

# Phoenix Tracker

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Getting Started . . . . .	1
1.2	Loading Phoenix Dataset . . . . .	1
<b>2</b>	<b>Activity</b>	<b>2</b>
2.1	Loading Dataset . . . . .	3
2.2	Most Active on 2017-02-06 . . . . .	5
2.3	Most Active in Last 7 Days . . . . .	5
2.4	Most Active in Last 30 Days . . . . .	7

## 1 Introduction

This project analyzes event data from the Phoenix Data Project

### 1.1 Getting Started

Load the necessary packages.

```
library(yaml)
library(tidyverse)
```

### 1.2 Loading Phoenix Dataset

The `phoenix.R` script provides a function that simplifies downloading and loading the dataset. We set the start date to the beginning of the year and load the dataset. The `phoenix_load()` function downloads the dataset if necessary.

```
source('R/phoenix.R')

config <- yaml.load_file("config.yml")

events <- phoenix_load(config, start_date = "2017-01-01")
```

Let's see what's in the dataset

```
str(events)
```

```
## 'data.frame': 104356 obs. of 26 variables:
## $ EventID : chr "2867898_v1.3.0" "2867899_v1.3.0" "2867900_v1.3.0" "2867901_v1.3.0" ..
## $ Date : Date, format: "2017-01-02" "2016-12-23" ...
## $ Year : int 2017 2016 2016 2017 2017 2017 2017 2017 2017 2017 ...
## $ Month : int 1 12 12 1 1 1 1 1 1 1 ...
## $ Day : int 2 23 29 2 2 2 2 1 1 2 ...
## $ SourceActorFull : chr "COG" "MNC" "IRN" "DEU" ...
## $ SourceActorEntity : chr "COG" "MNC" "IRN" "DEU" ...
## $ SourceActorRole : chr "" "" "" "" ...
## $ SourceActorAttribute: chr "" "CHE" "" "" ...
## $ TargetActorFull : chr "MED" "USAGOV" "SYR" "GBR" ...
## $ TargetActorEntity : chr "" "USA" "SYR" "GBR" ...
## $ TargetActorRole : chr "" "GOV" "" "" ...
## $ TargetActorAttribute: chr "" "" "" "" ...
## $ EventCode : chr "010" "010" "010" "042" ...
## $ EventRootCode : chr "01" "01" "01" "04" ...
## $ PentaClass : chr "0" "0" "0" "1" ...
## $ GoldsteinScore : num 0 0 0 1.9 -4.4 1.9 0 1 1.9 0 ...
## $ Issues : chr "" "" "" "" ...
## $ Lat : num NA 39.8 55.8 51.5 39.9 ...
## $ Lon : num NA -98.5 37.616 -0.126 116.397 ...
## $ LocationName : chr "" "United States" "Moscow" "London" ...
## $ StateName : chr "" "" "Moskva" "England" ...
## $ CountryCode : chr "" "USA" "RUS" "GBR" ...
## $ SentenceID : chr "586a9d7beaae1f0001eec49c_1" "5869ac06dc0402000134e7ef_0" "5869ac72d57..."
## $ URLs : chr "http://www.nation.co.ke/news/africa/DR-Congo-set-for-talks-on-implement..."
## $ NewsSources : chr "kenya_nation" "local_switzerland" "jpost_me" "yahoo_india;shanghai_na..."
```

Show selected columns

```
events %>%
  select(Date, SourceActorFull, TargetActorFull, EventCode, LocationName) %>%
  head()
```

```
##      Date SourceActorFull TargetActorFull EventCode LocationName
## 1 2017-01-02          COG           MED         010
## 2 2016-12-23        MNCHE        USAGOV         010 United States
## 3 2016-12-29          IRN           SYR         010      Moscow
## 4 2017-01-02          DEU           GBR         042       London
## 5 2017-01-02          CHN           HKG         130       Beijing
## 6 2017-01-02    USAELIGOV        USAGOV         042
```

---

*Last Updated: Feb 06, 2017 1:13 AM*

## 2 Activity

```
library(cshapes)
library(countrycode)
```

```
library(tidyverse)
library(lubridate)
library(broom)
library(yaml)
library(ggrepel)
```

