

data_intro

Clara Suong

18 February, 2019

Load libraries and set the working directory

```
rm(list = ls())
library(plyr)
library(dplyr)
library(dbplyr)
library(tidyverse)
library(tidyquant) # Loads tidyverse, tidquant, financial pkgs, xts/zoo
library(xts) #Time series
library(RMySQL) #For connecting to the database
#library(sjPlot) #For creating Word-compatible tables
#library(labelled)
library(htmlTable) #For creating Word-compatible tables
library(Hmisc)
library(expss) #For creating Word-compatible tables
library(sjlabelled)
library(lubridate)
library(foreign)
library(ggplot2)
library(reshape2)
library(readxl)
library(countrycode) #For reconciling different country codes across dataset
library(fuzzyjoin) #For reconciling different country codes across dataset
library(ISOcodes) #A package for ISO country codes
library(MASS) #Negative binomial models
library(stargazer)

getwd()
#setwd("/Users/clarahsuong/Dropbox/nyu_postdoc/ner/dataset_intro") #Set the working directory in the lo
#Create a folder named "data_comparison" in the working directory
```

Databases and external datasets

Our MySQL databases

- declassification_cables
- declassification_ddrs
- declassification_frus
- declassification_kissinger
- declassification_pdb
- declassification_clinton
- declassification_cabinet
- declassification_cpdoc

Key fields/variables in our database ‘declassification_cables’

- body
- subject
- date (year)
- classification
- urgency
- length
- (handling)
- (page_count)
- (line_count)
- office
- from_field
- to_field
- tag
- (Derived from TAGS below)
- country
- person
- topic
- frus_match
- Cf. label dictionary: <https://docs.google.com/document/d/13iM00ZfVzV-6mGw8YBGkJFnJFkLe011znb3LQenaIjM/edit?usp=sharing>

External datasets:

- Download the following datasets in the folder “data_comparison”
- COW country codes (cow): http://www.correlatesofwar.org/data-sets/cow-country-codes/cow-country-codes/at_download/file
- U.S. diplomatic representation (us_dip_rep; COW-compatible version): https://www.dropbox.com/sh/2wnklx04vblnmi1/AABmMxbxvja_JVStsxKD4F2Qa?dl=0
- U.S. diplomatic visits (us_dip_vis): <https://tinyurl.com/yyedcahu>
- U.S. diplomatic appointments (us_dip_app): https://static-content.springer.com/esm/art%3A10.1007%2Fs11558-017-9277-0/MediaObjects/11558_2017_9277_MOESM1_ESM.zip
- UN voting (un_vote): <https://dataverse.harvard.edu/dataset.xhtml?persistentId=hdl:1902.1/12379>
- Actor-level International Crisis Behavior (icb; Version 12): <https://sites.duke.edu/icbdata/data-collections/>
- U.S. diplomatic events (us_dip_evt): <https://www.dropbox.com/sh/d93tdqanxugvlg/AABYV97aMEE1Ydh0mkihigrSa?dl=0>

Exploring the database ‘declassification_cables’

Connecting to the database ‘declassification_cables’

Tables in the database ‘declassification_cables’

```
dbListTables(mydb)
```

```
## [1] "classification_countries" "classification_doc"
## [3] "classifications"         "concept_doc"
## [5] "concepts"                "countries"
## [7] "country_doc"             "doc_counts"
## [9] "docs"                    "from_to_sum"
## [11] "network_docs"            "network_nodes"
## [13] "office_doc"              "offices"
## [15] "person_doc"              "persons"
## [17] "reference_doc"           "tag_doc"
## [19] "tag_doc_staging"         "tagname_doc"
## [21] "tagnames"                "tags"
## [23] "tags_staging"            "tokens"
## [25] "top_classifications"     "top_countries"
## [27] "top_network"             "top_persons"
## [29] "top_topics"              "topic_doc"
## [31] "topic_token"             "topics"
## [33] "urgency"                 "urgency_doc"
```

#Note that many of the tables are yet to be populated (or in the process of being so).

