

ECONOMETRIC SOFTWARE - FREE VS. PAID SOLUTIONS

Nicolae-Marius JULA*

Abstract

Using software in econometric analysis empower the researches to find the answer to the problems faster than ever. The use of commercial software is very wide-spread, but the costs are always a barrier for a lot of researchers. Using free alternatives like R combines the power processing with affordability. We analyze in this paper some aspects related to main differences between commercial econometric software and free alternatives.

Keywords: *Econometric software, open source, E-Views, R, Bai-Perron*

1. Introduction

As Samuelson et al. stated in the paper¹, econometrics is “the quantitative analysis of actual economic phenomena based on the concurrent development of theory and observation, related by appropriate methods of inference”. There are a lot of practical applications of econometric methods, like forecasting macroeconomic indicators, analyzing firm’s factors, analyzing the relation between management techniques and results, analyzing the effect of dividend announcements and investors’ behavior, predicting the results in political campaigns, predicting the results of a marketing campaign and so on.

The methods used are in a continuous process of adapting, improving and getting more and more complex. The equations involved in the models trend to be more complex than ever and the data needed for estimations imply a huge computational power.

The estimations cannot be done in modern times without the help of complex software, doubled by complex processing power. On the market, the analysts have two options: to use paid software or free (open source) alternatives. In this paper, we shall present two examples of econometric software – the paid software – Econometric Views and the open source – R.

This article will identify the similarities and, mostly, the differences between open source software and proprietary software in econometric analysis.

2. Content

We start this comparison from the description presented by both of the software.

The latest version of Econometric Software is EViews8. Citing the website of the product, Eviews8 “offers academic researchers, corporations, government agencies and students access to powerful statistical, forecasting, and modeling tools through an innovative, easy-to-use object-oriented interface. EViews blends the best of modern software technology with cutting edge features. The result is a state-of-the art program that offers unprecedented power within a flexible, easy-to-use interface.”

(<http://www.eviews.com/EViews8/ev8main.html>)

* Lecturer, PhD, Faculty of Economic Studies, “Nicolae Titulescu” University, Bucharest (e-mail: mariusjula@univnt.ro).

¹ P. A. Samuelson, T. C. Koopmans, and J. R. N. Stone (1954). "Report of the Evaluative Committee for Econometrica," *Econometrica* 22(2), p. 142.

From the homepage of R software, we learn that “R is a language and environment for statistical computing and graphics(...) R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity. “ (<http://www.r-project.org/>)

Both software solutions are highly appreciated among users and both have strengths and weaknesses.

Open source solution vs. proprietary solution

According to Opensource.org, many people prefer open source software because they have more control over that kind of software.

There are two kind users: the advanced users, with programming skills, who can analyze the code and verify the integrity and consistency and the regular users, who aren't programmers, but also benefit from open source software, because they can use this software for many different purposes and, with some professional help, they can also adapt the solution to their needs.

Some analysts prefer to write their own code, to learn by manual writing formulas. They can learn from following the packages written by others. They can also share their work, their functions or packages with others, inviting comment, critique and also improvement.

Opensource.org suggest that some people prefer open source software because they consider it is more secure and stable than proprietary software. And this is because anyone can view and modify open source software errors or omissions can be easily identified and corrected. Because of the community and many programmers working on a particular open source software package, without any specific requirements for permission from original authors, open source software is generally fixed and updated with a higher rate than proprietary software.

Being open source, the source code is usually highly distributed and there are no concerns about sudden interruption of support. This is particularly important for long time projects.

Platform support

When using specialized software for analysis, one should consider the main operating system before acquiring a license.

The results when comparing the two products are:

Product	Windows	Mac OS	Linux	BSD	Unix
EViews	Yes	Yes	No	No	No
R	Yes	Yes	Yes	Yes	Yes

Table 1 – Comparison between supporting OS

If you are a UNIX user, the answer is already found. The high portability of R is one of the biggest engines that drive the popularity up.

User preferences

There were polls conducted among users for Top Analytics, Data Mining, Big Data software used (<http://www.kdnuggets.com/2012/05/top-analytics-data-mining-big-data-software.html>) and for the first time, the number of users of free/open source software exceeded the number of users of commercial software (according to the results from 2013): R

obtained 30.7% (increase from 23.3% in 2012) and Excel was second with 29.8% (increase from 21.8% in 2012). The study involved 798 voters and, according to the source, there were specific tools for removing unnatural votes.

