Democratic Erosion

Anthony Sutton

4/20/2022

Set arbitrary thresholds

```
dem_threshold = 0.5 #on vdem's 1-point scales
survival_threshold = 0.5 #as a probability
erosion_threshold = 1 #as multiple of standard deviations
treatment_threshold = 0.9 #percentile that distinguishes high clientelism/disinfo/polarization from low
                          #must hold high threshold else few democratic spells experience onset of "treatment"
                          # because they start the spell already above the threshold
lag_range = 10 #test time lags of dependent variables from 1 to this many years

#color scheme used in charts
dem_color = 'dodgerblue'
interlock_color = 'midnightblue'
client_color = 'sienna'
media_color = 'forestgreen'
disinfo_color = 'olivedrab3'
polar_color = 'tomato1'
```

Load libraries

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.4
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.0.4
```

```
## Warning: package 'tibble' was built under R version 4.0.4
```

```
## Warning: package 'tidyr' was built under R version 4.0.4
```

```
## Warning: package 'readr' was built under R version 4.0.4
```

```
## Warning: package 'purrr' was built under R version 4.0.4
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
## Warning: package 'stringr' was built under R version 4.0.4
```

```
## Warning: package 'forcats' was built under R version 4.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(scales) #for percent scales on charts
```

```
## Warning: package 'scales' was built under R version 4.0.4
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(knitr)  #for kable function to format regression results
```

```
## Warning: package 'knitr' was built under R version 4.0.4
```

```
library(stargazer)  #for displaying multiple model results
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
library(MatchIt) #for matching analysis
```

```
## Warning: package 'MatchIt' was built under R version 4.0.5
```

Load and shape data

This project uses V-Dem data:

Coppedge, Michael, John Gerring, Carl Henrik Knutsen, Staffan I. Lindberg, Jan Teorell, Nazifa Alizada, David Altman, et al. “V-Dem Dataset V11.1.” Varieties of Democracy Project, 2021. https://doi.org/10.23696/vdemds21 (https://doi.org/10.23696/vdemds21).

Starting from the raw data file provided by V-Dem, this chunk trims the dataframe to a manageable size. It need only be run once. Thereafter, the following code will directly load the trimmed file.)

```
##regime type data from V-Dem - one row per country year
#vdem <- read_csv('./data/V-Dem-CY-Full+Others-v11.1.csv'

##trim to a manageable file size by selecting only relevant variables
#vdem <- vdem %>% select(country_name, country_text_id, year, COWcode,
#                        v2x_polyarchy, v2x_libdem, v2x_partipdem, v2x_delibdem, v2x_egaldem,
#                        v2xnp_client, v2elvotbuy, v2dlencmps, v2psprlnks, v2x_elecreg,
#                        v2smonex, v2smmefra, v2smgovdom, v2smpardom, v2smfordom,
#                        v2cacamps, v2smpolsoc, v2smpolhate, v2psplats,
#                        e_migdpgro, e_migdppc,
#                        e_mipopula, e_miurbani,
#                        e_total_fuel_income_pc, e_total_resources_income_pc)

#write_csv(vdem, file = './data/vdem_trimmed.csv')
```

```
vdem <- read_csv('./data/vdem_trimmed.csv')
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   country_name = col_character(),
##   country_text_id = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```
vdem <- vdem %>% filter(!is.na(v2x_polyarchy))
```

Rescale variables to zero-to-one, and ensure 1 is the direction of more of that variable (not always more democratic)

```
vdem$v2cacamps <- rescale(vdem$v2cacamps, to = c(0,1))
vdem$v2smpolsoc <- 1 - rescale(vdem$v2smpolsoc)
vdem$v2smpolhate <- 1 - rescale(vdem$v2smpolhate)
vdem$v2smonex <- rescale(vdem$v2smonex, to = c(0,1)) #already runs low to high
vdem$v2smmefra <- 1 - rescale(vdem$v2smmefra, to = c(0,1))
vdem$v2smgovdom <- 1 - rescale(vdem$v2smgovdom, to = c(0,1))
vdem$v2smpardom <- 1 - rescale(vdem$v2smpardom, to = c(0,1))
vdem$v2smfordom <- 1 - rescale(vdem$v2smfordom, to = c(0,1))
vdem$v2psprlnks <- 1 - rescale(vdem$v2psprlnks, to = c(0,1))
vdem$v2dlencmps <- 1 - rescale(vdem$v2dlencmps, to = c(0,1))
vdem$v2psplats <- 1 - rescale(vdem$v2psplats, to = c(0,1))
```

Count GDP per capita in thousands of dollars, changing from single dollars

```
vdem$e_migdppc <- vdem$e_migdppc / 1000
```

Recalculate natural resources income as fraction of gdp

```
vdem$e_total_resources_percent <-
  case_when(is.na(vdem$e_migdppc) ~ as.numeric(NA),
            vdem$e_migdppc <= 0 ~ as.numeric(NA),
            TRUE ~ vdem$e_total_resources_income_pc / vdem$e_migdppc )
```

Multiply variables to interact in linear regressions (needed to run linear models within a loop)

```
vdem$clientXresources <- vdem$v2xnp_client * vdem$e_total_resources_percent #clientelism * natural resources
vdem$smmefraXsmpardom <- vdem$v2smmefra * vdem$v2smpardom #online fractionalization * party disinfo
vdem$smonexXsmmefra <- vdem$v2smonex * vdem$v2smmefra #online consumption * fractionalization
vdem$smonexXsmfordom <- vdem$v2smonex * vdem$v2smfordom #online consuption * foreign disinfo
vdem$smmefraXsmfordom <- vdem$v2smmefra * vdem$v2smfordom #online fractionalization * foreign disinfo
vdem$clientXcacamps <- vdem$v2cacamps * vdem$v2xnp_client #clientelism * political polarization
vdem$clientXsmpolsoc <- vdem$v2smpolsoc * vdem$v2xnp_client #clientelism * social polarization
vdem$smmefraXsmpolsoc <- vdem$v2smmefra * vdem$v2smpolsoc #online fractionalization * social polarization
```

Examine the data

```
skimr::skim(vdem)
```

Data summary

Name	vdem
Number of rows	25545
Number of columns	38
_______________________	
Column type frequency:	
character	2
numeric	36
________________________	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
country_name	0	1	4	32	0	199	0
country_text_id	0	1	3	3	0	199	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
year	0	1.00	1932.52	61.53	1789.00	1894.00	1943.00	1984.00	2020.00	▁▂▃▇▇
COWcode	510	0.98	443.29	250.82	2.00	225.00	435.00	652.00	950.00	▆▇▇▇▅
v2x_polyarchy	0	1.00	0.26	0.26	0.01	0.06	0.17	0.37	0.92	▇▃▁▁▂
v2x_libdem	992	0.96	0.22	0.23	0.00	0.06	0.12	0.29	0.89	▇▂▁▁▁
v2x_partipdem	419	0.98	0.16	0.18	0.00	0.03	0.08	0.22	0.80	▇▂▁▁▁
v2x_delibdem	6799	0.73	0.23	0.25	0.00	0.04	0.12	0.37	0.89	▇▂▁▁▁
v2x_egaldem	6799	0.73	0.24	0.23	0.01	0.06	0.14	0.33	0.88	▇▂▁▁▁
v2xnp_client	1413	0.94	0.55	0.27	0.02	0.34	0.58	0.79	0.99	▅▆▇▇▇
v2elvotbuy	20773	0.19	-0.02	1.39	-3.07	-1.12	-0.20	1.17	3.20	▂▇▆▇▂
v2dlencmps	97	1.00	0.51	0.20	0.00	0.35	0.48	0.67	1.00	▁▇▇▂
v2psprlnks	1475	0.94	0.52	0.21	0.00	0.37	0.55	0.69	1.00	▂▅▇▇▂
v2x_elecreg	0	1.00	0.59	0.49	0.00	0.00	1.00	1.00	1.00	▆▁▁▁▇
v2smonex	21804	0.15	0.52	0.19	0.00	0.39	0.53	0.66	1.00	▁▅▇▅▂
v2smmefra	21804	0.15	0.49	0.17	0.00	0.39	0.51	0.61	1.00	▁▃▇▅▁

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
v2smgovdom	21804	0.15	0.43	0.22	0.00	0.27	0.43	0.59	1.00	▃▇▇▅▁
v2smpardom	21804	0.15	0.42	0.19	0.00	0.28	0.41	0.57	1.00	▂▇▇▃▁
v2smfordom	21804	0.15	0.37	0.16	0.00	0.26	0.36	0.49	1.00	▂▇▆▁▁
v2cacamps	7643	0.70	0.45	0.17	0.00	0.32	0.46	0.58	1.00	▂▇▇▃▁
v2smpolsoc	21804	0.15	0.57	0.18	0.00	0.43	0.58	0.70	1.00	▁▅▇▇▂
v2smpolhate	21804	0.15	0.44	0.19	0.00	0.30	0.44	0.59	1.00	▂▇▇▅▁
v2psplats	1063	0.96	0.55	0.26	0.00	0.34	0.55	0.80	1.00	▃▆▇▆▇
e_migdpgro	11116	0.56	0.02	0.08	-1.00	0.00	0.02	0.05	2.72	▁▇▁▁▁
e_migdppc	10962	0.57	7.95	11.21	0.00	1.74	3.71	8.86	156.30	▇▁▁▁▁
e_mipopula	6874	0.73	19722.19	75293.76	17.90	1133.36	3624.00	9983.49	1262645.00	▇▁▁▁▁
e_miurbani	8294	0.68	0.26	0.22	0.00	0.12	0.22	0.34	2.59	▇▁▁▁▁
e_total_fuel_income_pc	14972	0.41	541.95	3230.78	0.00	0.00	3.44	81.25	81161.85	▇▁▁▁▁
e_total_resources_income_pc	15015	0.41	594.87	3236.42	0.00	0.23	28.04	188.91	81161.85	▇▁▁▁▁
e_total_resources_percent	16474	0.36	51.54	143.39	0.00	0.80	8.38	37.12	2596.83	▇▁▁▁▁
clientXresources	16476	0.36	29.48	97.08	0.00	0.16	2.99	16.04	1882.70	▇▁▁▁▁
smmefraXsmpardom	21804	0.15	0.21	0.13	0.00	0.11	0.20	0.30	0.78	▇▇▃▁▁
smonexXsmmefra	21804	0.15	0.26	0.12	0.00	0.17	0.25	0.33	0.74	▃▇▅▁▁
smonexXsmfordom	21804	0.15	0.20	0.12	0.00	0.11	0.17	0.27	1.00	▇▅▁▁▁
smmefraXsmfordom	21804	0.15	0.19	0.11	0.00	0.11	0.18	0.25	0.74	▇▇▃▁▁
clientXcacamps	7690	0.70	0.25	0.17	0.00	0.12	0.23	0.38	0.88	▇▇▅▁▁
clientXsmpolsoc	21804	0.15	0.28	0.19	0.00	0.11	0.27	0.41	0.84	▇▇▅▃▁
smmefraXsmpolsoc	21804	0.15	0.30	0.16	0.00	0.17	0.28	0.40	0.88	▅▇▅▂▁

Situate country-years within democratic spells

Label democratization and autocratization, which define spells

First, create empty variables

```
vdem$democratize <- as.logical(NA)
vdem$autocratize <- as.logical(NA)
vdem$first_appear <- as.logical(FALSE)
```

Second, loop over every country-year

```
for (i in seq_along(vdem$year)){
  #if no entry for the country in the prior year, label first year and skip to next country-year
  if (length(vdem$v2x_polyarchy[vdem$country_name == vdem$country_name[i] & vdem$year == (vdem$year[i] - 1)]) == 0) vdem$fir
st_appear[i] <- TRUE
  if (vdem$first_appear[i] == TRUE) next

  #otherwise, identify the polyarchy score for the same country in the preceding year
  pre1_polyarchy <- vdem %>%
    filter(country_name == vdem$country_name[i] & year == (vdem$year[i] - 1)) %>%
    pull(v2x_polyarchy)

  #call it democratization if a country is a democracy this year and was not in prior year
  vdem$democratize[i] <-
    case_when(pre1_polyarchy >= dem_threshold ~ FALSE,
              vdem$v2x_polyarchy[i] >= dem_threshold ~ TRUE,
              TRUE ~ FALSE
              )

  #call it autocratization if a country is an autocracy this year and was not in prior year
  vdem$autocratize[i] <-
    case_when(pre1_polyarchy < dem_threshold ~ FALSE,
              vdem$v2x_polyarchy[i] < dem_threshold ~ TRUE,
              TRUE ~ FALSE
    )
}
```

Third, for countries first appearing in data, determine whether democratized and view results

```
vdem$democratize <- if_else(vdem$first_appear == TRUE,
                            if_else(vdem$v2x_polyarchy >= dem_threshold,
                                    TRUE,
                                    FALSE),
                            vdem$democratize)
summary(vdem$democratize)
```

```
##    Mode   FALSE    TRUE
## logical   25357     188
```

Fourth, for countries first appearing in data, determine whether autocratized and view results

```
vdem$autocratize <- if_else(vdem$first_appear == TRUE,
                            if_else(vdem$v2x_polyarchy < dem_threshold,
                                    TRUE,
                                    FALSE),
                            vdem$autocratize)
summary(vdem$autocratize)
```

```
##    Mode   FALSE    TRUE
## logical   25201     344
```

Identify beginning of each democratic spell

```
vdem$dem_spell_start <- as.numeric(NA)
 #loop over every country-year
for (i in seq_along(vdem$year)){
 #ignore autocracies
 if (vdem$v2x_polyarchy[i] < dem_threshold) next
 #recognize democratization years as their own spell starts
 if (vdem$democratize[i] == TRUE) vdem$dem_spell_start[i] <- vdem$year[i]
 if (vdem$democratize[i] == TRUE) next

 vdem$dem_spell_start[i] <- vdem %>%
   filter(country_name == vdem$country_name[i] &
            vdem$year < vdem$year[i] &
            vdem$democratize == TRUE) %>%
   summarize(dem_spell_start = max(year)) %>%
   pull(dem_spell_start)
}

summary(vdem$dem_spell_start)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1849    1921    1952    1951    1990    2020   20528
```

Name democratic spells

```
vdem$dem_spell_name <- if_else(is.na(vdem$dem_spell_start),
                               as.character(NA),
                               paste(vdem$country_name, as.character(vdem$dem_spell_start)))
```

Check that every democratic country year is labeled with a spell start

```
sum(!is.na(vdem$dem_spell_start)) == sum(vdem$v2x_polyarchy >= dem_threshold)
```

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

Identify prior peak polyarchy within same democratic spell

```
vdem$dem_spell_peak = as.numeric(NA)
 ###loop over every country year
for (i in seq_along(vdem$year)){
  ###ignore autocracies
  if(is.na(vdem$dem_spell_start[i])) next
  ###examine polyarchy scores from each year during democratic spell, record highest
  vdem$dem_spell_peak[i] <- vdem %>%
    filter(country_name == vdem$country_name[i] &
             year >= vdem$dem_spell_start[i] &
             year <= vdem$year[i]) %>%
    summarize(peak_polyarchy = max(v2x_polyarchy), na.rm = TRUE) %>%
    pull(peak_polyarchy)
}
summary(vdem$dem_spell_peak)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.500   0.637   0.763   0.742   0.852   0.919   20528
```

Record running tally of length of democracy spell

```
vdem$dem_spell_running <- vdem$year - vdem$dem_spell_start
```

Identify eventual outcome of each democracy spell

```
vdem$dem_spell_outcome <- as.character(NA)

for (i in seq_along(vdem$year)){
vdem$dem_spell_outcome[i] <-
  if_else(vdem$v2x_polyarchy[i] >= dem_threshold,
          if_else(sum(vdem$autocratize[vdem$country_name == vdem$country_name[i] & vdem$year >= vdem$year[i]]) > 0,
                  'autocracy',
                  'democracy'),
          as.character(NA))
}
vdem$dem_spell_outcome <- as.factor(vdem$dem_spell_outcome)

summary(vdem$dem_spell_outcome)
```

```
## autocracy democracy      NA's
##      1132      3885     20528
```

Note length of spells at autocratization or last year of data

```
vdem <- vdem %>%
  group_by(dem_spell_name) %>%
  mutate(dem_spell_length = max(dem_spell_running)) %>%
  ungroup()
summary(vdem$dem_spell_length)
```

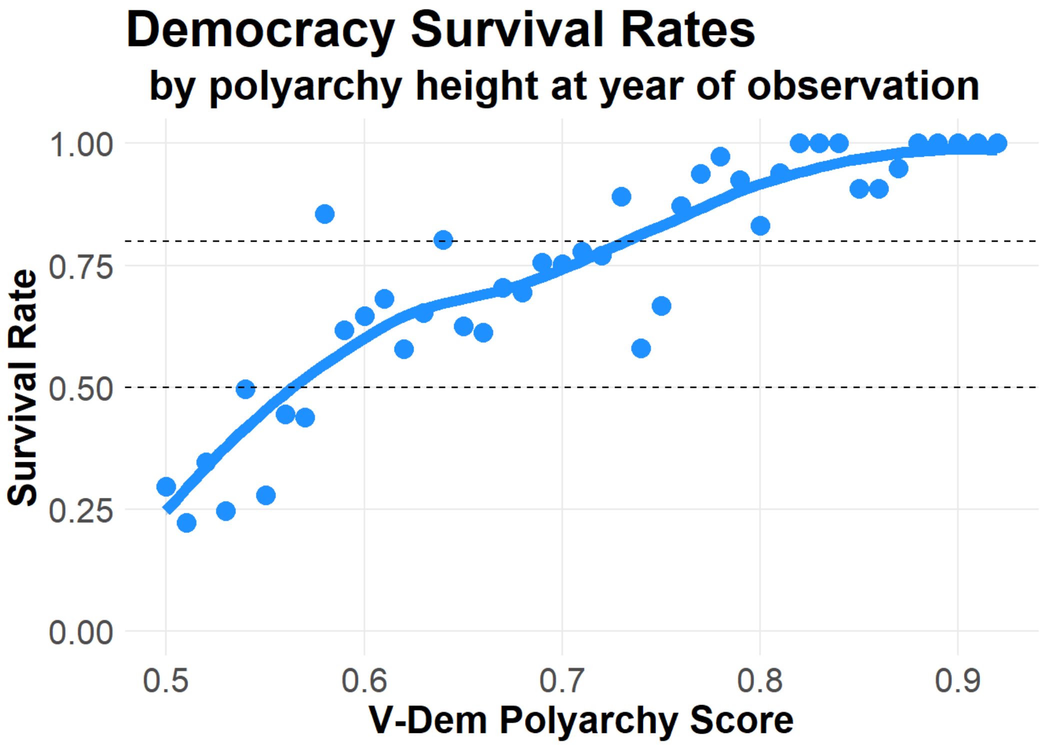
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   27.00   45.00   60.49   75.00  171.00   20528
```

Identify consolidated democracies

Count democracies as consolidated if their experience with democracy is sufficiently long, high, and broad.

Chart outcomes by polyarchy height

```
## `geom_smooth()` using formula 'y ~ x'
```



```
## `geom_smooth()` using formula 'y ~ x'
```

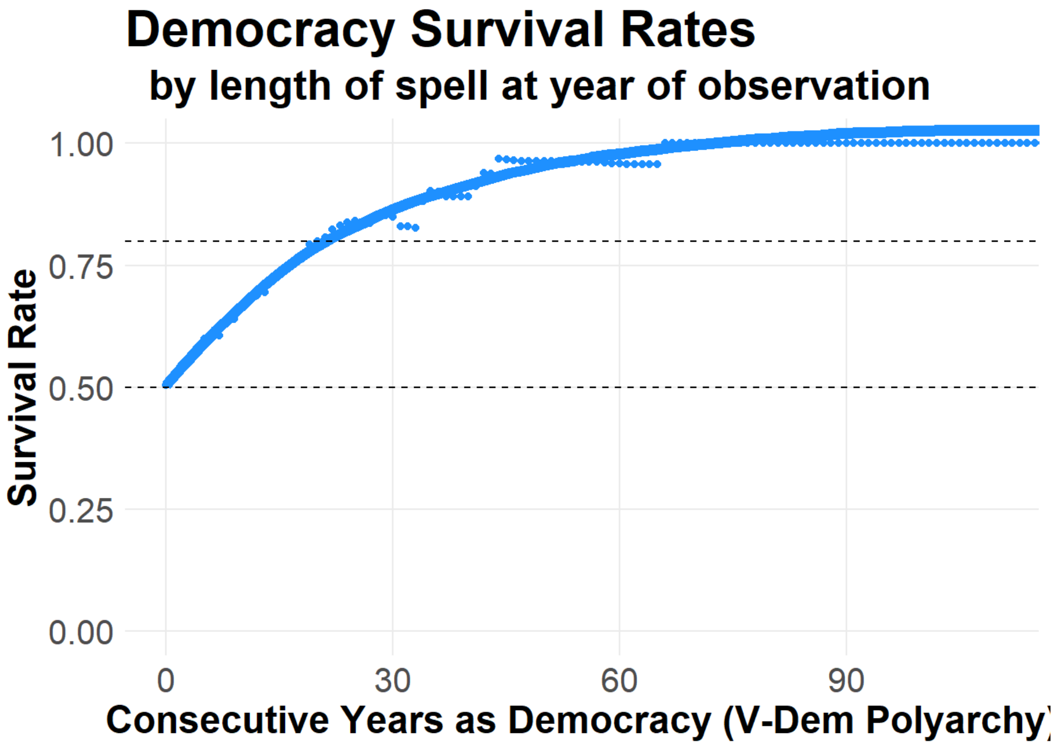
Calculate minimum polyarchy to predict survival above threshold

```
height_threshold <- vdem %>%
  filter(v2x_polyarchy >= dem_threshold) %>%
  group_by(polyarchy_cohort = round(v2x_polyarchy, digits = 2)) %>%
  summarize(outcome_rate = sum(dem_spell_outcome == 'democracy') / n()) %>%
  filter(outcome_rate < survival_threshold) %>%
  summarize(last_cohort = max(polyarchy_cohort)) %>%
  pull(last_cohort) + 0.01
height_threshold
```

```
## [1] 0.58
```

Chart outcome by democracy spell length

```
## `geom_smooth()` using formula 'y ~ x'
```



```
## `geom_smooth()` using formula 'y ~ x'
```

Calculate fraction of all democracy spells that lasted

```
vdem %>% filter(dem_spell_running == 0) %>%
  summarize(count = n(),
            stayed_democracies = sum(dem_spell_outcome == 'democracy'),
            survival_rate = stayed_democracies/count)
```

```
## # A tibble: 1 x 3
##   count stayed_democracies survival_rate
##   <int>              <int>         <dbl>
## 1   188                 95         0.505
```

Calculate minimum spell length to predict survival above threshold

```
length_threshold <- vdem %>%
  filter(v2x_polyarchy >= dem_threshold) %>%
  group_by(dem_spell_running) %>%
  summarize(outcome_rate = sum(dem_spell_outcome == 'democracy') / n()) %>%
  filter(outcome_rate >= survival_threshold) %>%
  summarize(last_cohort = min(dem_spell_running)) %>%
  pull(last_cohort)
length_threshold
```

```
## [1] 0
```

Compare vdem high level indexes Observe that the four other varieties of democracy tend to score lower than polyarchy

```
vdem %>% filter(v2x_polyarchy >= dem_threshold) %>%
  summarize(polyarchy = mean(v2x_polyarchy),
            liberal = mean(v2x_libdem, na.rm = TRUE),
            participatory = mean(v2x_partipdem, na.rm = TRUE),
            deliberative = mean(v2x_delibdem, na.rm = TRUE),
            egalitarian = mean(v2x_egaldem, na.rm = TRUE))
```

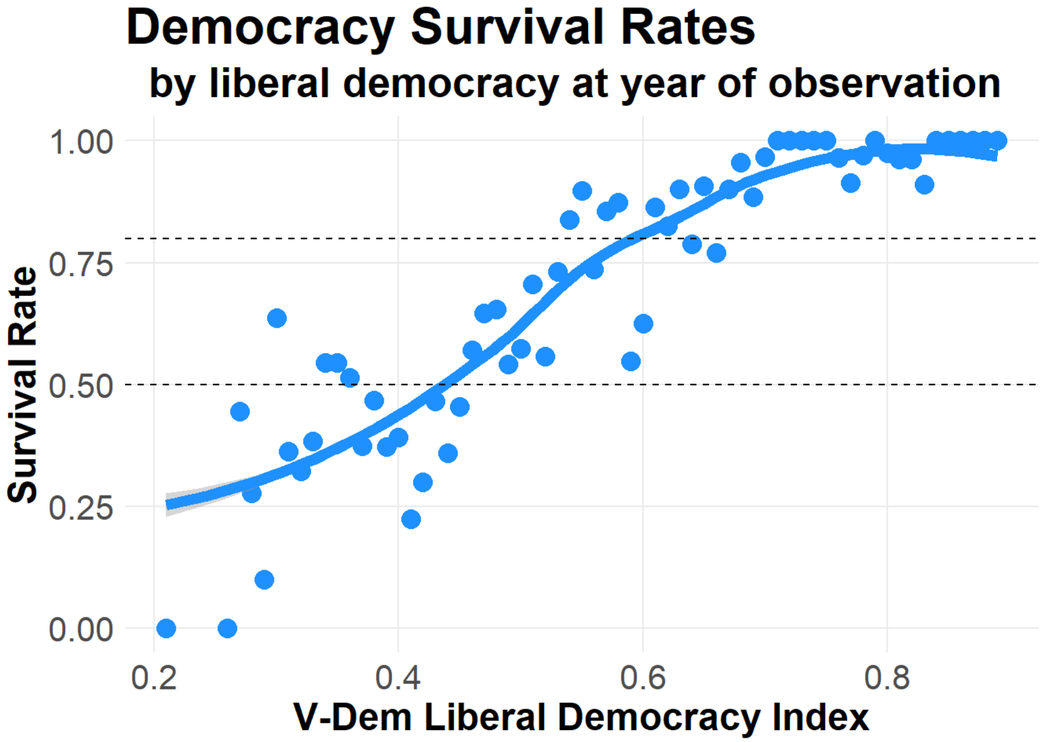
```
## # A tibble: 1 x 5
##   polyarchy liberal participatory deliberative egalitarian
##       <dbl>   <dbl>         <dbl>        <dbl>       <dbl>
## 1     0.730   0.623         0.489        0.608       0.579
```

```
vdem %>% filter(v2x_polyarchy >= dem_threshold) %>%
  group_by(libdem_cohort = round(v2x_libdem, digits = 2)) %>%
  mutate(outcome_rate = sum(dem_spell_outcome == 'democracy') / n()) %>%
  ggplot(aes(x = libdem_cohort, y = outcome_rate))+
  geom_point(size = 4, color = dem_color)+
  geom_smooth(method = 'loess', size = 2.5, color = dem_color)+
  coord_cartesian(ylim = c(0,1))+
  geom_hline(yintercept = 0.5, linetype = 'dashed')+
  geom_hline(yintercept = 0.8, linetype = 'dashed')+
  theme_minimal()+
  labs(title = 'Democracy Survival Rates',
       subtitle = '  by liberal democracy at year of observation',
       y = 'Survival Rate',
       x = 'V-Dem Liberal Democracy Index')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16))
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



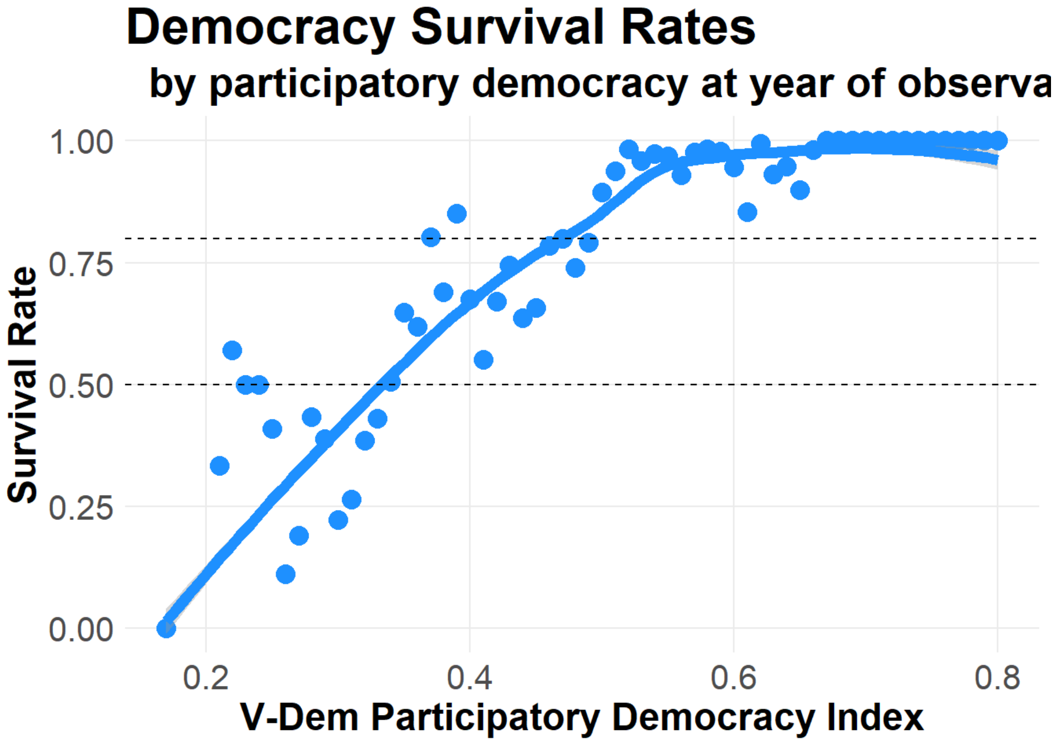
```
ggsave(filename = "./visuals/democracy_survival_libdem.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).

## Warning: Removed 1 rows containing missing values (geom_point).
```

```
vdem %>% filter(v2x_polyarchy >= dem_threshold) %>%
  group_by(partipdem_cohort = round(v2x_partipdem, digits = 2)) %>%
  mutate(outcome_rate = sum(dem_spell_outcome == 'democracy') / n()) %>%
  ggplot(aes(x = partipdem_cohort, y = outcome_rate))+
  geom_point(size = 4, color = dem_color)+
  geom_smooth(method = 'loess', size = 2.5, color = dem_color)+
  coord_cartesian(ylim = c(0,1))+
  geom_hline(yintercept = 0.5, linetype = 'dashed')+
  geom_hline(yintercept = 0.8, linetype = 'dashed')+
  theme_minimal()+
  labs(title = 'Democracy Survival Rates',
       subtitle = '  by participatory democracy at year of observation',
       y = 'Survival Rate',
       x = 'V-Dem Participatory Democracy Index')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
ggsave(filename = "./visuals/democracy_survival_partipdem.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

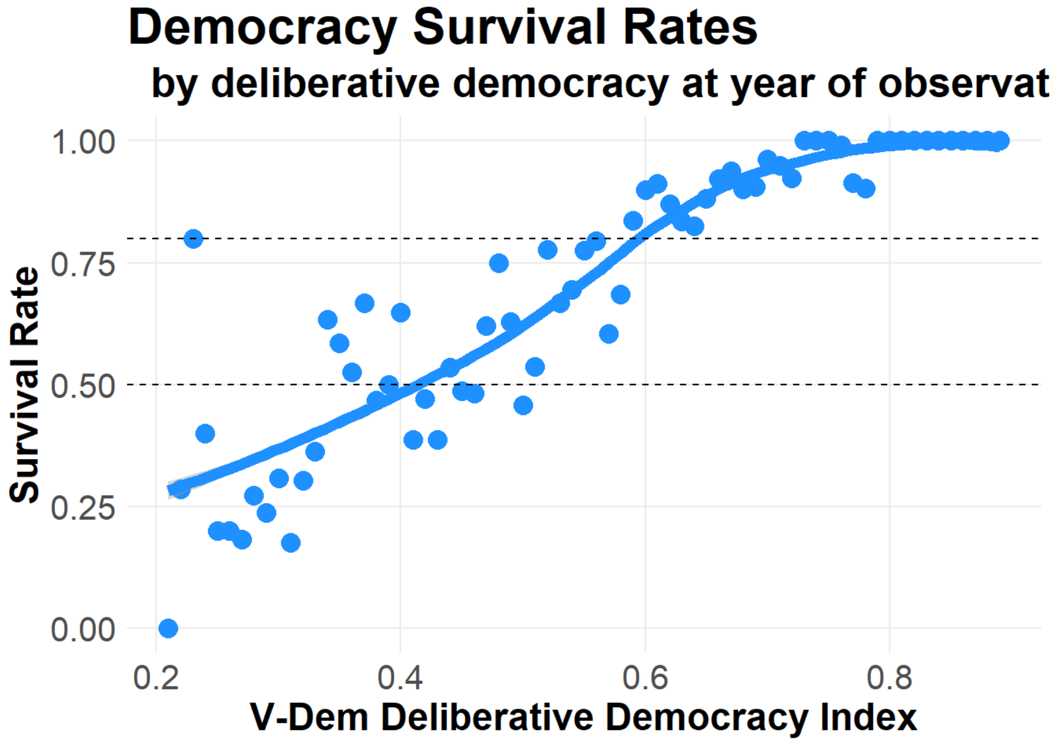
```
## `geom_smooth()` using formula 'y ~ x'
```

```
vdem %>% filter(v2x_polyarchy >= dem_threshold) %>%
  group_by(delibdem_cohort = round(v2x_delibdem, digits = 2)) %>%
  mutate(outcome_rate = sum(dem_spell_outcome == 'democracy') / n()) %>%
  ggplot(aes(x = delibdem_cohort, y = outcome_rate))+
  geom_point(size = 4, color = dem_color)+
  geom_smooth(method = 'loess', size = 2.5, color = dem_color)+
  coord_cartesian(ylim = c(0,1))+
  geom_hline(yintercept = 0.5, linetype = 'dashed')+
  geom_hline(yintercept = 0.8, linetype = 'dashed')+
  theme_minimal()+
  labs(title = 'Democracy Survival Rates',
       subtitle = '  by deliberative democracy at year of observation',
       y = 'Survival Rate',
       x = 'V-Dem Deliberative Democracy Index')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16))
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 164 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 164 rows containing missing values (geom_point).
```



```
ggsave(filename = "./visuals/democracy_survival_delibdem.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

```
## `geom_smooth()` using formula 'y ~ x'
```

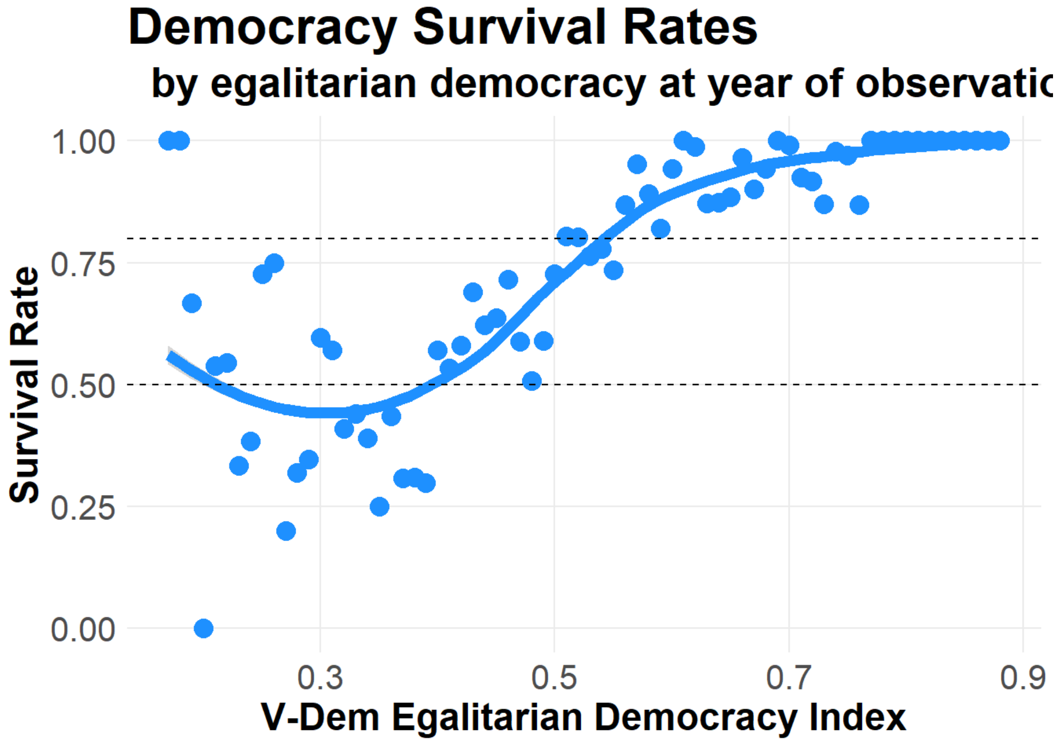
```
## Warning: Removed 164 rows containing non-finite values (stat_smooth).