## 2.1 Loading Dataset

Load Phoenix events and few other things we need for plotting.

```
source('R/phoenix.R')

config <- yaml.load_file("config.yml")

events <- phoenix_load(config, "2017-01-01")

country_centroids <- read_csv(config$google$centroids)

world_map <- tidy(cshp(as.Date("2016-06-30")), region = "COWCODE")
```

Function for summarizing events. Return value contains a list of nodes and edges representing dyadic events.

```
get_event_summary <- function(events, centroids, period = 0) {
  events <- events %>%
    filter(Date >= (max(Date) - days(period))) %>%
    filter(SourceActorRole == "GOV" & TargetActorRole == "GOV")

  nodes <- events %>%
    dplyr::mutate(TargetActorEntity = ifelse(SourceActorEntity == TargetActorEntity, NA, TargetActorEntity),
    gather(ActorType, ActorEntity, SourceActorEntity, TargetActorEntity) %>%
    filter(!is.na(ActorEntity), !(ActorEntity == "")) %>%
    group_by(ActorEntity) %>%
    dplyr::summarize(EventCount = n()) %>%
    dplyr::mutate(CountryCode = countrycode(ActorEntity, "iso3c", "cown"), EventCount) %>%
    select(CountryCode, Country = ActorEntity, EventCount) %>%
    left_join(centroids, by = "Country") %>%
    arrange(desc(EventCount))

  edges <- events %>%
    filter(!is.na(SourceActorEntity), !(SourceActorEntity == ""),
    !is.na(TargetActorEntity), !(TargetActorEntity == ""),
    SourceActorEntity != TargetActorEntity) %>%
    rowwise() %>%
    dplyr::mutate(Dyad = paste(sort(c(SourceActorEntity, TargetActorEntity)), collapse = "-")) %>%
    ungroup() %>%
    group_by(Dyad) %>%
    dplyr::summarize(EventCount = n()) %>%
    ungroup() %>%
    separate(Dyad, c("SideA", "SideB"), "-", remove = FALSE) %>%
    dplyr::mutate(CountryA = countrycode(SideA, "iso3c", "country.name"),
    CountryB = countrycode(SideB, "iso3c", "country.name")) %>%
    left_join(centroids, by = c("SideA" = "Country")) %>%

```

```

select(Dyad, SideA, SideB, CountryA, CountryB, EventCount, SideA_Latitude = Latitude, SideA_Longitude = Longitude) %>%
left_join(centroids, by = c("SideB" = "Country")) %>%
select(Dyad, SideA, SideB, CountryA, CountryB, EventCount, SideA_Latitude, SideA_Longitude, SideB_Latitude, SideB_Longitude) %>%
arrange(desc(EventCount))

return(list(nodes = nodes, edges = edges))
}

```

Function for plotting activity on a map.

```

plot_activity <- function(map, event_summary) {
  map <- map %>%
    dplyr::mutate(id = as.numeric(id)) %>%
    left_join(event_summary$nodes, by = c("id" = "CountryCode"))

  ggplot(map) +
    geom_map(map = map, aes(map_id = id, fill = EventCount), color = "gray", size = 0.5) +
    scale_fill_distiller(name = "Event Count", palette = "Blues", direction = 1, na.value = "white") +
    expand_limits(x = map$long, y = map$lat) +
    coord_cartesian() +
    geom_point(aes(x = Longitude, y = Latitude),
      data = event_summary$nodes,
      size = 0.5,
      color = "navy") +
    geom_curve(aes(x = SideA_Longitude,
      y = SideA_Latitude,
      xend = SideB_Longitude,
      yend = SideB_Latitude),
      data = event_summary$edges,
      size = 0.2,
      alpha = 0.5,
      color = "red") +
    geom_text_repel(aes(x = Longitude, y = Latitude, label = CountryName),
      data = head(event_summary$nodes, 10),
      force = 0.1,
      size = 3,
      fontface = "bold") +
    theme_minimal() +
    theme(legend.position = "bottom",
      legend.key.width = unit(5, "line"),
      axis.title = element_blank(),
      axis.text = element_blank(),
      panel.grid = element_blank())
}

```

This function simply formats the top 10 rows from the dataset in a pretty table.

```

show_top10 <- function(x) {
  x %>%
    head(n = 10) %>%
    knitr::kable()
}

