

Package ‘WhatIf’

October 12, 2022

Title Software for Evaluating Counterfactuals

Version 1.5-10

Date 2020-11-08

Author Heather Stoll <hstoll@polsci.ucsb.edu>, Gary King
<king@harvard.edu>, Langche Zeng <zeng@ucsd.edu>,
Christopher Gandrud <zelig.zee@gmail.com>, Ben Sabath

Maintainer Soubhik Barari <soubhikbarari@gmail.com>

Description Inferences about counterfactuals are essential for prediction, answering what if questions, and estimating causal effects. However, when the counterfactuals posed are too far from the data at hand, conclusions drawn from well-specified statistical analyses become based largely on speculation hidden in convenient modeling assumptions that few would be willing to defend. Unfortunately, standard statistical approaches assume the veracity of the model rather than revealing the degree of model-dependence, which makes this problem hard to detect. WhatIf offers easy-to-apply methods to evaluate counterfactuals that do not require sensitivity testing over specified classes of models. If an analysis fails the tests offered here, then we know that substantive inferences will be sensitive to at least some modeling choices that are not based on empirical evidence, no matter what method of inference one chooses to use. WhatIf implements the methods for evaluating counterfactuals discussed in Gary King and Langche Zeng, 2006, “The Dangers of Extreme Counterfactuals,” *Political Analysis* 14 (2) <[DOI:10.1093/pan/mpj004](https://doi.org/10.1093/pan/mpj004)>; and Gary King and Langche Zeng, 2007, “When Can History Be Our Guide? The Pitfalls of Counterfactual Inference,” *International Studies Quarterly* 51 (March) <[DOI:10.1111/j.1468-2478.2007.00445.x](https://doi.org/10.1111/j.1468-2478.2007.00445.x)>.

License GPL (>= 3)

URL <https://gking.harvard.edu/whatif>

BugReports <https://github.com/IQSS/WhatIf/issues>

Depends R (>= 2.3.1)

Imports lpSolve, pbmcapply, parallel

Suggests testthat

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2020-11-14 14:20:06 UTC

R topics documented:

peacecf	2
peacef	3
plot.whatif	3
print.summary.whatif	5
print.whatif	6
summary.whatif	7
whatif	8

Index	14
--------------	-----------

peacecf	<i>Counterfactual Replication Data from King and Zeng 2006b</i>
---------	---

Description

This data set is one of two that together allow the replication of the analysis in section 2.4 of King and Zeng 2006b. It contains data on 122 *counterfactuals* derived by King and Zeng 2006b from the factual Doyle and Sambanis 2000 data set, `peacef`. It should be used in conjunction with the latter.

Usage

```
data(peacecf)
```

Format

A data frame with dimensions 122-by-11. Columns are covariates and rows are data points (or units). The covariates are as in `peacef` with the exception of the key causal variable, `untype4`, which is transformed to `1 - untype4`.

Source

King and Zeng 2006b

References

King, Gary and Langche Zeng. 2006. "When Can History Be Our Guide? The Pitfalls of Counterfactual Inference." *International Studies Quarterly* 51 (March).

Doyle, Michael W. and Nicholas Sambanis. 2000. "International Peacebuilding: A Theoretical and Quantitative Analysis." *American Political Science Review* 94, no.4: 779–801.

peacef

Factual Replication Data from King and Zeng 2006b

Description

This data set is one of two that together allow the replication of the analysis in section 2.4 of King and Zeng 2006b. It contains *factual* data from Doyle and Sambanis 2000 on 124 post-WWII civil wars. It should be used in conjunction with the data set of counterfactuals derived from it, peacecf.

Usage

```
data(peacef)
```

Format

A data frame with dimensions 124-by-11. Columns are covariates and rows are data points (or units). The covariates are decade, wartype, logcost, wardur, factnum, factnumsq, trnsfcap, untype4, treaty, develop, and exp, in that order.

Source

King and Zeng 2006b

References

King, Gary and Langche Zeng. 2006. "When Can History Be Our Guide? The Pitfalls of Counterfactual Inference." *International Studies Quarterly* 51 (March).

