

# Package: wildwyong (via r-universe)

November 21, 2024

**Type** Package

**Title** Westfall-Young adjusted p-values for objects linear models via a wild bootstrap

**Version** 0.1.4

**Author** Alexander Fischer

**Maintainer** Alexander Fischer <alexander-fischer1801@t-online.de>

**Description** Implements Westfall-Young corrected p-values for objects of type 'fixest' and 'fixest\_multi' via a wild (cluster) bootstrap.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.0

**Imports** fixest, fwildclusterboot, dreamerr, wildr wolf

**Suggests** covr, testthat (>= 3.0.0), RStata, MASS

**Remotes** s3alfisc/fwildclusterboot, s3alfisc/wildr wolf

**Config/testthat/edition** 3

**Roxygen** list(markdown = TRUE)

**Repository** <https://s3alfisc.r-universe.dev>

**RemoteUrl** <https://github.com/s3alfisc/wildwyong>

**RemoteRef** HEAD

**RemoteSha** b49751c9c5c0e52e3d8ee0f3cbbd1312f9ad040d

## Contents

summary.wyong	2
wyong	2

<b>Index</b>	<b>5</b>
--------------	----------

---

summary.wyoung	<i>Summary method for objects of type wyoung</i>
----------------	--

---

**Description**

Summary method for objects of type wyoung

**Usage**

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

**Arguments**

object	An object of type wyoung
digits	Rounding of digits
...	misc. function arguments

---

wyoung	<i>Westfall-Young multiple hypotheses adjusted p-values</i>
--------	---

---

**Description**

Function implements the Westfall-Young multiple hypothesis correction procedure for objects of type `fixest_multi` (`fixest_multi` are objects created by `fixest::feols()` that use `feols()` multiple-estimation interface).

**Usage**

```
wyoung(
  models,
  param,
  B,
  R = NULL,
  r = 0,
  p_val_type = "two-tailed",
  weights_type = "rademacher",
  seed = NULL,
  engine = "R",
  nthreads = 1,
  bootstrap_type = NULL,
  ...
)
```

**Arguments**

<code>models</code>	An object of type <code>fixest_multi</code> or a list of objects of type <code>fixest</code>
<code>param</code>	The regression parameter to be tested
<code>B</code>	The number of bootstrap iterations
<code>R</code>	Hypothesis Vector giving linear combinations of coefficients. Must be either NULL or a vector of the same length as <code>param</code> . If NULL, a vector of ones of length <code>param</code> .
<code>r</code>	A numeric. Shifts the null hypothesis $H_0: \text{param} = r$ vs $H_1: \text{param} \neq r$
<code>p_val_type</code>	Character vector of length 1. Type of hypothesis test By default "two-tailed". Other options include "equal-tailed", ">" and "<".
<code>weights_type</code>	character or function. The character string specifies the type of bootstrap to use: One of "rademacher", "mammen", "norm" and "webb". Alternatively, type can be a function(n) for drawing wild bootstrap factors. "rademacher" by default. For the Rademacher distribution, if the number of replications <code>B</code> exceeds the number of possible draw combinations, $2^{(\text{number of clusters})}$ , then <code>boottest()</code> will use each possible combination once (enumeration).
<code>seed</code>	Integer. Sets the random seed. NULL by default.
<code>engine</code>	Should the wild cluster bootstrap run via <code>fwildclusterboot</code> 's R implementation or via <code>WildBootTests.jl</code> ? 'R' by default. The other option is 'WildBootTests.jl'.
<code>nthreads</code>	Integer. The number of threads to use.
<code>bootstrap_type</code>	Either "11", "13", "31", "33", or "fnw11".
<code>...</code>	additional function values passed to the bootstrap function.

**Value**

An object of type `wyoung`

**References**

- Westfall, Peter H., and S. Stanley Young. Resampling-based multiple testing: Examples and methods for p-value adjustment. Vol. 279. John Wiley & Sons, 1993.
- Clarke, Romano & Wolf (2019), STATA Journal. IZA working paper: <https://ftp.iza.org/dp12845.pdf>

**Examples**

```
library(fixest)
library(wildwyoung)

set.seed(12345)

N <- 1000
X1 <- rnorm(N)
Y1 <- 1 + 1 * X1 + rnorm(N)
Y2 <- 1 + 0.01 * X1 + rnorm(N)
Y3 <- 1 + 0.01 * X1 + rnorm(N)
Y4 <- 1 + 0.01 * X1 + rnorm(N)
```

```
B <- 999
# intra-cluster correlation of 0 for all clusters
cluster <- rep(1:50, N / 50)

data <- data.frame(Y1 = Y1,
                  Y2 = Y2,
                  Y3 = Y3,
                  Y4 = Y4,
                  X1 = X1,
                  cluster = cluster)

res <- feols(c(Y1, Y2, Y3) ~ X1, data = data, cluster = ~ cluster)
res_wyoung <- wyoung(models = res, param = "X1", B = B)
summary(res_wyoung)
```

# Index

summary . wyoung, [2](#)

wyoung, [2](#)