) %>% # total attacks that year (creates a new variable)
  count(iyear, wt=total-success) %>% # failed attempt
  cbind("failure") -> failure
colnames(failure)[3] <- "rate"

attack %>%
  count(iyear, wt = success) %>%
  cbind("success")-> success
colnames(success)[3] <- "rate"

rbind(failure,success) %>%
  ggplot(aes(iyear, n)) +
  geom_line(aes(group=rate, colour=rate))
```

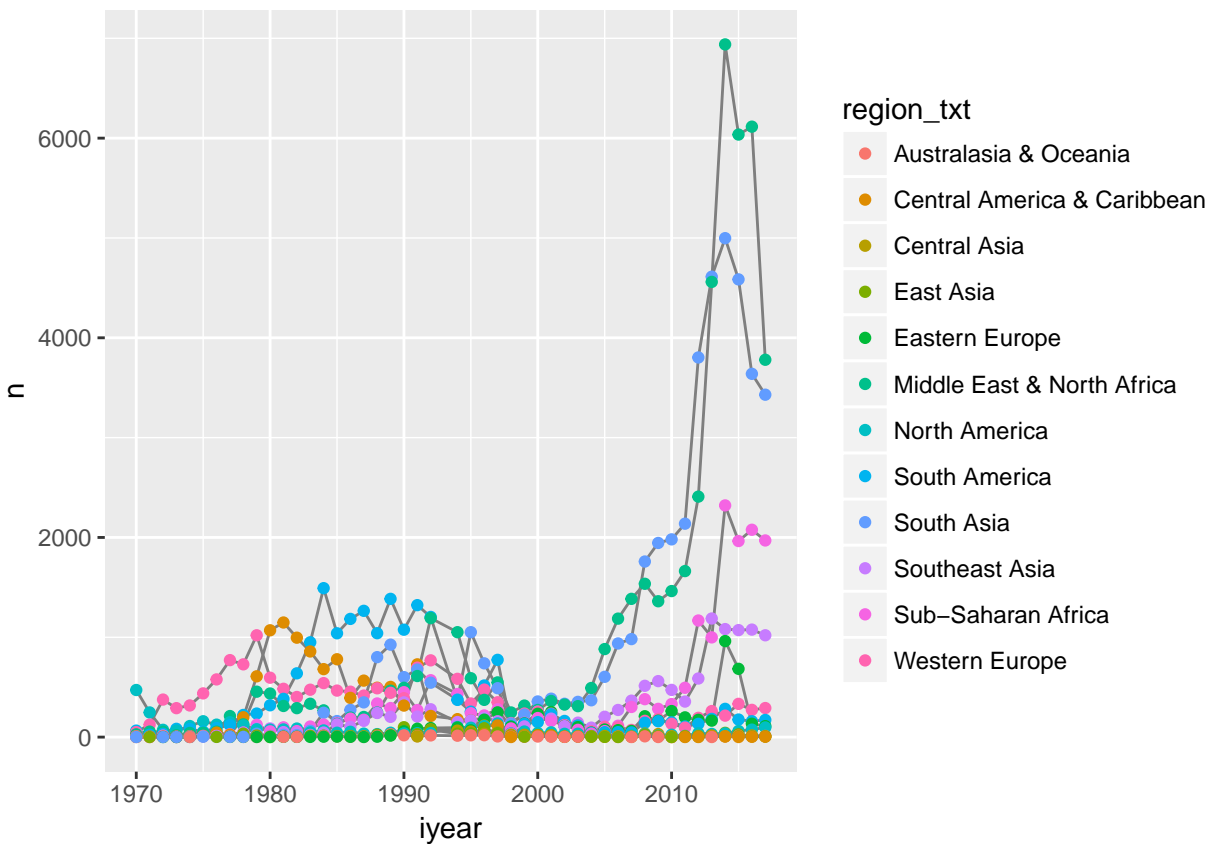


4] Success rate per region

