Data and Code Availability in Political Science Publications from 1995 to 2022

One-Click Reproduction

You can reproduce our results by interactively running `reproduction.R` or running `source("reproduction.R")` in the R console. This file makes sure that all packages are updated and installed and that brms is working properly.

If you have trouble getting rstan to work, see here for troubleshooting advice.

Reproduction tl;dr

The code below reproduces all analysis in the paper. The entire analysis takes about 30 minutes to complete.

```
# remove all generated files
unlink("output/*")
unlink("figs/*")

# run each component of the analysis
source("R/01-get-data.R")
source("R/02-clean-ranks-and-apsa-data.R")
source("R/03-create-model-data.R")
source("R/04-fit-models-and-make-figures.R")
```

Hand-Coded Data

Our hand-coded data categorizing the availability of data and code supporting a random sample of journal articles in political science is available as a publicly shared Google Sheet.

The script `R/01-get-data.R` automatically grabs the latest version of this spreadsheet from the web and saves it as `output/exists.csv`.

Files

Directory Structure

```
data  // all local data, our hand-coded data (exists.csv) are grabbed from web and
  |--- apsa-journals.xlsx
  |--- scimagojr 2022.csv
  |--- ssci-population.csv
figs  // all generated figures
  |--- fig01-all.png
  |--- fig02-apsa.png
  |--- fig03-sjr.png
  |--- fig04-all-cats.png
output // all generated files, except figures and README
  |--- exists.csv
  |--- journals.csv
  |--- model-data.csv
  |--- policies.csv
  |--- ranks.csv
  |--- stan-fit1.rds
  |--- stan-fit2.rds
  |--- stan-fit3.rds
  |--- stan-fit4.rds
  |--- works.csv
R
  |--- 01-get-data.R
  |--- 02-clean-ranks-and-apsa-data.R
  |--- 03-create-model-data.R
  |--- 04-fit-models-and-make-figures.R
exists.Rproj
README.md // generated from README.qmd
README.qmd
```

Session Information

```
  sessioninfo::session_info(info = "platform")
```

```
- Session info -------------------------------------------------------------
 setting  value
 version  R version 4.4.1 (2024-06-14)
 os       macOS Ventura 13.4
 system   aarch64, darwin20
 ui       X11
 language (EN)
 collate  en_US.UTF-8
 ctype    en_US.UTF-8
 tz       America/New_York
 date     2024-11-13
 pandoc   3.2 @ /Applications/RStudio.app/Contents/Resources/app/quarto/bin/tools/aarch64/ (v

-------------------------------------------------------------------------------
```

System Information

```
  # os info
  Sys.info() |>
    tibble::as_tibble(rownames = "par") |>
    dplyr::filter(par != "nodename")
```

```
# A tibble: 7 x 2
  par            value
  <chr>          <chr>
1 sysname        Darwin
2 release        22.5.0
3 version        Darwin Kernel Version 22.5.0: Mon Apr 24 20:53:19 PDT 2023; ro~
4 machine        arm64
5 login          root
6 user           carlislerainey
7 effective_user carlislerainey
```

```
  # cpu info
  benchmarkme::get_cpu()
```

```
$vendor_id
character(0)

$model_name
[1] "Apple M2 Max"

$no_of_cores
[1] 12
```

```
  # ram info
  ram_info <- system("sysctl hw.memsize", intern = TRUE)
  print(ram_info)
```

```
[1] "hw.memsize: 34359738368"
```

R Packages Used

```
  library(tidyverse)

  # Function to get package version
  get_version <- function(package_names){
    sapply(package_names, function(pkg) as.character(packageVersion(pkg)))
  }

  dep <- renv::dependencies("R/") %>%
    bind_rows(renv::dependencies("README.qmd")) %>%
    select(Package) %>%
    distinct() %>%
    arrange(Package) %>%
    mutate(Version = get_version(Package))
```

```
Finding R package dependencies ... Done!
Finding R package dependencies ... Done!
```

```
dep
```

```
       Package Version
1  benchmarkme   1.0.8
2         brms  2.22.0
3        dplyr   1.1.4
4      ggrepel   0.9.6
5    rcrossref   1.2.0
6       readxl   1.4.3
7      remotes   2.5.0
8         renv  1.0.11
9    rmarkdown    2.29
10      scales   1.3.0
11 sessioninfo   1.2.2
12      tibble   3.2.1
13   tidybayes   3.0.7
14   tidyverse   2.0.0
```

Write Script to Install Packages and Versions

```
# create a new script file
script_file <- file("R/install-packages.R", "w")

# write install.packages() commands for each package to the file
for (i in 1:nrow(dep)) {
  install_cmd <- sprintf("remotes::install_version('%s', version = '%s', repos = 'http://c
  cat(install_cmd, file = script_file, append = TRUE)
}

# close the file
close(script_file)
```