# Package ‘pampe’

## November 7, 2015

**Type** Package  
**Title** Implementation of the Panel Data Approach Method for Program Evaluation  
**Version** 1.1.2  
**Date** 2015-11-06  
**Author** Ainhoa Vega-Bayo  
**Maintainer** Ainhoa Vega-Bayo &lt;ainhoa.vega@gmail.com&gt;  
**Description** Implements the Panel Data Approach Method for program evaluation as developed in Hsiao, Ching and Ki Wan (2012). pampe estimates the effect of an intervention by comparing the evolution of the outcome for a unit affected by an intervention or treatment to the evolution of the unit had it not been affected by the intervention.  
**Depends** leaps  
**Enhances** xtable  
**License** GPL-2  
**NeedsCompilation** no  
**Repository** CRAN  
**Date/Publication** 2015-11-07 07:24:16

**R topics documented:**

- growth ................................................................. 2  
- pampe ................................................................. 2  
- pampe-internal ...................................................... 7  
- robustness ............................................................ 8

**Index** 9
Example Data for `pampe` function from the `pampe` package

Description

Dataset for the example from the `pampe` function of the `pampe` package. The growth dataset, obtained from the supplemental materials of Hsiao, Steve Ching and Ki Wan (2012) contains information on the quarterly real GDP growth rate of 24 countries (the donor pool) and Hong Kong from 1993 Q1 to 2008 Q1, computed as the change with respect to the same quarter in the previous year. The data is organized in standard cross-sectional data format, with the variables (the quarterly real GDP growth rate of the countries in the donor pool act as explanatory variables) extending across the columns and the quarters (time-periods) extending across rows. In this case the treated unit, Hong Kong, is in the first column whereas the possible controls (countries in the original donor pool) are in columns 2:25. It is important for the user to have his or her data in this standard format to correctly apply the methodology. Naming the rows and especially the columns is also strongly recommended. Be careful to avoid characters such as spacing when naming.

Usage

data(growth)

Format

A data frame with 61 observations on 25 variables (countries/units).

Source


See Also

`pampe`

Implementation of the Panel Data Approach for Program Evaluation

Description

Implements the Panel Data Approach for program evaluation as developed in Hsiao, Steve Ching and Ki Wan (2012). pampe estimates the effect of an intervention by comparing the evolution of the outcome for a unit affected by an intervention or treatment to the evolution of the unit had it not been affected by the intervention.
Usage

```r
pampe(time.pretr, time.tr, treated, controls = "All", data,
       nbest = 1, nvmax = length(controls), select = "AICc", placebos = FALSE)
```

data(growth)

Arguments

- **time.pretr**: The pre-treatment periods, up to the introduction of the treatment. For example, if you have 10 pre-treatment periods and the treatment was introduced in the 11th period, this argument could be 1:10.

- **time.tr**: The treatment period plus the periods for which you want to check the effect. For example, if the treatment was introduced in the 11th period and you want results for 9 more periods, it would be 11:20.

- **treated**: The treated unit, the unit that receives the intervention. It can be a name or the index of the column if the columns in the data matrix are named (recommended). For example, in the case of the growth data included in the package, the name of the treated unit is "HongKong", and it is in the first column of the data matrix, so this argument can be either "HongKong" or 1.

- **controls**: The units used as controls to calculate the counterfactual, that have not received the treatment. By default, all the remaining (after removing the treated) columns in the data matrix are included as columns, but specific controls can be specified using their column name, e.g. c("Australia", "Austria", "Canada") or their column index, e.g. 2:4.

- **data**: The name of the data matrix to be used, e.g. growth. The data matrix should be in standard cross-sectional format: with the variables (the controls/units in the donor pool pool act as explanatory variables) extending across the columns and the quarters (time-periods) extending across rows. It is important for the user to have his or her data in this standard format to correctly apply the function. Naming the rows and especially the columns is also strongly recommended. Be careful to avoid characters such as spacing when naming.

- **nbest**: The original method by Hsiao, Steve Ching and Ki Wan (2012) specifies to keep the best model in terms of R squared for each order, hence the default is one. However the user might choose to keep the best 2, 3... before moving on to the second step of the method by switching this argument default.

- **nvmax**: How many subsets of controls should the method check. The original method by Hsiao says to check all subsets up to the biggest size, hence the default; but if the pre-treatment period is too short that might not be possible. If it’s too big but attempt to apply the method regardless, it will throw out an error telling you to change this argument or reduce the number of controls.

- **select**: The model selection criteria for the second step of the method. Originally they propose either AICc (default) or AIC. The user can choose between those two or BIC as well.

