Code Appendix

Sam Tan, Wenhao Pan

2022-12-16

```
rm(list = ls())
knitr::opts_chunk$set(eval = FALSE)
# install.packages('skimr')
# install.packages('AER')
# install.packages('fastDummies')
# install.packages('sandwich')
# install.packages('stargazer')
# install.packages('tidyverse')
# install.packages('janitor')
library(tidyverse)
library(skimr)
library(AER) # AER::ivreg()
library(fastDummies) # dummycols()
library(sandwich)
library(stargazer)
library(janitor)
```

Data Cleaning

Clean the rainfall data

```
# install.packages('tidync')
library(tidync)
library(dplyr)
library(stringr)
rain = tidync('precip.mon.mean.nc') %>%
  hyper_tibble()

### filter nodes in Africa ###
rain = rain %>%
  filter(lat >= -40 & lat <= 40) %>%
  filter(lon >= 330 | (lon >= 0 & lon <= 60))

### extract year and month ###
rain$time = rain$time - min(rain$time)
rain$date = as.Date(rain$time, origin = '1979-01-01')
rain$year = as.numeric(format(rain$date, '%Y'))
rain = rain %>%
  filter(year <= 2000)

rain$month = as.numeric(format(rain$date, '%m'))
```

```
rain = rain %>%
  select(-c('time','date'))
rain$pos = paste(as.character(rain$lon), as.character(rain$lat))

node = read.csv('mss_rainfall.csv')
node = node %>%
  rename('Lon' = 'Longitude', 'Lat' = 'Latitude')

### clean the node data ###
for(i in 1:nrow(node)) {
  # clean longitude
  lon = node[i,'Lon']
  direction = str_extract(lon, '[EW]')
  val = as.double(str_extract(lon, '^(\\d+\\.\\d+)'))
  if (direction == 'E'){
    node[i, 'Lon'] = val
  } else {
    node[i, 'Lon'] = 360 - val
  }

  # clean latitude
  lat = node[i, 'Lat']
  direction = str_extract(lat, '[NS]')
  val = as.double(str_extract(lat, '^(\\d+\\.\\d+)'))
  if (direction == 'N'){
    node[i, 'Lat'] = val
  } else {
    node[i, 'Lat'] = -val
  }
}

### integrate the rainfall and node data ###
country_list = unique(node$Country)
for(i in 1:length(country_list)){
  country = country_list[i]
  country_node = node %>%
    filter(Country == country)
  node_pos = paste(country_node$Lon, country_node$Lat)
  country_rain = rain[rain$pos %in% node_pos,]
  country_rain$country = country
  country_rain = country_rain %>%
    select(-c('lon', 'lat', 'pos'))

  if (i == 1){
    final = country_rain
  } else {
    final = rbind(final, country_rain)
  }
}

### recover GPCP ###
leap_years = c('1980','1984','1988','1992','1996','2000')
month_total = numeric(nrow(final))
```

```r
for (i in 1:nrow(final)){
  obs = final[i,]
  year = obs$year
  month = obs$month
  precip = obs$precip

  if (month == '1'){
    month_total[i] = precip * 31
  } else if ((month == '2') & (year %in% leap_years)) {
    month_total[i] = precip * 29
  } else if (month == '2') {
    month_total[i] = precip * 28
  } else if (month == '3') {
    month_total[i] = precip * 31
  } else if (month == '4') {
    month_total[i] = precip * 30
  } else if (month == '5') {
    month_total[i] = precip * 31
  } else if (month == '6') {
    month_total[i] = precip * 30
  } else if (month == '7') {
    month_total[i] = precip * 31
  } else if (month == '8') {
    month_total[i] = precip * 31
  } else if (month == '9') {
    month_total[i] = precip * 30
  } else if (month == '10') {
    month_total[i] = precip * 31
  } else if (month == '11') {
    month_total[i] = precip * 30
  } else {
    month_total[i] = precip * 31
  }
}
final$month_total = month_total
final = final %>%
  select(-c('precip','month')) %>%
  group_by(country, year) %>%
  summarise(GPCP = sum(month_total))

n_nodes = node %>%
  group_by(Country) %>%
  summarise(n = n())

for (i in 1:nrow(final)){
  obs = final[i, ]
  country = obs$country
  GPCP = obs$GPCP
  n = n_nodes[n_nodes$Country == country,]$n
  final[i,'GPCP'] = GPCP / n
}

# adjust measures for Ethiopia
```

```
final = final %>%
  filter(!((country == 'Ethiopia, pre 1993' & year >= '1993') | (country == 'Ethiopia, post 1993' & yea
final[str_detect(final$country, 'Ethiopia'),]$country = 'Ethiopia'
final = final %>%
  arrange(country, year)

### recover features derived from GPCP ###
final = final %>%
  group_by(country) %>%
  mutate(
    GPCP_L = dplyr::lag(GPCP, n=1, default=NA),
    GPCP_L2 = dplyr::lag(GPCP, n=2, default=NA),
  ) %>%
  mutate(
    GPCP_G = (GPCP - GPCP_L) / GPCP_L,
    GPCP_G_L = (GPCP_L - GPCP_L2) / GPCP_L2,
    GPCP_G_FL = dplyr::lead(GPCP_G, n=1, default=NA)
  )

### assign COW country code ###
COW_code = read.csv('COW-country-codes.csv')
country_list = unique(final$country)
sub = COW_code[COW_code$StateNme %in% country_list,]
sub = sub[!duplicated(sub),]
sub[nrow(sub)+1,] = c('CDI','437',"Cote d'Ivoire")
sub[nrow(sub)+1,] = c('CON','484','Congo, Brazzaville')
sub[nrow(sub)+1,] = c('DRC','490','Congo, Kinshasa')
COW_code = sub %>%
  arrange(CCode)

code = integer(nrow(final))
for (i in 1:nrow(final)){
  obs = final[i,]
  country = obs$country
  code[i] = COW_code[COW_code$StateNme == country,]$CCode
}
final$ccode = code
```

