Package ‘wfe’

July 22, 2025

Type Package

Title Weighted Linear Fixed Effects Regression Models for Causal Inference

Version 1.9.1

Date 2019-04-17

Description Provides a computationally efficient way of fitting weighted linear fixed effects estimators for causal inference with various weighting schemes. Weighted linear fixed effects estimators can be used to estimate the average treatment effects under different identification strategies. This includes stratified randomized experiments, matching and stratification for observational studies, first differencing, and difference-in-differences. The package implements methods described in Imai and Kim (2017) ``When should We Use Linear Fixed Effects Regression Models for Causal Inference with Longitudinal Data?'', available at <https://imai.fas.harvard.edu/research/FEmatch.html>.

License GPL (>= 2)

Imports utils, arm, Matrix, MASS, methods

Depends R (>= 3.2.0)

Encoding UTF-8

LazyData true

BugReports https://github.com/insongkim/wfe/issues

NeedsCompilation yes

Author In Song Kim [aut, cre],
Kosuke Imai [aut]

Maintainer In Song Kim <insong@mit.edu>

Repository CRAN

Date/Publication 2019-04-17 21:50:03 UTC

Contents

pwfe *Fitting the Weighted Fixed Effects Model with Propensity Score Weighting*

Description

`pwfe` is used to fit weighted fixed effects model for causal inference after transforming outcome variable based on estimated propensity score. `pwfe` also derives the regression weights for different causal quantity of interest.

Usage

```
pwfe(formula, treat = "treat.name", outcome, data, pscore = NULL,
     unit.index, time.index = NULL, method = "unit", within.unit = TRUE,
     qoi = c("ate", "att"), estimator = NULL, C.it = NULL,
     White = TRUE, White.alpha = 0.05,
     hetero.se = TRUE, auto.se = TRUE, unbiased.se = FALSE,
     verbose = TRUE)
```

Arguments

`formula` a symbolic description of the model for estimating propensity score. The formula should not include dummmies for fixed effects. The details of model specifications are given under 'Details'.

`treat` a character string indicating the name of treatment variable used in the models. The treatment should be binary indicator (integer with 0 for the control group and 1 for the treatment group).

`outcome` a character string indicating the name of outcome variable.

`data` data frame containing the variables in the model.

`pscore` an optional character string indicating the name of estimated propensity score. Note that pre-specified propensity score should be bounded away from zero and one.

`unit.index` a character string indicating the name of unit variable used in the models. The index of unit should be factor.

`time.index` a character string indicating the name of time variable used in the models. The index of time should be factor.

`method` method for weighted fixed effects regression, either `unit` for unit fixed effects; `time` for time fixed effects. The default is `unit`.

`within.unit` a logical value indicating whether propensity score is estimated within unit. The default is `TRUE`.

`qoi` one of `"ate"` or `"att"`. The default is `"ate"`. `"fd"` and `"did"` are not compatible with `pwfe`.

`estimator` an optional character string `"fd"` indicating whether the first-difference estimator will be used.

`C.it` an optional non-negative numeric vector specifying relative weights for each unit of analysis.

`White` a logical value indicating whether White misspecification statistics should be calculated. The default is `TRUE`.

`White.alpha` level of functional specification test. See White (1980) and Imai . The default is `0.05`.

`hetero.se` a logical value indicating whether heteroskedasticity across units is allowed in calculating standard errors. The default is `TRUE`.

`auto.se` a logical value indicating whether arbitrary autocorrelation is allowed in calculating standard errors. The default is `TRUE`.

`unbiased.se` logical. If `TRUE`, bias-asjusted heteroskedasticity-robust standard errors are used. See Stock and Watson (2008). Should be used only for balanced panel. The default is `FALSE`.

`verbose` logical. If `TRUE`, helpful messages along with a progress report of the weight calculation are printed on the screen. The default is `TRUE`.

Details

To fit the weighted unit (time) fixed effects model with propensity score weighting, use the syntax for the formula, `~ x1 + x2`, where `x1` and `x2` are unit (time) varying covariates.

