thesis_v5

Camilla Zhang

2022-10-24

Import Libraries

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(countrycode)
library(stringr)
library(plm)
```

```
##
## Attaching package: 'plm'

## The following objects are masked from 'package:dplyr':
##
##     between, lag, lead
```

```
library(tidyr)
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following object is masked from 'package:plm':
##
##     between

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```r
library(ggplot2)
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(ggplot2)
library(ggthemes)
library(purrr)
```

```
##
## Attaching package: 'purrr'

## The following object is masked from 'package:data.table':
##
##     transpose
```

```r
library(rworldmap)
```

```
## Loading required package: sp

## ### Welcome to rworldmap ###

## For a short introduction type :   vignette('rworldmap')
```

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following object is masked from 'package:Hmisc':
##
##     describe

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

Import datasets

```
desta = read.csv("/Users/jayzee/Documents/qmss_fall22/thesis/data/desta.csv")
withdrawals = read.csv("/Users/jayzee/Documents/qmss_fall22/thesis/data/withdrawals.csv")
withdrawals = withdrawals[!duplicated(withdrawals),]  #remove duplicates for withdrawals
desta_dyads = read.csv("/Users/jayzee/Documents/qmss_fall22/thesis/data/desta_dyads_v3.csv")
desta_dyads1 = desta_dyads[!duplicated(desta_dyads),] #Remove duplicate entries in dyadic data
hdi = read.csv("/Users/jayzee/Documents/qmss_fall22/thesis/data/hdi_composite.csv", row.names=NULL)
vdem = read.csv("/Users/jayzee/Documents/qmss_fall22/thesis/data/vdem_folder/vdem.csv", row.names=NULL)
gallup = read.csv("/Users/jayzee/Documents/qmss_fall22/thesis/data/gallup.csv")
```

Test subset data (to ensure wrangling merging was successful)

```
#desta_dyads1 = subset(desta_dyads, base_treaty %in% c(1, 192, 65, 810, 562))
#desta_dyads1 = subset(desta_dyads, base_treaty %in% c(192, 1, 249))
```

Divide dataset between base, accession, and consolidated

```
#Base
base_ptas = subset(desta_dyads1, entry_type == "base_treaty")

#Consolidated
consolidated_ptas = subset(desta_dyads1, entry_type == "consolidated")

#Accession
accession_ptas = subset(desta_dyads1, entry_type == "accession")
```

###Merge DESTA data with dyadic data to aggregate by country-PTA-year

```
#Join DESTA indices with BASE PTAs based on base treaty number and year
merged_base = merge(base_ptas, desta[,c("base_treaty", "year", "depth_index", "depth_rasch")], by = c("

#Join DESTA indices with BASE PTAs based on *new* treaty number and year for consolidate PTAs that chan
```

```r
merged_con = merge(consolidated_ptas, desta[,c("number", "depth_index", "depth_rasch")], by = c("number
#Subset consolidated data where depth is not NA
con_non_nas = merged_con[!is.na(merged_con$depth_index),]
#Bind non NA consolidated data w/merged data
con_base = rbind(con_non_nas, merged_base)

#Subset CONSOLIDATED PTAs with NA and populate it (These are NAs b/c their depth/provisions didn't chan
con_nas <- merged_con[is.na(merged_con$depth_index),] #Subset
con_nas2 = merge(con_nas, desta[,c("base_treaty", "depth_index")], by = c("base_treaty"), all.x = TRUE)
dplyr::rename(depth_index = depth_index.y)
merged_con2 = subset(con_nas2, select = -c(depth_index.x)) #remove duplicate columns

# Join the CON BASE and (previously NA) CONSOLIDATED datasets together
con_base2 = rbind(merged_con2, con_base)
con_base2 <- con_base2[!is.na(con_base2$depth_index),]

# Remove duplicate rows based on country, base treaty, year, and depth index
con_base3 = con_base2[!duplicated(con_base2[c("iso1", "depth_index", "base_treaty", "year")]),]

# Rename depth index column to depth_filled
con_base3 = con_base3 %>% dplyr::rename(depth_filled = depth_index)
```

Appending accession countries

```r
#Add missing years by base treaty and fill PTA count with 0
desta2 = desta %>% group_by(base_treaty) %>% tidyr::complete(year = min(year):2021, fill = list(depth =

#Populate the depth based on previous year's depth index
desta2$depth_filled <- na.locf(desta2$depth_index)

#Create accession variable for the new DESTA
merged_acc = merge(accession_ptas, desta2[,c("depth_filled", "depth_rasch", "base_treaty", "year")], by

#Bind accession with con/base PTAs
merged_all = rbind(con_base3, merged_acc)

#Rename ISO column
merged_all = merged_all %>% dplyr::rename(iso3n = iso1)
```

Merge DESTA data with GWP incumbent support data

```r
#Merge by iso3 numeric country codes
gallup$iso3n = countryname(gallup$Geography, destination = 'iso3n')
```

```
## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest, : Some val

## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest, : The orig
##                      class. Filling-in bad matches with NA instead.
```

```
## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest, : Some va
```

```
#Merge by year
gallup = gallup %>%
  dplyr::rename(year = Time)

#Remove NAs where country and approval ratings are NA
gallup = subset(gallup, iso3n != "NA")
gallup = subset(gallup, Approve != "NA")

#Remove percentages in approval rating variable
gallup$Approve <- as.integer(gsub('%', '', gallup$Approve))

merged_all = merge(merged_all, gallup, by = c("iso3n", "year"), all.x = TRUE)
```

Find the mean and median of treaties

```
#Get average difference between year and entryforceyear
merged_all$difference_year = merged_all$entryforceyear-merged_all$year
mean(merged_all$difference_year, na.rm = T)
```

```
## [1] 1.6561
```

```
median(merged_all$difference_year, na.rm = T)
```

```
## [1] 1
```

```
#Percentage of trade agreements with a depth index
sum(unique(desta$base_treaty))/sum(unique(desta_dyads1$base_treaty))
```

```
## [1] 0.6653162
```

###Merge PTA data w/HDI + Add a binary indicator to signify if trade agreement is deep

```
## Merge HDI with desta data ##

#Subset HDI dataset
hdi = hdi %>% dplyr::select(iso3, country, region, hdi_1990, hdi_1991, hdi_1992, hdi_1993, hdi_1994, hd

#Produce iso3 numeric country codes for hdi_hdi dataset
hdi$iso3n = countryname(hdi$country, destination = 'iso3n')
```

```
## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest, : Some va
```

```
## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest, : The ori
##                       class. Filling-in bad matches with NA instead.
```

```r
#Merge HDI with PTA data
hdi_hdi = hdi %>%
  tidyr::pivot_longer(
    cols = starts_with("hdi_"),
    names_to = "year",
    names_prefix = "hdi_",
    values_to = "hdi_score"
  )

hdi_hdi = hdi_hdi %>% select(iso3n, year, hdi_score)
depth_data = merge(merged_all, hdi_hdi, by = c("iso3n", "year"), all.x = T) %>% mutate(developing = cas
                                                     hdi_score>= 0.8 ~ 0))

# Assign NAs in developing to a 1 or 0 based on current values
depth_data$developing2 = nafill(depth_data$developing, type = "nocb")
depth_data$developing3 = ifelse(depth_data$developing2 == 0, "Developed", "Developing")

#Signify that an agreement is deep w/binary variable
depth_data$deep = ifelse(depth_data$depth_filled >= mean(depth_data$depth_filled, na.rm = T), 1, 0)
```

#Descriptive Statistics and EDA

Data Visualization 1: PTAs signed and Average PTA depth over time b/w developed and developing

```r
#Count number of PTAs for each country_type-year (developing/not developing)
country_yr_n = depth_data %>% group_by(developing3,year)  %>% summarise(n = n())
```

```
## `summarise()` has grouped output by 'developing3'. You can override using the
## `.groups` argument.
```

```r
country_yr_n = subset(country_yr_n, !is.na(developing3)) #Remove NA rows
country_yr_n = subset(country_yr_n, !is.na(n))
country_yr_n
```

```
## # A tibble: 128 x 3
## # Groups:   developing3 [2]
##    developing3  year     n
##    <chr>       <int> <int>
##  1 Developed    1951     3
##  2 Developed    1954     2
##  3 Developed    1957     3
##  4 Developed    1958     2
##  5 Developed    1960     8
##  6 Developed    1961    10
##  7 Developed    1963     6
##  8 Developed    1965     4
##  9 Developed    1968     6
## 10 Developed    1969    19
## # ... with 118 more rows
```

```r
#Get average depth of PTAs for each country-year
country_yr_depth = depth_data %>% group_by(developing3,year) %>% summarise(avg_depth = mean(depth_fille
```

```
## `summarise()` has grouped output by 'developing3'. You can override using the
## `.groups` argument.
```

```r
country_yr_depth = subset(country_yr_depth, !is.na(developing3)) #Remove NA rows
country_yr_depth = subset(country_yr_depth, !is.na(avg_depth))
country_yr_depth
```

```
## # A tibble: 107 x 3
## # Groups:   developing3 [2]
##    developing3  year avg_depth
##    <chr>       <int>     <dbl>
##  1 Developed    1951      0
##  2 Developed    1954      0
##  3 Developed    1957      4
##  4 Developed    1958      4
##  5 Developed    1960      1.5
##  6 Developed    1961      1.3
##  7 Developed    1963      0.5
##  8 Developed    1965      1
##  9 Developed    1968      1.67
## 10 Developed    1969      1.37
## # ... with 97 more rows
```

```r
country_n = depth_data %>% group_by(country1) %>% summarise(n = n())
country_depth = depth_data %>% group_by(country1) %>% summarise(avg_depth = mean(depth_filled))
country_n
```

```
## # A tibble: 203 x 2
##    country1              n
##    <chr>             <int>
##  1 Afghanistan           3
##  2 Albania              12
##  3 Algeria              20
##  4 Andorra               5
##  5 Angola               22
##  6 Antigua & Barbuda    25
##  7 Argentina            41
##  8 Armenia              14
##  9 Australia            20
## 10 Austria              98
## # ... with 193 more rows
```

```r
#Add in developing country column
country_depth_m = merge(country_depth, depth_data[,c('developing3', 'country1')], by = 'country1', all.
country_depth_m = subset(country_depth_m[!duplicated(country_depth_m), ], !is.na(avg_depth))
country_depth_m
```

```
##                                country1 avg_depth developing3
```

```
## 1                  Afghanistan 0.6666667  Developing
## 4                      Albania 4.3333333  Developing
## 36                     Andorra 1.0000000   Developed
## 63           Antigua & Barbuda 2.3200000   Developed
## 87           Antigua & Barbuda 2.3200000  Developing
## 88                   Argentina 1.1707317   Developed
## 94                   Argentina 1.1707317  Developing
## 143                  Australia 4.6500000   Developed
## 261                 Azerbaijan 1.5000000  Developing
## 267                    Bahamas 2.4500000   Developed
## 273                    Bahamas 2.4500000  Developing
## 287                    Bahrain 2.6666667  Developing
## 289                    Bahrain 2.6666667   Developed
## 296                 Bangladesh 0.7777778  Developing
## 305                   Barbados 2.2000000  Developing
## 462                     Belize 2.3478261  Developing
## 505                     Bhutan 0.8000000  Developing
## 510                    Bolivia 2.2222222  Developing
## 528     Bosnia & Herzegovina 3.2500000  Developing
## 570                     Brazil 1.2631579  Developing
## 608                     Brunei 3.0588235   Developed
## 610                     Brunei 3.0588235  Developing
## 740                   Cambodia 2.7777778  Developing
## 773                     Canada 4.3181818   Developed
## 857                      Chile 3.9056604  Developing
## 870                      Chile 3.9056604   Developed
## 910                      China 4.0000000  Developing
## 930                   Colombia 2.8809524  Developing
## 1046              Cook Islands 1.8750000  Developing
## 1054                Costa Rica 3.5714286  Developing
## 1070                Costa Rica 3.5714286   Developed
## 1161                      Cuba 0.9130435  Developing
## 1450                  Dominica 2.2800000  Developing
## 1475        Dominican Republic 2.9166667  Developing
## 1487                   Ecuador 1.6538462  Developing
## 1551               El Salvador 2.5666667  Developing
## 1600                   Eritrea 2.5555556  Developing
## 1737             Faroe Islands 2.0714286  Developing
## 2029                   Georgia 3.0000000  Developing
## 2036                   Georgia 3.0000000   Developed
## 2261                   Grenada 2.1923077  Developing
## 2287                 Guatemala 2.8750000  Developing
## 2353                    Guyana 1.9642857  Developing
## 2381                     Haiti 2.5555556  Developing
## 2390                  Honduras 3.0740741  Developing
## 2417         Hong Kong SAR China 5.4000000   Developed
## 2491                   Iceland 4.4528302   Developed
## 2544                     India 1.5416667  Developing
## 2568                 Indonesia 2.3333333  Developing
## 2586                      Iran 0.4285714  Developing
## 2593                      Iraq 1.0000000  Developing
## 2704                    Israel 2.7142857   Developed
## 2712                    Israel 2.7142857  Developing
## 2848                   Jamaica 2.2500000  Developing
```