```
attack %>%
  group_by(iyear, region_txt) %>% # allow operations to be performed by groups
  count() %>% # In case, count success rate conditional on the year and country
  arrange(desc(n)) %>% # arrange in descending order
  print(n = 10) %>% # print
  ggplot(aes(iyear, n)) + # x axis- year, y = success rate
  geom_line(aes(group = region_txt), colour = "grey50") +
  geom_point(aes(colour = region_txt))
```

```
## # A tibble: 529 x 3
## # Groups:   iyear, region_txt [529]
##   iyear region_txt      n
##   <int> <chr>          <int>
## 1  2014 Middle East & North Africa 6939
## 2  2016 Middle East & North Africa 6115
## 3  2015 Middle East & North Africa 6036
## 4  2014 South Asia                4998
## 5  2013 South Asia                4612
## 6  2015 South Asia                4585
## 7  2013 Middle East & North Africa 4560
## 8  2012 South Asia                3803
## 9  2017 Middle East & North Africa 3780
## 10 2016 South Asia                3639
```

```
## # ... with 519 more rows
```

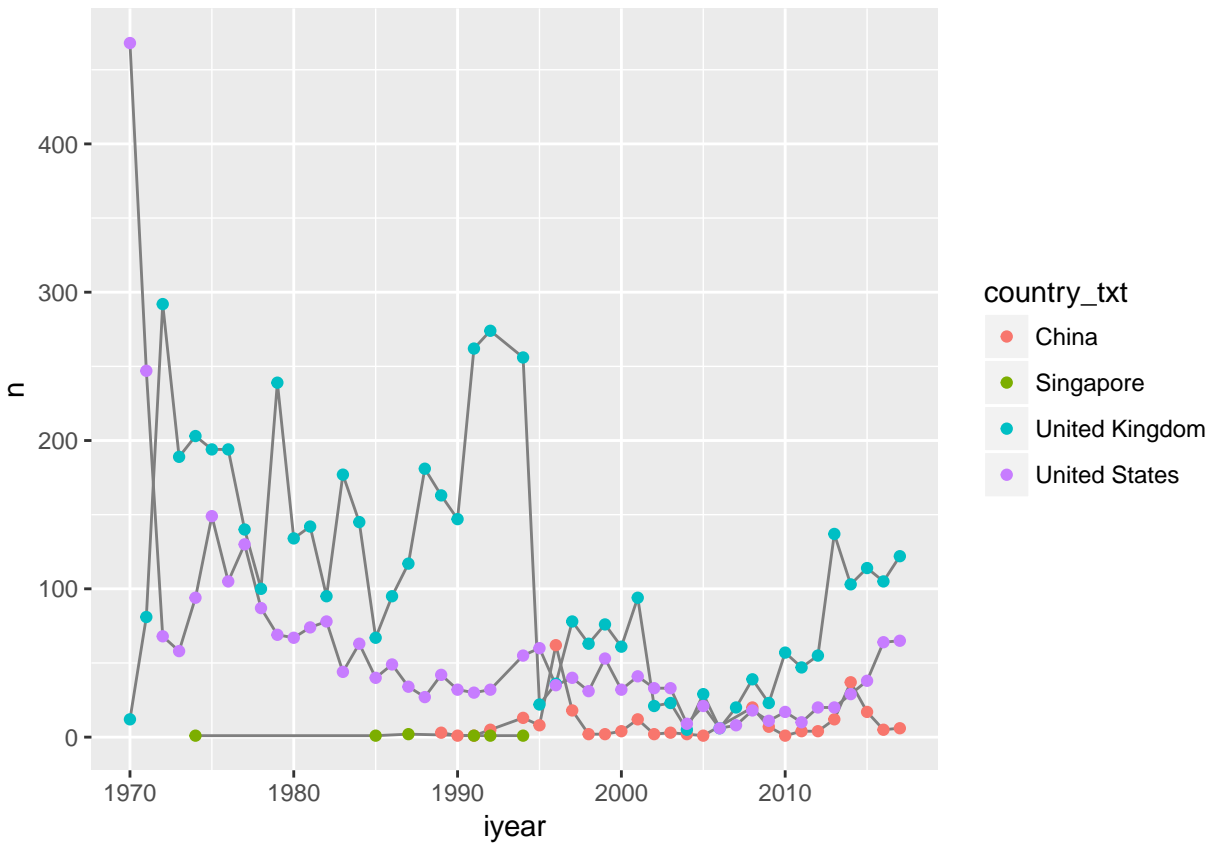


5] Analyze Success rate for selected countries

