

# Package ‘PRP’

January 20, 2025

**Title** Bayesian Prior and Posterior Predictive Replication Assessment

**Version** 0.1.1

**Author** Yi Zhao [aut, cre],  
Xiaoquan Wen [aut]

**Maintainer** Yi Zhao <zhayi@umich.edu>

**Description** Utilize the Bayesian prior and posterior predictive checking approach to provide a statistical assessment of replication success and failure. The package is based on the methods proposed in Zhao, Y., Wen X.(2021) <[arXiv:2105.03993](#)>.

**License** GPL-2

**Encoding** UTF-8

**Imports** mvtnorm, stats, graphics

**LazyData** true

**RoxygenNote** 7.1.1

**Depends** R (>= 2.10)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-12-13 08:20:05 UTC

## Contents

mortality . . . . .	2
posterior_prp . . . . .	2
prior_prp . . . . .	3
prob_to_r . . . . .	4
RPP_filtered . . . . .	5
severity . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

---

mortality

*Cardiovascular disease impact on the mortality of COVID-19*


---

### Description

This is a dataset containing several effect estimates and their standard errors for the impact of cardiovascular disease on the mortality of COVID-19 in the literature.

### Usage

```
data("mortality")
```

### Format

An object of class `data.frame` with 6 rows and 3 columns.

### Examples

```
data("mortality")
```

---

posterior\_prp

*Posterior Predictive Replication p-value Calculation*


---

### Description

Posterior Predictive Replication p-value Calculation

### Usage

```
posterior_prp(
  beta,
  se,
  L = 1000,
  r_vec = c(0, 8e-04, 0.006, 0.024),
  test = Q,
  print_test_dist = FALSE
)
```

### Arguments

beta	A vector, containing the estimates in the original study and the replication study.
se	A vector, containing the standard errors of the estimates in the original study and the replication study.
L	A value, determining the times of repeating simulation.

r_vec	A vector, defining the prior reproducible model. Each r value corresponds to a probability of sign consistency.
test	A function designed to calculate the test quantity, the default one is the Cochran's Q test statistics.
print_test_dist	A boolean, determining whether the simulated test statistics value difference will be plot as a histogram or not. Default is False.

**Value**

A list with the following components:

grid	Detailed grid values for the hyperparameters.
test_statistics	The test statistics used in calculating the replication p-value.
n_sim	The L value.
test_stats_dif	The difference between the simulated test statistics quantity and the original value.
pvalue	The resulting posterior predictive replication p-value.

**Examples**

```
data("mortality")
res = posterior_prp(beta = mortality$beta, se = mortality$se, test=Q)
names(res)
print(res$pvalue)
```

---

prior\_prp

*Prior Predictive Replication p-value Calculation*

---

**Description**

Assessing the prior predictive distribution and calculating the replication p-value based on it.

**Usage**

```
prior_prp(
  beta,
  se,
  r_vec = c(0, 8e-04, 0.006, 0.024),
  test = "two_sided",
  report_PI = FALSE
)
```

**Arguments**

beta	A 2-D vector, containing the estimates in the original study and the replication study.
se	A 2-D vector, containing the standard errors of the estimates in the original study and the replication study.
r_vec	A vector, defining the prior reproducible model. Each r value corresponds to a probability of sign consistency.
test	A string, determining which test statistics to utilize. If not specified, the default two-sided one will be used.
report_PI	A boolean, denoting whether the 95% predictive interval for the estimates be reported or not. This option is only valid for two-sided test statistics. The default is FALSE.

**Value**

A list with the following components:

grid	The detailed grid values for the hyperparameters.
test_statistics	The test statistics used in calculating the replication p-value.
pvalue	The resulting prior predictive replicaition p-value.
predictive_interval	The 95% predictive interval if required.

**Examples**

```
data("RPP_filtered")
attach(RPP_filtered)
rpp_pval<-sapply(1:nrow(RPP_filtered),function(x)
  prior_prp(beta=c(beta_orig[x], beta_rep[x]),se=c(se_orig[x], se_rep[x]))$pvalue)
```

---

prob_to_r	<i>Sign consistency probability and the value for r parameter 1-1 transformation</i>
-----------	--

---

**Description**

This function transforms the probability of simulated  $\beta_j$  having the same sign with the underlying true effect  $\beta_j$  to the corresponding heterogeneity  $r$  parameter value.

**Usage**

```
prob_to_r(p)
```

**Arguments**

`p` A value, the required probability of sign consistency.

**Value**

The corresponding heterogeneity parameter value.

---

RPP_filtered	<i>Filtered RPP data</i>
--------------	--------------------------

---

**Description**

This contains the RP:P data from the Open Science Collaboration project after filtering.

**Usage**

```
data("RPP_filtered")
```

**Format**

An object of class `data.frame` with 73 rows and 5 columns.

**Examples**

```
data("RPP_filtered")
```

---

severity	<i>Cardiovascular disease impact on the severe case rate of COVID-19</i>
----------	--

---

**Description**

This is a dataset containing several effect estimates and their standard errors for the impact of cardiovascular disease on the severe case rate of COVID-19 in the literature.

**Usage**

```
data("severity")
```

**Format**

An object of class `data.frame` with 6 rows and 3 columns.

**Examples**

```
data("severity")
```

# Index

## \* datasets

- mortality, 2
- RPP\_filtered, 5
- severity, 5

mortality, 2

posterior\_prp, 2

prior\_prp, 3

prob\_to\_r, 4

RPP\_filtered, 5

severity, 5