Clean the Fearon and Laitin (2003) data

```
# install.packages('haven')
library(haven)
FL = read_dta('FL.dta')
```

Clean GDNGD data

```
GDNGD = read.csv('GDNGD.csv')
COW_code$StateNme = replace(COW_code$StateNme, COW_code$StateNme == 'Congo, Brazzaville', 'Congo, Rep.'
COW_code$StateNme = replace(COW_code$StateNme, COW_code$StateNme == 'Congo, Kinshasa', 'Congo, Dem. Rep
COW_code$StateNme = replace(COW_code$StateNme, COW_code$StateNme == 'Gambia', 'Gambia, The')
code = integer(nrow(GDNGD))
for (i in 1:nrow(GDNGD)){
  obs = GDNGD[i,]
  country = obs$Country.Name

  if (country %in% COW_code$StateNme){
    code[i] = COW_code[COW_code$StateNme == country,]$CCode
  } else {
    code[i] = 'NA'
  }
}
GDNGD$ccode = code
GDNGD = GDNGD %>%
  filter(ccode != 'NA')
GDNGD = replace(GDNGD, GDNGD=='..',NA)
```

Integrate all data

```
### bring in GDNGD ###
TOT = numeric(nrow(final))
for (i in 1:nrow(final)) {
  obs = final[i,]
  if (obs$year == '2000'){
    TOT[i] = NA
  } else {
    TOT[i] = GDNGD[GDNGD$ccode == obs$ccode, paste('X',obs$year,sep='')]
  }
}
final$TOT_100 = as.double(TOT)
final = final %>%
  group_by(country) %>%
  mutate(
    TOT_100_L = dplyr::lag(TOT_100, n=1, default=NA),
  )
final$TOT_100_G = (final$TOT_100 - final$TOT_100_L) / final$TOT_100_L

### bring in the Fearon and Laitin (2003) ###
final = merge(final, FL, by=c('ccode','year'))
final = final %>%
  arrange(ccode, year)
final$GDP_G = (final$gdpen - final$gdpenl) / final$gdpenl
final = final %>%
  group_by(ccode) %>%
  mutate(
    GDP_G_L = dplyr::lag(GDP_G, n=1, default=NA),
  )
final$POLITY2L_6 = as.integer(final$polity2l >= 6)
final = final %>%
  group_by(ccode) %>%
  mutate(Y_0 = first(gdpen))
write.csv(final, 'clean_data.csv')
```

