FCI R Documentation

August 2018

Contents

Chapter 1

Algorithm Outline

1.1 Preparation

We start with a data matrix starting at time t

$$X_T^0 = [\mathbf{x}_{T,1}^0, \mathbf{x}_{T,2}^0 ... \mathbf{x}_{T,n}^0]$$

where $\mathbf{x}_{T,i}^0$ is the $i^{th}, i = 1, ...n$ independent variable of T periods starting at time 0 and $x_{t,i}$ is the t^{th} row and i^{th} column of X_T^0,

$$\mathbf{x}_{T,i}^0 = \begin{bmatrix} x_{1,i} \\ x_{2,i} \\ \vdots \\ x_{T,i} \end{bmatrix}$$

We also have $\mathbf{y}_T$ as the dependent variable of length T,

$$\mathbf{y}_T = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix}$$

1.1.1 Lags

We first create q lags for each $\mathbf{x}_{T,i}^0$. Let $\mathbf{x}_{T-q,i}^0$ be the lag q version of $\mathbf{x}_{T,i}^0$, where

$$\mathbf{x}_{T-q,i}^0 = \begin{bmatrix} - \\ \vdots \\ - \\ x_{1,i} \\ x_{2,i} \\ \vdots \\ x_{T-q,i} \end{bmatrix} \begin{array}{l} \left.\vphantom{\begin{matrix} - \\ \vdots \\ - \end{matrix}}\right\} q \text{ empty entries} \\ \\ \\ \\ \\ \end{array}$$

1.1.2 Standardization

We then standardize each column. Let $\mathbf{x}_{T-q,i}$ be the standardized version of $\mathbf{x}_{T-q,i}^0$.

$$\mathbf{x}_{T-q,i} = \frac{\mathbf{x}_{T-q,i}^0 - \overline{\mathbf{x}_{T-q,i}^0}}{s_{\mathbf{x}_{T-q,i}^0}}$$

where $\overline{\mathbf{x}_{T-q,i}^0}$ and $s_{\mathbf{x}_{T-q,i}^0}$ are the sample mean and sd respectively.

1.2 LRFCI Construction

We work with lags 0 to q in the LRFCI construction. Let X_t be the independent data matrix.

$X_T = [\mathbf{x}_{T,1}, \mathbf{x}_{T-1,1}, ..., \mathbf{x}_{T-q,1}, \mathbf{x}_{T,2}, \mathbf{x}_{T-1,2}, ..., \mathbf{x}_{T-q,2}, ...\mathbf{x}_{T,n}, \mathbf{x}_{T-1,n}, ..., \mathbf{x}_{T-q,n}]$

and also denote X_{T+h} be the matrix of column i as $\mathbf{x}_{T+h,i}$

$$\mathbf{x}_{T+h,i} = \begin{bmatrix} x_{1,i} \\ x_{2,i} \\ \vdots \\ x_{T,i} \\ x_{T+1,i} \\ \vdots \\ x_{T+h,i} \end{bmatrix}$$

1.2.1 Calculation

The algorithm is as the following:

Algorithm 1 LRFCI Construction

for each increasing update interval u_j, $j \leftarrow 0$ to N **do**
    Standardize X_{T+u_j} by its mean and sd
    Compute the weight vector: $\mathbf{w}_{u_j} = X^T_{T+u_j}\mathbf{y}_{T+u_j}$
    Set $\mathbf{w}_{u_j} = \frac{\mathbf{w}_{u_j}}{|\mathbf{w}_{u_j}|}$
    for $i = 1$ to n **do**
        for $k = 0$ to q **do**
            if $|w_{i,k,u_j}| = \max_q |w_{i,q,u_j}|$ **then**
                $w_{i,k,u_j} = w_{i,k,u_j}$
            else
                $w_{i,k,u_j} = 0$
            end if
        end for
    end for
    Standardize X_{T+u_j+h} by the mean and sd of X_{T+u_j}
    Compute the first factor $\mathbf{f}_{T+u_j+h} = X_{T+u_j+h}\mathbf{w}_{u_j}$
    Standardize $\mathbf{f}_{T+u_j+h}$ by the mean and sd of $\mathbf{f}_{T+u_j}$
    Set LRFCI $\mathbf{f}^{lr}_{u_j} = \mathbf{f}_{T+u_j+h} \times s_{\mathbf{y}_{T+u_j}} + \overline{\mathbf{y}_{T+u_j}}$
end for

