Replication Materials for "The Effective Power of Military Coalitions"

This archive contains all replication data and code for "The Effective Power of Military Coalitions: A Unified Theoretical and Empirical Model" by Brenton Kenkel and Kristopher W. Ramsay. For any questions about these materials, contact Brenton Kenkel at brenton.kenkel@gmail.com.

Running the Code

All analysis was performed using R 3.6.0 in a Debian 9 (stretch) environment with the following package versions:

- Amelia 1.7.5
- assertr 2.6
- countrycode 1.1.0
- dplyr 0.8.3
- foreach 1.4.4
- Formula 1.2-3
- geosphere 1.5-10
- haven 2.1.1
- lubridate 1.7.4
- maxLik 1.3-6
- purrr 0.3.2
- randtoolbox 1.30.0
- RColorBrewer 1.1-2
- Rcpp 1.0.1
- tidyverse 1.2.1
- WDI 2.6.0
- xtable 1.8-4

To replicate the analysis in an identical computing environment, build a Docker container using the `Dockerfile` provided in the replication materials.

1. Install Docker from https://docker.com.
2. At the command line, navigate to the directory where the replication files are stored. Build the Docker container for this project by running:

   ```
   docker build -t structwar .
   ```

   This should only take a couple of minutes to complete.
3. Run each script inside the Docker container, mounting the working directory on your local machine as `/tmp/replication` within the Docker container. For example, to run the first script:

   ```
   docker run --rm -it -v .:/tmp/replication structwar Rscript 01_fit_base.r
   ```

If you have the `make` utility available, we have supplied a `Makefile` to automatically run the scripts to generate all figures and tables. Simply run the following command:

```
make all
```

If you use `make`, output logs will be written to the `logs` subdirectory.

Depending on your system's Docker installation and permissions, you may need to run these commands with elevated privileges, i.e., by prepending `sudo` to the `docker build` and `docker run` (and/or `make`) commands.

File List

Code files

Estimated time to run listed in parentheses. Testing performed on a machine with 16 Intel Core i7-7820X processors.

- `01_fit_base.r`: Fit the baseline structural model. (3 minutes)
- `02_boot_base.r`: Bootstrap the baseline structural model. (6.5 hours)
- `03_fit_postwar.r`: Fit the model only using post-World War II data. (Described in Online Appendix J.) (2 minutes)
- `04_boot_postwar.r`: Bootstrap the model using post-World War II data. (6 hours)
- `05_fit_joiners.r`: Fit the model treating the bargaining stage as occurring only among dispute originators. (Described in Online Appendix G.) (8 minutes)
- `06_boot_joiners.r`: Bootstrap the model treating the bargaining stage as occurring only among dispute originators. (7.5 hours)
- `07_regression_tables.r`: Assemble the regression tables for all three models. (<1 minute)

  Table 3, Table A.4, and Table A.9 are generated by this script.
- `08_force_multipliers.r`: Analysis of estimated force multipliers for individual countries. (<1 minute)

  Figure A.1, Table 1, Table 2, and Table A.3 are generated by this script.
- `09_war_probabilities.r`: Analysis of the equilibrium probability of war in observed disputes. (<1 minute)

  Table 4 is generated by this script.
- `10_coalitions.r`: Analysis of counterfactual bargaining behavior and war effort under different coalition structures. (4 minutes)

Figure 2, Figure 3, Figure A.2, Figure A.3, Table 5, Table 6, Table A.5, Table A.6, and Table A.7 are generated by this script.

- `11_model_comparison.r`: Compare structural model fit to alternative models of capability aggregration. (<1 minute)

  Table A.8 is generated by this script.

- `12_descriptives.r`: Compile descriptive statistics about the data. (<1 minute)

  Table A.1 and Table A.2 are generated by this script.

- `13_equilibrium_illustrations.r`: Numerical analysis of the formal model. (<1 minute)

  Figure 1 is generated by this script.

- `OPTIONAL_remake_data.r`: Rebuild the analysis data from raw data files and perform multiple imputation for missing data. Because the replication archive contains the final analysis data, it is not necessary to run this script to replicate the statistical analysis. (2 hours)

- `backend_main.r`, `backend_main.cpp`: R and C++ code for the main structural estimator. These files are called by multiple other scripts and do not have any output on their own.

- `backend_joiners.r`, `backend_joiners.cpp`: R and C++ code for the alternative structural model where only dispute originators are treated as part of the bargaining stage. These files are called by multiple other scripts and do not have any output on their own.

The model fitting and bootstrap scripts (`01_fit_base.r` through `06_boot_joiners.r`) contain a flag called `CONFIRM_MODE`. If set to `TRUE`, then all numerical optimizations are started at the final values to confirm that they meet the first-order conditions. (In some iterations in the bootstrap scripts, the optimizer may run again due to numerical precision issues.) If set to `FALSE`, optimization is run from scratch, increasing computation time considerably. The default is `TRUE`.