Exploring the main table ‘docs’ from the database ‘declassification_cables’

```
## [1] "doc_nbr"                "auto_decaption"
## [3] "reference"               "capture_date"
## [5] "channel"                 "concepts"
## [7] "control_nbr"             "copy"
## [9] "date"                   "decaption_date"
## [11] "decaption_note"          "disp_action"
## [13] "disp_approved_on_date"   "disp_case"
## [15] "disp_comment"           "disp_date"
## [17] "disp_event"             "disp_history"
## [19] "disp_reason"            "disp_remarks"
## [21] "doc_source"             "drafter"
## [23] "enclosure"              "eo"
## [25] "errors"                 "expiration"
## [27] "film"                   "handling"
## [29] "isecure"                "legacy_key"
## [31] "line_count"             "litigationhistory"
## [33] "locator"                "messageid"
## [35] "office"                 "origclass"
## [37] "orighand"               "origpclass"
## [39] "origphand"              "page_count"
## [41] "pchannel"               "pclass"
## [43] "phandling"              "retention"
```

```

## [45] "review_action"          "review_content_flags"
## [47] "review_date"           "review_event"
## [49] "review_exemptions"     "review_media_id"
## [51] "review_release_date"   "review_release_event"
## [53] "review_transfer_date"  "review_withdrawn_fields"
## [55] "review_markings"       "sasid"
## [57] "secure"                "status"
## [59] "subject"               "to_field"
## [61] "vdkvgwkey"             "markings"
## [63] "body"                  "raw_body"
## [65] "nara_markings"         "type"
## [67] "format"                "from_field"
## [69] "class"                 "id"
## [71] "cable_type"            "source_path"
## [73] "body_markup"           "collection"
## [75] "title"                 "pdf"
## [77] "classification"        "composite_index"
## [79] "is_historic"           "frus_match"

##          Field          Type Null Key Default Extra
## 1          doc_nbr  varchar(30) YES      <NA>
## 2      auto_decaption  varchar(16) YES      <NA>
## 3          reference      text YES      <NA>
## 4      capture_date      date YES      <NA>
## 5          channel  varchar(32) YES      <NA>
## 6          concepts      text YES      <NA>
## 7      control_nbr  varchar(32) YES      <NA>
## 8          copy  varchar(32) YES      <NA>
## 9          date      date YES MUL      <NA>
## 10      decaption_date      text YES      <NA>
## 11      decaption_note  varchar(255) YES      <NA>
## 12      disp_action  varchar(32) YES      <NA>
## 13  disp_approved_on_date      date YES      <NA>
## 14      disp_case  varchar(32) YES      <NA>
## 15      disp_comment      text YES      <NA>
## 16      disp_date      date YES      <NA>
## 17      disp_event  varchar(8) YES      <NA>
## 18      disp_history      text YES      <NA>
## 19      disp_reason  varchar(32) YES      <NA>
## 20      disp_remarks  varchar(8) YES      <NA>
## 21      doc_source  varchar(16) YES      <NA>
## 22      drafter  varchar(64) YES      <NA>
## 23      enclosure      text YES      <NA>
## 24          eo  varchar(256) YES      <NA>
## 25      errors  varchar(32) YES      <NA>
## 26      expiration      date YES      <NA>
## 27          film  varchar(64) YES      <NA>
## 28      handling  varchar(32) YES      <NA>
## 29      isecure      int(11) YES      <NA>
## 30      legacy_key  varchar(128) YES      <NA>
## 31      line_count      int(11) YES      <NA>
## 32      litigationhistory      text YES      <NA>
## 33      locator  varchar(128) YES      <NA>
## 34      messageid  varchar(64) YES      <NA>

