Commercial meets Open Source

Tuning STATISTICA with R

Christian H. Weiβ
∗ Introduction ∗
Extremely powerful environment for statistical computing!
▶ Provides packages for different areas (data mining, econometrics, biostatistics, etc.).
▶ Offers methods from different disciplines (time series analysis, statistical process control, bootstrapping, cluster analysis, etc.).
▶ Reflects state-of-art in statistical sciences.
▶ Freely available!
... on the other hand:
R is not particularly user-friendly!

- No graphical user interface, where whole repertoire of methods fully integrated.
- Methods not available for users, who have not learnt the R language.
- No powerful spreadsheet environment, which enables intuitive way of data manipulation.
⇒ Potential users from applied sciences and industry often do not have the heart to work with R!

Users often prefer the comfort of a commercial package like STATISTICA.
Idea:

Combine the power of R with the comfort of STATISTICA!
Idea:

Use STATISTICA as an **easily operated interface** with a respectable **basic equipment of statistical procedures**.

Integrate **specialised statistical procedures and sophisticated techniques** offered by R into the user interface of STATISTICA.
Idea:

The user does data analysis in STATISTICA, using the readily available methods, and using **macros** written in Visual Basic, which access R for advanced computations.

⇒ **Use power of R** without the need to learn the R language!
∗ Procedure ∗
Required:

- Base version of **STATISTICA** with its **Visual Basic** development environment;
- **R** together with necessary packages;
After having installed STATISTICA, R, R DCOM Server, ...

the remaining steps (programming & application) are done within the user-interface of STATISTICA.
How can we write a STATISTICA macro, which is able to access functionalities offered by R?
Function AufrufDialog As Boolean

AufrufDialog = True

Begin Dialog Benutzerdialog 350,287,"ARL Calculator".,DialogBehandlung ' %GRID:10,7,1,1
  DropListBox 10,7,230,70,auswahl(),.ListBox
  Text 10,35,220,49,"Design of CUSUM and EWMA charts, computations done with R package "spc"
  OKButton 250,14,90,21,.OkKnopf
  CancelButton 250,42,90,21,.AbbrechenKnopf
  Text 10,98,150,14,"Required information:".,TextAngabe
  Text 30,126,160,14,"Tolerance/Sigma:".,Text1
  Text 30,161,160,14,"Control limits/Sigma:".,Text2
  Text 30,196,160,14,"Shift/Sigma:".,Text3
  TextBox 220,119,110,21,.TextBox1
  TextBox 220,154,110,21,.TextBox2
  TextBox 220,189,110,21,.TextBox3
  Text 10,238,60,14,"Result:".,TextResultat
  Text 30,266,160,14,"ARL:".,Text4
  TextBox 220,259,110,21,.TextBox4

End Dialog

Dim dlg As Benutzerdialog
InitialisierBenutzerdialog (dlg)
Visual Basic environment allows to easily design user dialogs, ...
Compared to a "standard" macro, only **one additional step** is necessary:

Include

**R DCOM ("StatConnector") libraries.**
Sub Main

' Erstelle Serverobjekt
Set rzugriff = New StatConnector

' Fehlerbehandlung:
On Error GoTo fehler

' Starte R:
rzugriff.Init ("R")
rzugriff.EvaluateNoReturn ("library(spc)")
STATISTICA and R – Christian H. Weiβ
Afterwards, a new type of object is available: \texttt{StatConnector} object.

This object allows to communicate with R.
StatConnector-objects offer a number of methods:

Dim rzugriff As StatConnector
Set rzugriff = New StatConnector
Starting R:
 rzugriff.Init("R")

Receive possible error messages:
 rzugriff.GetErrorText

Shut down connection:
 rzugriff.Close
Most important methods:

```r
obj.Evaluate("R command")
obj.EvaluateNoReturn("R command")

obj.SetSymbol("R variable", value)
obj.GetSymbol("R variable")
```
Example 1
STATISTICA offers a number of approaches from SQC:
In particular, STATISTICA offers a broad variety of control charts, including, e.g., EWMA and CUSUM charts.
Innerer Durchmesser (in mm) von Kolbenringen für Fahrzeugmotoren. Quelle: Montgomery (2005), Kapitel 5.
Reliable design of EWMA and CUSUM charts is not possible with simple $k\cdot\sigma$ rule.