## Warning: Removed 164 rows containing missing values (geom_point).
```

```
vdem %>% filter(v2x_polyarchy >= dem_threshold) %>%
  group_by(egaldem_cohort = round(v2x_egaldem, digits = 2)) %>%
  mutate(outcome_rate = sum(dem_spell_outcome == 'democracy') / n()) %>%
  ggplot(aes(x = egaldem_cohort, y = outcome_rate))+
  geom_point(size = 4, color = dem_color)+
  geom_smooth(method = 'loess', size = 2.5, color = dem_color)+
  coord_cartesian(ylim = c(0,1))+
  geom_hline(yintercept = 0.5, linetype = 'dashed')+
  geom_hline(yintercept = 0.8, linetype = 'dashed')+
  theme_minimal()+
  labs(title = 'Democracy Survival Rates',
       subtitle = '  by egalitarian democracy at year of observation',
       y = 'Survival Rate',
       x = 'V-Dem Egalitarian Democracy Index')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16))
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 164 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 164 rows containing missing values (geom_point).
```



```
ggsave(filename = "./visuals/democracy_survival_egaldem.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 164 rows containing non-finite values (stat_smooth).

## Warning: Removed 164 rows containing missing values (geom_point).
```

Calculate survival rate for democracies scoring well on all indexes

```
vdem %>% filter(v2x_polyarchy >= dem_threshold) %>%
  group_by(v2x_libdem >= dem_threshold &
             v2x_partipdem >= dem_threshold &
             v2x_delibdem >= dem_threshold &
             v2x_egaldem >= dem_threshold) %>%
  summarize(count = n(),
            stayed_democracies = sum(dem_spell_outcome == 'democracy'),
            success_rate = stayed_democracies/count)
```

```
## # A tibble: 3 x 4
##   `&...` count stayed_democracies success_rate
##   <lgl>  <int>              <int>        <dbl>
## 1 FALSE   2643               1601        0.606
## 2 TRUE    2368               2278        0.962
## 3 NA         6                  6        1
```

List all democratic spells that ever exceeded all thresholds in same year

```
broad_dem_spell_list <- vdem %>%
  filter(v2x_polyarchy >= dem_threshold,
         v2x_libdem >= dem_threshold &
           v2x_partipdem >= dem_threshold &
           v2x_delibdem >= dem_threshold &
           v2x_egaldem >= dem_threshold) %>%
  distinct(dem_spell_name) %>%
  pull(dem_spell_name)
```

Calculate survival rate predicted by whether broad democracy

```
vdem %>% filter(v2x_polyarchy >= dem_threshold) %>%
  distinct(dem_spell_name, .keep_all = TRUE)  %>%
  group_by(broad_dem = dem_spell_name %in% broad_dem_spell_list) %>%
  summarize(stayed_democratic = sum(dem_spell_outcome == 'democracy'),
            count = n(),
            survival_rate = stayed_democratic / count)
```

```
## # A tibble: 2 x 4
##   broad_dem stayed_democratic count survival_rate
##   <lgl>                 <int> <int>         <dbl>
## 1 FALSE                    45   132         0.341
## 2 TRUE                     50    56         0.893
```

Repeat process using individual thresholds predicting 80% survival Results are not very different - similar split of cases

```
high_broad_dem_spell_list <- vdem %>%
  filter(v2x_polyarchy >= 0.76,
         v2x_libdem >= 0.61 &
           v2x_partipdem >= 0.50 &
           v2x_delibdem >= 0.59 &
           v2x_egaldem >= 0.56) %>%
  distinct(dem_spell_name) %>%
  pull(dem_spell_name)

vdem %>% filter(v2x_polyarchy >= dem_threshold) %>%
  distinct(dem_spell_name, .keep_all = TRUE)  %>%
  group_by(high_broad_dem = dem_spell_name %in% high_broad_dem_spell_list) %>%
  summarize(stayed_democratic = sum(dem_spell_outcome == 'democracy'),
            count = n(),
            survival_rate = stayed_democratic / count)
```

```
## # A tibble: 2 x 4
##   high_broad_dem stayed_democratic count survival_rate
##   <lgl>                      <int> <int>         <dbl>
## 1 FALSE                         49   139         0.353
## 2 TRUE                          46    49         0.939
```

Label each country-year as consolidated or not …operationalizing consolidation by democracy spell length

```
vdem$consolidated_long <- case_when(vdem$v2x_polyarchy < dem_threshold ~ as.logical(NA),
                                    vdem$dem_spell_running >= length_threshold ~ TRUE,
                                    TRUE ~ FALSE)
summary(vdem$consolidated_long)
```

```
##    Mode    TRUE    NA's
## logical    5017   20528
```

…operationalizing consolidation by polyarchy height

```
vdem$consolidated_high <- case_when(vdem$v2x_polyarchy < dem_threshold ~ as.logical(NA),
                                    vdem$v2x_polyarchy >= height_threshold ~ TRUE,
                                    TRUE ~ FALSE)
summary(vdem$consolidated_high)
```

```
##    Mode   FALSE    TRUE    NA's 
## logical     890    4127   20528
```

…operationalizing consolidation by varietal breadth

```
vdem$consolidated_broad <- case_when(vdem$v2x_polyarchy < dem_threshold ~ as.logical(NA),
                                     (vdem$v2x_libdem >= dem_threshold &
                                        vdem$v2x_partipdem >= dem_threshold &
                                        vdem$v2x_delibdem >= dem_threshold &
                                        vdem$v2x_egaldem >= dem_threshold) ~ TRUE,
                                     TRUE ~ FALSE)
summary(vdem$consolidated_broad)
```

```
##    Mode   FALSE    TRUE    NA's 
## logical    2649    2368   20528
```

Examine overlap among consolidation measures

```
vdem %>% select(consolidated_long, consolidated_high, consolidated_broad) %>%
  na.omit() %>%
  cor()
```

```
## Warning in cor(.): the standard deviation is zero
```

```
##                    consolidated_long consolidated_high consolidated_broad
## consolidated_long                  1                NA                 NA
## consolidated_high                 NA         1.0000000          0.4390638
## consolidated_broad                NA         0.4390638          1.0000000
```

Create a variable for combined consolidation

```
vdem$consolidated_lhb <- case_when(vdem$v2x_polyarchy < dem_threshold ~ as.logical(NA),
                                   (vdem$consolidated_long == TRUE &
                                      vdem$consolidated_high == TRUE &
                                      vdem$consolidated_broad == TRUE) ~ TRUE,
                                   TRUE ~ FALSE)
summary(vdem$consolidated_lhb)
```

```
##    Mode   FALSE    TRUE    NA's 
## logical    2649    2368   20528
```

Count democratic spells that ever consolidated

```
vdem %>% filter(consolidated_lhb == TRUE) %>%
  distinct(dem_spell_name) %>%
  summarize(count = n())
```

```
## # A tibble: 1 x 1
##   count
##   <int>
## 1    56
```

Label erosion as any democratic country-year: *with polyarchy score at least X times 1 standard deviations below spell peak, where X is an arbitrarily defined multiple,* and so long as the spell ever peaked high enough to erode without autocratizing

```
vdem$erode <- case_when(vdem$v2x_polyarchy < dem_threshold ~ as.logical(NA),
                        vdem$dem_spell_peak < (dem_threshold + (erosion_threshold * sd(vdem$v2x_polyarchy[vdem$v2x_polyarchy
>= dem_threshold]))) ~ as.logical(NA),
                        vdem$dem_spell_peak - vdem$v2x_polyarchy >= (erosion_threshold * sd(vdem$v2x_polyarchy[vdem$v2x_poly
archy >= dem_threshold])) ~ TRUE,
                        TRUE ~ FALSE)
summary(vdem$erode)
```

```
##    Mode   FALSE    TRUE    NA's 
## logical    3762      74   21709
```

Count outcomes

```
vdem %>% filter(erode == TRUE) %>%
  group_by(dem_spell_outcome) %>%
  summarize(count = n())
```

```
## # A tibble: 2 x 2
##   dem_spell_outcome count
##   <fct>             <int>
## 1 autocracy            33
## 2 democracy            41
```

Label all country-years within spell, saying whether spell included erosion

```
vdem <- vdem %>%
  group_by(dem_spell_name) %>%
  mutate(dem_spell_erosion = case_when(max(vdem$v2x_polyarchy, na.rm = TRUE) < dem_threshold ~ as.logical(NA),
                                       max(vdem$dem_spell_peak, na.rm = TRUE) < (erosion_threshold * sd(vdem$v2x_polyarchy[v
dem$v2x_polyarchy >= dem_threshold])) ~ as.logical(NA),
                                       sum(erode, na.rm = TRUE) > 0 ~ TRUE,
                                       TRUE ~ FALSE)) %>%
  ungroup()
summary(vdem$dem_spell_erosion)
```

```
##    Mode   FALSE    TRUE 
## logical   24658     887
```

```
vdem$dem_spell_outcome <- case_when(vdem$dem_spell_outcome == 'autocracy' ~ 'autocracy',
                                    vdem$dem_spell_erosion == TRUE ~ 'erosion',
                                    TRUE ~ as.character(vdem$dem_spell_outcome))
vdem$dem_spell_outcome <- as.factor(vdem$dem_spell_outcome)
summary(vdem$dem_spell_outcome)
```

```
## autocracy democracy   erosion      NA's 
##      1132      3321       564     20528
```

Review available cases against which to build predictive models Count and itemize consolidated democracies, by outcome

```
vdem %>%
  filter(consolidated_lhb == TRUE) %>%
  group_by(dem_spell_outcome) %>%
  summarize(n_distinct(dem_spell_name))
```

```
## # A tibble: 3 x 2
##   dem_spell_outcome `n_distinct(dem_spell_name)`
##   <fct>                                    <int>
## 1 autocracy                                    6
## 2 democracy                                   42
## 3 erosion                                      8
```

```
interlocked_list <-
vdem %>%
  filter(consolidated_lhb == TRUE) %>%
  group_by('Democratic Spell' = dem_spell_name) %>%
  summarize(Outcome = case_when(sum(dem_spell_outcome == 'autocracy') > 0 ~ 'autocratized',
                                sum(dem_spell_erosion == TRUE) > 0 ~ 'eroded',
                                TRUE ~ 'remained democratic')) %>%
  arrange('Democratic Spell') %>%
  print(n = 100)
```

```
## # A tibble: 56 x 2
##    `Democratic Spell`            Outcome            
##    <chr>                         <chr>              
##  1 Argentina 1984                remained democratic
##  2 Armenia 2019                  remained democratic
##  3 Australia 1858                remained democratic
##  4 Austria 1945                  remained democratic
##  5 Belgium 1947                  remained democratic
##  6 Bolivia 1986                  autocratized       
##  7 Brazil 1987                   eroded             
##  8 Canada 1921                   remained democratic
##  9 Chile 1990                    remained democratic
## 10 Costa Rica 1950               remained democratic
## 11 Croatia 2000                  eroded             
## 12 Cyprus 1974                   remained democratic
## 13 Czech Republic 1990           remained democratic
## 14 Denmark 1902                  autocratized       
## 15 Denmark 1946                  remained democratic
## 16 Estonia 1993                  remained democratic
## 17 Finland 1918                  eroded             
## 18 France 1947                   eroded             
## 19 Germany 1949                  remained democratic
## 20 Greece 1975                   remained democratic
## 21 Hungary 1990                  autocratized       
## 22 Iceland 1904                  remained democratic
## 23 Ireland 1922                  remained democratic
## 24 Israel 1949                   remained democratic
## 25 Italy 1947                    remained democratic
## 26 Jamaica 1984                  remained democratic
## 27 Japan 1952                    remained democratic
## 28 Latvia 1923                   autocratized       
## 29 Latvia 1990                   remained democratic
## 30 Lithuania 1990                remained democratic
## 31 Luxembourg 1946               remained democratic
## 32 Malta 1963                    remained democratic
## 33 Mauritius 1968                eroded             
## 34 Netherlands 1946              remained democratic
## 35 New Zealand 1861              remained democratic
## 36 Norway 1946                   remained democratic
## 37 Panama 1991                   remained democratic
## 38 Peru 2001                     remained democratic
## 39 Poland 1990                   eroded             
## 40 Portugal 1976                 remained democratic
## 41 Romania 1991                  remained democratic
## 42 Slovakia 1993                 remained democratic
## 43 Slovenia 1990                 eroded             
## 44 South Africa 1995             remained democratic
## 45 South Korea 1988              eroded             
## 46 Spain 1978                    remained democratic
## 47 Suriname 1992                 remained democratic
## 48 Sweden 1922                   remained democratic
## 49 Switzerland 1849              remained democratic
## 50 Taiwan 1996                   remained democratic
## 51 Trinidad and Tobago 1962      remained democratic
## 52 United Kingdom 1919           remained democratic
## 53 United States of America 1921 remained democratic
## 54 Uruguay 1939                  autocratized       
## 55 Uruguay 1985                  remained democratic
## 56 Venezuela 1959                autocratized
```

```
write.table(interlocked_list, file = "./visuals/interlocked_list.txt", sep = ",", quote = FALSE, row.names = F)
```

Examine cases where consolidated regime autocratized Note that some (eg Denmark 1902) ended in foreign occupation

```
vdem %>% filter(consolidated_lhb == TRUE, dem_spell_outcome == 'autocracy') %>%
  distinct(dem_spell_name)
```

```
## # A tibble: 6 x 1
##   dem_spell_name
##   <chr>
## 1 Bolivia 1986
## 2 Venezuela 1959
## 3 Latvia 1923
## 4 Uruguay 1939
## 5 Denmark 1902
## 6 Hungary 1990
```

Examine cases where consolidated regime eroded but didn’t autocratize

```
vdem %>% filter(consolidated_lhb == TRUE, dem_spell_outcome == 'erosion') %>%
  distinct(dem_spell_name)
```

```
## # A tibble: 8 x 1
##   dem_spell_name
##   <chr>
## 1 Poland 1990
## 2 Brazil 1987
## 3 South Korea 1988
## 4 France 1947
## 5 Croatia 2000
## 6 Finland 1918
## 7 Mauritius 1968
## 8 Slovenia 1990
```

Spin off data frames for all democratic country-years and for all country-years within democratic spells that ever consolidated

```
vdem_dem = vdem %>% filter(v2x_polyarchy >= dem_threshold)
vdem <- vdem %>%
  group_by(dem_spell_name) %>%
  mutate(ever_consolidated = if_else(sum(consolidated_lhb == TRUE, na.rm = TRUE) > 0,
                                     TRUE,
                                     FALSE)) %>%
  ungroup()
vdem_con <- vdem %>%
  filter(ever_consolidated == TRUE)
```

Summary statistics

Year range - full data set

```
summary(vdem$year)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1789    1894    1943    1933    1984    2020
```

first year with democracy

```
vdem %>% filter(v2x_polyarchy >= dem_threshold) %>% summarize(min(year))
```

```
## # A tibble: 1 x 1
##   `min(year)`
##         <dbl>
## 1        1849
```

first year with consolidated democracy

```
vdem %>% filter(consolidated_lhb == TRUE) %>% summarize(min(year))
```

```
## # A tibble: 1 x 1
##   `min(year)`
##         <dbl>
## 1        1900
```

Range for clientelism variable

```
vdem %>% filter(!is.na(v2xnp_client)) %>% summarize(min(year), max(year), n_distinct(country_name))
```

```
## # A tibble: 1 x 3
##   `min(year)` `max(year)` `n_distinct(country_name)`
##         <dbl>       <dbl>                      <int>
## 1        1789        2020                        199
```

Range for polarization variable

```
vdem %>% filter(!is.na(v2cacamps)) %>% summarize(min(year), max(year), n_distinct(country_name))
```

```
## # A tibble: 1 x 3
##   `min(year)` `max(year)` `n_distinct(country_name)`
##         <dbl>       <dbl>                      <int>
## 1        1900        2020                        178
```

Range for online usage and fractionalization measures

```
vdem %>% filter(!is.na(smonexXsmmefra)) %>% summarize(min(year), max(year), n_distinct(country_name))
```

```
## # A tibble: 1 x 3
##   `min(year)` `max(year)` `n_distinct(country_name)`
##         <dbl>       <dbl>                      <int>
## 1        2000        2020                        179
```

Range for party disinformation

```
vdem %>% filter(!is.na(v2smpardom)) %>% summarize(min(year), max(year), n_distinct(country_name))
```

```
## # A tibble: 1 x 3
##   `min(year)` `max(year)` `n_distinct(country_name)`
##         <dbl>       <dbl>                      <int>
## 1        2000        2020                        179
```

Identify number of regimes - total

```
n_distinct(vdem$country_name)
```

```
## [1] 199
```

Countries with any democratic spells

```
vdem %>% filter(!is.na(dem_spell_name)) %>% summarize(n_distinct(country_name))
```

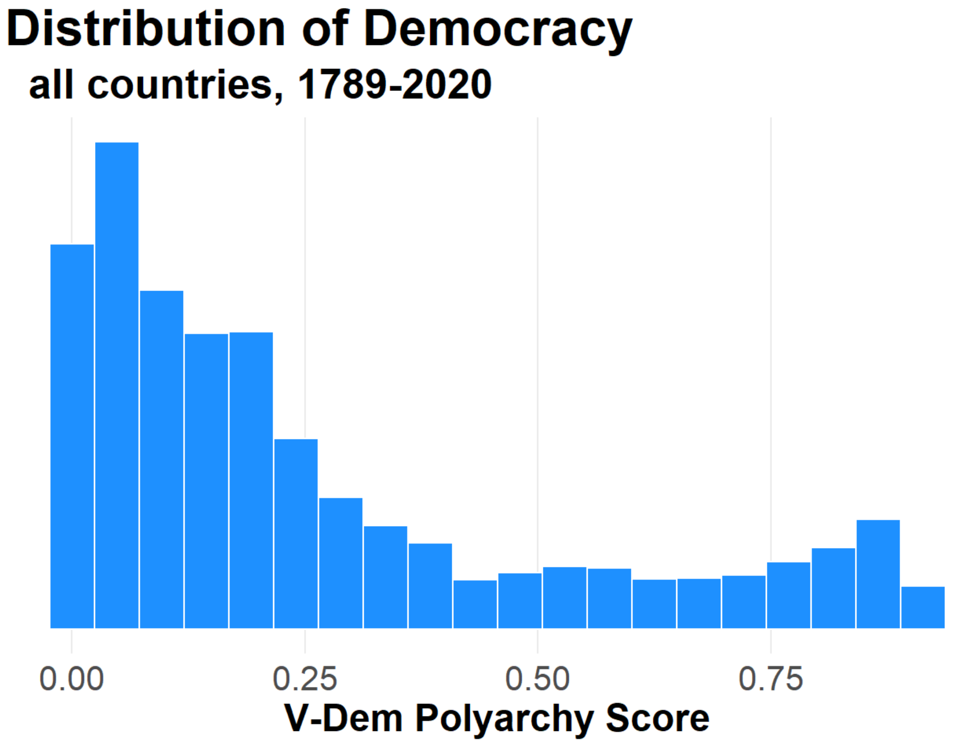
```
## # A tibble: 1 x 1
##   `n_distinct(country_name)`
##                        <int>
## 1                        121
```

Unique democratic spells

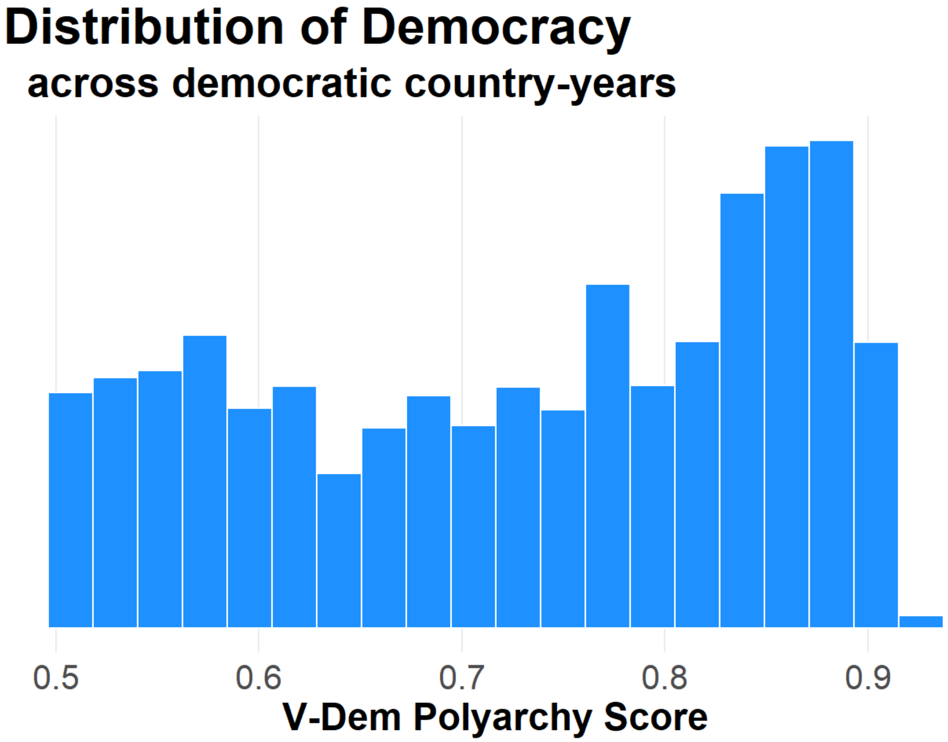
```
vdem %>% filter(!is.na(dem_spell_name)) %>% summarize(n_distinct(dem_spell_name))
```

```
## # A tibble: 1 x 1
##   `n_distinct(dem_spell_name)`
##                          <int>
## 1                          188
```

Distribution of variables

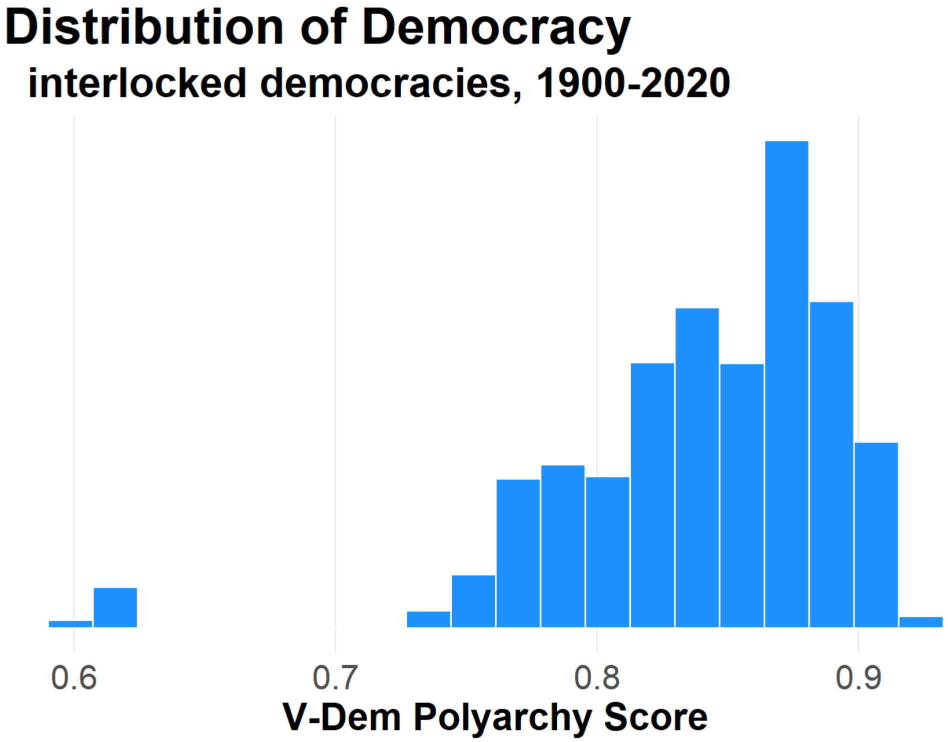


```
vdem %>% filter(v2x_polyarchy >= dem_threshold) %>%
ggplot(aes(x = v2x_polyarchy))+
  geom_histogram(bins = 20, fill = 'dodgerblue', color = 'white')+
  theme_minimal()+
  labs(title = 'Distribution of Democracy',
       subtitle = '  across democratic country-years',
       y = '',
       x = 'V-Dem Polyarchy Score')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank(),
        axis.text = element_text(size = 16),
        axis.text.y = element_blank())
```



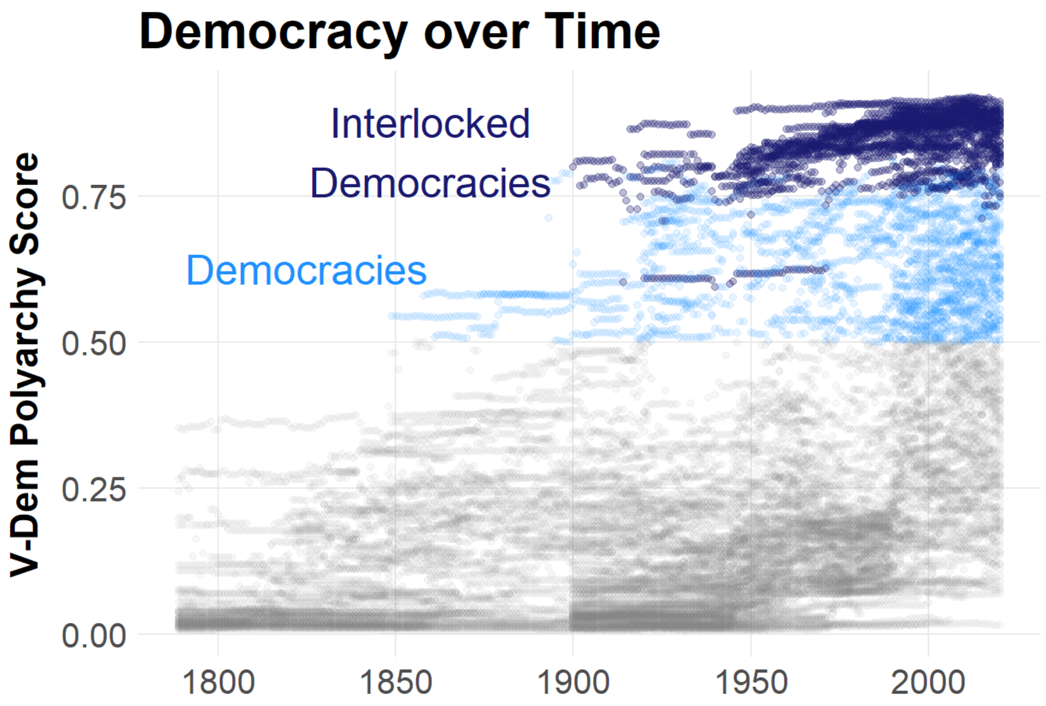
```
ggsave(filename = "./visuals/democracy_histogram_dems.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

```
ggplot(data = (vdem %>% filter(consolidated_lhb == TRUE)), aes(x = v2x_polyarchy))+
  geom_histogram(bins = 20, fill = 'dodgerblue', color = 'white')+
  theme_minimal()+
  labs(title = 'Distribution of Democracy',
       subtitle = '  interlocked democracies, 1900-2020',
       y = '',
       x = 'V-Dem Polyarchy Score')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank(),
        axis.text = element_text(size = 16),
        axis.text.y = element_blank())
```



```
ggsave(filename = "./visuals/democracy_histogram_interlocked.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

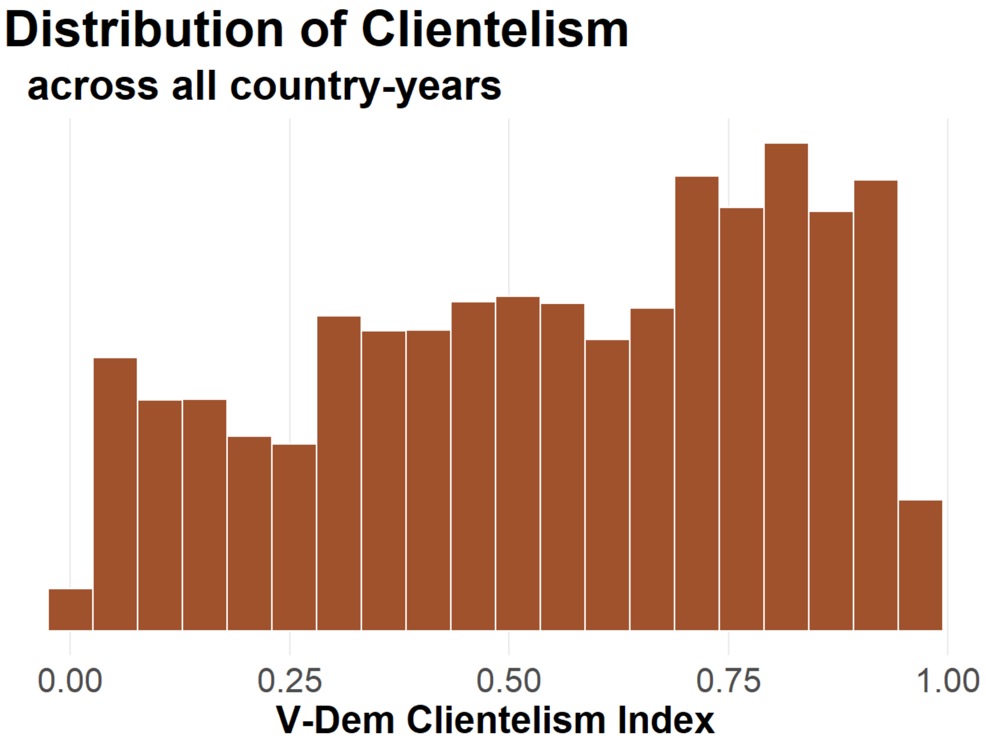
```
ggplot(data = vdem, aes(x = year, y = v2x_polyarchy))+
  geom_point(aes(color = consolidated_lhb == TRUE, alpha = consolidated_lhb == TRUE))+
  scale_color_manual(values = c(dem_color, interlock_color), na.value = 'gray50')+
  scale_alpha_manual(values = c(0.1, 0.3), na.value = 0.06)+
  theme_minimal()+
  labs(title = 'Democracy over Time',
       #subtitle = '  noting interlocked democracies',
       x = '',
       y = 'V-Dem Polyarchy Score')+
  annotate('text', label = 'Democracies', color = dem_color, size = 7, x = 1825, y = 0.625)+
  annotate('text', label = 'Interlocked\nDemocracies', color = interlock_color, size = 7, x = 1860, y = 0.825)+
  theme(legend.position = 'none',
        title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16),
        axis.title.y = element_text(margin = margin(r = 8)))
```