```

## 35	office	varchar(32)	YES	<NA>
## 36	origclass	varchar(32)	YES	<NA>
## 37	orighand	varchar(32)	YES	<NA>
## 38	origpclass	varchar(32)	YES	<NA>
## 39	origphand	varchar(32)	YES	<NA>
## 40	page_count	int(11)	YES	<NA>
## 41	pchannel	varchar(32)	YES	<NA>
## 42	pclass	text	YES	<NA>
## 43	phandling	text	YES	<NA>
## 44	retention	int(11)	YES	<NA>
## 45	review_action	text	YES	<NA>
## 46	review_content_flags	text	YES	<NA>
## 47	review_date	text	YES	<NA>
## 48	review_event	text	YES	<NA>
## 49	review_exemptions	text	YES	<NA>
## 50	review_media_id	text	YES	<NA>
## 51	review_release_date	text	YES	<NA>
## 52	review_release_event	text	YES	<NA>
## 53	review_transfer_date	text	YES	<NA>
## 54	review_withdrawn_fields	text	YES	<NA>
## 55	review_markings	text	YES	<NA>
## 56	sasid	int(11)	YES	<NA>
## 57	secure	varchar(32)	YES	<NA>
## 58	status	text	YES	<NA>
## 59	subject	text	YES	<NA>
## 60	to_field	text	YES	<NA>
## 61	vdkgwkey	text	YES	<NA>
## 62	markings	text	YES	<NA>
## 63	body	longtext	YES	<NA>
## 64	raw_body	longtext	YES	<NA>
## 65	nara_markings	text	YES	<NA>
## 66	type	varchar(16)	YES	<NA>
## 67	format	varchar(16)	YES	<NA>
## 68	from_field	text	YES	<NA>
## 69	class	text	YES	<NA>
## 70	id	varchar(128)	NO PRI	<NA>
## 71	cable_type	text	YES	<NA>
## 72	source_path	varchar(128)	YES	<NA>
## 73	body_markup	text	YES	<NA>
## 74	collection	varchar(16)	YES	<NA>
## 75	title	text	YES	<NA>
## 76	pdf	text	YES	<NA>
## 77	classification	varchar(32)	YES	<NA>
## 78	composite_index	int(11)	YES	0
## 79	is_historic	int(1)	YES	0
## 80	frus_match	varchar(35)	YES	<NA>

Examine the different country codes across datasets

```
#Re-connect to the database
driver = dbDriver("MySQL")
connection = dbConnect(driver,host='history-lab.org', password='XreadF403', user='de_reader')
mydb = dbConnect(driver,host='history-lab.org', password='XreadF403', user='de_reader', dbname='declass')
```

```

countries<-
  tbl(mydb, 'countries') %>%
  collect() #Note that there is no tag for "South Vietnam" but tag "VM" (id: 557) for "Vietnam" accordin

countries2<-
  tbl(mydb, 'countries') %>%
  collect() %>%
  mutate(country_id=as.integer(id)) %>%
  dplyr::select(-id)

cow<-read_csv("./data_comparison/cow/COW country codes.csv") %>%
  distinct()

## Parsed with column specification:
## cols(
##   StateAbb = col_character(),
##   CCode = col_integer(),
##   StateNme = col_character()
## )

#Merge ISO_3166_1 and ISO_3166_3 (ISO country codes for withdrawn countries). Note that this list often

iso_3166<-
  tibble::as_tibble(left_join(ISO_3166_1, ISO_3166_3, by = c("Alpha_3","Numeric","Name")))%>%
  mutate(Numeric=as.integer(Numeric)) %>%
  dplyr::select("Alpha_3",
    "Numeric",
    "Name",
    "Official_name",
    "Common_name")

#Check whether the variable "country_id" in the table "countries" is from ISO 3166.
#Derive COW country codes from the variable "name" in the table "countries."
countries2$cow_ccode<-countrycode(countries2$name, 'country.name', 'cown') #cow_ccode for Vietnam should

## Warning in countrycode(countries2$name, "country.name", "cown"): Some values were not matched unambig
countries2$coid2<-countrycode(countries2$country_id, 'iso3n', 'cown')
countries2$iso3n<-countrycode(countries2$name, 'country.name', 'iso3n') #iso3n for South Vietnam should

## Warning in countrycode(countries2$name, "country.name", "iso3n"): Some values were not matched unamb
## Warning in countrycode(countries2$name, "country.name", "iso3n"): Some strings were matched more than

all(countries2$country_id %in% iso_3166$Numeric)
all(iso_3166$Numeric %in% countries2$country_id)
setdiff(countries2$country_id, iso_3166$Numeric)
countries2[countries2$country_id %in% setdiff(countries2$country_id, iso_3166$Numeric),]
#Most of the items with a discrepancy between the database's country ids and iso-3166 numeric seem to b

#Focus on countries by dropping the observations with missing tag_id or COW country code.
countries2<-
  countries2[!is.na(countries2$cow_ccode) & !is.na(countries2$tag_id),] %>%
  mutate(cow_ccode= replace(cow_ccode, name=="Vietnam", 816)) %>% #Fix cow_ccode for Vietnam
  mutate(cow_ccode= replace(cow_ccode, name=="West Germany", 260)) %>% #Fix cow_ccode for West Germany
  left_join(cow, by=c("cow_ccode"="CCode")) %>%

```

```

rename(country_name=name,
        cow_stateabb=StateAbb,
        cow_statename=StateNme) %>%
dplyr::select(-iso3n)
#Germany and West Germany share their COW country codes.