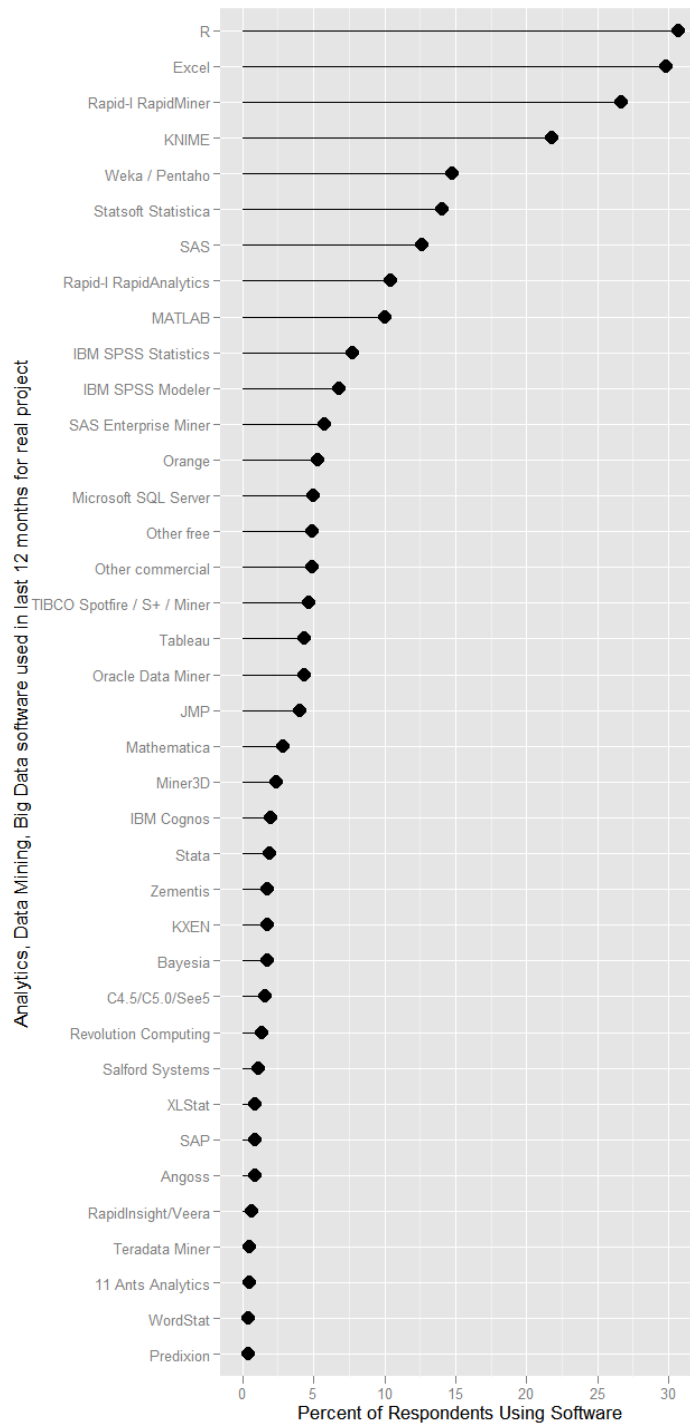


Figure 1 – Analytics, Data Mining, Big Data Software used in last 12 Month

Source: <http://www.kdnuggets.com/2012/05/top-analytics-data-mining-big-data-software.html>

Popularity evolution

In an article from www.r4stats.com, Robert A. Muenchen analyzed the popularity of data analysis software. The author started the paper with simple but important questions that

every user of data analysis software should ask before choosing the tool to work with. Questions like:

- Does it run natively on your computer?
- Does the software provide all the methods you use? If not, how extensible is it?
- Does that extensibility use its own language, or an external one (e.g. Python, R) that is commonly accessible from many packages?
- Does it fully support the style (programming vs. point-and-click) that you like?
- Are its visualization options (e.g. static vs. interactive) adequate for your problems?
- Does it provide output in the form you prefer (e.g. cut & paste into a word processor vs. LaTeX integration)?
- Does it handle large enough data sets?
- Do your colleagues use it so you can easily share data and programs?
- Can you afford it?

