tjbrailey\_wrangle\_data

POLI 191A/B

Thomas Brailey

13/09/2019

# Load Data

# set working directory.   
wd <- paste0(here::here(), '/data/')   
  
# parse out .xlsx worksheets   
files <- list.files(paste0(wd, 'PSED\_agreement.xlsx'))  
files <- files[]  
  
read\_excel\_allsheets <- function(filename) {   
 sheets <- readxl::excel\_sheets(filename)   
 x <- lapply(sheets, function(x) readxl::read\_excel(filename, sheet = x))   
 names(x) <- sheets   
 x   
}   
  
out <- lapply(paste0(here::here(), '/data/PSED\_agreement.xlsx'), read\_excel\_allsheets)  
basename(files)

## character(0)

# install datasets.  
# power-sharing-specific datasets.  
#psed\_prom <- out[[1]]$PSED\_agreement\_promises  
#psed\_prac <- out[[1]]$PSED\_agreement\_practices  
idc <- rio::import(paste0(wd, 'IDC\_country-year\_v1\_0.RData'))  
#impact <- rio::import(paste0(wd,'c\_656154-l\_1-k\_impact--version2.0.csv'))  
dtd <- rio::import(paste0(wd,'Democracy Timeseries Data January 2009 Excel2007.csv'))  
#epr <- rio::import(paste0(wd,'data-epr\_countryyear.csv'))  
#cah\_pshare\_and\_dem <- rio::import(paste0(wd,'pshare\_and\_democracy\_for\_world\_politics\_publication.dta'))  
#cah\_craft\_peace <- rio::import(paste0(wd,'book\_project\_49\_cases\_long\_time\_version\_feb\_2004.dta'))  
#bumba <- rio::import(paste0(wd,'Bumba(peaceduration).RData'))  
pax <- rio::import(paste0(wd,'pax\_20\_02\_2018\_1\_CSV.csv'))  
#di <- rio::import(paste0(wd,'Diplomatic Interventions data.dta'))  
#dme <- rio::import(paste0(wd,'DME data.dta'))  
  
# general datasets (the y value).  
vdem <- rio::import(paste0(wd,"V-Dem-CY-Full+Others-v9.csv"))  
dpi <- rio::import(paste0(wd,'DPI2012.xls'))  
#qog\_cs <- rio::import(paste0(wd,'qog\_std\_cs\_jan19.csv'))  
qog\_ts <- rio::import(paste0(wd,'qog\_std\_ts\_jan19.csv'))  
polityiv <- rio::import(paste0(wd,'p4v2017.xls'))  
ucdp <- rio::import(paste0(wd,'ucdp-prio-acd-191.xlsx'))  
wvs <- rio::import(paste0(wd, "F00008390-WVS\_Longitudinal\_1981\_2016\_r\_v20180912.rds"))  
wvs <- dplyr::as\_tibble(wvs)  
  
# clean workspace.  
rm(out, files, read\_excel\_allsheets)

# Set Baseline Data

tb <- idc  
  
tb$country[tb$country == "Cent. Af. Rep."] <- "Central African Republic"  
tb$country[tb$country == "Dom. Rep."] <- "Dominican Republic"  
tb$country[tb$country == "GDR"] <- "German Democratic Republic"  
tb$country[tb$country == "PRC"] <- "China"  
tb$country[tb$country == "ROK"] <- "South Korea"  
tb$country[tb$country == "S. Africa"] <- "South Africa"  
tb$country[tb$country == "Serbia and Montenegro"] <- "Montenegro"  
  
tb <- tb[!is.na(tb$country) & !is.na(tb$ifs),]  
  
tb$cowc <- countrycode::countrycode(tb$country, 'country.name', 'cowc')

## Warning in countrycode::countrycode(tb$country, "country.name", "cowc"): Some values were not matched unambiguously: Serbia

tb$cown <- countrycode::countrycode(tb$country, 'country.name', 'cown')