One can provide his/her own estimated `pscore` which can be used to transform the outcome varialbe. If so, one does not need to specify `formula`.

If `pscore` is not provided, `bayesglm` will be used to estimate propensity scores. If `within.unit = TRUE`, propensity score will be separately estimated within time (unit) when `method` is `unit` (`time`). Otherwise, propensity score will be estimated on entire data at once.

The estimated propensity scores will be used to transform the `outcome` variable as described in Imai and Kim (2018).

`pwfe` calculates weights based on different underlying causal quantity of interest: Average Treatment Effect (`qoi = "ate"`) or Average Treatment Effect for the Treated (`qoi = "att"`).

One can further set estimating methods: First-Difference (`estimator ="fd"`) or Difference-in-differences (`estimator = "did"`).

To specify different ex-ante weights for each unit of analysis, use non-negative weights `C.it`. For instance, using the survey weights for `C.it` enables the estimation fo the average treatement effect for the target population.

Value

pwfe returns an object of class "pwfe", a list that contains the components listed below.

The function summary (i.e., summary.pwfe) can be used to obtain a table of the results.

`coefficients`	a named vector of coefficients
`residuals`	the residuals, that is respons minus fitted values
`df`	the degree of freedom
`W`	weight matrix calculated from the model. Row and column indices can be found from unit.name, time.name.
`call`	the matched call
`causal`	causal quantity of interest
`estimator`	the estimating method
`unit.name`	a vector containing unique unit names
`unit.index`	a vector containing unique unit index number
`time.name`	a vector containing unique time names
`time.index`	a vector containing unique time index number
`method`	call of the method used
`vcov`	the variance covariance matrix
`White.alpha`	the alpha level for White specification test
`White.pvalue`	the p-value for White specification test
`White.stat`	the White statistics
`x`	the design matrix
`y`	the response vector
`mf`	the model frame

Author(s)

In Song Kim, Massachusetts Institute of Technology, <insong@mit.edu> and Kosuke Imai, Princeton University, <imai@harvard.edu>

References

Imai, Kosuke and In Song Kim. (2018) “When Should We Use Unit Fixed Effects Regression Models for Causal Inference with Longitudinal Data?" American Journal of Political Science, Forthcoming.

Stock, James and Mark Watson. (2008) “Heteroskedasticity-Robust Standard Errors for Fixed Effect Panel Data Regression” Econometrica, 76, 1.

White, Halbert. (1980) ‘Using Least Squares to Approximate Unknown Regression Functions.” International Economic Review, 21, 1, 149–170.

See Also

wfe for fitting weighted fixed effect models.

