Distr.
GENERAL

Working Paper No. 15
28 March 2008

ENGLISH ONLY

**UNITED NATIONS STATISTICAL COMMISSION and
ECONOMIC COMMISSION FOR EUROPE
CONFERENCE OF EUROPEAN STATISTICIANS**

**EUROPEAN COMMISSION
STATISTICAL OFFICE OF THE
EUROPEAN COMMUNITIES (EUROSTAT)**

**ORGANISATION FOR ECONOMIC COOPERATIO
AND DEVELOPMENT (OECD)
STATISTICS DIRECTORATE**

Meeting on the Management of Statistical Information Systems (MSIS 2008)
(Luxembourg, 7-9 April 2008)

Topic (iii): Exchange/sharing/re-use of components, common models among statistical offices

R: An Open Source Statistical Environment

Invited Paper

Prepared by Valentin Todorov, UNIDO

I. INTRODUCTION

1. The Open Source movement has changed dramatically the global software landscape in the recent decades. If we look at any software area we will find prominent representatives of the Open Source Software/Free Software (OSS/FS, also abbreviated as FLOSS/FOSS) like Linux, Apache, MySql, Perl, PHP, OpenOffice, Mozilla Firefox. The exact definition of what Open Source is and what it is not can be found at the home page of the Open Source Initiative, but briefly speaking, programs developed as OSS/FS are programs with a licenses giving the users the freedom to redistribute them in any form, to use them for any purpose, to have access to the complete source and have the freedom to modify it and to redistribute the modified programs, of course without having to pay any royalties to the original developers. An extensive quantitative evaluation of the Open Source approach can be found in Wheeler (2007).

2. In the world of commercial statistical software there are only a few very well known names that dominate - SAS, SPSS, STATA, S-PLUS, MATLAB. The situation with the free software is similar. Although there are hundreds and hundreds of free tools for solving a given statistical problems, if we talk about a comprehensive statistical environment which could be competitive to the dominating commercial packages, the choice is not much and we end always with R.

3. The goal of this paper is to present a brief overview of the Open Source statistical language and environment R, pointing out its advantages (and disadvantages) when compared to the commercial statistical packages dominating the market for statistical software. Since nowadays it is very easy to find any information (useful or not) in Internet, not many references are included and the included ones are either those that I have used for the preparation of this material or such that are not easy to find.

II. WHAT IS R

A. The R Platform

3. As described by the R-core development team on its web page, R is “a system for statistical computation and graphics. It provides, among other things, a programming language, high-level graphics, interfaces to other languages and debugging facilities.”

4. R is a GNU project, which was developed after and can be considered a different implementation of the S language and environment, with similar syntax and features. The S language was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues in the mid-seventies. One of the big names in the world of the commercial statistical software is S-Plus, which is a value added implementation of the S language and now is marketed by Insightful Corporation. Despite the very close similarities with S and the superficial similarities with the C language, actually the R engine is significantly influenced by Scheme, a Lisp dialect. Nevertheless, much code written for S runs unaltered under R.

5. The development of the R language and environment first started in 1990 as an experimental project by Ihaka and Gentleman, both from the laboratory of statistics at the University of Auckland (New Zealand), in 1993 a preliminary version of R was presented and already in 1995 R was released under the GNU Public License. Now the development of R is managed by the R-core team consisting of 17 members including John Chambers.

6. R provides a wide variety of statistical (linear and non-linear modelling, classical statistical tests, time-series analysis, classification, clustering, robust methods and many more) and graphical techniques.

7. A general collection of useful information for users on **all** platforms (Linux, Mac, Unix, Windows) can be found in R FAQ. Additionally there are two platform-specific FAQs for Windows and MacOS.

B. R Availability

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

9. R can be obtained as both source and binary (executable) forms from the Comprehensive R Archive Network (CRAN). The source files are available for a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux) as well as for Windows and MacOS for which are available also precompiled binary distributions of the base system and contributed packages.

10. The most recent release of R is version 2.6.2 (released on 8.February 2008) and pre-release versions of 2.7.0 are in progress.

11. A wide variety of add-on functionality (actually the normal way of extending R) is available from the same web page in the form of contributed R packages, which can be downloaded in source form or installed directly from the R console by using the *install.packages()* function (provided the computer is connected to the Internet).

