Foreign Assistance Choropleth Map

J. Dayton

5/19/2020

Financial Data

Financial data includes both: (Source: https://www.foreignassistance.gov/learn/understanding-the-data)

- obligated data – funds the U.S. government decides can be mobilized
- spent data – funds the U.S. government has mobilized to purchase goods and services.

U.S. government agencies report financial transaction data to ForeignAssistance.gov from their accounting systems on a quarterly basis. The fiscal years associated with obligated and spent transaction data on ForeignAssistance.gov represent the years in which those transactions took place. Transaction data is the most granular form of financial data. Transaction data represents every individual financial record in an agency's accounting system for program work with implementing partners and administrative expenses.

Obligated

Obligated data represents the funds that U.S. government agencies commit to spend on foreign assistance efforts. Agencies take their appropriated funds and commit them to specific projects or activities.

Obligations are also known as commitments because they are the funds which have been applied to activities that obligate/commit the U.S. government to make outlays either immediately or in the future for those given activities. Funds can be de-obligated, in which case they would appear as negative values in our dataset.

Spent

Spent data represents funds the U.S. government has outlaid, disbursed, or expended to purchase goods and services during a given quarter of a fiscal year.

Disbursements

Disbursements are the funds paid/outlaid by U.S. government agencies, by cash or cash equivalent, during the fiscal year to liquidate government obligations.

Expenditures

Expenditures are the actual disbursement of funds in return for goods and services.

STEP 0: Load Libraries

Data source:

http://foreignassistance.gov/

General Methodology / Outline

There are four general steps to creating these charts:

- 1. Import libraries and relevant data set.
- 2. Aggregate data by country (or other location).
- 3. Merge data from data set to map data.
- 4. Create the plot in ggplot2.

STEP 1: Import libraries and relevant data set.

Get the Foreign Assistance Data

The Foreign Assistance data is from https://www.foreignassistance.gov/developers. The Foreign Assistance files were downloaded in .csv for FY19 - the focus of this example.

```
data <- read.csv(paste0("./ForeignAssistance-FullDataSet/ForeignAssistance-FullDataSet-2018-and-Later.c
## FOR FY19
dfAll <- subset(data, Transaction...Fiscal.Year == 2019)
rm(data)  #Save memory - no further calls to this original data element

## FOR Disbursement and Expenditure
dfAll <- subset(dfAll, Transaction...Type %in% c("Disbursement", "Expenditure"))

## For CWMD
dfAll <- subset(dfAll, Transaction...U.S..Foreign.Assistance.Sector.Name == "Combating Weapons of Mass
unique(dfAll$Managing.Agency)
```

```
## [1] "U.S. Department of State"
## [2] "U.S. Department of Energy"
## [3] "U.S. Department of Defense"
## [4] "U.S. Department of Homeland Security"
```

```
unique(dfAll$Treasury.Account...Title)
```

```
## [1] "Nonproliferation, Anti-Terrorism, Demining and Related Programs, International Security Assista
## [2] "Defense Nuclear Nonproliferation, Energy"
## [3] "Cooperative Threat Reduction Account, Defense"
## [4] "Nonproliferation and Disarmament Fund, Funds Appropriated to the President"
```

```
print(paste0("The dimensions of the consolidated data are ", dim(dfAll), "."))
```

```
## [1] "The dimensions of the consolidated data are 1648."
## [2] "The dimensions of the consolidated data are 39."
```

```
kable(head(data.frame(Department = dfAll[, 3], Agency = dfAll[, 4], Location = dfAll[, 13], Value = dfAl
kable_styling(latex_options = c("striped", "scale_down"))
```

Department	Agency	Location	Value	FY	TransType	TreasAcctTitle
U.S. Department of State	Bureau of International Security and Nonproliferation	Worldwide	12137.00	2019	Disbursement	Nonproliferation, Anti-Terrorism, Demining and Related Programs, International Security Assistance, State
U.S. Department of Energy	National Nuclear Security Administration/Office of Defense Nuclear Non-Proliferation/Office of Global Material Security	Asia (South & Central Asia)	1485672.00	2019	Disbursement	Defense Nuclear Nonproliferation, Energy
U.S. Department of State	Bureau of International Security and Nonproliferation	Turkey	18023.00	2019	Disbursement	Nonproliferation, Anti-Terrorism, Demining and Related Programs, International Security Assistance, State
U.S. Department of State	Bureau of International Security and Nonproliferation	Worldwide	-27697.15	2019	Disbursement	Nonproliferation, Anti-Terrorism, Demining and Related Programs, International Security Assistance, State
U.S. Department of State	Bureau of International Security and Nonproliferation	Worldwide	270226.78	2019	Disbursement	Nonproliferation, Anti-Terrorism, Demining and Related Programs, International Security Assistance, State
U.S. Department of State	Bureau of International Security and Nonproliferation	Pakistan	0.00	2019	Disbursement	Nonproliferation, Anti-Terrorism, Demining and Related Programs, International Security Assistance, State

STEP 2. Aggregate data by country (or other location).

```
## For USG
dfUSG <- dfAll %>%
  group_by(Location = Award...Benefitting.Location) %>%
  summarise(Amount = sum(Transaction...Value), Initiatives = n())

## For DoD
dfDoD <- subset(dfAll, Managing.Agency == "U.S. Department of Defense")
unique(dfDoD$Managing.Agency...Office)
```

```
## [1] "Defense Threat Reduction Agency"
```

```
unique(dfDoD$Treasury.Account...Title)
```

```
## [1] "Cooperative Threat Reduction Account, Defense"
```

```
dfDoD <- dfDoD %>%
  group_by(Location = Award...Benefitting.Location) %>%
  summarise(Amount = sum(Transaction...Value), Initiatives = n())

region <- c("Africa (South of Sahara)", "America (North & Central America)", "America (South America)",

#Track areas not included in map
dfUSGRegion <- removeRegion(df = dfUSG, sort.col = dfUSG$Location, val = FALSE)
#Take out regional reference
dfUSG <- removeRegion(df = dfUSG, sort.col = dfUSG$Location)
#Track areas not included in map
dfDoDRegion <- removeRegion(df = dfDoD, sort.col = dfDoD$Location, val = FALSE)
#Take out regional reference
dfDoD <- removeRegion(df = dfDoD, sort.col = dfDoD$Location)
```