To answer these questions and to measure the popularity or market share for different analytic software, the author recommends some interesting approaches:

- job advertisements – considered one of the best tool to observe the trends
- scholarly articles – the young generation will represent the next wave of users
- books – the number and the quality may represent the current interest for a particular software
- website popularity and blogs – the effort to maintain up-to-date online content and the community is a good indicator
- other indicators: survey of use, discussion forum activity, programming activity, popularity measures, IT research firms reports, sales or downloads measures, competition use and growth in capability.

Regarding the last aspect, the author updated his research, and included the growth in R packages (more than 5000 available only on the official site), which is following a rapid parabolic arc.

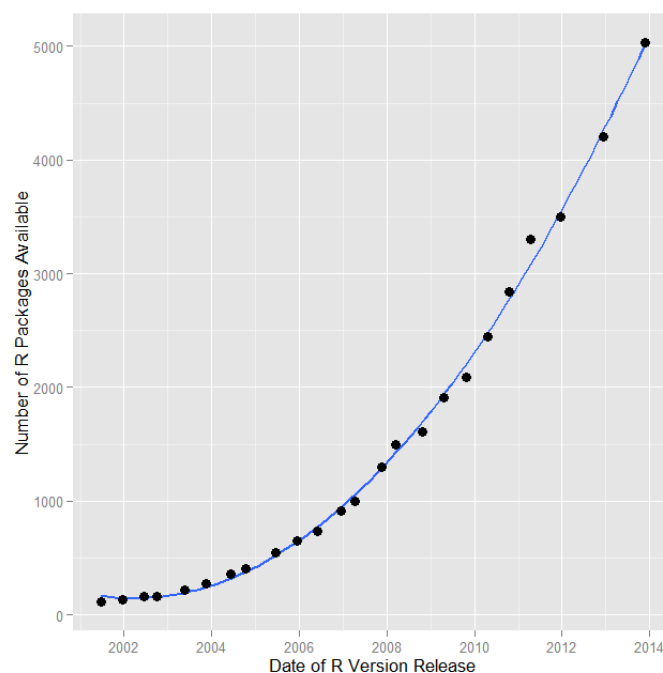


Figure 2 – Number of R Packages Available

Source: <http://www.r-bloggers.com/r-continues-its-rapid-growth/>

Eviews has most of the functions integrated in the main package. But, there are also add-ins available for download. As at the time of the writing of this paper, there were available around 60 add-ins. The interesting aspect is that for running some of them, one should have R installed on the computer. These packages, like TVAR (for estimating a threshold VAR), Mcontrol (A command line tool for solving model objects when there are multiple control and target variables, with or without inequality constraints), ExpSmooth (Performs an expanded set of exponential smoothing and forecasting techniques, including automatic model selection), BVAR (Performs a Litterman / Minnesota / Ko-Ko or Sims-Zha (1998) Bayesian VAR estimation), BMA (Computes different Bayesian Model Averaging methods including LM, GLM and Multinomial Logit models), aim_solve(Provides a way to simulate DSGE models within EViews) or BaiPerron (this add-in performs the Bai-Perron (1998) breakpoints test, as implemented in the R package "strucchange"), should work properly only when in R instance is installed on the computer. One can admit that there is a good opportunity for an analyst used to Eviews to have the opportunity to expand the research by including these new tools from other software.

Pricing

The users are highly influenced not only by the capabilities of the software, but also by the acquisition price. The pricing options for E-Views as the writing date of this paper are:

Single Copy Price

Version	Commercial	Academic
EViews Standard Edition 8	\$1,275	\$510
EViews Enterprise Edition 8	\$1,625	\$650

(<http://www.eviews.com/general/prices/prices.html>)

The difference between the two versions are, according the website: “The Enterprise Edition has all the features of the Standard Edition plus support for ODBC and support for proprietary data formats of commercial data and database vendors. Specifically support for ODBC (including Oracle, SQL Server and DB2), Datastream, EcoWin, FactSet, FAME (local and server), Global Insight DRIBase, Haver Analytics, and Moody’s Economy.com databases, is only provided in EViews Enterprise Edition.”

