GIS Data Solution

Alicia R. Chen

6/1/2021

```
packages <- c("lubridate", "data.table", "rgdal", "rgeos", "data.table",
              "dplyr", "readr", "ggplot2", "tools", "RColorBrewer", "grid")
lapply(packages, library, character.only = TRUE)
```

```
## [[1]]
## [1] "lubridate" "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [7] "methods"   "base"
##
## [[2]]
## [1] "data.table" "lubridate"  "stats"      "graphics"   "grDevices"
## [6] "utils"      "datasets"   "methods"    "base"
##
## [[3]]
##  [1] "rgdal"      "sp"         "data.table" "lubridate"  "stats"
##  [6] "graphics"   "grDevices"  "utils"      "datasets"   "methods"
## [11] "base"
##
## [[4]]
##  [1] "rgeos"      "rgdal"      "sp"         "data.table" "lubridate"
##  [6] "stats"      "graphics"   "grDevices"  "utils"      "datasets"
## [11] "methods"    "base"
##
## [[5]]
##  [1] "rgeos"      "rgdal"      "sp"         "data.table" "lubridate"
##  [6] "stats"      "graphics"   "grDevices"  "utils"      "datasets"
## [11] "methods"    "base"
##
## [[6]]
##  [1] "dplyr"      "rgeos"      "rgdal"      "sp"         "data.table"
##  [6] "lubridate"  "stats"      "graphics"   "grDevices"  "utils"
## [11] "datasets"   "methods"    "base"
##
## [[7]]
##  [1] "readr"      "dplyr"      "rgeos"      "rgdal"      "sp"
##  [6] "data.table" "lubridate"  "stats"      "graphics"   "grDevices"
## [11] "utils"      "datasets"   "methods"    "base"
##
## [[8]]
##  [1] "ggplot2"    "readr"      "dplyr"      "rgeos"      "rgdal"
##  [6] "sp"         "data.table" "lubridate"  "stats"      "graphics"
## [11] "grDevices"  "utils"      "datasets"   "methods"    "base"
##
```

```
## [[9]]
##  [1] "tools"      "ggplot2"    "readr"      "dplyr"      "rgeos"
##  [6] "rgdal"      "sp"         "data.table" "lubridate"  "stats"
## [11] "graphics"   "grDevices"  "utils"      "datasets"   "methods"
## [16] "base"
##
## [[10]]
##  [1] "RColorBrewer" "tools"        "ggplot2"      "readr"        "dplyr"
##  [6] "rgeos"        "rgdal"        "sp"           "data.table"   "lubridate"
## [11] "stats"        "graphics"     "grDevices"    "utils"        "datasets"
## [16] "methods"      "base"
##
## [[11]]
##  [1] "grid"         "RColorBrewer" "tools"        "ggplot2"      "readr"
##  [6] "dplyr"        "rgeos"        "rgdal"        "sp"           "data.table"
## [11] "lubridate"    "stats"        "graphics"     "grDevices"    "utils"
## [16] "datasets"     "methods"      "base"
```