Doyle, Michael W. and Nicholas Sambanis. 2000. "International Peacebuilding: A Theoretical and Quantitative Analysis." *American Political Science Review* 94, no.4: 779–801.

plot.whatif

Plot Cumulative Frequencies of Distances for "whatif" Objects

Description

Generates a cumulative frequency plot of distances from an object of class "whatif". The cumulative frequencies (the fraction of rows in the observed data set with either Gower or (squared) Euclidian distances to the counterfactuals less than the given value on the horizontal axis) appear on the vertical axis.

Usage

```
## S3 method for class 'whatif'  
plot(x, type = "f", numcf = NULL, eps = FALSE, ...)
```

Arguments

x	An object of class "whatif", the output of the function whatif.
type	A character string; the type of plot of the cumulative frequencies of the distances to be produced. Possible types are: "f" for cumulative frequencies only; "l" for LOWESS smoothing of cumulative frequencies only; and "b" for both cumulative frequencies and LOWESS smoothing. The default is "f".
numcf	A numeric vector; the specific counterfactuals to be plotted. Each element represents a counterfactual, specifically its row number from the matrix or data frame of counterfactuals. By default, all counterfactuals are plotted. Default is NULL.
eps	A Boolean; should an encapsulated postscript file be generated? Setting the argument equal to TRUE generates an .eps file, which is saved to your working directory with file name of form 'graph_' type '_' numcf '.eps', where 'type' and 'numcf' are the values of the respective arguments. Specifically, 'numcf' takes the value of the first element of the argument numcf unless all counterfactuals were plotted, in which case all appears in the place of 'numcf'. Default is FALSE, which instead prints the graph to the screen.
...	Further arguments passed to and from other methods.

Details

LOWESS scatterplot smoothing using the function lowess is plotted in blue. Counterfactuals in the convex hull are plotted with a solid line and counterfactuals outside of the convex hull with a dashed line.

Value

A graph printed to the screen or an encapsulated postscript file saved to your working directory. In the latter case, the file name has form 'graph_' type '_' numcf '.eps', where 'type' and 'numcf' are the values of the respective arguments.

Author(s)

Stoll, Heather <hstoll@polsci.ucsb.edu>, King, Gary <king@harvard.edu> and Zeng, Langche <zeng@ucsd.edu>

References

King, Gary and Langche Zeng. 2006. "The Dangers of Extreme Counterfactuals." *Political Analysis* 14 (2). Available from <https://gking.harvard.edu>.

King, Gary and Langche Zeng. 2007. "When Can History Be Our Guide? The Pitfalls of Counterfactual Inference." *International Studies Quarterly* 51 (March). Available from <https://gking.harvard.edu>.

See Also

[whatif](#), [summary.whatif](#), [print.whatif](#), [print.summary.whatif](#)

Examples

```
## Create example data sets and counterfactuals
my.cfact <- matrix(rnorm(3*5), ncol = 5)
my.data <- matrix(rnorm(100*5), ncol = 5)

## Evaluate counterfactuals
my.result <- whatif(data = my.data, cfact = my.cfact, mc.cores = 1)

## Plot cumulative frequencies for the first two counterfactuals (rows
## 1 and 2) in my.cfact
plot(my.result, type = "b", numcf = c(1, 2), mc.cores = 1)
```

```
print.summary.whatif Print "summary.whatif" Object
```

Description

Prints the information generated from the whatif output object by a call to summary, which is stored in an object of class "summary.whatif".

Usage

```
## S3 method for class 'summary.whatif'
print(x, ...)
```

Arguments

x An object of class "summary.whatif", the output of the function summary.whatif.
... Further arguments passed to and from other methods.

Value

A printout to the screen of the whatif information summarized in the summary.whatif output object.

Author(s)

Stoll, Heather <hstoll@polsci.ucsb.edu>, King, Gary <king@harvard.edu> and Zeng, Langche <zeng@ucsd.edu>

References

King, Gary and Langche Zeng. 2006. "The Dangers of Extreme Counterfactuals." *Political Analysis* 14 (2). Available from <https://gking.harvard.edu>.
King, Gary and Langche Zeng. 2007. "When Can History Be Our Guide? The Pitfalls of Counterfactual Inference." *International Studies Quarterly* 51 (March). Available from <https://gking.harvard.edu>.