There is also a student version, for 40\$, aimed to help students learning the econometric analysis, forecasting and statistics.

(<http://www.eviews.com/EViews8/EViews8Student/evstud8.html>)

R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form.

Differences in model approach

An analyst should always carefully read the “vignette” of a new package before using the functions in production. We found different approaches for some indicators and these differences were not only among different econometric software implementations, but inside the same software.

For example, in R, for the same statistic indicator, you can obtain different results when using different packages. You have to use functions within official CRAN repository and, even so, to read the way of the author decided to implement some indicators.

When using for research different software, you may have to compare the results. For example, the Akaike and Schwartz Specification Criterion (AIC and BIC) are different in R vs. EViews:

	R	Obs.	E-Views	Obs.
AIC	$-2l+k*mpar$	K = 2; Npar=degrees of freedom (ex., for univariate model: mpar=3)	$AIC = -\frac{2l}{n} + \frac{2(k+1)}{n}$	K – number of explicative variables
BIC	$-2l+k*mpar$	K=log(n); Npar= degrees of freedom (ex., for univariate model: mpar=3)	$SIC = -\frac{2l}{n} + \frac{(k+1)}{n} \ln(n)$	
Log- Likelihood function	$l = -\frac{n}{2} \left(1 + \ln(2\pi) + \ln \frac{\sum u_t^2}{n} \right)$			

Table 2 – AIC and BIC – R vs. EViews

Software Solution for Econometric Tests: Bai-Perron Breakpoints

In many applications it is useful to know when the structural change occurred. Treating the date of structural change—the breakdate—as an unknown parameter, the issues are how to estimate the breakpoint and how to obtain confidence intervals for the breakpoint.

A Highly possible breakpoint estimate is the date that yields the largest value of the Chow test sequence. Usually this is known to be a good estimate only in one special case—in linear regressions when the Chow test is constructed with the “homoskedastic” form of the covariance matrix.

In regression models, a suitable method to estimate the parameters—including the breakpoint—is least squares. Operationally, the sample is split at each possible breakpoint, the other parameters assessed by ordinary least squares and the sum of squared errors calculated and stored. The least squares breakpoints estimate is the date that minimizes the full-sample sum of squared errors (equivalently, minimizes the residual variance).

A theory of least squares estimation has been developed in a sequence of papers by Jushan Bai (alone or with coauthors). Bai (1994, 1997a) derives the asymptotic distribution of the breakpoint estimator and shows how to construct confidence intervals for the breakpoint.⁸ These confidence intervals are easy to calculate and hence are very useful in applications, as they indicate the degree of estimation accuracy. Bai, Lumsdaine and Stock (1998) extend this analysis to multiple time series with simultaneous structural breaks. They show that using multiple time series improves estimation precision. Bai and Perron (1998) discuss simultaneous estimation of multiple breakpoints.

Chong (1995) and Bai (1997b) show how to estimate multiple breakpoints sequentially. The key insight is that when there are multiple structural breaks, the sum of squared errors (as a function of the breakpoint) can have a local minimum near each breakpoint. Thus, the global minimum can be used as a breakpoint estimator, and the other local minima can be viewed (cautiously) as candidate breakpoint estimators. The sample is then split at the breakpoint estimate, and analysis continues on the subsamples. Bai (1997b) shows that important improvements are obtained by iterative refinements: re-estimation of breakpoints based on refined samples.

Hansen B., in his article from 2001, *The New Econometrics of Structural Change: Dating Breaks in U.S. Labor Productivity*, considers that “as yet, there have been few rigorous applications of the methodology described here. An important exception is Bai,

Lumsdaine and Stock (1998), who attempt to date the alleged slowdown of the early 1970s. Using U.S. quarterly data for 1959 through 1995 on real output, consumption and investment, they find no evidence of structural change when examining the individual series with univariate models, but find strong evidence in a joint vector autoregression, in which the output, consumption and investment variables are regressed on lagged values of output, consumption and investment. Their estimate of the breakpoint is the first quarter of 1969, and their 90 percent confidence interval puts the breakpoint between the second quarter of 1966 and the fourth quarter of 1971. An interesting common feature of our breakpoint estimates for labor productivity and the Bai, Lumsdaine and Stock (1998) estimate for real output is that they are both quite different than the widely accepted breakpoint of 1973. (Neither set of confidence intervals include 1973.) Another common feature is that both sets of estimates are consistent with a structural break in the late 1960s”.