```

## 2.2 Most Active on 2017-02-06

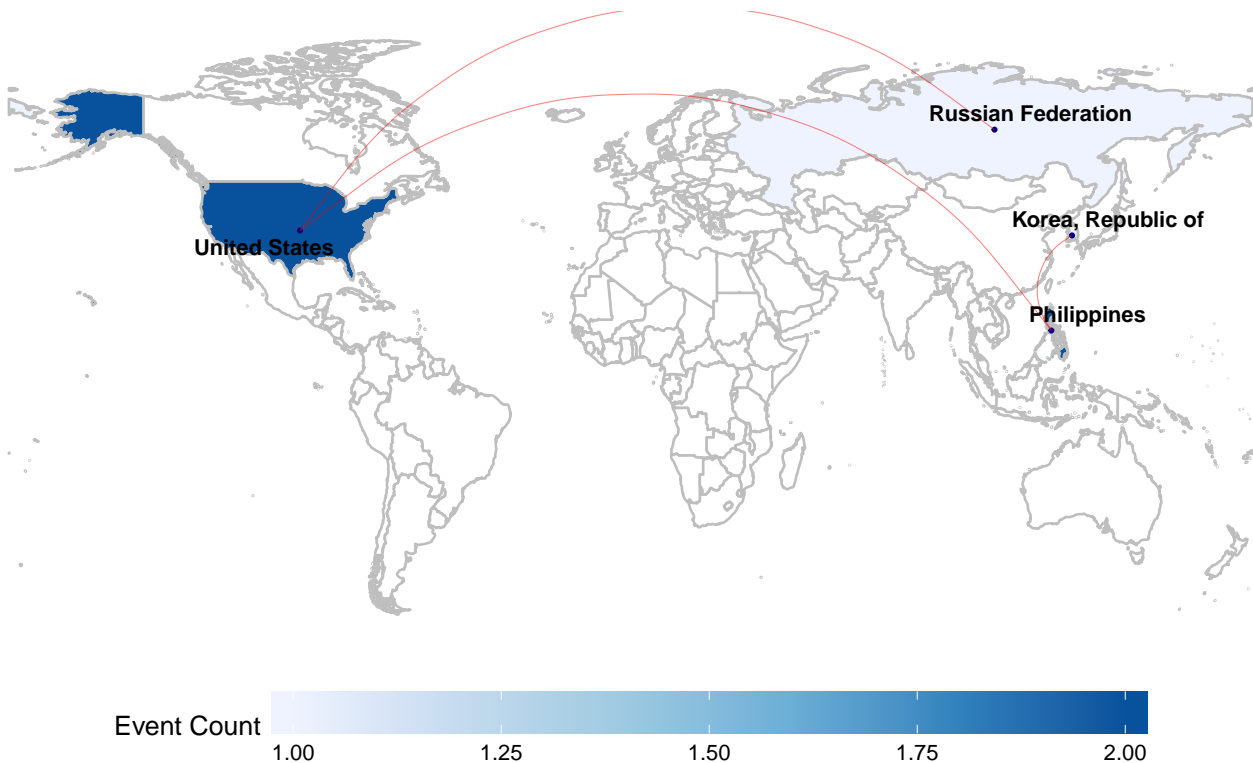
```
event_summary <- get_event_summary(events, country_centroids)
show_top10(select(event_summary$nodes, CountryName, EventCount))
```

CountryName	EventCount
Philippines	2
United States	2
Korea, Republic of	1
Russian Federation	1

```
show_top10(select(event_summary$edges, CountryA, CountryB, EventCount))
```

CountryA	CountryB	EventCount
Korea, Republic of	Philippines	1
Philippines	United States	1
Russian Federation	United States	1

```
plot_activity(world_map, event_summary)
```



