

# Package ‘DCPO’

May 7, 2026

**Version** 0.5.3

**Title** Dynamic Comparative Public Opinion

**Description** Estimates latent variables of public opinion cross-nationally and over time from sparse and incomparable survey data. 'DCPO' uses a population-level graded response model with country-specific item bias terms. Sampling is conducted with 'Stan'. References: Solt (2020) <[doi:10.31235/osf.io/d5n9p](https://doi.org/10.31235/osf.io/d5n9p)>.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**ByteCompile** true

**Depends** R (>= 3.4.0), Rcpp (>= 0.12.17), methods

**Imports** rstan (>= 2.18.1), rstantools (>= 2.0.0), beeper, dplyr, forcats, janitor, purrr, tibble, tidyr

**LinkingTo** StanHeaders (>= 2.18.0), rstan (>= 2.18.1), BH (>= 1.66.0-1), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.4.0)

**Suggests** knitr

**SystemRequirements** GNU make

**NeedsCompilation** yes

**RoxygenNote** 7.0.0

**Biarch** true

**Author** Frederick Solt [aut, cre],  
Trustees of Columbia University [cph]

**Maintainer** Frederick Solt <[frederick-solt@uiowa.edu](mailto:frederick-solt@uiowa.edu)>

**Repository** CRAN

**Date/Publication** 2020-05-29 12:50:02 UTC

## Contents

DCPO-package . . . . .	2
dcpo . . . . .	2

dcpo_xvt . . . . .	3
demsup_data . . . . .	4
get_xvt_results . . . . .	5
summarize_dcpo_results . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

DCPO-package	<i>DCPO: Dynamic Comparative Public Opinion</i>
--------------	---

---

## Description

DCPO estimates dynamic comparative public opinion as a latent variable from survey data

## References

Stan Development Team (2018). RStan: the R interface to Stan. R package version 2.18.2. <http://mc-stan.org>

---

dcpo	<i>Estimate Dynamic Comparative Public Opinion</i>
------	--

---

## Description

dcpo uses diverse survey data to estimate public opinion across countries and over time.

## Usage

```
dcpo(dcpo_input, chime = TRUE, ...)
```

## Arguments

dcpo_input	a data frame of survey items and marginals generated by <code>DCPOtools::dcpo_setup</code>
chime	play chime when complete?
...	arguments to be passed to <code>rstan::stan</code> . Defaults reset by dcpo are described below under details.

## Details

dcpo, when passed a data frame `dcpo_input` of survey marginals created by `dcpo_setup`, estimates a latent variable of public opinion. See `rstan::stan` for additional options; stan defaults reset by dcpo are `seed = 324`, `thin = 2`, `cores = min(stan_args$chains, parallel::detectCores()/2)`, and `control <- list(adapt_delta = 0.99, stepsize = 0.005, max_treedepth = 14)`

## Value

a stanfit object

**Examples**

```
out1 <- dcpo(demsup_data,
            chains = 1,
            iter = 300) # 1 chain/300 iterations for example purposes only; use defaults
```

---

dcpo\_xvt

*Cross-validation testing for DCPO*


---

**Description**

dcpo\_xvt performs a single cross-validation test for DCPO

**Usage**

```
dcpo_xvt(
  dcpo_input,
  fold_number = 1,
  number_of_folds = 10,
  fold_seed = 324,
  chime = TRUE,
  ...
)
```

**Arguments**

dcpo_input	a data frame of survey items and marginals generated by <code>DCPOtools::dcpo_setup</code>
fold_number	an integer indicating the number of the fold to treated as test data in the current analysis
number_of_folds	an integer indicating the total number of folds
fold_seed	a seed for reproducibly randomly assigning observations to folds; when a complete set of k-fold cross-validations is to be performed, the same seed should be used for all
chime	play chime when complete?
...	arguments to be passed to <code>rstan::stan</code> . See <code>dcpo</code> .

**Details**

dcpo\_xvt performs a single cross-validation test of a DCPO estimation. To perform a complete k-fold cross-validation, call it repeatedly, changing only the `fold_number` argument.

**Value**

a stanfit object

**Examples**

```
# Single cross-validation test with 25% test set
demsup_xvtest_25pct <- dcpo_xvt(demsup_data,
                               number_of_folds = 4,
                               iter = 300,
                               chains = 1) # 1 chain/300 iterations for example only; use defaults
```

---

demsup\_data

*Support for Democracy in 51 Survey Datasets*


---

**Description**

A dataset containing the prices and other attributes of almost 54,000 diamonds.