C. Is R harder to learn/use than other statistical packages?

12. One of most popular criticisms against R is that this statistical language is hard to learn, compared to the other statistical packages, like SAS and SPSS and it is said to have a very steep learning curve. Quoting Kabacoff (2008) ”I have been a hardcore SAS and SPSS programmer

for more than 25 years, a Systat programmer for 15 years and a Stata programmer for 2 years. But when I started learning R recently, I found it frustratingly difficult”. On the other hand, I have not used SAS for 25 years, but I have programmed in C as long, I have not used Systat or Stata for many years, but I have programmed in Fortran, Java, C# and many other (non statistical) programming languages. And for me R was not harder to learn than any of these (non statistical) languages.

13. As mentioned in Muenchen (2007), while SAS and SPSS have a wide variety of functions and procedures, all these fall into one of five categories and these are:
(a) Data input and management statements for reading, transforming and organizing the data
(b) Statistical and graphical procedures for analysing the data
(c) An output management system for formatting the output from statistical procedures or for customizing printed output. In SAS this is done by the Output Delivery System (ODS) while in SPSS it is done by the Output Management System (OMS).
(d) A macro language to allow creating of programs, i.e. repeatedly executing statements, functions and procedures
(e) A matrix language for creating new algorithms. This language is SAS/IML in SAS and SPSS Matrix in SPSS.

14. In SAS and SPSS these five areas are handled with different systems, but for the sake of simplicity the introductory training in these packages involves mainly the first two (data management, statistical analysis and graphics) and many of the users stay with this knowledge and never learn the more advanced topics. On the other hand, in R all these five areas are interrelated in such a way that the user must approach them in parallel, which could be difficult for the novice. But the integration of these five areas gives R a significant advantage in power which allowed most of the R procedures to be written in the same interpreted language and thus the source code of these procedures is available for viewing and modifying by the user.

D. R Graphics

15. One of the most important strengths of R is the ease with which simple exploratory graphics as well as well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

16. A simple example of basic R graphics is shown in ***Figure 1*** (produced by the code below) while ***Figure 2*** shows a trellis-type graphic (to my knowledge this type of graphics are not available in most of the other statistical packages). Finally, ***Figure 3*** shows an example of time series diagnostic graphics.

```
daten <- faithful[faithful$eruptions > 3, 2]
par(mfrow = c(2, 2))
hist(daten, freq = FALSE, breaks = 20, xlim = c(60,100))
boxplot(daten, main = "Boxplot")
plot(density(daten), main = "Estimated Density")
qqnorm(daten)
qqline(daten, col = "red")
```