```
attack %>%
  # With multiple conditions for filter, make sure to use %in% and not ==
  filter(parse_character(country_txt) %in% c("Singapore", "China", "United States", "United Kingdom")) %>%
  group_by(iyear, country_txt) %>% # allow operations to be performed by groups
  count() %>% # In case, count success rate conditional on the year and country
  print(n = 10) %>%
  ggplot(aes(iyear, n)) + # x axis- year, y = success rate
  geom_line(aes(group = country_txt), colour = "grey50") +
  geom_point(aes(colour = country_txt))
```

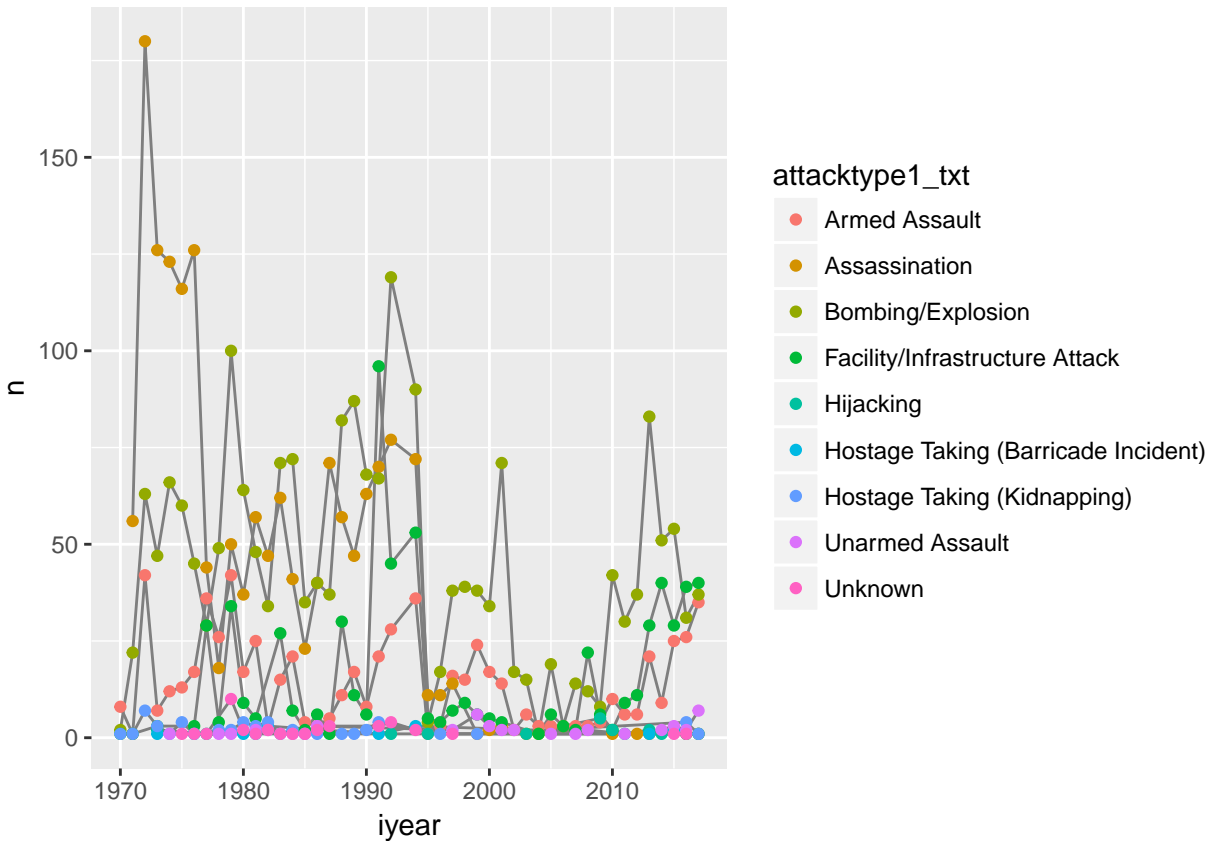
```
## # A tibble: 126 x 3
## # Groups:   iyear, country_txt [126]
##   iyear country_txt      n
##   <int> <chr>         <int>
## 1 1970 United Kingdom     12
## 2 1970 United States    468
## 3 1971 United Kingdom     81
## 4 1971 United States   247
## 5 1972 United Kingdom   292
## 6 1972 United States     68
## 7 1973 United Kingdom   189
```

