FORTUNE — A System for Interactive Statistical Graphics for Time Series

Thomas Kötter[1], Martin Theus[2]

[1] Humboldt University of Berlin, 10178 Berlin, Germany
e-mail: thomas@wiwi.hu-berlin.de

[2] AT&T Labs Research, Florham Park, NJ
e-mail: theus@research.att.com

Summary

The classical representation of time series data is the time series plot. It is the common means to investigate the structure of data which were measured over time. As long as the representation remains static, this kind of display hardly supports an exploratory analysis of the series. Although various optimisation criteria, e.g. Cleveland's (1993) banking to 45 degree can be used to find the best representation of a graph, the gained insight into the series might still be limited.

Interactive features like dynamic scaling or overlaying of different time series and components of time series support an exploratory analysis of time series strongly.

This paper presents the FORTUNE software package. The main focus of FORTUNE is on forecasting and transformation. In addition to the basic interactive features for time series, new display and decomposition techniques have been implemented in FORTUNE.

The RPC-interface to XploRe4 makes FORTUNE extendable in an easy way.

Keywords: Exploratory Time Series Analysis, Interactive Graphics, Interactive Decomposition, Forecasting

1 Interactivity for Time Series

1.1 What we do not need

Several packages — commercial packages include DataDesk, JMP, SAS-Insight, research software includes MANET, Voyager and XLispStat (cf. Wilhelm, Unwin & Theus (1996)) — offer interactive statistical graphics. This kind of interactivity is mainly based on the paradigm of linked highlighting. Every point or subgroup which is selected, or more generally, to which a particular attribute was assigned to, reflects this assignment in each plot or output in the package. This supports an analysis of multidimensional data strongly.

1.2 What we need

Although linked highlighting could be applied in exploratory time series analysis, the kind of interactivity needed for time series is distinct from the one described above.

The three main interactive actions in time series analysis comprise:

- Dynamic scaling in t and in y direction

  If one looks for similarities in one time series or between different time series, it might be helpful to scale the series first, in order to make the series comparable. Since the amount of scaling may depend on the particular time interval, scaling operations should be dynamic.

- Overlaying of different time series

  - free
  - constrained to the time-axis

  Overlaying makes the comparison of time series much easier. A free overlaying of series is equivalent to the free movement of the series in the workspace. The overlaying of components of time series needs a constraint to the time axis, since corresponding points on the time axis of, e.g. the series and its trend estimate, should be aligned correctly. The general concept of constraint based plotting of series is called hierarchical plotting, and will be introduced in section 3.2.

  Even today scientists still overlay plots of time series printed on transparencies to find similarities, or at least ignore any interactive approach. (cf. the recent paper by Becker et al. (1994), suggesting the so called Cave Plotting)

- Interrogation of individual points

  Understanding outliers, breakpoints as well as trends and seasonal patterns always need a precise knowledge of the time intervals where patterns occur. Interactive implementations of time series plots can offer interrogations easily by just clicking with the mouse on a particular point of the series, which will show all information of interest concerning that point.

Although these three demands can be implemented in modern computer systems, nearly all statistical packages fail to offer these features in an interactive fashion.

2 Other Systems

There exist two other packages, offering dynamic and interactive methods to analyse time series, which are presented briefly in this chapter:

2.1 DIAMOND FAST

DIAMOND FAST (Unwin & Wills (1988)) is software for Macintosh computers, which is able to handle many time series simultaneously. The user can arrange the series on different pages. Each series can be moved to any place on its page, be scaled dynamically and interrogated. Thus DIAMOND FAST meets the three basic demands on interactivity of time series. Even missing values can be handled properly. Simple arithmetic transformations can be performed on the series, including differences and smoothing. Each point can be interrogated to get date and value of that point. Figure 1 shows a sample DIAMOND FAST desktop. The left window shows a part of the currently loaded series of unemployment figures of German Bundesländer. The window on the upper right allows an editing of the series data of the currently selected series. The lower right window is the index window, which gives an overview of the complete page.

2.2 zXQ

The zXQ software (McDougall & Cook (1994)) uses the XGobi (cf. Cook et al. (1995)) graphical interface routines. Thus zXQ only displays a single plot in each window. Whereas DIAMOND FAST was designed to handle many series simultaneously, zXQ offers different representations of a single series. The user can open a diagnostic window for an ARMA model containing the autocorrelation function (ACF) and partial autocorrelation function (PACF) of the series as well as a window showing the frequency domain of the series. The base window allows dynamic scaling and other on-line transformations. These changes to the series are propagated to all other windows depending on the base window. Figure 2 shows a sample zXQ session. The upper left window shows the first differences of the initially loaded time series. The right hand window contains the base dialog box offering all transformation options for the loaded series, while the lower left window displays the power spectrum for the series of the first differences of the series in dB-scale. Applying 12th differences successively to the time series would eliminate the peaks in the spectrum by a single mouse click instantaneously as the monthly measured data includes a seasonal cycle of length 12.