Civil Conflict

```
main = readxl::read_excel("ucdp-prio-acd-3-2005.xls")
africa = read_csv('COW_code.csv')
monadic = main %>%
  select(Type, Intensity, YEAR, COW_location, Location) %>%
  pivot_wider(id_cols = c(YEAR,COW_location), names_from = Type,values_from = Intensity, values_fn = ma

monadic_africa_conflict = monadic %>%
  filter(COW_location %in% africa$CCode) %>%
  mutate(COW_location = as.numeric(COW_location))

monadic_africa_empty = map_df(1:nrow(africa), function(i){
  # For Namibia, only 9 years are observed (1990 - 1999), instead of all years. Source: Armed Conflict 
  if (africa$StateNme[i] == 'Namibia') {
    bind_cols(Location = africa$StateNme[i],
        COW_location = africa$CCode[i],
        YEAR = 1990:1999)
  }
  else{
    bind_cols(Location = africa$StateNme[i],
        COW_location = africa$CCode[i],
        YEAR = 1980:2000) # include 1980 and 2000 for the onset and offset variable
    }
})


monadic_africa_all = left_join(
  monadic_africa_empty,
  monadic_africa_conflict,
  by = c('COW_location','YEAR'))  %>%
  mutate_if(is.numeric,coalesce,0) %>% # The default is peace. No conflict.
  mutate(WAR_PRIO = ((Type3 == 3)|(Type4 == 3)),
         WAR_PRIO_ON = ((lag(WAR_PRIO == F, default = F) & (WAR_PRIO == T))),
         WAR_PRIO_OFF = ((lag(WAR_PRIO == T, default = F) & (WAR_PRIO == F))),
         MINOR_PRIO = ((Type3 == 1)|(Type3 == 2)|(Type4 == 1) | (Type4 == 2)),
         MINOR_PRIO_ON = ((lag(MINOR_PRIO == F, default = F) & (MINOR_PRIO == T))),
         MINOR_PRIO_OFF = ((lag(MINOR_PRIO == T, default = F) & (MINOR_PRIO == F))),
         ANY_PRIO = ((Type3 == 1)|(Type3 == 2)|(Type3 == 3)|(Type4 == 1) | (Type4 == 2) | (Type4 == 3))
         ANY_PRIO_ON = (lag(ANY_PRIO == F,default = F) & (ANY_PRIO == T)),
         ANY_PRIO_OFF = ((lag(ANY_PRIO == T, default = F) & (ANY_PRIO == F))))

monadic_africa_all$ANY_PRIO_ON[lag(monadic_africa_all$ANY_PRIO)] <- NA
monadic_africa_all$ANY_PRIO_OFF[!lag(monadic_africa_all$ANY_PRIO)] <- NA
monadic_africa_all$WAR_PRIO_ON[lag(monadic_africa_all$WAR_PRIO)] <- NA
monadic_africa_all$WAR_PRIO_OFF[!lag(monadic_africa_all$WAR_PRIO)] <- NA

monadic_africa_all =  monadic_africa_all %>%
    filter((YEAR >= 1981) & (YEAR <= 1999))

# Adding fixed effects and country specific effects
clean_df = read_csv('clean_data.csv')
merged_df = inner_join(x = clean_df, y = monadic_africa_all,by = c('year' = 'YEAR', 'ccode' = 'COW_loca
# fixed effect for each country (in dummies)
Iccode = fastDummies::dummy_cols(.data = merged_df$ccode,remove_selected_columns = T,remove_first_dummy

# extract the years for each country in dummies form
Iccyear = Iccode * (merged_df$year - 1978)

# merge the dummies to the main dataframe
names(Iccode) = paste0("Iccode", names(Iccode))
names(Iccyear) = paste0("Iccyear", names(Iccyear))
Iccyear = Iccyear %>% rowid_to_column("ID")
Iccode = Iccode %>% rowid_to_column("ID")
merged_df = merged_df %>% rowid_to_column("ID")
dummies_df = left_join(x = Iccode, y = Iccyear, by = 'ID')
master_df = left_join(merged_df,dummies_df,by = c('ID' = 'ID'))
master_df = master_df %>%
  filter(!is.na(GDP_G_L)) %>%
  filter(!(is.na(GDP_G))) %>%
  mutate(year_actual = year, year = year_actual - 1978)

# covariates
X = c('Y_0','polity2l','ethfrac','relfrac','Oil','lmtnest','lpopl1')
X_log = c('log_Y_0','polity2l','ethfrac','relfrac','Oil','lmtnest','lpopl1')

write.csv(x = master_df, file = 'master_df.csv')
```