In **line 3 and 4**, a weight vector as the following is produced:

$$\mathbf{w}_{u_j} = \begin{bmatrix} w_{1,0,u_j} \\ w_{1,1,u_j} \\ \vdots \\ w_{1,q,u_j} \\ w_{2,0,u_j} \\ w_{2,1,u_j} \\ \vdots \\ w_{2,q,u_j} \\ \vdots \\ w_{n,0,u_j} \\ w_{n,1,u_j} \\ \vdots \\ w_{n,q,u_j} \end{bmatrix}$$

where $w_{i,q,t}$ is the weight for the i^{th} independent variable with lags q at time t. The weight for the i^{th} independent variable including lags is

$$\mathbf{w}_{i,u_j} = \begin{bmatrix} w_{i,0,u_j} \\ w_{i,1,u_j} \\ \vdots \\ w_{i,q,u_j} \end{bmatrix}$$

In **line 5**, We want to keep only the weight with the largest magnitude. For example if the maximum weight occurs at the first lag, i.e. $\max |w_{i,q,u_j}| = |w_{i,1,u_j}|$ then we set other weights to zero and this sub-vector becomes

$$\mathbf{w'}_{i,u_j} = \begin{bmatrix} 0 \\ w_{i,1,u_j} \\ \vdots \\ 0 \end{bmatrix}$$

In **line 15**, out-of-sample data of the independent variables are included to create forecasting values,

$$\mathbf{f}_{T+u_j+h} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{T+u_j} \\ f_{T+u_j+1} \\ \vdots f_{T+u_j+h} \end{bmatrix}$$

The values $[f_{T+u_j+1}, f_{T+u_j+2}, ..., f_{T+u_j+h}]$ are forecasting values.

1.2.2 Index Concatenation

Since our goal is to create an index, the old values should be kept for consistency.

Algorithm 2 LRFCI Concatenation

```
for each update interval u_j, j ← 0 to N do
    for each update interval u_k, k > j do
        Set f^{lr}_{u_k} = f^{lr}_{u_j}
    end for
end for
```

For example, Let $f^{lr}_{u_i,t}$ be the t^{th} entry of $\mathbf{f}^{lr}_{u_i}$, in the first update,

$$\mathbf{f}^{lr}_{u_1} = \begin{bmatrix} \mathbf{f}^{lr}_{u_0} \\ f^{lr}_{u_1,T+u_0+1} \\ f^{lr}_{u_1,T+u_0+2} \\ \vdots \\ f^{lr}_{u_1,T+u_1+h} \end{bmatrix}$$

1.3 SRFCI Construction

For SRFCI, we exclude the lag 0 term. The data matrix of independent variables is

$$X_T = [\mathbf{x}_{T-1,1}, ..., \mathbf{x}_{T-q,1}, \mathbf{x}_{T-1,2}, ..., \mathbf{x}_{T-q,2}, ..., \mathbf{x}_{T-1,n}, ..., \mathbf{x}_{T-q,n}]$$

1.3.1 Calculation

The weight vector is obtained from OLS regression. The algorithm is as the following:

Algorithm 3 SRFCI Construction

1: **for** each increasing update interval u_j, $j \leftarrow 0$ to N **do**
2: Standardize X_{T+u_j} by its mean and sd
3: Set $c = 0$ and $k = 0$
4: **for** $i = 1$ to n **do**
5: Regress $\mathbf{y}_{T+u_j}$ on $[\mathbf{x}_{T+u_j-1,i}, ..., \mathbf{x}_{T+u_j-q,i}]$
6: Extract 3 coefficients with smallest p-value (including intercept)
7: Regress on the 3 selected variables
8: Drop insignificant coefficients with p-value larger than constant p_0
9: **if** there are at least one remaining variables **then**
10: $c = c + 1$
11: Save the remaining coefficient estimates as $\mathbf{w}_{i,u_j}$
12: Set other insignificant coefficient estimates to zero
13: **if** intercept β_0 is significant **then**
14: $k = k + \beta_0$
15: **end if**
16: **else**
17: Set $\mathbf{w}_{i,u_j} = 0$
18: **end if**
19: **end for**
20: Standardize X_{T+u_j+h} by the mean and sd of X_{T+u_j}
21: Compute $\mathbf{f}^{sr}_{T+u_j+h} = \frac{1}{c}(X_{T+u_j+h}\mathbf{w}_{u_j} + k)$
22: **end for**

In **line 4**, we are basically regressing the dependent variable on the lagged independent variables grouped by their original variable.