Examples

```
### NOTE: this example illustrates the use of wfe function with randomly
### generated panel data with arbitrary number of units and time.

## generate panel data with number of units = N, number of time = Time
## Not run:

N <- 10 # number of distinct units
Time <- 15 # number of distinct time

## generate treatment variable
treat <- matrix(rbinom(N*Time, size = 1, 0.25), ncol = N)
## make sure at least one observation is treated for each unit
while ((sum(apply(treat, 2, mean) == 0) > 0) | (sum(apply(treat, 2, mean) == 1) > 0) |
       (sum(apply(treat, 1, mean) == 0) > 0) | (sum(apply(treat, 1, mean) == 1) > 0)) {
  treat <- matrix(rbinom(N*Time, size = 1, 0.25), ncol = N)
}
treat.vec <- c(treat)

## unit fixed effects
alphai <- rnorm(N, mean = apply(treat, 2, mean))

## geneate two random covariates
x1 <- matrix(rnorm(N*Time, 0.5,1), ncol=N)
x2 <- matrix(rbeta(N*Time, 5,1), ncol=N)
pscore <- matrix(runif(N*Time, 0,1), ncol=N)
x1.vec <- c(x1)
x2.vec <- c(x2)
pscore <- c(pscore)

## generate outcome variable
y <- matrix(NA, ncol = N, nrow = Time)
for (i in 1:N) {
    y[, i] <- alphai[i] + treat[, i] + x1[,i] + x2[,i] + rnorm(Time)
}
y.vec <- c(y)

## generate unit and time index
unit.index <- rep(1:N, each = Time)
time.index <- rep(1:Time, N)

Data.str <- as.data.frame(cbind(y.vec, treat.vec, unit.index, x1.vec, x2.vec))
colnames(Data.str) <- c("y", "tr", "strata.id", "x1", "x2")

Data.obs <- as.data.frame(cbind(y.vec, treat.vec, unit.index, time.index, x1.vec, x2.vec, pscore))
colnames(Data.obs) <- c("y", "tr", "unit", "time", "x1", "x2", "pscore")


############################################################
# Example 1: Stratified Randomized Experiments
############################################################
```

```
## run the weighted fixed effect regression with strata fixed effect.
## Note: the quantity of interest is Average Treatment Effect ("ate")
## and the standard errors allow heteroskedasticity and arbitrary
## autocorrelation.


### Average Treatment Effect
ps.ate <- pwfe(~ x1+x2, treat = "tr", outcome = "y", data = Data.str,
               unit.index = "strata.id", method = "unit", within.unit = TRUE,
               qoi = "ate", hetero.se=TRUE, auto.se=TRUE)
## summarize the results
summary(ps.ate)

### Average Treatment Effect for the Treated
ps.att <- pwfe(~ x1+x2, treat = "tr", outcome = "y", data = Data.str,
               unit.index = "strata.id", method = "unit", within.unit = TRUE,
               qoi = "att", hetero.se=TRUE, auto.se=TRUE)
## summarize the results
summary(ps.att)


############################################################
# Example 2: Observational Studies with Unit Fixed-effects
############################################################

## run the weighted fixed effect regression with unit fixed effect.
## Note: the quantity of interest is Average Treatment Effect ("ate")
## and the standard errors allow heteroskedasticity and arbitrary
## autocorrelation.

### Average Treatment Effect
ps.obs <- pwfe(~ x1+x2, treat = "tr", outcome = "y", data = Data.obs,
               unit.index = "unit", time.index = "time",
               method = "unit", within.unit = TRUE,
               qoi = "ate", hetero.se=TRUE, auto.se=TRUE)

## summarize the results
summary(ps.obs)

## extracting weigths
summary(ps.obs)$Weights

### Average Treatment Effect with First-difference

ps.fd <- pwfe(~ x1+x2, treat = "tr", outcome = "y", data = Data.obs,
              unit.index = "unit", time.index = "time",
              method = "unit", within.unit = TRUE,
              qoi = "ate", estimator = "fd", hetero.se=TRUE, auto.se=TRUE)

## summarize the results
summary(ps.fd)
```

```
############################################################
# Example 3: Estimation with pre-specified propensity score
############################################################

### Average Treatment Effect with Pre-specified Propensity Scores

mod.ps <- pwfe(treat = "tr", outcome = "y", data = Data.obs, pscore = "pscore",
               unit.index = "unit", time.index = "time",
               method = "unit", within.unit = TRUE,
               qoi = "ate", hetero.se=TRUE, auto.se=TRUE)

## summarize the results
summary(mod.ps)

## End(Not run)
```

wfe *Fitting the Weighted Fixed Effects Model for Causal Inference*

Description

`wfe` is used to fit weighted fixed effects model for causal inference. `wfe` also derives the regression weights for different causal quantity of interest.

Usage

```
wfe(formula, data, treat = "treat.name",
    unit.index, time.index = NULL, method = "unit",
    dyad1.index = NULL, dyad2.index = NULL,
    qoi = "ate", estimator = NULL, C.it = NULL,
    hetero.se = TRUE, auto.se = TRUE,
    dyad.se = FALSE,
    White = TRUE, White.alpha = 0.05,
    verbose = TRUE, unbiased.se = FALSE, unweighted = FALSE,
    store.wdm = FALSE, maxdev.did = NULL,
    tol = sqrt(.Machine$double.eps))
```

Arguments

`formula` a symbolic description of the model to be fitted. The formula should not include dummmies for fixed effects. The details of model specifications are given under 'Details'.