```
## 8 1973 United States 58
## 9 1974 Singapore 1
## 10 1974 United Kingdom 203
## # ... with 116 more rows
```



6] Attack Type

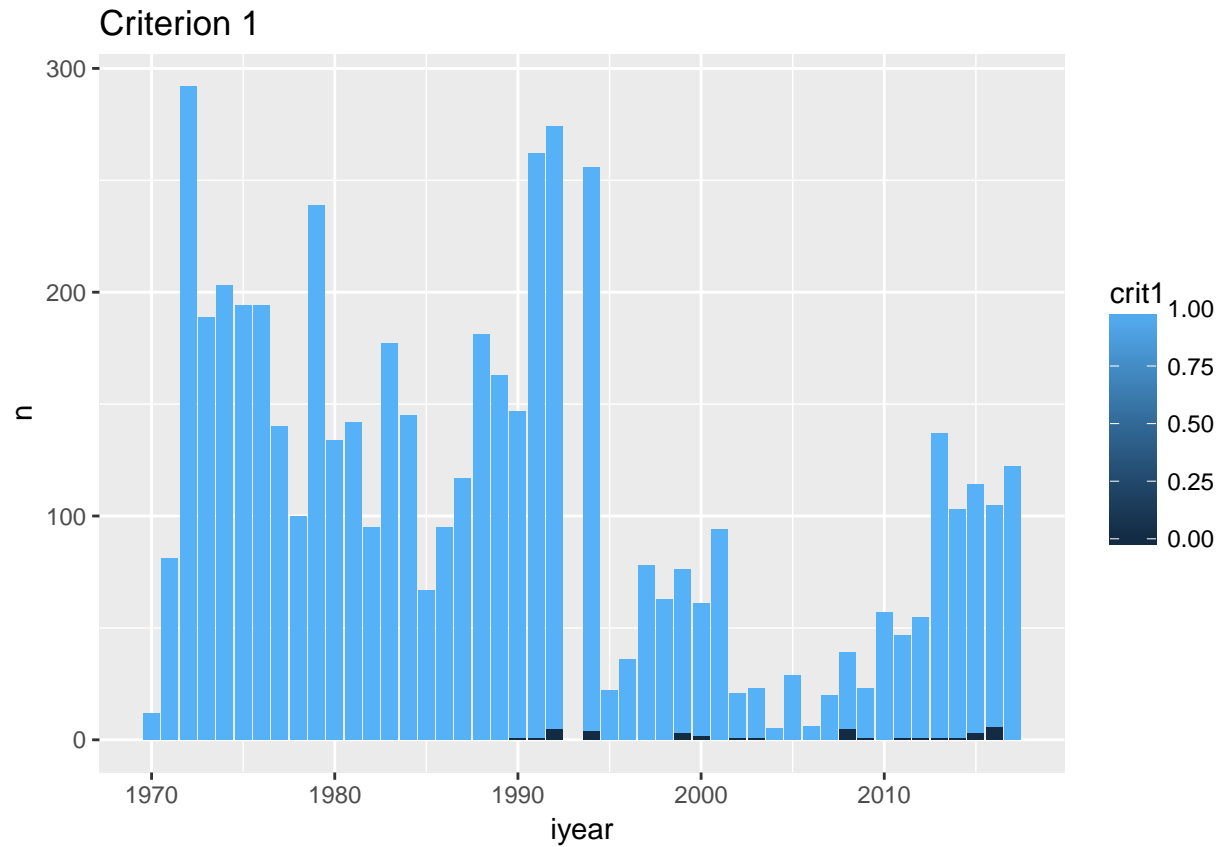
```
attack %>%
  # With multiple conditions for filter, make sure to use %in% and not ==
  filter(parse_character(country_txt) %in% c("United Kingdom")) %>% # select four countries
  group_by(iyear, attacktype1_txt) %>% # allow operations to be performed by groups
  count() %>% # In case, count success rate conditional on the year and attacktype1_txt
  ggplot(aes(iyear, n)) + # x axis- year, y = success rate
  geom_line(aes(group = attacktype1_txt, colour = "grey50")) +
  geom_point(aes(colour = attacktype1_txt))
```



