Package 'rsmatrix'

August 27, 2020

Title Matrices for Repeat-Sales Price Indexes

Version 0.1.0

Description A small package for calculating the matrices in Shiller (1991, <doi:10.1016/S1051-1377(05)80028-2>) that serve as the foundation for many repeat-sales price indexes.

Depends R (>= 4.0)

Imports Matrix, methods, stats

License MIT + file LICENSE

Encoding UTF-8

URL https://github.com/marberts/rsmatrix

LazyData true

NeedsCompilation no

Author Steve Martin [aut, cre, cph]

Maintainer Steve Martin <stevemartin041@gmail.com>

Repository CRAN

Date/Publication 2020-08-27 14:50:02 UTC

R topics documented:

`rsmatrix-package` *Matrices for Repeat-Sales Price Indexes*

Description

A small package for calculating the matrices in Shiller (1991) that serve as the foundation for many repeat-sales price indexes.

Details

Most repeat-sales price indexes used in practice require the matrices in Shiller (1991, sections I-II), e.g., S&P's Case-Shiller index, Statistics Canada's RPPI. The `rs_matrix()` function produces a function to construct these matrices. In most cases data need to be structured as sales pairs, which can be done with the `rs_pair()` function.

Note

The `'McSpatial'` package has some functionality for making repeat-sales indices. The functions in this package build off of those in the `'rsi'` package in Kirby-McGregor and Martin (2019), which also gives a good background on the theory of repeat-sales indexes.

Author(s)

Maintainer: Steve Martin <stevemartin041@gmail.com>

References

ILO, IMF, OECD, UN, World Bank, Eurostat. (2013). *Handbook on Residential Property Prices Indices (RPPIs)*. Eurostat.

Kirby-McGregor, M., and Martin, S. (2019). An R package for calculating repeat-sale price indices. *Romanian Statistical Review*, 3:17-33.

Shiller, R. J. (1991). Arithmetic repeat sales price estimators. *Journal of Housing Economics*, 1(1):110-126.

See Also

https://github.com/marberts/rsmatrix

`rs_matrix` *Shiller's repeat-sales matrices*

Description

Create a function to compute the Z, X, y, and Y matrices in Shiller (1991, sections I-II) from sales-pair data to calculate a repeat-sales price index.

Usage

```
rs_matrix(t2, t1, p2, p1, f = NULL, sparse = FALSE)
```

Arguments

`t2, t1`	A pair of vectors giving the time period of the second and first sale, respectively. Usually a vector of dates.
`p2, p1`	A pair of numeric vectors giving the price of the second and first sale, respectively.
`f`	An optional factor the same length as `t1` and `t2`, or a vector to be turned into a factor, that is used to group `t2` and `t1`.
`sparse`	Should sparse matrices from the `'Matrix'` package be used in the calculation (faster for large datasets), or regular dense matrices?

Details

The function returned by `rs_matrix()` computes a generalization of the matrices in Shiller (1991, sections I-II) that are applicable to grouped data. This is useful for calculating separate indexes for many, say, cities without needing an explicit loop.

The Z, X, and Y matrices are not well defined if either `t1` or `t2` have missing values, and an error is thrown if any inputs have missing values. Similarly, it should always be the case that `t2 > t1`, otherwise a warning is given.

Value

A function that takes a single argument naming the desired matrix. It returns one of two matrices (Z and X) or two vectors (y and Y), either regular matrices if `sparse = FALSE`, or sparse matrices of class `'dgCMatrix'` if `sparse = TRUE`.

References

Shiller, R. J. (1991). Arithmetic repeat sales price estimators. *Journal of Housing Economics*, 1(1):110-126.

See Also

`rs_pair` for turning sales data into sales pairs.

Examples

```
# Make some data
x <- data.frame(date = c(3, 2, 3, 2, 3, 3),
                date_prev = c(1, 1, 2, 1, 2, 1),
                price = 6:1,
                price_prev = 1)

# Calculate matrices
mat <- with(x, rs_matrix(date, date_prev, price, price_prev))
Z <- mat("Z") # Z matrix
X <- mat("X") # X matrix
y <- mat("y") # y vector
Y <- mat("Y") # Y vector

# Calculate the GRS index in Bailey, Muth, and Nourse (1963)
b <- solve(crossprod(Z), crossprod(Z, y))[, 1]
# or b <- qr.coef(qr(Z), y)
(grs <- exp(b) * 100)

# Covariance matrix
vcov <- rs_var(y - Z %*% b, Z)
sqrt(diag(vcov)) * grs # delta method

# Calculate the ARS index in Shiller (1991)
b <- solve(crossprod(Z, X), crossprod(Z, Y))[, 1]
# or b <- qr.coef(qr(crossprod(Z, X)), crossprod(Z, Y))
(ars <- 100 / b)

# Standard errors
vcov <- rs_var(Y - X %*% b, Z, X)
sqrt(diag(vcov)) * ars^2 / 100 # delta method

# Works with grouped data
x <- data.frame(date = c(3, 2, 3, 2),
                date_prev = c(2, 1, 2, 1),
                price = 4:1,
                price_prev = 1,
                group = c("a", "a", "b", "b"))

mat <- with(x, rs_matrix(date, date_prev, price, price_prev, group))
b <- solve(crossprod(mat("Z"), mat("X")), crossprod(mat("Z"), mat("Y")))[, 1]
100 / b
```