```
ggsave(filename = "./visuals/democracy_interlocked_over_time.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

```
vdem %>%
  ggplot(aes(x = v2xnp_client))+
  geom_histogram(bins = 20, fill = client_color, color = 'white')+
  theme_minimal()+
  labs(title = 'Distribution of Clientelism',
       subtitle = '  across all country-years',
       y = '',
       x = 'V-Dem Clientelism Index')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank(),
        axis.text = element_text(size = 16),
        axis.text.y = element_blank())
```

```
## Warning: Removed 1413 rows containing non-finite values (stat_bin).
```

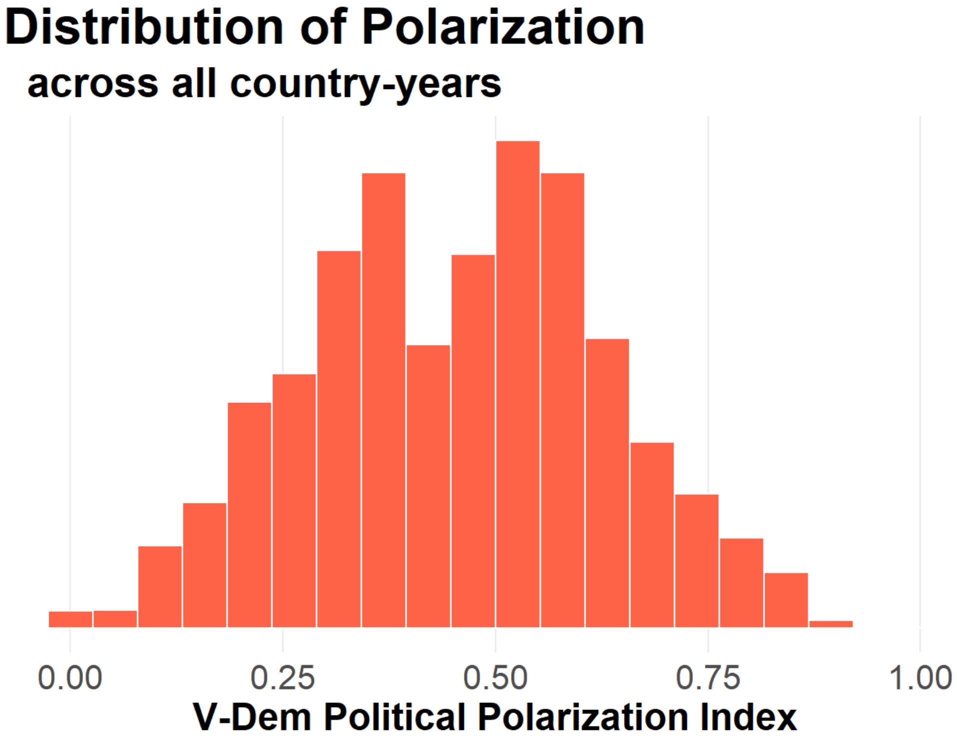


```
ggsave(filename = "./visuals/client_histogram.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

```
## Warning: Removed 1413 rows containing non-finite values (stat_bin).
```

```
vdem %>%
  ggplot(aes(x = v2cacamps))+
  geom_histogram(bins = 20, fill = polar_color, color = 'white')+
  theme_minimal()+
  labs(title = 'Distribution of Polarization',
       subtitle = '  across all country-years',
       y = '',
       x = 'V-Dem Political Polarization Index')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank(),
        axis.text = element_text(size = 16),
        axis.text.y = element_blank())
```

```
## Warning: Removed 7643 rows containing non-finite values (stat_bin).
```

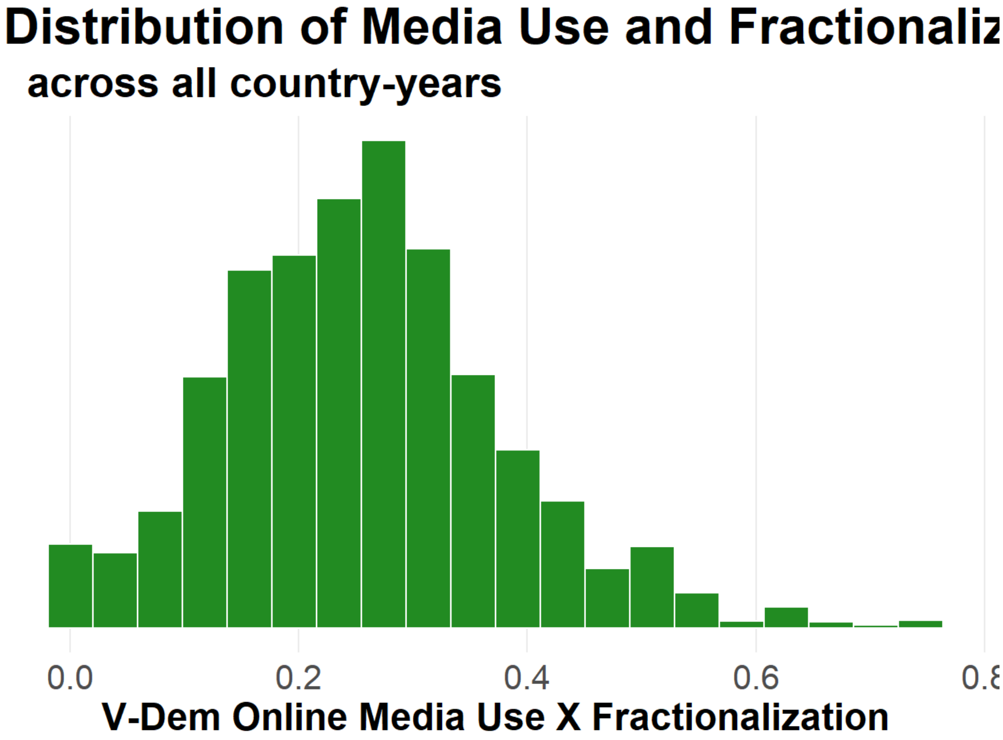


```
ggsave(filename = "./visuals/polar_histogram.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

```
## Warning: Removed 7643 rows containing non-finite values (stat_bin).
```

```
vdem %>%
  ggplot(aes(x = smonexXsmmefra))+
  geom_histogram(bins = 20, fill = media_color, color = 'white')+
  theme_minimal()+
  labs(title = 'Distribution of Media Use and Fractionalization',
       subtitle = '  across all country-years',
       y = '',
       x = 'V-Dem Online Media Use X Fractionalization')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank(),
        axis.text = element_text(size = 16),
        axis.text.y = element_blank())
```

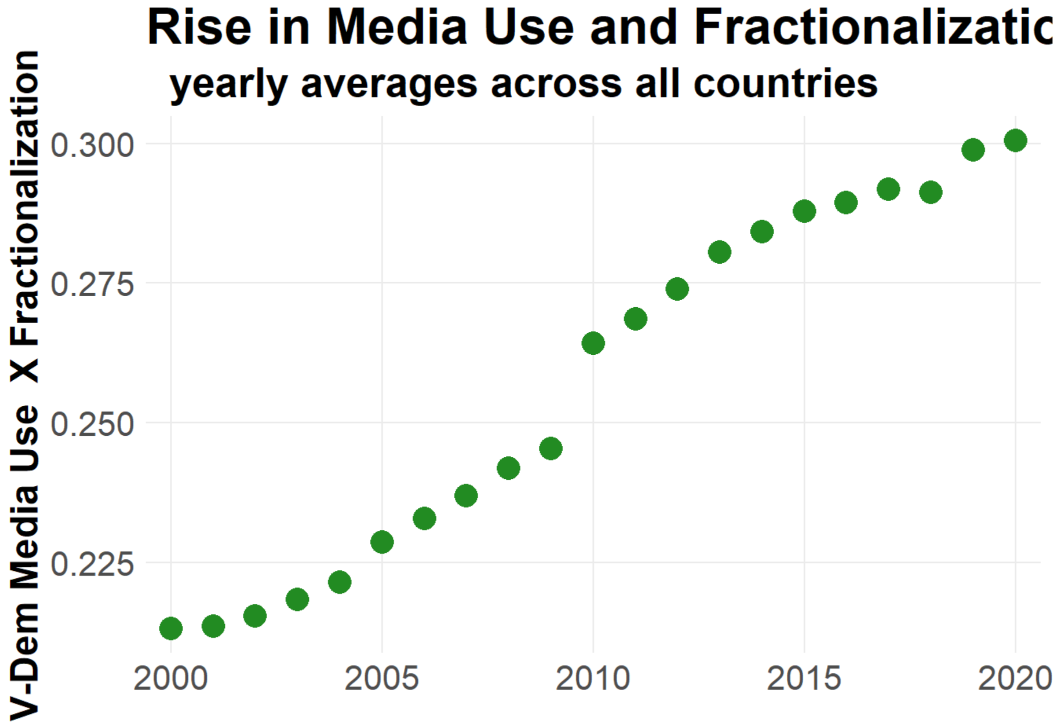
```
## Warning: Removed 21804 rows containing non-finite values (stat_bin).
```



```
ggsave(filename = "./visuals/media_histogram.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

```
## Warning: Removed 21804 rows containing non-finite values (stat_bin).
```

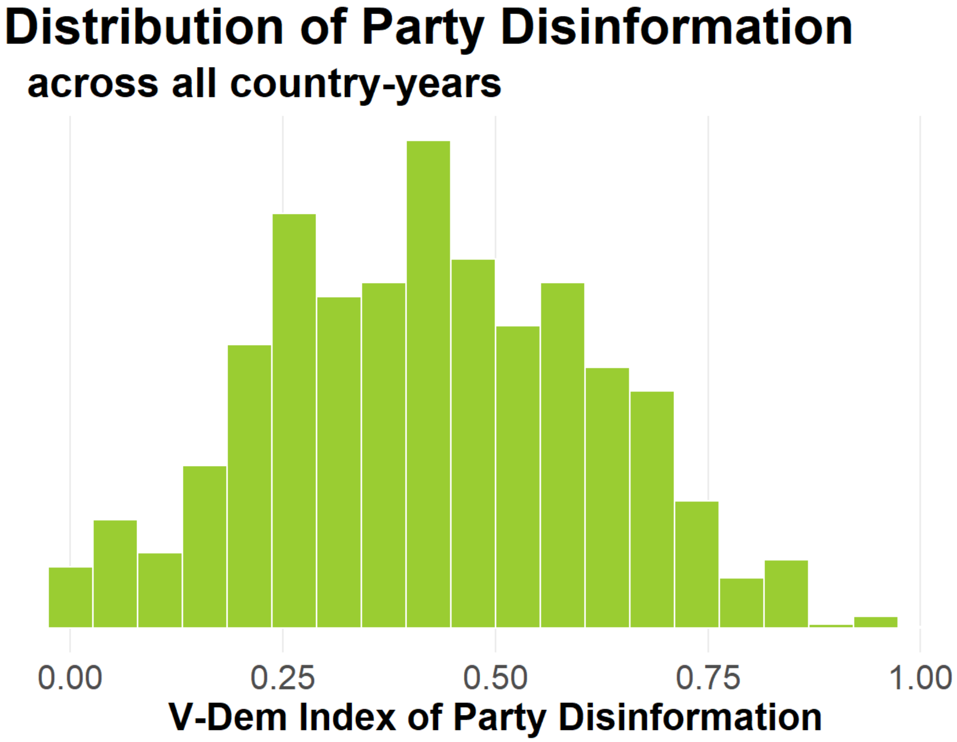
```
vdem %>%
  filter(!is.na(smonexXsmmefra)) %>%
  group_by(year_factor = as.factor(year)) %>%
  summarize(avg_smonexXsmmefra = mean(smonexXsmmefra)) %>%
  ggplot(aes(x = year_factor, y = avg_smonexXsmmefra))+
  geom_point(size = 5, color = media_color)+
  theme_minimal()+
  scale_x_discrete(breaks = c(2000,2005,2010,2015,2020))+
  labs(title = 'Rise in Media Use and Fractionalization',
       subtitle = '  yearly averages across all countries',
       y = 'V-Dem Media Use  X Fractionalization',
       x = '')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16))
```



```
ggsave(filename = "./visuals/media_chronology.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

```
vdem %>%
  ggplot(aes(x = v2smpardom))+
  geom_histogram(bins = 20, fill = disinfo_color, color = 'white')+
  theme_minimal()+
  labs(title = 'Distribution of Party Disinformation',
       subtitle = '  across all country-years',
       y = '',
       x = 'V-Dem Index of Party Disinformation')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank(),
        axis.text = element_text(size = 16),
        axis.text.y = element_blank())
```

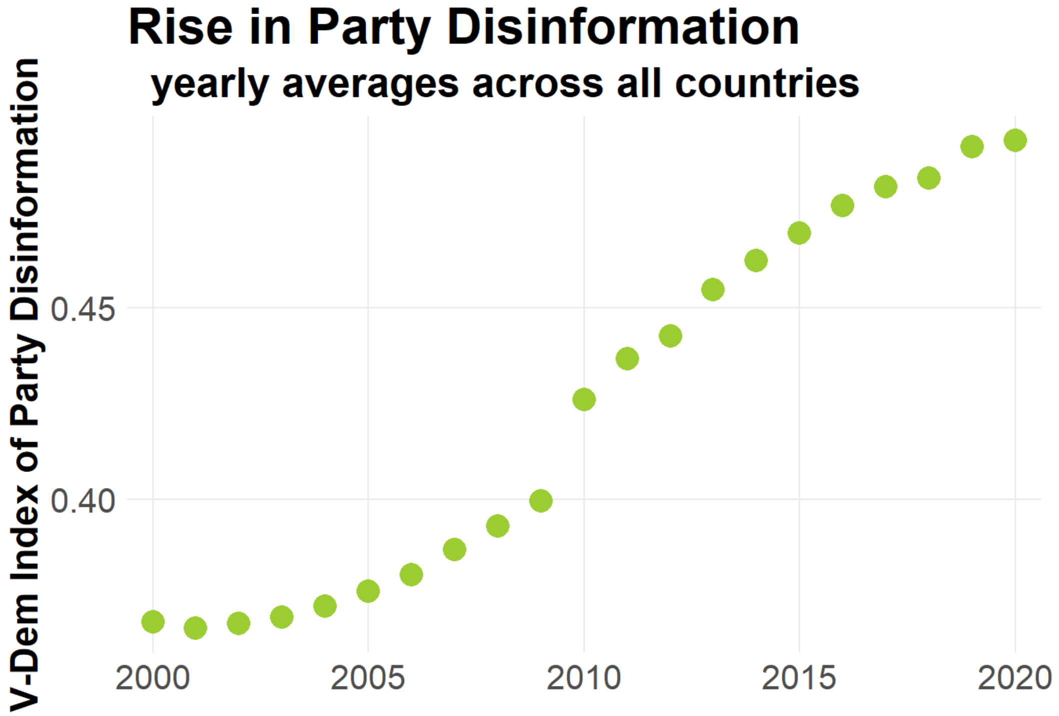
```
## Warning: Removed 21804 rows containing non-finite values (stat_bin).
```



```
ggsave(filename = "./visuals/par_disinfo_histogram.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

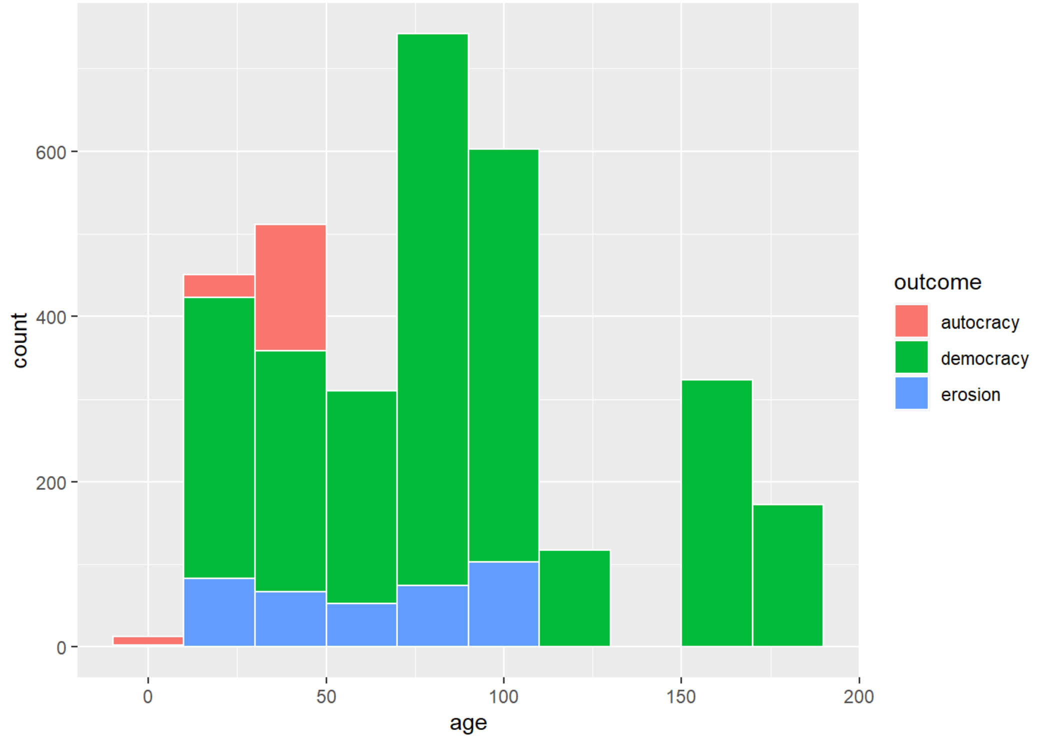
```
## Warning: Removed 21804 rows containing non-finite values (stat_bin).
```

```
vdem %>%
  filter(!is.na(v2smpardom)) %>%
  group_by(year_factor = as.factor(year)) %>%
  summarize(avg_v2smpardom = mean(v2smpardom)) %>%
  ggplot(aes(x = year_factor, y = avg_v2smpardom))+
  geom_point(size = 5, color = disinfo_color)+
  theme_minimal()+
  scale_x_discrete(breaks = c(2000,2005,2010,2015,2020))+
  labs(title = 'Rise in Party Disinformation',
       subtitle = '  yearly averages across all countries',
       y = 'V-Dem Index of Party Disinformation',
       x = '')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16))
```



```
ggsave(filename = "./visuals/par_disinfo_chronology.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

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



Demonstrate possibility of ephemeral spells when using a single threshold like 0.5

```
vdem %>% filter(country_name == 'Albania', between(year, 2003, 2020)) %>%
  select(country_name, year, v2x_polyarchy) %>%
  print(n = 20)
```

```
## # A tibble: 18 x 3
##    country_name  year v2x_polyarchy
##    <chr>        <dbl>         <dbl>
##  1 Albania       2003         0.499
##  2 Albania       2004         0.5
##  3 Albania       2005         0.522
##  4 Albania       2006         0.548
##  5 Albania       2007         0.551
##  6 Albania       2008         0.552
##  7 Albania       2009         0.551
##  8 Albania       2010         0.546
##  9 Albania       2011         0.546
## 10 Albania       2012         0.543
## 11 Albania       2013         0.51
## 12 Albania       2014         0.518
## 13 Albania       2015         0.521
## 14 Albania       2016         0.521
## 15 Albania       2017         0.538
## 16 Albania       2018         0.52
## 17 Albania       2019         0.494
## 18 Albania       2020         0.501
```

```
vdem %>% filter(country_name == 'Kosovo', between(year, 2001, 2020)) %>%
  select(country_name, year, v2x_polyarchy) %>%
  print(n = 20)
```

```
## # A tibble: 20 x 3
##    country_name  year v2x_polyarchy
##    <chr>        <dbl>         <dbl>
##  1 Kosovo        2001         0.246
##  2 Kosovo        2002         0.516
##  3 Kosovo        2003         0.517
##  4 Kosovo        2004         0.514
##  5 Kosovo        2005         0.494
##  6 Kosovo        2006         0.494
##  7 Kosovo        2007         0.5
##  8 Kosovo        2008         0.494
##  9 Kosovo        2009         0.495
## 10 Kosovo        2010         0.501
## 11 Kosovo        2011         0.47
## 12 Kosovo        2012         0.476
## 13 Kosovo        2013         0.48
## 14 Kosovo        2014         0.544
## 15 Kosovo        2015         0.6
## 16 Kosovo        2016         0.593
## 17 Kosovo        2017         0.529
## 18 Kosovo        2018         0.606
## 19 Kosovo        2019         0.609
## 20 Kosovo        2020         0.611
```

Predict eventual autocratization or erosion

Produce a few illustrative statistics

Highest polyarchy peak that later autocratized

```
vdem %>%
  filter(dem_spell_outcome == 'autocracy') %>%
  group_by(dem_spell_name) %>%
  summarize(farthest_faller = max(dem_spell_peak), last_year = max(year)) %>%
  arrange(desc(farthest_faller))
```

```
## # A tibble: 93 x 3
##    dem_spell_name      farthest_faller last_year
##    <chr>                         <dbl>     <dbl>
##  1 Denmark 1902                  0.873      1942
##  2 Hungary 1990                  0.859      2017
##  3 Estonia 1990                  0.809      1991
##  4 Czech Republic 1920           0.807      1938
##  5 Uruguay 1939                  0.803      1972
##  6 Bolivia 1986                  0.767      2019
##  7 Venezuela 1959                0.761      2002
##  8 Latvia 1923                   0.757      1933
##  9 India 1977                    0.756      2018
## 10 Suriname 1950                 0.752      1979
## # ... with 83 more rows
```

Distribution of lowest polyarchy scores after a country reached a high threshold

First identify mean and sd for all democratic country-years

```
dems_polyarchy_mean <- vdem %>%
  filter(v2x_polyarchy >= dem_threshold) %>%
  summarize(dems_polyarchy_mean = mean(v2x_polyarchy)) %>%
  pull(dems_polyarchy_mean)
dems_polyarchy_sd <- vdem %>%
  filter(v2x_polyarchy >= dem_threshold) %>%
  summarize(dems_polyarchy_sd = sd(v2x_polyarchy)) %>%
  pull(dems_polyarchy_sd)
```

Then identify all democratic spells that ever exceeded 1 sd above democratic mean

```
high_dems_reference_set <-
vdem %>%
  filter(v2x_polyarchy >= dems_polyarchy_mean + dems_polyarchy_sd) %>%
  group_by(country_name) %>%
  summarize(year_above_highthreshold = min(year))
```

Then log the lowest polyarchy score in a spell after it exceeded 1 sd above mean

```
high_dems_reference_set$low_after_high <- as.numeric(NA)
for (i in seq_along(high_dems_reference_set$country_name)){
  high_dems_reference_set$low_after_high[i] <-
    vdem %>% filter(country_name == high_dems_reference_set$country_name[i] &
                    year >= high_dems_reference_set$year_above_highthreshold[i]) %>%
    summarize(low_after_high = min(v2x_polyarchy, na.rm = TRUE)) %>%
    pull(low_after_high)
}
```

Identify lowest sinking democracies that previously exceeded 1 sd above mean

```
high_dems_reference_set %>% arrange(low_after_high)
```

```
## # A tibble: 35 x 3
##    country_name             year_above_highthreshold low_after_high
##    <chr>                                       <dbl>          <dbl>
##  1 Denmark                                      1916          0.206
##  2 Hungary                                      2002          0.465
##  3 Poland                                       1992          0.632
##  4 Brazil                                       1994          0.69
##  5 Croatia                                      2011          0.732
##  6 Slovenia                                     2009          0.748
##  7 Czech Republic                               1991          0.803
##  8 Chile                                        1994          0.813
##  9 Slovakia                                     2011          0.815
## 10 United States of America                     1981          0.815
## # ... with 25 more rows
```

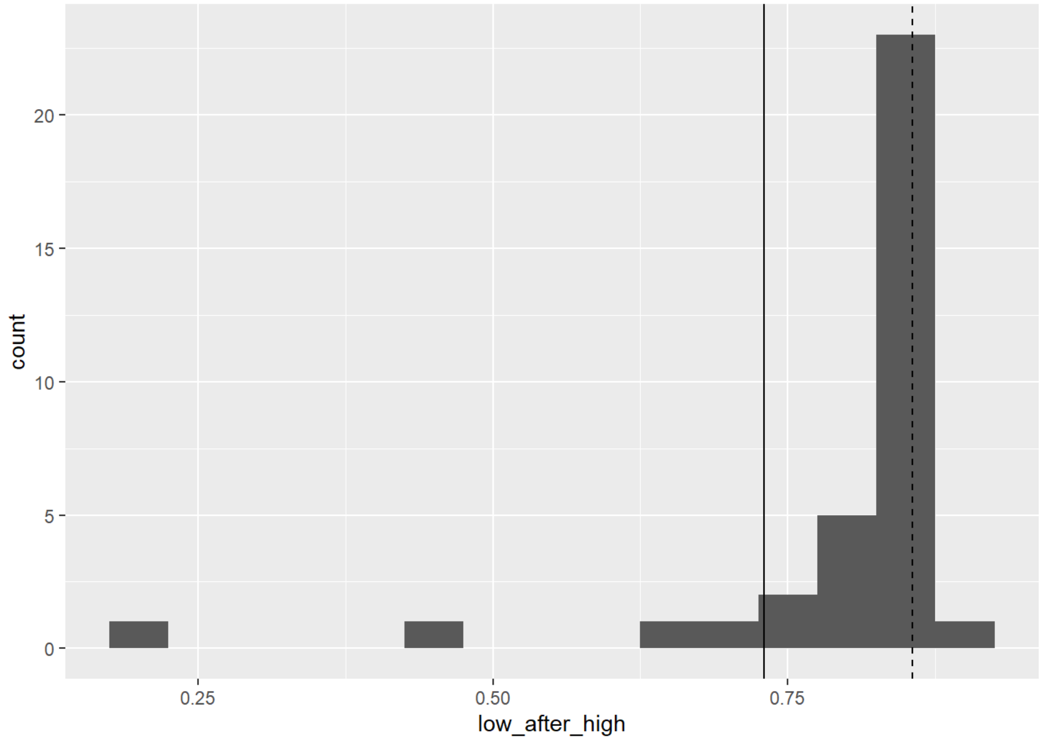
```
count(high_dems_reference_set, autocratized_after_high_threshold = low_after_high < dem_threshold)
```

```
## # A tibble: 2 x 2
##   autocratized_after_high_threshold     n
##   <lgl>                             <int>
## 1 FALSE                                33
## 2 TRUE                                  2
```

```
count(high_dems_reference_set, fell_below_avg_dem_after_high_threshold = low_after_high < dems_polyarchy_mean)
```

```
## # A tibble: 2 x 2
##   fell_below_avg_dem_after_high_threshold     n
##   <lgl>                                   <int>
## 1 FALSE                                      31
## 2 TRUE                                        4
```

Chart distribution of lowest polyarchy scores after exceeding 1 sd above mean



Examine independent variables

clientelism

```
summary(vdem$v2xnp_client) #clientelism index, rolls up psprlnks and dlencmps plus additional variables
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##  0.0180  0.3390  0.5760  0.5524  0.7890  0.9870    1413
```

```
summary(vdem$v2psprlnks) #how parties link to constituents
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##  0.0000  0.3699  0.5526  0.5234  0.6883  1.0000    1475
```

```
summary(vdem$v2dlencmps) #particularistic social spending
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##  0.0000  0.3538  0.4848  0.5088  0.6710  1.0000      97
```

mutual correlations

```
vdem %>% select(v2xnp_client, v2psprlnks, v2dlencmps) %>%
  na.omit() %>%
  cor()
```

```
##              v2xnp_client v2psprlnks v2dlencmps
## v2xnp_client    1.0000000  0.8969867  0.7415204
## v2psprlnks      0.8969867  1.0000000  0.5247795
## v2dlencmps      0.7415204  0.5247795  1.0000000
```

information control

```
summary(vdem$v2smonex) # online media consumption
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.000   0.387   0.527   0.523   0.660   1.000   21804
```

```
summary(vdem$v2smmefra) # online media fractionalization
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.000   0.390   0.510   0.495   0.609   1.000   21804
```

```
summary(vdem$v2smgovdom) # government disseminates false info
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.000   0.272   0.427   0.429   0.594   1.000   21804
```

```
summary(vdem$v2smpardom) # party disseminates false info
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.000   0.276   0.411   0.424   0.567   1.000   21804
```

```
summary(vdem$v2smfordom) # foreign governments inject false info
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.000   0.258   0.359   0.373   0.491   1.000   21804
```

mutual correlations

```
vdem %>% select(v2smonex, v2smmefra, v2smgovdom, v2smpardom, v2smfordom) %>%
  na.omit() %>%
  cor()
```

```
##               v2smonex   v2smmefra  v2smgovdom v2smpardom v2smfordom
## v2smonex    1.00000000 -0.05243952 -0.36156206 -0.2753955 0.00981006
## v2smmefra  -0.05243952  1.00000000  0.08457634  0.1192835 0.17738122
## v2smgovdom -0.36156206  0.08457634  1.00000000  0.8276953 0.43776226
## v2smpardom -0.27539549  0.11928347  0.82769534  1.0000000 0.41088195
## v2smfordom  0.00981006  0.17738122  0.43776226  0.4108819 1.00000000
```

polarization

```
summary(vdem$v2cacamps) # political polarization extends into society
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   0.323   0.464   0.455   0.576   1.000    7643
```

```
summary(vdem$v2smpolsoc) # societal polarization
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   0.430   0.580   0.568   0.699   1.000   21804
```

```
summary(vdem$v2smpolhate) # parties use hate speech
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   0.300   0.441   0.443   0.590   1.000   21804
```

mutual correlations

```
vdem %>% select(v2cacamps, v2smpolsoc, v2smpolhate) %>%
  na.omit() %>%
  cor()
```

```
##             v2cacamps v2smpolsoc v2smpolhate
## v2cacamps   1.0000000  0.6446798   0.6035975
## v2smpolsoc  0.6446798  1.0000000   0.5557396
## v2smpolhate 0.6035975  0.5557396   1.0000000
```

Model to correlate corrosive factors to reductions in democracy

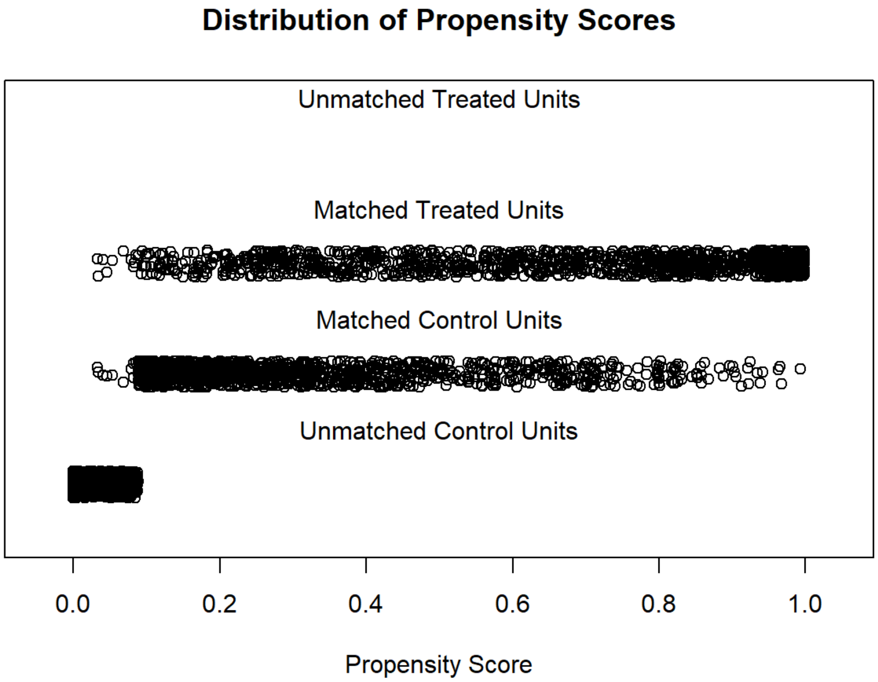
nomenclature *m1 = all democratic cases, predict autocratization* m2 = consolidated cases, predict autocratization *m3 = all democratic cases, predict erosion* m4 = consolidated cases, predict erosion *m5 = consolidated cases, predict autocratization using matching for controls* m6 = consolidated cases, predict erosion using matching for controls *m7 = consolidated cases, predict [autocratization or erosion]* m8 = consolidated cases, predict [autocratization or erosion] using matching

Whenever matches are made as control groups, plots show the propensity distribution, which suggests whether matches are good.

model clientelism

```
cm1 <- lm(dem_spell_outcome == 'autocracy' ~ v2xnp_client + v2x_polyarchy + e_migdppc + as.factor(year), data = vdem_dem)
cm2 <- lm(dem_spell_outcome == 'autocracy' ~ v2xnp_client + v2x_polyarchy + e_migdppc + as.factor(year), data = vdem_con)
cm3 <- lm(dem_spell_erosion == TRUE ~ v2xnp_client + v2x_polyarchy + e_migdppc + as.factor(year), data = vdem_dem)
cm4 <- lm(dem_spell_erosion == TRUE ~ v2xnp_client + v2x_polyarchy + e_migdppc + as.factor(year), data = vdem_con)
```

```
match_client <- matchit(v2xnp_client >= mean(vdem_con$v2xnp_client, na.rm = TRUE) ~
                          v2x_polyarchy + e_migdppc + as.factor(year),
                        data = vdem_con %>% filter(!is.na(v2x_polyarchy), !is.na(e_migdppc)),
                        method = 'nearest', distance = 'glm')
plot(match_client, type = "jitter", interactive = FALSE)
```

