Package ‘ArCo’
November 5, 2017

Title Artificial Counterfactual Package
Version 0.3-1
Date 2017-11-05
Maintainer Gabriel F. R. Vasconcelos <gabrielrvsc@yahoo.com.br>

BugReports https://github.com/gabrielrvsc/ARCO/issues

Description Set of functions to analyse and estimate Artificial Counterfactual models from Carvalho, Masini and Medeiros (2016) <DOI:10.2139/ssrn.2823687>.

License MIT + file LICENSE
Encoding UTF-8
LazyData TRUE
RoxygenNote 6.0.1

Imports Matrix, glmnet, boot

NeedsCompilation no

Author Yuri R. Fonseca [aut], Ricardo Masini [aut], Marcelo C. Medeiros [aut], Gabriel F. R. Vasconcelos [aut, cre]

Repository CRAN

Date/Publication 2017-11-05 12:29:56 UTC

R topics documented:

  data.q1 ................................................................. 2
  data.q2 ................................................................. 2
  estimate_t0 .......................................................... 3
  fitArCo .............................................................. 5
  inflationNFP ......................................................... 8
  panel_to_ArCo_list ................................................. 8
  plot.fitArCo ......................................................... 9

Index 11
**Description**

This data contains 100 observations of 20 variables generated using the dgp from Carvalho, Masini and Medeiros (2016). Each variable is one ArCo unit. The intervention took place on the first unit at t0=51 by adding a constant equal to 0.628, which is one standard deviation of the treated unit variable before the intervention.

**Usage**

```r
data(data.q1)
```

**Format**

A matrix with 100 rows and 20 variables.

**References**


---

**Description**

This data is a list with two matrixes, each have 100 observations and 6 variables. It was generated using the dgp from Carvalho, Masini and Medeiros (2016). In the ArCo context, each matrix is a variable and each variable in the matrices is a unit. The intervention took place on the first unit at t0=51 by adding constants of 0.840 and 0.511 (one standard deviation before the intervention) on variables (matrixes) 1 and 2.

**Usage**

```r
data(data.q2)
```

**Format**

A list with 2 matrixes of 100 rows and 6 variables.

**References**

estimate_t0

Estimates the intervention time on a given treated unit

Description

Estimates the intervention time on a given treated unit based on any model supplied by the user.

Usage

\[
\text{estimate}_t0(\text{data, fn = NULL, p.fn = NULL, start = 0.4, end = 0.9, treated.unit = 1, lag = 0, xreg = NULL, ...})
\]

Arguments

- **data**: A list of matrixes or data frames of length \(q\). Each matrix is \(T \times n\) and it contains observations of a single variable for all units and all periods of time. Even in the case of a single variable \((q=1)\), the matrix must be inside a list.
- **fn**: The function used to estimate the first stage model. This function must receive only two arguments in the following order: \(X\) (independent variables), \(y\) (dependent variable). If the model requires additional arguments they must be supplied inside the function fn. If not supplied the default is the lm function.
- **p.fn**: The forecasting function used to estimate the counterfactual using the first stage model (normally a predict function). This function also must receive only two arguments in the following order: \(\text{model}\) (model estimated in the first stage), \(\text{newdata}\) (out of sample data to estimate the second stage). If the prediction requires additional arguments they must be supplied inside the function p.fn.
- **start**: Initial value of \(\lambda_0\) to be tested.
- **end**: Final value of \(\lambda_0\) to be tested.
- **treated.unit**: Single number indicating the unit where the intervention took place.
- **lag**: Number of lags in the first stage model. Default is 0, i.e. only contemporaneous variables are used.
- **Xreg**: Exogenous controls.
- **...**: Additional arguments used in the function fn.

Details

This description may be useful to clarify the notation and understand how the arguments must be supplied to the functions.

- **units**: Each unit is indexed by a number between \(1, \ldots, n\). They are for example: countries, states, municipalities, firms, etc.
- **Variables**: For each unit and for every time period \(t = 1, \ldots, T\) we observe \(q_i \geq 1\) variables. They are for example: GDP, inflation, sales, etc.
- **Intervention**: The intervention took place only in the treated unit at time \(t_0 = \lambda_0 \times T\), where \(\lambda_0\) is in \((0, 1)\).
Value

A list with the following items:

- `t0` Estimated t0.
- `delta.norm` The norm of the delta corresponding to t0.
- `call` The matched call.