```

Cable Traffic

Download, save, or load the tables for tags and docs (doc_id and date) in the working directory and count the number of cables tagged for each country

```

#Re-connect to the database
#driver = dbDriver("MySQL")
#connection = dbConnect(driver,host='history-lab.org', password='XreadF403', user='de_reader')
#mydb = dbConnect(driver,host='history-lab.org', password='XreadF403', user='de_reader', dbname='declass')

tags<-
tbl(mydb, 'tags') %>%
dplyr::select(id, tag, category) #%>% collect()

tag_doc<-
tbl(mydb, 'tag_doc') #%>% collect()

doc_date2<-
tbl(mydb, 'docs') %>%
dplyr::select(id, date) %>%
rename(doc_id=id) #%>% collect()

#tag_doc2<-
# tag_doc%>%
# inner_join(tags, by = c("tag_id"="id")) %>%
# inner_join(doc_date2, by = "doc_id") %>%
# mutate(year=lubridate::year(date),
#         month=lubridate::month(date),
#         date=lubridate::ymd(date),
#         ym=as.yearmon(paste(year, month),"%Y %m")
#         ) %>%
# collect()
#save(tag_doc2, file = "tag_doc2.RData")
load("tag_doc2.RData")

#tag_doc2_country <-
# tag_doc2 %>%
# inner_join(countries2, by="tag_id")
#save(tag_doc2_country, file = "tag_doc2_country.RData")
load("tag_doc2_country.RData")

#cable_n_country_date<-
# tag_doc2_country %>%

```

```

# group_by(.dots=c("tag_id",
#                  "country_id",
#                  "cow_ccode",
#                  "country_name",
#                  "date")) %>%
# tally()
#save(cable_n_country_date, file = "cable_n_country_date.RData")
load("cable_n_country_date.RData")

## # A tibble: 1 x 8
##   country_name deleted official tag_id country_id cow_ccode cow_stateabb
##   <chr>          <int>    <int> <int>      <int>      <dbl> <chr>
## 1 China          0        1   386        156        710. CHN
## # ... with 1 more variable: cow_statename <chr>

## # A tibble: 2 x 8
##   country_name deleted official tag_id country_id cow_ccode cow_stateabb
##   <chr>          <int>    <int> <int>      <int>      <dbl> <chr>
## 1 North Korea    0        1   453        408        731. PRK
## 2 South Korea    0        1   454        410        732. ROK
## # ... with 1 more variable: cow_statename <chr>

## # A tibble: 1 x 8
##   country_name deleted official tag_id country_id cow_ccode cow_stateabb
##   <chr>          <int>    <int> <int>      <int>      <dbl> <chr>
## 1 Vietnam        0        1   557        704        816. DRV
## # ... with 1 more variable: cow_statename <chr>

## # A tibble: 1 x 8
##   country_name deleted official tag_id country_id cow_ccode cow_stateabb
##   <chr>          <int>    <int> <int>      <int>      <dbl> <chr>
## 1 Afghanistan    0        1   342         4        700. AFG
## # ... with 1 more variable: cow_statename <chr>

## # A tibble: 1 x 8
##   country_name deleted official tag_id country_id cow_ccode cow_stateabb
##   <chr>          <int>    <int> <int>      <int>      <dbl> <chr>
## 1 Iran            0        1   440        364        630. IRN
## # ... with 1 more variable: cow_statename <chr>

## # A tibble: 3 x 8
##   country_name deleted official tag_id country_id cow_ccode cow_stateabb
##   <chr>          <int>    <int> <int>      <int>      <dbl> <chr>
## 1 Germany          0        1  1283        276        255. GMY
## 2 East Germany     1        0   419        278        265. GDR
## 3 West Germany     1        0   418        280        260. GFR
## # ... with 1 more variable: cow_statename <chr>

## # A tibble: 1 x 8
##   country_name deleted official tag_id country_id cow_ccode cow_stateabb
##   <chr>          <int>    <int> <int>      <int>      <dbl> <chr>
## 1 Egypt            0        1   402        818        651. EGY
## # ... with 1 more variable: cow_statename <chr>