1.3.2 Index Concatenation

The SRFCI $\mathbf{f}^{sr}$ concatenation process differs from LRFCI with an extra mean adjustment step since $\mathbf{f}^{sr}$ is not standardized.

Algorithm 4 SRFCI Concatenation

for each update interval u_j, $j \leftarrow 0$ to N **do**
    for each update interval u_k, $k > j$ **do**
        Set $\mathbf{f}^{sr}_{u_k} = \mathbf{f}^{sr}_{u_j} - \overline{\mathbf{f}^{sr}}_{u_j} + \overline{\mathbf{f}^{sr}}_{u_k}$
    end for
end for

Chapter 2

Documentation

2.1 LRFCI_main

This script includes the input section and the 'run' section to produce LRFCI.

2.1.1 Path Variables

These variables specify the folders to be used. They can be the same.

Arguments

primary_directory: the main directory

library_directory: the directory for the scripts

data_directory: the directory for data files

output_directory: the director for output files

Example

primary_directory = 'C:/Users/Ho Tak Yui/Documents/FCI Project/R code_andy'

library_directory = 'C:/Users/Ho Tak Yui/Documents/FCI Project/R code_andy/implementation'

data_directory = 'C:/Users/Ho Tak Yui/Documents/FCI Project/R code_andy/data'

output_directory = primary_directory

2.1.2 Input File Names

dependent_filename: the y variable in a txt file

indicator_filename: the X variables in a txt file

Example

dependent_filename = 'ch_lmpiwpi.txt'

indicator_filename = 'g_ori.txt'

2.1.3 Function Parameters

These parameters are passed to the scripts to produce LRFCI.

Arguments

fbgn: begin date

fend: end date

forecasting_horizon: the forecasting period for the fci

maxLag: the maximum lags to be created for the independent variables

ncomp: the number of latent factors: some functions are not implemented for $n > 1$

update_interval: the update period number for the moving window; usually the same as forecasting_horizone

n_update: the number of updates in the moving window

Example

fbgn = "1993-07-01"

fend = "2005-12-01"

forecasting_horizon = 12

maxLag = 6

ncomp = 1

update_interval = forecasting_horizon

n_update = 10

2.1.4 Output File Names

Arguments

save_output: TRUE/FALSE; specifies if you want to produce output files

lrfci_filename: the file name for lrfci

weight_filename: the file name for weight of independent variables

filtered_weight_filename: the file name for the filtered weight table

Example

save_output = TRUE

lrfci_filename = 'lrfci.csv'

weight_filename = 'lr_weight.csv'

filtered_weight_filename = 'lr_filtered_weight.csv'

2.1.5 Script Function Calls

The remaining scripts call the functions in common.R and LRFCI.R to produce the index.

data_read

data_lag_expansion

mov_win_pls

flatten_wgt

w_sparse

filtered_table

conca

2.2 SRFCI_main

This script includes the input section and the 'run' section to produce SRFCI and is similar to LRFCI_main.

2.2.1 Path Variables

These variables specify the folders to be used. They can be the same.

Arguments

primary_directory: the main directory

library_directory: the directory for the scripts

data_directory: the directory for data files

output_directory: the director for output files

Example

primary_directory = 'C:/Users/Ho Tak Yui/Documents/FCI Project/R code_andy'

library_directory = 'C:/Users/Ho Tak Yui/Documents/FCI Project/R code_andy/implementation'

data_directory = 'C:/Users/Ho Tak Yui/Documents/FCI Project/R code_andy/data'

output_directory = primary_directory

2.2.2 Input File Names

dependent_filename: the y variable in a txt file

indicator_filename: the X variables in a txt file

Example

dependent_filename = 'ch_gr12ir.txt'

indicator_filename = 'g_ori.txt'

2.2.3 Function Parameters

These parameters are passed to the scripts to produce LRFCI.