Let's suppose we have to find the breakpoints in regression relationships. From the description found in R help: “The foundation for estimating breaks in time series regression models was given by Bai (1994) and was extended to multiple breaks by Bai (1997ab) and Bai & Perron (1998). “breakpoints” implements the algorithm described in Bai & Perron (2003) for simultaneous estimation of multiple breakpoints. The distribution function used for the confidence intervals for the breakpoints is given in Bai (1997b). The ideas behind this implementation are described in Zeileis et al. (2003).”

Let a series of 150 records, created in Microsoft Excel. We define a function Y_t as following:

$$Y_t = \begin{cases} 2 + 3X_t + e_t, & \text{for } t = 1 \dots 50 \\ 300 - 3X_t + e_t, & \text{for } t = 51 \dots 100 \\ -100 + X_t + e_t, & \text{for } t = 101 \dots 150 \end{cases}$$

If we want to compute the breakpoints for this, we can use the following code in R:

```
#reading data from MS Excel - copy from Clipboard
read.excel <- function(header=TRUE,...)
{
  read.table("clipboard",sep="\t",header=header,...)
}

#loading data into variable "dat"
bp=read.excel()

#displaying first 6 records
head(bp)

a<-lm(V1~seq(1,dim(bp)[1],by=1), data=bp)
summary(a)
#creating a time variable equal with series dimension
time<-seq(1,dim(bp)[1],by=1)

#loading library "strucchange"
require(strucchange)

#computing breakpoints
```

breakpoints(V1~1+time, data=bp, h=0.1)

The results indicate:

Breakpoints at observation number:
52 97

In EViews, the procedure is:

- we estimate an equation like: $V1 = C(1) + C(2)*@TREND$
- From View -> Stability Diagnostics -> Multiple Breakpoint Test

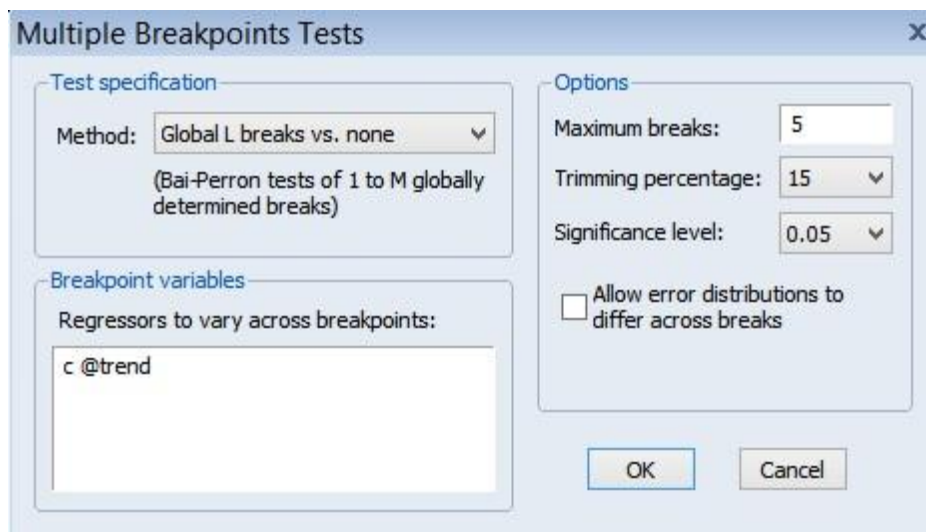


Figure 3 – Breakpoints in Eviews8

- The obtained results are:

Equation: EQ01 Workfile: BAI_PERRON::Untitled\

View Proc Object Print Name Freeze Estimate Forecast Stats Resids

Sample: 1 150
 Included observations: 150
 Breakpoint variables: C @TREND
 Break test options: Trimming 0.15, Max. breaks 5, Sig. level 0.05

Sequential F-statistic determined breaks:	5
Significant F-statistic largest breaks:	5
UDmax determined breaks:	2
WDmax determined breaks:	2