**Usage**

```
demsup_data
```

**Format**

A list of 15 elements

**K** an integer, the total number of countries in the data

**T** an integer, the total number of years in the data

**Q** an integer, the total number of distinct survey questions in the data

**R** an integer, the maximum number of response cutpoints in any survey question in the data

**N** an integer, the number of KTQR observations

**kk** a numeric vector of length N, the country of each observation

**tt** a numeric vector of length N, the year of each observation

**qq** a numeric vector of length N, the question of each observation

**rr** a numeric vector of length N, the response cutpoint of each observation

**y\_r** a numeric vector of length N, the number of respondents who provided a response above the relevant cutpoint for each observation

**n\_r** a numeric vector of length N, the total number of respondents for each observation

**fixed\_cutp** a QxR matrix, a truth table indicating the question-cutpoint to be fixed at difficulty .5

**use\_delta** a QxK tibble, a truth table indicating whether item difficulty should be estimated to vary by question-country to account for potential item-response bias

**data** an Nx14 tibble, the aggregate survey response dataset in its original format

**data\_args** a list of length 3, indicating the arguments passed to DCPOtools::format\_dcpo to generate demsup\_data from demsup\_data\$data

## Details

Data on aggregate support for democracy reported in 51 survey datasets in 998 country-years, formatted for use with the functions of the DCPO package

## Source

demsup\_data replicates the data employed in Claassen, Christopher. 2019. "Estimating Smooth Country-Year Panels of Public Opinion." Political Analysis 27(1):1-20. See <https://github.com/fsolt/DCPOtools>.

---

get_xvt_results	<i>Get results of DCPO cross-validation testing</i>
-----------------	---

---

## Description

get\_xvt\_results performs a single cross-validation test for dcpo's estimates of cross-national public opinion

## Usage

```
get_xvt_results(dcpo_xvt_output, ci = 80)
```

## Arguments

dcpo_xvt_output	output from a single call to DCPO: : dcpo_xvt or a k-fold test list of such output generated by purrr::map
ci	an integer indicating the desired width of credible interval for coverage testing; 80 is the default.

## Value

a stanfit object

## Examples

```
# Single cross-validation test with 25% test set
demsup_xvtest_25pct <- dcpo_xvt(demsup_data,
  chime = FALSE,
  number_of_folds = 4,
  iter = 300,
  chains = 1) # 1 chain/300 iterations for example only; use defaults

get_xvt_results(demsup_xvtest_25pct)
```

---

 summarize\_dcpo\_results

*Extract DCPO Results*


---

## Description

summarize\_dcpo\_results is a convenience function that produces summary statistics of the main parameters of a DCPO stanfit object along with the relevant identifying information (country, year, question, and cutpoint).

## Usage

```
summarize_dcpo_results(
  dcpo_input,
  dcpo_output,
  pars = c("theta", "sigma", "alpha", "beta", "delta"),
  probs = c(0.1, 0.9)
)
```

## Arguments

dcpo_input	the data frame of survey items and marginals generated by <code>DCPOtools::dcpo_setup</code> previously passed to <code>DCPO::dcpo</code> to generate the stanfit object passed as <code>dcpo_output</code>
dcpo_output	a stanfit object output by <code>DCPO::dcpo</code>
pars	a character vector of parameter names to be summarized from the DCPO model: theta (mean public opinion), sigma (polarization in public opinion), alpha (question dispersion), beta (question-cutpoint difficulty), and/or delta (country-specific question bias)
probs	a numeric vector of quantiles of interest; the default is <code>c(.1, .9)</code>

## Value

a tibble

## Examples

```
out1 <- dcpo(demsup_data,
  chime = FALSE,
  chains = 1,
  iter = 300) # 1 chain/300 iterations for example purposes only; use defaults

theta_results <- summarize_dcpo_results(dcpo_input = demsup_data,
  dcpo_output = out1,
  pars = "theta")
```

# Index

## \* datasets

demsup\_data, [4](#)

DCPO (DCPO-package), [2](#)

dcpo, [2](#)

DCPO-package, [2](#)

dcpo\_xvt, [3](#)

demsup\_data, [4](#)

get\_xvt\_results, [5](#)

summarize\_dcpo\_results, [6](#)