

Package: lfe2fixest (via r-universe)

October 31, 2024

Title Converts `lfe::felm()` To The `fixest::feols()` Equivalents

Version 0.0.1

Description What the package does (one paragraph).

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

URL <https://github.com/grantmcdermott/lfe2fixest>,
<http://grantmcdermott.com/lfe2fixest>

BugReports <https://github.com/grantmcdermott/lfe2fixest/issues>

Suggests fixest, knitr, lfe, modelsummary, rmarkdown

VignetteBuilder knitr

Repository <https://grantmcdermott.r-universe.dev>

RemoteUrl <https://github.com/grantmcdermott/lfe2fixest>

RemoteRef HEAD

RemoteSha 85f71f3f54c84bddeba6837da001239a7c94beb8

Contents

lfe2fixest	2
Index	5

lfe2fixest	<i>Converts lfe::felmm() commands into their fixest::feols() equivalents.</i>
------------	---

Description

Takes an R script with `lfe::felmm()` commands, converts them into their `fixest::feols()` equivalents, and then exports the resulting script to disk. Conversion is the only thing it does. Neither the input nor output script are run.

Usage

```
lfe2fixest(infile = NULL, outfile = NULL, verbose = FALSE, robust = FALSE)
```

```
felmm2feols(infile = NULL, outfile = NULL, verbose = FALSE, robust = FALSE)
```

Arguments

<code>infile</code>	An R script containing <code>lfe::felmm()</code> commands. Required.
<code>outfile</code>	File or connection to write the resulting R script (i.e. with <code>fixest::feols()</code> conversion) to. Can be the same as the input script, in which case the latter will obviously be overwritten. Can also be left blank in which case nothing will be written to disk and the output will simply be printed to screen.
<code>verbose</code>	Logical. Should the result be printed to screen. Defaults to <code>FALSE</code> unless <code>outfile</code> above is left blank.
<code>robust</code>	Logical. By default, iid errors will be used unless cluster variables have been specified in the <code>felmm()</code> formula(s). If users would like HC-robust standard errors, they should specify <code>TRUE</code> . Will be ignored if the <code>felmm</code> formula contains cluster variables, since the errors will then default to cluster-robust.

Details

`lfe::felmm()` and `fixest::feols()` provide "fixed-effects" estimation routines for high-dimensional data. Both methods are highly optimised, although `feols()` is newer and tends to be quite a bit faster. The syntax between these two methods is similar, if not quite offering drop-in replacement. This function aims to automate the conversion process; ignoring non-relevant arguments and differing options between the two, while doing its best to ensure that the resulting scripts will produce the same output.

Note that the conversion only handles (or attempts to handle) the actual model calls. No attempt is made to convert downstream objects or functions like regression table construction. Although, you will probably be okay if you use a modern table-generating package like `modelsummary`.

Other limitations include: (1) The function more or less implements a literal translation of the relevant `felmm` model. It doesn't support translation for some of the specialised syntax that `feols()` offers, e.g. multiple estimation and varying slopes. Everything should still work even if the literal translation doesn't yield all of the additional performance boosts and tricks that `feols()` offers. (2) The function assumes that users always provide a dataset in their model calls; i.e. regressions with

global variables are not supported. (3) Similarly, models that are constructed programmatically (e.g. with `Formula()`) are not supported.

Value

An R script.

See Also

[felm](#), [feols](#).

Examples

```
## Not run:
## Write a (deliberately messy) lfe script
lfe_string = "
library(lfe)
library(modelsummary)

## Our toy dataset
aq = airquality
names(aq) = c('y', 'x1', 'x2', 'x3', 'mnth', 'dy')

## Simple OLS (no FEs)
mod1 = felm(y ~ x1 + x2, aq)

## Add a FE and cluster variable
mod2 = felm(y ~ x1 + x2 |
            dy |
            0 |
            mnth, aq)

## Add a second cluster variable and some estimation options
mod3 = felm(y ~ x1 + x2 |
            dy |
            0 |
            dy + mnth,
            cmethod = 'reghdfe',
            exactDOF = TRUE,
            aq)

## IV reg with weights
mod4 = felm(y ~ 1 |
            dy |
            (x1 ~ x3) |
            mnth,
            weights = aq$x2,
            data = aq
            )

## Regression table
mods = list(mod1, mod2, mod3, mod4)
```

```
msummary(mods, gof_omit = 'Pseudo|Within|Log|IC', output = 'markdown')
"
writeLines(lfe_string, 'lfe_script.R')

## Covert to fixest equivalents
lfe2fixest('lfe_script.R') ## no output file provided, will print to screen
lfe2fixest('lfe_script.R', 'fixest_script.R') ## write converted script to disk

## Check equivalence

## First the lfe version
source('lfe_script.R', print.eval = TRUE)

## Then the fixest conversion
source('fixest_script.R', print.eval = TRUE)

## Clean up
file.remove(c('lfe_script.R', 'fixest_script.R'))

## End(Not run)
## Not run:
## For people that like options, there's the felm2feols() alias...
felm2fixest('another_felm_script.R')

## End(Not run)
```

Index

felm, [3](#)
felm2feols (lfe2fixest), [2](#)
feols, [3](#)
lfe2fixest, [2](#)