Breaks	F-statistic	Scaled F-statistic	Weighted F-statistic	Critical Value
1 *	168.7513	337.5026	337.5026	11.47
2 *	2549.436	5098.872	5998.365	9.75
3 *	1742.416	3484.831	4781.222	8.36
4 *	1320.860	2641.721	4214.261	7.19
5 *	1054.144	2108.287	4133.685	5.85

UDMax statistic*	5098.872	UDMax critical value**	11.70
WDMax statistic*	5998.365	WDMax critical value**	12.81

* Significant at the 0.05 level.
 ** Bai-Perron (Econometric Journal, 2003) critical values.

Estimated break dates:
 1: 60
 2: 53, 98

Figure 4 – Breakpoints in Eviews8 – Results

Source: Author's calculations

We have to emphasize the slight differences between the two software, regarding the results. In R, the breakpoints are the number of observations that are the last in one segment. In EViews, as the help section stated: “results follow the EViews convention in defining breakdates to be the first date of the subsequent regime”

3. Conclusions

Companies and individual researchers are considered rational in their economic behavior, so they will tend to go for that particular software which delivers best results with minimum costs. When we refer to cost, we should consider not only acquisition, maintaining and updating costs for the software, but also training and supporting materials, user community and online/offline support. Also the time spent from data input to outputs should be considered. The diversity of supported data and the output customization options can weight in the choosing process.

With the R environment becoming more and more divers, the solution for commercial software should be either to adopt a better pricing strategy or to come with other services, like a plus on support system, free training courses or outstanding computing performance that should justify the invested money.

If in the past there were reservations in using open source software because of the lack of support, there are companies that can provide the requested help for adopting or creating a suitable software solution, for a fee.

The threat is not only for software like E-Views, but for all commercial software which share a common pool of users. The competition is even stronger in a field like data analysis because of the reduced number of total possible users, compared with other domains.

References

- Bai J., 1994. "Least Squares Estimation of a Shift in Linear Processes." *Journal of Time Series Analysis*. 15:5, pp. 453–72.
- Bai J., Lumsdaine R., Stock J.H., 1998. "Testing for and Dating Common Breaks in Multivariate Time Series." *Review of Economic Studies*. July, 64:3, pp. 395–432.
- Bai J., Perron P., 1998. "Estimating and Testing Linear Models with Multiple Structural Changes." *Econometrica*. January, 66:1, pp. 47–78.
- Bai J., 1997a. "Estimation of a Change Point in Multiple Regression Models." *Review of Economics and Statistics*. November, 79:4, pp. 551–63.
- Bai J., 1997b. "Estimating Multiple Breaks One at a Time." *Econometric Theory*. June, 13:3, pp. 315–52.
- Goldstein, H. (1995), *Multilevel Statistical Models*, 2nd edition. London: Edward Arnold.
- Hansen B.E. (2001), "The New Econometrics of Structural Change: Dating Breaks in U.S. Labor Productivity", *Journal of Economic Perspectives—Volume 15, Number 4*, p.117-128
- Muenchen R.A. (2013), *The Popularity of Data Analysis Software*, <http://r4stats.com/articles/popularity/>
- Perron, P., 1989. "The Great Crash, the Oil-Price Shock, and the Unit-Root Hypothesis." *Econometrica*. November, 57:6, pp. 1361–401.
- Perron, P., 1997. "Further Evidence on Breaking Trend Functions in Macroeconomic Variables." *Journal of Econometrics*. October, 80:2, pp. 355–85.
- Perron, P., Vogelsang T.J., 1992. "Nonstationarity and Level Shifts with an Application to Purchasing Power Parity." *Journal of Business and Economic Statistics*. July, 10:3, pp. 301–20.
- Rabe-Hesketh, Sophia and Anders Skrondal (2012), *Multilevel and Longitudinal Modeling Using Stata*. 3rd edition. Volume I: Continuous Responses. College Station, Texas: Stata Press.
- Raudenbush, S.W. and Bryk, A.S. (2002), *Hierarchical Linear Models. Applications and Data Analysis Methods*, 2nd edition. Sage: Thousand Oaks. <http://www.kdnuggets.com/2012/05/top-analytics-data-mining-big-data-software.html>
- Samuelson P. A., Koopmans T. C., Stone J. R. N. (1954). "Report of the Evaluative Committee for Econometrica," *Econometrica* 22(2), p. 142
- www.eviews.com
- www.opensource.com/resources/what-open-source
- www.r4stats.com
- www.r-project.org

