

# Package ‘qualCI’

February 20, 2015

**Version** 0.1

**Date** 2014-07-01

**Title** Causal Inference with Qualitative and Ordinal Information on Outcomes

**Author** Konstantin Kashin <kkashin@fas.harvard.edu>, Adam Glynn <aglynn@emory.edu>, Nahomi Ichino <nichino@umich.edu>

**Maintainer** Konstantin Kashin <kkashin@fas.harvard.edu>

**Depends** R (>= 2.15.0)

**Imports** combinat

**Description** Exact one-sided p-values and confidence intervals for an outcome variable defined on an interval measurement scale with only qualitative and ordinal information available.

**License** GPL (>= 2)

**BugReports** <https://github.com/kkashin/qualCI/issues>

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-07-03 20:59:31

## R topics documented:

pluralityPairs . . . . .	2
pluralitySets . . . . .	3
quade . . . . .	4
quade.prep . . . . .	5
qualCI . . . . .	6
<b>Index</b>	<b>8</b>

---

pluralityPairs      *Glynn and Ichino (2014b) replication data for paired design*

---

### Description

Replication data for Glynn and Ichino (2014b). Contains the dataframe `pluralityPairs`, corresponding to a matched pairs design.

### Usage

```
pluralityPairs
```

### Format

The dataframe has 4 variables for 8 countries in sub-Saharan Africa that held transitional multi-party presidential elections in the 1990s:

- `country`: country name (string).
- `pair`: pair identifier (numeric). The 8 countries are matched into 4 pairs.
- `plurality`: binary treatment indicator (numeric), where a country is coded 1 if it used plurality rules with one round only for its transitional multi-party presidential elections in the 1990s. A country is coded 0 if the electoral rules for the transitional election had a run-off provision to eliminate weaker candidates and hold a second round election if no candidate met a vote threshold in the first round.
- `OppHarRankPair`: within-pair rank for the country's level of opposition harassment by the incumbent regime in the lead up to the transitional multi-party presidential election (numeric). Within each pair, the country that had greater opposition harassment is coded 1; the other country is coded 0. The rank is determined by qualitative reports of whether the incumbent regime targeted only opposition leaders or supporters or both; whether opposition members were detained and their treatment during detention, such as torture or death; whether violence was threatened or used; and the extent of any violence towards the opposition.

### Source

Glynn and Ichino (2014a,b) contain fuller descriptions of opposition harassment that are the basis for the ranks and details on the sample and matching.

### References

Glynn, Adam N., and Nahomi Ichino. 2014a. [Supplementary Information \(online appendix\) for "Using Qualitative Information to Improve Causal Inference."](#) *American Journal of Political Science*.

Glynn, Adam N., and Nahomi Ichino. 2014b. "Using Qualitative Information to Improve Causal Inference." *American Journal of Political Science*. Forthcoming.

### See Also

[quade](#).

---

pluralitySets

*Glynn and Ichino (2014b) replication data for full matching design*

---

### Description

Replication data for Glynn and Ichino (2014b). Contains the dataframe `pluralitySets`, corresponding to a full matching design.

### Usage

```
pluralitySets
```

### Format

The dataframe has 4 variables for 9 countries in sub-Saharan Africa that held transitional multi-party presidential elections in the 1990s:

- `country`: country name (string).
- `set`: set identifier (numeric). The 9 countries are matched into 3 sets with varying numbers of treated and control units.
- `plurality`: binary treatment indicator (numeric), where a country is coded 1 if it used plurality rules with one round only for its transitional multi-party presidential elections in the 1990s. A country is coded 0 if the electoral rules for the transitional election had a run-off provision to eliminate weaker candidates and hold a second round election if no candidate met a vote threshold in the first round.
- `OppHarRankSet`: within-set rank for the country's level of opposition harassment by the incumbent regime in the lead up to the transitional multi-party presidential election (numeric). Within each set, the country that had the lowest level of opposition harassment is coded 1. The country with the highest level of opposition harassment in the set is assigned the rank equal to the number of units in the set. The rank is determined by qualitative reports of whether the incumbent regime targeted only opposition leaders or supporters or both; whether opposition members were detained and their treatment during detention, such as torture or death; whether violence was threatened or used; and the extent of any violence towards the opposition.

### Source

Glynn and Ichino (2014a,b) contain fuller descriptions of opposition harassment that are the basis for the ranks and details on the sample and full matching.

### References

- Glynn, Adam N., and Nahomi Ichino. 2014a. [Supplementary Information \(online appendix\) for "Using Qualitative Information to Improve Causal Inference."](#) *American Journal of Political Science*. (weblink)
- Glynn, Adam N., and Nahomi Ichino. 2014b. "Using Qualitative Information to Improve Causal Inference." *American Journal of Political Science*. Forthcoming.

**See Also**

[quade.](#)

---

quade

*Calculate Quade's statistic and one-sided p-value.*

---

**Description**

Calculates Quade's statistic and exact one-sided p-value for a stratified completely randomized design with arbitrary probability of treatment assignment.

**Usage**

```
quade(obj)
```

**Arguments**

`obj` an object of class "matchedSets".

**Value**

quade returns an object of class "quade", which consists of the following components:

<code>Qobs</code>	observed test statistic.
<code>permutations</code>	a permutation matrix of possible test statistics and associated probabilities.
<code>pval</code>	p-value.
<code>sets</code>	the "matchedSets" object input into the function.