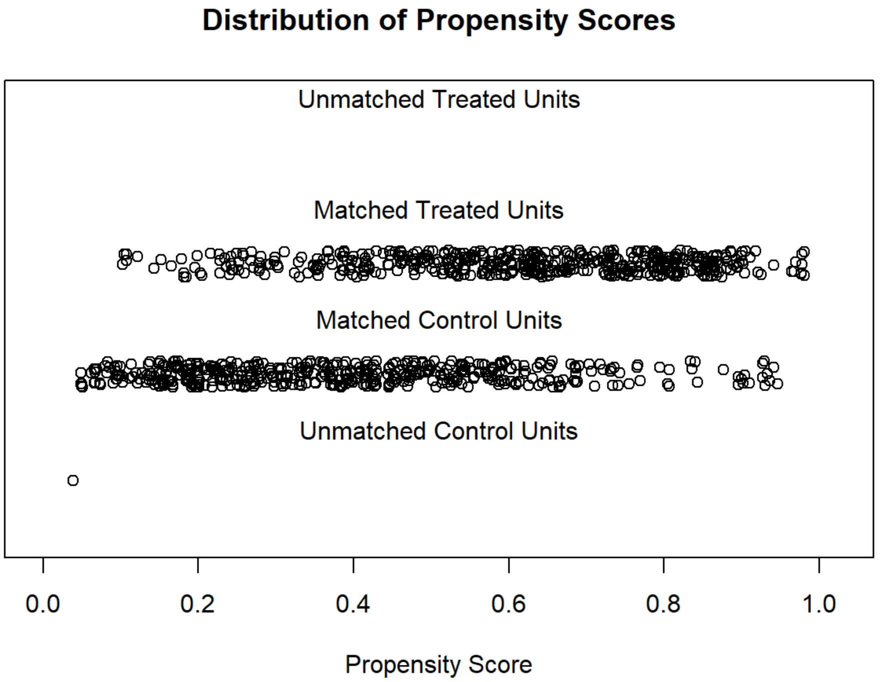


```
vdem_client <- match.data(match_client)
cm5 <- lm(dem_spell_outcome == 'autocracy' ~ v2xnp_client + v2x_polyarchy + e_migdppc + as.factor(year),
          data = vdem_client,
          weights = weights)

cm6 <- lm(dem_spell_erosion == TRUE ~ v2xnp_client + v2x_polyarchy + e_migdppc + as.factor(year),
          data = vdem_client,
          weights = weights)

cm7 <- lm(dem_spell_outcome != 'democracy' ~ v2xnp_client + v2x_polyarchy + e_migdppc + as.factor(year), data = vdem_con)

cm8 <- lm(dem_spell_outcome != 'democracy' ~ v2xnp_client + v2x_polyarchy + e_migdppc + as.factor(year),
          data = vdem_client,
          weights = weights)
```

model media

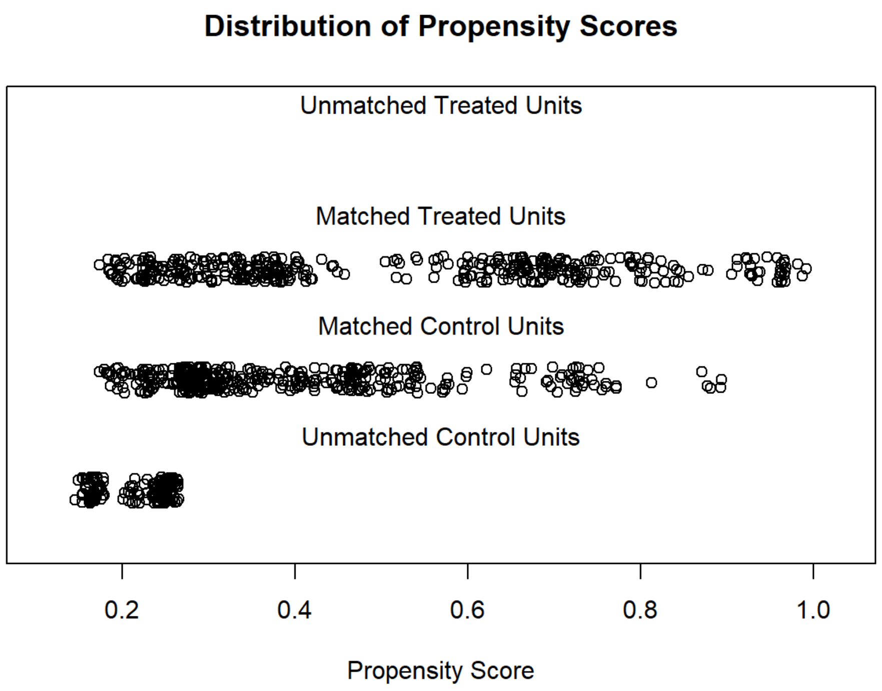
```
mm1_full <- lm(dem_spell_outcome == 'autocracy' ~ v2smonex + v2smmefra + v2smgovdom + v2smpardom + v2smfordom + v2x_polyarch
y + e_migdppc + as.factor(year), data = vdem_dem)

mm2_full <- lm(dem_spell_outcome == 'autocracy' ~ v2smonex + v2smmefra + v2smgovdom + v2smpardom + v2smfordom + v2x_polyarch
y + e_migdppc + as.factor(year), data = vdem_con)

mm3_full <- lm(dem_spell_erosion == TRUE ~ v2smonex + v2smmefra + v2smgovdom + v2smpardom + v2smfordom + v2x_polyarchy + e_m
igdppc + as.factor(year), data = vdem_dem)

mm4_full <- lm(dem_spell_erosion == TRUE ~ v2smonex + v2smmefra + v2smgovdom + v2smpardom + v2smfordom + v2x_polyarchy + e_m
igdppc + as.factor(year), data = vdem_con)

mm7_full <- lm(dem_spell_outcome != 'democracy' ~ v2smonex + v2smmefra + v2smgovdom + v2smpardom + v2smfordom + v2x_polyarch
y + e_migdppc + as.factor(year), data = vdem_con)
```

```
match_consume <- matchit(v2smonex >= mean(vdem_con$v2smonex, na.rm = TRUE) ~
                           v2x_polyarchy + e_migdppc + as.factor(year),
                         data = vdem_con %>% filter(!is.na(v2smonex), !is.na(v2x_polyarchy), !is.na(e_migdppc)),
                         method = 'nearest', distance = 'glm')
plot(match_consume, type = "jitter", interactive = FALSE)
```



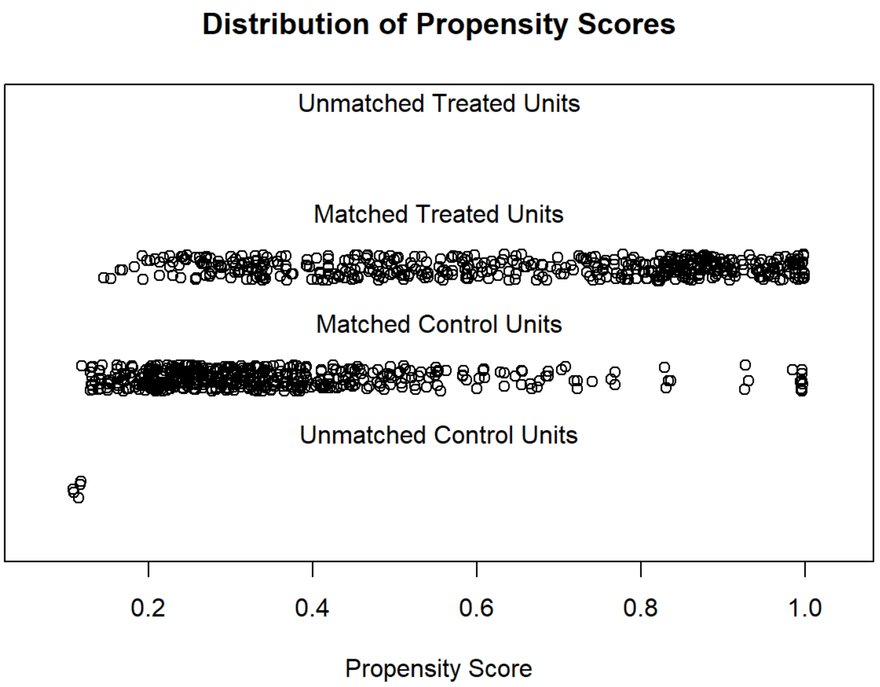
```
vdem_consume <- match.data(match_consume)
mm5_consume <- lm(dem_spell_outcome == 'autocracy' ~ v2smonex + v2x_polyarchy + e_migdppc + as.factor(year),
                  data = vdem_consume,
                  weights = weights)

mm8_consume <- lm(dem_spell_outcome != 'democracy' ~ v2smonex + v2x_polyarchy + e_migdppc + as.factor(year),
                  data = vdem_consume,
                  weights = weights)
```

```
match_fraction <- matchit(v2smmefra >= mean(vdem_con$v2smmefra, na.rm = TRUE) ~
                            v2x_polyarchy + e_migdppc + as.factor(year),
                          data = vdem_con %>% filter(!is.na(v2smmefra), !is.na(v2x_polyarchy), !is.na(e_migdppc)),
                          method = 'nearest', distance = 'glm')
plot(match_fraction, type = "jitter", interactive = FALSE)
```



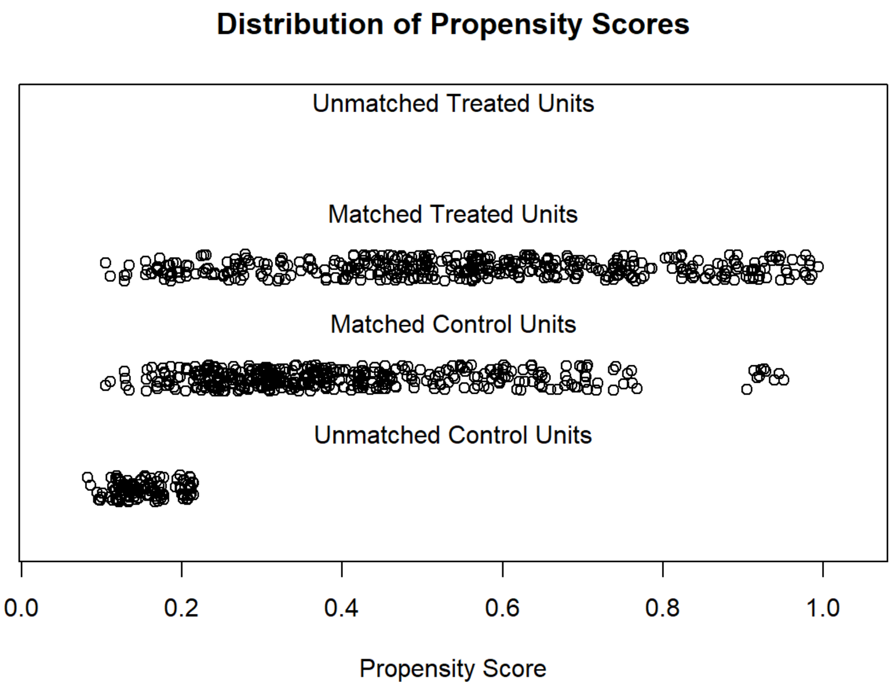
```
vdem_fraction <- match.data(match_fraction)
mm5_fraction <- lm(dem_spell_outcome == 'autocracy' ~ v2smmefra + v2x_polyarchy + e_migdppc + as.factor(year),
                   data = vdem_fraction,
                   weights = weights)

mm8_fraction <- lm(dem_spell_outcome != 'democracy' ~ v2smmefra + v2x_polyarchy + e_migdppc + as.factor(year),
                   data = vdem_fraction,
                   weights = weights)
```

```
match_gov_disinfo <- matchit(v2smgovdom >= mean(vdem_con$v2smgovdom, na.rm = TRUE) ~
                               v2x_polyarchy + e_migdppc + as.factor(year),
                             data = vdem_con %>% filter(!is.na(v2smgovdom), !is.na(v2x_polyarchy), !is.na(e_migdppc)),
                             method = 'nearest', distance = 'glm')
plot(match_gov_disinfo, type = "jitter", interactive = FALSE)
```



```
vdem_gov_disinfo <- match.data(match_gov_disinfo)
mm5_gov <- lm(dem_spell_outcome == 'autocracy' ~ v2smgovdom + v2x_polyarchy + e_migdppc + as.factor(year),
              data = vdem_gov_disinfo,
              weights = weights)

mm8_gov <- lm(dem_spell_outcome != 'democracy' ~ v2smgovdom + v2x_polyarchy + e_migdppc + as.factor(year),
              data = vdem_gov_disinfo,
              weights = weights)
```

```
match_par_disinfo <- matchit(v2smpardom >= mean(vdem_con$v2smpardom, na.rm = TRUE) ~
                              v2x_polyarchy + e_migdppc + as.factor(year),
                            data = vdem_con %>% filter(!is.na(v2smpardom), !is.na(v2x_polyarchy), !is.na(e_migdppc)),
                            method = 'nearest', distance = 'glm')
plot(match_par_disinfo, type = "jitter", interactive = FALSE)
```

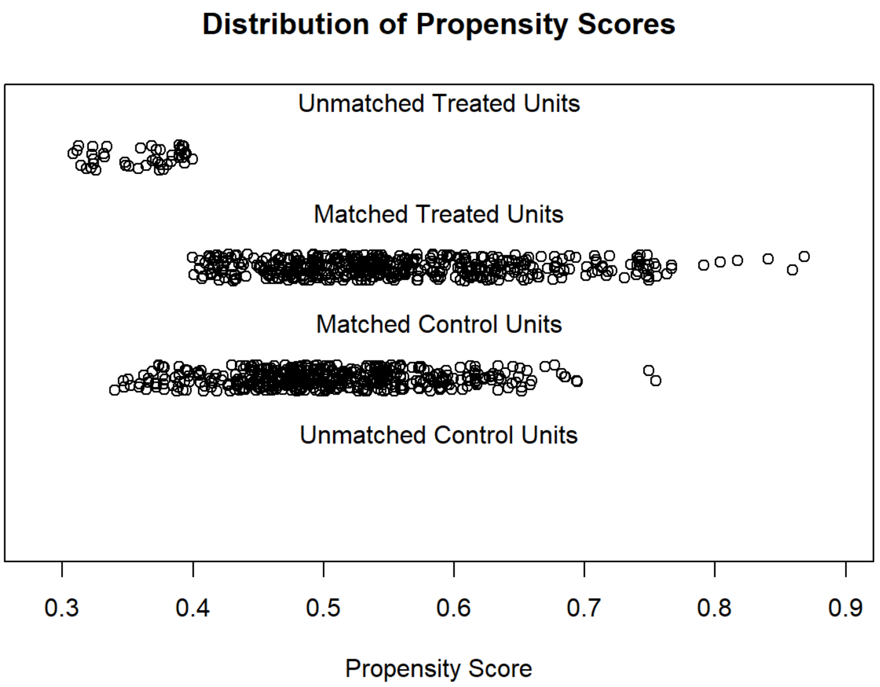


```
vdem_par_disinfo <- match.data(match_par_disinfo)
mm5_par <- lm(dem_spell_outcome == 'autocracy' ~ v2smpardom + v2x_polyarchy + e_migdppc + as.factor(year),
              data = vdem_gov_disinfo,
              weights = weights)

mm8_par <- lm(dem_spell_outcome != 'democracy' ~ v2smpardom + v2x_polyarchy + e_migdppc + as.factor(year),
              data = vdem_gov_disinfo,
              weights = weights)
```

```
match_for_disinfo <- matchit(v2smfordom >= mean(vdem_con$v2smfordom, na.rm = TRUE) ~
                               v2x_polyarchy + e_migdppc + as.factor(year),
                             data = vdem_con %>% filter(!is.na(v2smfordom), !is.na(v2x_polyarchy), !is.na(e_migdppc)),
                             method = 'nearest', distance = 'glm')
```

```
## Warning: Fewer control units than treated units; not all treated units will get
## a match.
```

```
plot(match_for_disinfo, type = "jitter", interactive = FALSE)
```



```
vdem_for_disinfo <- match.data(match_for_disinfo)
mm5_for <- lm(dem_spell_outcome == 'autocracy' ~ v2smfordom + v2x_polyarchy + e_migdppc + as.factor(year),
              data = vdem_for_disinfo,
              weights = weights)

mm8_for <- lm(dem_spell_outcome != 'democracy' ~ v2smfordom + v2x_polyarchy + e_migdppc + as.factor(year),
              data = vdem_for_disinfo,
              weights = weights)
```

model polarization

```
pm1_full <- lm(dem_spell_outcome == 'autocracy' ~ v2cacamps + v2smpolsoc + v2smpolhate + v2x_polyarchy + e_migdppc + as.fact
or(year), data = vdem_dem)

pm2_full <- lm(dem_spell_outcome == 'autocracy' ~ v2cacamps + v2smpolsoc + v2smpolhate + v2x_polyarchy + e_migdppc + as.fact
or(year), data = vdem_con)

pm3_full <- lm(dem_spell_erosion == TRUE ~ v2cacamps + v2smpolsoc + v2smpolhate + v2x_polyarchy + e_migdppc + as.factor(yea
r), data = vdem_dem)

pm4_full <- lm(dem_spell_erosion == TRUE ~ v2cacamps + v2smpolsoc + v2smpolhate + v2x_polyarchy + e_migdppc + as.factor(yea
r), data = vdem_con)

pm7_full <- lm(dem_spell_outcome != 'democracy' ~ v2cacamps + v2smpolsoc + v2smpolhate + v2x_polyarchy + e_migdppc + as.fact
or(year), data = vdem_con)
```

Search for most telling time lag

Write custom function to find optimal time lag for regression on any set of variables

```
test_lags <- function(df = vdem, vars) {
  model_results <- data.frame(time_lag = c(1:lag_range),
                              regression_coef = as.numeric(NA),
                              p_value = as.numeric(NA))

  for (i in 1:lag_range){
    wdf <- df %>%
      group_by(country_name) %>%
      arrange(year) %>%
      mutate(v2x_polyarchy_lagged = lead(v2x_polyarchy, n = i)) %>%
      ungroup()
    wdf$polyarchy_change <- wdf$v2x_polyarchy_lagged - wdf$v2x_polyarchy
    wdf$year_factor <- as.factor(wdf$year)
    wdf <- wdf %>% filter(consolidated_lhb == TRUE)

    wm <- lm(polyarchy_change ~ ., data = wdf[,c('polyarchy_change', vars, 'year_factor')])
    model_results$regression_coef[i] <- summary(wm)$coefficients[2,'Estimate'] #coefficient
    model_results$p_value[i] <- summary(wm)$coefficients[2, 4] #p-value
  }
  print(model_results)
  print(model_results[model_results %>%
                        filter(p_value < 0.05) %>%
                        summarize(strongest_prediction = time_lag[which.max(abs(regression_coef))]) %>%
                        pull(strongest_prediction),])
}
```

Find lag length that provides most powerful predictions

```
test_lags(vars = c('v2xnp_client', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef      p_value
## 1         1    -0.002171956 4.107440e-01
## 2         2    -0.008182059 3.829922e-02
## 3         3    -0.014921392 2.421120e-03
## 4         4    -0.023807698 7.257002e-05
## 5         5    -0.035905516 2.064600e-06
## 6         6    -0.052321111 1.640576e-08
## 7         7    -0.070977743 2.770100e-11
## 8         8    -0.088682286 1.755841e-13
## 9         9    -0.106521491 1.020662e-15
## 10       10    -0.126081336 3.235842e-18
##    time_lag regression_coef      p_value
## 10       10      -0.1260813 3.235842e-18
```

```
test_lags(vars = c('v2xnp_client', 'v2x_polyarchy', 'e_migdppc', 'e_total_resources_percent'))
```

```
##    time_lag regression_coef      p_value
## 1         1    -0.001265312 6.913361e-01
## 2         2    -0.006716302 1.254640e-01
## 3         3    -0.015314388 4.388301e-03
## 4         4    -0.027275333 4.042069e-05
## 5         5    -0.042365976 6.559206e-07
## 6         6    -0.061631480 6.718832e-09
## 7         7    -0.082861540 8.471123e-12
## 8         8    -0.104219084 1.198987e-14
## 9         9    -0.124593053 3.651384e-17
## 10       10    -0.143293795 2.997080e-19
##    time_lag regression_coef     p_value
## 10       10      -0.1432938 2.99708e-19
```

```
test_lags(vars = c('v2smonex', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef      p_value
## 1         1     0.007059152 0.0270670545
## 2         2     0.016122424 0.0034515176
## 3         3     0.023996604 0.0006723478
## 4         4     0.029887641 0.0004380013
## 5         5     0.037282340 0.0002821480
## 6         6     0.039704934 0.0008019212
## 7         7     0.039660108 0.0036476994
## 8         8     0.039581431 0.0107421165
## 9         9     0.039722827 0.0195131874
## 10       10     0.041885477 0.0239451736
##    time_lag regression_coef    p_value
## 10       10      0.04188548 0.02394517
```

```
test_lags(vars = c('v2smmefra', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef      p_value
## 1         1    -0.006897193 1.030458e-02
## 2         2    -0.014366644 1.970553e-03
## 3         3    -0.025245932 2.330051e-05
## 4         4    -0.035689962 7.697008e-07
## 5         5    -0.045078973 2.783976e-07
## 6         6    -0.054069063 9.809973e-08
## 7         7    -0.063428176 6.489858e-08
## 8         8    -0.072367028 7.959904e-08
## 9         9    -0.076935709 2.444471e-07
## 10       10    -0.080586937 1.030030e-06
##    time_lag regression_coef     p_value
## 10       10     -0.08058694 1.03003e-06
```

```
test_lags(vars = c('v2smgovdom', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef      p_value
## 1         1     -0.01045911 4.025469e-03
## 2         2     -0.02500303 6.679325e-05
## 3         3     -0.03726036 4.118019e-06
## 4         4     -0.04682185 2.124217e-06
## 5         5     -0.05462565 6.951004e-06
## 6         6     -0.06395582 5.941611e-06
## 7         7     -0.08285730 3.431825e-07
## 8         8     -0.10109816 5.442275e-08
## 9         9     -0.10800437 1.408645e-07
## 10       10     -0.11218123 7.648336e-07
##    time_lag regression_coef      p_value
## 10       10      -0.1121812 7.648336e-07
```

```
test_lags(vars = c('v2smpardom', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef      p_value
## 1         1    -0.009224761 7.577419e-03
## 2         2    -0.021022740 4.206994e-04
## 3         3    -0.031800906 3.274207e-05
## 4         4    -0.043393163 2.742730e-06
## 5         5    -0.054963386 1.009209e-06
## 6         6    -0.063106869 1.369906e-06
## 7         7    -0.080639751 9.587390e-08
## 8         8    -0.097383881 2.062874e-08
## 9         9    -0.103248096 7.411622e-08
## 10       10    -0.105982299 6.305982e-07
##    time_lag regression_coef      p_value
## 10       10      -0.1059823 6.305982e-07
```

```
test_lags(vars = c('v2smfordom', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef   p_value
## 1         1   -0.0006079735 0.7972186
## 2         2   -0.0013990435 0.7322687
## 3         3   -0.0026465003 0.6173097
## 4         4   -0.0044035915 0.4939897
## 5         5   -0.0060865521 0.4389872
## 6         6   -0.0094220597 0.3030629
## 7         7   -0.0132949406 0.2117689
## 8         8   -0.0176455307 0.1508007
## 9         9   -0.0194106026 0.1534970
## 10       10   -0.0195598164 0.1926372
## [1] time_lag        regression_coef p_value
## <0 rows> (or 0-length row.names)
```

```
test_lags(vars = c('v2cacamps', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef      p_value
## 1         1     -0.00778844 1.568723e-05
## 2         2     -0.01574449 5.046599e-09
## 3         3     -0.02319069 5.488117e-12
## 4         4     -0.03072676 9.347133e-14
## 5         5     -0.03947713 3.559478e-14
## 6         6     -0.04722405 1.759313e-13
## 7         7     -0.05341070 5.944604e-13
## 8         8     -0.05896493 2.094447e-12
## 9         9     -0.06162551 3.350261e-11
## 10       10     -0.06398357 3.488101e-10
##    time_lag regression_coef      p_value
## 10       10     -0.06398357 3.488101e-10
```

```
test_lags(vars = c('v2smpolsoc', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef      p_value
## 1         1     -0.01024134 1.344282e-04
## 2         2     -0.01994241 1.654886e-05
## 3         3     -0.03627168 1.033385e-09
## 4         4     -0.05411840 4.674330e-14
## 5         5     -0.07039449 7.864749e-16
## 6         6     -0.08253868 4.702087e-16
## 7         7     -0.09491823 1.036418e-15
## 8         8     -0.10759172 3.414394e-15
## 9         9     -0.11347050 1.172968e-13
## 10       10     -0.11817911 3.942099e-12
##    time_lag regression_coef      p_value
## 10       10      -0.1181791 3.942099e-12
```

```
test_lags(vars = c('v2smpolhate', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef      p_value
## 1         1    -0.007572887 3.934801e-03
## 2         2    -0.019678393 1.366778e-05
## 3         3    -0.030176854 1.936795e-07
## 4         4    -0.041480196 3.029438e-09
## 5         5    -0.054538977 1.487382e-10
## 6         6    -0.063063624 2.017152e-10
## 7         7    -0.074128293 1.161203e-10
## 8         8    -0.085742936 1.052161e-10
## 9         9    -0.090595633 8.500912e-10
## 10       10    -0.092581466 1.729489e-08
##    time_lag regression_coef      p_value
## 10       10     -0.09258147 1.729489e-08
```

Find optimal lag for models with interacted variables

```
test_lags(vars = c('smmefraXsmpardom', 'v2smmefra', 'v2smpardom', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef   p_value
## 1         1    -0.001536971 0.9367958
## 2         2     0.001006722 0.9759252
## 3         3    -0.010519447 0.8042706
## 4         4    -0.019583323 0.7000894
## 5         5    -0.018550816 0.7627030
## 6         6    -0.030985053 0.6618731
## 7         7    -0.064138462 0.4293523
## 8         8    -0.076886270 0.4052607
## 9         9     0.003467467 0.9728909
## 10       10     0.121424344 0.2839144
## [1] time_lag        regression_coef p_value
## <0 rows> (or 0-length row.names)
```

```
test_lags(vars = c('smonexXsmmefra', 'v2smonex', 'v2smmefra', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef      p_value
## 1         1      0.02557159 1.367775e-01
## 2         2      0.05515466 6.204924e-02
## 3         3      0.09166869 1.478932e-02
## 4         4      0.11801827 9.010680e-03
## 5         5      0.16915111 1.932078e-03
## 6         6      0.23250697 2.119309e-04
## 7         7      0.29532616 4.420301e-05
## 8         8      0.34817164 2.558216e-05
## 9         9      0.38443607 2.848675e-05
## 10       10      0.42340991 2.934350e-05
##    time_lag regression_coef     p_value
## 10       10       0.4234099 2.93435e-05
```

```
test_lags(vars = c('smonexXsmfordom', 'v2smonex', 'v2smfordom', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef      p_value
## 1         1      0.03746043 1.067961e-02
## 2         2      0.06927505 6.136609e-03
## 3         3      0.11569505 4.069194e-04
## 4         4      0.15494846 1.012951e-04
## 5         5      0.20147817 3.835995e-05
## 6         6      0.24640460 1.738880e-05
## 7         7      0.28291906 2.688074e-05
## 8         8      0.31121178 8.281177e-05
## 9         9      0.30213931 7.493284e-04
## 10       10      0.27246808 6.640054e-03
##   time_lag regression_coef      p_value
## 8        8       0.3112118 8.281177e-05
```

```
test_lags(vars = c('smmefraXsmfordom', 'v2smmefra', 'v2smfordom', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef   p_value
## 1         1     0.005522687 0.6945666
## 2         2     0.003445445 0.8870753
## 3         3     0.009086858 0.7735736
## 4         4     0.017750227 0.6445086
## 5         5     0.030548776 0.5160568
## 6         6     0.024577265 0.6543070
## 7         7     0.019909220 0.7557193
## 8         8     0.020629094 0.7810523
## 9         9     0.051893281 0.5289065
## 10       10     0.094009078 0.3022285
## [1] time_lag        regression_coef p_value
## <0 rows> (or 0-length row.names)
```

```
test_lags(vars = c('clientXsmpolsoc', 'v2xnp_client', 'v2smpolsoc', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef      p_value
## 1         1     -0.04246675 2.198426e-02
## 2         2     -0.07745627 1.518765e-02
## 3         3     -0.16857473 2.981983e-05
## 4         4     -0.23831143 7.089545e-07
## 5         5     -0.30382628 2.282901e-07
## 6         6     -0.42974632 2.106248e-10
## 7         7     -0.57428649 1.977604e-13
## 8         8     -0.72824357 7.552510e-16
## 9         9     -0.78961468 6.594667e-14
## 10       10     -0.79962795 7.180087e-11
##    time_lag regression_coef      p_value
## 10       10       -0.799628 7.180087e-11
```

```
test_lags(vars = c('smmefraXsmpolsoc', 'v2smmefra', 'v2smpolsoc', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef      p_value
## 1         1     -0.01675575 2.628387e-01
## 2         2     -0.02598889 3.137141e-01
## 3         3     -0.07214884 2.944267e-02
## 4         4     -0.12957741 1.112671e-03
## 5         5     -0.18258162 1.483805e-04
## 6         6     -0.24629695 9.651515e-06
## 7         7     -0.32748160 4.085780e-07
## 8         8     -0.40529647 5.477176e-08
## 9         9     -0.42386972 6.388408e-07
## 10       10     -0.41896191 1.754477e-05
##   time_lag regression_coef      p_value
## 9        9      -0.4238697 6.388408e-07
```

```
test_lags(vars = c('clientXresources', 'v2xnp_client', 'e_total_resources_percent', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef      p_value
## 1         1   -4.401133e-05 6.319324e-01
## 2         2   -2.019045e-04 1.098435e-01
## 3         3   -6.501267e-04 2.517579e-05
## 4         4   -1.218141e-03 1.339120e-10
## 5         5   -1.790080e-03 1.473238e-13
## 6         6   -2.291298e-03 3.089945e-14
## 7         7   -2.636616e-03 1.633529e-14
## 8         8   -2.881960e-03 4.319227e-14
## 9         9   -3.047738e-03 2.838959e-13
## 10       10   -3.155436e-03 2.466773e-12
##    time_lag regression_coef      p_value
## 10       10    -0.003155436 2.466773e-12
```

```
test_lags(vars = c('clientXcacamps', 'v2xnp_client', 'v2cacamps', 'v2x_polyarchy', 'e_migdppc'))
```

```
##    time_lag regression_coef    p_value
## 1         1     -0.01032098 0.46128418
## 2         2     -0.03917919 0.06045946
## 3         3     -0.06142165 0.01873535
## 4         4     -0.06132791 0.05510263
## 5         5     -0.04239796 0.29868553
## 6         6     -0.08688511 0.08436272
## 7         7     -0.13964929 0.01696154
## 8         8     -0.15590050 0.01901577
## 9         9     -0.12279398 0.09812380
## 10       10     -0.08661776 0.29229080
##   time_lag regression_coef    p_value
## 8        8      -0.1559005 0.01901577
```