See Also

[whatif](#), [plot.whatif](#), [summary.whatif](#), [print.whatif](#)

Examples

```
## Create example data sets and counterfactuals
my.cfact <- matrix(rnorm(3*5), ncol = 5)
my.data <- matrix(rnorm(100*5), ncol = 5)

## Evaluate counterfactuals
my.result <- whatif(data = my.data, cfact = my.cfact, mc.cores = 1)

## Print summary output object
my.result.sum <- summary(my.result)
print(my.result.sum)
```

print.whatif	<i>Print "whatif" Object</i>
--------------	------------------------------

Description

Prints the information produced by the function `whatif`, an object of class "whatif", to the screen.

Usage

```
## S3 method for class 'whatif'
print(x, print.dist = FALSE, print.freq = FALSE, ...)
```

Arguments

<code>x</code>	An object of class "whatif", the output of the function <code>whatif</code> .
<code>print.dist</code>	A Boolean; should the matrix of pairwise distances between each counterfactual and data point be printed to the screen, if it was returned? Default is FALSE.
<code>print.freq</code>	A Boolean; should the matrix of cumulative frequencies of distances for each counterfactual be printed to the screen? Default is FALSE.
<code>...</code>	Further arguments passed to and from other methods.

Value

A printout to the screen of the information contained in the `whatif` output object.

Author(s)

Stoll, Heather <hstoll@polsci.ucsb.edu>, King, Gary <king@harvard.edu> and Zeng, Langche <zeng@ucsd.edu>

References

King, Gary and Langche Zeng. 2006. "The Dangers of Extreme Counterfactuals." *Political Analysis* 14 (2). Available from <https://gking.harvard.edu>.

King, Gary and Langche Zeng. 2007. "When Can History Be Our Guide? The Pitfalls of Counterfactual Inference." *International Studies Quarterly* 51 (March). Available from <https://gking.harvard.edu>.

See Also

[whatif](#), [plot.whatif](#), [summary.whatif](#), [print.summary.whatif](#)

Examples

```
## Create example data sets and counterfactuals
my.cfact <- matrix(rnorm(3*5), ncol = 5)
my.data <- matrix(rnorm(100*5), ncol = 5)

## Evaluate counterfactuals
my.result <- whatif(data = my.data, cfact = my.cfact, mc.cores = 1)

## Print output object
print(my.result)
```

summary.whatif	<i>Summary of "whatif" Object</i>
----------------	-----------------------------------

Description

Summarizes the information produced by the function `whatif`. The summary generated is returned as an output object and also printed to the screen.

Usage

```
## S3 method for class 'whatif'
summary(object, ...)
```

Arguments

<code>object</code>	An object of class "whatif", the output of the function <code>whatif</code> .
<code>...</code>	Further arguments passed to and from other methods.

Value

An object of class "summary.whatif", a list containing the following five elements:

<code>call</code>	The original call to <code>whatif</code> .
<code>m</code>	A scalar. The total number of counterfactuals evaluated.

<code>m.inhull</code>	A scalar. The number of counterfactuals evaluated that are in the convex hull of the observed covariate data.
<code>mean.near</code>	A scalar. The average percentage of data nearby each counterfactual, where the average is taken over all counterfactuals.
<code>sum.df</code>	A data frame with three columns and m rows, where m is the number of counterfactuals. The first column, <code>cfact</code> , indexes the counterfactuals. The second column, <code>in.hull</code> , contains the results of the convex hull test. The third column, <code>per.near</code> , contains the percentage of data points nearby each counterfactual.

This object is printed to the screen.

Author(s)

Stoll, Heather <hstoll@polsci.ucsb.edu>, King, Gary <king@harvard.edu> and Zeng, Langche <zeng@ucsd.edu>

References

- King, Gary and Langche Zeng. 2006. "The Dangers of Extreme Counterfactuals." *Political Analysis* 14 (2). Available from <https://gking.harvard.edu>.
- King, Gary and Langche Zeng. 2007. "When Can History Be Our Guide? The Pitfalls of Counterfactual Inference." *International Studies Quarterly* 51 (March). Available from <https://gking.harvard.edu>.