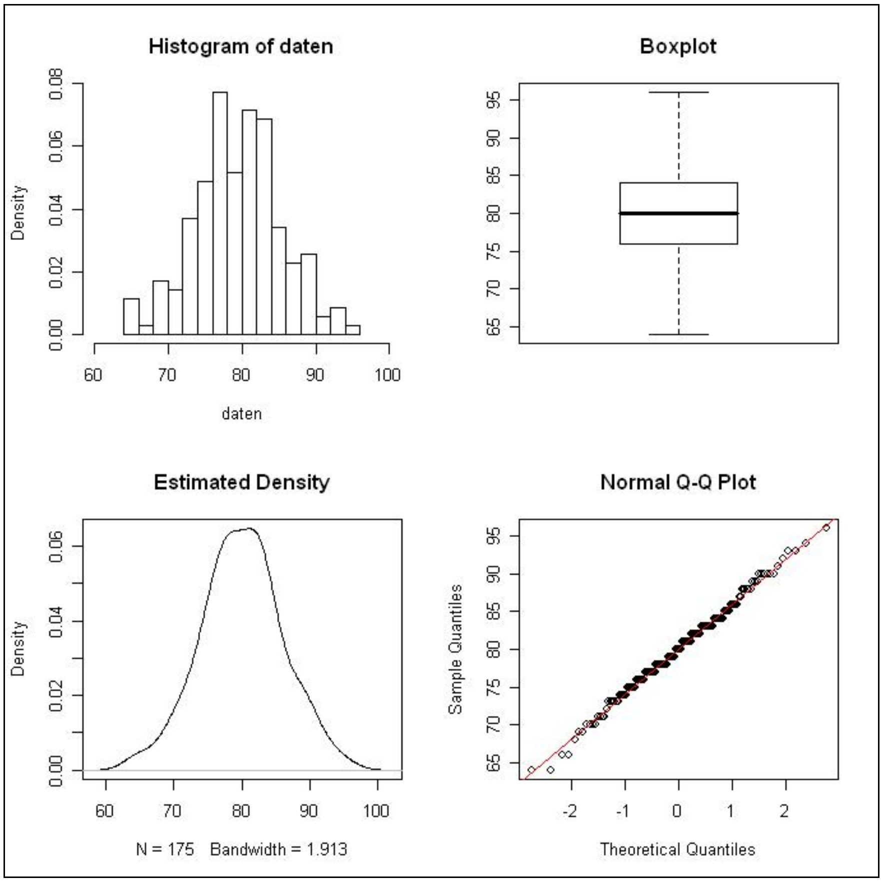


Figure 1:A simple example of R basic graphics

17. R can produce graphics in many formats, including:
(a) On screen
(b) PDF files for including in LATEX or for direct distribution
(c) PNG or JPEG bitmap formats for the WEB
(d) On Windows, metafiles for Word, PowerPoint, and similar programs

18. An exciting example of the R graphics capabilities is the R Graph Gallery which aims to present many different graphics fully created with the programming environment R. Graphs are gathered in a MySQL database and browsable through PHP. An excellent reference to R Graphics is the book of Paul Murrell, a member of the R Core Development Team who has not only been the main author of the *grid* package but has also been responsible for several recent enhancements to the underlying R graphics engine.

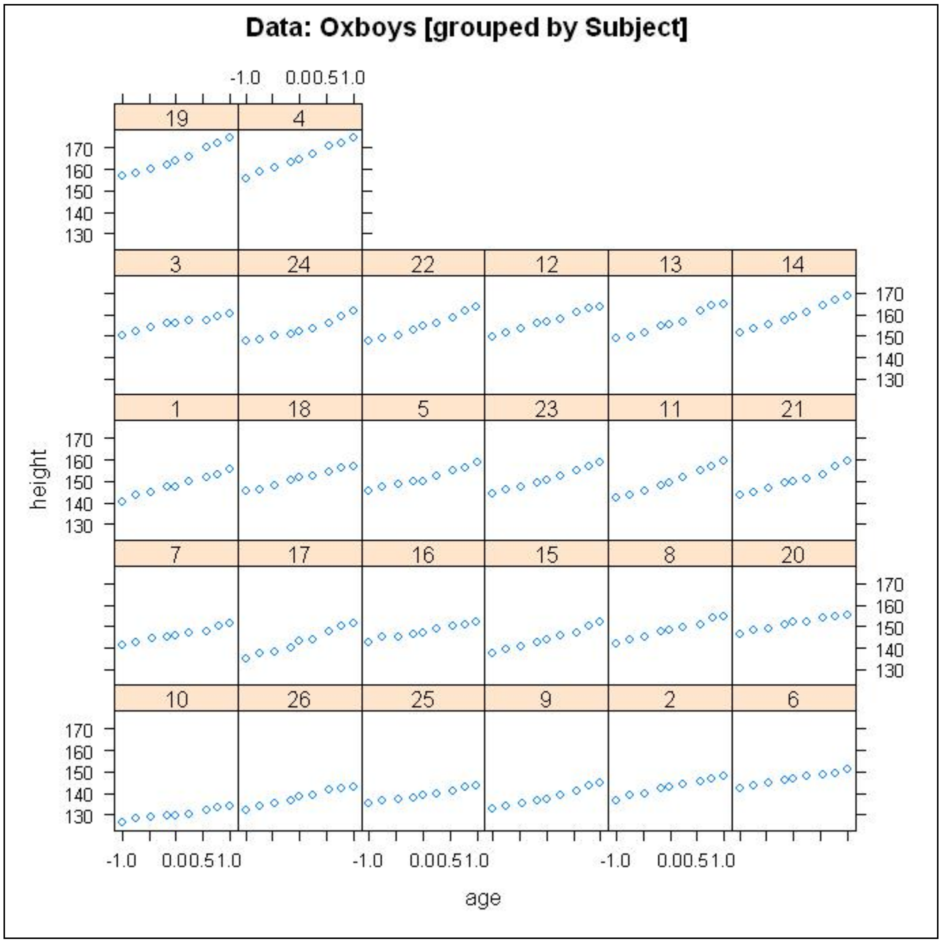


Figure 2:An example of multipanel graphic display (trellis graphics) in R

E. R Extensibility (R Packages)

19. One of the most important features of the R language and one of the main topics in which R beats the commercial version S-Plus, is its extensibility by creating packages of functions and data. This was one of the key features that has contributed to the R's growth. The first step in R programming is writing R functions for performing a given repetitive action. Next, to facilitate the reusability of such functions they can be combined into a package.