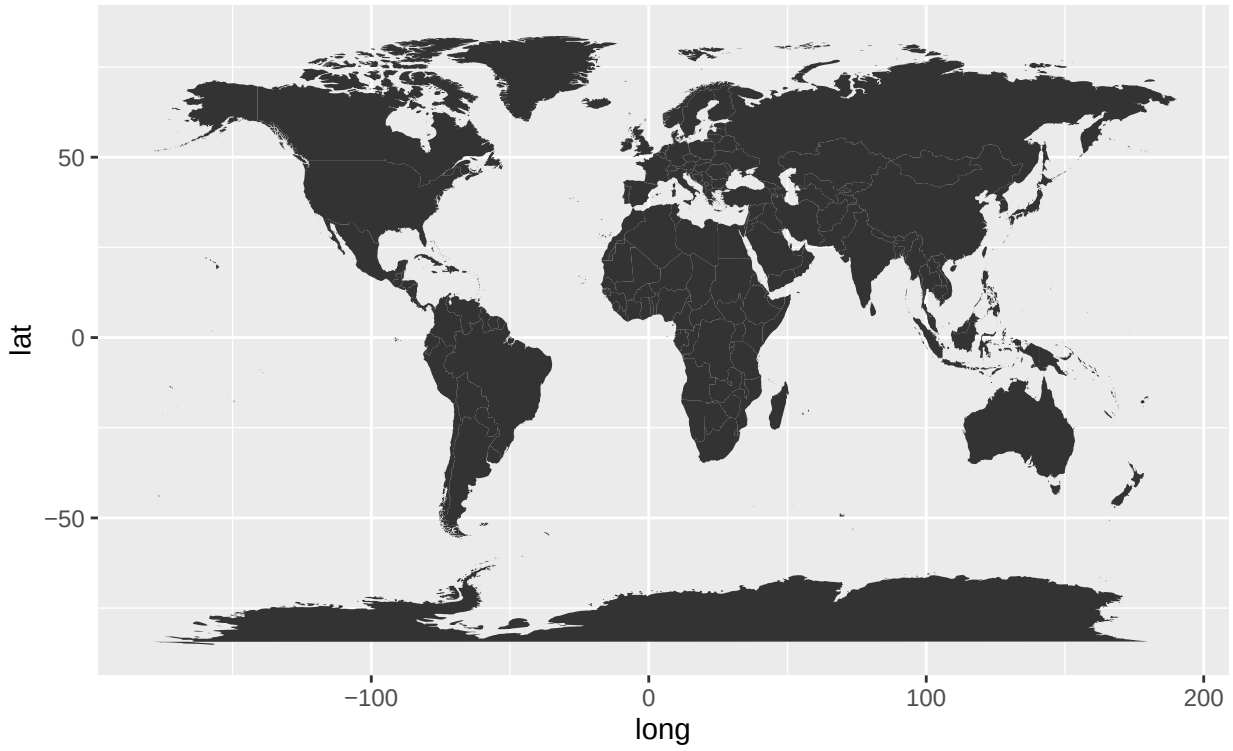
```
# load data
df <- fread("./all_flow_classes.csv")

# examine types of Chinese investments included:
df %>% group_by(flow_class) %>% summarise(n())
```

```
## # A tibble: 3 x 2
##   flow_class               `n()`
##   <chr>                    <int>
## 1 ODA-like                  4315
## 2 OOF-like                  1140
## 3 Vague (Official Finance)   735
```

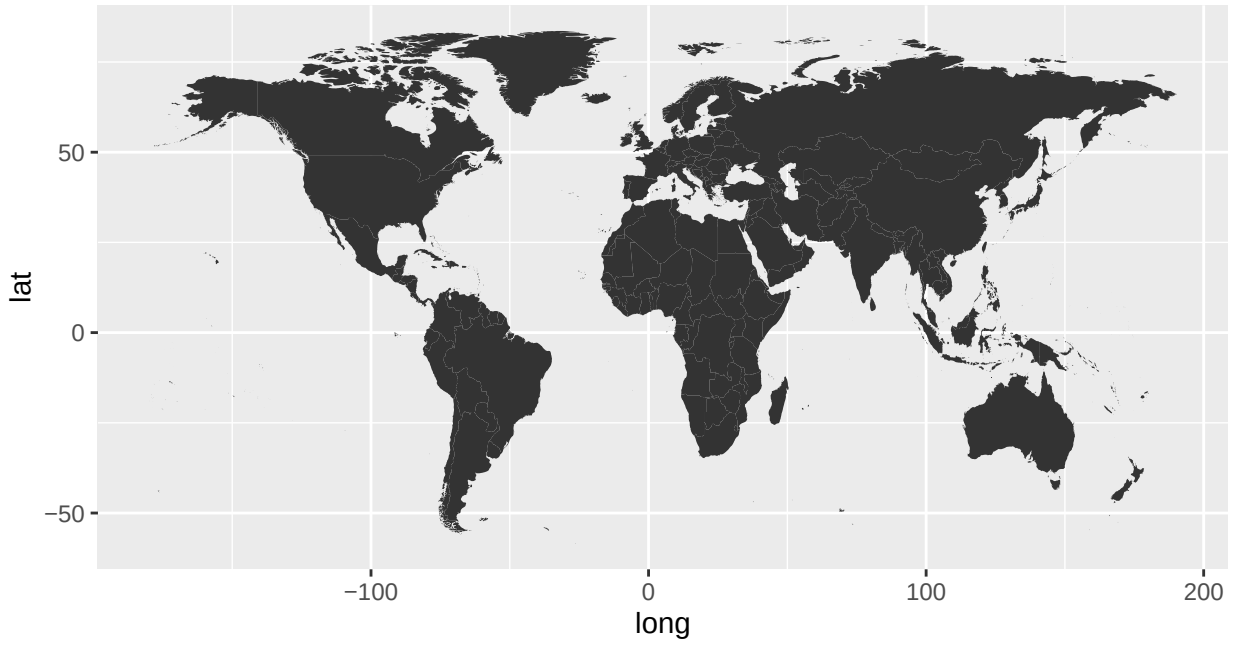
```
# load project descriptions
desc <- fread("./project_descriptions_and_sources.csv")

# merge by project_id
final_df <- plyr::join(df, desc, by = 'project_id')
final_df <- final_df[!(is.na(final_df$latitude)==TRUE),]
```