Arguments

fbgn: begin date

fend: end date

forecasting_horizon: the forecasting period for the fci

start_lag: the starting lag

maxLag: the maximum lags to be created for the independent variables

update_interval: the update period number for the moving window; usually the same as forecasting_horizone

n_update: the number of updates in the moving window

p_value: the filtered threshold for the independent variables

lag_length: max_lag - start_lag + 1

Example

fbgn = "1993-07-01"

fend = "2005-12-01"

forecasting_horizon = 12

start_lag = 1

maxLag = 6

ncomp = 1

update_interval = forecasting_horizon

n_update = 5

p_value = .1

lag_length = max_lag - start_lag + 1

2.2.4 Output File Names

Arguments

save_output: TRUE/FALSE; specifies if you want to produce output files

fci_filename: the file name for fci

unconca_filename: the file name for weight of independent variables

filtered_weight_filename: the file name for the filtered weight table

Example

save_output = TRUE

fci_filename = 'srfci.csv'

unconca_filename = 'sr_unconca.csv'

filtered_weight_filename = 'sr_filtered_weight.csv'

2.2.5 Script Function Calls

The remaining scripts call the functions in common.R and LRFCI.R to produce the index.

data_read

data_lag_expansion

gen_samp

leading_filter

weight_table

sr_conca

2.3 Common

Functions used by both LRSCI and SRSCI

2.3.1 packages_check

Description

Install packages

Usage

packages_check(x)

Arguments

x: package name

Value

None

2.3.2 data_read

Description

read the data file from .txt into R and format it into xts class

Usage

data_read(filename)

Arguments

filename: the txt file path

Value

an xts object

2.3.3 sample_interval

Description

create a sample time interval from data or start_period/end_period or both

Usage

sample_interval(data, start_period = time(first(data)), end_period = time(last(data)), start_increment = 0, end_increment = 0)

Arguments

data: an xts object; can be NULL

start_period: a date

end_period: a date

start_increment: number of months added to start_period

end_increment: number of months added to end_period

Value

an interval e.g. "1990-01-01/2018-03-01"

Example

sample_interval(data)

sample_interval(NULL, "1990-01-01", "2018-03-01")

sample_interval(NULL, "1990-01-01", "2018-03-01", 12, 12)

2.3.4 standardize_data

Description

enable out-of-sample standardization by subtracting mean and divided by std dev

Usage

standardize_data(data, insample_interval, outsample_interval = insample_interval)

Arguments

data: an xts object

insample_interval: an xts interval

outsample_interval: an xts interval

Value

a rescaled xts object

Example

standardize_data(data, "1990-01-01/2018-03-01")

standardize_data(data, "1990-01-01/2017-03-01", "1991-01-01/2018-03-01")

2.3.5 weight_table

Description

reformat the weight matrix to a table of filtered weight

Usage

weight_table(weight, table_fmt='multi')

Arguments

weight: the weight matrix of n(variables) rows and t(period) columns

table_fmt: 'multi': multi level indexing and more human readable / 'single': single level indexing and more machine-readable

Value

a dataframe object

2.4 LRFCI

Functions used in producing LRCFI.

2.4.1 pls1

Description

pls1 algo copied from wikipedia which is not actually used

Usage

pls1(X, y, l=1)

Arguments

X: matrix of independent variables

y: vector of the dependent variable

l: the limit on the number of latent factors in the regression

Value

a matrix of least square estimate (b0, b1)

2.4.2 matpls1

Description

pls1 algo copied from Eviews

Usage

matpls1(X, y, l=1)

Arguments

X: matrix of independent variables

y: vector of the dependent variable

l: the limit on the number of latent factors in the regression

Value

a list containing

t: the score vector

w: the weight vector

p: the loading vector

2.4.3 modifiedW

Description

produces the modified weight from the weight vector: the raw weight is corresponding to the deflated independent variables (weight); the modified weight is corresponding to the original independent variables (loading)

Usage

modifiedW(weight, loading, ncomp, n=dim(weight)[1])

Arguments

weight: the weight vector (see 2.4.2)

loading: the loading vector (see 2.4.2)

ncomp: the number of latent factors

n: number of variables in the weight vector

Value

a modified weight vector (matrix if n>1)

2.4.4 sparseW

Description

filtered the lag with largest loading (modified weight); weights of others are set to zero

Usage

sparseW(weight, ncomp, maxLag)

Arguments

weight: the weight vector (see 2.4.2, 2.4.3)

ncomp: the number of latent factors

maxLag: the maximum lag of independent variables

Value

a sparse weight vector (matrix if n>1)

2.4.5 insig_stats

Description

label the insignificant independent variables by the magnitude of the modified weight

Usage

sparseW(weight, ncomp, maxLag)

Arguments

weight: the weight vector (see 2.4.2, 2.4.3, 2.4.4)

ncomp: the number of latent factors

Value

a list containing

w_filtered: a weight vector of non-zero weights

insig_indicators: a filtered weight vector with elements of each absolute value less than 0.05

insig_indicators1: a filtered weight vector with elements of each value less than 0.6 quantile and larger than 0.4 quantile

2.4.6 all_matpls1

Description

an aggregate function corresponding to sub_all_matpls1 in EViews.