20. The R package mechanism was first designed to help the developers to encapsulate related programs, data and documentation and distribute them to the users. Now this mechanism is the natural way of extending R. Numerous researchers create R packages and post them on the special area of the Comprehensive R Archive Network (CRAN). The R environment provides tools for downloading and installing packages, for creating packages from scratch and extending them, for writing and incorporating online help pages as well as extended documentation in PDF format. A package can contain R code, documentation files in a special format which provide both online help as well as printed manual and example data sets, but could contain also compiled C or Fortran source code which is automatically compiled when building the package.

21. The package 'check' tool is invaluable for the package developer and a positive result from the check is a must for posting a package on CRAN. Apart from the formal code validation, the check procedure includes running all examples from every help page and will build and test the package vignettes, if available. It is possible to write test cases in the form of R programs, which

will be run, and the output will be compared with the previously stored output. This guarantees that no side effects, which broke already working code, appeared with the last changes to the package.

22. Currently more than 1300 packages exist on CRAN covering a wide variety of statistical methods and algorithms, including the newest achievements in the statistical science. There are about a dozen ‘base’ packages, which together with the packages denoted as ‘recommended’ are included in all binary distributions of R.

F. R and the Others (Interfaces)

23. When using a statistical system we must have in mind that this is not done in isolation and the system must be able to communicate with other systems in order to import data for analysis, to export data for further processing (use the right tool for the right work) and to export results for report writing.

24. A rich variety of facilities for data import and export as well as for communication with databases, other statistical systems and programming languages are available either in R itself or through packages available from CRAN.

(a) The easiest data format to import into R is a simple text file but reading XML, spreadsheet like data, e.g. from Excel is also possible;
(b) The recommended package foreign provides import facilities for reading data in the format of the statistical packages Minitab, SAS, S-Plus, SPSS, STATA, Systat and Octave as well as export capability for writing STATA files, while the package matlab provides emulation for Matlab;
(c) Working with large data sets could be a problem in R (if the data do not fit in the RAM of the computer) but the interface to RDBMS could help in such cases. Another limitation is that R does not easily support concurrent access to data, i.e. if more than one user is accessing, and perhaps updating, the same data, the changes made by one user will not be visible to the others. This could also be solved by using the interface to relational databases. There are several packages available on CRAN for communication with RDBMSs, providing different levels of abstraction. All have functions to select data within the database via SQL queries, and to retrieve the result as a whole, as a data frame or in pieces (usually as groups of rows). Most packages are tied to a particular database – ROracle, RMySQL, RSQLite, RmSQL, RPgSQL, while the package RODBC provides a generic access to any ODBC capable relational database.
(d) R is an interpreted language and some very computation intensive algorithms could be slow – in this case a native code implemented in C or FORTRAN is the right solution;
(e) R has no real (statistical) GUI, which is often criticized by the proponents of point-and-click statistical packages, but if it is necessary to develop a nice specialized graphical user interface, this could be implemented in Java and R will do the computations in the background