```


Import and load the data for U.S. diplomatic representation

```
us_dip_rep<-
  read_csv("./data_comparison/moyeretal2016/Pardee Center Diplomatic Representation_COW 20190208.csv", )
# read_excel("./data_comparison/moyeretal2016/Diplomatic_Exchange_V3.16.16.xlsx") %>% #Non-compatible
  mutate(Year=as.numeric(Year)) %>%
  rename(year = Year) %>%
  filter(Country=="United States of America" & year>1969 & year<1981) %>%
# filter(Country=="United States of America") %>%
  mutate(date = ymd(paste0(year, "-", 06, "-", 01))) %>% #us_dip_rep[is.na(us_dip_rep$cow_ccode),] COW
  left_join(countries2, by=c("Destination"="cow_statename"))

## Parsed with column specification:
## cols(
##   Destination = col_character(),
##   Country = col_character(),
##   Year = col_integer(),
##   `Embassy Old` = col_integer(),
##   `Focus Old` = col_integer(),
##   `Embassy New` = col_integer(),
##   `Focus New` = col_integer(),
##   LOR = col_double(),
##   Location = col_character()
## )
```

Import and load the data on U.S. diplomatic visits

```
#Save the dta file as a csv file
#write.csv(read.dta("diplomatic_core.replication.dta"),"./data_comparison/lebovic_saunders_2016/diploma

#Note that the paper (LEBOVIC AND SAUNDERS 2016) mentions the variable for crisis (Crisis Shocks from t

us_dip_vis<-
  read_csv("./data_comparison/lebovic_saunders_2016/diplomatic_core.replication.csv") %>%
  filter(year>1969 & year<1981) %>%
  dplyr::select(cowid, year, bi_PRE, bi_SOS, mil_ratio, USmilaid, allies, USdefense, USdefense_EUR, USt
  left_join(cow, by=c("cowid"="CCode")) %>%
  mutate(un_member= ifelse(!is.na(UNpart),1, NA)) #Check whether newly admitted states participate in U

## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   mil_ratio = col_double(),
##   mil_ex = col_double(),
##   USmilaid = col_double(),
##   UStrade = col_double(),
##   energypc = col_double(),
##   gdppc = col_double(),
##   USalign = col_double(),
##   UNpart = col_double(),
##   energypc_TRU = col_double(),
##   gdppc_CLI = col_double(),
```

```
##   gdppc_GWB = col_double(),
##   gdppc_OBA = col_double(),
##   Rmil_rat = col_double(),
##   Rmil_rat_SOS_L1 = col_double(),
##   Rmil_rat_PRE_L1 = col_double(),
##   Rmil_rat_SOS_L14 = col_double(),
##   Rmilaid = col_double(),
##   Rmilaid_SOS_L1 = col_double(),
##   Rmilaid_PRE_L1 = col_double(),
##   Rmilaid_SOS_L14 = col_double()
##   # ... with 4 more columns
## )

## See spec(...) for full column specifications.
#a<-us_dip_vis[us_dip_vis$UNpart==0,]
```