Analysis

T2

```
# model 1: Economic Growth Rate ~ Growth in rainfall at time t + Growth in rainfall at t - 1.
m1 = lm(formula = GDP_G ~ GPCP_G + GPCP_G_L, data = master_df)
m1.se = sqrt(diag(sandwich::vcovCL(m1, cluster = ~ccode,type = 'HC1')))

# model 2: Economic Growth Rate ~ Growth in rainfall, t +  Growth in rainfall, t - 1 + covariates


# model formula for m2
m2formula = reformulate(termlabels = c(grep('Iccyear',x = names(master_df),value = T),'GPCP_G','GPCP_G_
m2 = lm(formula = m2formula, data = master_df)
m2.se = sqrt(diag(sandwich::vcovCL(m2, cluster = ~ccode,type = 'HC1')))

# m3
m3formula = reformulate(termlabels = c(grep('Iccode|Iccyear',x = names(master_df),value = T),'GPCP_G','

m3 = lm(formula = m3formula, data = master_df)
m3.se = sqrt(diag(sandwich::vcovCL(m3, cluster = ~ccode,type = 'HC1')))

# m4
m4formula = reformulate(termlabels = c(grep('Iccode|Iccyear',x = names(master_df),value = T),'GPCP_G','

m4 = lm(formula = m4formula, data = master_df)
m4.se = sqrt(diag(sandwich::vcovCL(m4, cluster = ~ccode,type = 'HC1')))
```

```
# m5
m5formula = reformulate(termlabels = c(grep('Iccode|Iccyear',x = names(master_df),value = T),'GPCP_G','G

m5 = lm(formula = m5formula, data = master_df)
m5.se = sqrt(diag(sandwich::vcovCL(m5, cluster = ~ccode,type = 'HC1')))

# table 2
stargazer(m1,m2,m3,m4,m5,se=list(m1.se,m2.se,m3.se,m4.se,m5.se),header = F, type = 'text', omit = c('Ic
```

T4: OLS VERSUS 2SLS

```
rformula = reformulate(termlabels = c(X,'GDP_G', 'GDP_G_L', 'year'),response = 'ANY_PRIO')
probit_model = glm(rformula, family = binomial(link = "probit"), data = master_df)
probit.se = sqrt(diag(sandwich::vcovCL(probit_model, cluster = ~ccode,type = 'HC1')))
rmse1 = sqrt(mean(probit_model$residuals^2))

rformula2 = reformulate(termlabels = c(X,'GDP_G', 'GDP_G_L', 'year'),response = 'ANY_PRIO')
ols2 = lm(rformula2, data = master_df)
ols2.se = sqrt(diag(sandwich::vcovCL(ols2, cluster = ~ccode,type = 'HC1')))
rmse2 = sqrt(mean(ols2$residuals^2))


#master_df = master_df %>% mutate(log_Y_0 = log(Y_0))


rformula3 = reformulate(termlabels = c(grep('Iccyear',x = names(master_df),value = T),X,'GDP_G', 'GDP_G
ols3 = lm(rformula3, data = master_df)
ols3.se = sqrt(diag(sandwich::vcovCL(ols3, cluster = ~ccode,type = 'HC1')))
rmse3 = sqrt(mean(ols3$residuals^2))


# OLS 4:
rformula4 = reformulate(termlabels = c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GDP_G', 'G
ols4 = lm(rformula4, data = master_df)
ols4.se = sqrt(diag(sandwich::vcovCL(ols4, cluster = ~ccode,type = 'HC1')))
rmse4 = sqrt(mean(ols4$residuals^2))


# IV 2SLS 5:
# stage 1:
m5formula1 = reformulate(c(grep('Iccyear',x = names(master_df),value = T),'GPCP_G','GPCP_G_L',X))
# stage 2:
m5formula2 = reformulate(c(grep('Iccyear',x = names(master_df),value = T),X,'GDP_G_L', "GDP_G"),response

iv2sls_5 = ivreg(formula = m5formula2, instruments = m5formula1, data = master_df)
iv2sls_5.se = sqrt(diag(sandwich::vcovCL(iv2sls_5, cluster = ~ccode,type = 'HC1')))
rmse5 = sqrt(mean(iv2sls_5$residuals^2))

# IV 2SLS 6:
# stage 1:
m6formula1 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GPCP_G', "GPCP_G_L"))
```

```
# stage 2:
m6formula2 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GDP_G_L', "GDP_G"),res
#m5_2 = lm(m5formula2, data = master_df)

iv2sls_6 = ivreg(formula = m6formula2, instruments = m6formula1, data = master_df)
iv2sls_6.se = sqrt(diag(sandwich::vcovCL(iv2sls_6, cluster = ~ccode,type = 'HC1')))
rmse6 = round(sqrt(mean(iv2sls_6$residuals^2)),3)

# IV 2SLS 7:
# stage 1:
m7formula1 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GPCP_G', "GPCP_G_L"))
# stage 2:
m7formula2 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GDP_G_L', "GDP_G"),res

iv2sls_7 = ivreg(formula = m7formula2, instruments = m7formula1, data = master_df)
iv2sls_7.se = sqrt(diag(sandwich::vcovCL(iv2sls_7, cluster = ~ccode,type = 'HC1')))
rmse7 = round(sqrt(mean(iv2sls_7$residuals^2)),3)

# Summary table 4:
stargazer(probit_model,ols2,ols3,ols4,iv2sls_5,iv2sls_6,iv2sls_7,se = list(probit.se, ols2.se, ols3.se,
```

