Replication of Satyanath, Voigtländer, and Voth (2017)

```
packages = c("dplyr", "readr", "haven", "estimatr",
             "conleyreg", "lmtest", "sandwich", "tibble",
             "quantreg")
sapply(packages, library, character.only = TRUE)
```

Load in the data:

```
nazi_data = read_dta("../datasets/4-satyanath-voigtlander-voth/Dataset_Bowling_Replication_J
```

Generate relevant variables:

```
nazi_int = nazi_data |>
  mutate(lnNSentry_total = log(1 + NSentry_total), # log entry rates
         lnNSentry_FU_total = log(1 + NSentry_FU_total),
         lnclubs_all = log(clubs_all),
         pop2 = (pop25 / 1000)^2, # population
         pop3 = (pop25 / 1000)^3,
         lnpop25 = log(pop25),
         pop25_quintiles = ntile(pop25, 5), # size quintile dummy
         lnpop_density = log(pop_density), # population density
         dummy_maps = as.integer(area_source == "Maps"),
         i_popden_maps = lnpop_density * dummy_maps)
```

The following code reproduces Column 1 of Panel B of Table 4. The first equation follows Satyanath et al., the second introduces share Protestant as an additional control. The final equation reproduces Column 3 of the same panel as a median regression.

```r
landweimar = nazi_int |> pull(landweimar)
mod1 = lm(pcNSentry_std ~ clubs_all_pc + lnpop25 + share_cath25 + bcollar25,
          data = nazi_int)
rob1 = coeftest(mod1, vcov = vcovCL(mod1, cluster = landweimar))[, "Std. Error"]

mod2 = lm(pcNSentry_std ~ clubs_all_pc + lnpop25 + share_cath25 + bcollar25 +
            share_prot25,
          data = nazi_int)
rob2 = coeftest(mod2, vcov = vcovCL(mod2, cluster = landweimar))[, "Std. Error"]

mod3 = rq(pcNSentry_std ~ clubs_all_pc + lnpop25 + share_cath25 + bcollar25,
          data = nazi_int)
summary(mod3, se = "boot", R = 1000)
```

```
Call: rq(formula = pcNSentry_std ~ clubs_all_pc + lnpop25 + share_cath25 +
    bcollar25, data = nazi_int)

tau: [1] 0.5

Coefficients:
             Value    Std. Error t value  Pr(>|t|)
(Intercept)  -0.83010  0.72666   -1.14234  0.25454
clubs_all_pc  0.07130  0.05635    1.26550  0.20702
lnpop25       0.16657  0.05813    2.86522  0.00457
share_cath25 -0.89615  0.18599   -4.81839  0.00000
bcollar25    -2.27175  0.41413   -5.48564  0.00000
```

Extract first principal component of government stability measures. I recoded the value representing the percentage of time a state was governed by at least one party from the 'Weimar coalition' (SPD, DDP, Zentrum) from 0 to 1 for Bavaria to reflect the agreement between the BVP and Zentrum.

```r
pca_data = nazi_int |>
  dplyr::select(landweimar, govt_longest_perc,
                party_longest_perc,
                weimar_coalition_perc) |>
  distinct() |>
  filter(!landweimar %in% c("Lübeck", "Saarland", "Bremen")) |>
  mutate(weimar_coalition_perc = case_when(
    landweimar == "Bayern" ~ 1,
```

```
    TRUE ~ weimar_coalition_perc
  )) |>
  column_to_rownames(var = "landweimar")

stability = princomp(pca_data, cor = TRUE) # extract first principal component
govt_stability = as.data.frame(stability$scores) |>
  rownames_to_column(var = "landweimar") |>
  dplyr::select(-Comp.2, -Comp.3) |>
  arrange(desc(Comp.1)) |>
  rename(pc1 = Comp.1)
#  mutate(stability1 = as.integer(PC1 > median(PC1)), # make non-Prussia median unstable
#         stability2 = as.integer(PC1 > -0.5)) # make it stable

nazi_pca = left_join(nazi_int, govt_stability, by = "landweimar") |>
  filter(!landweimar %in% c("Lübeck", "Saarland", "Bremen", "Preußen")) |>
  mutate(stability1 = as.integer(pc1 >= median(pc1)),
         stability2 = as.integer(pc1 > median(pc1)))
```

The following regressions replicate Columns 3 and 4 of Table 7. I report two possible specifications: in the first two regressions, the non-Prussian state with the median principal component value is coded as stable, while in the next two, it is coded as unstable.

```
#Median state stable
mod3 = lm(pcNSentry_PRS_std ~ clubs_all_pc + share_cath25 + lnpop25 + bcollar25,
                  data = nazi_pca |> filter(stability1 == 0))
rob3 = coeftest(mod3, vcov = vcovHC(mod3, type = "HC1"))[, "Std. Error"]

mod4 = lm(pcNSentry_PRS_std ~ clubs_all_pc + share_cath25 + lnpop25 + bcollar25,
                  data = nazi_pca |> filter(stability1 == 1))
rob4 = coeftest(mod4, vcov = vcovHC(mod4, type = "HC1"))[, "Std. Error"]

#Median state unstable
mod5 = lm(pcNSentry_PRS_std ~ clubs_all_pc + share_cath25 + lnpop25 + bcollar25,
                  data = nazi_pca |> filter(stability2 == 0))
rob5 = coeftest(mod5, vcov = vcovHC(mod5, type = "HC1"))[, "Std. Error"]

mod6 = lm(pcNSentry_PRS_std ~ clubs_all_pc + share_cath25 + lnpop25 + bcollar25,
                  data = nazi_pca |> filter(stability2 == 1))
rob6 = coeftest(mod6, vcov = vcovHC(mod6, type = "HC1"))[, "Std. Error"]
```

The following code replicates Column 5 of Table 7. Instead of treating pc1 as a binary indicator, I keep it as a continuous variable and report the results.

```r
nazi_pca1 = nazi_pca |>
  mutate(prussia = landweimar == "Preußen")

mod4 = lm(pcNSentry_PRS_std ~ pc1*(clubs_all_pc + share_cath25 +
            lnpop25 + bcollar25 + prussia),
          data = nazi_pca1)
coeftest(mod4, vcov = vcovHC(mod2, type = "HC1"))
```

```
t test of coefficients:

              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.894277   1.006652 -0.8884  0.37656
clubs_all_pc  0.176490   0.053066  3.3259  0.00125 **
share_cath25 -1.129235   0.548408 -2.0591  0.04219 *
lnpop25       0.143411   0.057288  2.5033  0.01399 *
bcollar25    -0.652492   0.482408 -1.3526  0.17937
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```