`data` data frame containing the variables in the model.

`treat` a character string indicating the name of treatment variable used in the models. The treatment should be binary indicator (integer with 0 for the control group and 1 for the treatment group).

`unit.index` a character string indicating the name of unit variable used in the models. The index of unit should be factor.

`time.index` a character string indicating the name of time variable used in the models. The index of time should be factor.

`method` method for weighted fixed effects regression, either `unit` for unit fixed effects; `time` for time fixed effects. The default is `unit`. For two-way weighted fixed effects regression models, set method to the default value `unit`.

`dyad1.index` a character string indicating the variable name of first unit of a given dyad. The default is `NULL`. This is required to calculate robust standard errors with dyadic data.

`dyad2.index` a character string indicating the variable name of second unit of a given dyad. The default is `NULL`. This is required to calculate robust standard errors with dyadic data.

`qoi` one of `"ate"` or `"att"`. The default is `"ate"`. If set to `"att"` in implementing `"fd"` and `"did"` estimators, the comparison of the treated observation is restricted to the control observation from the previous time period but not with the control observation from the next time period.

`estimator` an optional character string indicating the estimating method. One of `"fd"`, `"did"`, or `"Mdid"`. `"fd"` is for First-Difference Design. `"did"` is for multi-period Difference-in-Differences design. The default is `NULL`. Setting estimator to be `"Mdid"` implements the Difference-in-Differences design with Matching on the pretreatment outcome variables.

`C.it` an optional non-negative numeric vector specifying relative weights for each unit of analysis. If not specified, the weights will be calculated based on the estimator and quantity of interest.

`hetero.se` a logical value indicating whether heteroskedasticity across units is allowed in calculating standard errors. The default is `TRUE`.

`auto.se` a logical value indicating whether arbitrary autocorrelation is allowed in calculating standard errors. The default is `TRUE`.

`dyad.se` a logical value indicating whether correlations across dyads exist. The default is `FALSE`.

`White` a logical value indicating whether White misspecification statistics should be calculated. The default is `TRUE`.

`White.alpha` level of functional specification test. See White (1980) and Imai and Kim (2018). The default is `0.05`.

`verbose` logical. If `TRUE`, helpful messages along with a progress report of the weight calculation are printed on the screen. The default is `TRUE`.

`unbiased.se` logical. If `TRUE`, bias-asjusted heteroskedasticity-robust standard errors are used. See Stock and Watson (2008). Should be used only for balanced panel. The default is `FALSE`.

`unweighted` logical. If `TRUE`, standard unweighted fixed effects model is estimated. The default is `FALSE`. Note: users do not need to specify `qoi` when unweighted=`TRUE`. For standard two-way fixed effects model (unit and time), set estimator=`"did"` and unweighted=`"TRUE"`.

`store.wdm` logical. If `TRUE`, weighted demeaned dataframe will be stored. The default is `FALSE`.

`maxdev.did` an optional positive numeric value specifying the maximum deviation in pre-treatment outcome when `"Mdid"` is implemented. The default is `NULL`, which implements nearest-neighbor matching.

`tol` a relative tolerance to detect zero singular values for generalized inverse. The default is sqrt(.Machine$double.eps)

Details

To fit the weighted unit (time) fixed effects model, use the syntax for the formula, `y ~ x1 + x2`, where `y` is a dependent variable and `x1` and `x2` are unit (time) varying covariates.

`wfe` calculates weights based on different underlying causal quantity of interest: Average Treatment Effect (`qoi = "ate"`) or Average Treatment Effect for the Treated (`qoi = "att"`).