- **placebos**: Options between "controls", "time", or "Both". The "controls" placebo method reassigns the treatment to all controls and computes the treatment effect, the "time" placebo reassigns treatment to a time period previous to the treatment and computes the treatment effect as well. You can then plot the results.
Details

As proposed by Hsiao, Steve Ching and Ki Wan (2012), the panel data approach method for program evaluation estimates the effect of a treatment or policy intervention in the aggregate outcome of a single affected unit by exploiting the dependence among cross-sectional units to construct a counterfactual of the treated unit(s), which is an estimate of how the affected unit would have developed in the absence of an intervention. The estimated effect of the policy intervention is therefore simply the difference between the actual observed outcome of the treated unit and this estimated counterfactual.

The way they propose to estimate the outcome of the treated unit under no treatment, Y^0_1t, is to use the following modeling strategy: use R^2 (or likelihood values) in order to select the best OLS estimator for Y^0_1t using j out of the J units in the donor pool, denoted by M(j)* for j=1, ..., J; then choose M(m)* from M(1)*, ..., M(J)* in terms of a model selection criterion, like AICc, AIC or BIC. Note that the method calculates OLS models of up to J+1 parameters; so that if the length of the pre-treatment period t=1, 2, ..., T'-1 is not of a much higher order than that, the regressions M(J-1)*, M(J)* can not be calculated because there are not enough degrees of freedom.

To avoid this problem, the pampe package proposes the following slight modification to the previously outlined modeling strategy: use R^2 in order to select the best OLS estimator for Y^0_1t using j out of the J units in the donor pool, denoted by M(j)* for j=1, ..., T_0-4; then choose M(m)* from M(1)*, ..., M(T_0-4)* in terms of a model selection criterion (in our case AICc). Note that the key difference is that while we allowed models up to M(J)*, this is now modified to allow models up to M(T_0-4)*, with T_0-4<J, which allows for at least 3 degrees of freedom. This is implemented through the default value of nvmax, which is equal to J, or if not possible, to J-4. The user can of course override this default.

The results of the function include, among others, the optimal model chosen by the strategy (an object of class lm) and the estimated treatment effects (difference between the actual outcome and the counterfactual outcome) for the placebo tests if the user chooses to carry them out.

Value

A list of objects, including:

- controls: a named vector of the controls finally included in the model
- model: an object of class lm with the optimal model. Usual methods such as fitted(), residuals(), ... can be used on it.
- counterfactual: the path of the estimated counterfactual for the time.pretr and time.tr periods.
- placebo.ctrl: if the user has chosen to perform the placebo test which reassigns the treatment to the controls in the original donor pool, this element is a list which includes another two elements: mspe and tr.effect. The first includes the mspe for the pre-treatment period (time.pretr) and the second is the estimated treatment effect for the treated unit in the first column and for the countries in the original donor pool in the remaining columns.
- placebo.time: same as placebo.ctrl but with the reassignment of the treatment in time, to periods in the pre-treatment period.
- data: initial data, for further use.
Note

Check the references for more information on the placebo studies. The leaps package by Lumley is required for pampe to run properly.

Author(s)

Ainhoa Vega-Bayo

References


See Also

data, summary

Examples

```r
# Load the sample dataset
data(growth)

# First we apply the method to the political integration of Hong Kong
# as done by Hsiao et al. (2012) using AICc as selection criteria
tr <- time.tr
ct <- possible.ctrls
fit <- pampe(time.pretr, time.tr, treated, controls, data=growth)

# The function automatically prints out a summary of the optimal model
summary(fit)
```
The method `plot()` works on object of class `pampe` to produce a plot of the actual evolution of the treated unit together with the predicted counterfactual path. A simple plot call to our saved `pampe` object would give us the desired plot:

```r
plot(pol.integ)
```

### User-generated plot

A plot of the estimated counterfactual together with the actual outcome:

```r
matplot(c(time.pretr, time.tr), cbind(growth[c(time.pretr, time.tr), 1], pol.integ$counterfactual), type="l", ylab="GDP growth", xlab="", ylim=c(-0.15, 0.15), col=1, lwd=3, xaxt="n")
axis(1, at=c(time.pretr, time.tr)[c(seq(2, length(c(time.pretr, time.tr)), by=2))], labels=c(rownames(growth)[c(time.pretr, time.tr)[c(seq(2, length(c(time.pretr, time.tr)), by=2))]]), las=3)
title(xlab="Quarter", mgp=c(3.6, 0.5, 0))
legend("bottomright", c("Hong Kong", "predicted Hong Kong"), col=1, lty=c(1, 2), lwd=3)
```

Now we include placebo tests:

```r
pol.integ.placebos <- pampe(time.pretr=time.pretr, time.tr=time.tr, treated=treated, controls=possible.ctrls, data=growth, placebos="Both")
```

We can use the plot method again and check the results in the viewer:

```r
plot(pol.integ.placebos)
```

### Or create user-generated plots

Plot of the placebos-controls:

```r
mspe <- pol.integ.placebos$placebo.ctrl$mspe
linewidth <- matrix(2, 1, ncol(mspe)-1)
linewidth <- append(linewidth, 5, after = 0)
```