Import and load the data on US diplomatic events

```
us_dip_evt<-
  read_csv("./data_comparison/carter2018/AmericanDiplomacyDataset.csv")

## Warning: Missing column names filled in: 'X1' [1]
## Parsed with column specification:
## cols(
##   X1 = col_integer(),
##   date = col_date(format = ""),
##   source = col_character(),
##   target = col_character(),
##   code = col_integer(),
##   label = col_character(),
##   quote = col_character(),
##   topic = col_integer(),
##   scale = col_double(),
##   vcoop = col_integer(),
##   mcoop = col_integer(),
##   vcon = col_integer(),
##   mcon = col_integer(),
##   threat = col_integer(),
##   vconeconomic = col_integer(),
##   mconeconomic = col_integer(),
##   year = col_integer()
## )

#Check whether the country abbreviations are from COW or ISO-3166.
all(us_dip_evt$source %in% countries2$cow_stateabb)
all(us_dip_evt$target %in% countries2$cow_stateabb)
all(us_dip_evt$source %in% iso_3166$Alpha_3)
all(us_dip_evt$target %in% iso_3166$Alpha_3)

setdiff(us_dip_evt$source, countries2$cow_stateabb)
setdiff(us_dip_evt$target, countries2$cow_stateabb)
setdiff(us_dip_evt$source, iso_3166$Alpha_3)
setdiff(us_dip_evt$target, iso_3166$Alpha_3)
```

```

all(setdiff(us_dip_evt$target, iso_3166$Alpha_3) %in% us_dip_evt$target)
all(setdiff(us_dip_evt$source, iso_3166$Alpha_3) %in% us_dip_evt$source)
all(setdiff(us_dip_evt$target, iso_3166$Alpha_3) %in% iso_3166$Alpha_3)
all(setdiff(us_dip_evt$source, iso_3166$Alpha_3) %in% iso_3166$Alpha_3)

us_dip_evt[us_dip_evt$target %in% setdiff(us_dip_evt$target, iso_3166$Alpha_3),]
us_dip_evt[us_dip_evt$source %in% setdiff(us_dip_evt$source, iso_3166$Alpha_3),]

#It looks like the variables "target" and "source" in the dataset are "the 2-character FIPS10-4 country

us_dip_evt2<-
  read_csv("./data_comparison/carter2018/AmericanDiplomacyDataset.csv") %>%
  # filter(1969<year & year<1980) %>%
  mutate(ym=as.yearmon(date, "%Y %m"),
         cow_ccode_source=countrycode(source, 'iso3c', 'cown'),
         cow_ccode_target=countrycode(target, 'iso3c', 'cown')) %>%
  mutate(cow_ccode_source= replace(cow_ccode_source, source=="YUG", 345), #The function countrycode did
         cow_ccode_target = replace(cow_ccode_target, target=="YUG", 345), #The function countrycode did
         cow_ccode_source= replace(cow_ccode_source, source=="SER", 345), #The function countrycode did
         cow_ccode_target = replace(cow_ccode_target, target=="SER", 345)) #The function countrycode did

## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:
## cols(
##   X1 = col_integer(),
##   date = col_date(format = ""),
##   source = col_character(),
##   target = col_character(),
##   code = col_integer(),
##   label = col_character(),
##   quote = col_character(),
##   topic = col_integer(),
##   scale = col_double(),
##   vcoop = col_integer(),
##   mcoop = col_integer(),
##   vcon = col_integer(),
##   mcon = col_integer(),
##   threat = col_integer(),
##   vconeconomic = col_integer(),
##   mconeconomic = col_integer(),
##   year = col_integer()
## )

## Warning in countrycode(source, "iso3c", "cown"): Some values were not matched unambiguously: AUH, BM
## Warning in countrycode(target, "iso3c", "cown"): Some values were not matched unambiguously: AUH, BM

#Typos in the dataset
#Make empathetic comment

#Entities with missing COW country codes are non-state. Let's drop or fix them.
table(us_dip_evt[is.na(us_dip_evt$cow_ccode_source),]$source)

```

```
## Warning: Unknown or uninitialised column: 'cow_ccode_source'.
## Warning in is.na(us_dip_evt$cow_ccode_source): is.na() applied to non-(list
## or vector) of type 'NULL'
## Warning: Length of logical index must be 1 or 48518, not 0
table(us_dip_evt[is.na(us_dip_evt$cow_ccode_target),]$target)

## Warning: Unknown or uninitialised column: 'cow_ccode_target'.
## Warning in is.na(us_dip_evt$cow_ccode_target): is.na() applied to non-(list
## or vector) of type 'NULL'
## Warning: Length of logical index must be 1 or 48518, not 0
#AUH: Abu Dhabi before it became part of UAE in December 1971.
#BMU: Bermuda
#CYM: Cayman Islands
#HKG: Hong Kong before it became part of China in 1997
#PSE: Palestine
#SER: Yugoslavia? (https://wits.worldbank.org/wits/wits/witshelp/content/codes/country\_codes.htm)
#TBT: Tibet (Dalai Lama visited the US on Sep.3-Oct.21, 1979 (https://www.dalailama.com/the-dalai-lama/
#TMP: East Timor (https://wits.worldbank.org/wits/wits/witshelp/content/codes/country\_codes.htm)
#VMN: a typo for VNM (Vietnam)?
#YUG: Transitional reservations for former Yugoslavia/Serbia and Montenegro (https://en.wikipedia.org/w
```