Instead: Consider ARL performance of charts.
However: STATISTICA does not allow to compute ARLs!


⇒ Tune STATISTICA with R!
Macro “ARLwithR.svb”:

Dim robj As StatConnector
Set robj = New StatConnector

Load spc-package:
robj.EvaluateNoReturn("library(spc)")
Compute ARL of EWMA chart:

```r
robj.Evaluate("xewma.arl(l=0.1, c=2.7, mu=0.0, sided="two", limits="vacl")")
```
Compute limits of EWMA chart:

```r
robj.Evaluate("xewma.crit(l=0.1,L0=370,
sided="two",
limits="vacl")")
```
※ Example 2 ※
STATISTICA and R – Christian H. Weiß

Image: A screenshot of a software interface titled "Time Series Analysis: EuStockMarkets.sta". The interface includes options for variables, with entries for DAX, SMI, CAC, and FTSE. The window also contains sections for Quick and Missing data, with options for ARIMA & autocorrelation functions, Seasonal decomposition (Census 1), Interrupted time series analysis, X11/Y2k (Census 2) - monthly, Exponential smoothing & forecasting, and X11/Y2k (Census 2) - quarterly. There are also options for Distributed lags analysis and Spectral (Fourier) analysis.

Description
Contains the daily closing prices of major European stock indices.

Single Series ARIMA: EuStockMarkets.sta

<table>
<thead>
<tr>
<th>Lock</th>
<th>Variable</th>
<th>Long variable (series) name</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>LogReturn</td>
<td>=Dif(v5:1)</td>
</tr>
</tbody>
</table>

Number of backups per variable (series): 3

Quick | Advanced | Autocorrelations | Review series

ARIMA model parameters
- Estimate constant: Yes
- Seasonal lag: 12
- p - Autoregressive: 2
- P - Seasonal: 1
- q - Moving average: 4
- Q - Seasonal: 3

Transform variable (series) prior to analysis
- Natural Log: Yes
- Power transform: 2.0
- Difference: 1. Log: 0
- N of passes: 0
- 2. Log: 0
- N of passes: 0

Plot of variable: LogReturn

Autocorrelation Function (EuStockMarkets.sta)
STATISTICA offers a large number of methods from time series analysis. E.g., it is able to fit any type of ARIMA model. However, STATISTICA not able to fit GARCH models!


⇒ Tune STATISTICA with R!
Macro “GARCHwithR.svb”:

Dim robj As StatConnector
Set robj = New StatConnector

Load tseries-package:
robj.EvaluateNoReturn("library(tseries)")
Submit data to R, assign it to R variable called "data":

robj.SetSymbol("data", spreadsht.Data)
Ask R to fit a GARCH(1,1) model:

```r
robjects::EvaluateNoReturn("data.garch<-garch(data, order=c(1,1))")
```
Ask R for ...

maximized log-likelihood:
robj.Evaluate("logLik(daten.garch)")

estimated coefficients:
robj.Evaluate("coef(daten.garch)")

estimated covariance matrix:
robj.Evaluate("vcov(daten.garch)")

...
estimated residuals:
robj.Evaluate("residuals(daten.garch)")

Use these results and prepare STATISTICA output:

Description
Contains the daily closing prices of major European stock indices:
Germany DAX (Ibis), Switzerland SMI, France CAC, and UK FTSE.
The data are sampled in business time, i.e., weekends and holidays are omitted.

Format
A multivariate time series with 1860 observations on 4 variables. The object is of class "nts".

Fitted GARCH(1,1) model: AIC 11922.7751257722, BIC 11939.3568935303.

<table>
<thead>
<tr>
<th>1</th>
<th>Estimate</th>
<th>2</th>
<th>Std. error</th>
<th>3</th>
<th>t value</th>
<th>4</th>
<th>p value (2-sided)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a0</td>
<td>4.639E-01</td>
<td>7.559E-07</td>
<td>6.136E+00</td>
<td>8.42E-10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a1</td>
<td>0.0683289</td>
<td>0.0112507</td>
<td>6.073E+00</td>
<td>1.253E-09</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b1</td>
<td>0.889067</td>
<td>0.0165202</td>
<td>53.817</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Latest Developments
Above approach for accessing R can be realized with any version of STATISTICA.