## Warning in countrycode::countrycode(tb$country, "country.name", "cown"): Some values were not matched unambiguously: Serbia

# Some values were not matched unambiguously: Serbia  
  
tb <- tb %>%  
 dplyr::select(country,   
 cowc,   
 cown,   
 year,   
 gwno,   
 ifs,   
 mveto,  
 gcman,  
 gcimp,  
 auton,  
 jrevman,  
 relconstd,  
 relconstp,  
 milleg,  
 partynoethnic,  
 jtenure,  
 jconst,  
 gcseats1,  
 gcseats2,  
 gcseats3,  
 unity,  
 resman,  
 resseats,  
 resseats2,  
 resseatsimp,  
 miman,  
 subtax,  
 subed,  
 subpolice,  
 state,  
 muni  
 )

## DTD (Norris, 2008)

dtd\_psp\_sub <- dtd %>%  
 dplyr::select(Nation,  
 Year,  
 Natcode,  
 NatWVS,  
 Natmap,  
 Natabrv,  
 Coalition,  
 coalition2,  
 coalitiongov,  
 VANstand,  
 Admin,  
 mixed,  
 proportional,  
 roseprop,  
 xrcomp  
 ) %>%  
 dplyr::rename(country = Nation,  
 year = Year)  
  
# Some values were not matched unambiguously:  
  
dtd\_psp\_sub$country <- countrycode::countrycode(dtd\_psp\_sub$country, 'country.name', 'country.name')

## Warning in countrycode::countrycode(dtd\_psp\_sub$country, "country.name", : Some values were not matched unambiguously:

dtd\_psp\_sub$cowc <- countrycode::countrycode(dtd\_psp\_sub$country, 'country.name', 'cowc')  
dtd\_psp\_sub$cown <- countrycode::countrycode(dtd\_psp\_sub$country, 'country.name', 'cown')   
  
dtd\_psp\_sub <- dtd\_psp\_sub %>%  
 dplyr::select(-country)

## PAX (Bell, C. and Badanjak, S., 2019)

pax\_psp\_sub <- pax %>%  
 dplyr::select(Con,  
 Dat,  
 Agt,  
 PpsAut,  
 TpsAut,  
 PpsEx,  
 CenBan,  
 PpsOro,  
 PpsOthPr,  
 StRef  
 ) %>%  
 dplyr::rename(country = Con,  
 year = Dat) %>%  
 dplyr::mutate(year = as.numeric(stringr::str\_sub(year, start= -4))) %>%  
 dplyr::group\_by(PpsAut,  
 TpsAut,  
 PpsEx,  
 CenBan,  
 PpsOro,  
 PpsOthPr,  
 StRef,   
 country, year) %>%  
 dplyr::summarise(Agt = paste(Agt, collapse=", ")) %>%   
 dplyr::ungroup() %>%  
 dplyr::mutate(country = sub("\\/.\*", "", country)) %>%  
 dplyr::select(-Agt)  
  
# Some values were not matched unambiguously: Kurds-Kurdistan, United Nations  
# Some values were not matched unambiguously: Palestinian Territories, Serbia  
# Some values were not matched unambiguously: Palestinian Territories, Serbia  
  
pax\_psp\_sub$country <- countrycode::countrycode(pax\_psp\_sub$country, 'country.name', 'country.name')

## Warning in countrycode::countrycode(pax\_psp\_sub$country, "country.name", : Some values were not matched unambiguously: Kurds-Kurdistan, United Nations

pax\_psp\_sub$cown <- countrycode::countrycode(pax\_psp\_sub$country, 'country.name', 'cown')

## Warning in countrycode::countrycode(pax\_psp\_sub$country, "country.name", : Some values were not matched unambiguously: Palestinian Territories, Serbia

pax\_psp\_sub$cowc <- countrycode::countrycode(pax\_psp\_sub$country, 'country.name', 'cowc')