Import and load the data on international crises

```
icb<-
  read_csv("./data_comparison/icb/icb2v12.csv") %>%
#Rule out crises that end before 1970 and crises that start after 1979
  filter(yrterm>1969 & systrgyr<1980) %>%
  mutate(ym_term=as.yearmon(paste(yrterm, moterm),"%Y %m"),
         ym_trg=as.yearmon(paste(systrgyr, systrgmo), "%Y %m"),
         duration=(ym_term-ym_trg)*12) %>% #Some crises are missing the day of occurence but most have

## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   icb2 = col_character(),
##   actor = col_character(),
##   crisname = col_character()
## )

## See spec(...) for full column specifications.
#Note that the longest duration of a crisis (from a trigger event to its termination) is 3 years.
#max(icb$yrterm-icb$systrgyr, na.rm=TRUE)
```

Compare the monthly cable traffic with the data on international crises

Compare the yearly cable traffic with the country-year data on diplomatic visits

```
cable_n_vis_country_y<-
  cable_n_country_date %>%
  mutate(year=lubridate::year(date)) %>%
  group_by(year, cow_ccode, country_name) %>%
  summarise(y_n = sum(n)) %>%
  rename(cowid=cow_ccode) %>%
  inner_join(us_dip_vis, by=c("year", "cowid"))

glm.nb_pre1 <- glm.nb(y_n~bi_PRE, data = cable_n_vis_country_y)
glm.nb_sos1 <- glm.nb(y_n~bi_SOS, data = cable_n_vis_country_y)

#Slightly modified from the analysis in Lebovic and Saunders (2016)' Tables 3 and 4
glm.nb_pre2 <- glm.nb(y_n~bi_PRE+mil_ratio+USmilaid+allies+USdefense+USdefense_EUR+UStrade+energypc+USa
#Cluster by cowid

glm.nb_sos2 <- glm.nb(y_n~bi_SOS+mil_ratio+USmilaid+allies+USdefense+USdefense_EUR+UStrade+energypc+USa
#Cluster by cowid

glm.nb_pre3 <- glm.nb(y_n~bi_PRE+mil_ratio+USmilaid+allies+USdefense+USdefense_EUR+UStrade+energypc+USa
#Cluster by cowid

glm.nb_sos3 <- glm.nb(y_n~bi_SOS+mil_ratio+USmilaid+allies+USdefense+USdefense_EUR+UStrade+energypc+USa
#Cluster by cowid

#Check model assumptions and robustness (https://stats.idre.ucla.edu/r/dae/negative-binomial-regression,
#Descriptive statistics
stargazer(as.data.frame(cable_n_vis_country_y),
  type = "html",
  title="Descriptive statistics",
  digits=1,
  out="desc_cable_n_vis_country_year.html"#,
  #covariate.labels=c("Miles/(US)gallon",)
)

#Negative binominal regression models
stargazer(glm.nb_pre1, glm.nb_pre2, glm.nb_pre3,
  type="html",
  dep.var.labels=c("Number of Cables Tagged")#,
  #covariate.labels=c("Gross horsepower", "Rear axle ratio", "Four foward gears", "Five forward ge
  #out="cable_n_pres_visit.html"
)

stargazer(glm.nb_sos1, glm.nb_sos2, glm.nb_sos3,
  type="html",
```

```

    dep.var.labels=c("Number of Cables Tagged") #,
    #covariate.labels=c("Gross horsepower", "Rear axle ratio", "Four foward gears", "Five forward ge
    #out="cable_n_sos_visit.html"
  )

stargazer(glm.nb_pre1, glm.nb_pre2, glm.nb_pre3, glm.nb_sos1, glm.nb_sos2, glm.nb_sos3,
  type="html",
  dep.var.labels=c("Number of Cables Tagged"),
  #covariate.labels=c("Gross horsepower", "Rear axle ratio", "Four foward gears", "Five forward ge
  out="cable_n_visit.html")

```