Only few days ago, the new release MR-3 for STATISTICA, version 8 occurred, see www.statsoft.com.

→ several new approaches for interacting with R!
Essentially, four main innovations:

- Run R scripts straight from STATISTICA.
- Call R scripts from STATISTICA macro.
- New commands for R scripts to simplify data transfer between R and STATISTICA.
- New commands for SVB macros to simplify data transfer between R and STATISTICA.
Run R scripts from STATISTICA:
→ Simply open file with extension .r or .s.
Then run script like usual SVB macro.
Output in workbook:

- A report (≈ RTF file) with console output.
- Graphs generated by `plot` as separate metafiles.
Extend these R scripts with the new commands offered by STATISTICA:

```r
library(tseries)
p<-1
q<-1
varnr<-6
n<-p+q+1
model<-paste("Fitted GARCH(".p,".".q." model",sep="")
dater<-ActiveDataSet[varnr]
plot(dater[,"data_type"])
daten.nrow<-nrow(daten)
daten.garch <- garch(daten, order = c(p,q))
daten.garch.logLik<-logLik(daten.garch)
AIC<-2*daten.garch.logLik + 2*n
#AIC<-sapply(daten.garch, AIC)
BIC<-2*daten.garch.logLik + log(daten.nrow-1)*n
RouteOutput(summary(daten.garch)[2:2][[1]],"Fitted Model",paste(model,"with AIC",AIC,"and BIC",RouteOutput(as.matrix(summary(daten.garch)[1:1][[1]]),"Estimated Residuals",paste("Estimated Re RouteOutput(summary(daten.garch)[5:5][[1]],"Jarque-Bera Test",paste("Jarque-Bera Test of",model RouteOutput(summary(daten.garch)[6:6][[1]],"Box-Ljung Test",paste("Box-Ljung Test of",model))
# RouteOutput(vcov(daten.garch), "Estimated Covariance Matrix",paste("Estimated Covariance Matrix
```
Important new commands for R scripts:

**ActiveDataSet** `[FromVar:ToVar]`  
**Spreadsheet** ("*path*")

→ Access STATISTICA data file.

**RouteOutput**(R *table*, *name*, *header*)

→ Transfer R tables to STATISTICA tables, display them separately in a workbook (optional: with name “*name*”, header “*header*”).
Call R script from SVB macro:

```r
# Werte abfragen:
p <- CLng(dlg.TextBoxa)  # Parameter p
q <- CLng(dlg.TextBoxb)  # Parameter q

# Fehlerbehandlung bzgl. Eingabe:
If p<0 Then
    MsgBox "p is not a"+vbCrLf+"non-negative integer value!".
    _
    vbCritical+vbokOnly, "Error"
    GoTo ende
End If

If q<0 Then
    MsgBox "q is not a"+vbCrLf+"non-negative integer value!".
    _
    vbCritical+vbokOnly, "Error"
    GoTo ende
End If

Dim oColl As New Collection
oColl.Add(tabelle, "daten")
oColl.Add(VarListe(1), "varnr")
oColl.Add(p, "ppar")
oColl.Add(q, "qpaa")

Dim oMacro As Macro
Set oMacro=Macros.Open("D:\Eigene Dateien\KategorialeZeitreihen\Vortraege\useR! 2008\Vortrag\GARCH.r")
oMacro.ExecuteWithArgument(oColl)
```
Dim oMacro As Macro
Set oMacro = Macros.Open("path")

Run macro by one of following approaches:
oMacro.Execute
oMacro.ExecuteWithArgument(oColl)
oMacro.executeNoRouteOutput(oColl)
oMacro::Execute

Just execute R macro, output controlled from R script, e.g., using command RouteOutput.
Like before, but submit initial values through newly created SVB Collection object:

```vba
Dim oColl As New Collection
oColl.Add(STCAobj, "name")
```

“name” ist variable’s name in R.
oMacro.executeNoRouteOutput(oColl)

Like before, but no immediate output to workbook. Instead:

Returns an object of newly created type StaDocCollection. Items of this object can be processed in SVB macro.
References
http://cran.r-project.org/contrib/extra/dcom/

http://cran.r-project.org/src/contrib/Descriptions/spc.html


http://cran.r-project.org/src/contrib/Descriptions/tseries.html

Thank You
for Your Interest!