Plot of the placebos-in-time:

```r
matplot(c(time.pretr, time.tr), pol.integ.placebos$placebo.in.time$tr.effect, type="l", xlab="", ylab="GDP growth gap", col=c("red", matrix(1, 1, ncol(mspe)-1)), lty=1, lwd=linewidth, ylim=c(-0.35, 0.2), xaxt="n")
axis(1, at=c(time.pretr, time.tr)[c(seq(4, length(c(time.pretr, time.tr)), by=4))],
labels=c(rownames(growth)[c(time.pretr, time.tr)[c(seq(4, length(c(time.pretr, time.tr)), by=4))]]), las=3)
title(xlab="Quarter", mgp=c(3.6, 0.5, 0))
legend("topleft", c("Hong Kong", "Controls"), col=c("red", 1), lty=c(1, 2), lwd=c(5, 2))
```

The estimated effect for Hong Kong does not appear to be significant since it is not an outlier.

### Plot of the placebos-in-time

For example let's plot the first reassignment:

```r
placebo.in.time1 <- pol.integ.placebos$placebo.in.time$tr.effect[, 2] + growth[c(time.pretr, time.tr), 1]
matplot(c(time.pretr, time.tr), cbind(growth[c(time.pretr, time.tr), 1],
pol.integ.placebos$counterfactual, placebo.in.time1), type="l", ylab="GDP growth",
xlim=c(-0.25, 0.2), col=1, lwd=3, xaxt="n")
axis(1, at=c(time.pretr, time.tr)[c(seq(4, length(c(time.pretr, time.tr)), by=4))],
labels=c(rownames(growth)[c(time.pretr, time.tr)[c(seq(4, length(c(time.pretr, time.tr)),
by=4))]]), las=3)
title(xlab="Quarter", mgp=c(3.6, 0.5, 0))
```
legend("bottomleft",c("actual", "predicted", paste("placebo", colnames(pol.integ.placebos$placebo.time$str.effect)[2], sep=" ")), col=1, lty=c(1,2,3), lwd=3)
abline(y=time.pretr[length(time.pretr)], lty=2, lwd=3)
abline(y=which(colnames(pol.integ.placebos$placebo.time$str.effect)[2]==rownames(growth)), lty=3, lwd=3)

## We can also plot the gaps all at the same time
mspe <- pol.integ.placebos$placebo.time-mspe
linewidth <- matrix(2, 1, ncol(mspe)-1)
linewidth <- append(linewidth, 5, after = 0)

matplot(c(time.pretr, time.tr), pol.integ.placebos$placebo.time$str.effect, type="l", xlab="", ylab="GDP growth gap", col=c("red", matrix(1, 1, ncol(mspe)-1)), lty=c(1, matrix(2, 1, ncol(mspe)-1)), lwd=linewidth, ylim=c(-0.35, 0.2), xaxt="n")
axis(1, at=c(time.pretr, time.tr)[c(seq(4, length(c(time.pretr, time.tr)), by=4))], labels=c(rownames(growth)[c(time.pretr, time.tr)[c(seq(4, length(c(time.pretr, time.tr)), by=4))]]), las=3)
title(xlab="Quarter", mgp=c(3.6,0.5,0))
legend("topleft",c("Hong Kong", "Controls"),col=c("red", 1),lty=c(1,2),lwd=c(5,2))

## Leave-one-out robustness check
robust <- robustness(pol.integ)
plot(robust)

---

**Implementation of the Panel Data Approach for Program Evaluation**

**Description**

Description of internal function for the pampe package, regsubsets2aic. Takes the models estimated using regsubsets and orders them according to a model selection criteria like AIC, AICc, or BIC.

**Usage**

regsubsets2aic(x, y, z)

**Arguments**

- `x` A matrix of predictors
- `y` A response vector
- `z` Output of regsubsets

**See Also**

pampe.leaps
robustness

Robustness check for the Implementation of the Panel Data Approach for Program Evaluation

Description

Implements a leave-one-out robustness check for the Panel Data Approach for program evaluation as developed in Hsiao, Steve Ching and Ki Wan (2012). Robustness must be run after pampe and will automatically and iteratively remove each one of the controls resulted in pampe to check whether the results obtained in pampe are robust or are driven by a single control.

Usage

robustness(pampe.object)

Arguments

pampe.object The object resulting from previously running the pampe function.

Note

Check the references in pampe for more information on the placebo studies. The leaps package by Lumley is required for pampe to run properly.

Author(s)

Ainhoa Vega-Bayo

See Also

pampe leaps
Index

*Topic **datasets**
  growth, 2

growth, 2

leaps, 5, 7, 8

pampe, 2, 2, 7, 8
pampe-internal, 7
plot.pampe (pampe), 2
plot.robustness (robustness), 8
print.summary.pampe (pampe), 2

regsubsets2aic (pampe-internal), 7
robustness, 8

summary.pampe (pampe), 2

xtable.pampe (pampe), 2
xtable.robustness (robustness), 8