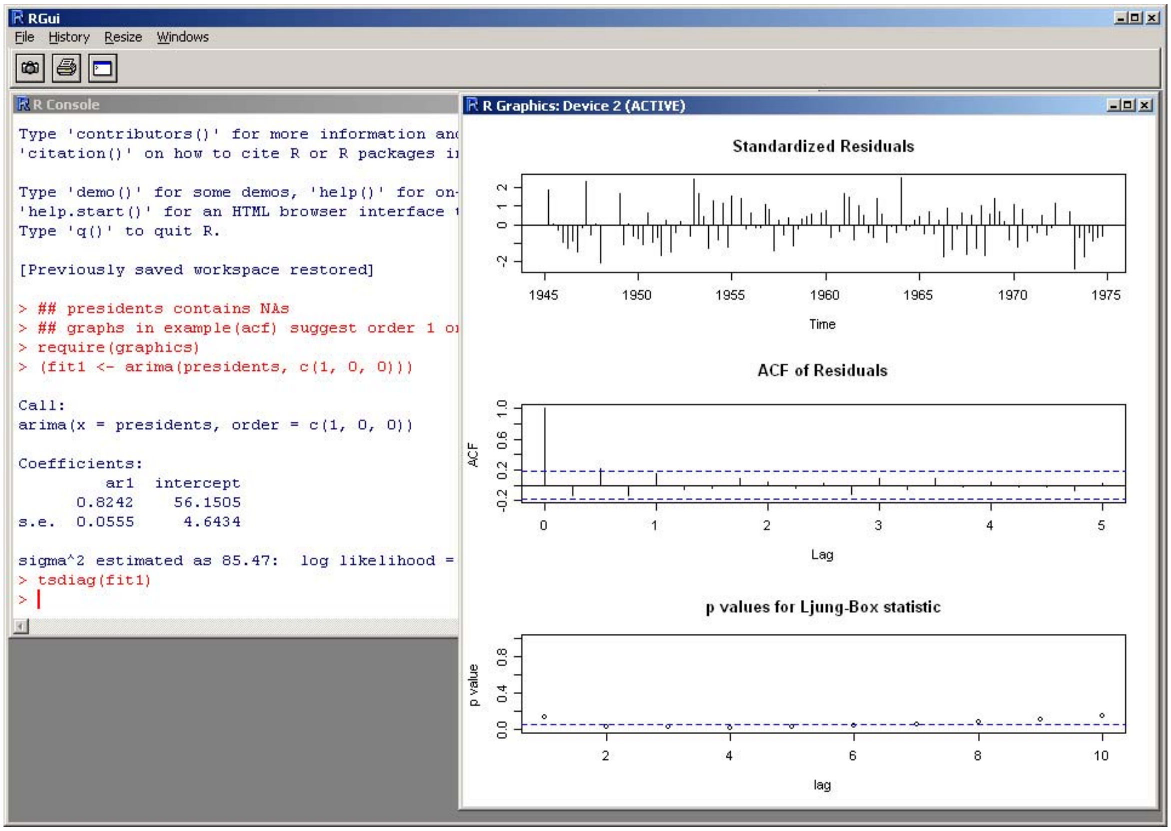
G. R for Time Series

25. R has extensive facilities for analysing time series in the packages *stats, tseries, zoo, its* (irregular time series), *ast* (not yet on CRAN), *pastecs* (for analysing space-time ecological time series) and *lmtest*. Vito Ricci has compiled a reference card of the most popular time series functions – see Ricci (2008). The package *stats* includes classical time series modelling tools - *arima*() for ARIMA modelling and Box-Jenkins-type analysis. For fitting structural time series is available *StructTS*() in *stats* and for time series filtering and decomposition can be used *decompose*() and *HoltWinters*(). The package *forecast* supplements the tools available in the *stats* package by providing additional forecast methods, and graphical tools for displaying and

analysing the forecasts. Further information on time series functions and packages in R can be found in the Task View Econometrics - http://cran.r-project.org/web/views/Econometrics.html.

26. The functionality for analysing monthly or lower frequency time series data which is implemented in the software packages **TRAMO/SEATS** (Time series Regression with ARIMA Noise, Missing values and Outliers/Signal Extraction in ARIMA Time Series) and **X-12-ARIMA** seasonal adjustment software of the US Census Bureau is easily accessible through the *Gretl* library – see Cottrell (2008).

27. In ***Figure 3*** is shown an example of time series analysis functions **-** *arima*() for fitting an ARIMA model to a univariate time series and *tsdiag*() for plotting time series analysis diagnostics



```
Type 'contributors()' for more information anc
'citation()' on how to cite R or R packages i1

Type 'demo()' for some demos, 'help()' for on-
'help.start()' for an HTML browser interface 1
Type 'q()' to quit R.

[Previously saved workspace restored]

> ## presidents contains NAs
> ## graphs in example(acf) suggest order 1 o1
> require(graphics)
> (fit1 <- arima(presidents, c(1, 0, 0)))

Call:
arima(x = presidents, order = c(1, 0, 0))

Coefficients:
         ar1  intercept
      0.8242    56.1505
s.e.  0.0555     4.6434

sigma^2 estimated as 85.47:  log likelihood =
> tsdiag(fit1)
> 
```

Figure 3: Example of time series analysis functions: arima() for fitting an ARIMA model to a univariate time series and tsdiag() for plotting time series analysis diagnostics

H. R for Survey Analysis

28. Complex survey samples are usually analysed by specialized software packages. From the most well known general-purpose statistical packages Stata provides much more comprehensive support for analysing survey data than SAS and SPSS and could successfully compete with the specialized packages. In R functionality for survey analysis is offered by several add-on packages, the most popular being the *survey* package. Detailed information can be found in the manuals of the package as well as from its home page, maintained by the author, Thomas Lumley at http://faculty.washington.edu/tlumley/survey/, but here is a brief overview:

(a) Designs incorporating stratification, clustering, and possibly multistage sampling, allowing unequal sampling probabilities or weights; multistage stratified random sampling with or without replacements

(b) Summary statistics: means, totals, ratios, quantiles, contingency tables, regression models, for the whole sample and for domains
(c) Variances by Taylor liberalization or by replicate weights (BRR, jack-knife, bootstrap, or user-supplied)
(d) Post-stratification and raking
(e) Graphics: histograms, hexbin scatterplots, smoothers.