```
## 2872                            Japan 6.0588235   Developed
## 2889                           Jordan 1.4222222  Developing
## 2977                         Kiribati 1.9090909  Developing
## 2988                           Kosovo 4.5000000        <NA>
## 2992                           Kuwait 2.0000000  Developing
## 2993                           Kuwait 2.0000000   Developed
## 3018                             Laos 2.1875000  Developing
## 3105                          Lebanon 2.0000000  Developing
## 3186                    Liechtenstein 4.3584906   Developed
## 3428                  Macao SAR China 3.5000000  Developing
## 3484                         Malaysia 3.0800000  Developing
## 3492                         Malaysia 3.0800000   Developed
## 3509                         Maldives 0.6666667  Developing
## 3592                 Marshall Islands 2.4000000  Developing
## 3697 Micronesia (Federated States of) 2.4000000  Developing
## 3702                          Moldova 2.4705882  Developing
## 3719                           Monaco 1.0000000  Developing
## 3720                         Mongolia 3.5000000  Developing
## 3722                       Montenegro 4.4000000   Developed
## 3724                       Montenegro 4.4000000  Developing
## 3727                       Montserrat 2.2500000  Developing
## 3801                  Myanmar (Burma) 2.7000000  Developing
## 3841                            Nauru 2.0000000  Developing
## 3848                            Nepal 0.6000000  Developing
## 3970                      New Zealand 4.6250000   Developed
## 3986                        Nicaragua 2.6923077  Developing
## 4050                             Niue 1.8750000   Developed
## 4058                      North Korea 0.0000000   Developed
## 4059                  North Macedonia 3.8888889  Developing
## 4077                           Norway 4.2857143   Developed
## 4133                             Oman 2.8750000  Developing
## 4141                         Pakistan 1.5000000  Developing
## 4155                            Palau 2.4000000  Developing
## 4160          Palestinian Territories 1.5000000   Developed
## 4166                           Panama 2.5937500  Developing
## 4190                           Panama 2.5937500   Developed
## 4216                         Paraguay 1.6428571  Developing
## 4244                             Peru 3.5348837  Developing
## 4287                      Philippines 2.5263158  Developing
## 4475                            Qatar 1.8888889   Developed
## 4476                            Qatar 1.8888889  Developing
## 4609                       San Marino 1.0000000  Developing
## 4631                     Saudi Arabia 1.8000000  Developing
## 4637                     Saudi Arabia 1.8000000   Developed
## 4664                           Serbia 2.5882353  Developing
## 4719                        Singapore 4.4444444   Developed
## 4728                        Singapore 4.4444444  Developing
## 4961                      South Korea 4.6086957   Developed
## 5085                        Sri Lanka 1.5454545  Developing
## 5096                St. Kitts & Nevis 2.3750000  Developing
## 5120                        St. Lucia 2.2800000  Developing
## 5145         St. Vincent & Grenadines 2.3750000  Developing
## 5313                      Switzerland 4.0169492   Developed
## 5372                            Syria 0.8571429  Developing
```

```
## 5393                        Taiwan 5.6000000  Developing
## 5437                      Thailand 2.6000000  Developing
## 5457                   Timor-Leste 2.0000000  Developing
## 5480                         Tonga 1.7500000  Developing
## 5492             Trinidad & Tobago 2.0714286  Developing
## 5498             Trinidad & Tobago 2.0714286   Developed
## 5548                        Turkey 2.8604651  Developing
## 5554                        Turkey 2.8604651   Developed
## 5591                  Turkmenistan 1.0000000  Developing
## 5596                        Tuvalu 1.9090909  Developing
## 5630                       Ukraine 2.5714286  Developing
## 5651          United Arab Emirates 1.8333333   Developed
## 5654          United Arab Emirates 1.8333333  Developing
## 5789                 United States 5.8421053   Developed
## 5808                       Uruguay 1.4864865  Developing
## 5824                       Uruguay 1.4864865   Developed
## 5845                    Uzbekistan 1.4285714  Developing
## 5852                       Vanuatu 1.8181818  Developing
## 5903                       Vietnam 3.6315789  Developing
## 5927                         Yemen 1.6666667  Developing
```

```
country_n_m = merge(country_n, depth_data[,c('developing3', 'country1')], by = 'country1', all.x = TRUE)
country_n_m = subset(country_n_m[!duplicated(country_n_m), ], !is.na(n))
country_n_m
```

```
##                             country1   n developing3
## 1                        Afghanistan   3  Developing
## 4                            Albania  12  Developing
## 16                           Algeria  20  Developing
## 36                           Andorra   5   Developed
## 41                            Angola  22  Developing
## 63                 Antigua & Barbuda  25   Developed
## 87                 Antigua & Barbuda  25  Developing
## 88                         Argentina  41   Developed
## 94                         Argentina  41  Developing
## 129                          Armenia  14  Developing
## 143                        Australia  20   Developed
## 163                          Austria  98   Developed
## 261                       Azerbaijan   6  Developing
## 267                          Bahamas  20   Developed
## 273                          Bahamas  20  Developing
## 287                          Bahrain   9  Developing
## 289                          Bahrain   9   Developed
## 296                       Bangladesh   9  Developing
## 305                         Barbados  25  Developing
## 330                          Belarus  15  Developing
## 343                          Belarus  15   Developed
## 345                          Belgium 117   Developed
## 462                           Belize  23  Developing
## 485                            Benin  20  Developing
## 505                           Bhutan   5  Developing
## 510                          Bolivia  18  Developing
## 528               Bosnia & Herzegovina  12  Developing
## 540                         Botswana  30  Developing
```

```
## 570                        Brazil  38  Developing
## 608                        Brunei  17   Developed
## 610                        Brunei  17  Developing
## 625                      Bulgaria  68  Developing
## 661                      Bulgaria  68   Developed
## 693                  Burkina Faso  23  Developing
## 716                       Burundi  24  Developing
## 740                      Cambodia   9  Developing
## 749                      Cameroon  24  Developing
## 773                        Canada  22   Developed
## 795                    Cape Verde  14  Developing
## 809      Central African Republic  24  Developing
## 833                          Chad  24  Developing
## 857                         Chile  53  Developing
## 870                         Chile  53   Developed
## 910                         China  20  Developing
## 930                      Colombia  42  Developing
## 972                       Comoros  19  Developing
## 991          Congo - Brazzaville  24  Developing
## 1015            Congo - Kinshasa  31  Developing
## 1046                Cook Islands   8  Developing
## 1054                  Costa Rica  28  Developing
## 1070                  Costa Rica  28   Developed
## 1082               Côte d’Ivoire  24  Developing
## 1106                     Croatia  55   Developed
## 1108                     Croatia  55  Developing
## 1161                        Cuba  23  Developing
## 1184                      Cyprus  72   Developed
## 1233                      Cyprus  72  Developing
## 1256                     Czechia  70   Developed
## 1282                     Czechia  70  Developing
## 1326                     Denmark 105   Developed
## 1431                    Djibouti  19  Developing
## 1450                    Dominica  25  Developing
## 1475          Dominican Republic  12  Developing
## 1487                     Ecuador  26  Developing
## 1513                       Egypt  38  Developing
## 1551                 El Salvador  30  Developing
## 1581           Equatorial Guinea  19  Developing
## 1600                     Eritrea   9  Developing
## 1609                     Estonia  73   Developed
## 1619                     Estonia  73  Developing
## 1682                    Eswatini  36  Developing
## 1718                    Ethiopia  19  Developing
## 1737               Faroe Islands  14  Developing
## 1751                        Fiji  16  Developing
## 1767                     Finland 104   Developed
## 1871                      France 118  Developing
## 1872                      France 118   Developed
## 1989                       Gabon  24  Developing
## 2013                      Gambia  16  Developing
## 2029                     Georgia  12  Developing
## 2036                     Georgia  12   Developed
## 2041                     Germany 117   Developed
```

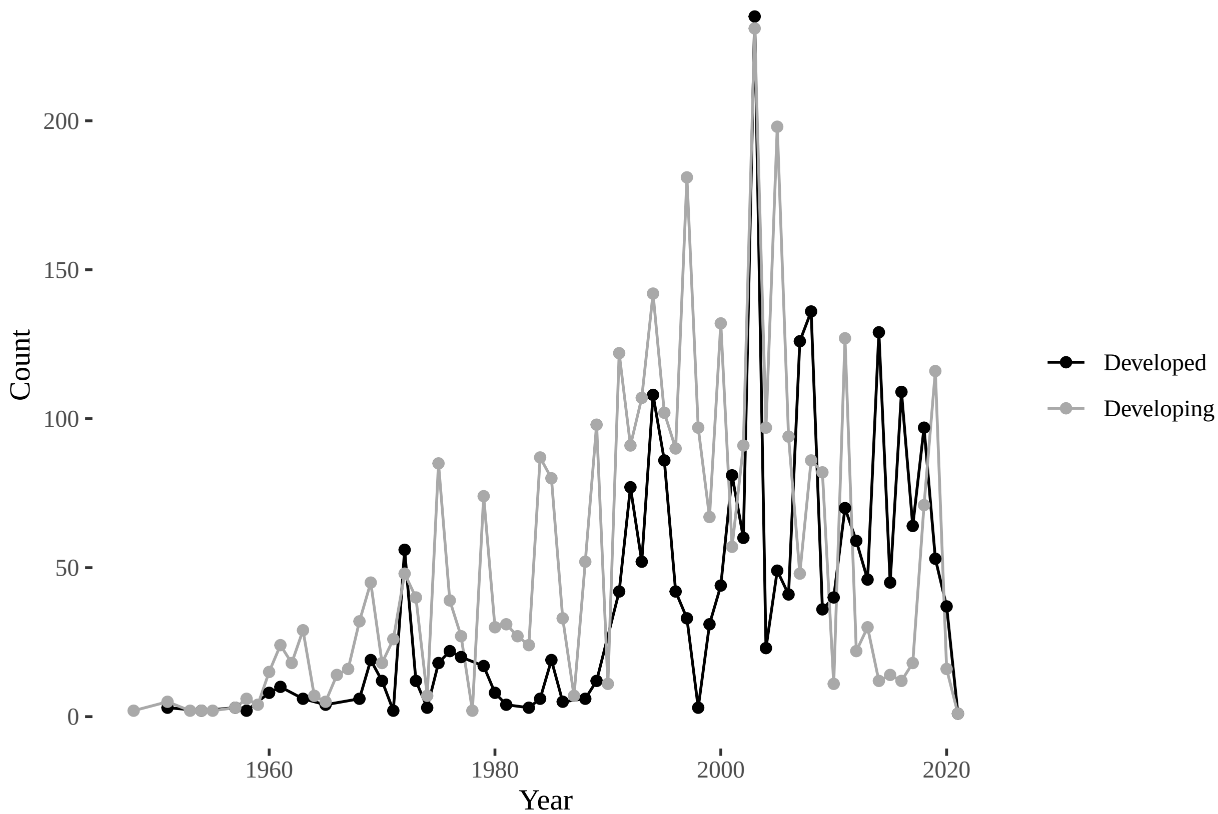
```
## 2158                              Ghana  19  Developing
## 2177                             Greece  84   Developed
## 2180                             Greece  84  Developing
## 2261                            Grenada  26  Developing
## 2287                          Guatemala  32  Developing
## 2319                             Guinea  18  Developing
## 2337                      Guinea-Bissau  16  Developing
## 2353                             Guyana  28  Developing
## 2381                              Haiti   9  Developing
## 2390                           Honduras  27  Developing
## 2417                 Hong Kong SAR China   5   Developed
## 2422                            Hungary  69   Developed
## 2424                            Hungary  69  Developing
## 2491                            Iceland  53   Developed
## 2544                              India  24  Developing
## 2568                          Indonesia  18  Developing
## 2586                               Iran   7  Developing
## 2593                               Iraq   8  Developing
## 2601                            Ireland 103   Developed
## 2602                            Ireland 103  Developing
## 2704                             Israel  28   Developed
## 2712                             Israel  28  Developing
## 2732                              Italy 116   Developed
## 2733                              Italy 116  Developing
## 2848                            Jamaica  24  Developing
## 2872                              Japan  17   Developed
## 2889                             Jordan  45  Developing
## 2934                         Kazakhstan  20  Developing
## 2954                              Kenya  23  Developing
## 2977                           Kiribati  11  Developing
## 2988                             Kosovo   4        <NA>
## 2992                             Kuwait  10  Developing
## 2993                             Kuwait  10   Developed
## 3002                         Kyrgyzstan  16  Developing
## 3018                               Laos  16  Developing
## 3034                             Latvia  71  Developing
## 3035                             Latvia  71   Developed
## 3105                            Lebanon  15  Developing
## 3120                            Lesotho  32  Developing
## 3152                            Liberia  19  Developing
## 3171                              Libya  15  Developing
## 3186                      Liechtenstein  53   Developed
## 3239                          Lithuania  72   Developed
## 3241                          Lithuania  72  Developing
## 3311                         Luxembourg 117  Developing
## 3312                         Luxembourg 117   Developed
## 3428                    Macao SAR China   2  Developing
## 3430                         Madagascar  24  Developing
## 3454                             Malawi  30  Developing
## 3484                           Malaysia  25  Developing
## 3492                           Malaysia  25   Developed
## 3509                           Maldives   3  Developing
## 3512                               Mali  23  Developing
## 3535                              Malta  57   Developed
```