STEP 3. Merge data from data set to map data.

```
#Get the country data
data("countryExData", envir = environment(), package = "rworldmap")
dfMergeUSG <- joinCountryData2Map(dfUSG, joinCode = "NAME",
                                  nameJoinColumn = "Location", verbose = "TRUE")
```

```
## 63 codes from your data successfully matched countries in the map
## 0 codes from your data failed to match with a country code in the map
##      failedCodes failedCountries
## 180 codes from the map weren't represented in your data
```

```r
#Take care of NA values
# dfMergeUSG$Amount[is.na(dfMergeUSG[["Amount"]])] <- 0
#Scale Amount to $M
dfMergeUSG$Amount <- dfMergeUSG$Amount / 1000000

#Get country data
dfMergeDOD <- joinCountryData2Map(dfDoD, joinCode = "NAME",
                                  nameJoinColumn = "Location", verbose = "TRUE")
```

```
## 33 codes from your data successfully matched countries in the map
## 0 codes from your data failed to match with a country code in the map
##      failedCodes failedCountries
## 210 codes from the map weren't represented in your data
```

```r
#Take care of NA values
# dfMergeDOD$Amount[is.na(dfMergeDOD[["Amount"]])] <- 0
#Scale Amount to $M
dfMergeDOD$Amount <- dfMergeDOD$Amount / 1000000
```

STEP 4. Create the plot(s).

```r
mapDevice() #create world map shaped window
mapCountryData(dfMergeUSG, nameColumnToPlot = "Amount")
```

```r
#getting class intervals
classInt <- classIntervals(dfMergeUSG[["Amount"]], n = 7, style = "jenks")
catMethod = classInt[["brks"]]

#getting colours
colourPalette <- brewer.pal(7, "Greens")

#plot map
mapDevice() #create world map shaped window
mapParams <- mapCountryData(dfMergeUSG, nameColumnToPlot = "Amount",
                            addLegend = FALSE, catMethod = catMethod,
                             mapTitle = "US Government FY19 Foreign Assistance - CWMD Sector ($M)",
                            colourPalette = colourPalette, oceanCol = "white",
                            missingCountryCol = "grey98",
                            borderCol = "grey10")
#adding legend
do.call(addMapLegend , c(mapParams, legendLabels = "all", legendWidth = 0.5,
                         legendIntervals = "data",
                         legendMar = 2))
```

```r
dfUSGRegion <- dfUSGRegion %>% arrange(desc(Amount)) %>%
   mutate(Amount = round(Amount / 1000000, 2)) %>%
   select(Location, Initiatives, Amount)
colnames(dfUSGRegion) <- c("Region", "Initiatives", "Amount ($M)")
kable(dfUSGRegion)
```

Region	Initiatives	Amount ($M)
Worldwide	752	184.45
Europe	74	39.89
Asia (South & Central Asia)	46	20.68
Asia (Far East Asia)	51	10.93
Asia (Middle East Asia)	11	7.02
Africa (South of Sahara)	24	5.51
America (South America)	7	5.20
America (North & Central America)	8	1.14
Oceania	7	-0.06

```r
#getting class intervals
classInt <- classIntervals(dfMergeDOD[["Amount"]], n = 7, style = "jenks")
catMethod = classInt[["brks"]]

#getting colours
colourPalette <- brewer.pal(7, "BuPu")

#plot map
mapDevice() #create world map shaped window
mapParams <- mapCountryData(dfMergeDOD, nameColumnToPlot = "Amount",
                            addLegend = FALSE, catMethod = catMethod,
                             mapTitle = "Department of Defense FY19 Foreign Assistance - CWMD Sector ($
                            colourPalette = colourPalette, oceanCol = "white",
                            missingCountryCol = "grey99",
                            borderCol = "grey10")
#adding legend
do.call(addMapLegend , c(mapParams, legendLabels = "all", legendWidth = 0.5,
                         legendIntervals = "data",
                         legendMar = 2))
```

```r
#plot map
mapDevice() #create world map shaped window
mapParams <- mapCountryData(dfMergeDOD, nameColumnToPlot = "Amount",
                            addLegend = FALSE, catMethod = catMethod,
                             mapTitle = "Department of Defense FY19 Foreign Assistance - CWMD Sector ($
                            colourPalette = colourPalette, oceanCol = "white",
                            missingCountryCol = "grey99",
                            borderCol = "grey10",
                            xlim = c(85, 100),
                            ylim = c(-60, 90))
#adding legend
do.call(addMapLegend , c(mapParams, legendLabels = "all", legendWidth = 0.5,
                         legendIntervals = "data",
                         legendMar = 2))
```

```r
dfDoDRegion <- dfDoDRegion %>% arrange(desc(Amount)) %>%
   mutate(Amount = round(Amount / 1000000, 2)) %>%
   select(Location, Initiatives, Amount)
colnames(dfDoDRegion) <- c("Region", "Initiatives", "Amount ($M)")
kable(dfDoDRegion)
```

Region	Initiatives	Amount ($M)
Worldwide	10	90.50
Europe	2	7.82
Asia (Middle East Asia)	1	1.20
America (South America)	1	0.25