## Warning in countrycode::countrycode(pax\_psp\_sub$country, "country.name", : Some values were not matched unambiguously: Palestinian Territories, Serbia

pax\_psp\_sub <- pax\_psp\_sub %>%  
 dplyr::select(-country)

## V-Dem

vdem\_psp\_sub <- vdem %>%  
 dplyr::select(country\_name, country\_id, country\_text\_id, year, e\_miinterc, e\_civil\_war) %>%   
 dplyr::rename(country = country\_name)  
vdem\_psp\_sub$country <- countrycode::countrycode(vdem\_psp\_sub$country, 'country.name', 'country.name')

## Warning in countrycode::countrycode(vdem\_psp\_sub$country, "country.name", : Some values were not matched unambiguously: Republic of Vietnam, WÃ¼rtemberg

vdem\_psp\_sub$cown <- countrycode::countrycode(vdem\_psp\_sub$country, 'country.name', 'cown')

## Warning in countrycode::countrycode(vdem\_psp\_sub$country, "country.name", : Some values were not matched unambiguously: Brunswick, Hamburg, Hesse-Darmstadt, Hesse-Kassel, Hong Kong SAR China, Nassau, Oldenburg, Palestinian Territories, Piedmont-Sardinia, Saxe-Weimar-Eisenach, Serbia

vdem\_psp\_sub$cowc <- countrycode::countrycode(vdem\_psp\_sub$country, 'country.name', 'cowc')

## Warning in countrycode::countrycode(vdem\_psp\_sub$country, "country.name", : Some values were not matched unambiguously: Brunswick, Hamburg, Hesse-Darmstadt, Hesse-Kassel, Hong Kong SAR China, Nassau, Oldenburg, Palestinian Territories, Piedmont-Sardinia, Saxe-Weimar-Eisenach, Serbia

vdem\_psp\_sub <- vdem\_psp\_sub %>%  
 dplyr::select(-country)

## QoG

qog\_ts\_psp\_sub <- qog\_ts %>%  
 dplyr::select(ccode,   
 cname,   
 year,   
 fe\_etfra,  
 iaep\_ebbp,  
 gle\_gdp,  
 bti\_ci,  
 cspf\_sfi,  
 gtm\_unit,  
 ccp\_hr,  
 ffp\_hr,  
 iiag\_phr,  
 dpi\_housesys,  
 dpi\_sensys,  
 jw\_bicameral,  
 bti\_ig,  
 vdem\_partipdem,  
 iaep\_nr,  
 bti\_sop,  
 gol\_est,  
 gol\_mt,  
 iaep\_es,  
 no\_ef,  
 no\_ce,  
 iaep\_eccdt,  
 iaep\_ecdl,  
 iaep\_eml,  
 iaep\_epmf,  
 iaep\_evp,  
 iaep\_lcre,  
 iaep\_lego,  
 iaep\_lrit,  
 wbgi\_pve  
 ) %>%  
 dplyr::rename(country = cname)  
qog\_ts\_psp\_sub$country[qog\_ts\_psp\_sub$country == "Micronesia"] <- "Federated States of Micronesia"  
qog\_ts\_psp\_sub$country[qog\_ts\_psp\_sub$country == "Serbia and Montenegro"] <- "Montenegro"  
qog\_ts\_psp\_sub$country <- countrycode::countrycode(qog\_ts\_psp\_sub$country, 'country.name', 'country.name')

## Warning in countrycode::countrycode(qog\_ts\_psp\_sub$country, "country.name", : Some values were not matched unambiguously: Tibet

qog\_ts\_psp\_sub$cown <- countrycode::countrycode(qog\_ts\_psp\_sub$country, 'country.name', 'cown')

## Warning in countrycode::countrycode(qog\_ts\_psp\_sub$country, "country.name", : Some values were not matched unambiguously: Serbia

qog\_ts\_psp\_sub$cowc <- countrycode::countrycode(qog\_ts\_psp\_sub$country, 'country.name', 'cowc')