```
## 3554                           Malta  57  Developing
## 3592                Marshall Islands   5  Developing
## 3597                      Mauritania  23  Developing
## 3620                       Mauritius  29  Developing
## 3627                       Mauritius  29   Developed
## 3649                          Mexico  48  Developing
## 3697 Micronesia (Federated States of)   5  Developing
## 3702                         Moldova  17  Developing
## 3719                          Monaco   1  Developing
## 3720                        Mongolia   2  Developing
## 3722                      Montenegro   5   Developed
## 3724                      Montenegro   5  Developing
## 3727                      Montserrat  16  Developing
## 3743                         Morocco  33  Developing
## 3776                      Mozambique  25  Developing
## 3801                 Myanmar (Burma)  10  Developing
## 3811                         Namibia  30  Developing
## 3841                           Nauru   7  Developing
## 3848                           Nepal   5  Developing
## 3853                     Netherlands 117   Developed
## 3970                     New Zealand  16   Developed
## 3986                       Nicaragua  26  Developing
## 4012                           Niger  22  Developing
## 4034                         Nigeria  16  Developing
## 4050                            Niue   8   Developed
## 4058                     North Korea   1   Developed
## 4059                 North Macedonia  18  Developing
## 4077                          Norway  56   Developed
## 4133                            Oman   8  Developing
## 4141                        Pakistan  14  Developing
## 4155                           Palau   5  Developing
## 4160         Palestinian Territories   6   Developed
## 4166                          Panama  32  Developing
## 4190                          Panama  32   Developed
## 4198                Papua New Guinea  18  Developing
## 4216                        Paraguay  28  Developing
## 4244                            Peru  43  Developing
## 4287                     Philippines  19  Developing
## 4306                          Poland  68   Developed
## 4315                          Poland  68  Developing
## 4374                        Portugal 101  Developing
## 4385                        Portugal 101   Developed
## 4475                           Qatar   9   Developed
## 4476                           Qatar   9  Developing
## 4484                         Romania  64  Developing
## 4491                         Romania  64   Developed
## 4548                          Russia  21  Developing
## 4558                          Russia  21   Developed
## 4569                          Rwanda  24  Developing
## 4593                           Samoa  16  Developing
## 4609                      San Marino   5  Developing
## 4614             São Tomé & Príncipe  17  Developing
## 4631                    Saudi Arabia  10  Developing
## 4637                    Saudi Arabia  10   Developed
```

```
## 4641                      Senegal  23  Developing
## 4664                       Serbia  17  Developing
## 4681                   Seychelles  19   Developed
## 4682                   Seychelles  19  Developing
## 4700                 Sierra Leone  19  Developing
## 4719                    Singapore  36   Developed
## 4728                    Singapore  36  Developing
## 4755                     Slovakia  71   Developed
## 4760                     Slovakia  71  Developing
## 4826                     Slovenia  74  Developing
## 4827                     Slovenia  74   Developed
## 4900              Solomon Islands  14  Developing
## 4914                      Somalia  17  Developing
## 4931                 South Africa  30  Developing
## 4961                  South Korea  23   Developed
## 4984                  South Sudan   4  Developing
## 4988                        Spain  97   Developed
## 4990                        Spain  97  Developing
## 5085                    Sri Lanka  11  Developing
## 5096            St. Kitts & Nevis  24  Developing
## 5120                    St. Lucia  25  Developing
## 5145     St. Vincent & Grenadines  24  Developing
## 5169                        Sudan  25  Developing
## 5194                     Suriname  18  Developing
## 5212                       Sweden 101   Developed
## 5313                  Switzerland  59   Developed
## 5372                        Syria  21  Developing
## 5393                       Taiwan   5  Developing
## 5398                   Tajikistan   9  Developing
## 5407                     Tanzania  30  Developing
## 5437                     Thailand  20  Developing
## 5457                  Timor-Leste   3  Developing
## 5460                         Togo  20  Developing
## 5480                        Tonga  12  Developing
## 5492            Trinidad & Tobago  28  Developing
## 5498            Trinidad & Tobago  28   Developed
## 5520                      Tunisia  28  Developing
## 5548                       Turkey  43  Developing
## 5554                       Turkey  43   Developed
## 5591                 Turkmenistan   5  Developing
## 5596                       Tuvalu  11  Developing
## 5607                       Uganda  23  Developing
## 5630                      Ukraine  21  Developing
## 5651         United Arab Emirates  12   Developed
## 5654         United Arab Emirates  12  Developing
## 5663               United Kingdom 126   Developed
## 5789                United States  19   Developed
## 5808                      Uruguay  37  Developing
## 5824                      Uruguay  37   Developed
## 5845                   Uzbekistan   7  Developing
## 5852                      Vanuatu  11  Developing
## 5863                    Venezuela  40  Developing
## 5903                      Vietnam  19  Developing
## 5922               Western Sahara   5  Developing
```

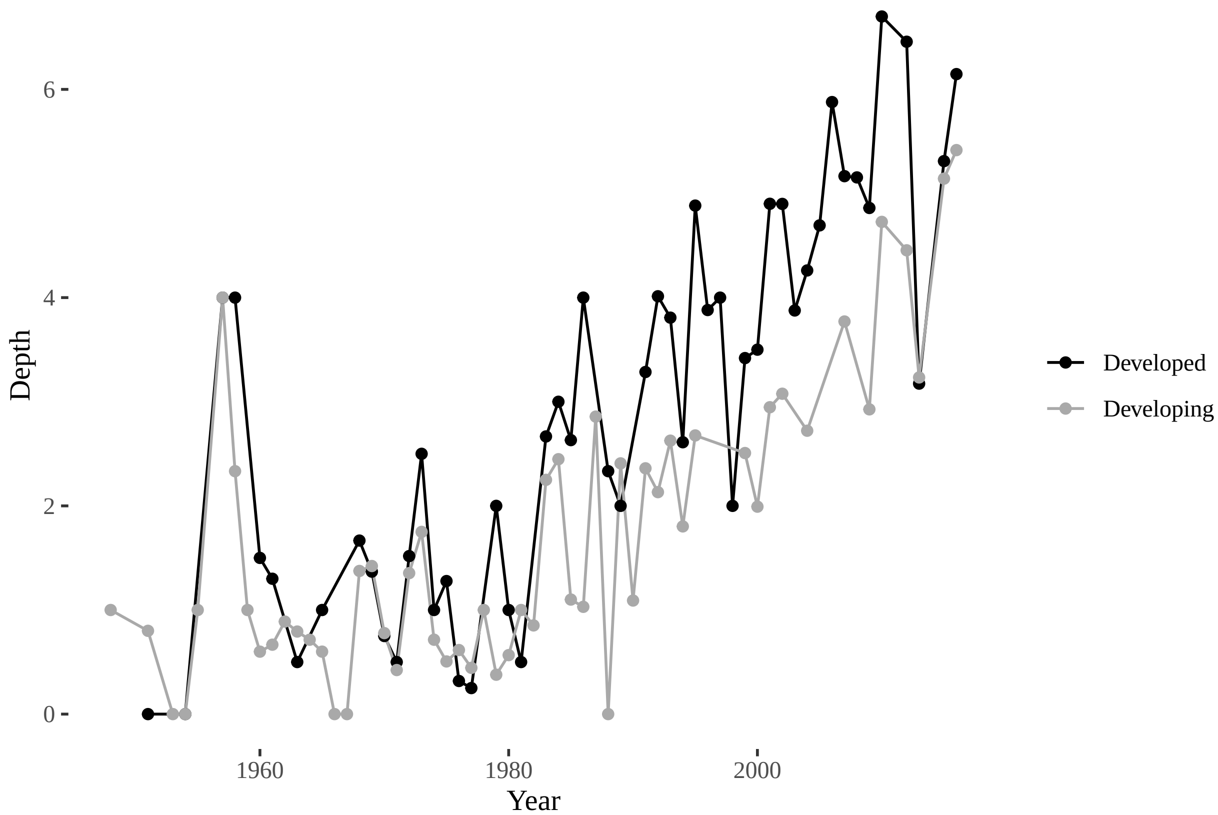
```
## 5927                          Yemen   3  Developing
## 5930                         Zambia  27  Developing
## 5957                       Zimbabwe  33  Developing
```

```
#Distribution of average depth index by developing-year
depth_dy = ggplot(country_yr_depth, aes(x = year, y = avg_depth, color = as.factor(developing3), group =
  geom_line()+
  geom_point()+
  scale_color_manual(values=c('Black','Dark Grey'))+
  ylab("Depth")+
  xlab("Year")+
  labs(color = "") +
  scale_fill_discrete(labels=c('Developed', 'Developing')) +
  theme_tufte()

# Distribution of average count by developing-year
count_dy = ggplot(country_yr_n, aes(x = year, y = n, color = as.factor(developing3), group = as.factor(d
  geom_line()+
  geom_point()+
  scale_color_manual(values=c('Black','Dark Grey'))+
  ylab("Count")+
  xlab("Year")+
  labs(color = "") +
  scale_fill_discrete(labels=c('Developed', 'Developing')) +
  theme_tufte()

count_dy
```



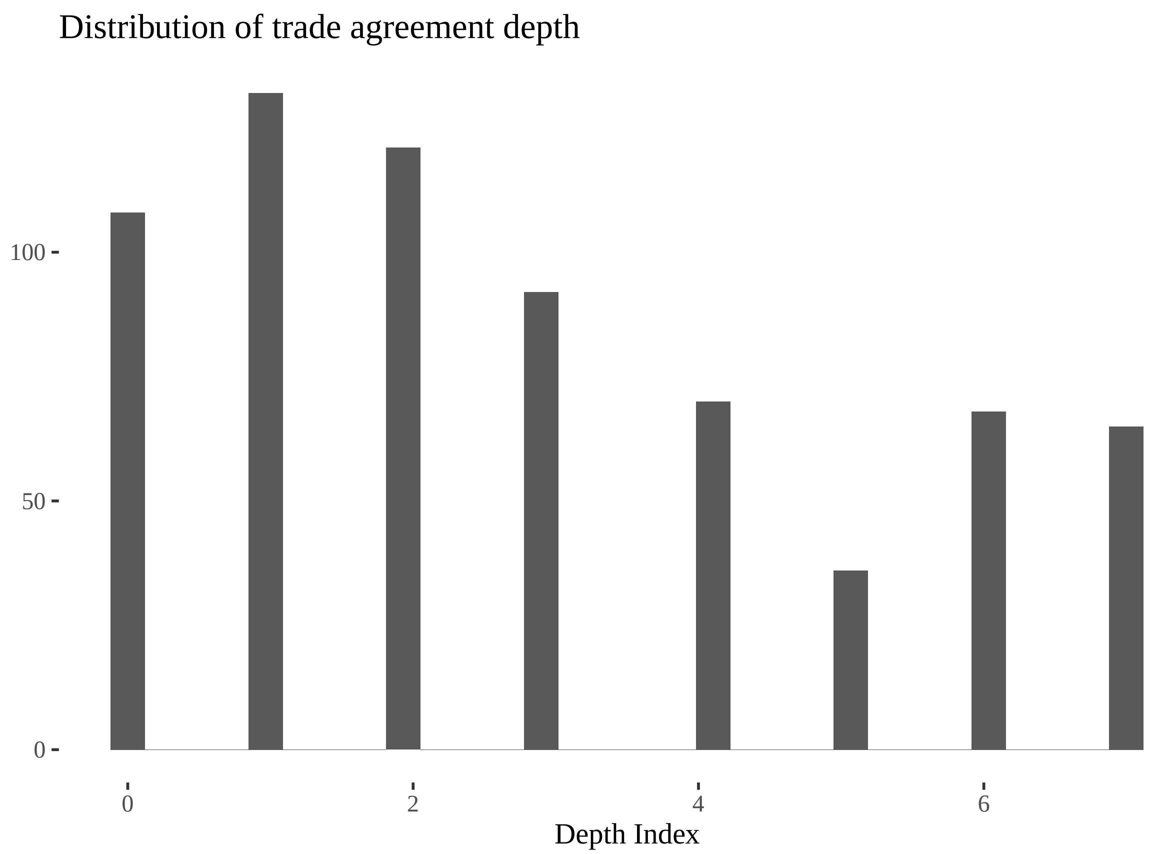
200
150
100
50
0
Count
1960
1980
2000
2020
Year
Developed
Developing