Q1

```
world_map <- map_data("world")
ggplot() +
  geom_polygon(data = world_map,
               aes(x=long, y = lat, group = group)) +
  coord_fixed(1.3)
```

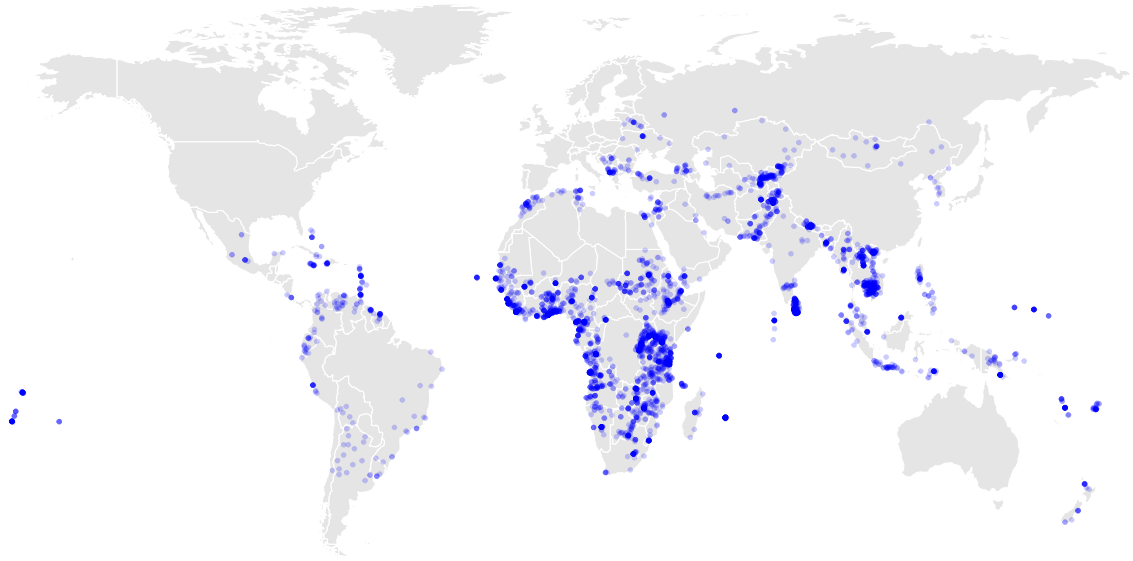



```
#remove Antarctica
world_map <- world_map[world_map$region!="Antarctica",]
ggplot() +
  geom_polygon(data = world_map,
               aes(x=long, y = lat, group = group)) +
  coord_fixed(1.3)
```



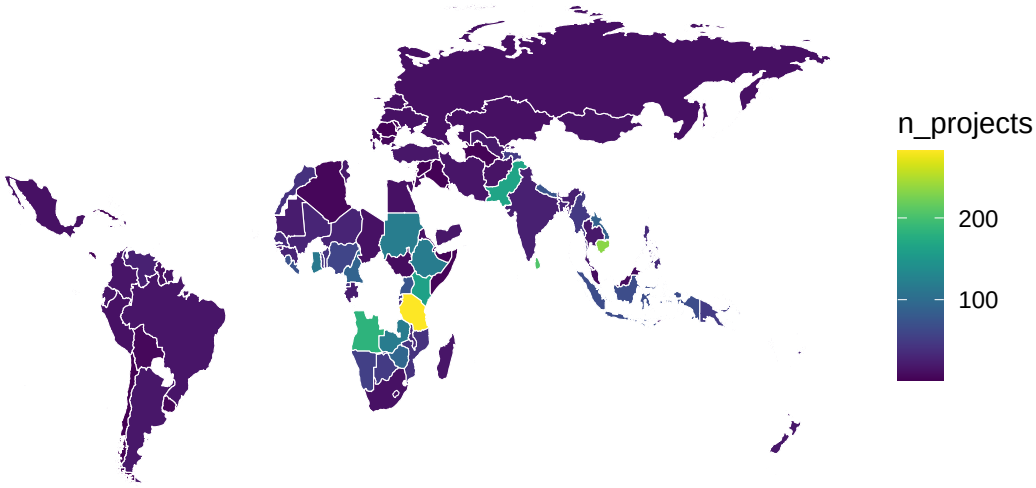

```
# plot Chinese investments
ggplot() +
  geom_polygon(data = world_map,
```

```
              aes(x=long, y = lat, group = group),
              fill='gray90', size=0.2, color='white') +
  coord_fixed(1.3) +
  theme_void() +
  geom_point(data = final_df,
             aes(x=longitude,y=latitude),
             color='blue',alpha=0.2, size=0.3)
```