29. Other relevant R packages are *pps, sampling, sampfling,* all of which focus on design, in particular PPS sampling without replacement.

I. R and SDMX

30. No, there is nothing of the kind but this is not much different from the other statistical software packages. There exists a proposal for a new data exchange format for statistical data based on XML, which is called StatDataML – see Meyer (2004) – and a corresponding implementation in the R package *StatDataMl*.

J. R and the Outliers (Robust Statistics in R)

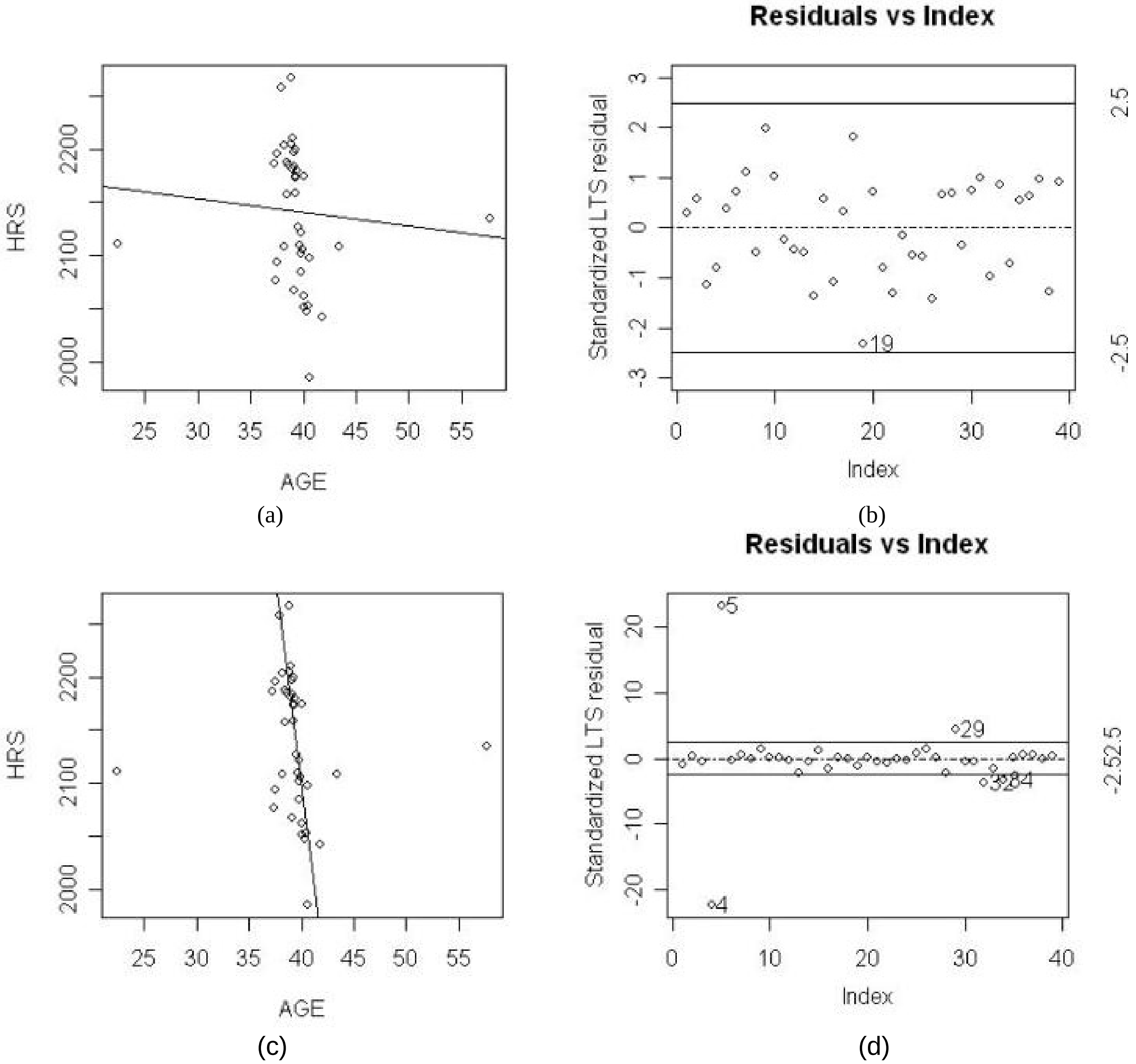
31. Atypical observations, which are inconsistent with the rest of the data or deviate from the postulated model, usually called outliers, are likely to appear often in the data sets under consideration. Unfortunately most of the classical statistical methods are very sensitive to such data. Therefore *robust* statistical methods are developed whose main goal is to produce reasonable results even when one or more outliers may appear anywhere in the data.