```
depth_dy
```



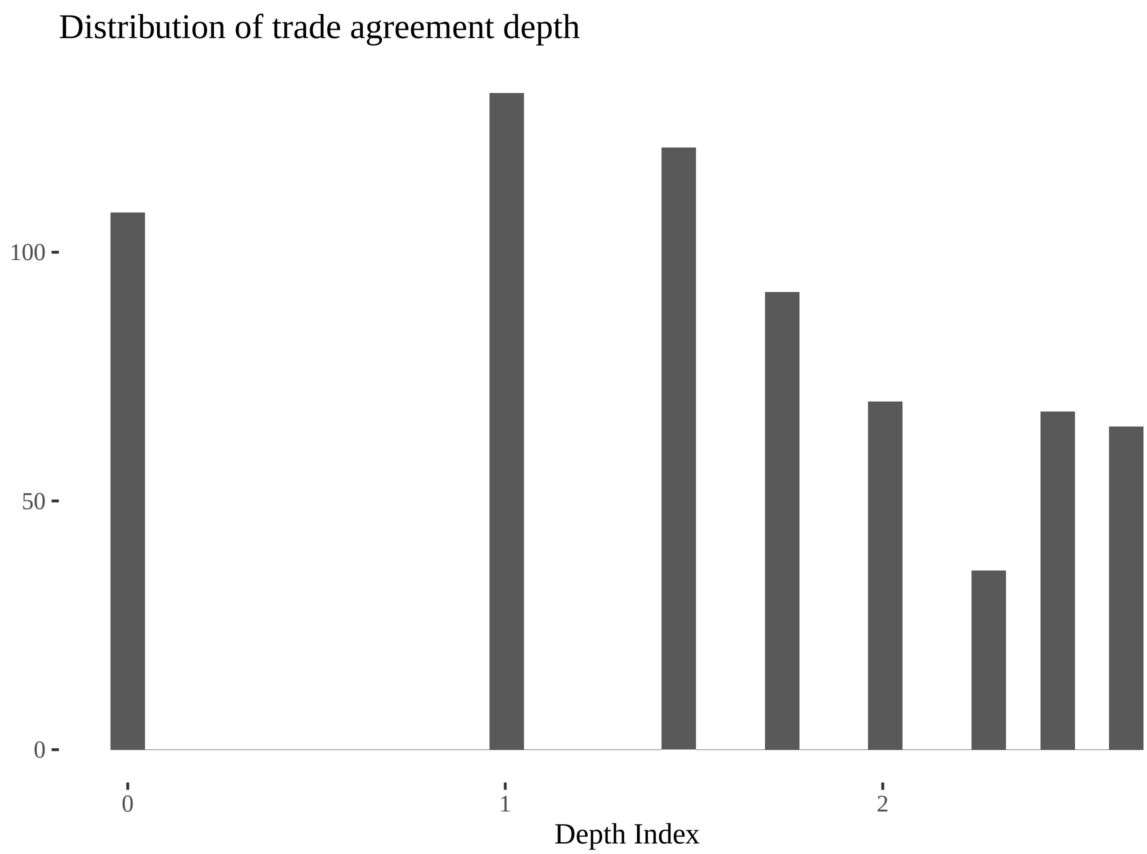

```r
# Get distribution of depth index
 depth_distrib = ggplot(desta, aes(x = depth_index)) +
  geom_histogram()+
  labs(title="Distribution of trade agreement depth", x = "Depth Index", y =  "")+
  theme_tufte()

# Get distribution of depth index (sqrt transformation)
depth_distrib_sqrted = ggplot(desta, aes(x = sqrt(depth_index))) +
  geom_histogram()+
  labs(title="Distribution of trade agreement depth", x = "Depth Index", y =  "")+
  theme_tufte()


depth_distrib
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```




```
depth_distrib_sqrted
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



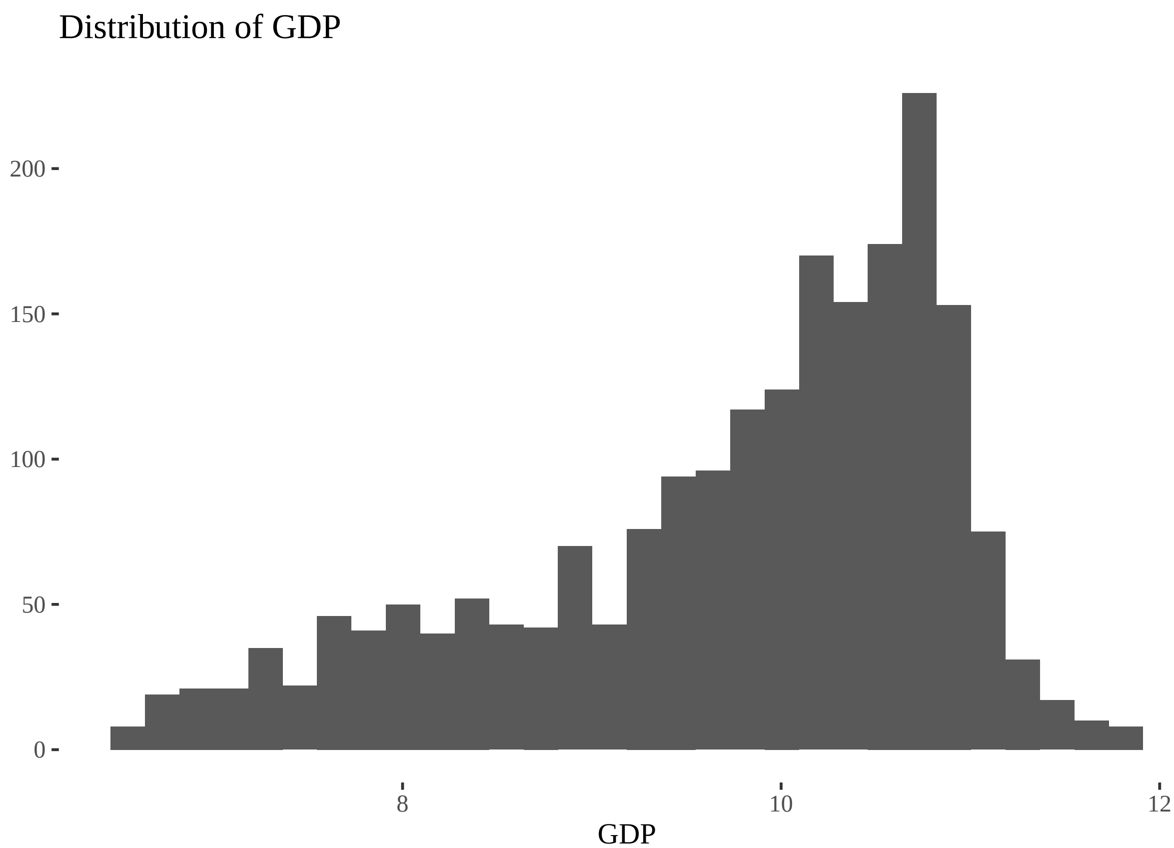
#Add 3 control variables

###1. Adding GDP

```
#Merge GDP with pta data
hdi_gdp = hdi %>%
  tidyr::pivot_longer(
    cols = starts_with("gnipc_"),
    names_to = "year",
    names_prefix = "gnipc_",
    values_to = "gdp"
  )

hdi_gdp = hdi_gdp %>% select(iso3n, year, gdp)
depth_data = merge(depth_data, hdi_gdp, by = c("iso3n", "year"), all.x = T)

# Get distribution of GDP
gdp_distrib = ggplot(depth_data, aes(x = log(gdp))) +
  geom_histogram()+
  labs(title="Distribution of GDP", x = "GDP", y =  "")+
  theme_tufte()
gdp_distrib
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 3911 rows containing non-finite values (stat_bin).
```

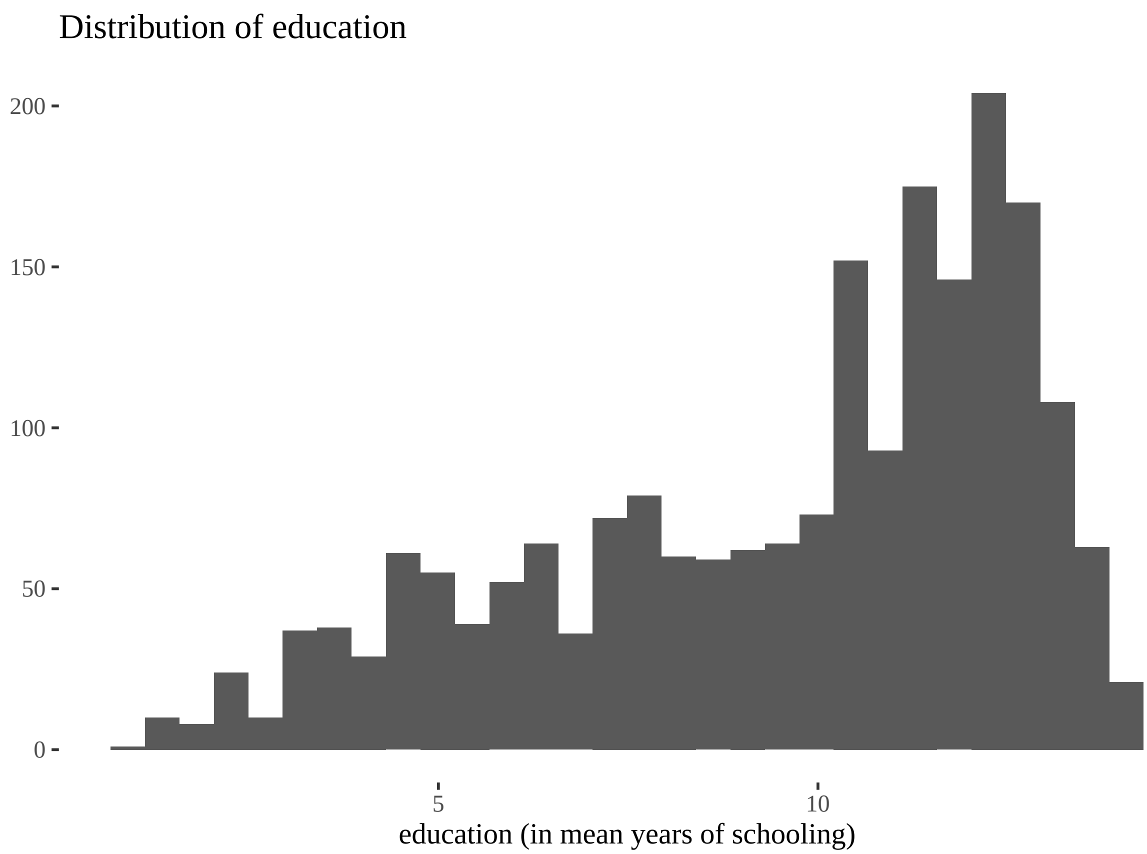


###2. Adding mean yrs of schooling (EDU)