Run models at best time lag of 10 years, and export results

Record change in polyarchy scores over 10 years

```
vdem <- vdem %>%
  group_by(country_name) %>%
  arrange(year) %>%
  mutate(v2x_polyarchy_lagged = lead(v2x_polyarchy, n = 10)) %>%
  ungroup()
vdem$polyarchy_change <- vdem$v2x_polyarchy_lagged - vdem$v2x_polyarchy
```

```
cm_print <- lm(polyarchy_change ~ v2xnp_client + v2x_polyarchy + e_migdppc + as.factor(year),
               data = (vdem %>% filter(consolidated_lhb == TRUE)))
summary(cm_print)
```

```
## 
## Call:
## lm(formula = polyarchy_change ~ v2xnp_client + v2x_polyarchy + 
##     e_migdppc + as.factor(year), data = (vdem %>% filter(consolidated_lhb == 
##     TRUE)))
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.65650 -0.01036  0.00248  0.01821  0.18067 
## 
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)    
## (Intercept)          0.2639423  0.0611067   4.319 1.65e-05 ***
## v2xnp_client        -0.1260813  0.0143300  -8.798  < 2e-16 ***
## v2x_polyarchy       -0.3019518  0.0302795  -9.972  < 2e-16 ***
## e_migdppc            0.0001535  0.0001819   0.844  0.39903    
## as.factor(year)1901 -0.0059621  0.0789907  -0.075  0.93984    
## as.factor(year)1902 -0.0174066  0.0684083  -0.254  0.79918    
## as.factor(year)1903 -0.0172970  0.0684081  -0.253  0.80041    
## as.factor(year)1904 -0.0328960  0.0684076  -0.481  0.63066    
## as.factor(year)1905 -0.0454286  0.0684076  -0.664  0.50672    
## as.factor(year)1906 -0.0524600  0.0684078  -0.767  0.44326    
## as.factor(year)1907 -0.0625470  0.0684078  -0.914  0.36067    
## as.factor(year)1908 -0.0559947  0.0684077  -0.819  0.41316    
## as.factor(year)1909 -0.0344340  0.0684077  -0.503  0.61477    
## as.factor(year)1910 -0.0001825  0.0684082  -0.003  0.99787    
## as.factor(year)1911  0.0006799  0.0684085   0.010  0.99207    
## as.factor(year)1912  0.0052352  0.0684095   0.077  0.93901    
## as.factor(year)1913  0.0070370  0.0684093   0.103  0.91808    
## as.factor(year)1914 -0.0125133  0.0645479  -0.194  0.84631    
## as.factor(year)1915  0.0275837  0.0790104   0.349  0.72704    
## as.factor(year)1916  0.0231734  0.0684093   0.339  0.73484    
## as.factor(year)1917  0.0080528  0.0790129   0.102  0.91883    
## as.factor(year)1918  0.0211587  0.0684099   0.309  0.75713    
## as.factor(year)1919  0.0095859  0.0644962   0.149  0.88186    
## as.factor(year)1920 -0.0224239  0.0624546  -0.359  0.71961    
## as.factor(year)1921 -0.0220214  0.0624551  -0.353  0.72443    
## as.factor(year)1922 -0.0248844  0.0624546  -0.398  0.69035    
## as.factor(year)1923 -0.0273451  0.0624540  -0.438  0.66155    
## as.factor(year)1924 -0.0271396  0.0624540  -0.435  0.66394    
## as.factor(year)1925 -0.0321845  0.0612006  -0.526  0.59903    
## as.factor(year)1926 -0.0364021  0.0612003  -0.595  0.55205    
## as.factor(year)1927 -0.0306547  0.0624541  -0.491  0.62360    
## as.factor(year)1928 -0.0303072  0.0624543  -0.485  0.62755    
## as.factor(year)1929 -0.0417779  0.0624549  -0.669  0.50363    
## as.factor(year)1930 -0.0953741  0.0624546  -1.527  0.12692    
## as.factor(year)1931 -0.1070457  0.0624546  -1.714  0.08671 .  
## as.factor(year)1932 -0.0998926  0.0624555  -1.599  0.10991    
## as.factor(year)1933 -0.1387251  0.0624560  -2.221  0.02647 *  
## as.factor(year)1934 -0.2050065  0.0624561  -3.282  0.00105 ** 
## as.factor(year)1935 -0.1730113  0.0624573  -2.770  0.00566 ** 
## as.factor(year)1936 -0.0126991  0.0624594  -0.203  0.83891    
## as.factor(year)1937 -0.0075131  0.0624596  -0.120  0.90427    
## as.factor(year)1938 -0.0078020  0.0624597  -0.125  0.90061    
## as.factor(year)1939  0.0005037  0.0624675   0.008  0.99357    
## as.factor(year)1940 -0.0081037  0.0645638  -0.126  0.90013    
## as.factor(year)1941  0.0028204  0.0684213   0.041  0.96712    
## as.factor(year)1942 -0.0006174  0.0684162  -0.009  0.99280    
## as.factor(year)1943  0.0268410  0.0645198   0.416  0.67745    
## as.factor(year)1944  0.0082025  0.0624906   0.131  0.89558    
## as.factor(year)1945  0.0117292  0.0624879   0.188  0.85113    
## as.factor(year)1946  0.0087253  0.0588822   0.148  0.88222    
## as.factor(year)1947  0.0081149  0.0583429   0.139  0.88940    
## as.factor(year)1948  0.0081436  0.0583428   0.140  0.88901    
## as.factor(year)1949  0.0022039  0.0579657   0.038  0.96967    
## as.factor(year)1950  0.0048810  0.0574786   0.085  0.93234    
## as.factor(year)1951  0.0093524  0.0575778   0.162  0.87099
## as.factor(year)1952  0.0103439  0.0575784   0.180  0.85745
## as.factor(year)1953  0.0111945  0.0575791   0.194  0.84587
## as.factor(year)1954  0.0103302  0.0575806   0.179  0.85764
## as.factor(year)1955 -0.0020603  0.0574818  -0.036  0.97141
## as.factor(year)1956  0.0068344  0.0574833   0.119  0.90537
## as.factor(year)1957  0.0087817  0.0574845   0.153  0.87860
## as.factor(year)1958 -0.0009517  0.0574846  -0.017  0.98679
## as.factor(year)1959 -0.0012220  0.0574845  -0.021  0.98304
## as.factor(year)1960 -0.0005821  0.0574869  -0.010  0.99192
## as.factor(year)1961  0.0062971  0.0573220   0.110  0.91254
## as.factor(year)1962  0.0141880  0.0573229   0.248  0.80454
## as.factor(year)1963 -0.0043549  0.0573252  -0.076  0.93945
## as.factor(year)1964 -0.0065562  0.0573273  -0.114  0.90896
## as.factor(year)1965 -0.0094230  0.0574108  -0.164  0.86965
## as.factor(year)1966 -0.0072767  0.0573318  -0.127  0.89902
## as.factor(year)1967 -0.0100061  0.0573327  -0.175  0.86147
## as.factor(year)1968  0.0275013  0.0574147   0.479  0.63200
## as.factor(year)1969  0.0265690  0.0574200   0.463  0.64363
## as.factor(year)1970  0.0252193  0.0573476   0.440  0.66016
## as.factor(year)1971  0.0257530  0.0573545   0.449  0.65348
## as.factor(year)1972  0.0192824  0.0572966   0.337  0.73651
## as.factor(year)1973  0.0163556  0.0573775   0.285  0.77564
## as.factor(year)1974  0.0185640  0.0573079   0.324  0.74603
## as.factor(year)1975  0.0209578  0.0572416   0.366  0.71431
## as.factor(year)1976  0.0222176  0.0571896   0.388  0.69770
## as.factor(year)1977  0.0210486  0.0571944   0.368  0.71290
## as.factor(year)1978  0.0208055  0.0571999   0.364  0.71610
## as.factor(year)1979  0.0223514  0.0571489   0.391  0.69576
## as.factor(year)1980  0.0241574  0.0571015   0.423  0.67230
## as.factor(year)1981  0.0222005  0.0570542   0.389  0.69724
## as.factor(year)1982  0.0205296  0.0570562   0.360  0.71903
## as.factor(year)1983  0.0196951  0.0570600   0.345  0.73001
## as.factor(year)1984  0.0198270  0.0569819   0.348  0.72792
## as.factor(year)1985  0.0198820  0.0570312   0.349  0.72742
## as.factor(year)1986  0.0199839  0.0569511   0.351  0.72571
## as.factor(year)1987  0.0187763  0.0569576   0.330  0.74170
## as.factor(year)1988  0.0179711  0.0569637   0.315  0.75243
## as.factor(year)1989  0.0178781  0.0569698   0.314  0.75370
## as.factor(year)1990  0.0208257  0.0569117   0.366  0.71446
## as.factor(year)1991  0.0171322  0.0567669   0.302  0.76284
## as.factor(year)1992  0.0104047  0.0567083   0.183  0.85444
## as.factor(year)1993  0.0097368  0.0566920   0.172  0.86365
## as.factor(year)1994  0.0076478  0.0566811   0.135  0.89269
## as.factor(year)1995  0.0093289  0.0566896   0.165  0.86931
## as.factor(year)1996  0.0122490  0.0566825   0.216  0.82894
## as.factor(year)1997  0.0124957  0.0566932   0.220  0.82558
## as.factor(year)1998  0.0198625  0.0567167   0.350  0.72623
## as.factor(year)1999  0.0204035  0.0567035   0.360  0.71902
## as.factor(year)2000  0.0189015  0.0566816   0.333  0.73882
## as.factor(year)2001  0.0193698  0.0566725   0.342  0.73255
## as.factor(year)2002  0.0178411  0.0566797   0.315  0.75297
## as.factor(year)2003  0.0108019  0.0566840   0.191  0.84889
## as.factor(year)2004  0.0042769  0.0566902   0.075  0.93987
## as.factor(year)2005  0.0045042  0.0567063   0.079  0.93670
## as.factor(year)2006 -0.0014942  0.0567096  -0.026  0.97898
## as.factor(year)2007 -0.0054809  0.0567493  -0.097  0.92307
## as.factor(year)2008 -0.0096058  0.0567401  -0.169  0.86558
## as.factor(year)2009 -0.0181736  0.0567034  -0.321  0.74863
## as.factor(year)2010 -0.0275656  0.0567158  -0.486  0.62701
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05585 on 1764 degrees of freedom
##   (490 observations deleted due to missingness)
## Multiple R-squared:  0.191,  Adjusted R-squared:  0.1392
## F-statistic: 3.687 on 113 and 1764 DF,  p-value: < 2.2e-16
```

```
pm_print <- lm(polyarchy_change ~ v2cacamps + v2x_polyarchy + e_migdppc + as.factor(year),
               data = (vdem %>% filter(consolidated_lhb == TRUE)))
summary(pm_print)
```

```
##
## Call:
## lm(formula = polyarchy_change ~ v2cacamps + v2x_polyarchy + e_migdppc +
##     as.factor(year), data = (vdem %>% filter(consolidated_lhb ==
##     TRUE)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66175 -0.01102  0.00262  0.01936  0.17529
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.2201316  0.0623379   3.531 0.000425 ***
## v2cacamps           -0.0639836  0.0101356  -6.313 3.49e-10 ***
## v2x_polyarchy       -0.2557492  0.0301529  -8.482  < 2e-16 ***
## e_migdppc            0.0007871  0.0001628   4.834 1.46e-06 ***
## as.factor(year)1901 -0.0063481  0.0811548  -0.078 0.937661
## as.factor(year)1902 -0.0122335  0.0702850  -0.174 0.861843
## as.factor(year)1903 -0.0123178  0.0702852  -0.175 0.860901
## as.factor(year)1904 -0.0279501  0.0702850  -0.398 0.690924
## as.factor(year)1905 -0.0406174  0.0702851  -0.578 0.563412
## as.factor(year)1906 -0.0475798  0.0702854  -0.677 0.498529
## as.factor(year)1907 -0.0573788  0.0702856  -0.816 0.414405
## as.factor(year)1908 -0.0506104  0.0702853  -0.720 0.471580
## as.factor(year)1909 -0.0291222  0.0702854  -0.414 0.678676
## as.factor(year)1910  0.0046241  0.0702860   0.066 0.947553
## as.factor(year)1911  0.0054399  0.0702862   0.077 0.938318
## as.factor(year)1912  0.0112158  0.0702865   0.160 0.873236
## as.factor(year)1913  0.0129704  0.0702863   0.185 0.853614
## as.factor(year)1914  0.0183769  0.0702861   0.261 0.793771
## as.factor(year)1915  0.0327327  0.0811731   0.403 0.686818
## as.factor(year)1916  0.0331676  0.0702875   0.472 0.637070
## as.factor(year)1917  0.0225025  0.0811980   0.277 0.781713
## as.factor(year)1918  0.0317390  0.0702873   0.452 0.651642
## as.factor(year)1919  0.0152618  0.0662641   0.230 0.817873
## as.factor(year)1920 -0.0109910  0.0641703  -0.171 0.864025
## as.factor(year)1921 -0.0099576  0.0641703  -0.155 0.876702
## as.factor(year)1922 -0.0138371  0.0641687  -0.216 0.829297
## as.factor(year)1923 -0.0165580  0.0641689  -0.258 0.796409
## as.factor(year)1924 -0.0165361  0.0641692  -0.258 0.796674
## as.factor(year)1925 -0.0186876  0.0628809  -0.297 0.766358
## as.factor(year)1926 -0.0231867  0.0628810  -0.369 0.712368
## as.factor(year)1927 -0.0204375  0.0641697  -0.318 0.750151
## as.factor(year)1928 -0.0202344  0.0641702  -0.315 0.752554
## as.factor(year)1929 -0.0315473  0.0641707  -0.492 0.623055
## as.factor(year)1930 -0.0831425  0.0641751  -1.296 0.195304
## as.factor(year)1931 -0.0946521  0.0641746  -1.475 0.140420
## as.factor(year)1932 -0.0873777  0.0641750  -1.362 0.173519
## as.factor(year)1933 -0.1269553  0.0641744  -1.978 0.048058 *
## as.factor(year)1934 -0.1931024  0.0641755  -3.009 0.002660 **
## as.factor(year)1935 -0.1609911  0.0641764  -2.509 0.012214 *
## as.factor(year)1936 -0.0001497  0.0641780  -0.002 0.998139
## as.factor(year)1937  0.0047716  0.0641788   0.074 0.940742
## as.factor(year)1938  0.0044840  0.0641790   0.070 0.944308
## as.factor(year)1939  0.0119723  0.0641813   0.187 0.852045
## as.factor(year)1940 -0.0006834  0.0663249  -0.010 0.991780
## as.factor(year)1941  0.0066631  0.0702948   0.095 0.924494
## as.factor(year)1942  0.0018967  0.0702909   0.027 0.978476
## as.factor(year)1943  0.0176327  0.0662747   0.266 0.790229
## as.factor(year)1944  0.0049258  0.0642017   0.077 0.938852
## as.factor(year)1945  0.0084271  0.0641987   0.131 0.895581
## as.factor(year)1946  0.0141242  0.0605102   0.233 0.815464
## as.factor(year)1947  0.0176562  0.0599661   0.294 0.768460
## as.factor(year)1948  0.0176144  0.0599663   0.294 0.768994
## as.factor(year)1949  0.0132345  0.0595850   0.222 0.824253
## as.factor(year)1950  0.0108537  0.0591738   0.183 0.854489
## as.factor(year)1951  0.0163325  0.0592916   0.275 0.782996
## as.factor(year)1952  0.0164182  0.0592888   0.277 0.781876
## as.factor(year)1953  0.0166860  0.0592880   0.281 0.778408
## as.factor(year)1954  0.0127590  0.0592836   0.215 0.829622
## as.factor(year)1955  0.0005807  0.0591728   0.010 0.992171
## as.factor(year)1956  0.0095169  0.0591740   0.161 0.872247
## as.factor(year)1957  0.0097024  0.0591724   0.164 0.869776
## as.factor(year)1958 -0.0003738  0.0591728  -0.006 0.994960
## as.factor(year)1959  0.0002006  0.0591747   0.003 0.997296
## as.factor(year)1960  0.0009714  0.0591802   0.016 0.986906
## as.factor(year)1961  0.0063154  0.0589855   0.107 0.914749
## as.factor(year)1962  0.0146566  0.0589863   0.248 0.803797
## as.factor(year)1963 -0.0051907  0.0589881  -0.088 0.929890
## as.factor(year)1964 -0.0078755  0.0589886  -0.134 0.893806
## as.factor(year)1965 -0.0128456  0.0590786  -0.217 0.827897
## as.factor(year)1966 -0.0086562  0.0589948  -0.147 0.883365
## as.factor(year)1967 -0.0100143  0.0589998  -0.170 0.865239
## as.factor(year)1968  0.0302479  0.0590966   0.512 0.608830
## as.factor(year)1969  0.0288465  0.0591027   0.488 0.625561
## as.factor(year)1970  0.0258404  0.0590166   0.438 0.661551
## as.factor(year)1971  0.0251876  0.0590199   0.427 0.669605
## as.factor(year)1972  0.0172749  0.0589524   0.293 0.769535
## as.factor(year)1973  0.0133505  0.0590408   0.226 0.821132
## as.factor(year)1974  0.0157569  0.0589630   0.267 0.789322
## as.factor(year)1975  0.0175207  0.0588838   0.298 0.766085
## as.factor(year)1976  0.0179721  0.0588236   0.306 0.760004
## as.factor(year)1977  0.0165919  0.0588284   0.282 0.777947
## as.factor(year)1978  0.0156136  0.0588300   0.265 0.790732
## as.factor(year)1979  0.0178163  0.0587765   0.303 0.761836
## as.factor(year)1980  0.0185391  0.0587144   0.316 0.752231
## as.factor(year)1981  0.0180863  0.0586690   0.308 0.757910
## as.factor(year)1982  0.0160928  0.0586697   0.274 0.783891
## as.factor(year)1983  0.0149579  0.0586725   0.255 0.798801
## as.factor(year)1984  0.0140693  0.0585805   0.240 0.810228
## as.factor(year)1985  0.0130771  0.0586288   0.223 0.823524
## as.factor(year)1986  0.0144151  0.0585466   0.246 0.805545
## as.factor(year)1987  0.0126742  0.0585495   0.216 0.828648
## as.factor(year)1988  0.0115358  0.0585548   0.197 0.843844
## as.factor(year)1989  0.0114115  0.0585610   0.195 0.845522
## as.factor(year)1990  0.0125271  0.0584856   0.214 0.830424
## as.factor(year)1991  0.0102553  0.0583376   0.176 0.860478
## as.factor(year)1992  0.0023270  0.0582694   0.040 0.968149
## as.factor(year)1993  0.0006398  0.0582455   0.011 0.991238
## as.factor(year)1994 -0.0020627  0.0582302  -0.035 0.971746
## as.factor(year)1995 -0.0007778  0.0582364  -0.013 0.989345
## as.factor(year)1996  0.0008599  0.0582190   0.015 0.988218
## as.factor(year)1997  0.0004121  0.0582267   0.007 0.994353
## as.factor(year)1998  0.0086795  0.0582580   0.149 0.881584
## as.factor(year)1999  0.0093704  0.0582444   0.161 0.872207
## as.factor(year)2000  0.0074601  0.0582178   0.128 0.898052
## as.factor(year)2001  0.0071921  0.0582026   0.124 0.901670
## as.factor(year)2002  0.0054700  0.0582081   0.094 0.925142
## as.factor(year)2003 -0.0010390  0.0582148  -0.018 0.985763
## as.factor(year)2004 -0.0072362  0.0582241  -0.124 0.901107
## as.factor(year)2005 -0.0079632  0.0582346  -0.137 0.891250
## as.factor(year)2006 -0.0151959  0.0582287  -0.261 0.794148
## as.factor(year)2007 -0.0205118  0.0582608  -0.352 0.724831
## as.factor(year)2008 -0.0255707  0.0582439  -0.439 0.660697
## as.factor(year)2009 -0.0328608  0.0582142  -0.564 0.572501
## as.factor(year)2010 -0.0422561  0.0582300  -0.726 0.468137
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05738 on 1702 degrees of freedom
##   (552 observations deleted due to missingness)
## Multiple R-squared:  0.1748, Adjusted R-squared:   0.12
## F-statistic:  3.19 on 113 and 1702 DF,  p-value: < 2.2e-16
```

```
mm_print <- lm(polyarchy_change ~ (v2smonex * v2smmefra) + v2x_polyarchy + e_migdppc + as.factor(year),
               data = (vdem %>% filter(consolidated_lhb == TRUE)))
summary(mm_print)
```

```
##
## Call:
## lm(formula = polyarchy_change ~ (v2smonex * v2smmefra) + v2x_polyarchy +
##     e_migdppc + as.factor(year), data = (vdem %>% filter(consolidated_lhb ==
##     TRUE)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.33372 -0.01525  0.00725  0.02704  0.12855
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.3771129  0.0591672   6.374 4.39e-10 ***
## v2smonex            -0.1435644  0.0489123  -2.935 0.003496 **
## v2smmefra           -0.3734141  0.0704576  -5.300 1.78e-07 ***
## v2x_polyarchy       -0.2980822  0.0593615  -5.021 7.28e-07 ***
## e_migdppc            0.0008964  0.0001947   4.603 5.35e-06 ***
## as.factor(year)2001 -0.0007395  0.0101378  -0.073 0.941880
## as.factor(year)2002 -0.0027655  0.0101383  -0.273 0.785146
## as.factor(year)2003 -0.0097682  0.0101396  -0.963 0.335851
## as.factor(year)2004 -0.0150477  0.0101461  -1.483 0.138714
## as.factor(year)2005 -0.0170500  0.0101561  -1.679 0.093852 .
## as.factor(year)2006 -0.0249224  0.0101127  -2.464 0.014077 *
## as.factor(year)2007 -0.0306594  0.0101839  -3.011 0.002747 **
## as.factor(year)2008 -0.0357741  0.0101387  -3.528 0.000459 ***
## as.factor(year)2009 -0.0430098  0.0100800  -4.267 2.40e-05 ***
## as.factor(year)2010 -0.0537532  0.0101570  -5.292 1.85e-07 ***
## v2smonex:v2smmefra   0.4234099  0.1003430   4.220 2.93e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04727 on 473 degrees of freedom
##   (1879 observations deleted due to missingness)
## Multiple R-squared:  0.232,  Adjusted R-squared:  0.2076
## F-statistic: 9.525 on 15 and 473 DF,  p-value: < 2.2e-16
```

```
dm_print <- lm(polyarchy_change ~ v2smpardom + v2x_polyarchy + e_migdppc + as.factor(year),
               data = (vdem %>% filter(consolidated_lhb == TRUE)))
summary(dm_print)
```

```
## 
## Call:
## lm(formula = polyarchy_change ~ v2smpardom + v2x_polyarchy + 
##     e_migdppc + as.factor(year), data = (vdem %>% filter(consolidated_lhb == 
##     TRUE)))
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.34350 -0.01073  0.00846  0.02467  0.08753 
## 
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)    
## (Intercept)          0.3798724  0.0619857   6.128 1.87e-09 ***
## v2smpardom          -0.1059823  0.0209862  -5.050 6.31e-07 ***
## v2x_polyarchy       -0.4509116  0.0716063  -6.297 6.91e-10 ***
## e_migdppc            0.0013649  0.0001907   7.159 3.12e-12 ***
## as.factor(year)2001 -0.0008341  0.0103738  -0.080 0.935950    
## as.factor(year)2002 -0.0025880  0.0103742  -0.249 0.803105    
## as.factor(year)2003 -0.0093627  0.0103754  -0.902 0.367307    
## as.factor(year)2004 -0.0165216  0.0103819  -1.591 0.112187    
## as.factor(year)2005 -0.0167516  0.0103875  -1.613 0.107482    
## as.factor(year)2006 -0.0240548  0.0103402  -2.326 0.020420 *  
## as.factor(year)2007 -0.0294464  0.0104104  -2.829 0.004874 ** 
## as.factor(year)2008 -0.0335229  0.0103585  -3.236 0.001296 ** 
## as.factor(year)2009 -0.0395899  0.0102898  -3.847 0.000136 ***
## as.factor(year)2010 -0.0463477  0.0103436  -4.481 9.33e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.04837 on 475 degrees of freedom
##   (1879 observations deleted due to missingness)
## Multiple R-squared:  0.1924, Adjusted R-squared:  0.1703 
## F-statistic: 8.706 on 13 and 475 DF,  p-value: 5.036e-16
```

```
fullm_print <- lm(polyarchy_change ~ v2xnp_client + v2cacamps + (v2smonex * v2smmefra) + v2smpardom + v2x_polyarchy + e_migd
ppc + as.factor(year),
                  data = (vdem %>% filter(consolidated_lhb == TRUE)))
summary(fullm_print)
```

```
## 
## Call:
## lm(formula = polyarchy_change ~ v2xnp_client + v2cacamps + (v2smonex * 
##     v2smmefra) + v2smpardom + v2x_polyarchy + e_migdppc + as.factor(year), 
##     data = (vdem %>% filter(consolidated_lhb == TRUE)))
## 
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.312842 -0.018080  0.006805  0.027267  0.123852 
## 
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)    
## (Intercept)          0.6559590  0.0819691   8.003 1.01e-14 ***
## v2xnp_client        -0.0782605  0.0261500  -2.993 0.002914 ** 
## v2cacamps           -0.0540624  0.0169010  -3.199 0.001476 ** 
## v2smonex            -0.1001227  0.0537775  -1.862 0.063271 .  
## v2smmefra           -0.2657501  0.0762963  -3.483 0.000543 ***
## v2smpardom          -0.0686580  0.0264894  -2.592 0.009849 ** 
## v2x_polyarchy       -0.6231303  0.0874250  -7.128 3.99e-12 ***
## e_migdppc            0.0006972  0.0002284   3.052 0.002402 ** 
## as.factor(year)2001 -0.0005901  0.0099929  -0.059 0.952935    
## as.factor(year)2002 -0.0020251  0.0099943  -0.203 0.839517    
## as.factor(year)2003 -0.0086445  0.0099954  -0.865 0.387575    
## as.factor(year)2004 -0.0149662  0.0100021  -1.496 0.135263    
## as.factor(year)2005 -0.0161523  0.0100114  -1.613 0.107349    
## as.factor(year)2006 -0.0229539  0.0099759  -2.301 0.021843 *  
## as.factor(year)2007 -0.0274735  0.0100710  -2.728 0.006617 ** 
## as.factor(year)2008 -0.0326377  0.0100265  -3.255 0.001217 ** 
## as.factor(year)2009 -0.0394360  0.0099598  -3.960 8.70e-05 ***
## as.factor(year)2010 -0.0476007  0.0100889  -4.718 3.16e-06 ***
## v2smonex:v2smmefra   0.3306004  0.1089605   3.034 0.002549 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.04605 on 459 degrees of freedom
##   (1890 observations deleted due to missingness)
## Multiple R-squared:  0.2915, Adjusted R-squared:  0.2637 
## F-statistic: 10.49 on 18 and 459 DF,  p-value: < 2.2e-16
```

```
threem_print <- lm(polyarchy_change ~ v2xnp_client + v2cacamps + v2smpardom + v2x_polyarchy + e_migdppc + as.factor(year),
                         data = (vdem %>% filter(consolidated_lhb == TRUE)))
summary(threem_print)
```

```
##
## Call:
## lm(formula = polyarchy_change ~ v2xnp_client + v2cacamps + v2smpardom +
##     v2x_polyarchy + e_migdppc + as.factor(year), data = (vdem %>%
##     filter(consolidated_lhb == TRUE)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.31958 -0.01657  0.00469  0.02727  0.09864
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.5994535  0.0797838   7.513 3.01e-13 ***
## v2xnp_client        -0.0905121  0.0261841  -3.457 0.000597 ***
## v2cacamps           -0.0804011  0.0161435  -4.980 8.98e-07 ***
## v2smpardom          -0.0619082  0.0256106  -2.417 0.016023 *
## v2x_polyarchy       -0.6516805  0.0876776  -7.433 5.20e-13 ***
## e_migdppc            0.0007638  0.0002188   3.491 0.000527 ***
## as.factor(year)2001 -0.0003829  0.0101902  -0.038 0.970039
## as.factor(year)2002 -0.0017205  0.0101914  -0.169 0.866010
## as.factor(year)2003 -0.0082111  0.0101924  -0.806 0.420881
## as.factor(year)2004 -0.0153181  0.0101990  -1.502 0.133802
## as.factor(year)2005 -0.0150119  0.0102058  -1.471 0.141993
## as.factor(year)2006 -0.0213745  0.0101660  -2.103 0.036045 *
## as.factor(year)2007 -0.0253107  0.0102575  -2.468 0.013967 *
## as.factor(year)2008 -0.0301263  0.0102075  -2.951 0.003324 **
## as.factor(year)2009 -0.0369153  0.0101313  -3.644 0.000299 ***
## as.factor(year)2010 -0.0436345  0.0102086  -4.274 2.33e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04696 on 462 degrees of freedom
##   (1890 observations deleted due to missingness)
## Multiple R-squared:  0.2584, Adjusted R-squared:  0.2343
## F-statistic: 10.73 on 15 and 462 DF,  p-value: < 2.2e-16
```

Save out model results

```
stargazer(cm_print, pm_print, mm_print, dm_print, fullm_print,
          title = 'Solvents of Democracy',
          keep = c('polyarchy_change', 'v2xnp_client', 'v2cacamps', 'v2smonex', 'v2smmefra', 'v2smpardom',
                   'v2x_polyarchy', 'e_migdppc'),
          dep.var.labels = 'Change in Polyarchy Score After 10 Years',
          model.names = TRUE,
          covariate.labels = c('Clientelism', 'Polarization', 'Online Media Consumption', 'Online Media Fractionalization',
'Party Disinformation',
                               'Level of Democracy', 'GDP Per Capita', 'Online Media Consumption X Fractionalization'),
          nobs = TRUE,
          type = 'html',
          out = './models/democracy_erosion_model_results.doc')
```

```
##
## <table style="text-align:center"><caption><strong>Solvents of Democracy</strong></caption>
## <tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"></td><td colspan
="5"><em>Dependent variable:</em></td></tr>
## <tr><td></td><td colspan="5" style="border-bottom: 1px solid black"></td></tr>
## <tr><td style="text-align:left"></td><td colspan="5">Change in Polyarchy Score After 10 Years</td></tr>
## <tr><td style="text-align:left"></td><td colspan="5"><em>OLS</em></td></tr>
## <tr><td style="text-align:left"></td><td>(1)</td><td>(2)</td><td>(3)</td><td>(4)</td><td>(5)</td></tr>
## <tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left">Clientelism</td><td>
-0.126<sup>***</sup></td><td></td><td></td><td></td><td>-0.078<sup>***</sup></td></tr>
## <tr><td style="text-align:left"></td><td>(0.014)</td><td></td><td></td><td></td><td>(0.026)</td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">Polarization</td><td></td><td>-0.064<sup>***</sup></td><td></td><td></td><td>-0.054<sup>*
**</sup></td></tr>
## <tr><td style="text-align:left"></td><td></td><td>(0.010)</td><td></td><td></td><td>(0.017)</td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">Online Media Consumption</td><td></td><td></td><td>-0.144<sup>***</sup></td><td></td><td>
-0.100<sup>*</sup></td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td>(0.049)</td><td></td><td>(0.054)</td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">Online Media Fractionalization</td><td></td><td></td><td>-0.373<sup>***</sup></td><td></t
d><td>-0.266<sup>***</sup></td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td>(0.070)</td><td></td><td>(0.076)</td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">Party Disinformation</td><td></td><td></td><td></td><td>-0.106<sup>***</sup></td><td>-0.0
69<sup>***</sup></td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td>(0.021)</td><td>(0.026)</td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">Level of Democracy</td><td>-0.302<sup>***</sup></td><td>-0.256<sup>***</sup></td><td>-0.2
98<sup>***</sup></td><td>-0.451<sup>***</sup></td><td>-0.623<sup>***</sup></td></tr>
## <tr><td style="text-align:left"></td><td>(0.030)</td><td>(0.030)</td><td>(0.059)</td><td>(0.072)</td><td>(0.087)</td></tr
>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">GDP Per Capita</td><td>0.0002</td><td>0.001<sup>***</sup></td><td>0.001<sup>***</sup></td
><td>0.001<sup>***</sup></td><td>0.001<sup>***</sup></td></tr>
## <tr><td style="text-align:left"></td><td>(0.0002)</td><td>(0.0002)</td><td>(0.0002)</td><td>(0.0002)</td><td>(0.0002)</td
></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">Online Media Consumption X Fractionalization</td><td></td><td></td><td>0.423<sup>***</sup
></td><td></td><td>0.331<sup>***</sup></td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td>(0.100)</td><td></td><td>(0.109)</td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td><td></td></tr>
## <tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left">Observations</td><td
>1,878</td><td>1,816</td><td>489</td><td>489</td><td>478</td></tr>
## <tr><td style="text-align:left">R<sup>2</sup></td><td>0.191</td><td>0.175</td><td>0.232</td><td>0.192</td><td>0.291</td>
</tr>
## <tr><td style="text-align:left">Adjusted R<sup>2</sup></td><td>0.139</td><td>0.120</td><td>0.208</td><td>0.170</td><td>0.
264</td></tr>
## <tr><td style="text-align:left">Residual Std. Error</td><td>0.056 (df = 1764)</td><td>0.057 (df = 1702)</td><td>0.047 (df
= 473)</td><td>0.048 (df = 475)</td><td>0.046 (df = 459)</td></tr>
## <tr><td style="text-align:left">F Statistic</td><td>3.687<sup>***</sup> (df = 113; 1764)</td><td>3.190<sup>***</sup> (df
= 113; 1702)</td><td>9.525<sup>***</sup> (df = 15; 473)</td><td>8.706<sup>***</sup> (df = 13; 475)</td><td>10.491<sup>***</s
up> (df = 18; 459)</td></tr>
## <tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"><em>Note:</em></td><
td colspan="5" style="text-align:right"><sup>*</sup>p<0.1; <sup>**</sup>p<0.05; <sup>***</sup>p<0.01</td></tr>
## </table>
```

```
stargazer(cm_print, pm_print, dm_print, threem_print,
          title = 'Solvents of Democracy',
          keep = c('polyarchy_change', 'v2xnp_client', 'v2cacamps', 'v2smpardom',
                   'v2x_polyarchy', 'e_migdppc'),
          dep.var.labels = 'Change in Polyarchy Score After 10 Years',
          model.names = TRUE,
          covariate.labels = c('Clientelism', 'Polarization', 'Party Disinformation',
                               'Level of Democracy', 'GDP Per Capita'),
          nobs = TRUE,
          type = 'html',
          out = './models/corrosive_factors_model_results.doc')
```

```
##
## <table style="text-align:center"><caption><strong>Solvents of Democracy</strong></caption>
## <tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"></td><td colspan
="4"><em>Dependent variable:</em></td></tr>
## <tr><td></td><td colspan="4" style="border-bottom: 1px solid black"></td></tr>
## <tr><td style="text-align:left"></td><td colspan="4">Change in Polyarchy Score After 10 Years</td></tr>
## <tr><td style="text-align:left"></td><td colspan="4"><em>OLS</em></td></tr>
## <tr><td style="text-align:left"></td><td>(1)</td><td>(2)</td><td>(3)</td><td>(4)</td></tr>
## <tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left">Clientelism</td><td>
-0.126<sup>***</sup></td><td></td><td></td><td>-0.091<sup>***</sup></td></tr>
## <tr><td style="text-align:left"></td><td>(0.014)</td><td></td><td></td><td>(0.026)</td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">Polarization</td><td></td><td>-0.064<sup>***</sup></td><td></td><td>-0.080<sup>***</sup>
</td></tr>
## <tr><td style="text-align:left"></td><td></td><td>(0.010)</td><td></td><td>(0.016)</td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">Party Disinformation</td><td></td><td></td><td>-0.106<sup>***</sup></td><td>-0.062<sup>**
</sup></td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td>(0.021)</td><td>(0.026)</td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">Level of Democracy</td><td>-0.302<sup>***</sup></td><td>-0.256<sup>***</sup></td><td>-0.4
51<sup>***</sup></td><td>-0.652<sup>***</sup></td></tr>
## <tr><td style="text-align:left"></td><td>(0.030)</td><td>(0.030)</td><td>(0.072)</td><td>(0.088)</td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">GDP Per Capita</td><td>0.0002</td><td>0.001<sup>***</sup></td><td>0.001<sup>***</sup></td
><td>0.001<sup>***</sup></td></tr>
## <tr><td style="text-align:left"></td><td>(0.0002)</td><td>(0.0002)</td><td>(0.0002)</td><td>(0.0002)</td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
## <tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left">Observations</td><td
>1,878</td><td>1,816</td><td>489</td><td>478</td></tr>
## <tr><td style="text-align:left">R<sup>2</sup></td><td>0.191</td><td>0.175</td><td>0.192</td><td>0.258</td></tr>
## <tr><td style="text-align:left">Adjusted R<sup>2</sup></td><td>0.139</td><td>0.120</td><td>0.170</td><td>0.234</td></tr>
## <tr><td style="text-align:left">Residual Std. Error</td><td>0.056 (df = 1764)</td><td>0.057 (df = 1702)</td><td>0.048 (df
= 475)</td><td>0.047 (df = 462)</td></tr>
## <tr><td style="text-align:left">F Statistic</td><td>3.687<sup>***</sup> (df = 113; 1764)</td><td>3.190<sup>***</sup> (df
= 113; 1702)</td><td>8.706<sup>***</sup> (df = 13; 475)</td><td>10.731<sup>***</sup> (df = 15; 462)</td></tr>
## <tr><td colspan="5" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"><em>Note:</em></td><
td colspan="4" style="text-align:right"><sup>*</sup>p<0.1; <sup>**</sup>p<0.05; <sup>***</sup>p<0.01</td></tr>
## </table>
```