32. As an example let us consider the "Wages and Hours" data set available at http://lib.stat.cmu.edu/DASL/. The data are from a national sample of 6000 households with a male head earning less than $15,000 annually in 1966. The data were classified into 39 demographic groups (if the cases with missing data are removed only 28 groups remain). The study was undertaken to estimate the labour supply (average hours) from the available data. There are 9 independent variables but for the sake of the example we will consider only one - the average age of the respondents, i.e. if y = labour supply and x = average age of the respondents we will fit the model $y = \beta_0 + \beta_1 x$.

33. ***Figure 4*** (a) shows a scatterplot of the data with the line fitted by the Ordinary Least Squares (OLS) method. It is clearly seen that two observations are outliers – assuming that the measurements are correct, the average age of the people in group 3 is very low compared to the others while the age of the people in group 4 is on average too high. The line shown in (a) does not fit well the data since it is attracted by the outliers. The outliers fall inside the tolerance band in the residual plot presented in (b). The line fitted by the robust Least Trimmed Squares (LTS) method, presented in (c) resists the outliers and follows the majority of the data. The corresponding residual plot shown in (d) clearly identifies the outliers. For further examples based on this data set as well as other information and references about robust statistics see Hubert et al. (2004).

34. SAS and Stata have functions for robust regression (*PROC ROBUSTREG* and *rreg* respectively) while SPSS has no such capability. In R robust methods are available in many packages, the most well known being *MASS, robustbase, rrcov, robust.* The computation and the diagnostic plots shown in ***Figure 4*** were produced by the function *ltsReg()* from package *robustbase.*

35. A method for visualization of missing data and robust imputation is developed by Templ et al. (2008) and implemented in R (not yet on CRAN)



(a) (b) (c) (d)

Figure 4: Wages and Hours data: Scatter plot of the data with LS (a) and LTS (c) line as well as their residual plots – (b) and (d) respectively

K. More R

36. There are many more topics about R, which deserve our attention but were not considered here because of time and space limitations. A brief list is in order.

(a) **R and the WEB** – there are several projects that provide possibility to use R as a service over the Web. The first one was Rweb – see Rweb Home Page, which provides three types of interface: a simple text version, a more sophisticated JavaScript based interface and a point-and-click interface, mainly suitable for teaching statistics. Information about other Web projects for R is available from R FAQ.

(b) **R and the Missing -** Missing values in both character and numeric variables are represented by the symbol NA (not available) and most modelling functions offer options for dealing with missing values. Advanced handling of missing values is available through a number of packages:

i. *mvnmle*: ML estimation for multivariate normal data with missing values;

ii. *mitools*: Tools for multiple imputation of missing data and to perform analyses and combine results from multiple-imputation datasets;

iii. *mice* - Multivariate Imputation by Chained Equations;
iv. *EMV*: Estimation of Missing Values for a Data Matrix

(c) **R GUI.** As already mentioned, R has no (statistical) GUI and this is a reason often to be criticized by the proponents of the point-and-click statistical packages. Nevertheless also statistical GUIs are emerging and already several packages exist like *Rcomander* and *Sciviews*.

(d) **R Objects.** One of the main features of R is that it is an object oriented (although not in the sense C++, Java and C# are) language.

III. SUMMARY

37. In the following *Table 1* is presented a summary of the main differences between R and SAS and SPSS. The second part of the table is a comparison of the SAS and SPSS products to the approximate equivalent R packages. The table was compiled mainly from Muenchen (2007) but only the topics, which are of interest for our work, are considered.