Dependencies

standardize_data 2.3.4

matpls1 2.4.2

modifiedW 2.4.3

sparseW 2.4.4

insig_stats 2.4.5

Usage

all_matpls1(data, dependent, in_samp, out_samp=in_samp, ncomp=1, method='plsr')

Arguments

data: an xts containing independent variables (X)

dependent: an xts containing dependent variables (y)

in_samp: in-the-sample period

out_samp: out-of-sample period if applicable

ncomp: the number of latent factors

method: 'eviews': the EViews method; 'plsr': the R package plsr

Value

a list containing

loadings: the loading vector

weight: the weight vector

scores: the score vector

w_modified: the modified weight vector

w_sparse: the sparse weight vector

sparse_factor: independent variables times sparse weight

w_filtered: a weight vector of non-zero weights

insig_indicators: a filtered weight vector with elements of each absolute value less than 0.05

insig_indicators1: a filtered weight vector with elements of each value less than 0.6 quantile and larger than 0.4 quantile

2.4.7 mov_win_pls

Description

a moving window implementation of all_matpls1

Usage

mov_win_pls(data, dependent, fbgn, fend, forecasting_horizon, update_interval, n_update, ncomp=1)

Arguments

data: an xts containing independent variables (X)

dependent: an xts containing dependent variables (y)

in_samp: in-the-sample period

out_samp: out-of-sample period if applicable

ncomp: the number of latent factors

method: 'eviews': the EViews method; 'plsr': the R package plsr

Value

a list containing

loadings: the loading vector

weight: the weight vector

scores: the score vector

w_modified: the modified weight vector

w_sparse: the sparse weight vector

sparse_factor: independent variables times sparse weight

w_filtered: a weight vector of non-zero weights

insig_indicators: a filtered weight vector with elements of each absolute value less than 0.05

insig_indicators1: a filtered weight vector with elements of each value less than 0.6 quantile and larger than 0.4 quantile

2.4.8 conca

Description

concatenate the factor and align a single output the list of pls results

Usage

conca(ls, fend, update_interval, attr = 'sparse_factor')

Arguments

ls: the list of factor to be concatenated

fend: the end date

update_interval: the number of months to be updated

attr: the variable name to be concatenated

Value

the concatenated factor matrix. Each column is an update.

2.4.9 flatten_wgt

Description

flatten a list: convert from a list to a matrix

Usage

flatten_wgt(ls, attr='weight')

Arguments

ls: the list of to be flattened

attr: the variable name to be flattened

Value

a weight matrix. Each column is an update.

2.5 SRFCI

Functions used in producing SRCFI.

2.5.1 gen_samp

Description

generate the sample pairs for moving window iteration

Usage

gen_samp(fbgn, fend, forecasting_horizon, max_lag, update_interval, n_update)

Arguments

fbgn: begin date

fend: end date

forecasting_horizon: the forecasting period for the fci

maxLag: the maximum lags to be created for the independent variables

update_interval: the update period number for the moving window; usually the same as forecasting_horizon

n_update: the number of updates in the moving window

Value

a matrix of sample interval containing training periods and forecasting periods

2.5.2 regr

Description

1. produce ols regression coefficients
2. select the most significant three
3. filter the three variables by p-value

Usage

regr(data_lag, dependent, samp, p_value = .1)

Arguments

data_lag: matrix of independent variables (and lags)

dependent: vector of the dependent variable

samp: the sampling interval

p_value: the filtering threshold

Value

a list containing

weight: filtered ols estimates as the weight vector

const: accumulated significant intercepts in every regression

ind_count: the number of regression with at least one significant coeff (including intercept)

2.5.3 leading_filter

Description

a moving window implementation to produce weight and factor

Usage

leading_filter(data_lag, dependent, sample, p_value = .1)

Arguments

data_lag: matrix of independent variables (and lags)

dependent: vector of the dependent variable

sample: the sampling period table produced by gen_samp 2.5.1

Value

a list containing

factor_ts: the latent factor time series matrix. Each column is an iteration. Each row is a time period.