```
#Merge school with PTA data
hdi_edu = hdi %>%
  tidyr::pivot_longer(
    cols = starts_with("mys_"),
    names_to = "year",
    names_prefix = "mys_",
    values_to = "edu"
  )

hdi_edu = hdi_edu %>% select(iso3n, year, edu)
depth_data = merge(depth_data, hdi_edu, by = c("iso3n", "year"), all.x = T)

edu_distrib = ggplot(depth_data, aes(x = edu)) +
  geom_histogram()+
  labs(title="Distribution of education", x = "education (in mean years of schooling)", y =  "")+
  theme_tufte()
edu_distrib
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 3924 rows containing non-finite values (stat_bin).
```



###3. Adding Regime Type

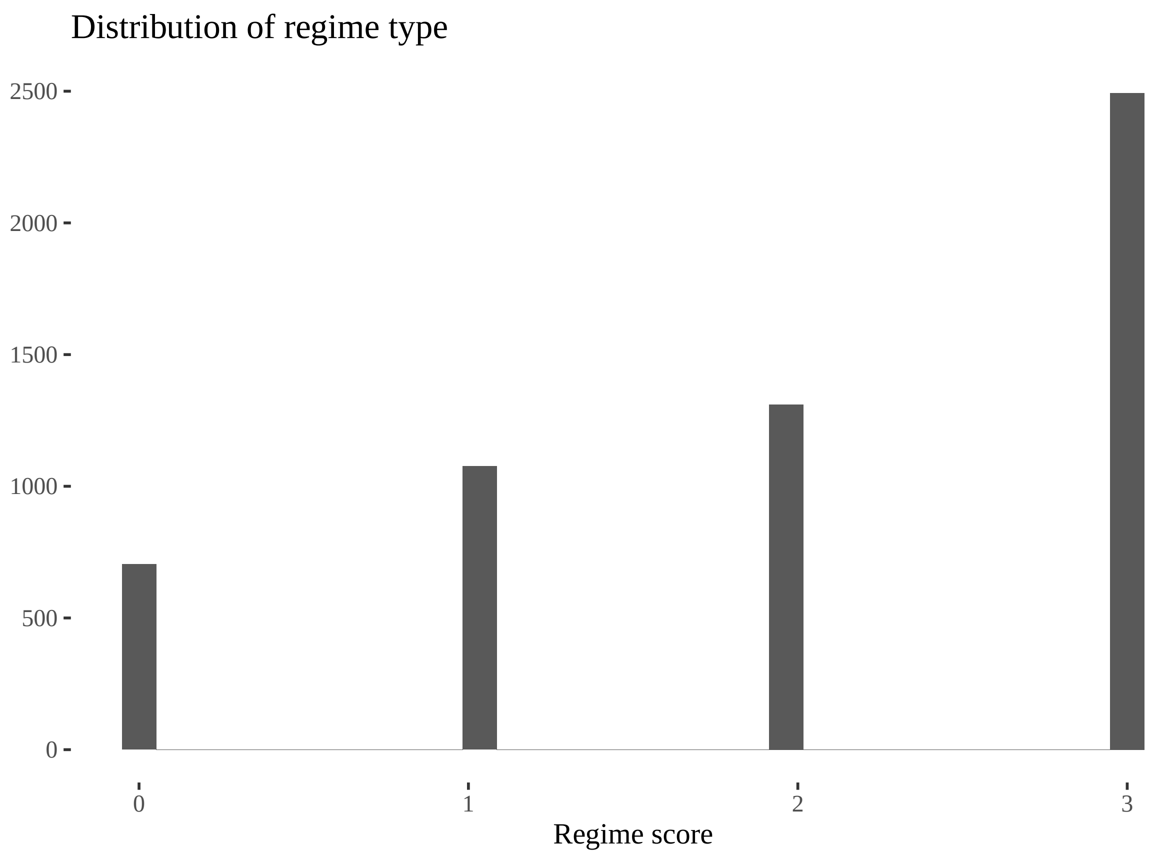
```
#Convert vdem country names to iso3n
vdem = vdem[,c("country_name", "country_text_id", "country_id", "year", "v2x_regime")]
vdem$iso3n = countrycode(vdem$country_text_id, origin = 'iso3c', destination = 'iso3n')
```

```
## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest, : Some va
```

```
#merge regime with pta data
depth_data = merge(depth_data, vdem[,c("iso3n", "v2x_regime", "year")], by = c("iso3n", "year"), all.x =

regime_distrib = ggplot(depth_data, aes(x = v2x_regime)) +
  geom_histogram()+
  labs(title="Distribution of regime type", x = "Regime score", y =  "")+
  theme_tufte()
regime_distrib
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 405 rows containing non-finite values (stat_bin).
```




```r
#Subset data down to 2005-2021 and non-NA values
depth_data2 = depth_data[depth_data$year > 2004,]
depth_data2 = depth_data2[!is.na(depth_data2$Approve),]
length(unique(depth_data2$iso3n))
```

```
## [1] 136
```

```r
length(unique(depth_data2[depth_data2$developing == 1,]$iso3n))
```

```
## [1] 95
```

Models

PTA Depth Level Analysis (w/year as independent variable)

Model 1: Depth w/Lags and logged Depth/GDP

```r
fe_mod1 <- plm(Approve ~ Lag(sqrt(depth_filled), shift = 2) +
                 developing +
                 Lag(log(gdp)) +
                 Lag(edu) +
```

```
                   Lag(v2x_regime) +
                   year +
                   Lag(sqrt(depth_filled), shift = 1)*developing,
                     data = subset(depth_data2),
                     index = c("iso3n", "year"),
                     model = "within",
                     effect = "twoways")
```

```
## Warning in pdata.frame(data, index): duplicate couples (id-time) in resulting pdata.frame
##  to find out which, use, e.g., table(index(your_pdataframe), useNA = "ifany")
```

```
summary(fe_mod1)
```

```
## Twoways effects Within Model
##
## Call:
## plm(formula = Approve ~ Lag(sqrt(depth_filled), shift = 2) +
##     developing + Lag(log(gdp)) + Lag(edu) + Lag(v2x_regime) +
##     year + Lag(sqrt(depth_filled), shift = 1) * developing, data = subset(depth_data2),
##     effect = "twoways", model = "within", index = c("iso3n",
##         "year"))
##
## Unbalanced Panel: n = 115, T = 1-41, N = 1051
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -28.60419  -4.65243  -0.23262   4.56816  30.93603
##
## Coefficients:
##                                               Estimate Std. Error t-value
## Lag(sqrt(depth_filled), shift = 2)             0.87279    0.63439  1.3758
## developing                                    13.74492    3.50334  3.9234
## Lag(log(gdp))                                  3.17080    1.49594  2.1196
## Lag(edu)                                      -0.53333    0.51770 -1.0302
## Lag(v2x_regime)                               -2.92486    1.05741 -2.7661
## Lag(sqrt(depth_filled), shift = 1)             0.19039    0.92860  0.2050
## developing:Lag(sqrt(depth_filled), shift = 1) -1.26553    1.26461 -1.0007
##                                                Pr(>|t|)
## Lag(sqrt(depth_filled), shift = 2)             0.169225
## developing                                    9.386e-05 ***
## Lag(log(gdp))                                  0.034308 *
## Lag(edu)                                       0.303191
## Lag(v2x_regime)                                0.005788 **
## Lag(sqrt(depth_filled), shift = 1)             0.837595
## developing:Lag(sqrt(depth_filled), shift = 1)  0.317225
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    78735
## Residual Sum of Squares: 75947
## R-Squared:      0.035404
## Adj. R-Squared: -0.10812
## F-statistic: 4.7924 on 7 and 914 DF, p-value: 2.6076e-05
```

Country-Count Level Analysis (w/year as independent variable)

```
#Count all PTAs for each country-year (to get change)
desta_dyads1 = desta_dyads1 %>% rename(iso3n = iso1)
pta_count = desta_dyads1 %>% dplyr::group_by(iso3n, year) %>% dplyr::summarise(count_pta = n())
```

```
## `summarise()` has grouped output by 'iso3n'. You can override using the
## `.groups` argument.
```

```
#Count all PTA withdrawals for each country-year
withdrawals = withdrawals %>% rename(iso3n = iso1)
withdrawals_count = withdrawals %>% dplyr::group_by(iso3n, year) %>% dplyr::summarise(count_withdrawals
```

```
## `summarise()` has grouped output by 'iso3n'. You can override using the
## `.groups` argument.
```

```
#Merge withdrawals column to desta_country_yr
count_data = merge(pta_count, withdrawals_count, by = c("year", "iso3n"), all.x = TRUE, all.y = TRUE)

#Add missing years
count_data <- setDT(count_data)[CJ(iso3n=iso3n, year=seq(min(year), 2021), unique=TRUE),
              on=.(iso3n, year)]

#fill PTA count with 0
count_data <- count_data %>% mutate(count_pta = ifelse(is.na(count_pta), 0, count_pta),
                                    count_withdrawals = ifelse(is.na(count_withdrawals), 0

#Add the PTAs
count_data = count_data %>% group_by(iso3n) %>%
    mutate(across(count_pta, ~ accumulate(., `+`)))
count_data = count_data %>% group_by(iso3n) %>%
    mutate(across(count_withdrawals, ~ accumulate(., `+`)))

#Create total PTA column
count_data$total_ptas = count_data$count_pta - count_data$count_withdrawals

## Merge approval ratings w/data ##
count_data = merge(count_data, gallup, by = c("iso3n", "year"), all.x = TRUE)

## Merge HDI w/data ##
count_data = merge(count_data, hdi_hdi, by = c("iso3n", "year"), all.x = T) %>% mutate(developing = case
                                          hdi_score>= 0.8 ~ 0))

## Merge Control variables w/data ##
count_data = merge(count_data, hdi_gdp, by = c("iso3n", "year"), all.x = T)
count_data = merge(count_data, hdi_edu, by = c("iso3n", "year"), all.x = T)
count_data = merge(count_data, vdem[,c("iso3n", "v2x_regime", "year")], by = c("iso3n", "year"), all.x =

# Assign NAs in developing to a 1 or 0 based on current values
count_data$developing2 = nafill(count_data$developing, type = "nocb")
count_data$developing3 = ifelse(count_data$developing2 == 0, "Developed", "Developing")
```

```r
#Remove duplicate country-yr
count_data$unique_id <- paste(count_data$iso3n,count_data$year) # concatenate to make unique ID
count_data$duplicate = duplicated(count_data$unique_id) # generate the duplicate variable
count_data = count_data[count_data$duplicate != "TRUE", ]

count_data$duplicate = duplicated(count_data$unique_id) # generate the duplicate variable

#Subset data down to 2005-2021 and non-NA values
count_data2 = count_data[count_data$year > 2004,]
count_data2 = count_data2[!is.na(count_data2$Approve),]
```

Model 1: Change in PTA membership count

```r
#Build FE model on PTA count
fe_mod2 <- plm(Approve ~ Lag(total_ptas, shift = 2) +
                 Lag(log(gdp)) +
                 Lag(edu) +
                 Lag(v2x_regime) +
                 year +
                 Lag(total_ptas, shift = 2)*developing,
                   data = count_data2,
                   index = c("iso3n", "year"),
                   model = "within",
                   effect = "twoways")
summary(fe_mod2)
```

```
## Twoways effects Within Model
##
## Call:
## plm(formula = Approve ~ Lag(total_ptas, shift = 2) + Lag(log(gdp)) +
##     Lag(edu) + Lag(v2x_regime) + year + Lag(total_ptas, shift = 2) *
##     developing, data = count_data2, effect = "twoways", model = "within",
##     index = c("iso3n", "year"))
##
## Unbalanced Panel: n = 150, T = 1-16, N = 1789
##
## Residuals:
##     Min.  1st Qu.   Median  3rd Qu.     Max.
## -46.1413  -6.2951   0.0000   6.2754  42.3982
##
## Coefficients:
##                                         Estimate Std. Error t-value Pr(>|t|)
## Lag(total_ptas, shift = 2)             0.1122734  0.0269712  4.1627 3.31e-05
## Lag(log(gdp))                          0.8398364  1.0287689  0.8164 0.414420
## Lag(edu)                              -0.1925297  0.3393447 -0.5674 0.570550
## Lag(v2x_regime)                       -1.5967628  0.7182508 -2.2231 0.026345
## developing                             5.3332493  1.9434561  2.7442 0.006133
## Lag(total_ptas, shift = 2):developing -0.0074945  0.0439901 -0.1704 0.864742
##
## Lag(total_ptas, shift = 2)            ***
## Lag(log(gdp))
```