Topic	SAS	SPSS	R
Output Management System	Rarely used for routine work		Output is easily passed from one function to another to do further processing and obtain more results
Macro language	A special language used for performing repetitive tasks and adding new functionality. The new functions are not run in the same way as the built-in procedures		R itself is a programming language. The added new functions are run exactly in the same way as the built-in ones.
Matrix language	A special language used for adding new functionality. The new functions are not run in the same way as the built-in procedures		The base R itself is a vector- and matrix-based language, and it ships with many powerful tools for doing matrix manipulations. These are complemented by the packages Matrix and SparseM.
Publishing results	Cut and paste to a Word processor or exporting to a file		There are possibilities to produce Tex output (including graphics) using the Sweave package
Data size	Limited by the size of the disk		Limited by the size of the RAM, (not trivial) usage of databases for large data sets is possible
Data structure	Rectangular data set		Rectangular data frame, vector, list
Interface to other programming languages	Not available		R can be easily mixed with Fortran, C, C++ and Java

Source code	Not available		The source code of R as well as of the R packages is a part of the distribution
Basics	SAS®	SPSS Base™	R
Data Access	SAS/ACCESS®	SPSS Data Access Pack™	DBI, RODBC, foreign
Data Mining	Enterprise Miner™	Clementine®	rattle, arules, FactoMineR
Geographic Information Systems /Mapping	SAS/GIS®, SAS/Graph®	SPSS Maps™ (no full GIS)	maps, mapdata, mapproj, GRASS via spgrass6, RColorBrewer, see Spatial in Task Views
GUI	Enterprise Guide®	SPSS Base™	JGR, R Commander, pmg, Sciviews
Graphics	SAS/GRAPH®	SPSS Base™	ggplot, gplots, grid, lotrix, graphics, gridBase, hexbin, lattice, vcd, vioplot, scatterplot3d, geneplotter, Rgraphics,
Dynamic Graphics	SAS/INSIGHT®	None	GGobi via rggobi iPlots, Mondrian via Rserve
Matrix/Linear Algebra	SAS/IML®	SPSS Matrix™	R, matlab, Matrix, sparseM
Missing Values Imputation	SAS/STAT®: MI	SPSS Missing Values Analysis™	aregImpute (Hmisc), EMV, fit.mult.impute (Design), mice, mitools, mvnmle
Sampling, Complex or Survey	SAS/STAT®: surveymeans, etc.	SPSS Complex Samples™	pps, sampfling, sampling, spsurvey, survey
Time Series	SAS/ETS®	SPSS Trends™ Expert Modeler	Many (> 40) packages - see Task View Econometrics.

Table 1: Functionality of R, SAS and SPSS

IV. REFERENCES

Banfield, J. (1999) Rweb:Web-based Statistical Analysis, *Journal of Statistical Software*, **4**, 1

Cottrell, A. (2008) Gnu Regression, Econometrics and Time-series Library, URL:http://gretl.sourceforge.net/index.html

Grunsky, E.C., (2002) R: a data analysis and statistical programming environment–an emerging tool for the geosciences, *Computers & Geosciences*, **28** 10, pp 1219-1222.

Hornik, K and Leisch’ F, (2005) R Version 2.1.0, C*omputational Statistics*, **20** 2 pp 197-202

Kabacoff, R. (2008) Quick-R for SAS and SPSS users, available from http://www.statmethods.net/index.html

López-de-Lacalle, J, (2006) The R-computing language: Potential for Asian economists, *Journal of Asian Economics*, 17 6, pp 1066-1081

Meyer, D, Leisch, F., Hothorn, T. and Hornik, K., StatDataML: An XML Format for Statistical Data, URL: http://citeseer.ist.psu.edu/546737.html

Muenchen, R. (2007), R for SAS and SPSS users, URL: http://oit.utk.edu/scc/RforSAS&SPSSusers.pdf

Muenchen, R. (2007), Comparison of SAS and SPSS Products with R Packages and Functions, mailto:BobM@utk.edu

Murrel, P. (2005) R Graphics, Chapman & Hall

R Development Core Team (2007) R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0. URL: http://www.r-project.org/

Templ, M and Filzmoser, F (2008), Visualisation of Missing Values and Robust Imputation in Environmental Surveys, submitted for publication

Vito R. (2008) R Functions for Time Series, URL: http://cran.r-project.org/doc/contrib/Ricci-refcard-ts.pdf

Wheeler, D.A., (2007) Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers! URL: http://www.dwheeler.com/oss_fs_why.html