`rs_var` *Robust variance matrix for repeat-sales index*

Description

Compute a cluster-robust variance matrix for a linear regression, with or without instruments, where clustering occurs along one dimension. Useful for calculating a variance matrix when a regression is calculated manually.

Usage

```
rs_var(u, Z, X = Z, ids = seq_len(nrow(X)), df)
```

Arguments

u	An nx1 vector of residuals from a linear regression.
Z	An nxk matrix of instruments.
X	An nxk matrix of covariates.
ids	A vector of length n that groups observations in u. By default each observation belongs to its own group.
df	An optional degrees of freedom correction. Default is Stata's degrees of freedom correction.

Details

This function calculates the standard robust variance matrix for a linear regression, as in Manski (1988, section 8.1.2) or White (2001, Theorem 6.3); that is, (Z'X)^-1 V (X'Z)^-1. It is useful when a regression is calculated by hand. This generalizes the variance matrix proposed by Shiller (1991, section II) when a property sells more than twice.

This function gives the same result as `vcovHC(x,type = 'sss',cluster = 'group')` from the `'plm'` package.

Value

A kxk matrix.

References

Manski, C. (1988). *Analog Estimation Methods in Econometrics*. Chapman and Hall.

Shiller, R. J. (1991). Arithmetic repeat sales price estimators. *Journal of Housing Economics*, 1(1):110-126.

White, H. (2001). *Asymptotic Theory for Econometricians* (revised edition). Emerald Publishing.

Examples

```
# Makes some groups in mtcars
mtcars$clust<- letters[1:4]

# Matrices for regression
x <- model.matrix(~ cyl + disp, mtcars)
y <- matrix(mtcars$mpg)

# Regression coefficients
b <- solve(crossprod(x), crossprod(x, y))

# Residuals
r <- y - x %*% b
```

```
# Robust variance matrix
vcov <- rs_var(r, x, ids = mtcars$clust)

## Not run:
# Same as plm
library(plm)
mdl <- plm(mpg ~ cyl + disp, mtcars, model = 'pooling', index = 'clust')
vcov2 <- vcovHC(mdl, type = 'sss', cluster = 'group')
vcov - vcov2

## End(Not run)
```

`sales pairs` *Make/unmake sales pairs*

Description

Turn a long dataset with repeat sales information into a dataset of sales pairs, or turn a wide dataset of sales pairs into a long data set with a single price and date variable.

Usage

```
rs_pair(x, id = "id", date = "date", price = "price")

rs_unpair(x, id = "id", date = "date", date_prev = "date_prev",
          price = "price", price_prev = "price_prev")
```

Arguments

`x`	A data.frame, or something that can be coerced into one.
`id`	Column name in `x` for a unique id.
`date`	Column name in `x` for a sales date.
`price`	Column name in `x` for a sales price.
`date_prev`	Column name in `x` for a previous sales date.
`price_prev`	Column name in `x` for a previous sales price.

Details

Making sales pairs takes consecutive sales for each `id` and turns them into pairs of consecutive sales. This is equivalent to first-differencing the data.

Observations with missing values and duplicated observations are not removed prior to creating sales pairs. Making sales pairs is generally not well defined with missing and duplicated values, and these should be removed prior to making pairs.

Extra columns with data that do not varying over time for each id are preserved when making pairs.

When unmaking sales pairs, the previous price and date for the earliest sale are extracted for each id, then concatenated (row-wise) back into x. Extra columns are expanded in the long format.

As with making sales pairs, un-making sales pairs is generally not well defined with missing and duplicated values, and these should be removed prior to un-making pairs.

Value

A data.frame.

See Also

rs_matrix for using sales pairs to make a repeat-sales index.

Examples

```
# Turn into a 'wide' dataset of sales pairs
x <- data.frame(id = c(1, 1, 1, 2, 2),
                date = c(1, 2, 3, 1, 3),
                price = c(1, 3, 2, 1, 1))
rs_pair(x)

# Turn into a 'long' dataset of sales pairs
x <- data.frame(id = c(1, 1, 2),
                date = c(2, 3, 3),
                date_prev = c(1, 2, 1),
                price = 3:1,
                price_prev = c(1, 3, 1))
rs_unpair(x)
```

Index