**Author(s)**

Konstantin Kashin <kkashin@fas.harvard.edu>

**Examples**

```
data(pluralityPairs)
between.ranks <- c(3,4,2,1)
dat <- quade.prep(data=pluralityPairs, set="pair", treatment="plurality",
withinRank="OppHarRank", unit="country", betweenRank=between.ranks)
qout <- quade(dat)
qout
```

```
data(pluralitySets)
between.ranks <- c(2,3,1)
dat <- quade.prep(data=pluralitySets, set="set", treatment="plurality",
withinRank="OppHarRank", unit="country", betweenRank=between.ranks)
qout <- quade(dat)
qout
head(qout$permutations)
```

---

quade.prep	<i>Prepare data for qualitative causal inference.</i>
------------	---

---

### Description

Function to prepare data for the `quade` function and hence the `qualCI` function. Processes a data frame with set identifiers, treatment indicators, and within-set ranks, along with vector of between-set ranks to create a "matchedSets" object.

### Usage

```
quade.prep(data, set, treatment, withinRank, unit=NULL, betweenRank)
```

### Arguments

<code>data</code>	an object of class "data.frame" containing set, treatment, and within-set rank variables.
<code>set</code>	the column name of data that contains the set identifier. It cannot contain any NAs.
<code>treatment</code>	the column name of data that contains the treatment indicator. Treatment indicator must be binary and cannot contain any NAs.
<code>withinRank</code>	the column name of data that contains the within-set ranks. It cannot contain any NAs and must be numeric. See below for details.
<code>unit</code>	the column name of data that contains the unit names. May be set to NULL, in which case the unit's name is set to its within-set position.
<code>betweenRank</code>	a (named) numeric vector of between-set ranks. Names on vector should correspond to set identifiers. If <code>betweenRank</code> object has no names attribute, an element's name is set to its position in the vector.

### Details

Function for preparing data for the `quade` function. Each set should contain at least one treated unit and at least one control unit.

Within-set ranks must consist of a sequence of consecutive integers starting with either 0 or 1. Ties are not permitted. Typical practice uses 0 and 1 to rank units in a matched pairs design and uses consecutive integers starting with 1 to rank units in a more general matched sets design.

Between-set ranks should be consecutive integers starting with 1. The names on the vector should correspond to the set nomenclature in `data[, set]`.

### Value

`prepareData` returns an object of class "matchedSets", which is a named list of length equal to the number of sets. A given list element (corresponding to a set) consists of the following components:

<code>possibleTreat</code>	a matrix of possible treatment vectors, where rows correspond to different treatment permutations and columns correspond to the units.
----------------------------	--

prob	a vector of probabilities corresponding to the probability of receiving a particular treatment permutation from the matrix of possible treatments. Length is equal to the number of rows in possibleTreat.
obsTreat	a named vector denoting observed treatment status.
withinRank	a named vector denoting within-set ranks of units.
rank	numeric, denoting between-set rank of given set.

**Author(s)**

Konstantin Kashin <kkashin@fas.harvard.edu>

**Examples**

```
data(pluralityPairs)
between.ranks <- c(3,4,2,1)
dat <- quade.prep(data=pluralityPairs, set="pair", treatment="plurality",
withinRank="OppHarRank", unit="country", betweenRank=between.ranks)
dat
```

```
data(pluralitySets)
between.ranks <- c(2,3,1)
dat <- quade.prep(data=pluralitySets, set="set", treatment="plurality",
withinRank="OppHarRank", unit="country", betweenRank=between.ranks)
dat
dat[[1]]
```

---

qualCI

*Calculate qualitative confidence intervals.*

---

**Description**

Calculates one-sided qualitative confidence intervals based on a sign test for a matched pairs design and a stratified rank sum test for a general matched sets design.

**Usage**

```
qualCI(obj)
```

**Arguments**

obj an object of class "quade".

**Details**

For a matched pairs design, the function identifies the difference between the units in the lowest ranked pair as the qualitative confidence interval, along with the confidence level from the sign test. For a general matched sets design, the function prompts the user to identify the hardest-to-rank pair from a pool of candidate pairs (all treatment-control pairs where the treated unit has a within-set rank that is immediately above the rank for the control unit). The function then returns the difference between the units in this pair as the qualitative confidence interval, along with the confidence level from the stratified rank sum test.

**Value**

qualCI returns an object of class "qualCI", which consists of the following components:

qualci	pair of units whose difference represents the qualitative confidence interval
set	set from which pair was drawn
ci.level	confidence interval level based on sign rank test for matched pairs and the stratified rank sum test for more general matched sets

**Author(s)**

Konstantin Kashin <kkashin@fas.harvard.edu>

**Examples**

```
data(pluralityPairs)
between.ranks <- c(3,4,2,1)
dat <- quade.prep(data=pluralityPairs, set="pair", treatment="plurality",
  withinRank="OppHarRank", unit="country", betweenRank=between.ranks)
qout <- quade(dat)
qualCI(qout)
```

```
data(pluralitySets)
between.ranks <- c(2,3,1)
dat <- quade.prep(data=pluralitySets, set="set", treatment="plurality",
  withinRank="OppHarRank", unit="country", betweenRank=between.ranks)
qout <- quade(dat)
# not run because requires user input
# qualCI(qout)
```

# Index

\*Topic **causal inference**

pluralityPairs, [2](#)

pluralitySets, [3](#)

quade, [4](#)

quade.prep, [5](#)

qualCI, [6](#)

\*Topic **datasets**

pluralityPairs, [2](#)

pluralitySets, [3](#)

pluralityPairs, [2](#)

pluralitySets, [3](#)

print.matchedSets (quade.prep), [5](#)

quade, [2](#), [4](#), [4](#), [5](#), [6](#)

quade.prep, [5](#)

qualCI, [5](#), [6](#)