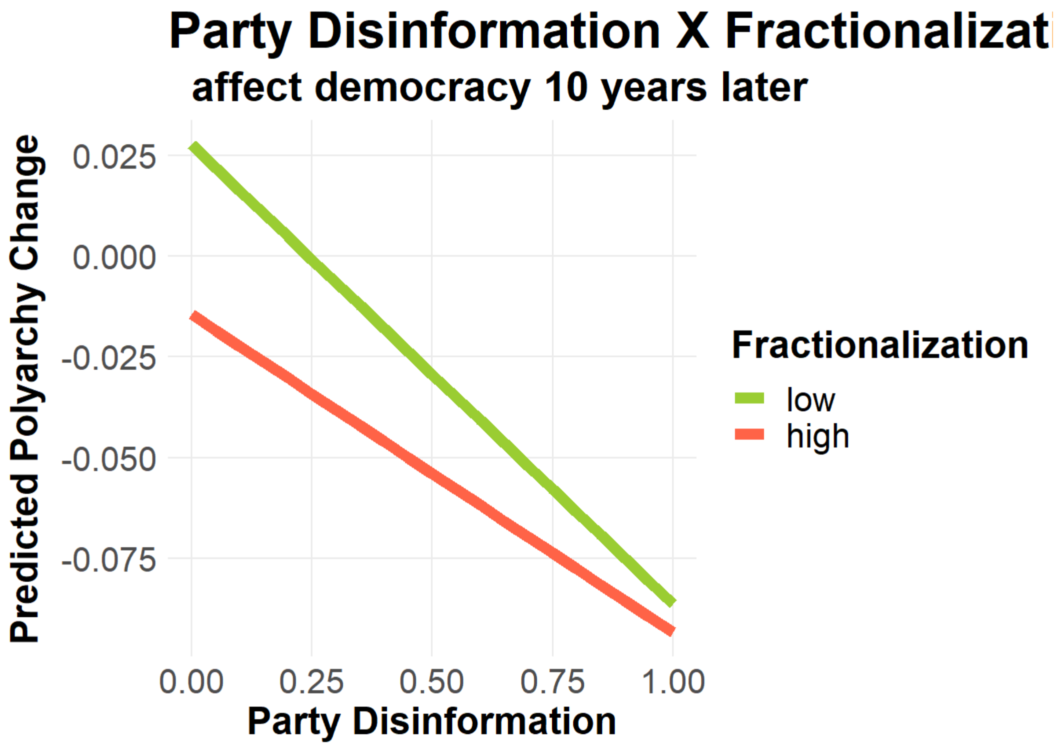
Chart interacted variables

Preliminary terms

```
lag_years = 10
point_scale = seq(from = 0, to = 1, by = 0.05)
df = vdem_con
```

Chart party disinformation by media fractionalization

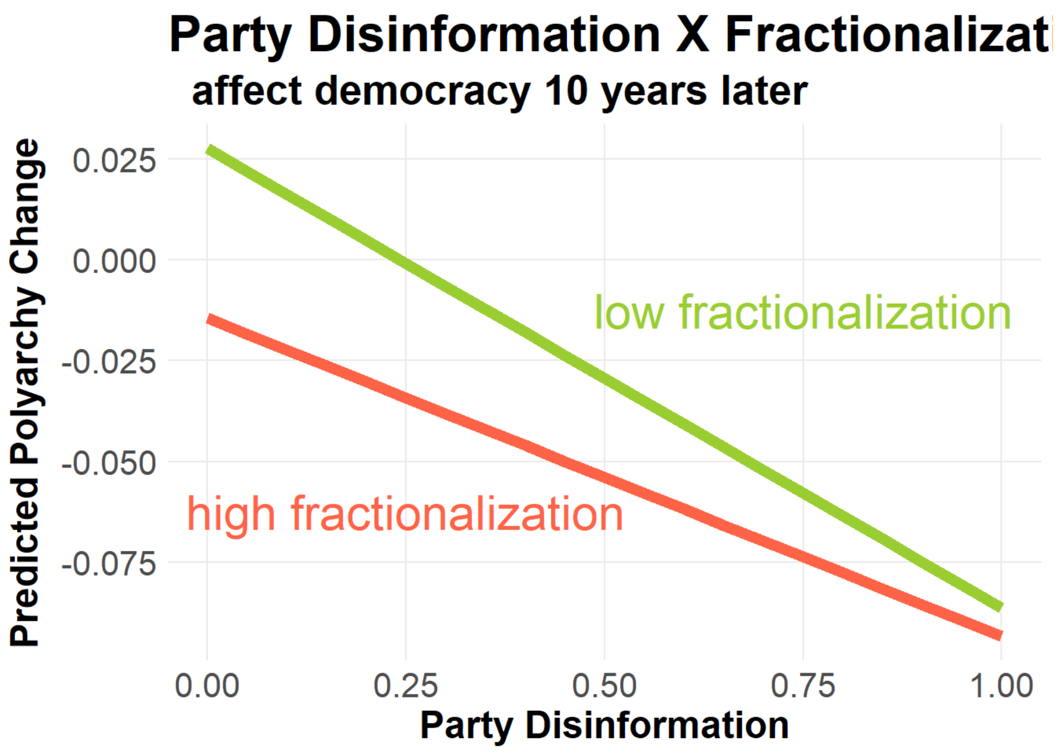
```
mimir <- data.frame(v2smpardom = rep(point_scale, times = length(point_scale)),
                    v2smmefra = rep(point_scale, each = length(point_scale)),
                    v2x_polyarchy = median(vdem_con$v2x_polyarchy[df$v2x_polyarchy >= 0.5], na.rm = TRUE),
                    e_migdppc = median(vdem_con$e_migdppc[df$v2x_polyarchy >= 0.5], na.rm = TRUE))
wdf <- vdem %>%
  group_by(country_name) %>%
  arrange(year) %>%
  mutate(v2x_polyarchy_lagged = lead(v2x_polyarchy, n = lag_years)) %>%
  ungroup()%>%
  filter(consolidated_lhb == TRUE)
wdf$polyarchy_change <- wdf$v2x_polyarchy_lagged - wdf$v2x_polyarchy
wm <- lm(polyarchy_change ~ v2smpardom * v2smmefra + v2x_polyarchy + e_migdppc, data = wdf)

mimir <- mimir %>% mutate(expected_polyarchy = predict(object = wm, newdata = mimir))
mimir %>% filter(v2smmefra %in% c(0.25, 0.75)) %>%
ggplot(aes(x = v2smpardom, y = expected_polyarchy, color = as.factor(v2smmefra)))+
  geom_line(size = 2.5)+
  scale_color_manual(values = c(disinfo_color, polar_color),
                     labels = c('low', 'high'))+
  theme_minimal()+
  labs(title = 'Party Disinformation X Fractionalization',
       subtitle = '  affect democracy 10 years later',
       y = 'Predicted Polyarchy Change',
       x = 'Party Disinformation',
       color = 'Fractionalization')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        axis.title.y = element_text(margin = margin(r = 8)),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 16))
```



```
ggsave(filename = "./visuals/model_pardisinfo_fract_inter.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

Repeat chart with internal labels

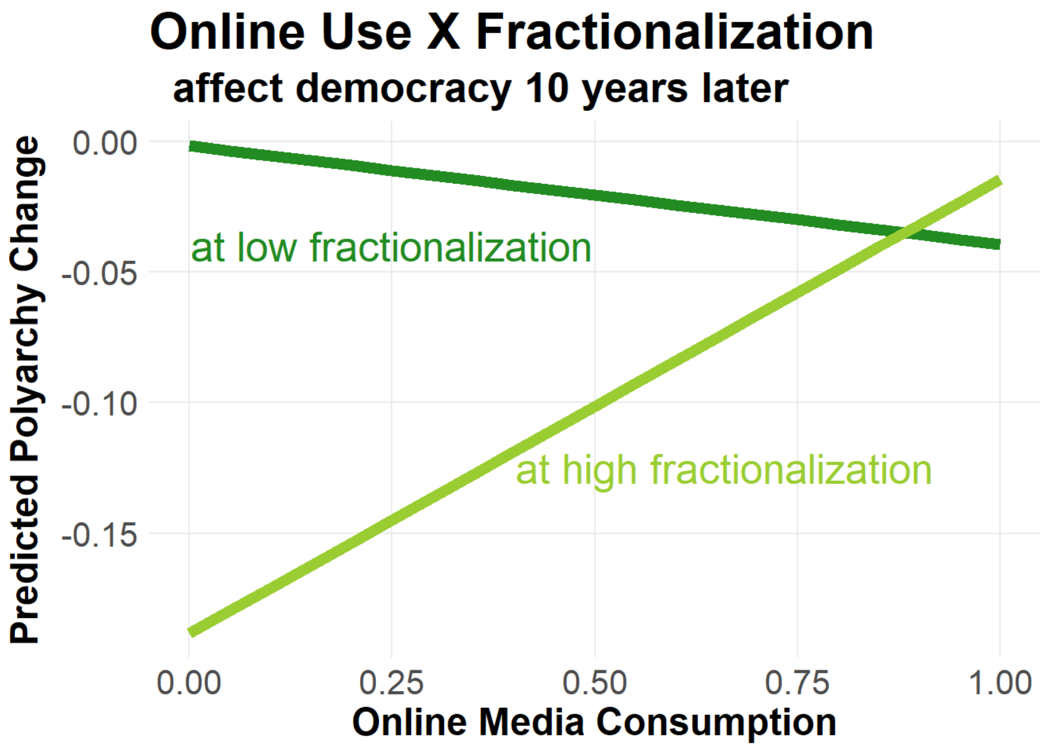
```
mimir %>% filter(v2smmefra %in% c(0.25, 0.75)) %>%
  ggplot(aes(x = v2smpardom, y = expected_polyarchy, color = as.factor(v2smmefra)))+
  geom_line(size = 2.5)+
  scale_color_manual(values = c(disinfo_color, polar_color),
                     labels = c('low', 'high'))+
  theme_minimal()+
  labs(title = 'Party Disinformation X Fractionalization',
       subtitle = '  affect democracy 10 years later',
       y = 'Predicted Polyarchy Change',
       x = 'Party Disinformation',
       color = 'Fractionalization')+
  annotate('text', label = 'low fractionalization', size = 8, color = disinfo_color, x = 0.75, y = -0.0125)+
  annotate('text', label = 'high fractionalization', size = 8, color = polar_color, x = 0.25, y = -0.0625)+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        axis.title.y = element_text(margin = margin(r = 8)),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16),
        legend.position = 'none')
```



```
ggsave(filename = "./visuals/model_pardisinfo_fract_inter_nolegend.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

Chart online consumption by media fractionalization

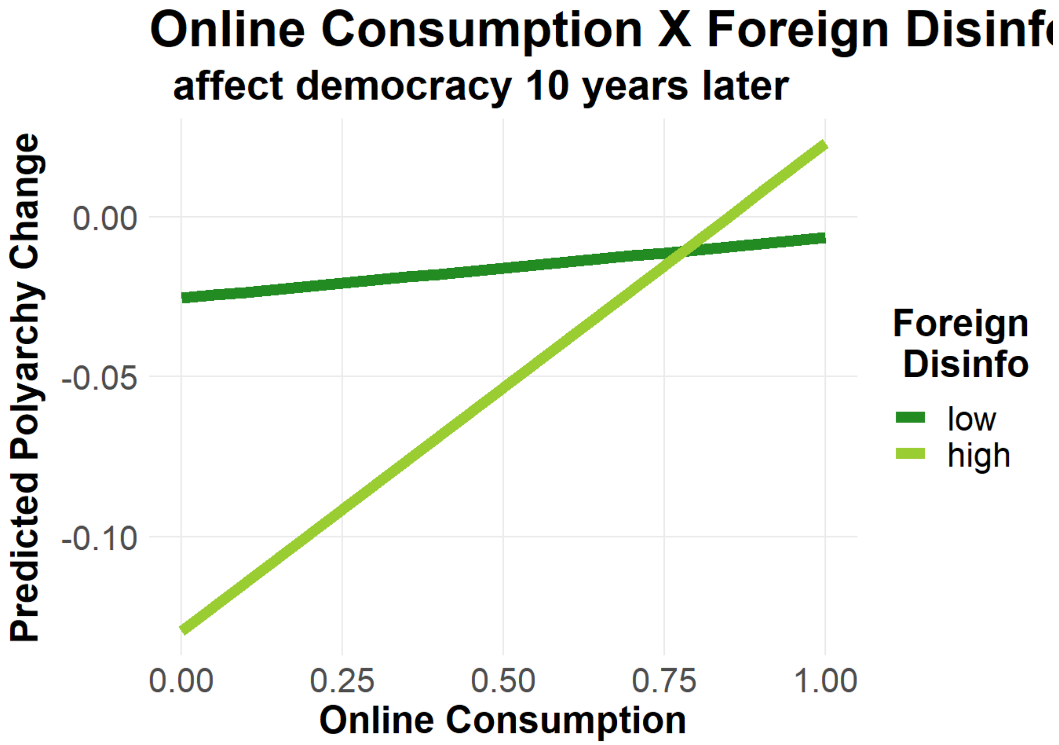
```
mimir <- data.frame(v2smonex = rep(point_scale, times = length(point_scale)),
                    v2smmefra = rep(point_scale, each = length(point_scale)),
                    v2x_polyarchy = median(vdem_con$v2x_polyarchy[df$v2x_polyarchy >= 0.5], na.rm = TRUE),
                    e_migdppc = median(vdem_con$e_migdppc[df$v2x_polyarchy >= 0.5], na.rm = TRUE),
                    year = as.factor(2010))
wdf <- vdem %>%
  group_by(country_name) %>%
  arrange(year) %>%
  mutate(v2x_polyarchy_lagged = lead(v2x_polyarchy, n = lag_years)) %>%
  ungroup() %>%
  filter(consolidated_lhb == TRUE)
wdf$polyarchy_change <- wdf$v2x_polyarchy_lagged - wdf$v2x_polyarchy
wm <- lm(polyarchy_change ~ v2smonex * v2smmefra + v2x_polyarchy + e_migdppc + as.factor(year), data = wdf)

mimir <- mimir %>% mutate(expected_polyarchy = predict(object = wm, newdata = mimir))
mimir %>% filter(v2smmefra %in% c(0.25, 0.75)) %>%
  ggplot(aes(x = v2smonex, y = expected_polyarchy, color = as.factor(v2smmefra)))+
  geom_line(size = 2.5)+
  scale_color_manual(guide = 'none',
                     values = c(media_color, disinfo_color))+
  theme_minimal()+
  labs(title = 'Online Use X Fractionalization',
       subtitle = '  affect democracy 10 years later',
       y = 'Predicted Polyarchy Change',
       x = 'Online Media Consumption')+
  annotate('text', label = 'at low fractionalization', color = media_color, size = 7, x = 0.25, y = -0.04)+
  annotate('text', label = 'at high fractionalization', color = disinfo_color, size = 7, x = 0.66, y = -0.125)+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        axis.title.y = element_text(margin = margin(r = 8)),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16),
        legend.position = 'none')
```



```
ggsave(filename = "./visuals/model_online_fract_inter.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

Chart online consumption by foreign disinformation

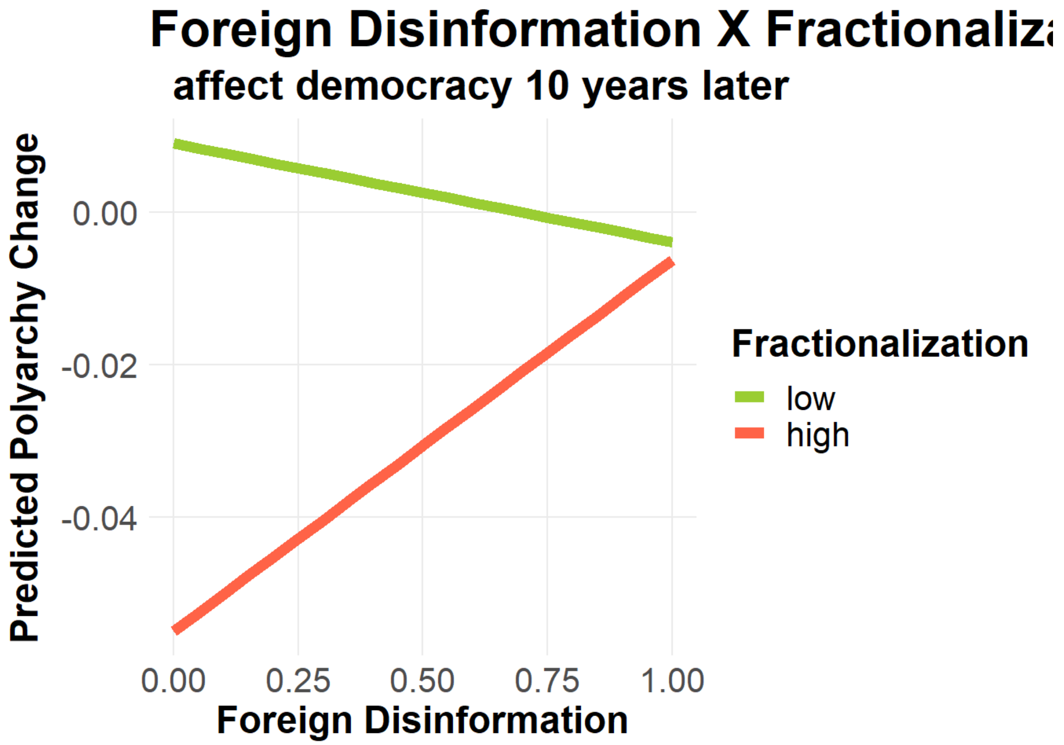
```
mimir <- data.frame(v2smonex = rep(point_scale, times = length(point_scale)),
                    v2smfordom = rep(point_scale, each = length(point_scale)),
                    v2x_polyarchy = median(vdem_con$v2x_polyarchy[df$v2x_polyarchy >= 0.5], na.rm = TRUE),
                    e_migdppc = median(vdem_con$e_migdppc[df$v2x_polyarchy >= 0.5], na.rm = TRUE))
wdf <- vdem %>%
  group_by(country_name) %>%
  arrange(year) %>%
  mutate(v2x_polyarchy_lagged = lead(v2x_polyarchy, n = lag_years)) %>%
  ungroup()%>%
  filter(consolidated_lhb == TRUE)
wdf$polyarchy_change <- wdf$v2x_polyarchy_lagged - wdf$v2x_polyarchy
wm <- lm(polyarchy_change ~ v2smonex * v2smfordom + v2x_polyarchy + e_migdppc, data = wdf)

mimir <- mimir %>% mutate(expected_polyarchy = predict(object = wm, newdata = mimir))
mimir %>% filter(v2smfordom %in% c(0.25, 0.75)) %>%
  ggplot(aes(x = v2smonex, y = expected_polyarchy, color = as.factor(v2smfordom)))+
  geom_line(size = 2.5)+
  scale_color_manual(values = c(media_color, disinfo_color),
                     labels = c('low', 'high'))+
  theme_minimal()+
  labs(title = 'Online Consumption X Foreign Disinformation',
       subtitle = '  affect democracy 10 years later',
       y = 'Predicted Polyarchy Change',
       x = 'Online Consumption',
       color = 'Foreign\n Disinfo')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        axis.title.y = element_text(margin = margin(r = 8)),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 16))
```



```
ggsave(filename = "./visuals/model_online_fordisinfo_inter.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

Chart foreign disinformation by media fractionalization

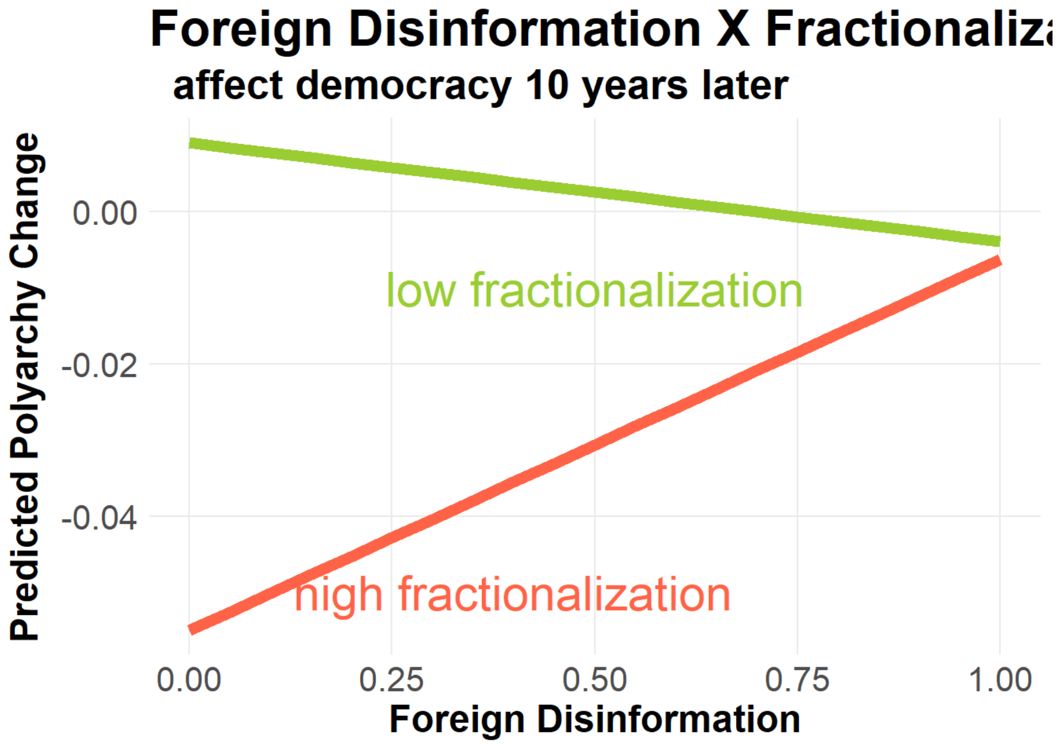
```
mimir <- data.frame(v2smfordom = rep(point_scale, times = length(point_scale)),
                    v2smmefra = rep(point_scale, each = length(point_scale)),
                    v2x_polyarchy = median(vdem_con$v2x_polyarchy[df$v2x_polyarchy >= 0.5], na.rm = TRUE),
                    e_migdppc = median(vdem_con$e_migdppc[df$v2x_polyarchy >= 0.5], na.rm = TRUE))
wdf <- vdem %>%
  group_by(country_name) %>%
  arrange(year) %>%
  mutate(v2x_polyarchy_lagged = lead(v2x_polyarchy, n = lag_years)) %>%
  ungroup()%>%
  filter(consolidated_lhb == TRUE)
wdf$polyarchy_change <- wdf$v2x_polyarchy_lagged - wdf$v2x_polyarchy
wm <- lm(polyarchy_change ~ v2smfordom * v2smmefra + v2x_polyarchy + e_migdppc, data = wdf)

mimir <- mimir %>% mutate(expected_polyarchy = predict(object = wm, newdata = mimir))
mimir %>% filter(v2smmefra %in% c(0.25, 0.75)) %>%
  ggplot(aes(x = v2smfordom, y = expected_polyarchy, color = as.factor(v2smmefra)))+
  geom_line(size = 2.5)+
  scale_color_manual(values = c(disinfo_color, polar_color),
                     labels = c('low', 'high'))+
  theme_minimal()+
  labs(title = 'Foreign Disinformation X Fractionalization',
       subtitle = '  affect democracy 10 years later',
       y = 'Predicted Polyarchy Change',
       x = 'Foreign Disinformation',
       color = 'Fractionalization')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        axis.title.y = element_text(margin = margin(r = 8)),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 16))
```



```
ggsave(filename = "./visuals/model_fordisinfo_fract_inter.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

Repeat with internal lables

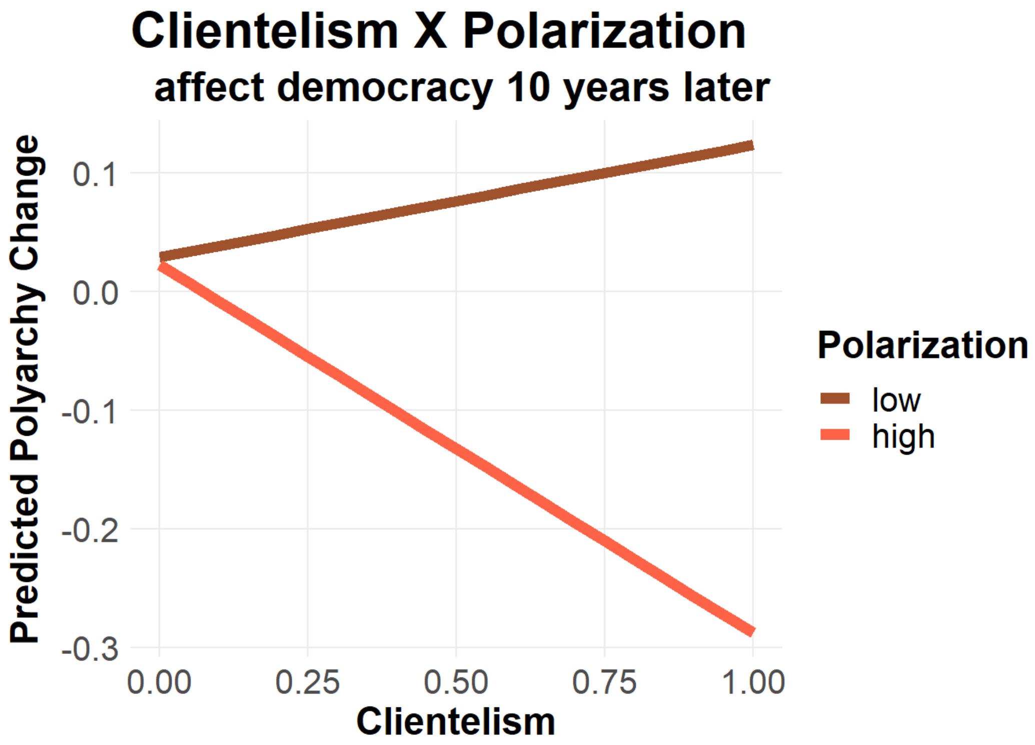
```
mimir %>% filter(v2smmefra %in% c(0.25, 0.75)) %>%
  ggplot(aes(x = v2smfordom, y = expected_polyarchy, color = as.factor(v2smmefra)))+
  geom_line(size = 2.5)+
  scale_color_manual(values = c(disinfo_color, polar_color),
                     labels = c('low', 'high'))+
  theme_minimal()+
  labs(title = 'Foreign Disinformation X Fractionalization',
       subtitle = '  affect democracy 10 years later',
       y = 'Predicted Polyarchy Change',
       x = 'Foreign Disinformation',
       color = 'Fractionalization')+
  annotate('text', label = 'low fractionalization', size = 8, color = disinfo_color, x = 0.5, y = -0.01)+
  annotate('text', label = 'high fractionalization', size = 8, color = polar_color, x = 0.4, y = -0.05)+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        axis.title.y = element_text(margin = margin(r = 8)),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16),
        legend.position = 'none')
```



```
ggsave(filename = "./visuals/model_fordisinfo_fract_inter_nolegend.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

Chart clientelism by polarized society

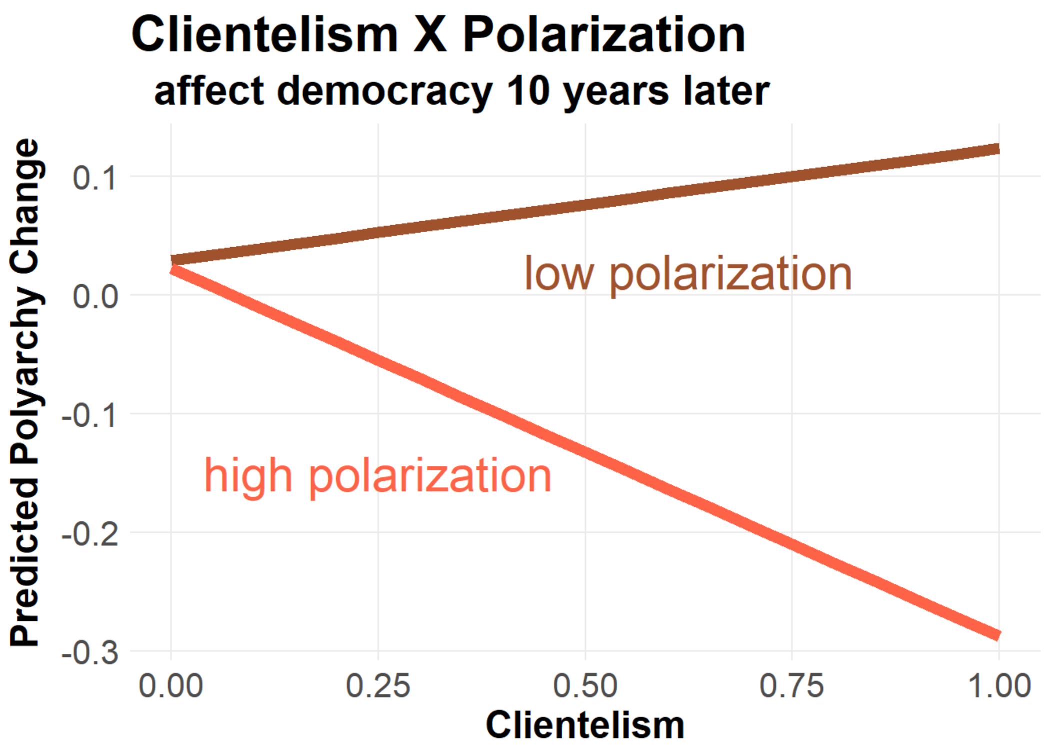
```
mimir <- data.frame(v2xnp_client = rep(point_scale, times = length(point_scale)),
                    v2smpolsoc = rep(point_scale, each = length(point_scale)),
                    v2x_polyarchy = median(vdem_con$v2x_polyarchy[df$v2x_polyarchy >= 0.5], na.rm = TRUE),
                    e_migdppc = median(vdem_con$e_migdppc[df$v2x_polyarchy >= 0.5], na.rm = TRUE))
wdf <- vdem %>%
  group_by(country_name) %>%
  arrange(year) %>%
  mutate(v2x_polyarchy_lagged = lead(v2x_polyarchy, n = lag_years)) %>%
  ungroup()%>%
  filter(consolidated_lhb == TRUE)
wdf$polyarchy_change <- wdf$v2x_polyarchy_lagged - wdf$v2x_polyarchy
wm <- lm(polyarchy_change ~ v2xnp_client * v2smpolsoc + v2x_polyarchy + e_migdppc, data = wdf)

mimir <- mimir %>% mutate(expected_polyarchy = predict(object = wm, newdata = mimir))
mimir %>% filter(v2smpolsoc %in% c(0.25, 0.75)) %>%
  ggplot(aes(x = v2xnp_client, y = expected_polyarchy, color = as.factor(v2smpolsoc)))+
  geom_line(size = 2.5)+
  scale_color_manual(values = c(client_color, polar_color),
                     labels = c('low', 'high'))+
  theme_minimal()+
  labs(title = 'Clientelism X Polarization',
       subtitle = '  affect democracy 10 years later',
       y = 'Predicted Polyarchy Change',
       x = 'Clientelism',
       color = 'Polarization')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        axis.title.y = element_text(margin = margin(r = 8)),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 16))
```



```
ggsave(filename = "./visuals/model_client_pol_inter.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

Repeat with internal labels

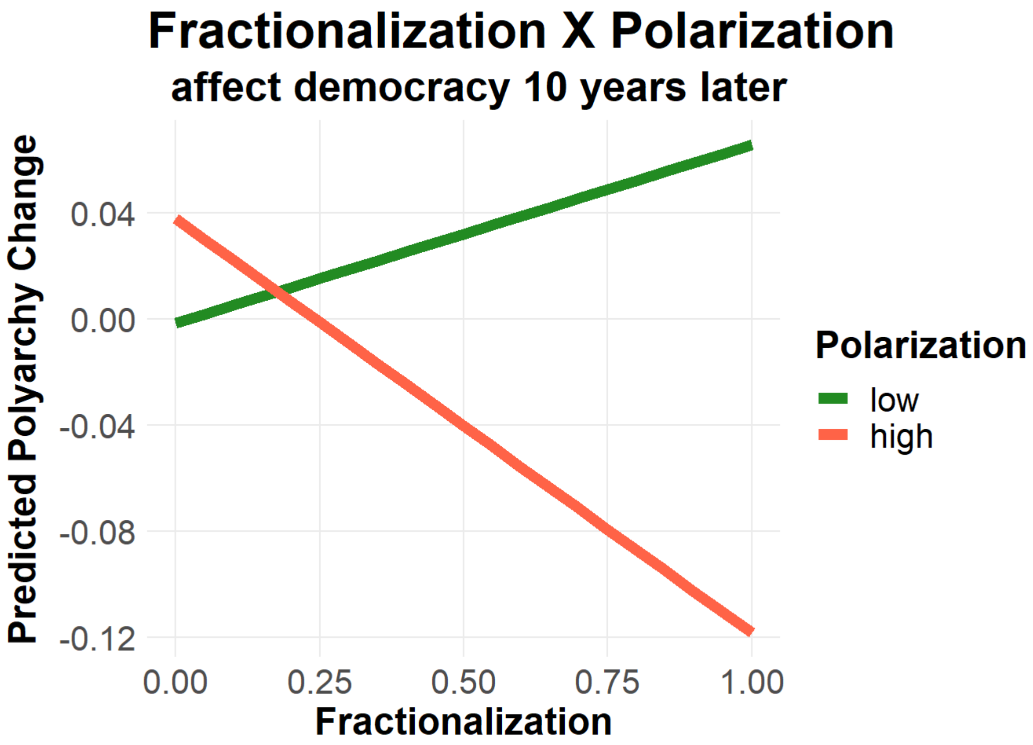
```
mimir %>% filter(v2smpolsoc %in% c(0.25, 0.75)) %>%
  ggplot(aes(x = v2xnp_client, y = expected_polyarchy, color = as.factor(v2smpolsoc)))+
  geom_line(size = 2.5)+
  scale_color_manual(values = c(client_color, polar_color),
                     labels = c('low', 'high'))+
  theme_minimal()+
  labs(title = 'Clientelism X Polarization',
       subtitle = '  affect democracy 10 years later',
       y = 'Predicted Polyarchy Change',
       x = 'Clientelism',
       color = 'Polarization')+
  annotate('text', label = 'low polarization', size = 8, color = client_color, x = 0.625, y = 0.02)+
  annotate('text', label = 'high polarization', size = 8, color = polar_color, x = 0.25, y = -0.15)+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        axis.title.y = element_text(margin = margin(r = 8)),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16),
        legend.position = 'none')
```