## 2.3 Most Active in Last 7 Days

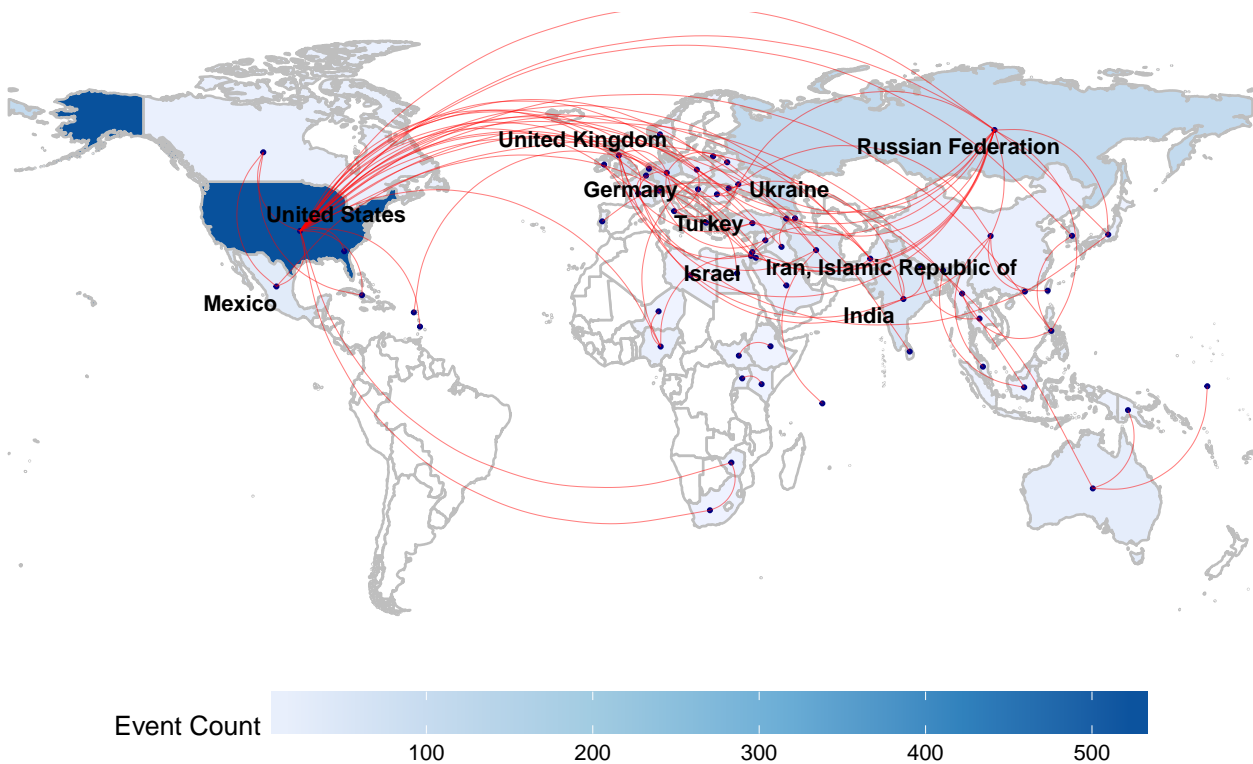
```
event_summary <- get_event_summary(events, country_centroids, period = 7)
show_top10(select(event_summary$nodes, CountryName, EventCount))
```

CountryName	EventCount
United States	525
Russian Federation	113
United Kingdom	81
Israel	77
Germany	76
India	57
Iran, Islamic Republic of	51
Turkey	42
Mexico	37
Ukraine	33

```
show_top10(select(event_summary$edges, CountryA, CountryB, EventCount))
```

CountryA	CountryB	EventCount
Iran, Islamic Republic of	United States	43
United Kingdom	United States	35
Germany	Turkey	34
Israel	United States	34
Russian Federation	United States	33
Korea, Republic of	United States	24
Australia	United States	22
India	United States	20
Mexico	United States	19
United Kingdom	Israel	17

```
plot_activity(world_map, event_summary)
```



## 2.4 Most Active in Last 30 Days

```
event_summary <- get_event_summary(events, country_centroids, period = 30)
show_top10(select(event_summary$nodes, CountryName, EventCount))
```

CountryName	EventCount
United States	2167
United Kingdom	407
Russian Federation	404
Israel	243
Germany	236
Iran, Islamic Republic of	197
China	186
India	177
Mexico	161
Turkey	143

```
show_top10(select(event_summary$edges, CountryA, CountryB, EventCount))
```

CountryA	CountryB	EventCount
Russian Federation	United States	224
United Kingdom	United States	188
Israel	United States	152
Mexico	United States	111
Iran, Islamic Republic of	United States	109
India	United States	78
Germany	United States	54
Germany	France	44
Germany	Turkey	41
United Kingdom	Turkey	40

```
plot_activity(world_map, event_summary)
```

