Constructing the analysis dataset

Seth Warner,* 8-14-2025

* ChatGPT 5 was used to help prepare parts of Sections 1a, 1b, 2a, and 2b. All text-generated information was reviewed and verified by the author.

The analysis.RData file in this repository is a bespoke dataset that can be used to reproduce the tables and figures in the text of "Whose Party is This?" Its observations trace back to publicly available survey data (specifically, the ANES and CES) and its contextual variables on party ideology are assembled from additional sources.

This document provides instructions on how the dataset can be reconstructed for researchers wishing to replicate analysis or study a different question using some of the same data.

Contents

1. Survey data
    a. ANES
    b. CES
    c. Operational ideology
2. Party ideology data
    a. CFscores
    b. DW-NOMINATE
    c. Americans for Democratic Action
    d. NPAT (state legislators)
    e. Party-in-electorate
    f. Standardization
3. Final steps

1. Survey data

In the analysis dataset, each observation ties to a survey response to the American National Election Study (ANES) or Cooperative Election Study (CES). In addition, 15 of the variables are also drawn from these survey datasets. They are:

> year, state, case_id1, R_party, dem_place, gop_place, self_place, weight, nonwhite, bach, male, over50, st_knowledge, and ideology_score

The sections below indicate how to create each of these variables from their original datasets, with separate sections for knowledge and operational ideology, which involved more complex calculations.

a. ANES

Working with the ANES was the simpler of the two survey datasets, as all relevant variables were included in the cumulative file. Here are how they were converted to the variables in my analysis datset.

year

Set to `VCF0004` (survey year) with no recode beyond what's in the file.

state

Set to `VCF0901b` (state identifier) with no additional recode.

case_id1

String concatenation of year and respondent ID: `paste(year, VCF0006a, sep = "")`, with "anes " as its prefix.

R_party (respondent's party identification)

From `VCF0301`:

If `VCF0301 < 4` → `"Democrat"`

If `VCF0301 > 4` → `"Republican"`

Else → `"Independent"`

If `VCF0301 == 0` → `NA` (missing)

dem_place (respondent's placement of Democrats)

From `VCF0503`; set to `NA` if value is `0` or `8`; otherwise keep the numeric value.

gop_place (respondent's placement of Republicans)

From `VCF0504`; set to `NA` if value is `0` or `8`; otherwise keep the numeric value.

self_place (respondent's self-placement)

From `VCF0803`; if `9` → recoded to midpoint `4`; if `0` → `NA`; otherwise keep the numeric value.

weight

Set to `VCF0009z`.

nonwhite

From `VCF0105a` (race): if `9` → `NA`; else indicator `VCF0105a > 1` (TRUE/FALSE).

bach (bachelor's degree indicator)

From `VCF0110`; equals `1` if `VCF0110 == 4`, else `0`.

male

From `VCF0104`; equals `1` if `VCF0104 == 1`, else `0`.

over50

From `VCF0101` (age); equals `1` if `VCF0101 > 50`, else `0`.

st_knowledge

Recode VCF9260, VCF9261, and VCF9262 so that values of -9 are NA, and find the number of variables that == 1 (this indicates correct answer). Then, grouping by year, apply the scale command to standardize that value so that st_knowledge is the z-score of (# correct | year).

b. CES

Assembling data from the Cooperative Election Study (CES) was a little more complicated, as the cumulative file did not contain respondents' ideological placements of themselves and the parties, or their responses to political knowledge questions.

First, I harmonized variable names and coding from the 2010, 2014, and 2018 waves of the CES according to the table below.

Harmonized variable	**2018 CES variable**	**2014 CES variable**	**2010 CES variable**	**Recode rule**
self_place	CC18_334A	CC334A	CC334A	Values > 7 → NA
gop_place	CC18_334E	CC334L	CC334E	Values > 7 → NA
dem_place	CC18_334D	CC334K	CC334D	Values > 7 → NA

Harmonized variable	2018 CES variable	2014 CES variable	2010 CES variable	Recode rule
case_id1	caseid	V101	V100	Prefix w/ “ces ”
house_party	CC18_309a	CC14_309a	CC309a	Values > 2 → NA
sen_party	CC18_309b	CC14_309b	CC309b	Values > 2 → NA
stsen_party	CC18_309c	CC14_309c	CC309c	Values > 2 → NA
sthou_party	CC18_309d	CC14_309d	CC309d	Values > 2 → NA

Then, using the original caseid variable, I merged the harmonized variables into the CES cumulative file, which allowed me to more centrally create the other variables that appear in the analysis dataset.