T3

```
t3_m1formula = reformulate(termlabels = c(grep('Iccode|Iccyear',x = names(master_df),value = T),'GPCP_G

t3_m1 = lm(formula = t3_m1formula, data = master_df)
t3_m1.coef = t3_m1$coefficients
t3_m1.se = sqrt(diag(sandwich::vcovCL(t3_m1, cluster = ~ccode,type = 'HC1')))


t3_m2formula = reformulate(termlabels = c(grep('Iccode|Iccyear',x = names(master_df),value = T),'GPCP_G

t3_m2 = lm(formula = t3_m2formula, data = master_df)
t3_m2.coef = t3_m2$coefficients
t3_m2.se = sqrt(diag(sandwich::vcovCL(t3_m2, cluster = ~ccode,type = 'HC1')))

stargazer(t3_m1,t3_m2,header = F, type = 'text', omit = c('Iccyear','Iccode'), omit.labels = c("Country-
```

T5: Interactions between Economic Growth and Country Characteristics Dependent Variable: Civil Conflict 25 Deaths

```
# IV 2SLS (1): Interactions between Economic Growth and Democracy on Civil Con ict

## stage 1:
t5_f1 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GPCP_G','GPCP_G_L', "polit
## stage 2:
t5_f2 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GDP_G_L', "GDP_G", "polity

iv2sls_1 = ivreg(formula = t5_f2, instruments = t5_f1, data = master_df)
iv2sls_1.se = sqrt(diag(sandwich::vcovCL(iv2sls_1, cluster = ~ccode,type = 'HC1'))) # Cluster Robust St
```

```
# IV 2SLS (2): Interactions between Economic Growth and Log Per Capital Income on Civil Con ict

## stage 1:
t5_f1 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GPCP_G','GPCP_G_L', "Y_0*GI
## stage 2:
t5_f2 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GDP_G_L', "GDP_G", "Y_0*GDI

iv2sls_2 = ivreg(formula = t5_f2, instruments = t5_f1, data = master_df)
iv2sls_2.se = sqrt(diag(sandwich::vcovCL(iv2sls_2, cluster = ~ccode,type = 'HC1'))) # Cluster Robust St
rmse2 = sqrt(mean(iv2sls_2$residuals^2))

# IV 2SLS (3): Interactions between Economic Growth and ethnolinguistic fractionalization on Civil Con i

## stage 1:
t5_f1 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GPCP_G','GPCP_G_L', "ethfra
## stage 2:
t5_f2 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GDP_G_L', "GDP_G", "ethfrac

iv2sls_3 = ivreg(formula = t5_f2, instruments = t5_f1, data = master_df)
iv2sls_3.se = sqrt(diag(sandwich::vcovCL(iv2sls_3, cluster = ~ccode,type = 'HC1')))
rmse3 = sqrt(mean(iv2sls_3$residuals^2))

# IV 2SLS (4): Interactions between Economic Growth and being an oil-exporting country on Civil Con ict

## stage 1:
t5_f1 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GPCP_G','GPCP_G_L', "Oil*GI
## stage 2:
t5_f2 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GDP_G_L', "GDP_G", "Oil*GDI

iv2sls_4 = ivreg(formula = t5_f2, instruments = t5_f1, data = master_df)
iv2sls_4.se = sqrt(diag(sandwich::vcovCL(iv2sls_4, cluster = ~ccode,type = 'HC1')))
rmse4 = sqrt(mean(iv2sls_4$residuals^2))


# IV 2SLS (5): Interactions between Economic Growth and being an mountainous country on Civil Con ict

## stage 1:
t5_f1 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GPCP_G','GPCP_G_L', "lmtnes
## stage 2:
t5_f2 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GDP_G_L', "GDP_G", "lmtnest

iv2sls_5 = ivreg(formula = t5_f2, instruments = t5_f1, data = master_df)
iv2sls_5.se = sqrt(diag(sandwich::vcovCL(iv2sls_5, cluster = ~ccode,type = 'HC1')))
rmse5 = sqrt(mean(iv2sls_5$residuals^2))

stargazer(iv2sls_1,iv2sls_2,iv2sls_3,iv2sls_4,iv2sls_5,se = list(iv2sls_1.se, iv2sls_2.se, iv2sls_3.se,
```