## Warning in countrycode::countrycode(qog\_ts\_psp\_sub$country, "country.name", : Some values were not matched unambiguously: Serbia

qog\_ts\_psp\_sub <- qog\_ts\_psp\_sub %>%  
 dplyr::select(-country)

## DPI

dpi\_psp\_sub <- dpi %>%  
 dplyr::select(countryname,   
 year,  
 system,  
 pr,  
 sensys,  
 eiec  
 ) %>%  
 dplyr::rename(country = countryname) %>%  
 dplyr::mutate(year = as.numeric(year))  
dpi\_psp\_sub$country[dpi\_psp\_sub$country == "Cent. Af. Rep."] <- "Central African Republic"  
dpi\_psp\_sub$country[dpi\_psp\_sub$country == "Dom. Rep."] <- "Dominican Republic"  
dpi\_psp\_sub$country[dpi\_psp\_sub$country == "GDR"] <- "German Democratic Republic"  
dpi\_psp\_sub$country[dpi\_psp\_sub$country == "PRC"] <- "China"  
dpi\_psp\_sub$country[dpi\_psp\_sub$country == "PRK"] <- "North Korea"  
dpi\_psp\_sub$country[dpi\_psp\_sub$country == "ROK"] <- "South Korea"  
dpi\_psp\_sub$country[dpi\_psp\_sub$country == "S. Africa"] <- "South Africa"  
dpi\_psp\_sub$country <- countrycode::countrycode(dpi\_psp\_sub$country, 'country.name', 'country.name')  
dpi\_psp\_sub$cowc <- countrycode::countrycode(dpi\_psp\_sub$country, 'country.name', 'cowc')  
dpi\_psp\_sub$cown <- countrycode::countrycode(dpi\_psp\_sub$country, 'country.name', 'cown')  
  
dpi\_psp\_sub <- dpi\_psp\_sub %>%  
 dplyr::select(-country)

## World Value Survery

wvs\_psp\_sub <- wvs %>%   
 dplyr::select(S003, S020,  
 A035,  
   
 A124\_01, A124\_02,A124\_03,A124\_04,A124\_05,A124\_06,A124\_07,A124\_08,A124\_09,A124\_10,A124\_11,A124\_12,A124\_13,A124\_14,A124\_15,A124\_16,A124\_17,A124\_18,A124\_19,A124\_20,A124\_21,A124\_22,A124\_23,A124\_24,A124\_25,A124\_27,A124\_28,A124\_29,A124\_30,A124\_31,A124\_32,A124\_33,A124\_34,A124\_35,A124\_36,A124\_37,A124\_38,A124\_42,A124\_43,A124\_45,A124\_46,A124\_47,A124\_48,A124\_49,A124\_50,A124\_51,A124\_52,A124\_53,A124\_54,A124\_55,A124\_56,A124\_57,A124\_58,A124\_59,A124\_60,A124\_61,  
   
E069\_11, E069\_12,  
  
E114, E115, E116, E117,  
  
E124)  
  
wvs\_psp\_sub[wvs\_psp\_sub == -4] <- NA  
  
wvs\_psp\_sub$country <- countrycode::countrycode(wvs$S003, "iso3n", "country.name", nomatch = "unknown")  
wvs\_psp\_sub$cowc <- countrycode::countrycode(wvs$S003, "iso3n", "cowc")

## Warning in countrycode::countrycode(wvs$S003, "iso3n", "cowc"): Some values were not matched unambiguously: 101, 275, 344, 630, 688, 914

wvs\_psp\_sub$cown <- countrycode::countrycode(wvs$S003, "iso3n", "cown")