```
## Lag(edu)
## Lag(v2x_regime)                          *
## developing                               **
## Lag(total_ptas, shift = 2):developing
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    197850
## Residual Sum of Squares: 193170
## R-Squared:      0.023672
## Adj. R-Squared: -0.078909
## F-statistic: 6.53836 on 6 and 1618 DF, p-value: 7.7985e-07
```

##Country Count-Level Analysis II: Change in deeper agreements

Model 2: Change in (deeper) PTA membership count

```
#obtain a list of PTAs that have a depth > mean of depth
deep_ptas = subset(desta, desta$depth_index > mean(desta$depth_index))$base_treaty


#Subset desta dyads and withdrawals based on if the base_treaty is greater than mean in desta
desta_dyads2 = desta_dyads1 %>% filter(number %in% deep_ptas)

#Count all PTAs for each country-year (to get change)
pta_count = desta_dyads2 %>% dplyr::group_by(iso3n, year) %>% dplyr::summarise(count_pta = n())
```

```
## `summarise()` has grouped output by 'iso3n'. You can override using the
## `.groups` argument.
```

```
#Count all PTA withdrawals for each country-year
withdrawals2 = withdrawals %>% filter(number %in% deep_ptas)
withdrawals_count = withdrawals2 %>% dplyr::group_by(iso3n, year) %>% dplyr::summarise(count_withdrawals
```

```
## `summarise()` has grouped output by 'iso3n'. You can override using the
## `.groups` argument.
```

```
#Merge withdrawals column to desta_country_yr
deep_count_data = merge(pta_count, withdrawals_count, by = c("year", "iso3n"), all.x = TRUE, all.y = TRU

#Add missing years
deep_count_data <- setDT(deep_count_data)[CJ(iso3n=iso3n, year=seq(min(year), 2021), unique=TRUE),
              on=.(iso3n, year), roll=F]

#fill PTA count with 0
deep_count_data <- deep_count_data %>% mutate(count_pta = ifelse(is.na(count_pta), 0, count_pta),
                                              count_withdrawals = ifelse(is.na(count_withdrawals),

#Add the PTAs
deep_count_data = deep_count_data %>% group_by(iso3n) %>%
    mutate(across(count_pta, ~ accumulate(., `+`)))
```

```r
deep_count_data = deep_count_data %>% group_by(iso3n) %>%
    mutate(across(count_withdrawals, ~ accumulate(., `+`)))

#Create total PTA column
deep_count_data$total_ptas = deep_count_data$count_pta - deep_count_data$count_withdrawals

## Merge approval ratings w/data ##
deep_count_data = merge(deep_count_data, gallup, by = c("iso3n", "year"), all.x = TRUE)

## Merge HDI w/data ##
deep_count_data = merge(deep_count_data, hdi_hdi, by = c("iso3n", "year"), all.x = T) %>% mutate(develop
                                              hdi_score>= 0.8 ~ 0))
# Assign NAs in developing to a 1 or 0 based on current values
deep_count_data$developing2 = nafill(deep_count_data$developing, type = "nocb")
deep_count_data$developing3 = ifelse(deep_count_data$developing2 == 0, "Developed", "Developing")


## Merge Control variables w/data ##
deep_count_data = merge(deep_count_data, hdi_gdp, by = c("iso3n", "year"), all.x = T)
deep_count_data = merge(deep_count_data, hdi_edu, by = c("iso3n", "year"), all.x = T)
deep_count_data = merge(deep_count_data, vdem[,c("iso3n", "v2x_regime", "year")], by = c("iso3n", "year

#Remove duplicate country-yr
deep_count_data$unique_id <- paste(deep_count_data$iso3n,deep_count_data$year) # concatenate to make un
deep_count_data$duplicate = duplicated(deep_count_data$unique_id) # generate the duplicate variable
deep_count_data = deep_count_data[deep_count_data$duplicate != "TRUE", ]

#Remove NAs
deep_count_data = deep_count_data[!is.na(deep_count_data$Approve),]

#Subset to 2005-2021 and rows without missing data
deep_count_data2 = deep_count_data[deep_count_data$year > 2004,]
deep_count_data2 = deep_count_data2[!is.na(deep_count_data2$Approve),]
```

```r
#Build FE model on PTA count
fe_mod3 <- plm(Approve ~
                 Lag((total_ptas), shift = 2) +
                 Lag(log(gdp)) +
                 Lag(edu) +
                 Lag(v2x_regime) +
                 year +
                 Lag((total_ptas), shift = 2)*developing,
                   data = deep_count_data2,
                   index = c("iso3n", "year"),
                   model = "within",
                   effect = "twoways")
summary(fe_mod3)
```

```
## Twoways effects Within Model
##
## Call:
## plm(formula = Approve ~ Lag((total_ptas), shift = 2) + Lag(log(gdp)) +
##     Lag(edu) + Lag(v2x_regime) + year + Lag((total_ptas), shift = 2) *
```

```
##     developing, data = deep_count_data2, effect = "twoways",
##     model = "within", index = c("iso3n", "year"))
##
## Unbalanced Panel: n = 138, T = 1-16, N = 1670
##
## Residuals:
##     Min.  1st Qu.   Median  3rd Qu.     Max.
## -46.3328  -6.6103   0.0000   6.4403  43.3067
##
## Coefficients:
##                                          Estimate Std. Error t-value  Pr(>|t|)
## Lag((total_ptas), shift = 2)             0.175624   0.049684  3.5348 0.0004204
## Lag(log(gdp))                            2.502682   1.265465  1.9777 0.0481466
## Lag(edu)                                -0.956728   0.424180 -2.2555 0.0242460
## Lag(v2x_regime)                         -1.619210   0.750670 -2.1570 0.0311614
## developing                               5.393012   1.733275  3.1115 0.0018965
## Lag((total_ptas), shift = 2):developing -0.029620   0.070736 -0.4187 0.6754643
##
## Lag((total_ptas), shift = 2)            ***
## Lag(log(gdp))                           *
## Lag(edu)                                *
## Lag(v2x_regime)                         *
## developing                              **
## Lag((total_ptas), shift = 2):developing
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    189670
## Residual Sum of Squares: 184590
## R-Squared:      0.026776
## Adj. R-Squared: -0.074991
## F-statistic: 6.92848 on 6 and 1511 DF, p-value: 2.8135e-07
```

Tests for fixed effects

```r
#Test if we need to control for individual effects
plmtest(fe_mod1, effect="individual", type="bp")
```

```
## Warning in pdata.frame(data, index): duplicate couples (id-time) in resulting pdata.frame
##  to find out which, use, e.g., table(index(your_pdataframe), useNA = "ifany")
```

```
##
##  Lagrange Multiplier Test - (Breusch-Pagan)
##
## data:  Approve ~ Lag(sqrt(depth_filled), shift = 2) + developing + Lag(log(gdp)) +  ...
## chisq = 2264.6, df = 1, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

```r
plmtest(fe_mod2, effect="individual", type="bp")
```

```
##
##  Lagrange Multiplier Test - (Breusch-Pagan)
##
## data:  Approve ~ Lag(total_ptas, shift = 2) + Lag(log(gdp)) + Lag(edu) +  ...
## chisq = 3891.5, df = 1, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

```
plmtest(fe_mod3, effect="individual", type="bp")
```

```
##
##  Lagrange Multiplier Test - (Breusch-Pagan)
##
## data:  Approve ~ Lag((total_ptas), shift = 2) + Lag(log(gdp)) + Lag(edu) +  ...
## chisq = 3521.5, df = 1, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

```
#Test if we need to control for time effects
plmtest(fe_mod1, effect="time", type="bp")
```

```
## Warning in pdata.frame(data, index): duplicate couples (id-time) in resulting pdata.frame
##  to find out which, use, e.g., table(index(your_pdataframe), useNA = "ifany")
```

```
##
##  Lagrange Multiplier Test - time effects (Breusch-Pagan)
##
## data:  Approve ~ Lag(sqrt(depth_filled), shift = 2) + developing + Lag(log(gdp)) +  ...
## chisq = 6.2849, df = 1, p-value = 0.01218
## alternative hypothesis: significant effects
```

```
plmtest(fe_mod2, effect="time", type="bp")
```

```
##
##  Lagrange Multiplier Test - time effects (Breusch-Pagan)
##
## data:  Approve ~ Lag(total_ptas, shift = 2) + Lag(log(gdp)) + Lag(edu) +  ...
## chisq = 7.8443, df = 1, p-value = 0.005098
## alternative hypothesis: significant effects
```

```
plmtest(fe_mod3, effect="time", type="bp")
```

```
##
##  Lagrange Multiplier Test - time effects (Breusch-Pagan)
##
## data:  Approve ~ Lag((total_ptas), shift = 2) + Lag(log(gdp)) + Lag(edu) +  ...
## chisq = 7.8658, df = 1, p-value = 0.005038
## alternative hypothesis: significant effects
```

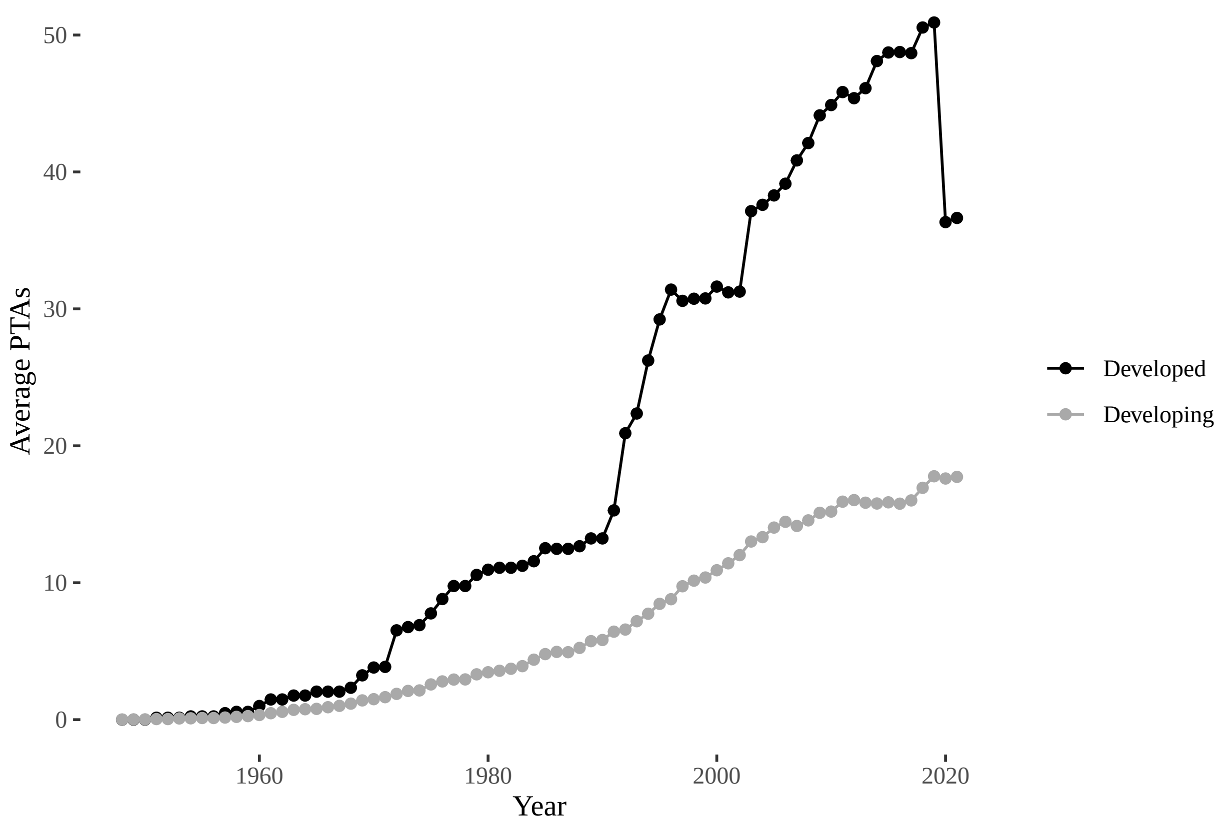
Summary Statistics Pt 2

Average PTA memberships per country over time b/w developed and developing

```
#Count number of PTAs for each country type-year (developing/not developing)

country_yr_n = count_data %>% group_by(developing3,year)  %>% summarise(n = mean(total_ptas))
```

```
## `summarise()` has grouped output by 'developing3'. You can override using the
## `.groups` argument.
```

```
country_yr_n = subset(country_yr_n, !is.na(developing3)) #Remove NA rows
country_yr_n = subset(country_yr_n, !is.na(n))

# Average PTAs per year between 1990-2021 by developing country
count_dy = ggplot(country_yr_n, aes(x = year, y = n, color = as.factor(developing3), group = as.factor(
  geom_line()+
  geom_point()+
  scale_color_manual(values=c('Black','Dark Grey'))+
  ylab("Average PTAs")+
  xlab("Year")+
  labs(color = "") +
  theme_tufte()
count_dy
```