See Also

`fitArCo`

Examples

```r
# Example for q=Q

data(data.q1)
# = First unit was treated on t=51 by adding
# a constant equal to one standard deviation.

data=list(data.q1) # = Even if q=Q the data must be in a list

# = Fitting the ArCo using linear regression

# = creating fn and p.fn function = 
fn=function(X,y){
  return(lm(y-X))
}
p.fn=function(model,newdata){
  b=coef(model)
  return(cbind(1,newdata)%*%b)
}

t0a=estimate_t0(data = data, fn = fn, p.fn = p.fn, treated.unit = 1 )

# Example for q=R

data(data.q2) # data is already a list

t0b=estimate_t0(data = data.q2,fn = fn, p.fn = p.fn, treated.unit = 1, start=0.4)
```
**fitArCo**

Estimates the ArCo using the model selected by the user

**Description**

Estimates the Artificial Counterfactual using any model supplied by the user, calculates the most relevant statistics and allows for the counterfactual confidence intervals to be estimated by block bootstrap.

The model must be supplied by the user through the arguments fn and p.fn. The first determines which function will be used to estimate the model and the second determines the forecasting function. For more details see the examples and the description on the arguments.

**Usage**

```r
fitArCo(data, fn = NULL, p.fn = NULL, treated.unit, t0, lag = 0,
        xreg = NULL, alpha = 0.05, boot.cf = FALSE, R = 100, l = 3,
        VCOV.type = c("iid", "var", "nw", "varhac"), VCOV.lag = 1,
        bandwidth.kernel = NULL, kernel.type = c("QuadraticSpectral", "Truncated",
         "Bartlett", "Parzen", "TukeyHanning"), VHAC.max.lag = 5,
        prewhitening.kernel = FALSE, ...)
```

**Arguments**

- **data**
  - A list of matrixes or data frames of length q. Each matrix is T X n and it contains observations of a single variable for all units and all periods of time. Even in the case of a single variable (q=1), the matrix must be inside a list.

- **fn**
  - The function used to estimate the first stage model. This function must receive only two arguments in the following order: X (independent variables), y (dependent variable). If the model requires additional arguments they must be supplied inside the function fn. If not supplied the default is the lm function.

- **p.fn**
  - The forecasting function used to estimate the counterfactual using the first stage model (normally a predict funtion). This function also must receive only two arguments in the following order: model (model estimated in the first stage), newdata (out of sample data to estimate the second stage). If the prediction requires additional arguments they must be supplied inside the function p.fn.

- **treated.unit**
  - Single number indicating the unit where the intervention took place.

- **t0**
  - Single number indicating the intervention period.

- **lag**
  - Number of lags in the first stage model. Default is 0, i.e. only contemporaneous variables are used.

- **xreg**
  - Exogenous controls.

- **alpha**
  - Significance level for the delta confidence bands.

- **boot.cf**
  - Should bootstrap confidence intervals for the counterfactual be calculated (default=FALSE).

- **R**
  - Number of bootstrap replications in case boot.cf=TRUE.
Block length for the block bootstrap.

Type of covariance matrix for the delta. "iid" for standard covariance matrix, "var" or "varhac" to use prewhitened covariance matrix using VAR models, "varhac" selects the order of the VAR automatically and "nw" for Newey West. In the last case the user may select the kernel type and combine the kernel with the VAR prewhitening. For more details see Andrews and Monahan (1992).

Lag used on the robust covariance matrix if VCOV.type is different from "iid".

Kernel bandwidth. If NULL the bandwidth is automatically calculated.

Kernel to be used for VCOV.type="nw".

Maximum lag of the VAR in case VCOV.type="varhac".

If TRUE and VCOV.type="nw", the covariance matrix is calculated with prewhitening (default=FALSE).

Additional arguments used in the function fn.

Details

This description may be useful to clarify the notation and understand how the arguments must be supplied to the functions.

- units: Each unit is indexed by a number between 1, . . ., n. They are for example: countries, states, municipalities, firms, etc.
- Variables: For each unit and for every time period t = 1, . . ., T we observe q ≥ 1 variables. They are for example: GDP, inflation, sales, etc.
- Intervention: The intervention took place only in the treated unit at time t0 = λ0 * T, where λ0 is in (0,1).

Value

An object with S3 class fitArCo.

cf estimated counterfactual
fitted.values In sample fitted values for the pre-treatment period.
model A list with q estimated models, one for each variable. Each element in the list is the output of the fn function.
delta The delta statistics and its confidence interval.
p.value ArCo p-value.
data The data used.
t0 The intervention period used.
treated.unit The treated unit used.
omega Residual standard deviation.
residuals model residuals.
boot.cf A list with the bootstrap result (boot.cf=TRUE) or logical FALSE (boot.cf=FALSE). In the first case, each element in the list refers to one bootstrap replication of the counterfactual, i.e. the list length is R.
call The matched call.
References


See Also

plot, estimate_t0, panel_to_ArCo_list

Examples

# Example for q=1
# First unit was treated on t=UQ by adding a constant equal to one standard deviation

data(data.q1)
# = First unit was treated on t=51 by adding a constant equal to one standard deviation

data=list(data.q1) # Even if q=1 the data must be in a list

# = creating fn and p.fn function =
fn=function(x,y){
  return(lm(y~x))
}
p.fn=function(model,newdata){
  b=coef(model)
  return(cbind(1,newdata) %*% b)
}