Annex

Eviews – Equation Estimation

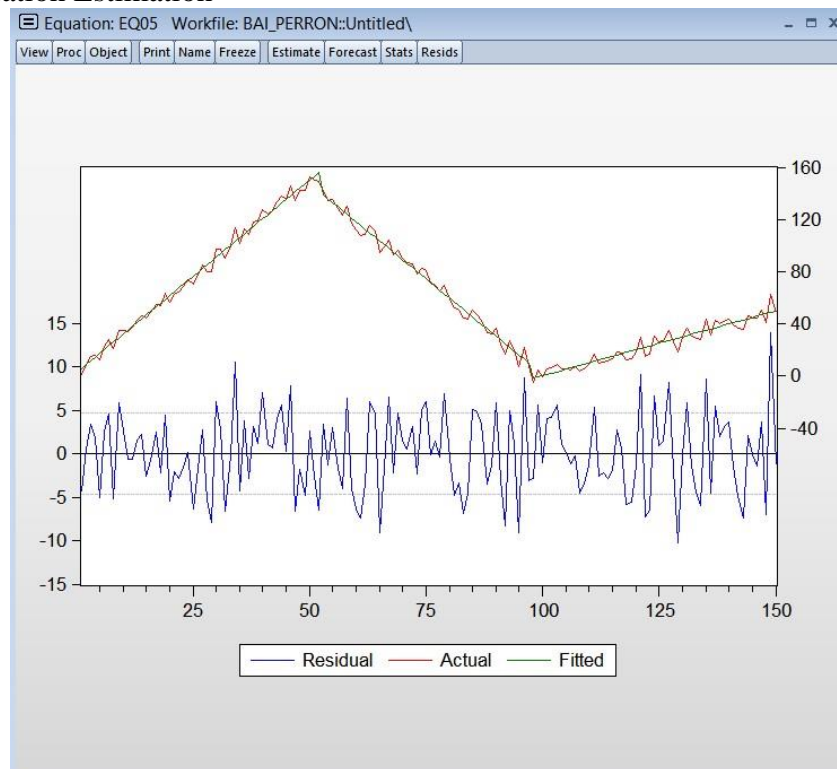


Figure 5 – Estimation in Eviews8

Source: Author's calculations

R source code for breakpoints

```
> #reading data from MS Excel - copy from Clipboard
> read.excel <- function(header=TRUE,...)
+ {
+ read.table("clipboard",sep="\t",header=header,...)
+ }
> #loading data into variable "dat"
> bp=read.excel()
> #displaying first 6 records
> head(bp)
      v1
1  1.329643
2  9.071669
3 14.984144
4 16.271895
5 12.382279
6 22.829745
> a<-lm(v1~seq(1,dim(bp)[1],by=1), data=bp)
> summary(a)
```

```
Call:
lm(formula = v1 ~ seq(1, dim(bp)[1], by = 1), data = bp)
```

```
Residuals:
    Min     1Q  Median     3Q     Max
-89.54 -28.74  -4.44  25.92  82.85
```

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	91.30382	6.46533	14.122	< 2e-16 ***
seq(1, dim(bp)[1], by = 1)	-0.43080	0.07428	-5.799	3.89e-08 ***

 Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 39.39 on 148 degrees of freedom
 Multiple R-squared: 0.1852, Adjusted R-squared: 0.1797
 F-statistic: 33.63 on 1 and 148 DF, p-value: 3.886e-08

```
> #creating a time variable equal with series dimension
> time<-seq(1,dim(bp)[1],by=1)
> #time
> #loading library "strucchange"
> require(strucchange)
> #computing breakpoints
> breakpoints(v1~1+time, data=bp, h=0.1)
```

Optimal 3-segment partition:

Call:

```
breakpoints.formula(formula = v1 ~ 1 + time, h = 0.1, data = bp)
```

Breakpoints at observation number:

52 97

Corresponding to breakdates:

0.3466667 0.6466667