Descriptive Stats Table

```
count_data_ds = subset(count_data, year < 2022 & year > 2004) %>% select(iso3n, Approve, total_ptas, gdp
summary_table = psych::describe(count_data_ds)


count_data_developing = subset(count_data_ds, developing == 1)
count_data_developed = subset(count_data_ds, developing == 0)

summary_table_developing = psych::describe(count_data_developing)
summary_table_developed = psych::describe(count_data_developed)

summary_table
```

```
##            vars    n     mean       sd   median  trimmed      mad    min
## iso3n         1 3757   432.67   252.26   430.00   431.05   323.21   4.00
## Approve       2 1812    46.83    19.08    45.00    45.94    19.27   4.00
## total_ptas    3 3757    23.49    22.91    19.00    19.31    13.34  -1.00
## gdp           4 3278 18823.49 20456.01 11421.91 15120.89 12284.09 731.79
## edu           5 3220     8.36     3.24     8.62     8.49     3.96   0.56
## v2x_regime    6 2969     1.62     0.98     2.00     1.65     1.48   0.00
## developing    7 3216     0.70     0.46     1.00     0.74     0.00   0.00
##                  max     range  skew kurtosis     se
## iso3n         900.00    896.00  0.04    -1.18   4.12
## Approve        99.00     95.00  0.39    -0.36   0.45
## total_ptas    129.00    130.00  1.94     3.98   0.37
## gdp        146829.70 146097.91  1.88     4.44 357.29
## edu            14.13     13.57 -0.28    -0.97   0.06
## v2x_regime      3.00      3.00 -0.04    -1.03   0.02
## developing      1.00      1.00 -0.85    -1.28   0.01
```

```
summary_table_developing
```

```
##            vars    n    mean      sd  median trimmed     mad    min      max
## iso3n         1 2237  436.23  257.67  430.00  434.85  335.07   4.00   894.00
## Approve       2 1108   48.41   19.71   47.00   47.65   20.76   4.00    99.00
## total_ptas    3 2237   18.68    9.41   19.00   18.24    8.90   0.00    53.00
## gdp           4 2237 8170.76 6602.02 6535.79 7383.89 6253.54 731.79 87527.32
## edu           5 2237    6.98    2.79    7.11    7.00    3.24   0.56    12.88
## v2x_regime    6 2010    1.34    0.75    1.00    1.37    1.48   0.00     3.00
## developing    7 2237    1.00    0.00    1.00    1.00    0.00   1.00     1.00
##               range  skew kurtosis     se
## iso3n        890.00  0.04    -1.22   5.45
## Approve       95.00  0.32    -0.42   0.59
## total_ptas    53.00  0.53     0.46   0.20
## gdp        86795.53  2.42    18.04 139.59
## edu           12.32 -0.04    -0.91   0.06
## v2x_regime     3.00  0.01    -0.40   0.02
## developing     0.00   NaN      NaN   0.00
```

```r
summary_table_developed
```

```
##            vars   n     mean       sd   median  trimmed      mad      min
## iso3n         1 979   412.50   245.96   410.00   410.98   311.35     8.00
## Approve       2 687    44.31    17.89    42.00    43.35    17.79     7.00
## total_ptas    3 979    45.91    31.74    41.00    43.40    35.58     1.00
## gdp           4 979 43156.05 20161.69 40275.25 40752.34 18181.38 12802.15
## edu           5 979    11.52     1.52    11.84    11.66     1.44     6.22
## v2x_regime    6 903     2.29     1.06     3.00     2.49     0.00     0.00
## developing    7 979     0.00     0.00     0.00     0.00     0.00     0.00
##                  max     range  skew kurtosis     se
## iso3n         858.00    850.00  0.04    -1.14   7.86
## Approve        98.00     91.00  0.48    -0.29   0.68
## total_ptas    129.00    128.00  0.57    -0.77   1.01
## gdp        146829.70 134027.55  1.51     3.82 644.37
## edu            14.13      7.91 -0.78     0.20   0.05
## v2x_regime      3.00      3.00 -1.22     0.01   0.04
## developing      0.00      0.00   NaN      NaN   0.00
```

```r
#Obtain average number of deep trade agreements and by country-type
deep_count_data_ds = subset(deep_count_data, year < 2022 & year > 2004) %>% select(iso3n, total_ptas, de
deep_summary_table = psych::describe(deep_count_data_ds)
deep_count_data_developing = subset(deep_count_data_ds, developing == 1)
deep_count_data_developed = subset(deep_count_data_ds, developing == 0)
deep_summary_table_developing = psych::describe(deep_count_data_developing)
deep_summary_table_developed = psych::describe(deep_count_data_developed)
deep_summary_table
```

```
##            vars    n   mean     sd median trimmed    mad min max range  skew
## iso3n         1 1693 428.68 245.87    410  424.54 305.42   8 894   886  0.14
## deep_ptas     2 1693  13.28  15.30      6   10.23   5.93   0  58    58  1.53
## developing    3 1676   0.59   0.49      1    0.61   0.00   0   1     1 -0.37
##            kurtosis   se
## iso3n         -1.13 5.98
## deep_ptas      1.12 0.37
## developing    -1.87 0.01
```

```r
deep_summary_table_developing
```

```
##            vars   n   mean     sd median trimmed    mad min max range skew
## iso3n         1 989 438.49 251.10    430  434.72 320.24   8 894   886 0.12
## deep_ptas     2 989   4.93   3.85      4    4.33   2.97   0  28    28 1.88
## developing    3 989   1.00   0.00      1    1.00   0.00   1   1     0  NaN
##            kurtosis   se
## iso3n         -1.18 7.98
## deep_ptas      4.39 0.12
## developing      NaN 0.00
```

```r
deep_summary_table_developed
```

```
##            vars   n   mean     sd median trimmed    mad min max range skew
```

```
## iso3n         1 687 418.86 237.12    392  416.78 280.21   8 858    850 0.12
## deep_ptas     2 687  25.53  17.42     20   24.79  19.27   1  58     57 0.35
## developing    3 687   0.00   0.00      0    0.00   0.00   0   0      0  NaN
##            kurtosis   se
## iso3n         -1.05 9.05
## deep_ptas     -1.28 0.66
## developing      NaN 0.00
```