T6: Economic shocks on the onset of civil conflict

```
# IV 2SLS 1:
# stage 1:
m1formula1 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GPCP_G','GPCP_G_L'))
# stage 2:
```

```
m1formula2 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GDP_G_L', "GDP_G"),re

iv2sls_1 = ivreg(formula = m1formula2, instruments = m1formula1, data = master_df)
iv2sls_1.se = sqrt(diag(sandwich::vcovCL(iv2sls_1, cluster = ~ccode,type = 'HC1')))
rmse1 = sqrt(mean(iv2sls_1$residuals^2))

# IV 2SLS 2:
# stage 1:
m2formula1 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GPCP_G','GPCP_G_L'))
# stage 2:
m2formula2 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GDP_G_L', "GDP_G"),re

iv2sls_2 = ivreg(formula = m2formula2, instruments = m2formula1, data = master_df)
iv2sls_2.se = sqrt(diag(sandwich::vcovCL(iv2sls_2, cluster = ~ccode,type = 'HC1')))
rmse2 = sqrt(mean(iv2sls_2$residuals^2))


stargazer(iv2sls_1,iv2sls_2,header = F, type = 'text', omit = c('Iccyear','Iccode'), omit.labels = c("C
```

TC1: Summary Statistcs (Appendix)

```
tablec1 = master_df %>%
  group_by(Location) %>%
  summarise(Year_count = length(year),
            Minor = sum(ANY_PRIO),
            War = sum(WAR_PRIO)) %>%
  arrange((Location)) %>%
  janitor::adorn_totals()



tablec1 %>% knitr::kable(col.names = c('Country', 'Total Years', "Years of Civil Con ict  25 Deaths (PRI
```

T1

```
library(dplyr)
library(tibble)
main_df = read.csv('master_df.csv')
table_1 = main_df  %>%
  summarise(
    mean = c(
      mean(ANY_PRIO, na.rm = T),
      mean(ANY_PRIO_ON,na.rm = T),
      mean(ANY_PRIO_OFF,na.rm = T),
      mean(WAR_PRIO, na.rm = T),
      mean(WAR_PRIO_ON,na.rm = T),
      mean(WAR_PRIO_OFF,na.rm = T),
      mean(GPCP,na.rm = T),
      mean(GPCP_G,na.rm = T),
      mean(GPCP_G_L,na.rm = T),
      mean(GDP_G,na.rm = T),
      mean(GDP_G_L,na.rm = T),
```

```r
      mean(Y_0,na.rm = T),
      mean(polity2l,na.rm = T),
      mean(ethfrac,na.rm = T),
      mean(relfrac,na.rm = T),
      mean(Oil,na.rm = T),
      mean(lmtnest,na.rm=T),
      mean(lpopl1,na.rm = T),
      mean(TOT_100_G,na.rm = T)
    ),
    sd = c(
      sd(ANY_PRIO,na.rm = T),
      sd(ANY_PRIO_ON,na.rm = T),
      sd(ANY_PRIO_OFF,na.rm = T),
      sd(WAR_PRIO,na.rm = T),
      sd(WAR_PRIO_ON,na.rm = T),
      sd(WAR_PRIO_OFF,na.rm = T),
      sd(GPCP,na.rm = T),
      sd(GPCP_G,na.rm = T),
      sd(GPCP_G_L,na.rm = T),
      sd(GDP_G,na.rm = T),
      sd(GDP_G_L,na.rm = T),
      sd(Y_0,na.rm = T),
      sd(polity2l,na.rm = T),
      sd(ethfrac,na.rm = T),
      sd(relfrac,na.rm = T),
      sd(Oil,na.rm = T),
      sd(lmtnest,na.rm=T),
      sd(lpopl1,na.rm = T),
      sd(TOT_100_G,na.rm = T)
    ),
    obs = c(
      length(na.omit(ANY_PRIO)),
      length(na.omit(ANY_PRIO_ON)),
      length(na.omit(ANY_PRIO_OFF)),
      length(na.omit(WAR_PRIO)),
      length(na.omit(WAR_PRIO_ON)),
      length(na.omit(WAR_PRIO_OFF)),
      length(na.omit(GPCP)),
      length(na.omit(GPCP_G)),
      length(na.omit(GPCP_G_L)),
      length(na.omit(GDP_G)),
      length(na.omit(GDP_G_L)),
      length(na.omit(Y_0)),
      length(na.omit(polity2l)),
      length(na.omit(ethfrac)),
      length(na.omit(relfrac)),
      length(na.omit(Oil)),
      length(na.omit(lmtnest)),
      length(na.omit(lpopl1)),
      length(na.omit(TOT_100_G))
    )
  ) %>%
  add_column(.before = T, Label = c(
    'ANY_PRIO', 'ANY_PRIO_ON','ANY_PRIO_OFF','WAR_PRIO','WAR_PRIO_ON',
    'WAR_PRIO_OFF','GPCP','GPCP_G','GPCP_G_L','GDP_G','GDP_G_L','Y_0',
    'polity2l','ethfrac','relfrac','Oil','lmtnest','lpopl1','TOT_100_G')) %>%
  knitr::kable()
```