See Also

[whatif](#), [plot.whatif](#), [print.whatif](#), [print.summary.whatif](#)

Examples

```
## Create example data sets and counterfactuals
my.cfact <- matrix(rnorm(3*5), ncol = 5)
my.data <- matrix(rnorm(100*5), ncol = 5)

## Evaluate counterfactuals
my.result <- whatif(data = my.data, cfact = my.cfact, mc.cores = 1)

## Print summary
summary(my.result)
```

whatif

Counterfactual Evaluation

Description

Implements the methods described in King and Zeng (2006a, 2006b) for evaluating counterfactuals.

Usage

```
whatif(formula = NULL, data, cfact, range = NULL, freq = NULL, nearby = 1,
       distance = "gower", miss = "list", choice = "both", return.inputs = FALSE,
       return.distance = FALSE, mc.cores = detectCores(), ...)
```

Arguments

- | | |
|---------|--|
| formula | An optional formula without a dependent variable that is of class "formula" and that follows standard R conventions for formulas, e.g. $\sim x1 + x2$. Allows you to transform or otherwise re-specify combinations of the variables in both data and cfact. To use this parameter, both data and cfact must be coercible to data frames; the variables of both data and cfact must be labeled; and all variables appearing in formula must also appear in both data and cfact. Otherwise, errors are returned. The intercept is automatically dropped. Default is NULL. |
| data | <p>May take one of the following forms:</p> <ol style="list-style-type: none"> 1. A R model output object, such as the output from calls to <code>lm</code>, <code>glm</code>, and <code>zelig</code>. If it is not a <code>zelig</code> object, such an output object must be a list. It must additionally have either a <code>formula</code> or <code>terms</code> component and either a <code>data</code> or <code>model</code> component; if it does not, an error is returned. Of the latter, <code>whatif</code> first looks for <code>data</code>, which should contain either the original data set supplied as part of the model call (as in <code>glm</code>) or the name of this data set (as in <code>zelig</code>), which is assumed to reside in the global environment. If <code>data</code> does not exist, <code>whatif</code> then looks for <code>model</code>, which should contain the model frame (as in <code>lm</code>). The intercept is automatically dropped from the extracted <i>observed covariate data</i> set if the original model included one. 2. A n-by-k non-character (logical or numeric) matrix or data frame of <i>observed covariate data</i> with n data points or units and k covariates. All desired variable transformations and interaction terms should be included in this set of k covariates unless <code>formula</code> is alternatively used to produce them. However, an intercept should not be. Such a matrix may be obtained by passing model output (e.g., output from a call to <code>lm</code>) to <code>model.matrix</code> and excluding the intercept from the resulting matrix if one was fit. Note that <code>whatif</code> will attempt to coerce data frames to their internal numeric values. Hence, data frames should only contain logical, numeric, and factor columns; character columns will lead to an error being returned. 3. A string. Either the complete path (including file name) of the file containing the data or the path relative to your working directory. This file should be a white space delimited text file. If it contains a header, you must include a column of row names as discussed in the help file for the R function <code>read.table</code>. The data in the file should be as otherwise described in (2). <p>Missing data is allowed and will be dealt with via the argument <code>missing</code>. It should be flagged using R's standard representation for missing data, NA.</p> |
| cfact | A R object or a string. If a R object, a m -by- k non-character matrix or data frame of <i>counterfactuals</i> with m counterfactuals and the same k covariates (in the same order) as in <code>data</code> . However, if <code>formula</code> is used to select a subset of the k covariates, then <code>cfact</code> may contain either only these $j \leq k$ covariates or the complete set of k covariates. An intercept should not be included as one of |

the covariates. It will be automatically dropped from the counterfactuals generated by **Zelig** if the original model contained one. Data frames will again be coerced to their internal numeric values if possible. If a string, either the complete path (including file name) of the file containing the counterfactuals or the path relative to your working directory. This file should be a white space delimited text file. See the discussion under data for instructions on dealing with a header. All counterfactuals should be fully observed: if you supply counterfactuals with missing data, they will be list-wise deleted and a warning message will be printed to the screen.