7] Attack Agenda

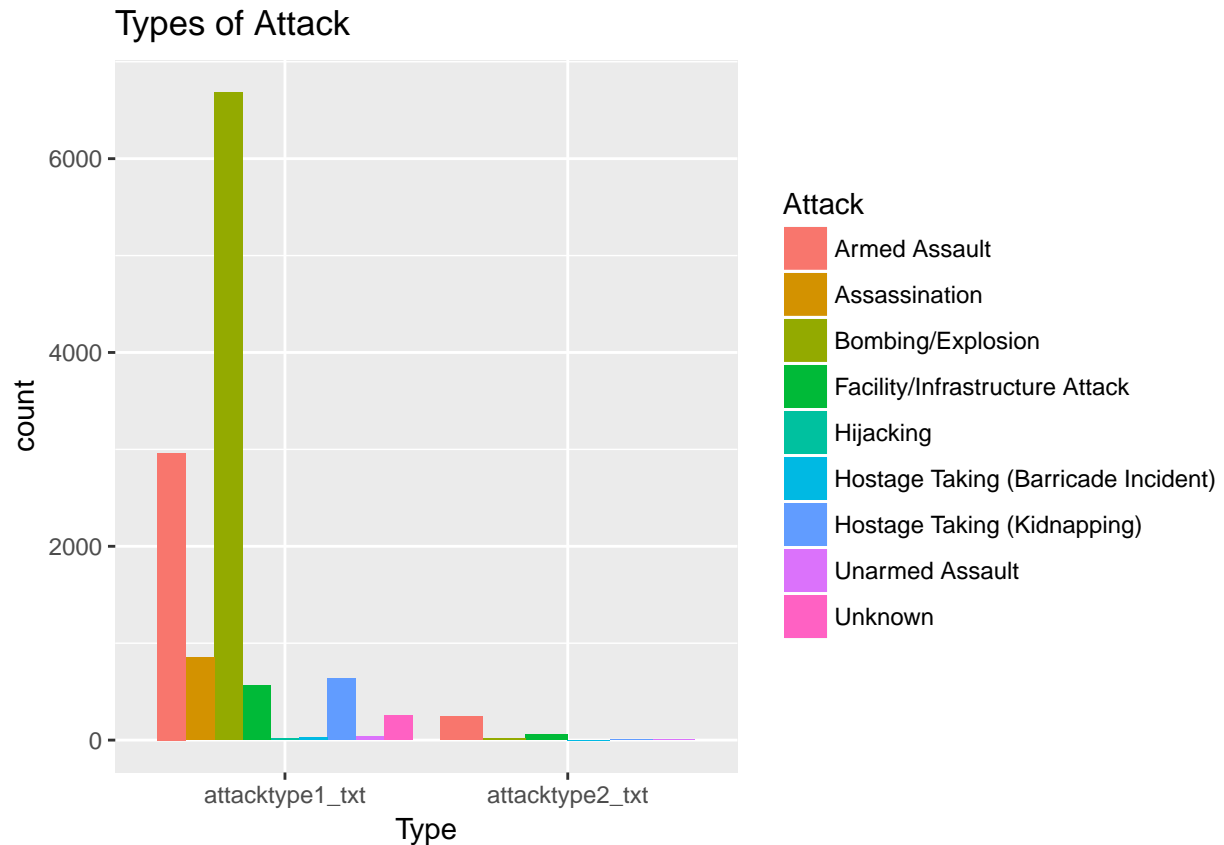
```
# Criterion 1: POLITICAL, ECONOMIC, RELIGIOUS, OR SOCIAL GOAL (CRIT1)
# 1 = The incident meets Criterion 1.
# 0 = "No The incident does not meet Criterion 1.
attack %>%
# With multiple conditions for filter, make sure to use %in% and not ==
filter(parse_character(country_txt) %in% c("United Kingdom")) %>% # select four countries
group_by(iyear,crit1) %>% # allow operations to be performed by groups
count() %>% # In case, count success rate conditional on the year and crit1
print(n=5) %>%
ggplot(aes(iyear, n, fill=crit1)) + # x axis- year, y = success rate
geom_bar(stat="identity") + # stacking on top of each other
ggtitle("Criterion 1")
```