Robustness Check

```
# The IV-2SLS  xed-effects results for the 25-death threshold are robust to dropping one country at a ti

# stage 1:
m6formula1 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GPCP_G', "GPCP_G_L"))
# stage 2:
m6formula2 = reformulate(c(grep('Iccyear|Iccode',x = names(master_df),value = T),'GDP_G_L', "GDP_G"),re


subset_iv = map_df(1:41, function(i){

 .df =  master_df  %>% filter(ccode != unique(ccode)[i])
  .model = AER::ivreg(formula = m6formula2, instruments = m6formula1, data = .df)
  GDP_G.coef = .model$coefficient['GDP_G']
  GDP_G_L.coef = .model$coefficient['GDP_G_L']

  GDP_G.se = sqrt(diag(sandwich::vcovCL(x = .model, cluster = .df$ccode,type = 'HC1',)))['GDP_G']
  GDP_G_L.se = sqrt(diag(sandwich::vcovCL(x = .model, cluster = .df$ccode,type = 'HC1',)))['GDP_G_L']
  p.value.GDP_G = (1 - pnorm(q = abs(GDP_G.coef/GDP_G.se)))*2
  p.value.GDP_G_L = (1 - pnorm(q = abs(GDP_G_L.coef/GDP_G_L.se)))*2

  names(p.value.GDP_G) = 'p.value.GDP_G'
  names(p.value.GDP_G_L) = 'p.value.GDP_G_L'
  names(GDP_G.coef) = 'GDP_G.coef'
  names(GDP_G_L.coef) = 'GDP_G_L.coef'
  names(GDP_G.se) = 'GDP_G.se'
  names(GDP_G_L.se) = 'GDP_G_L.se'

  return(c(GDP_G.coef,
         GDP_G_L.coef,
         p.value.GDP_G,
         GDP_G.se,
         GDP_G_L.se,
         p.value.GDP_G_L))
})

# coef cients on economic growth
range(subset_iv[,'GDP_G.coef'])
# coef cients on lagged economic growth
range(subset_iv[,'GDP_G_L.coef'])
# p.value on economic growth
range(subset_iv[,'p.value.GDP_G'])
# coef cients on lagged economic growth
range(subset_iv[,'p.value.GDP_G_L'])
```