Analysis data

These files are all in RData format, to be read into R using the `load()` command.

Main analysis datasets (full descriptions in `CODEBOOK.pdf`):

- `kr_analysis_dispute.rda`: Dispute-level observations and variables.
- `kr_analysis_participant.rda`: Dispute participant-level observations and variables.

Additional datasets used for post-estimation counterfactual simulations:

- `kr_dyad_year.rda`: Dyad-year data with missing values imputed 10x.

- `kr_mid_distance.rda`: Data on state distance to MID locations.
- `kr_state_year.rda`: State-year data with missing values imputed 10x.

Data files containing model results/coefficients:

- `confirm_boot_base.rda`: Starting values for `02_boot_base.r` in `CONFIRM_MODE`.
- `confirm_boot_joiners.rda`: Starting values for `06_boot_joiners.r` in `CONFIRM_MODE`.
- `confirm_boot_postwar.rda`: Starting values for `04_boot_postwar.r` in `CONFIRM_MODE`.
- `confirm_fit_base.rda`: Starting values for `01_fit_base.r` in `CONFIRM_MODE`.
- `confirm_fit_joiners.rda`: Starting values for `05_fit_joiners.r` in `CONFIRM_MODE`.
- `confirm_fit_postwar.rda`: Starting values for `03_fit_postwar.r` in `CONFIRM_MODE`.
- `startvals_fit_base.rda`: Starting values for `01_fit_base.r` when not in `CONFIRM_MODE`.

Raw data from other sources

- `LatLong2011.raw`: State latitudes and longitudes in tab-separated format. Obtained from the source code of the EUGene program (https://eugenesoftware.la.psu.edu/).
- `MIDLOCA_2.0.csv`: Militarized Interstate Dispute locations in comma-separated format. Obtained from the Correlates of War Project (https://correlatesofwar.org/data-sets/midloc/).
- `NMC_5_0.csv`: National Material Capbilities data in comma-separated format. Obtained from the Correlates of War Project (https://correlatesofwar.org/data-sets/national-material-capabilities/).
- `atop4_01dy.csv`: Dyad-year Alliance Treaty Obligations and Provisions data in comma-separated format. Obtained from the Alliance Treaty Obligations and Provisions project (http://www.atopdata.org/).
- `atop_sscores.csv`: Dyad-year S-scores computed from ATOP alliances. Obtained from the Alliance Treaty Obligations and Provisions project (http://www.atopdata.org/).
- `contdir.csv`: Dyadic data on state contiguity in comma-separated format. Obtained from the Correlates of War Project (https://correlatesofwar.org/data-sets/direct-contiguity/).

- `gml-mida-2.1.1.csv`: Dispute-level Militarized Interstate Dispute data in comma-separated format. Obtained from the Gibler-Miller-Little MID fork (http://svmiller.com/gml-mid-data/).
- `gml-midb-2.1.1.csv`: Participant-level Militarized Interstate Dispute data in comma-separated format. Obtained from the Gibler-Miller-Little MID fork (http://svmiller.com/gml-mid-data/).
- `gml-ndy-disputes-2.1.1.csv`: Nondirected dispute-year Militarized Interstate Dispute data in comma-separated format. Obtained from the Gibler-Miller-Little MID fork (http://svmiller.com/gml-mid-data/).
- `majors2016.csv`: State-level major power status in comma-separated format. Obtained from the Correlates of War Project (https://correlatesofwar.org/data-sets/state-system-membership/).
- `mpd2018.dta`: State-year historical GDP data from Maddison Project in STATA format. Obtained from the Groningen Growth and Development Centre (https://www.rug.nl/ggdc/historicaldevelopment/maddison/?lang=en).
- `p4v2017.sav`: State-year Polity IV regime type data in SPSS format. Obtained from the Polity IV Project (https://www.systemicpeace.org/inscrdata.html).
- `pwt90.dta`: State-year historical GDP data from Penn World Tables in STATA format. Obtained from the Groningen Growth and Development Centre (https://www.rug.nl/ggdc/productivity/pwt/?lang=en).
- `states2016.csv`: State system membership data in comma-separated format. Obtained from the Correlates of War Project (https://correlatesofwar.org/data-sets/state-system-membership/).
- `wdi-download.csv`: State-year data from the World Bank's World Development Indicators. Obtained using the R package WDI.
- `wernerpeaceyears.csv`: Dyadic data on years since a dispute prior to 1816, compiled by Suzanne Werner, in comma-separated format. Obtained from the source code of the EUGene program (https://eugenesoftware.la.psu.edu/).

Other files

- `CODEBOOK.txt`, `CODEBOOK.pdf`: Codebook for the analysis data.
- `Dockerfile`: Use to build the Docker container replicating the original analysis environment. See the "Running the Code" section above.
- `Makefile`: Use with the GNU make utility (https://gnu.org/software/make) to run all scripts, generating all figures and tables. See the "Running the Code" section above.

- `README.txt`, `README.pdf`: This file.