range	An optional numeric vector of length k , where k is the number of covariates. Each element represents the range of the corresponding covariate for use in calculating Gower distances. Use this argument when covariate data do not represent the population of interest, such as selection by stratification or experimental manipulation. By default, the range of each covariate is calculated from the data (the difference of its maximum and minimum values in the sample), which is appropriate when a simple random sampling design was used. To supply your own range for the k th covariate, set the k th element of the vector equal to the desired range and all other elements equal to NA. Default is NULL.
freq	An optional numeric vector of any positive length, the elements of which comprise a set of distances. Used in calculating cumulative frequency distributions for the distances of the data points from each counterfactual. For each such distance and counterfactual, the cumulative frequency is the fraction of observed covariate data points with distance to the counterfactual less than or equal to the supplied distance value. The default varies with the distance measure used. When the Gower distance measure is employed, frequencies are calculated for the sequence of Gower distances from 0 to 1 in increments of 0.05. When the Euclidian distance measure is employed, frequencies are calculated for the sequence of Euclidian distances from the minimum to the maximum observed distances in twenty equal increments, all rounded to two decimal places. Default is NULL.
nearby	An optional scalar indicating which observed data points are considered to be nearby (i.e., withing 'nearby' geometric variances of) the counterfactuals. Used to calculate the summary statistic returned by the function: the fraction of the observed data nearby each counterfactual. By default, the geometric variance of the covariate data is used. For example, setting nearby to 2 will identify the proportion of data points within two geometric variances of a counterfactual. Default is NULL.
distance	An optional string indicating which of two distance measures to employ. The choices are either "gower", Gower's non-parametric distance measure (G^2), which is suitable for both qualitative and quantitative data; or "euclidian", squared Euclidian distance, which is only suitable for quantitative data. The default is "gower".
miss	An optional string indicating the strategy for dealing with missing data in the observed covariate data set. whatif supports two possible missing data strategies: "list", list-wise deletion of missing cases; and "case", ignoring missing data case-by-case. Note that if "case" is selected, cases with missing values are deleted listwise for the convex hull test and for computing Euclidian distances,

but pairwise deletion is used in computing the Gower distances to maximally use available information. The user is strongly encouraged to treat missing data using specialized tools such as *Amelia* prior to feeding the data to `whatif`. Default is "list".

<code>choice</code>	An optional string indicating which analyses to undertake. The options are either "hull", only perform the convex hull membership test; "distance", do not perform the convex hull test but do everything else, such as calculating the distance between each counterfactual and data point; or "both", undertake both the convex hull test and the distance calculations (i.e., do everything). Default is "both".
<code>return.inputs</code>	A Boolean; should the processed observed covariate and counterfactual data matrices on which all <code>whatif</code> computations are performed be returned? Processing refers to internal <code>whatif</code> operations such as the subsetting of covariates via formula, the deletion of cases with missing values, and the coercion of data frames to numeric matrices. Primarily intended for diagnostic purposes. If TRUE, these matrices are returned as a list. Default is FALSE.
<code>return.distance</code>	A Boolean; should the matrix of distances between each counterfactual and data point be returned? If TRUE, this matrix is returned as part of the output; if FALSE, it is not. Default is FALSE due to the large size that this matrix may attain.
<code>mc.cores</code>	The number of cores to use for the convex hull test, i.e. at most how many child processes will be run simultaneously. Must be at least one, and parallelization requires at least two cores. The default is set by <code>detectCores</code> .
<code>...</code>	Further arguments passed to and from other methods.

Details

This function is the primary tool for evaluating your counterfactuals. Specifically, it:

1. Determines whether or not your counterfactuals are in the convex hull of the observed covariate data.
2. Computes the distance of your counterfactuals from each of the n observed covariate data points. The default distance function used is Gower's non-parametric measure.
3. Computes a summary statistic for each counterfactual based on the distances in (2): the fraction of observed covariate data points with distances to your counterfactual less than a value you supply. By default, this value is taken to be the geometric variability of the observed data.
4. Computes the cumulative frequency distribution of each counterfactual for the distances in (2) using values that you supply. By default, Gower distances from 0 to 1 in increments of 0.05 are used.