weight_ts: the weight matrix. Each column is an iteration.

const_ts: an array of accumulated significant intercepts in each iteration

ind_count_ts: an array of the number of regression with at least one significant coeff (including intercept) in each iteration

2.5.4 sr_conca

Description

the short run index concatenation function to keep the old value of the index

Usage

sr_conca(factor, ind_count, fend, update_interval, forecasting_horizon, n_update = dim(factor)[2]-1)

Arguments

factor: the factor matrix produced by leading_filter 2.5.3

ind_count: the array of the number of regression with at least one significant coeff produced by leading_filter 2.5.3; this is used to average the factor

fend: end date

forecasting_horizon: the forecasting period for the fci

update_interval: the update period number for the moving window

n_update: the number of updates in the moving window; should be the same as number of columns of factor minus one

Value

a list containing

conca: the concatenated index

unconca: the un-concatenated index

Chapter 3

Note

3.1 LRFCI Note

3.1.1 Regression

We start with a data matrix

$$X = [x_1, x_2, ...x_n]$$

where x_i is the $i^{th}, i = 1, ...n$ independent variable of T periods. We also have y as the dependent variable of length T. We first create q lags for each x_i and **standardize** each column.

$$X_t = [x_{1,t}, x_{1,t-1}, ..., x_{1,t-q}, x_{2,t}, x_{2,t-1}, ..., x_{2,t-q}, ...x_{1,t}, x_{n,t-1}, ..., x_{n,t-q}]$$

Since we take $q = 6$, the dimension of X_t is $T \times (7n)$. The main idea is to regress y on each column of X_t one-by-one.

$$y_t = x_{i,t-q}\beta_{i,t-q} + \epsilon_{i,t-q} \quad i = 1, ...n \text{ and } q = 0, ...6$$

and the OLS estimate of β is

$$\hat{\beta_{i,t-q}} = \frac{\sum_{t=1}^{T} x_{i,t-q}y_t}{\sum_{t=1}^{T} x_{i,t-q}^2}$$

Since X is standardized, $\sum x_i \to 0$ and $\sum x_i^2 \to T$

$$\hat{\beta_{i,t-q}} = \frac{1}{T}\sum_{t=1}^{T} x_{i,t-q}y_t$$

Consider $X_t^T y_t$,

$$X_t^T y_t = \begin{bmatrix} x_{1,t}^T \\ x_{1,t-1}^T \\ \vdots \\ x_{1,t-q}^T \\ x_{2,t}^T \\ x_{2,t-1}^T \\ \vdots \\ x_{2,t-q}^T \\ \vdots \\ x_{n,t}^T \\ x_{n,t-1}^T \\ \vdots \\ x_{n,t-q}^T \end{bmatrix} y_t$$

and

$$x_{i,t-q}^T y_t = \sum_t x_{i,t-q} y_t = T\hat{\beta_{i,t-q}}$$

Then the weight vector $w = X^T y / |X^T y|$ is just a re-scaled version of $\hat{\beta}$.

3.1.2 Factor Construction

Consider

$$\begin{aligned} f =& Xw = [x_1, ..., x_n] \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \\ =& x_1 w_1 + ... x_n w_n \end{aligned}$$

Then the variance becomes

$$Var(f) = \sum_{i=1}^{n} w_i^2 Var(x_i) + \sum_{i \neq j} w_i w_j Cov(x_i, x_j)$$

Since x_is are standardized and further if we normalize the weight vector, i.e. $|w|_2 = 1$,

$$Var(f) = 1 + \sum_{i \neq j} w_i w_j Corr(x_i, x_j) \neq 1$$

3.2 SRFCI Note

Regress the dependent variable y on an independent variable (with lags) x_i, $i = 1, ..., n$:

$$y = \beta_{0i} + \beta_{1i} x_i + \epsilon_i$$

to obtain an estimator of y:

$$\hat{y}_i = \hat{\beta_{0i}} + \hat{\beta_{1i}} x_i$$

The estimator of y is then

$$\hat{y} = \frac{1}{n} \sum_{i=1}^{n} \hat{y}_i = \frac{1}{n} \sum_{i=1}^{n} \hat{\beta_{0i}} + \frac{1}{n} \sum_{i=1}^{n} \hat{\beta_{1i}} x_i$$