Robustness Check 1: RE and Hausman Test

```
re_mod1 <- plm(Approve ~ Lag(sqrt(depth_filled), shift = 2) +
                  developing +
                  Lag(log(gdp)) +
                  Lag(edu) +
                  Lag(v2x_regime) +
                  year +
                  Lag(sqrt(depth_filled), shift = 2)*developing,
                    data = depth_data2,
                    index = c("iso3n", "year"),
                    model = "random")
```

```
## Warning in pdata.frame(data, index): duplicate couples (id-time) in resulting pdata.frame
##  to find out which, use, e.g., table(index(your_pdataframe), useNA = "ifany")
```

```
summary(re_mod1)
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Call:
## plm(formula = Approve ~ Lag(sqrt(depth_filled), shift = 2) +
##     developing + Lag(log(gdp)) + Lag(edu) + Lag(v2x_regime) +
##     year + Lag(sqrt(depth_filled), shift = 2) * developing, data = depth_data2,
##     model = "random", index = c("iso3n", "year"))
##
## Unbalanced Panel: n = 127, T = 1-45, N = 1187
##
## Effects:
##                   var std.dev share
## idiosyncratic  83.434   9.134 0.329
## individual    170.512  13.058 0.671
## theta:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4268  0.7599  0.8491  0.8043  0.8641  0.8963
##
## Residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -30.827  -6.160  -0.589  -0.279   5.141  30.623
##
## Coefficients:
```

```
##                                              Estimate Std. Error z-value
## (Intercept)                                  14.25903    9.36064  1.5233
## Lag(sqrt(depth_filled), shift = 2)            1.24273    0.89539  1.3879
## developing                                   11.27945    3.08129  3.6606
## Lag(log(gdp))                                 4.40458    1.25375  3.5131
## Lag(edu)                                     -0.57598    0.41904 -1.3745
## Lag(v2x_regime)                              -2.56999    0.86006 -2.9881
## year2007                                     -9.74463    1.87601 -5.1943
## year2008                                     -7.59034    1.79000 -4.2404
## year2009                                     -2.77415    2.02635 -1.3690
## year2010                                     -6.25250    2.23840 -2.7933
## year2011                                     -6.69832    1.83714 -3.6461
## year2012                                     -9.58504    1.88533 -5.0840
## year2013                                    -11.49920    2.01065 -5.7191
## year2014                                     -5.65009    1.82629 -3.0938
## year2015                                     -7.26638    2.05320 -3.5390
## year2016                                     -5.87405    1.85078 -3.1738
## year2017                                     -2.93015    2.02520 -1.4468
## year2018                                     -4.55991    1.93813 -2.3527
## year2019                                     -6.75585    1.98394 -3.4053
## year2020                                     -0.50455    2.47653 -0.2037
## year2021                                     -0.50266    9.69908 -0.0518
## Lag(sqrt(depth_filled), shift = 2):developing  -0.77513    1.21359 -0.6387
##                                              Pr(>|z|)
## (Intercept)                                 0.1276844
## Lag(sqrt(depth_filled), shift = 2)          0.1651634
## developing                                  0.0002516 ***
## Lag(log(gdp))                               0.0004429 ***
## Lag(edu)                                    0.1692809
## Lag(v2x_regime)                             0.0028068 **
## year2007                                    2.054e-07 ***
## year2008                                    2.231e-05 ***
## year2009                                    0.1709880
## year2010                                    0.0052174 **
## year2011                                    0.0002663 ***
## year2012                                    3.696e-07 ***
## year2013                                    1.071e-08 ***
## year2014                                    0.0019764 **
## year2015                                    0.0004016 ***
## year2016                                    0.0015044 **
## year2017                                    0.1479416
## year2018                                    0.0186359 *
## year2019                                    0.0006610 ***
## year2020                                    0.8385626
## year2021                                    0.9586674
## Lag(sqrt(depth_filled), shift = 2):developing 0.5230154
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    150370
## Residual Sum of Squares: 104060
## R-Squared:      0.31011
## Adj. R-Squared: 0.29767
## Chisq: 139.401 on 21 DF, p-value: < 2.22e-16
```

```
phtest(fe_mod1, re_mod1)
```

```
##
##  Hausman Test
##
## data:  Approve ~ Lag(sqrt(depth_filled), shift = 2) + developing + Lag(log(gdp)) +  ...
## chisq = 5.5147, df = 5, p-value = 0.3563
## alternative hypothesis: one model is inconsistent
```

```
#Build RE model on PTA count
re_mod2 <- plm(Approve ~ Lag(total_ptas, shift = 2) +
                 Lag(log(gdp)) +
                 Lag(edu) +
                 Lag(v2x_regime) +
                 year +
                 Lag(total_ptas, shift = 2)*developing,
                   index = c("iso3n", "year"),
                   data = count_data2,
                   model = "random")
summary(re_mod2)
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Call:
## plm(formula = Approve ~ Lag(total_ptas, shift = 2) + Lag(log(gdp)) +
##     Lag(edu) + Lag(v2x_regime) + year + Lag(total_ptas, shift = 2) *
##     developing, data = count_data2, model = "random", index = c("iso3n",
##     "year"))
##
## Unbalanced Panel: n = 150, T = 1-16, N = 1789
##
## Effects:
##                  var std.dev share
## idiosyncratic 119.39   10.93 0.369
## individual    204.33   14.29 0.631
## theta:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3927  0.7926  0.8064  0.7909  0.8123  0.8123
##
## Residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -38.622  -7.435  -0.300  -0.204   7.287  40.893
##
## Coefficients:
##                                      Estimate Std. Error z-value  Pr(>|z|)
## (Intercept)                         46.809586   7.092730  6.5997 4.121e-11
## Lag(total_ptas, shift = 2)           0.105946   0.025567  4.1439 3.415e-05
## Lag(log(gdp))                        0.527763   0.976794  0.5403 0.5889896
## Lag(edu)                            -0.117621   0.317865 -0.3700 0.7113569
## Lag(v2x_regime)                     -1.975418   0.688075 -2.8709 0.0040926
## year2007                            -6.475900   2.011671 -3.2192 0.0012856
```

```
## year2008                              -7.977187   1.929976 -4.1333 3.576e-05
## year2009                              -4.618786   1.947631 -2.3715 0.0177165
## year2010                              -2.600020   1.933626 -1.3446 0.1787435
## year2011                              -4.315251   1.891212 -2.2817 0.0225048
## year2012                              -6.531863   1.911761 -3.4167 0.0006339
## year2013                              -7.411835   1.907815 -3.8850 0.0001023
## year2014                              -5.008581   1.897934 -2.6390 0.0083160
## year2015                              -5.921453   1.907305 -3.1046 0.0019052
## year2016                              -7.044292   1.912568 -3.6832 0.0002304
## year2017                              -4.835093   1.904151 -2.5392 0.0111094
## year2018                              -5.479969   1.927653 -2.8428 0.0044716
## year2019                              -5.537924   1.926663 -2.8744 0.0040485
## year2020                              -3.059051   1.995993 -1.5326 0.1253754
## year2021                              -5.775588   1.977068 -2.9213 0.0034858
## developing                             5.442108   1.849826  2.9420 0.0032615
## Lag(total_ptas, shift = 2):developing  0.002379   0.042700  0.0557 0.9555695
##
## (Intercept)                           ***
## Lag(total_ptas, shift = 2)            ***
## Lag(log(gdp))
## Lag(edu)
## Lag(v2x_regime)                       **
## year2007                              **
## year2008                              ***
## year2009                              *
## year2010
## year2011                              *
## year2012                              ***
## year2013                              ***
## year2014                              **
## year2015                              **
## year2016                              ***
## year2017                              *
## year2018                              **
## year2019                              **
## year2020
## year2021                              **
## developing                            **
## Lag(total_ptas, shift = 2):developing
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    243920
## Residual Sum of Squares: 217390
## R-Squared:      0.11204
## Adj. R-Squared: 0.10148
## Chisq: 83.9012 on 21 DF, p-value: 1.781e-09
```

```
phtest(fe_mod2, re_mod2)
```

```
##
##  Hausman Test
##
## data:  Approve ~ Lag(total_ptas, shift = 2) + Lag(log(gdp)) + Lag(edu) +  ...
```

```
## chisq = 11.162, df = 6, p-value = 0.08349
## alternative hypothesis: one model is inconsistent
```

```r
#Build RE model on deep PTA count
re_mod3 <- plm(Approve ~
                 Lag((total_ptas), shift = 2) +
                 Lag(log(gdp)) +
                 Lag(edu) +
                 Lag(v2x_regime) +
                 year +
                 Lag((total_ptas), shift = 2)*developing,
                   data = deep_count_data2,
                   index = c("iso3n", "year"),
                   model = "random")
summary(re_mod3)
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Call:
## plm(formula = Approve ~ Lag((total_ptas), shift = 2) + Lag(log(gdp)) +
##     Lag(edu) + Lag(v2x_regime) + year + Lag((total_ptas), shift = 2) *
##     developing, data = deep_count_data2, model = "random", index = c("iso3n",
##     "year"))
##
## Unbalanced Panel: n = 138, T = 1-16, N = 1670
##
## Effects:
##                  var std.dev share
## idiosyncratic 122.16   11.05 0.397
## individual    185.80   13.63 0.603
## theta:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3702  0.7806  0.7951  0.7802  0.8013  0.8013
##
## Residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -38.390  -7.712  -0.332  -0.176   7.406  42.338
##
## Coefficients:
##                                          Estimate Std. Error z-value  Pr(>|z|)
## (Intercept)                             37.913207   8.275042  4.5816 4.614e-06
## Lag((total_ptas), shift = 2)             0.199364   0.046736  4.2657 1.992e-05
## Lag(log(gdp))                            2.315460   1.173569  1.9730 0.0484947
## Lag(edu)                                -0.966802   0.388992 -2.4854 0.0129405
## Lag(v2x_regime)                         -1.918614   0.713409 -2.6894 0.0071589
## year2007                                -6.720449   2.049188 -3.2796 0.0010397
## year2008                                -8.354086   1.973469 -4.2332 2.304e-05
## year2009                                -5.386991   2.013859 -2.6750 0.0074738
## year2010                                -2.943046   1.996011 -1.4745 0.1403567
## year2011                                -4.191081   1.938596 -2.1619 0.0306247
## year2012                                -6.802329   1.962044 -3.4670 0.0005264
## year2013                                -7.774464   1.956685 -3.9733 7.089e-05
## year2014                                -5.223449   1.953163 -2.6744 0.0074874
```

```
## year2015                               -5.851429   1.963170 -2.9806 0.0028768
## year2016                               -7.147218   1.968485 -3.6308 0.0002825
## year2017                               -4.902006   1.958175 -2.5034 0.0123022
## year2018                               -5.439082   1.981102 -2.7455 0.0060422
## year2019                               -5.276716   1.981043 -2.6636 0.0077308
## year2020                               -2.479643   2.051641 -1.2086 0.2268111
## year2021                               -5.306122   2.032751 -2.6103 0.0090459
## developing                              5.138630   1.600045  3.2116 0.0013202
## Lag((total_ptas), shift = 2):developing -0.038860   0.067142 -0.5788 0.5627457
##
## (Intercept)                             ***
## Lag((total_ptas), shift = 2)            ***
## Lag(log(gdp))                           *
## Lag(edu)                                *
## Lag(v2x_regime)                         **
## year2007                                **
## year2008                                ***
## year2009                                **
## year2010
## year2011                                *
## year2012                                ***
## year2013                                ***
## year2014                                **
## year2015                                **
## year2016                                ***
## year2017                                *
## year2018                                **
## year2019                                **
## year2020
## year2021                                **
## developing                              **
## Lag((total_ptas), shift = 2):developing
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    233400
## Residual Sum of Squares: 206640
## R-Squared:      0.11684
## Adj. R-Squared: 0.10559
## Chisq: 95.8182 on 21 DF, p-value: 1.5716e-11
```

```
phtest(fe_mod3, re_mod3)
```

```
##
##  Hausman Test
##
## data:  Approve ~ Lag((total_ptas), shift = 2) + Lag(log(gdp)) + Lag(edu) +  ...
## chisq = 12.035, df = 6, p-value = 0.06119
## alternative hypothesis: one model is inconsistent
```

Robustness Check 2: Deep Trade Agreement = index of 4+

```
#obtain a list of PTAs that have a depth > mean of depth
deep_ptas = subset(desta, desta$depth_index > 4)$base_treaty


#Subset desta dyads and withdrawals based on if the base_treaty is greater than mean in desta
desta_dyads2 = desta_dyads1 %>% filter(number %in% deep_ptas)

#Count all PTAs for each country-year (to get change)
pta_count = desta_dyads2 %>% dplyr::group_by(iso3n, year) %>% dplyr::summarise(count_pta = n())
```

```
## `summarise()` has grouped output by 'iso3n'. You can override using the
## `.groups` argument.
```

```
#Count all PTA withdrawals for each country-year
withdrawals2 = withdrawals %>% filter(number %in% deep_ptas)
withdrawals_count = withdrawals2 %>% dplyr::group_by(iso3n, year) %>% dplyr::summarise(count_withdrawal
```

```
## `summarise()` has grouped output by 'iso3n'. You can override using the
## `.groups` argument.
```

```
#Merge withdrawals column to desta_country_yr
deep_count_data = merge(pta_count, withdrawals_count, by = c("year", "iso3n"), all.x = TRUE, all.y = TRU

#Add missing years
deep_count_data <- setDT(deep_count_data)[CJ(iso3n=iso3n, year=seq(min(year), 2021), unique=TRUE),
              on=.(iso3n, year), roll=F]

#fill PTA count with 0
deep_count_data <- deep_count_data %>% mutate(count_pta = ifelse(is.na(count_pta), 0, count_pta),
                                             count_withdrawals = ifelse(is.na(count_withdrawals), 

#Add the PTAs
deep_count_data = deep_count_data %>% group_by(iso3n) %>%
    mutate(across(count_pta, ~ accumulate(., `+`)))
deep_count_data = deep_count_data %>% group_by(iso3n) %>%
    mutate(across(count_withdrawals, ~ accumulate(., `+`)))

#Create total PTA column
deep_count_data$total_ptas = deep_count_data$count_pta - deep_count_data$count_withdrawals

## Merge approval ratings w/data ##
deep_count_data = merge(deep_count_data, gallup, by = c("iso3n", "year"), all.x = TRUE)

## Merge HDI w/data ##
deep_count_data = merge(deep_count_data, hdi_hdi, by = c("iso3n", "year"), all.x = T) %>% mutate(develo
                                                hdi_score>= 0.8 ~ 0))
# Assign NAs in developing to a 1 or 0 based on current values
deep_count_data$developing2 = nafill(deep_count_data$developing, type = "nocb")
deep_count_data$developing3 = ifelse(deep_count_data$developing2 == 0, "Developed", "Developing")
```

```r
## Merge Control variables w/data ##
deep_count_data = merge(deep_count_data, hdi_gdp, by = c("iso3n", "year"), all.x = T)
deep_count_data = merge(deep_count_data, hdi_edu, by = c("iso3n", "year"), all.x = T)
deep_count_data = merge(deep_count_data, vdem[,c("iso3n", "v2x_regime", "year")], by = c("iso3n", "year

#Remove duplicate country-yr
deep_count_data$unique_id <- paste(deep_count_data$iso3n,deep_count_data$year) # concatenate to make un
deep_count_data$duplicate = duplicated(deep_count_data$unique_id) # generate the duplicate variable
deep_count_data = deep_count_data[deep_count_data$duplicate != "TRUE", ]

#Subset to 2005-2021 and rows without missing data
deep_count_data3 = deep_count_data[deep_count_data$year > 2004,]
deep_count_data3 = deep_count_data3[!is.na(deep_count_data3$Approve),]

#Build FE model on PTA count
fe_mod4 <- plm(Approve ~
                 Lag((total_ptas), shift = 2) +
                 Lag(log(gdp)) +
                 Lag(edu) +
                 Lag(v2x_regime) +
                 year,
                   data = deep_count_data3,
                   index = c("iso3n", "year"),
                   model = "within",
                   effect = "twoways")
summary(fe_mod4)
```

```
## Twoways effects Within Model
##
## Call:
## plm(formula = Approve ~ Lag((total_ptas), shift = 2) + Lag(log(gdp)) +
##     Lag(edu) + Lag(v2x_regime) + year, data = deep_count_data3,
##     effect = "twoways", model = "within", index = c("iso3n",
##         "year"))
##
## Unbalanced Panel: n = 89, T = 1-16, N = 1146
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -46.77562  -6.66829   0.30063   6.24184  43.25041
##
## Coefficients:
##                                Estimate Std. Error t-value Pr(>|t|)
## Lag((total_ptas), shift = 2)  0.174743   0.061231  2.8538 0.004405 **
## Lag(log(gdp))                 2.840682   1.675690  1.6952 0.090331 .
## Lag(edu)                     -0.891252   0.543202 -1.6407 0.101155
## Lag(v2x_regime)              -0.974526   1.002871 -0.9717 0.331408
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    128780
## Residual Sum of Squares: 126820
```

```
## R-Squared:      0.015282
## Adj. R-Squared: -0.086226
## F-statistic: 4.02711 on 4 and 1038 DF, p-value: 0.003019
```

"'