One can further set estimating methods: First-Difference (`estimator ="fd"`) or Difference-in-differences (`estimator = "did"`). For the two-way fixed effects model, set `estimator = "did"`

To specify different ex-ante weights for each unit of analysis, use non-negative weights `C.it`. For instance, using the survey weights for `C.it` enables the estimation fo the average treatement effect for the target population.

An object of class "wfe" contains vectors of unique unit(time) names and unique unit(time) indices.

Value

`wfe` returns an object of class "wfe", a list that contains the components listed below.

The function `summary` (i.e., `summary.wfe`) can be used to obtain a table of the results.

`coefficients` a named vector of coefficients

`residuals` the residuals, that is respons minus fitted values

`df` the degree of freedom

`W` a dataframe containing unit and time indices along with the weights used for the observation. If method=`unit`, integer numbers corresponding to the order of input data will be used for generating time index.

`Num.nonzero` Number of observations with non-zero weights

`call` the matched call

`causal` causal quantity of interest

`estimator` the estimating method

`units` a dataframe containing unit names used for `W`

`times` a dataframe containing time names used for `W`

`method` call of the method used

`vcov` the variance covariance matrix

`White.alpha` the alpha level for White specification test

`White.pvalue` the p-value for White specification test

`White.stat` the White statistics

`X`	the design matrix
`Y`	the response vector
`X.wdm`	the demeaned design matrix
`Y.wdm`	the demeaned response vector
`mf`	the model frame where the last column is the weights used for the analysis

Author(s)

In Song Kim, Massachusetts Institute of Technology, <`insong@mit.edu`> and Kosuke Imai, Princeton University, <`imai@harvard.edu`>

References

Imai, Kosuke and In Song Kim. (2018) “When Should We Use Unit Fixed Effects Regression Models for Causal Inference with Longitudinal Data?" American Journal of Political Science, Forthcoming.

Aronow, Peter M., Cyrus Samii, and Valentina A. Assenova (2015) “Cluster–robust Variance Estimation for Dyadic Data." Political Analysis 23, no. 4, 564–577.

Stock, James and Mark Watson. (2008) “Heteroskedasticity-Robust Standard Errors for Fixed Effect Panel Data Regression” Econometrica, 76, 1.

White, Halbert. (1980) “Using Least Squares to Approximate Unknown Regression Functions.” International Economic Review, 21, 1, 149–170.