Re-analysis: Inversion Method

```
# Anderson-Rubin interval for the effect of Lagged GDP Growth (%) on Civil Conflict (# of conflicts >= 

# do a sequence of hypothesis tests
get_AR_pvals = function(y,z,d,with_cov=T, betas = seq(-10,10,0.01)){
  .data  = master_df

  if (with_cov == F) {
    rformula = reformulate(c(grep('Iccyear|Iccode',x = names(.data),value = T), z),response = 'adj_y')
  }
  else(
    rformula = reformulate(c(grep('Iccyear',x = names(.data),value = T),X, z),response = 'adj_y')
  )

  D = master_df[[d]]
  Y = master_df[[y]]
  p.values = map_dbl(betas, function(b){
    # b is the coefficient of interest.
    .data$adj_y = Y - b*D
    reg = glm(rformula,data = .data,family = 'gaussian')
    coef.z = reg$coefficients[z]
    se.z = sqrt(diag(hccm(reg))[z])
    Tstat = coef.z/se.z
    p.value = (1 - pt(q = abs(Tstat),df = reg$df.residual))*2
    return(p.value)
  })
  return(p.values)
}

# AFR test
betas = seq(-10,5, 0.1)
AR_pvals.Y1.z1.d1 = get_AR_pvals(y = 'ANY_PRIO', z  = 'GPCP_G_L', d = 'GDP_G_L', with_cov = T, betas = 

point.est.Y1.z1.d1 =  betas[which.max(AR_pvals.Y1.z1.d1)]
CI.Y1.z1.d1 = range(betas[AR_pvals.Y1.z1.d1 >= 0.05])
plot(AR_pvals.Y1.z1.d1 ~ betas, type = "l",
     xlab = "coefficient of GDP Growth",
     ylab = "p-value")
abline(h = 0.05, lty = 2, col = "grey")
abline(v = point.est.Y1.z1.d1, lty = 2, col = "grey")
ARCI = CI.Y1.z1.d1
abline(v = ARCI[1], lty = 2, col = "grey")
abline(v = ARCI[2], lty = 2, col = "grey")
```

IPW

```
master_df = read.csv('master_df.csv')
set.seed(156)

OS_est = function(z, y, x, out.family = gaussian,
                  truncpscore = c(0, 1))
{
     ## fitted propensity score
     pscore   = glm(z ~ x, family = binomial)$fitted.values
     pscore   = pmax(truncpscore[1], pmin(truncpscore[2], pscore))

     ## fitted potential outcomes
     outcome1 = glm(y ~ x, weights = z,
                    family = out.family)$fitted.values
     outcome0 = glm(y ~ x, weights = (1 - z),
                    family = out.family)$fitted.values

     ## regression imputation estimator
     ace.reg  = mean(outcome1 - outcome0)
     ## IPW estimators
     ace.ipw0 = mean(z*y/pscore - (1 - z)*y/(1 - pscore))
     ace.ipw  = mean(z*y/pscore)/mean(z/pscore) -
                   mean((1 - z)*y/(1 - pscore))/mean((1 - z)/(1 - pscore))
     ## doubly robust estimator
     res1     = y - outcome1
     res0     = y - outcome0
     ace.dr   = ace.reg + mean(z*res1/pscore - (1 - z)*res0/(1 - pscore))

     return(c(ace.reg, ace.ipw0, ace.ipw, ace.dr))
}


OS_ATE = function(z, y, x, n.boot = 2*10^2,
                     out.family = gaussian, truncpscore = c(0, 1))
{
     point.est  = OS_est(z, y, x, out.family, truncpscore)

     ## nonparametric bootstrap
     n.sample   = length(z)
     x          = as.matrix(x)
     boot.est   = replicate(n.boot,
                  {id.boot = sample(1:n.sample, n.sample, replace = TRUE)
                  OS_est(z[id.boot], y[id.boot], x[id.boot, ],
                         out.family, truncpscore)})

     boot.se    = apply(boot.est, 1, sd)

     res        = rbind(point.est, boot.se)
     rownames(res) = c("est", "se")
     colnames(res) = c("reg", "HT", "Hajek", "DR")

     return(res)
}


Z = as.integer((master_df$GDP_G_L) < 0)
Y = master_df$ANY_PRIO
X = as.matrix(master_df[, c(85,61,51,56,49,36)])

pscore = glm(Z ~ X, family = binomial)$fitted.values
```

```
hist(pscore)
hist(pscore[Z==1])
hist(pscore[Z==0])

causaleffects = OS_ATE(Z, Y, X,
                       out.family = binomial, n.boot = 1000, truncpscore = c(0.1,0.9))
round(causaleffects, 3)
rbind(causaleffects[1, ] - 1.96*causaleffects[2, ],
      causaleffects[1, ] + 1.96*causaleffects[2, ])
```

“‘