```
## # A tibble: 63 x 3
## # Groups:   iyear, crit1 [63]
##   iyear crit1     n
##   <int> <int> <int>
## 1  1970     1    12
## 2  1971     1    81
## 3  1972     1   292
## 4  1973     1   189
## 5  1974     1   203
## # ... with 58 more rows
```



8] Most frequent attack

```
attack %>%
  select(iyear, attacktype1_txt, attacktype2_txt) %>% # select variable
  filter(iyear == 2013) %>% # subset by rows based on condition
  gather(attacktype1_txt, attacktype2_txt, key = 'Type', value = 'Attack') %>% # transform wide to long
  filter(Attack != ".") %>% # remove all '.'
  ggplot(aes(x=Type, fill=Attack)) + geom_bar(position = "dodge") +# plot
  ggtitle("Types of Attack")
```



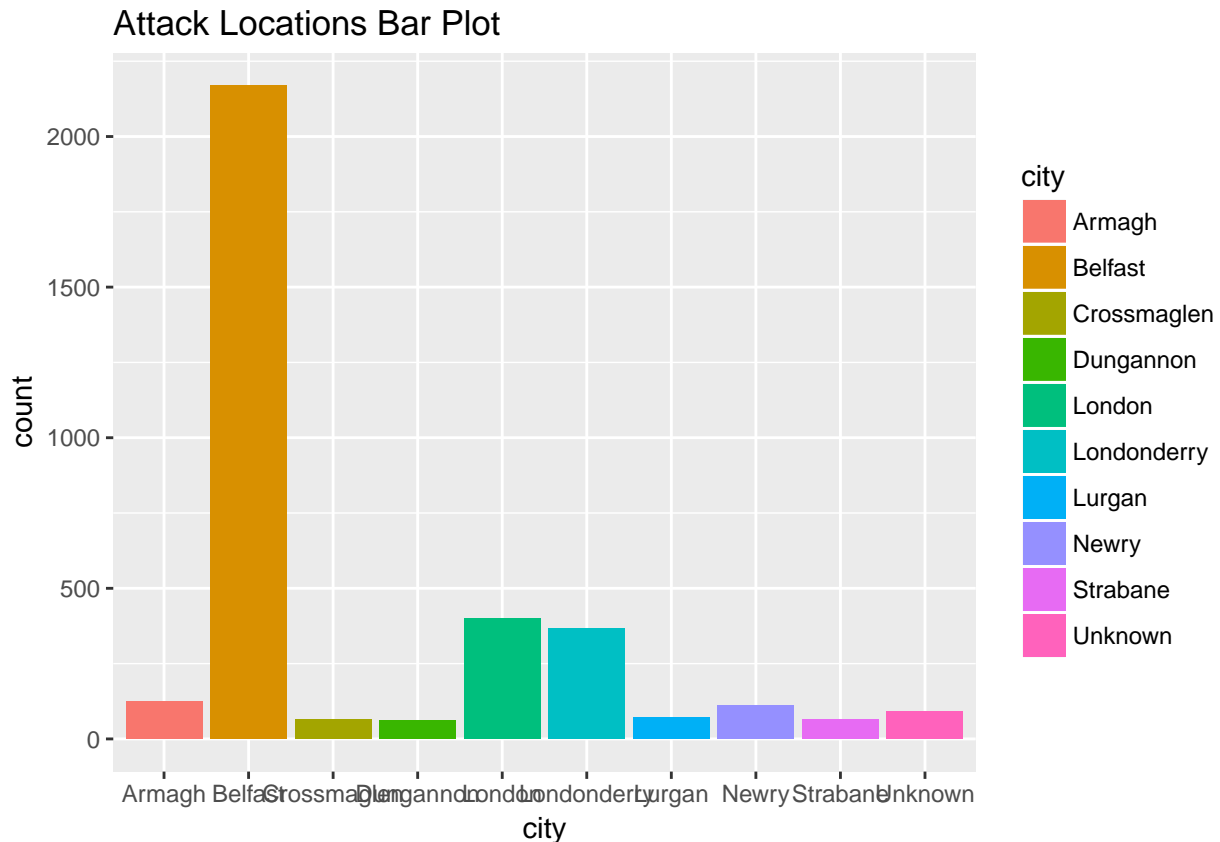
9] Attack Locations Bar Plot