See Also

`pwfe` for fitting weighted fixed effects models with propensity score weighting

Examples

```
### NOTE: this example illustrates the use of wfe function with randomly
### generated panel data with arbitrary number of units and time.

## generate panel data with number of units = N, number of time = Time
N <- 10 # number of distinct units
Time <- 15 # number of distinct time

## treatment effect
beta <- 1

## generate treatment variable
treat <- matrix(rbinom(N*Time, size = 1, 0.25), ncol = N)
## make sure at least one observation is treated for each unit
while ((sum(apply(treat, 2, mean) == 0) > 0) | (sum(apply(treat, 2, mean) == 1) > 0) |
       (sum(apply(treat, 1, mean) == 0) > 0) | (sum(apply(treat, 1, mean) == 1) > 0)) {
  treat <- matrix(rbinom(N*Time, size = 1, 0.25), ncol = N)
}
treat.vec <- c(treat)

## unit fixed effects
```

```
alphai <- rnorm(N, mean = apply(treat, 2, mean))

## geneate two random covariates
x1 <- matrix(rnorm(N*Time, 0.5,1), ncol=N)
x2 <- matrix(rbeta(N*Time, 5,1), ncol=N)
x1.vec <- c(x1)
x2.vec <- c(x2)
## generate outcome variable
y <- matrix(NA, ncol = N, nrow = Time)
for (i in 1:N) {
    y[, i] <- alphai[i] + treat[, i] + x1[,i] + x2[,i] + rnorm(Time)
}
y.vec <- c(y)

## generate unit and time index
unit.index <- rep(1:N, each = Time)
time.index <- rep(1:Time, N)

Data.str <- as.data.frame(cbind(y.vec, treat.vec, unit.index, x1.vec, x2.vec))
colnames(Data.str) <- c("y", "tr", "strata.id", "x1", "x2")

Data.obs <- as.data.frame(cbind(y.vec, treat.vec, unit.index, time.index, x1.vec, x2.vec))
colnames(Data.obs) <- c("y", "tr", "unit", "time", "x1", "x2")

############################################################
# Example 1: Stratified Randomized Experiments
############################################################

## run the weighted fixed effect regression with strata fixed effect.
## Note: the quantity of interest is Average Treatment Effect ("ate")
## and the standard errors allow heteroskedasticity and arbitrary
## autocorrelation.


### Average Treatment Effect
mod.ate <- wfe(y~ tr+x1+x2, data = Data.str, treat = "tr",
               unit.index = "strata.id", method = "unit",
               qoi = "ate", hetero.se=TRUE, auto.se=TRUE)
## summarize the results
summary(mod.ate)

### Average Treatment Effect for the Treated
mod.att <- wfe(y~ tr+x1+x2, data = Data.str, treat = "tr",
               unit.index = "strata.id", method = "unit",
               qoi = "att", hetero.se=TRUE, auto.se=TRUE)
## summarize the results
summary(mod.att)


############################################################
# Example 2: Observational Studies with Unit Fixed-effects
############################################################
```

```
## run the weighted fixed effect regression with unit fixed effect.
## Note: the quantity of interest is Average Treatment Effect (”ate”)
## and the standard errors allow heteroskedasticity and arbitrary
## autocorrelation.

mod.obs <- wfe(y~ tr+x1+x2, data = Data.obs, treat = ”tr”,
               unit.index = ”unit”, time.index = ”time”, method = ”unit”,
               qoi = ”ate”, hetero.se=TRUE, auto.se=TRUE,
               White = TRUE, White.alpha = 0.05)

## summarize the results
summary(mod.obs)

## extracting weigths
summary(mod.obs)$W


## Not run:
###################################################################
# Example 3: Observational Studies with differences-in-differences
###################################################################

## run difference-in-differences estimator.
## Note: the quantity of interest is Average Treatment Effect (”ate”)
## and the standard errors allow heteroskedasticity and arbitrary
## autocorrelation.

mod.did <- wfe(y~ tr+x1+x2, data = Data.obs, treat = ”tr”,
               unit.index = ”unit”, time.index = ”time”, method = ”unit”,
               qoi = ”ate”, estimator =”did”, hetero.se=TRUE, auto.se=TRUE,
               White = TRUE, White.alpha = 0.05, verbose = TRUE)

## summarize the results
summary(mod.did)

## extracting weigths
summary(mod.did)$W

#########################################################################
# Example 4: DID with Matching on Pre-treatment Outcomes
#########################################################################

## implements matching on pre-treatment outcomes where the maximum
## deviation is specified as 0.5

mod.Mdid <- wfe(y~ tr+x1+x2, data = Data.obs, treat = ”tr”,
               unit.index = ”unit”, time.index = ”time”, method = ”unit”,
               qoi = ”ate”, estimator =”Mdid”, hetero.se=TRUE, auto.se=TRUE,
               White = TRUE, White.alpha = 0.05, maxdev.did = 0.5, verbose = TRUE)

## summarize the results
summary(mod.Mdid)
```

```
## Note: setting the maximum deviation to infinity (or any value
## bigger than the maximum pair-wise difference in the outcome) will
## return the same result as Example 3.

dev <- 1000+max(Data.obs$y)-min(Data.obs$y)
mod.did2 <- wfe(y~ tr+x1+x2, data = Data.obs, treat = "tr",
               unit.index = "unit", time.index = "time", method = "unit",
               qoi = "ate", estimator ="Mdid", hetero.se=TRUE, auto.se=TRUE,
               White = TRUE, White.alpha = 0.05, maxdev.did = dev, verbose = TRUE)

## summarize the results
summary(mod.did2)
mod.did2$coef[1] ==  mod.did$coef[1]

## End(Not run)
```

Index