## Warning in countrycode::countrycode(wvs$S003, "iso3n", "cown"): Some values were not matched unambiguously: 101, 275, 344, 630, 688, 914

wvs\_psp\_sub <- wvs\_psp\_sub %>%  
 dplyr::select(-country)  
  
  
wvs\_n\_per\_year <- wvs\_psp\_sub %>%  
 dplyr::group\_by(S020) %>%  
 dplyr::summarize(obs=n())

## Warning: Calling `n()` without importing or prefixing it is deprecated, use `dplyr::n()`.  
## This warning is displayed once per session.

wvs\_n\_by\_country <- wvs\_psp\_sub %>%  
 dplyr::group\_by(S020, cowc) %>%  
 dplyr::summarize(states = n())

# Join Data

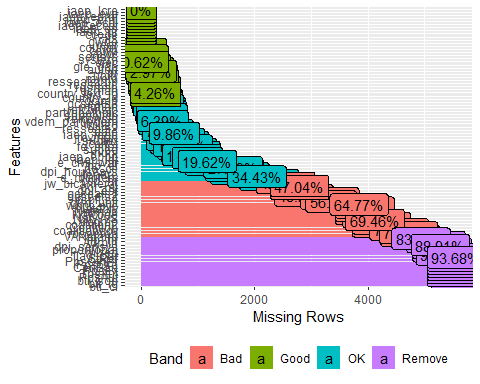
tb\_1 <- dplyr::left\_join(tb, dtd\_psp\_sub, by = c("cown", "cowc", "year"))  
tb\_2 <- dplyr::left\_join(tb\_1, qog\_ts\_psp\_sub, by = c("cown", "cowc", "year"))  
tb\_3 <- dplyr::left\_join(tb\_2, pax\_psp\_sub, by = c("cown", "cowc", "year"))  
tb\_4 <- dplyr::left\_join(tb\_3, dpi\_psp\_sub, by = c("cown", "cowc", "year"))  
tb\_5 <- dplyr::left\_join(tb\_4, vdem\_psp\_sub, by = c("cown", "cowc", "year"))  
  
# Collapse duplicate country/years whie retaining values.  
tb\_6 <- tb\_5 %>%   
 dplyr::group\_by(country, year) %>%  
 dplyr::summarise\_all(dplyr::funs(dplyr::first(na.omit(.))))

## Warning: funs() is soft deprecated as of dplyr 0.8.0  
## Please use a list of either functions or lambdas:   
##   
## # Simple named list:   
## list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`:   
## tibble::lst(mean, median)  
##   
## # Using lambdas  
## list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## This warning is displayed once per session.