```
attack %>%
  select(iyear, country_txt, city) %>% # select variable
  filter(country_txt == "United Kingdom") %>%
  group_by(city) %>% # group by city
  count() %>% # count the number of times a city appear
  arrange(desc(n)) %>% # subset by rows based on condition
  head(n=10) -> state
state
```

```
## # A tibble: 10 x 2
## # Groups:   city [10]
##   city      n
##   <chr>    <int>
## 1 Belfast  2170
## 2 London   399
## 3 Londonderry 366
## 4 Armagh   125
## 5 Newry    112
## 6 Unknown   90
## 7 Lurgan    71
## 8 Crossmaglen 65
## 9 Strabane  65
## 10 Dungannon 62
```



```
## plot top 10
attack %>%
  filter(country_txt == "United Kingdom") %>%
  select(iyear, city)%>%
  filter(city %in% state$city) %>%
  ggplot(aes(x=city, fill=city)) + geom_bar() + # plot
  ggtitle("Attack Locations Bar Plot")
```

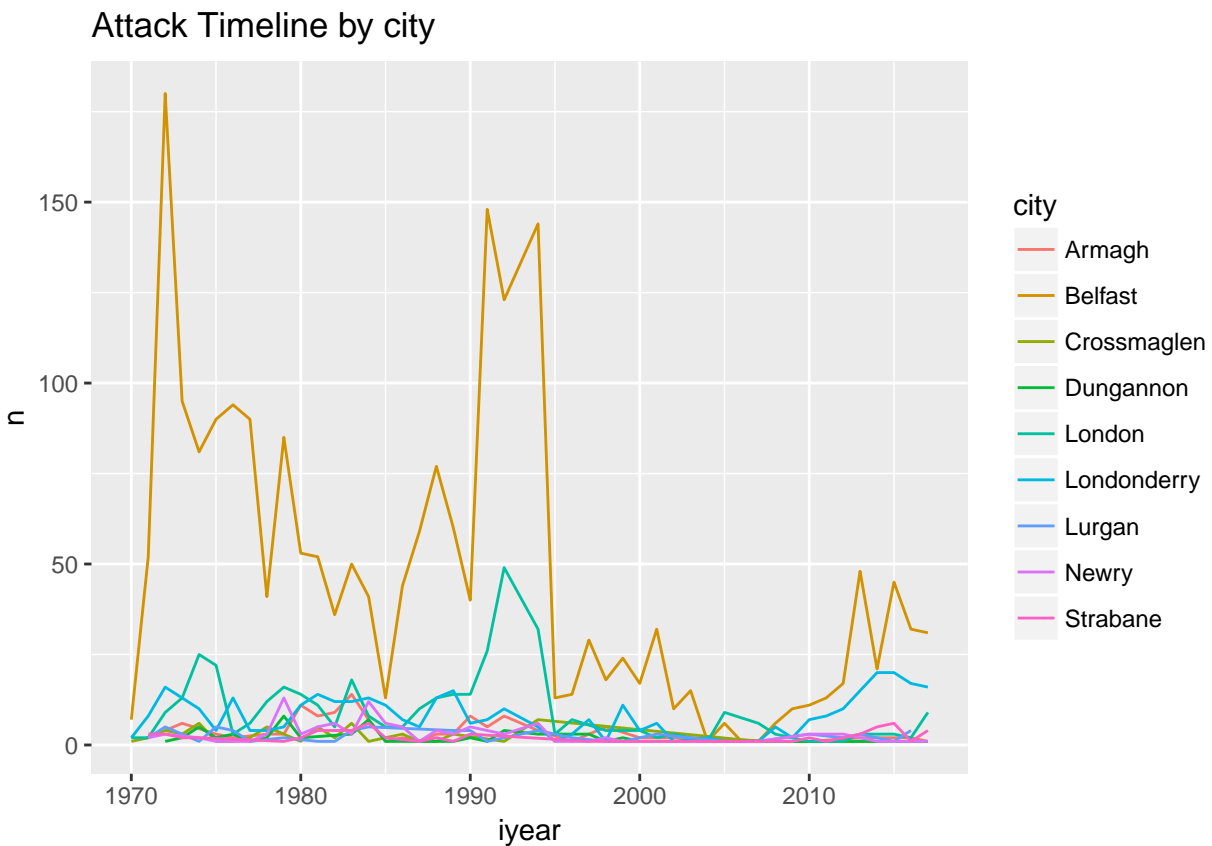


10] Attack Timeline by city

```
attack%>%
  select(iyear, country_txt, city) %>% # select variable
  filter(country_txt == "United Kingdom") %>%
  group_by(city) %>% # group by city
  count() %>% # count the number of times a city appear
  arrange(desc(n))%>% # subset by rows based on condition
  head(n=10) -> state ## plot top 10

attack %>%
  filter(country_txt == "United Kingdom") %>%
  select(iyear, city)%>%
  filter(city %in% state$city) %>% # select top 10
  filter(city != "Unknown") %>% # remove unknown
  group_by(iyear,city) %>% #group to do calculations
```

```
count() %>% # count
ggplot(aes(x=iyyear, y=n)) + geom_line(aes(color=city)) + # plot
ggtitle("Attack Timeline by city")
```



11] Conclusion

From the above example it can be concluded that key Data Science skills which comes handy in any data explorations are below 1] Data Understanding - In depth understanding of the data 2] Technical knowhow - Well versed with technology for data wrangling 3] Data Cleanup and Data imputation - Skilled in advance data imputations techniques to fill missing data 4] Data Visualization - Ability to visualize data on multiple faucets to gain additional insights 5] Data Comparison - Compare meaningful metrics side by side to set important statistics apart 6] Critical Thinking - Ability to come up with multiple questions and sought answers for those through data statistics