R_party (respondent’s party identification)

From `pid7`:

If `pid7 < 4` → `"Democrat"`

If `pid7 > 4` and `< 8` → `"Republican"`

Else → `"Independent"`

nonwhite

From `race`: coded as `TRUE` if `race != 1` (i.e., not White), else `FALSE`.

bach (bachelor’s degree indicator)

From `educ`: `1` if `educ >= 5` (college degree or higher), else `0`.

over50

From `age`: `1` if `age > 50`, else `0`.

male

From `gender`: `1` if `gender == 1` (male), else `0`.

state

From numeric state FIPS codes in `state`: converted to USPS two-letter postal abbreviations using the `fips()` function from the **cdlTools** package (`fips(state, to = "Abbreviation")`).

weight

Keep the CES variable called "weight" unaltered

st_knowledge

CES does not give indicator for whether response was correct, but I found that the majority was correct at federal level for all years, and for a few dozen state-years that I spot checked. (The "not sure" option, chosen by 20-40% per question, appears to ensure a high aggregate score.) As such, I create a function that identifies the majority value for a variable (not sure is coded NA), and I group first by year to count "correct" answers at the federal level, and then by state-year to do so at the state level. I calculate the number correct per respondent, and group again within year to standardize the number, so that **st_knowledge** is the z-score for (# correct | year).

c. Operational ideology

The ideology_score variable is not used directly in the analysis but is used in the next section to calculate the party-in-electorate ideology by year. As described in the text, I use an IRT-based scaling of policy views to place respondents on a left-right ideology scale. The steps involved look like this:

i. For the ANES, I identified the following variables in the cumulative file as describing the respondents' policy views:

VCF0805, VCF0806, VCF0808, VCF0809, VCF0839, VCF9038, VCF0814, VCF0815, VCF0816, VCF0817, VCF0818, VCF0819, VCF0830, VCF0867, VCF9037, VCF9230, VCF9231, VCF0823, VCF0828, VCF0841, VCF0843, VCF0844, VCF9232, VCF9233, VCF0832, VCF0833, VCF0837, VCF0838, VCF0876, VCF0877, VCF0878, VCF0879, VCF9043, VCF9051, VCF9236, VCF9237, VCF9238, VCF0842, VCF0886, VCF0887, VCF0888, VCF0889, VCF0890, VCF0891, VCF0892, VCF0893, VCF0894, VCF9046, VCF9047, VCF9048, VCF9049, VCF9050, VCF9117, VCF9118, VCF9119, VCF9120, VCF9121

After checking the coding structure, I cleaned them by recoding values of 0, 9, or negative numbers to NA, and then used the mirt command in R to create a IRT model that explains their dimensionality. Next, I use the fscores command to extract their first factor, and save these as **ideology_score** in the dataset.

ii. For the CES, I download the policy cumulative file, which only includes the case IDs and responses to all policy questions have been asked in the survey's history. Several of these are not coded in a left-right fashion, while others are scaled to be much wider than typical multi-option questions are. I recode these as follows:

In enviro_vs_jobs, answer code 6 was recoded to 3.
In guns_scale, answer code 2 was recoded to 5.
In gaymarriage_scale, 5 was recoded to NA.
In incometax_vs_salestax, I recoded all values above 50 to 1, and otherwise 0.
In spending_vs_tax, I recoded all values above 50 to 1, and otherwise 0.
I removed the spending_cuts_most and spending_cuts_least variables.

From there, I applied the mirt and fscores commands to the dataset as described above to create **ideology_score**.

iii. Lastly, because the ANES and CES policy variables were asked on different issues, and in different years, I had to adjust for the fact that the resulting ideology scores were not directly comparable. To limit the degree of slippage, I used a reference distribution technique to match CES scores by decile to the distribution from the ANES fielded two years prior. To keep you from guessing, here is the function I used.

```
adjust_ideology_score <- function(ces_df, anes_df, ces_year, anes_year) {

  # Filter the dataframes for the specified years
  ces_subset <- ces_df %>% filter(year == ces_year)
 anes_subset <- anes_df %>% filter(year == anes_year)

  # Calculate deciles for anes ideology_score
  anes_deciles <- quantile(anes_subset$ideology_score, probs = seq(0, 1,
0.1), na.rm = T)

   # Map ces ideology_score to the deciles of anes ideology_score
ces_subset$ideology_score <- cut(ces_subset$ideology_score, breaks =
quantile(ces_subset$ideology_score, probs = seq(0, 1, 0.1), na.rm = T),
labels = FALSE)



  # Assign new ideology_score based on anes deciles
  ces_subset$ideology_score <- anes_deciles[ces_subset$ideology_score +
1]

  return(ces_subset)
}
```

```
# Apply the function to the relevant years
ces_2010 <- adjust_ideology_score(ces, anes, 2010, 2008)
ces_2014 <- adjust_ideology_score(ces, anes, 2014, 2012)
ces_2018 <- adjust_ideology_score(ces, anes, 2018, 2016)
```

2. Party ideology data

Working with the party ideology data was considerably less immersive, as the data were largely pre-packaged and I only needed to create one or two variables from them each.