Value

An object of class "whatif", a list consisting of the following six or seven elements:

<code>call</code>	The original call to <code>whatif</code> .
-------------------	--

inputs	A list with two elements, data and cfact. Only present if return.inputs was set equal to TRUE in the call to whatif. The first element is the processed observed covariate data matrix on which all whatif computations were performed. The second element is the processed counterfactual data matrix.
in.hull	A logical vector of length m , where m is the number of counterfactuals. Each element of the vector is TRUE if the corresponding counterfactual is in the convex hull and FALSE otherwise.
dist	A m -by- n numeric matrix, where m is the number of counterfactuals and n is the number of data points (units). Only present if return.distance was set equal to TRUE in the call to whatif. The $[i, j]$ th entry of the matrix contains the distance between the i th counterfactual and the j th data point.
geom.var	A scalar. The geometric variability of the observed covariate data.
sum.stat	A numeric vector of length m , where m is the number of counterfactuals. The m th element contains the summary statistic for the corresponding counterfactual. This summary statistic is the fraction of data points with distances to the counterfactual less than the argument nearby, which by default is the geometric variability of the covariates.
cum.freq	A numeric matrix. By default, the matrix has dimension m -by-21, where m is the number of counterfactuals; however, if you supplied your own frequencies via the argument freq, the matrix has dimension m -by- f , where f is the length of freq. Each row of the matrix contains the cumulative frequency distribution for the corresponding counterfactual calculated using either the distance measure-specific default set of distance values or the set that you supplied (see the discussion under the argument freq). Hence, the $[i, j]$ th entry of the matrix is the fraction of data points with distances to the i th counterfactual less than or equal to the value represented by the j th column. The column names contain these values.

Note

This function requires the **IpSolve** package.

Author(s)

Stoll, Heather <hstoll@polsci.ucsb.edu>, King, Gary <king@harvard.edu> and Zeng, Langche <zeng@ucsd.edu>

References

- King, Gary and Langche Zeng. 2006. "The Dangers of Extreme Counterfactuals." *Political Analysis* 14 (2). Available from <https://gking.harvard.edu>.
- King, Gary and Langche Zeng. 2007. "When Can History Be Our Guide? The Pitfalls of Counterfactual Inference." *International Studies Quarterly* 51 (March). Available from <https://gking.harvard.edu>.

See Also

[plot.whatif](#), [summary.whatif](#), [print.whatif](#), [print.summary.whatif](#)

Examples

```
## Create example data sets and counterfactuals
my.cfact <- matrix(rnorm(3*5), ncol = 5)
my.data <- matrix(rnorm(100*5), ncol = 5)

## Evaluate counterfactuals
my.result <- whatif(data = my.data, cfact = my.cfact, mc.cores = 1)

## Evaluate counterfactuals and supply own gower distances for
## cumulative frequency distributions
my.result <- whatif(cfact = my.cfact, data = my.data,
                    freq = c(0, .25, .5, 1, 1.25, 1.5), mc.cores = 1)
```

Index

- * **datasets**
 - peacecf, [2](#)
 - peacecf, [3](#)
- * **hplot**
 - plot.whatif, [3](#)
- * **htest**
 - whatif, [8](#)
- * **models**
 - whatif, [8](#)
- * **print**
 - print.summary.whatif, [5](#)
 - print.whatif, [6](#)
 - summary.whatif, [7](#)
- * **regression**
 - whatif, [8](#)

detectCores, [11](#)

peacecf, [2](#)

peacecf, [3](#)

plot.whatif, [3](#), [6–8](#), [12](#)

print.summary.whatif, [4](#), [5](#), [7](#), [8](#), [12](#)

print.whatif, [4](#), [6](#), [6](#), [8](#), [12](#)

summary.whatif, [4](#), [6](#), [7](#), [7](#), [12](#)

whatif, [4](#), [6–8](#), [8](#)