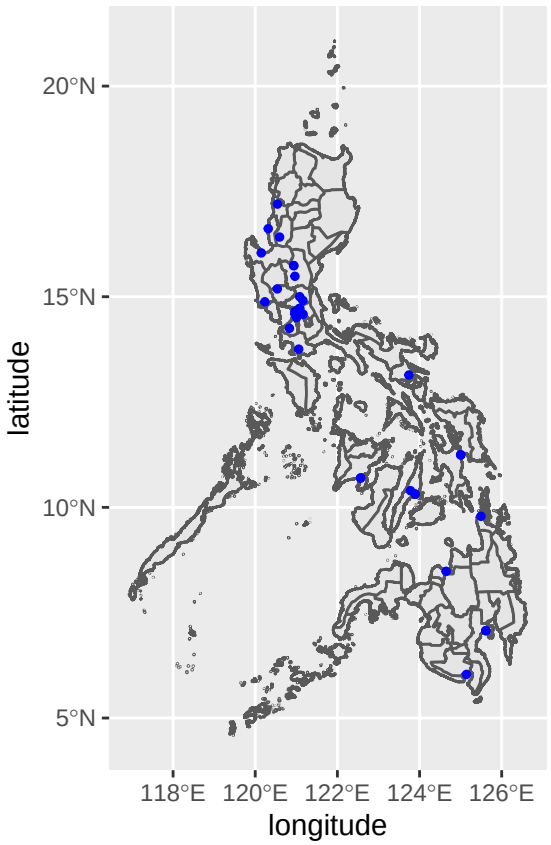
There are lots of ways to play around with visualizing this data. One example is this:

```
# some projects have multiple recipients
# split this data such that each row is one recipient
by_recipient <- splitstackshape::cSplit(final_df, "recipients", sep="|", "long")
by_recipient <- by_recipient[by_recipient$recipients != "Africa, regional",]

# calculate number of projects per country
by_recipient <- by_recipient %>% count(recipients)
colnames(by_recipient) <- c('region', 'n_projects')

# merge with world map
merged <- inner_join(world_map, by_recipient, by = "region")

ggplot() +
  geom_polygon(data = merged,
               aes(x=long, y = lat, group = group, fill=n_projects),
               size=0.2, color='white') +
  coord_fixed(1.3) +
  theme_void() +
  viridis::scale_fill_viridis()
```



Q2

```
ph <- readRDS("../../Solutions (NOT FOR FELLOWS)/gadm36_PHL_1_sf.rds")

# get projects in philippines
ph_projects <- final_df[grepl("Philippines", final_df$recipients)==TRUE,]

ggplot() +
  geom_sf(data = ph) +
  geom_point(ph_projects,
             mapping=aes(x=longitude,y=latitude), color='blue',size=1) +
  coord_sf()
```



Q3

```
#my_key <- "YOUR OWN KEY"
library(ggmap)
register_google(key = my_key)

jakarta_coord <- geocode(location = "Jakarta", output = "more", source = "google")
id_projects <- final_df[grepl("Indonesia", final_df$recipients)==TRUE,]
id_projects$jak_lon <- jakarta_coord$lon
id_projects$jak_lat <- jakarta_coord$lat

id_projects$distance<-geosphere::distHaversine(id_projects[,c("longitude", "latitude")],
                                               id_projects[,c("jak_lon", "jak_lat")])
id_projects$distance_in_miles <- id_projects$distance / 1609

summary(id_projects$distance_in_miles)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##    0.4065  264.0267  412.6843  607.1598  980.5363 2147.6488
```