The first five sections below indicate how I create the non-standardized variables by party and year in the analysis dataset: fed_cf, nominate, ada, state_cf, and state_npat, ntl_symbolic, ntl_operational, state_symbolic, state_operational. Then, the last section discusses how I standardize them "across" and "within" parties.

a. CFscores

Filter the recipient dataset: Keep only rows where **gwinner** == 'W', **seat** is one of "federal:house", "federal:senate", "state:lower", or "state:upper", and **party** is either '100' (Democrat) or '200' (Republican). Then recode party to "Democrat" if **party** == '100' and "Republican" if **party** == '200'.

Create fed_cf: From the filtered data, keep rows where **seat** contains 'federal', then group by **cycle** and **party**, and compute fed_cf = median(**recipient.cfscore**, na.rm = TRUE).

Create state_cf: From the filtered data, keep rows where **seat** contains 'state', then group by **cycle**, **state**, and **party**, and compute state_cf = median(**recipient.cfscore**, na.rm = TRUE).

b. DW-NOMINATE

Using the csv produced by the default settings on https://voteview.com/data, **create a `year` variable** by setting **`year`** = `1788 + 2 * congress`, where **`congress`** is the congressional session number.

Filter the dataset to keep only rows where **`year`** ≥ 1990 and **`party_code`** is either `100` (Democrat) or `200` (Republican).

Create `party` so that **`party`** = `'Democrat'` if **`party_code`** == 100, and **`party`** = `'Republican'` if **`party_code`** == 200.

Group by `year` and `party`.

Calculate the median NOMINATE score within each group: **`nominate`** = median(**`nominate_dim1`**, na.rm = TRUE).

c. Americans from Democratic Action

These data were collected manually for several past years from the ADA website, so use the ada_party_averages.csv that I posted to the repository for direct replication. Each row is a party-year. To make the House-to-Senate proportions similar to other measures, create a new variable **ada** that is (4*house + senate)/5. Then take 100 – ada to create a measure where higher values are more conservative. Finally, group by party and year to create the ada variable in the analysis dataset.

d. NPAT state legislative

Define the analysis years by creating a vector of even-numbered years from 1994 to 2020, naming it **even_years.**

Create the following helper function to identify legislators who served in a given year:

```
served_in_year <- function(data, year) {
  cols <- grepl(paste(year), names(data))
 rowSums(!is.na(data[, cols])) > 0
}
```

Loop over each year in **even_years**:

Filter npat to legislators where served_in_year(npat, year) is TRUE.
Group by **st** (state) and **party**.
Use **summarize** to create one row per state–party combination for that year with:
year = the loop year.
state_npat = median(np_score, na.rm = TRUE).

e. Party-in-electorate

Take the survey files as prepared in the last section, and bind them together using shared variables year, state, case_id1, R_party, dem_place, gop_place, self_place, weight, nonwhite, bach, male, over50, st_knowledge, and ideology_score.

First, grouping by **year** and **R_party**, collect the mean **self_place** as the **ntl_symbolic** value for that party-year, and the median **ideology_score** as **ntl_operational**.

Then, do the same thing but grouping by state as well to create **state_symbolic** and **state_operational**.

When taking averages, be sure to incorporate survey weights as well.

f. Standardization

The main text differentiated ideology across parties, on the standard left-right scale, versus ideology within parties, based on how each party has moved over time.

Standardizing across parties is relatively straightforward. For all party ideology variables created in sub-sections (a) through (e), apply the scale command and prefix the new variable with “std_”

Within-party standardization uses the same approach, but requires that the data be grouped by party first. The resulting variable should be prefixed with “std_party_”. Here is an example:

```
state_ideology <- state_ideology %>%
  group_by(party, state) %>%
  mutate(across(c(state_symbolic, state_operational, state_npat, state_cf),
           ~ scale(.)[,1], .names = "std_party_{col}")) %>%
  ungroup()
```

3. Final steps

The final steps to creating the analysis dataset are to:

Append the ANES and CES datasets to one another based on shared variables:

year, state, case_id1, R_party, dem_place, gop_place, self_place, weight, nonwhite, bach, male, over50, st_knowledge, and ideology_score.

Duplicate the dataset.

In one dataset, create the variable **party** and set it to “Democrat”. In the other, do the same and set it to “Republican”.

Append the datasets so that the new one refers to each ANES and CES respondent twice; one observation will be used for their placement of the Republicans and the other for their placement of the Democrats.

Create new variable **party_place**, which is **dem_place** if party == ‘Democrat’ and **gop_place** if party == ‘Republican’.

Create new variable **same_party**, which is 1 if **party == R_party** and 0 otherwise.

Import the party ideology variables created in the previous section, merging them based on variables **party, year,** and (as necessary) **state** on a many-to-one basis.

Create an **adjusted_weight** variable that proportionally adjusts the respondent’s survey weight down/upward based on the sample available for that year, so that each year’s sample has the same cumulative “voice” in the results. Here is how I did it:

```
desired_total_weight <- 1000  # or any other constant value

# Adjust the weights in the dataset
analysis <- analysis %>%
  group_by(year) %>%
  mutate(
    total_weight = sum(weight, na.rm = TRUE),
    adjusted_weight = weight / total_weight * desired_total_weight
  ) %>%
  ungroup()
```