```
ggsave(filename = "./visuals/model_client_pol_inter_nolegend.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

Chart media fractionalization by polarized society

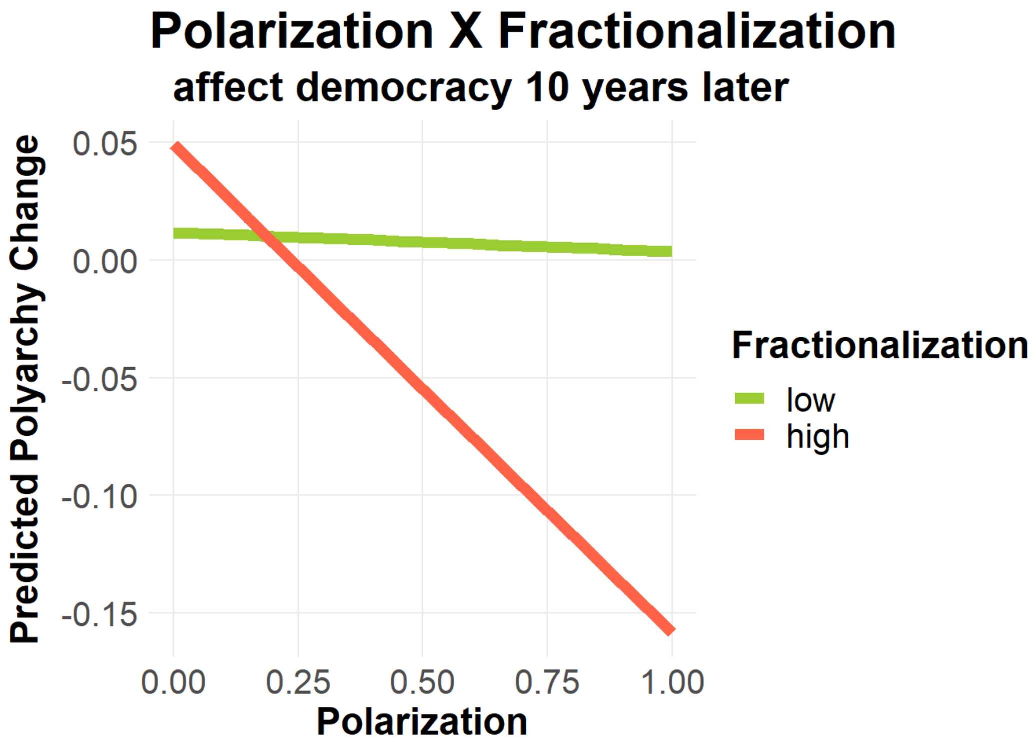
```
mimir <- data.frame(v2smmefra = rep(point_scale, times = length(point_scale)),
                    v2smpolsoc = rep(point_scale, each = length(point_scale)),
                    v2x_polyarchy = median(vdem_con$v2x_polyarchy[df$v2x_polyarchy >= 0.5], na.rm = TRUE),
                    e_migdppc = median(vdem_con$e_migdppc[df$v2x_polyarchy >= 0.5], na.rm = TRUE))
wdf <- vdem %>%
  group_by(country_name) %>%
  arrange(year) %>%
  mutate(v2x_polyarchy_lagged = lead(v2x_polyarchy, n = lag_years)) %>%
  ungroup()%>%
  filter(consolidated_lhb == TRUE)
wdf$polyarchy_change <- wdf$v2x_polyarchy_lagged - wdf$v2x_polyarchy
wm <- lm(polyarchy_change ~ v2smmefra * v2smpolsoc + v2x_polyarchy + e_migdppc, data = wdf)

mimir <- mimir %>% mutate(expected_polyarchy = predict(object = wm, newdata = mimir))
mimir %>% filter(v2smpolsoc %in% c(0.25, 0.75)) %>%
  ggplot(aes(x = v2smmefra, y = expected_polyarchy, color = as.factor(v2smpolsoc)))+
  geom_line(size = 2.5)+
  scale_color_manual(values = c(media_color, polar_color),
                     labels = c('low', 'high'))+
  theme_minimal()+
  labs(title = 'Fractionalization X Polarization',
       subtitle = '  affect democracy 10 years later',
       y = 'Predicted Polyarchy Change',
       x = 'Fractionalization',
       color = 'Polarization')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        axis.title.y = element_text(margin = margin(r = 8)),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 16))
```



```
ggsave(filename = "./visuals/model_fract_pol_inter.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

Chart polarized society by media fractionalization

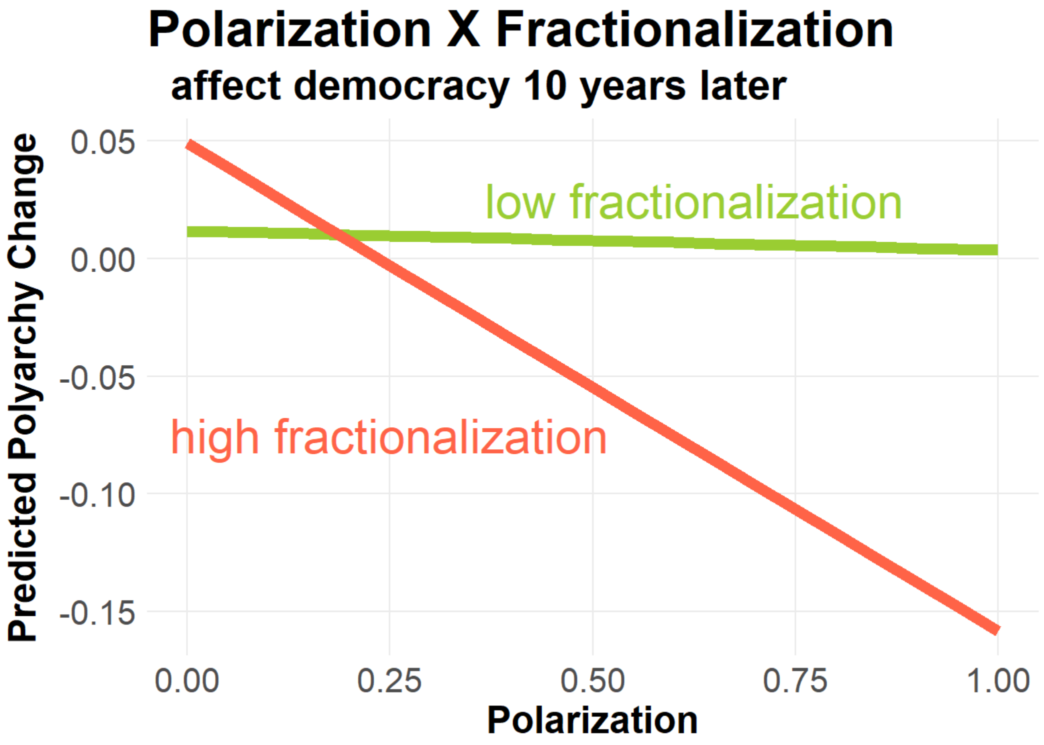
```
mimir <- data.frame(v2cacamps = rep(point_scale, times = length(point_scale)),
                     v2smmefra = rep(point_scale, each = length(point_scale)),
                    v2x_polyarchy = median(vdem_con$v2x_polyarchy[df$v2x_polyarchy >= 0.5], na.rm = TRUE),
                    e_migdppc = median(vdem_con$e_migdppc[df$v2x_polyarchy >= 0.5], na.rm = TRUE))
wdf <- vdem %>%
  group_by(country_name) %>%
  arrange(year) %>%
  mutate(v2x_polyarchy_lagged = lead(v2x_polyarchy, n = lag_years)) %>%
  ungroup()%>%
  filter(consolidated_lhb == TRUE)
wdf$polyarchy_change <- wdf$v2x_polyarchy_lagged - wdf$v2x_polyarchy
wm <- lm(polyarchy_change ~ (v2cacamps * v2smmefra) + v2x_polyarchy + e_migdppc, data = wdf)

mimir <- mimir %>% mutate(expected_polyarchy = predict(object = wm, newdata = mimir))
mimir %>% filter(v2smmefra %in% c(0.25, 0.75)) %>%
  ggplot(aes(x = v2cacamps, y = expected_polyarchy, color = as.factor(v2smmefra)))+
  geom_line(size = 2.5)+
  scale_color_manual(values = c(disinfo_color, polar_color),
                     labels = c('low', 'high'))+
  theme_minimal()+
  labs(title = 'Polarization X Fractionalization',
       subtitle = '  affect democracy 10 years later',
       y = 'Predicted Polyarchy Change',
       x = 'Polarization',
       color = 'Fractionalization')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        axis.title.y = element_text(margin = margin(r = 8)),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 16))
```



```
ggsave(filename = "./visuals/model_pol_fract_inter.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

Repeat with internal labels

```
mimir %>% filter(v2smmefra %in% c(0.25, 0.75)) %>%
  ggplot(aes(x = v2cacamps, y = expected_polyarchy, color = as.factor(v2smmefra)))+
  geom_line(size = 2.5)+
  scale_color_manual(values = c(disinfo_color, polar_color),
                     labels = c('low', 'high'))+
  theme_minimal()+
  labs(title = 'Polarization X Fractionalization',
       subtitle = '  affect democracy 10 years later',
       y = 'Predicted Polyarchy Change',
       x = 'Polarization',
       color = 'Fractionalization')+
  annotate('text', label = 'low fractionalization', size = 8, color = disinfo_color, x = 0.625, y = 0.025)+
  annotate('text', label = 'high fractionalization', size = 8, color = polar_color, x = 0.25, y = -0.075)+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        axis.title.y = element_text(margin = margin(r = 8)),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16),
        legend.position = 'none')
```



```
ggsave(filename = "./visuals/model_pol_fract_inter_nolegend.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

Difference in difference charts

Observe polyarchy relative to treatments

dif in dif - clientelism set level that counts as being “treated” with clientelism

```
clientelism_threshold <- as.numeric(quantile(vdem_con$v2xnp_client, probs = treatment_threshold, na.rm = TRUE))
```

Label years relative to onset of clientelism treatment only count as treatment if happens while inside a democratic spell, and if first year of spell did not see high clientelism but year count extends ten years prior and after the treatment, regardless of democratic status in those years

```
vdem$year_rel_client <- as.numeric(NA)
vdem$client_control_poly <- as.numeric(NA)
for (i in seq_along(vdem$year)){
  #skip country-years that are not in democratic spells or lack score
  if (is.na(vdem$dem_spell_name[i])) next
  if (is.na(vdem$v2xnp_client[i])) next
  #skip country-years that are in democratic spells that began with high score
  if (vdem$v2xnp_client[!is.na(vdem$dem_spell_name) &
                        vdem$dem_spell_name == vdem$dem_spell_name[i] &
                        !is.na(vdem$dem_spell_running) &
                        vdem$dem_spell_running == 0] > clientelism_threshold) next

  #find earliest year within each dem spell that crosses treatment threshold.
  client_year_zero <- vdem %>% filter(dem_spell_name == vdem$dem_spell_name[i]) %>%
    filter(v2xnp_client >= clientelism_threshold) %>%
    summarize(client_year_zero = min(year, na.rm = TRUE)) %>%
    pull(client_year_zero)

  #renumber infinite values as nulls
  #value is Inf if spell never crossed threshold
  client_year_zero <- if_else((client_year_zero == Inf | client_year_zero == -Inf),
                              as.numeric(NA),
                              client_year_zero)

  if (is.na(client_year_zero)) next

  #label year of onset as relative year zero
  vdem$year_rel_client[i] <- if_else(vdem$year[i] == client_year_zero,
                                     0,
                                     as.numeric(NA))
}
for (i in seq_along(vdem$year)){
  #locate treatment onset within 10 years of each country-year
  client_year_zero <- vdem %>%
    filter(country_name == vdem$country_name[i] &
             year >= (vdem$year[i] - 10) &
             year <= (vdem$year[i] + 10) &
             year_rel_client == 0) %>%
    summarize(client_year_zero = min(year)) %>%
    pull(client_year_zero)

  #renumber infinite values as nulls
  #value is Inf if spell never crossed threshold
  client_year_zero <- if_else((client_year_zero == Inf | client_year_zero == -Inf),
                              as.numeric(NA),
                              client_year_zero)

  if (is.na(client_year_zero)) next

  #label all preceding and following years in spell, relative to that zero
  vdem$year_rel_client[i] = vdem$year[i] - client_year_zero

  #log control value of clientelism among all consolidated democracies in same absolute year
  vdem$client_control_poly[i] <- if_else(is.na(vdem$year_rel_client[i]),
                                         as.numeric(NA),
                                         mean(vdem$v2x_polyarchy[vdem$v2x_polyarchy >= dem_threshold &
                                                                   vdem$year == vdem$year[i] &
                                                                   vdem$country_name != vdem$country_name[i]],
                                              na.rm = TRUE))
}
summary(vdem$year_rel_client)
```

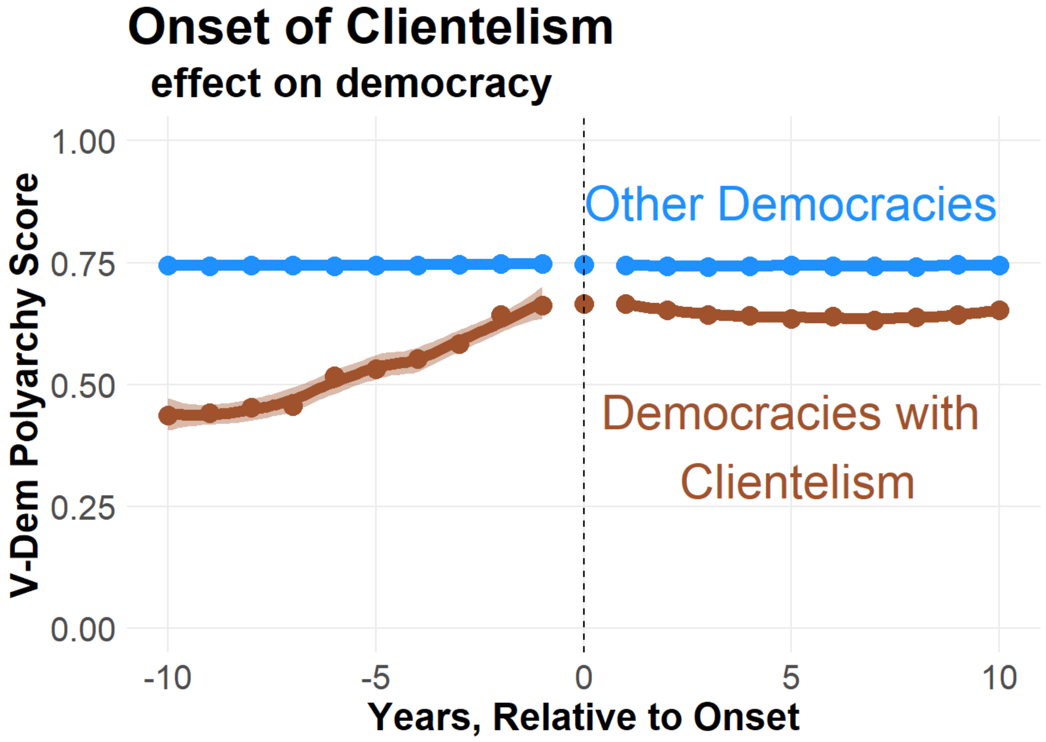
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## -10.000  -5.000   0.000  -0.241   5.000  10.000   25122
```

```
summary(vdem$client_control_poly)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.671   0.738   0.743   0.745   0.751   0.789   25122
```

Chart dif in dif - clientelism

```
vdem %>%
  filter(!is.na(year_rel_client)) %>%
  group_by(year_rel_client) %>%
  summarize(polyarchy_client = mean(v2x_polyarchy, na.rm = TRUE),
            polyarchy_control = mean(client_control_poly, na.rm = TRUE)) %>%
  ggplot(aes(x = year_rel_client))+
  geom_point(aes(y = polyarchy_client), color = client_color, size = 4)+
  geom_smooth(data = . %>% filter(year_rel_client < 0),
              method = 'loess',
    aes(y = polyarchy_client), color = client_color, size = 2.5, fill = client_color)+
  geom_smooth(data = . %>% filter(year_rel_client > 0),
              method = 'loess',
              aes(y = polyarchy_client), color = client_color, size = 2.5, fill = client_color)+
  geom_point(aes(y = polyarchy_control), color = dem_color, size = 4)+
  geom_smooth(data = . %>% filter(year_rel_client < 0),
              method = 'loess',
              aes(y = polyarchy_control), color = dem_color, size = 2.5, fill = dem_color)+
  geom_smooth(data = . %>% filter(year_rel_client > 0),
              method = 'loess',
              aes(y = polyarchy_control), color = dem_color, size = 2.5, fill = dem_color)+
  coord_cartesian(xlim = c(-10,10), ylim = c(0,1))+
  geom_vline(xintercept = 0, linetype = 'dashed')+
  annotate('text', label = 'Democracies with\n Clientelism', x = 5, y = 0.375, color = client_color, size = 8)+
  annotate('text', label = 'Other Democracies', x = 5, y = 0.875, color = dem_color, size = 8)+
  theme_minimal()+
  labs(title = 'Onset of Clientelism',
       subtitle = '  effect on democracy',
       y = 'V-Dem Polyarchy Score',
       x = 'Years, Relative to Onset')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16))
```



```
ggsave(filename = "./visuals/client_dif.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

dif in dif - polarization set level that counts as being “treated” with polarization

```
polarization_threshold <- as.numeric(quantile(vdem_con$v2cacamps, probs = treatment_threshold, na.rm = TRUE))
```

Label years relative to onset of polarization treatment only count as treatment if happens while inside a democratic spell, and if first year of spell did not see high polarization but year count extends ten years prior and after the treatment, regardless of democratic status in those years

```
vdem$year_rel_polar <- as.numeric(NA)
vdem$polar_control_poly <- as.numeric(NA)
for (i in seq_along(vdem$year)){
  #skip country-years that are not in democratic spells or lack score
  if (is.na(vdem$dem_spell_name[i])) next
  if (is.na(vdem$v2cacamps[i])) next

  #find earliest year within each dem spell that crosses treatment threshold.
  polar_year_zero <- vdem %>% filter(dem_spell_name == vdem$dem_spell_name[i]) %>%
    filter(v2cacamps >= polarization_threshold) %>%
    summarize(polar_year_zero = min(year, na.rm = TRUE)) %>%
    pull(polar_year_zero)

  #skip country-years that are in democratic spells that began with high score
  if (vdem$year[!is.na(vdem$dem_spell_name) &
                vdem$dem_spell_name == vdem$dem_spell_name[i] &
                !is.na(vdem$dem_spell_running) &
                vdem$dem_spell_running == 0] == polar_year_zero) next

  #renumber infinite values as nulls
  #value is Inf if spell never crossed threshold
  polar_year_zero <- if_else((polar_year_zero == Inf | polar_year_zero == -Inf),
                             as.numeric(NA),
                             polar_year_zero)

  if (is.na(polar_year_zero)) next

  #label year of onset as relative year zero
  vdem$year_rel_polar[i] <- if_else(vdem$year[i] == polar_year_zero,
                                    0,
                                    as.numeric(NA))
}
for (i in seq_along(vdem$year)){
  #locate treatment onset within 10 years of each country-year
  polar_year_zero <- vdem %>%
    filter(country_name == vdem$country_name[i] &
             year >= (vdem$year[i] - 10) &
             year <= (vdem$year[i] + 10) &
             year_rel_polar == 0) %>%
    summarize(polar_year_zero = min(year)) %>%
    pull(polar_year_zero)

  #renumber infinite values as nulls
  #value is Inf if spell never crossed threshold
  polar_year_zero <- if_else((polar_year_zero == Inf | polar_year_zero == -Inf),
                             as.numeric(NA),
                             polar_year_zero)

  if (is.na(polar_year_zero)) next

  #label all preceding and following years in spell, relative to that zero
  vdem$year_rel_polar[i] = vdem$year[i] - polar_year_zero

  #log control value of polarization among all consolidated democracies in same absolute year
  vdem$polar_control_poly[i] <- if_else(is.na(vdem$year_rel_polar[i]),
                                        as.numeric(NA),
                                        mean(vdem$v2x_polyarchy[vdem$v2x_polyarchy >= dem_threshold &
                                                                  vdem$year == vdem$year[i] &
                                                                  vdem$country_name != vdem$country_name[i]],
                                             na.rm = TRUE))
}
summary(vdem$year_rel_polar)
```

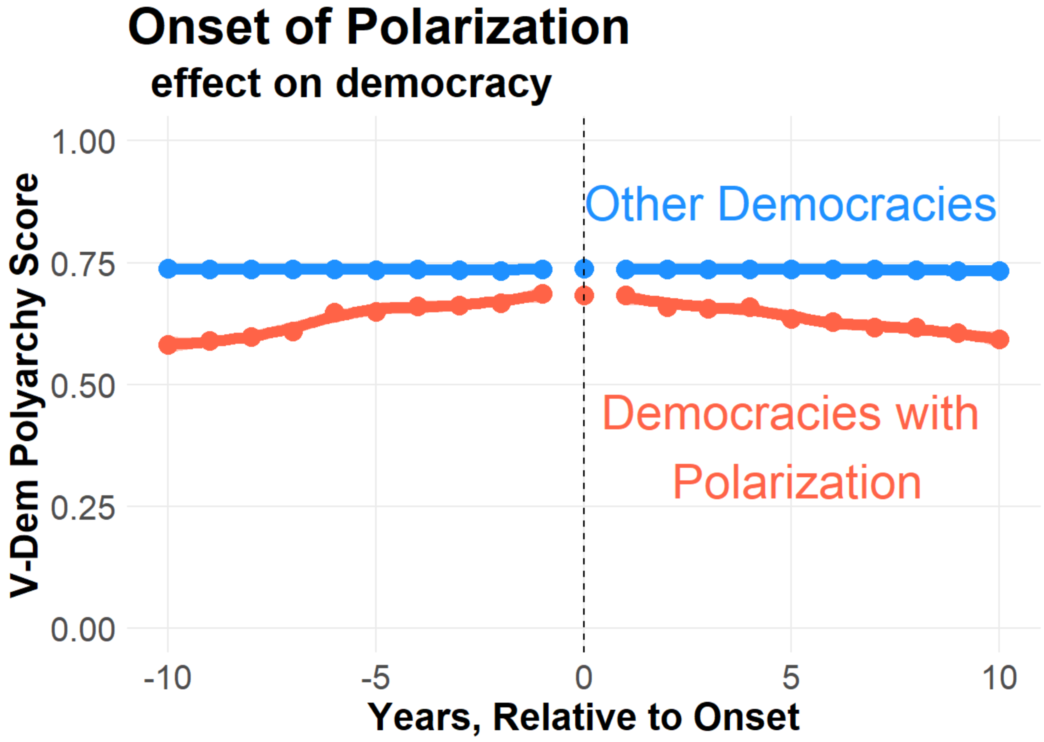
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
## -10.000  -5.000   0.000  -0.214   5.000  10.000   24857
```

```
summary(vdem$polar_control_poly)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.580   0.736   0.740   0.736   0.747   0.787   24857
```

Chart dif in dif - polarization

```
vdem %>%
  filter(!is.na(year_rel_polar)) %>%
  group_by(year_rel_polar) %>%
  summarize(polyarchy_polar = mean(v2x_polyarchy, na.rm = TRUE),
            polyarchy_control = mean(polar_control_poly, na.rm = TRUE)) %>%
  ggplot(aes(x = year_rel_polar))+
  geom_point(aes(y = polyarchy_polar), color = polar_color, size = 4)+
  geom_smooth(data = . %>% filter(year_rel_polar < 0),
              method = 'loess',
              aes(y = polyarchy_polar), color = polar_color, size = 2.5, fill = polar_color)+
  geom_smooth(data = . %>% filter(year_rel_polar > 0),
              method = 'loess',
              aes(y = polyarchy_polar), color = polar_color, size = 2.5, fill = polar_color)+
  geom_point(aes(y = polyarchy_control), color = dem_color, size = 4)+
  geom_smooth(data = . %>% filter(year_rel_polar < 0),
              method = 'loess',
              aes(y = polyarchy_control), color = dem_color, size = 2.5, fill = dem_color)+
  geom_smooth(data = . %>% filter(year_rel_polar > 0),
              method = 'loess',
              aes(y = polyarchy_control), color = dem_color, size = 2.5, fill = dem_color)+
  coord_cartesian(xlim = c(-10,10), ylim = c(0,1))+
  geom_vline(xintercept = 0, linetype = 'dashed')+
  annotate('text', label = 'Democracies with\n Polarization', x = 5, y = 0.375, color = polar_color, size = 8)+
  annotate('text', label = 'Other Democracies', x = 5, y = 0.875, color = dem_color, size = 8)+
  theme_minimal()+
  labs(title = 'Onset of Polarization',
       subtitle = '  effect on democracy',
       y = 'V-Dem Polyarchy Score',
       x = 'Years, Relative to Onset')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16))
```



```
ggsave(filename = "./visuals/polar_dif.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

dif in dif - party disinfo

Set level that counts as being “treated” with par_disinfo

```
par_disinfo_threshold <- as.numeric(quantile(vdem_con$v2smpardom, probs = treatment_threshold, na.rm = TRUE))
```

Label years relative to onset of party disinfo treatment only count as treatment if happens while inside a democratic spell, and if first year of spell did not see high par_disinfo but year count extends ten years prior and after the treatment, regardless of democratic status in those years

```
vdem$year_rel_par_disinfo <- as.numeric(NA)
vdem$par_disinfo_control_poly <- as.numeric(NA)
for (i in seq_along(vdem$year)){
  #skip country-years that are not in democratic spells or lack score
  if (is.na(vdem$dem_spell_name[i])) next
  if (is.na(vdem$v2smpardom[i])) next

  #find earliest year within each dem spell that crosses treatment threshold.
  par_disinfo_year_zero <- vdem %>% filter(dem_spell_name == vdem$dem_spell_name[i]) %>%
    filter(v2smpardom >= par_disinfo_threshold) %>%
    summarize(par_disinfo_year_zero = min(year, na.rm = TRUE)) %>%
    pull(par_disinfo_year_zero)

  #skip country-years that are in democratic spells that began with high score
  if (vdem$year[!is.na(vdem$dem_spell_name) &
                vdem$dem_spell_name == vdem$dem_spell_name[i] &
                !is.na(vdem$dem_spell_running) &
                vdem$dem_spell_running == 0] == par_disinfo_year_zero) next

  #renumber infinite values as nulls
  #value is Inf if spell never crossed threshold
  par_disinfo_year_zero <- if_else((par_disinfo_year_zero == Inf | par_disinfo_year_zero == -Inf),
                                   as.numeric(NA),
                                   par_disinfo_year_zero)

  if (is.na(par_disinfo_year_zero)) next

  #label year of onset as relative year zero
  vdem$year_rel_par_disinfo[i] <- if_else(vdem$year[i] == par_disinfo_year_zero,
                                          0,
                                          as.numeric(NA))
}
for (i in seq_along(vdem$year)){
  #locate treatment onset within 10 years of each country-year
  par_disinfo_year_zero <- vdem %>%
    filter(country_name == vdem$country_name[i] &
             year >= (vdem$year[i] - 10) &
             year <= (vdem$year[i] + 10) &
             year_rel_par_disinfo == 0) %>%
    summarize(par_disinfo_year_zero = min(year)) %>%
    pull(par_disinfo_year_zero)

  #renumber infinite values as nulls
  #value is Inf if spell never crossed threshold
  par_disinfo_year_zero <- if_else((par_disinfo_year_zero == Inf | par_disinfo_year_zero == -Inf),
                                   as.numeric(NA),
                                   par_disinfo_year_zero)

  if (is.na(par_disinfo_year_zero)) next

  #label all preceding and following years in spell, relative to that zero
  vdem$year_rel_par_disinfo[i] = vdem$year[i] - par_disinfo_year_zero

  #log control value of par_disinfo among all consolidated democracies in same absolute year
  vdem$par_disinfo_control_poly[i] <- if_else(is.na(vdem$year_rel_par_disinfo[i]),
                                              as.numeric(NA),
                                              mean(vdem$v2x_polyarchy[vdem$v2x_polyarchy >= dem_threshold &
                                                                        vdem$year == vdem$year[i] &
                                                                        vdem$country_name != vdem$country_name[i]],
                                                   na.rm = TRUE))
}
summary(vdem$year_rel_par_disinfo)
```

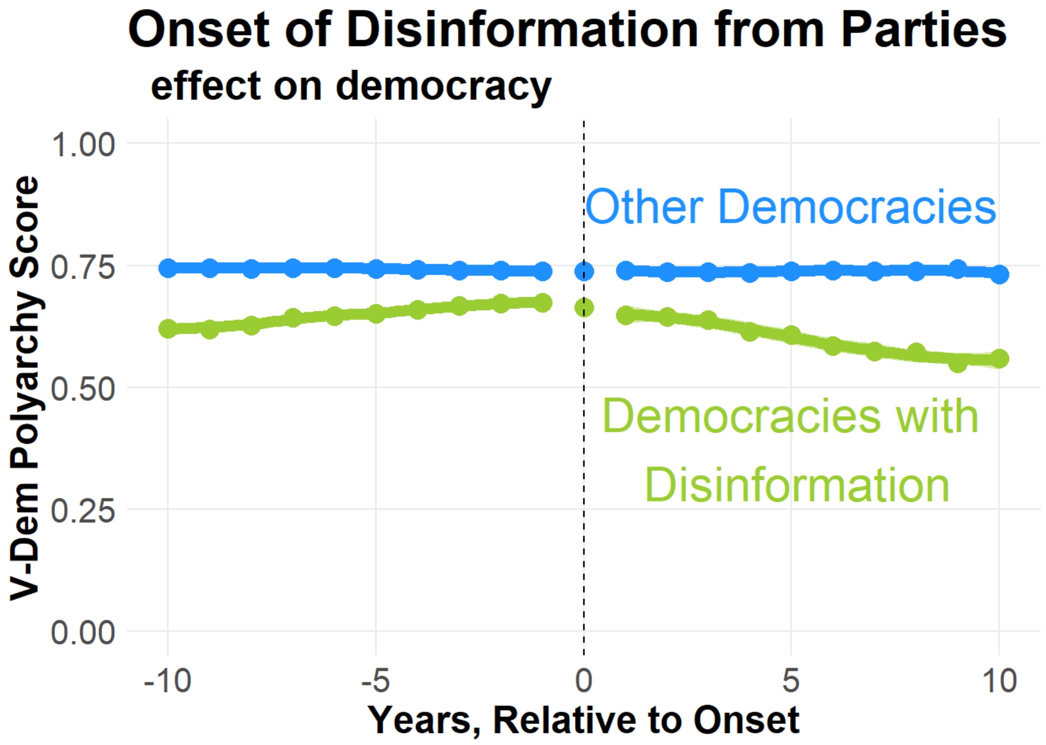
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## -10.000  -6.000  -1.000  -1.079   3.000  10.000   24851
```

```
summary(vdem$par_disinfo_control_poly)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.724   0.735   0.740   0.740   0.746   0.758   24851
```