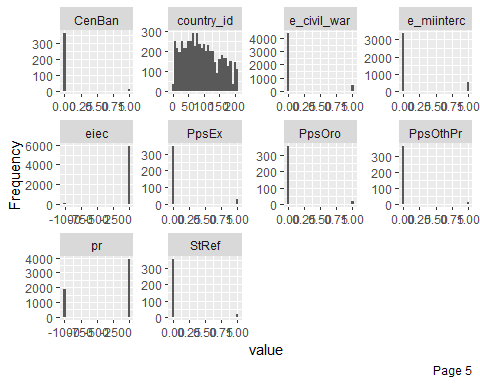
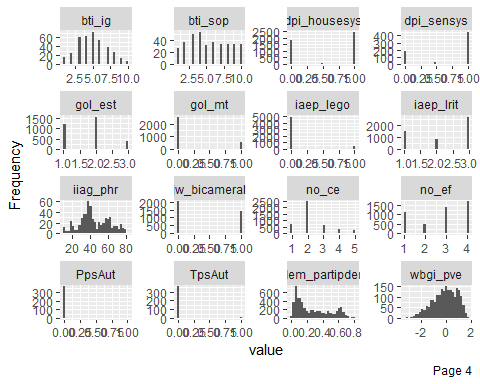
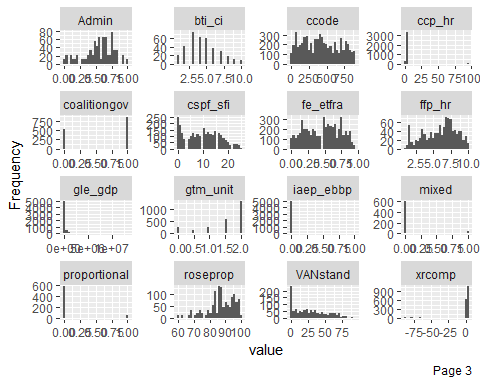
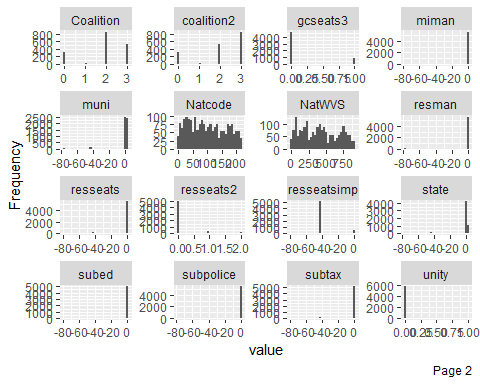
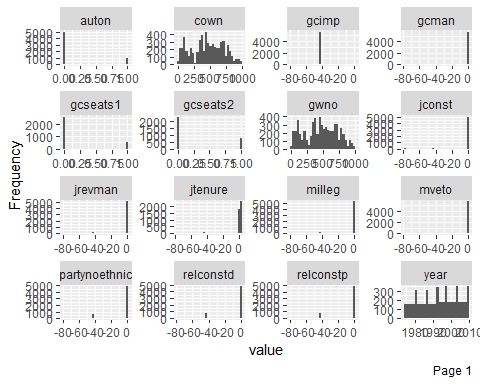
saveRDS(tb\_6, paste0(here::here(), '/data/tjbrailey\_psp.rds'))  
  
rm(tb\_1, tb\_2, tb\_3, tb\_4, tb\_5)

# Visualize Data

DataExplorer::plot\_missing(tb\_6)



DataExplorer::plot\_histogram(tb\_6)



png(file = paste0(here::here(), '/vis/tjbrailey\_psp\_datexp.png'),  
 width = 2000,  
 height = 1000)  
DataExplorer::plot\_missing(tb\_6)  
dev.off()

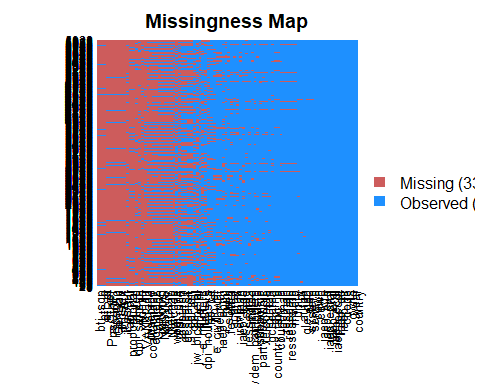
## png   
## 2

Amelia::missmap(tb\_6)

## Warning in if (class(obj) == "amelia") {: the condition has length > 1 and  
## only the first element will be used

## Warning: Unknown or uninitialised column: 'arguments'.  
  
## Warning: Unknown or uninitialised column: 'arguments'.

## Warning: Unknown or uninitialised column: 'imputations'.



png(file = paste0(here::here(), "/vis/tjbrailey\_psp\_missingness.png"),  
 width = 2000,   
 height = 1000)  
Amelia::missmap(tb\_6)

## Warning in if (class(obj) == "amelia") {: the condition has length > 1 and  
## only the first element will be used

## Warning: Unknown or uninitialised column: 'arguments'.  
  
## Warning: Unknown or uninitialised column: 'arguments'.

## Warning: Unknown or uninitialised column: 'imputations'.

dev.off()

## png   
## 2