ArCo=fitArCo(data = data, fn = fn, p.fn = p.fn, treated.unit = 1, t0 = 51)

# Example for q=2
# First unit was treated on t=51 by adding constants of one standard deviation for the first and second variables

data(data.q2) # data is already a list

# = Fitting the ArCo using the package glmnet =
# Quadratic Spectral kernel weights for two lags =
# Bartlett kernel weights for two lags =
require(glmnet)
set.seed(123)
ArCo2=fitArCo(data = data.q2, fn = cv.glmnet, p.fn = predict,treated.unit = 1, t0 = 51,
inflationNFP  

Dataset used on the empirical example by Carvalho, Masini and Medeiros (2016).

Description

This is the data from the *nota fiscal paulista* (NFP) example from Carvalho, Masini and Medeiros (2016). The variables are the food away from home component of the inflation and the GDP for 9 metropolitan areas in Brazil. Each variable is represented by a matrix inside the list. The treated unit is the Sao Paulo metropolitan area, which is the first column in each matrix. The treatment took place at $t_0 = 34$.

Usage

data(inflationNFP)

Format

A list with two matrixes of 56 rows and 9 variables.

References


panel_to_ArCo_list  

Transforms a balanced panel into a list of matrices compatible with the fitArCo function

Description

Transforms a balanced panel into a list of matrices compatible with the fitArCo function. The user must identify the columns with the time, the unit identifier and the variables.

Usage

panel_to_ArCo_list(panel, time, unit, variables)

Arguments

panel  
Balanced panel in a data.frame with columns for units and time.

time  
Name or index of the time column.

unit  
Name or index of the unit column.

variables  
Names or indexes of the columns containing the variables.
See Also
  fitArCo

Examples

  # = Generate a small panel as example =#
  set.seed(123)
  time=sort(rep(1:100,2))
  unit=rep(c("u1","u2"),100)
  v1=rnorm(200)
  v2=rnorm(200)
  panel=data.frame(time=time,unit=unit,v1=v1,v2=v2)
  head(panel)

  data=panel_to_ArCo_list(panel,time="time",unit="unit",variables = c("v1","v2"))
  head(data$v1)

plot.fitArCo  Plots realized values and the counterfactual estimated by the fitArCo function

Description

Plots realized values and the counterfactual estimated by the fitArCo function. The plotted variables
will be on the same level as supplied to the fitArCo function.

Usage

  ## S3 method for class 'fitArCo'
  plot(x, ylab = NULL, main = NULL, plot = NULL,
       ncol = 1, display.fitted = FALSE, y.min = NULL, y.max = NULL,
       confidence.bands = FALSE, alpha = 0.05, ...)

Arguments

  x  An ArCo object estimated using the fitArCo function.
  ylab  n dimensional character vector, where n is the length of the plot argument or
         n=q if plot=NULL.
  main  n dimensional character vector, where n is the length of the plot argument or
         n=q if plot=NULL.
  plot  n dimensional numeric vector where each element represents an ArCo unit. If
        NULL, all units will be plotted. If, for example, plot=c(1,2,5) only units 1 2 and
        5 will be plotted according to the order specified by the user on the fitArCo.
  ncol  Number of columns when multiple plots are displayed.
  display.fitted  If TRUE the fitted values of the first step estimation are also plotted (default=FALSE).
y.min  n dimensional numeric vector defining the lower bound for the y axis. n is the length of the plot argument or n=q if plot=NULL

y.max  n dimensional numeric vector defining the upper bound for the y axis. n is the length of the plot argument or n=q if plot=NULL

confidence.bands  TRUE to plot the counterfactual confidence bands (default=FALSE). If the ArCo was estimated without bootstrap this argument will be forced to FALSE.

alpha  Significance level for the confidence bands.

...  Other graphical parameters to plot.

See Also

fitArCo

Examples

# Example based on the q=1 fitArCo

data(data.q1)
data=list(data.q1)
fn=function(X,Y){
  return(lm(Y~X))
}
p.fn=function(model,newdata){
  b=coef(model)
  return(cbind(1,newdata) %*% b)}

ArCo=fitArCo(data=data,fn=fn,p.fn=p.fn,treated.unit=1,t0=51)
plot(ArCo)
Index

+Topic **ArCo**
  fitArCo, 5

+Topic **datasets**
  data.q1, 2
  data.q2, 2
  inflationNFP, 8

  data.q1, 2
  data.q2, 2

  estimate_t0, 3, 7

  fitArCo, 4, 5, 9, 10

  inflationNFP, 8

  panel_to_ArCo_list, 7, 8
  plot, 7
  plot.fitArCo, 9