Chart dif in dif - disinfo

```
vdem %>%
  filter(!is.na(year_rel_par_disinfo)) %>%
  group_by(year_rel_par_disinfo) %>%
  summarize(polyarchy_par_disinfo = mean(v2x_polyarchy, na.rm = TRUE),
            polyarchy_control = mean(par_disinfo_control_poly, na.rm = TRUE)) %>%
  ggplot(aes(x = year_rel_par_disinfo))+
  geom_point(aes(y = polyarchy_par_disinfo), color = disinfo_color, size = 4)+
  geom_smooth(data = . %>% filter(year_rel_par_disinfo < 0),
              method = 'loess',
              aes(y = polyarchy_par_disinfo), color = disinfo_color, size = 2.5, fill = disinfo_color)+
  geom_smooth(data = . %>% filter(year_rel_par_disinfo > 0),
              method = 'loess',
              aes(y = polyarchy_par_disinfo), color = disinfo_color, size = 2.5, fill = disinfo_color)+
  geom_point(aes(y = polyarchy_control), color = dem_color, size = 4)+
  geom_smooth(data = . %>% filter(year_rel_par_disinfo < 0),
              method = 'loess',
              aes(y = polyarchy_control), color = dem_color, size = 2.5, fill = dem_color)+
  geom_smooth(data = . %>% filter(year_rel_par_disinfo > 0),
              method = 'loess',
              aes(y = polyarchy_control), color = dem_color, size = 2.5, fill = dem_color)+
  coord_cartesian(xlim = c(-10,10), ylim = c(0,1))+
  geom_vline(xintercept = 0, linetype = 'dashed')+
  annotate('text', label = 'Democracies with\n Disinformation', x = 5, y = 0.375, color = disinfo_color, size = 8)+
  annotate('text', label = 'Other Democracies', x = 5, y = 0.875, color = dem_color, size = 8)+
  theme_minimal()+
  labs(title = 'Onset of Disinformation from Parties',
       subtitle = '  effect on democracy',
       y = 'V-Dem Polyarchy Score',
       x = 'Years, Relative to Onset')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16))
```



```
ggsave(filename = "./visuals/par_disinfo_dif.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

dif in dif - online consumption X fractionalization

Set level that counts as being “treated” with frac_inter

```
frac_inter_threshold <- as.numeric(quantile(vdem_con$smonexXsmmefra, probs = treatment_threshold, na.rm = TRUE))
```

Label years relative to onset of frac_inter treatment only count as treatment if happens while inside a democratic spell, and if first year of spell did not see high frac_inter but year count extends ten years prior and after the treatment, regardless of democratic status in those years

```
vdem$year_rel_frac_inter <- as.numeric(NA)
vdem$frac_inter_control_poly <- as.numeric(NA)
for (i in seq_along(vdem$year)){
  #skip country-years that are not in democratic spells or lack score
  if (is.na(vdem$dem_spell_name[i])) next
  if (is.na(vdem$smonexXsmmefra[i])) next

  #find earliest year within each dem spell that crosses treatment threshold.
  frac_inter_year_zero <- vdem %>% filter(dem_spell_name == vdem$dem_spell_name[i]) %>%
    filter(smonexXsmmefra >= frac_inter_threshold) %>%
    summarize(frac_inter_year_zero = min(year, na.rm = TRUE)) %>%
    pull(frac_inter_year_zero)

  #skip country-years that are in democratic spells that began with high score
  if (vdem$year[!is.na(vdem$dem_spell_name) &
                vdem$dem_spell_name == vdem$dem_spell_name[i] &
                !is.na(vdem$dem_spell_running) &
                vdem$dem_spell_running == 0] == frac_inter_year_zero) next

  #renumber infinite values as nulls
  #value is Inf if spell never crossed threshold
  frac_inter_year_zero <- if_else((frac_inter_year_zero == Inf | frac_inter_year_zero == -Inf),
                                  as.numeric(NA),
                                  frac_inter_year_zero)

  if (is.na(frac_inter_year_zero)) next

  #label year of onset as relative year zero
  vdem$year_rel_frac_inter[i] <- if_else(vdem$year[i] == frac_inter_year_zero,
                                         0,
                                         as.numeric(NA))
}
for (i in seq_along(vdem$year)){
  #locate treatment onset within 10 years of each country-year
  frac_inter_year_zero <- vdem %>%
    filter(country_name == vdem$country_name[i] &
             year >= (vdem$year[i] - 10) &
             year <= (vdem$year[i] + 10) &
             year_rel_frac_inter == 0) %>%
    summarize(frac_inter_year_zero = min(year)) %>%
    pull(frac_inter_year_zero)

  #renumber infinite values as nulls
  #value is Inf if spell never crossed threshold
  frac_inter_year_zero <- if_else((frac_inter_year_zero == Inf | frac_inter_year_zero == -Inf),
                                  as.numeric(NA),
                                  frac_inter_year_zero)

  if (is.na(frac_inter_year_zero)) next

  #label all preceding and following years in spell, relative to that zero
  vdem$year_rel_frac_inter[i] = vdem$year[i] - frac_inter_year_zero

  #log control value of frac_inter among all consolidated democracies in same absolute year
  vdem$frac_inter_control_poly[i] <- if_else(is.na(vdem$year_rel_frac_inter[i]),
                                             as.numeric(NA),
                                             mean(vdem$v2x_polyarchy[vdem$v2x_polyarchy >= dem_threshold &
                                                                       vdem$year == vdem$year[i] &
                                                                       vdem$country_name != vdem$country_name[i]],
                                                  na.rm = TRUE))
}
summary(vdem$year_rel_frac_inter)
```

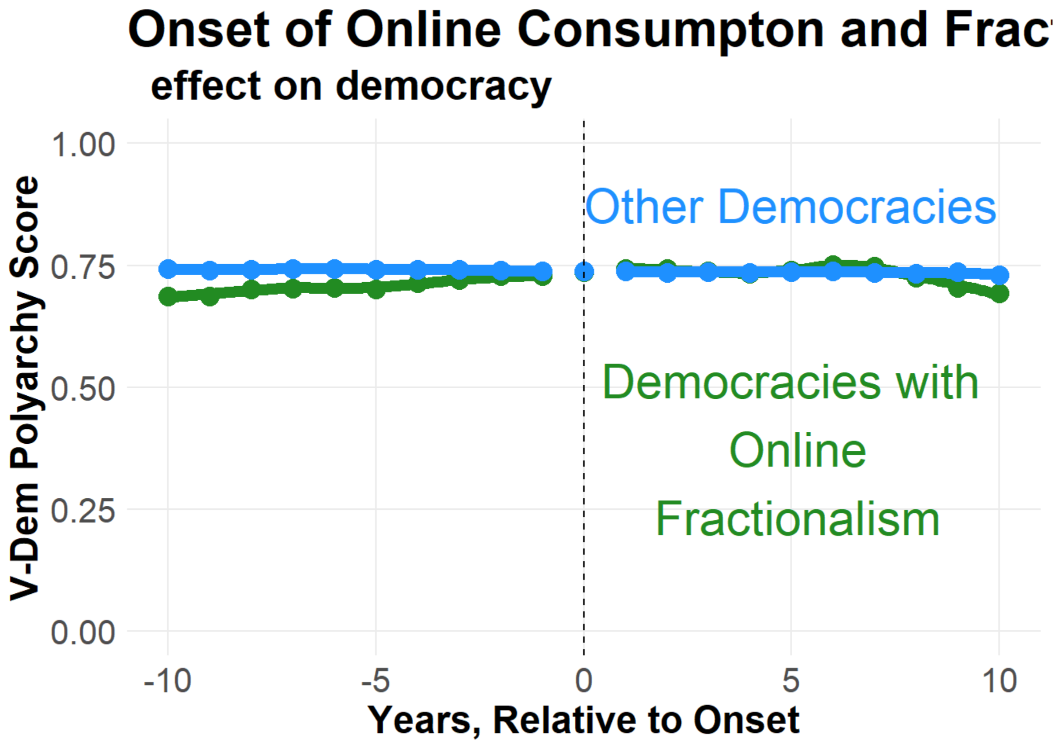
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
## -10.000  -6.000  -1.000  -1.118   3.000  10.000   25256
```

```
summary(vdem$frac_inter_control_poly)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.724   0.733   0.738   0.739   0.744   0.756   25256
```

Chart dif in dif - online consumption times fractionalization

```
vdem %>%
  filter(!is.na(year_rel_frac_inter)) %>%
  group_by(year_rel_frac_inter) %>%
  summarize(polyarchy_frac_inter = mean(v2x_polyarchy, na.rm = TRUE),
            polyarchy_control = mean(frac_inter_control_poly, na.rm = TRUE)) %>%
  ggplot(aes(x = year_rel_frac_inter))+
  geom_point(aes(y = polyarchy_frac_inter), color = media_color, size = 4)+
  geom_smooth(data = . %>% filter(year_rel_frac_inter < 0),
              method = 'loess',
              aes(y = polyarchy_frac_inter), color = media_color, size = 2.5, fill = media_color)+
  geom_smooth(data = . %>% filter(year_rel_frac_inter > 0),
              method = 'loess',
              aes(y = polyarchy_frac_inter), color = media_color, size = 2.5, fill = media_color)+
  geom_point(aes(y = polyarchy_control), color = dem_color, size = 4)+
  geom_smooth(data = . %>% filter(year_rel_frac_inter < 0),
              method = 'loess',
              aes(y = polyarchy_control), color = dem_color, size = 2.5, fill = dem_color)+
  geom_smooth(data = . %>% filter(year_rel_frac_inter > 0),
              method = 'loess',
              aes(y = polyarchy_control), color = dem_color, size = 2.5, fill = dem_color)+
  coord_cartesian(xlim = c(-10,10), ylim = c(0,1))+
  geom_vline(xintercept = 0, linetype = 'dashed')+
  annotate('text', label = 'Democracies with\n Online\n Fractionalism', x = 5, y = 0.375, color = media_color, size = 8)+
  annotate('text', label = 'Other Democracies', x = 5, y = 0.875, color = dem_color, size = 8)+
  theme_minimal()+
  labs(title = 'Onset of Online Consumpton and Fractionalism',
       subtitle = '  effect on democracy',
       y = 'V-Dem Polyarchy Score',
       x = 'Years, Relative to Onset')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18, face = 'bold'),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16))
```



```
ggsave(filename = "./visuals/media_dif.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

Charts for country case studies

Check GDP per capita to ensure comparability

```
vdem %>% filter(year == 1995,
                country_name %in% c('Venezuela', 'Brazil', 'Uruguay', 'Hungary', 'Poland', 'Estonia')) %>%
  select(country_name, e_migdppc)
```

```
## # A tibble: 6 x 2
##   country_name e_migdppc
##   <chr>            <dbl>
## 1 Poland            9.41
## 2 Brazil            8.95
## 3 Venezuela        14.4
## 4 Uruguay          12.1
## 5 Estonia          12.9
## 6 Hungary          10.1
```

Write generic code for country plots

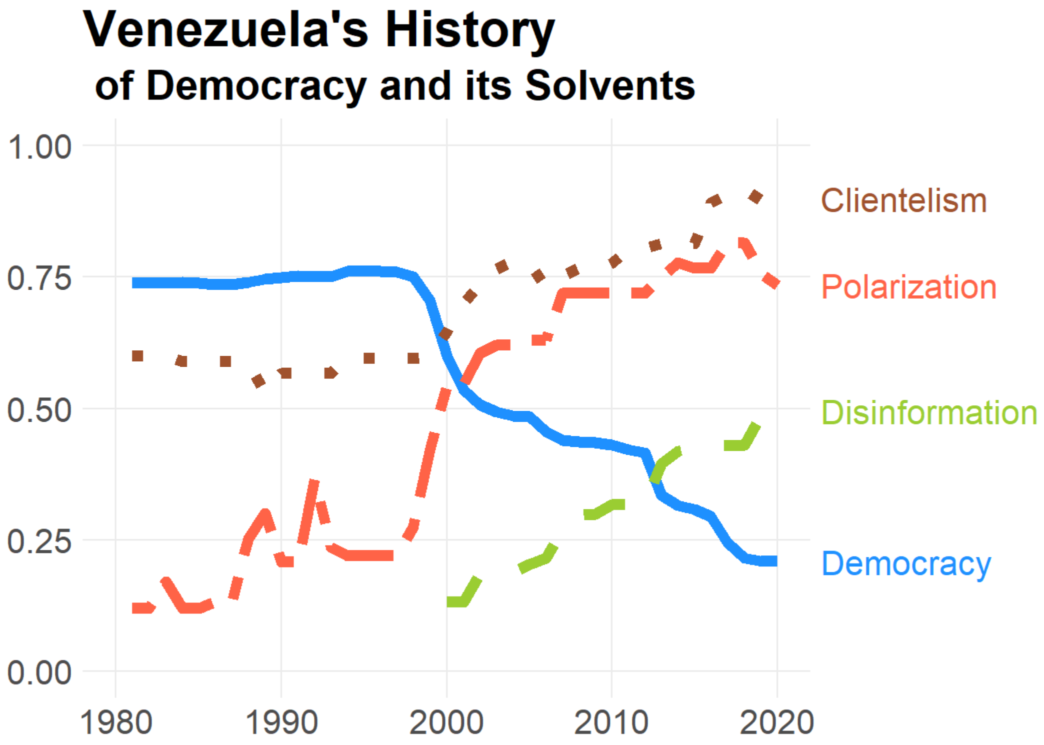
```
plot_case <- function(country_case){
  max_year <- max(vdem$year, na.rm = TRUE)
  case_labels_df <- data.frame(variable = c('Democracy', 'Clientelism', 'Polarization', 'Disinformation'),
                               last_value = c(vdem$v2x_polyarchy[vdem$year == max_year &
                                                                   vdem$country_name == country_case],
                                              vdem$v2xnp_client[vdem$year == max_year&
                                                                  vdem$country_name == country_case],
                                              vdem$v2cacamps[vdem$year == max_year&
                                                               vdem$country_name == country_case],
                                              vdem$v2smonex[vdem$year == max_year&
                                                              vdem$country_name == country_case] *
                                                vdem$v2smpardom[vdem$year == max_year&
                                                                  vdem$country_name == country_case]),
                               color = c(dem_color, client_color, polar_color, disinfo_color))

  ggplot(data = (vdem %>% filter(country_name == country_case &
                                   year >= 1981)),
         aes(x = year))+
    geom_line(aes(y = v2x_polyarchy), color = dem_color, size = 2.5)+
    geom_line(aes(y = v2xnp_client), color = client_color, size = 2.5, linetype = 'dotted')+
    geom_line(aes(y = v2cacamps), color = polar_color, size = 2.5, linetype = 'twodash')+
    geom_line(aes(y = (v2smonex * v2smpardom)), color = disinfo_color, size = 2.5, linetype = 'dashed')+
    coord_cartesian(xlim = c(1980, 2020), ylim = c(0,1))+
    scale_y_continuous(sec.axis = dup_axis(
      breaks = case_labels_df$last_value,
      labels = case_labels_df$variable))+
    theme_minimal()+
    labs(title = paste0(country_case, "'s History"),
         subtitle  = " of Democracy and its Solvents")+
    theme(title = element_text(size = 20, face = 'bold'),
          axis.title = element_blank(),
          panel.grid.minor = element_blank(),
          axis.text = element_text(size = 16),
          axis.text.y.right = element_text(size = 16, color = c(dem_color, client_color, polar_color, disinfo_color)))
}
```

Apply to desired cases

```
plot_case(country_case = 'Venezuela')
```

```
## Warning: Vectorized input to `element_text()` is not officially supported.
## Results may be unexpected or may change in future versions of ggplot2.
```

```
## Warning: Removed 19 row(s) containing missing values (geom_path).
```



```
ggsave(filename = "./visuals/case_venezuela.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

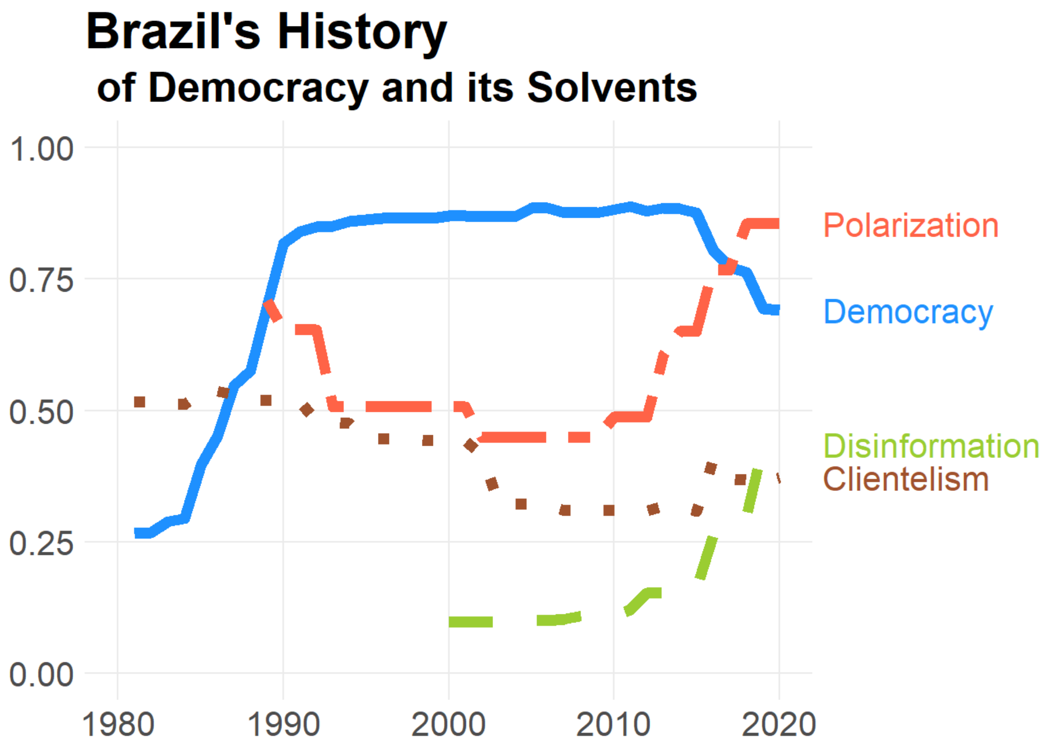
```
## Warning: Removed 19 row(s) containing missing values (geom_path).
```

```
plot_case(country_case = 'Brazil')
```

```
## Warning: Vectorized input to `element_text()` is not officially supported.
## Results may be unexpected or may change in future versions of ggplot2.
```

```
## Warning: Removed 8 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 19 row(s) containing missing values (geom_path).
```



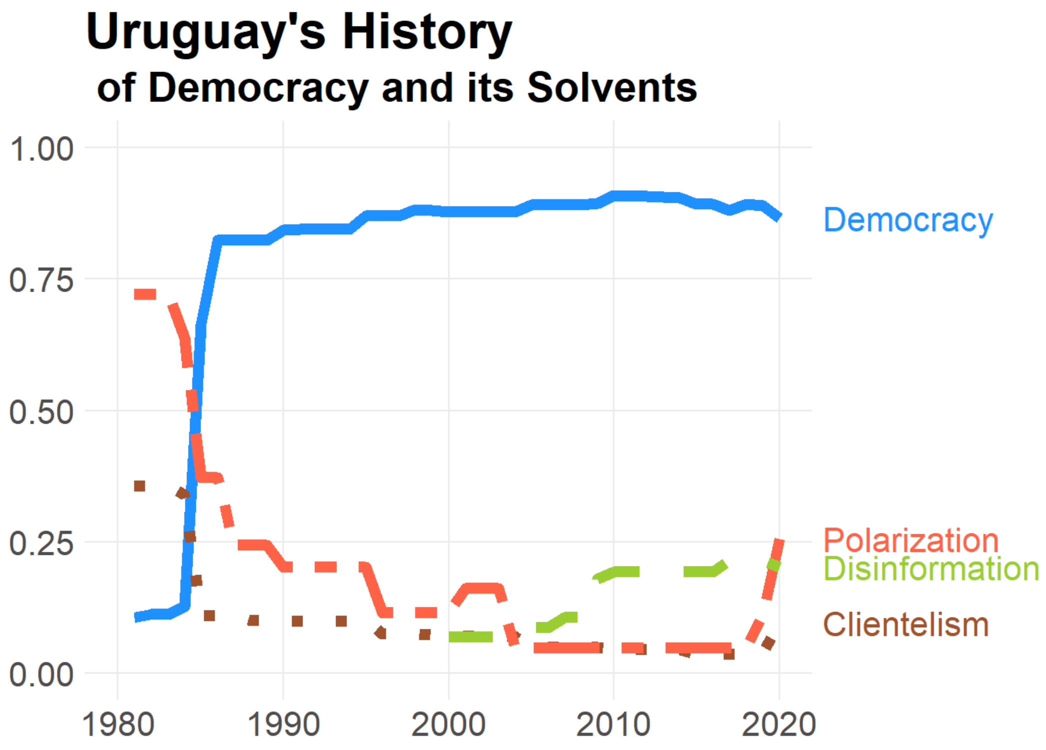
```
ggsave(filename = "./visuals/case_brazil.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

```
## Warning: Removed 8 row(s) containing missing values (geom_path).

## Warning: Removed 19 row(s) containing missing values (geom_path).
```

```
plot_case(country_case = 'Uruguay')
```

```
## Warning: Vectorized input to `element_text()` is not officially supported.
## Results may be unexpected or may change in future versions of ggplot2.
```

```
## Warning: Removed 19 row(s) containing missing values (geom_path).
```



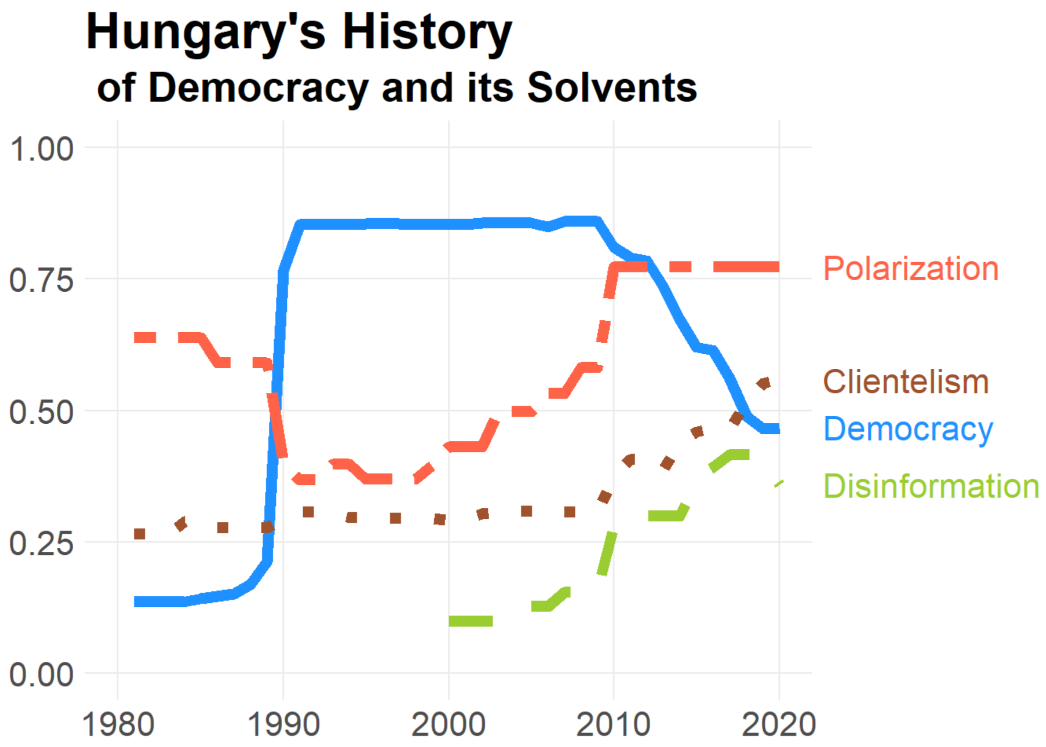
```
ggsave(filename = "./visuals/case_uruguay.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

```
## Warning: Removed 19 row(s) containing missing values (geom_path).
```

```
plot_case(country_case = 'Hungary')
```

```
## Warning: Vectorized input to `element_text()` is not officially supported.
## Results may be unexpected or may change in future versions of ggplot2.
```

```
## Warning: Removed 19 row(s) containing missing values (geom_path).
```



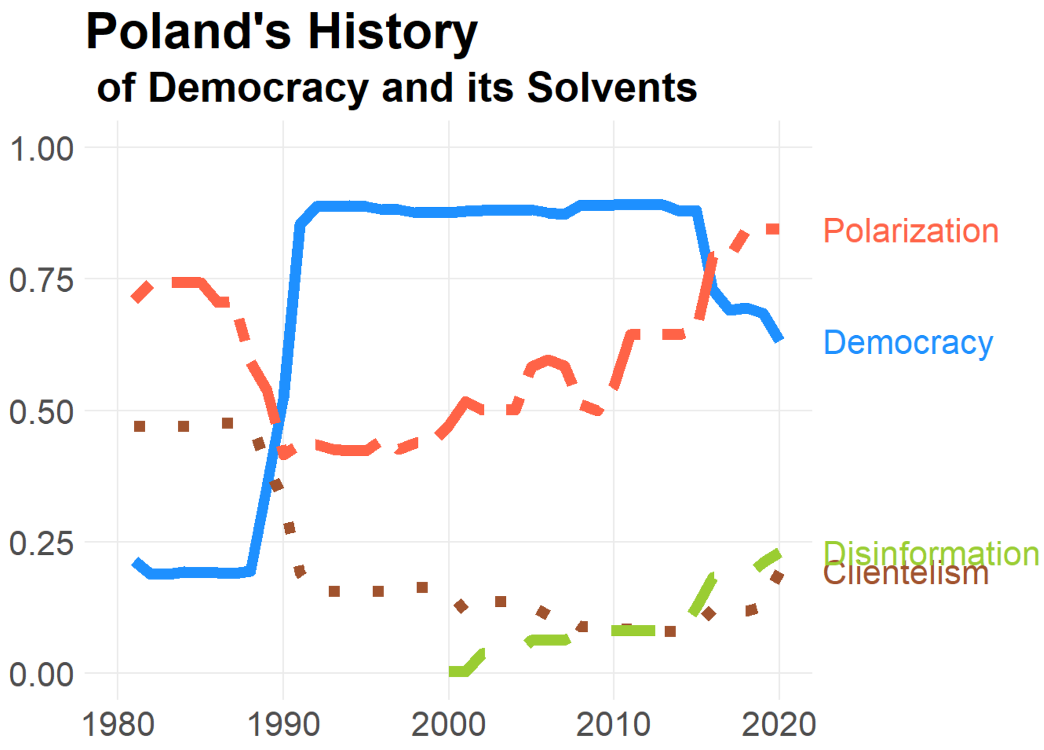
```
ggsave(filename = "./visuals/case_hungary.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

```
## Warning: Removed 19 row(s) containing missing values (geom_path).
```

```
plot_case(country_case = 'Poland')
```

```
## Warning: Vectorized input to `element_text()` is not officially supported.
## Results may be unexpected or may change in future versions of ggplot2.
```

```
## Warning: Removed 19 row(s) containing missing values (geom_path).
```



```
ggsave(filename = "./visuals/case_poland.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

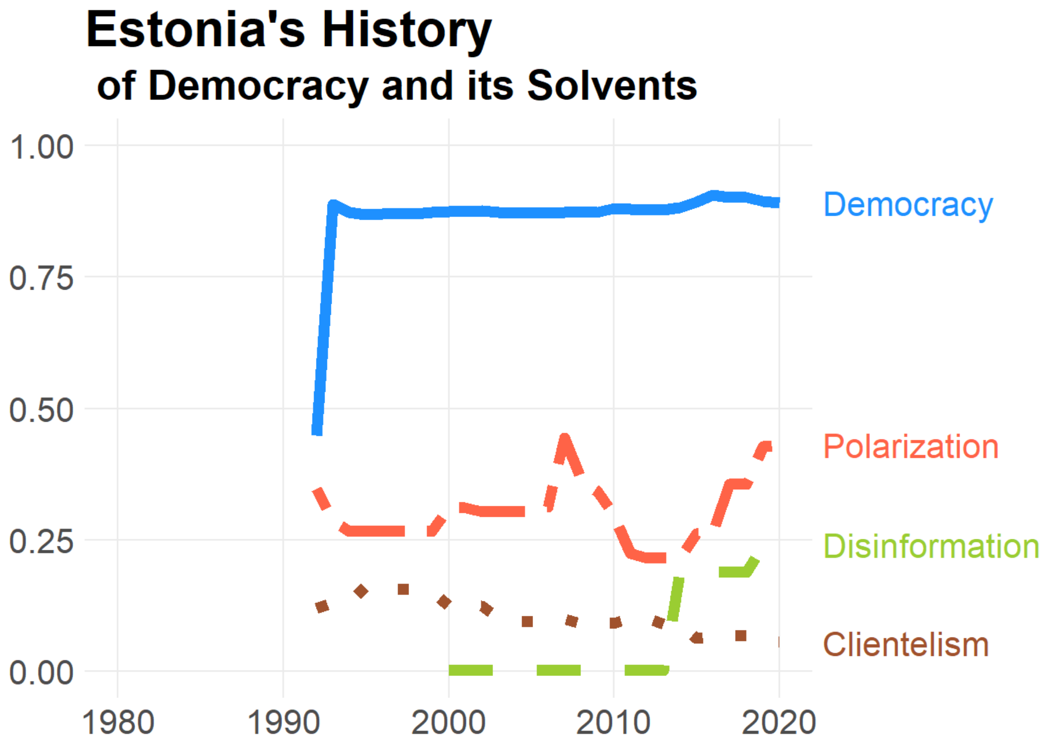
```
## Warning: Removed 19 row(s) containing missing values (geom_path).
```

```
plot_case(country_case = 'Estonia')+
  scale_x_continuous(limits = c(1992, 2020))
```

```
## Warning: Vectorized input to `element_text()` is not officially supported.
## Results may be unexpected or may change in future versions of ggplot2.
```

```
## Warning: Removed 2 row(s) containing missing values (geom_path).

## Warning: Removed 2 row(s) containing missing values (geom_path).

## Warning: Removed 2 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 10 row(s) containing missing values (geom_path).
```



```
ggsave(filename = "./visuals/case_estonia.jpg",
       width = 10,
       height = 6,
       units = 'in')
```

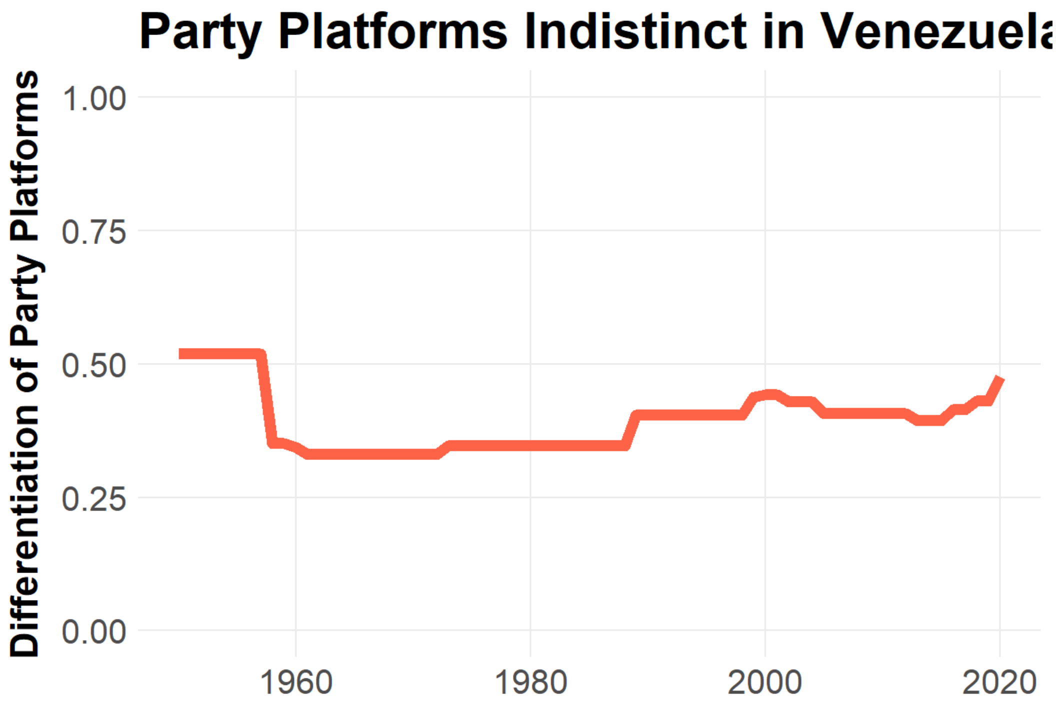
```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 2 row(s) containing missing values (geom_path).

## Warning: Removed 2 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 10 row(s) containing missing values (geom_path).
```

Verify claim about low distinctiveness of party platforms in Brazil

```
vdem %>% filter(country_name == 'Venezuela', year >= 1950) %>%
  ggplot(aes(x = year, y = v2psplats))+
  geom_line(color = polar_color, size = 2.5)+
  theme_minimal()+
  coord_cartesian(ylim = c(0,1))+
  labs(title = 'Party Platforms Indistinct in Venezuela',
       y = 'Differentiation of Party Platforms',
       x = '')+
  theme(title = element_text(size = 20, face = 'bold'),
        axis.title = element_text(size = 18),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 16),
        axis.title.y = element_text(margin = margin(r = 8)))
```


Party Platforms Indistinct in Venezuela
Differentiation of Party Platforms
1.00
0.75
0.50
0.25
0.00
1960
1980
2000
2020