3 FORTUNE Functionality

The design of FORTUNE was mainly influenced by DIAMOND FAST. Whereas DIAMOND FAST was developed for Macintosh computers only, FORTUNE, as well as zXQ, was developed for UNIX-workstations running an X11 window manager.

3.1 I/O-Functions

FORTUNE supports two different data formats. An ASCII format, which is also used to import series, and a FORTUNE specific binary format. After importing the data of the series, a dialog box prompts for various meta information on the series. In Figure 3 the questionnaire to query the meta information of a series is shown. All procedures inside FORTUNE can access this information. For an imported series the comment tag is automatically filled with the filename, number of observations and the extreme values of the series. It can additionally hold any arbitrary information provided by the user. Whenever a new series is

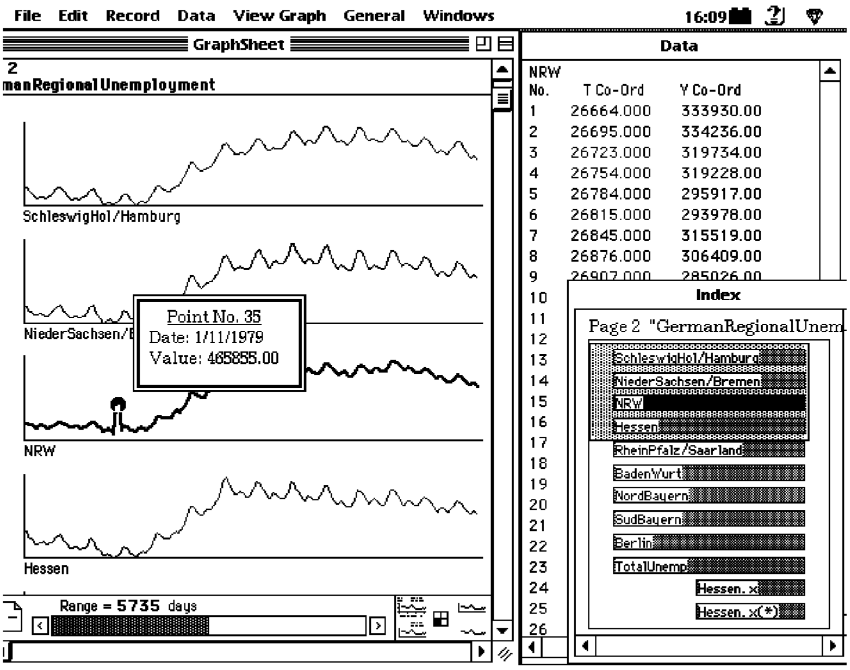

Figure 1: A sample DIAMOND FAST desktop

Figure 2: A screen shot of a zXQ session

generated from an existing series, these information are inherited and updated according to the generation. Besides importing a single series a whole session, i.e. a collection of series and derived series can be saved and reloaded.

3.2 The Workspace

FORTUNE uses a single page as its workspace. Unlike DIAMOND FAST, where each page has a fixed size, the workspace in FORTUNE is — theoretically — of infinite size, i.e. the user can drag a series to any place. A typical scenario is shown in Figure 4. Basic interactions with the series, which are invoked by a right mouse click on the series, include:

- Moving a series
- Cut-/Copying out a subseries
- Deletion of series

An interrogation of any point can be achieved by additionally pressing the shift key.

Additionally a zoom function allows a scaling of series to any size and aspect ratio, which always applies to all series in the workspace. Whereas it is possible to scale a time series in y-direction dynamically, there is no possibility to change the scale of an individual series along the time-axis. This would be a contradiction to the tidy-up concept inside FORTUNE, since all tidy-up functions adjust all series to a common time axis. Simple tidy-up arranges all series in a list, according to their y-position before the tidy-up function was invoked. Since the scale in y may be very different — e.g. a comparison of different components of a trend-season-decomposition — it is possible to choose the same plot size for all series, which overrides the common $y-$scale.

In many situations it may be of interest that tidy-up operations are only applied to certain time series. Therefore the user can choose the special option "group by selection". While this option applies, all tidy up operations are only performed on the currently selected series. This option may be mixed with the hierarchical plotting option, which is the most important and extremely efficient tidy-up operation:

Hierarchical Plotting of Time Series

Hierarchical plotting of time series arises in the context of time series decomposition, breakpoint detection and forecasting. All these components are used to describe the structure of a series. When hierarchical tidy up is chosen, all estimates for trend components and breakpoints are plotted onto the series, they were calculated from. The same applies for forecasts. Forecasts are plotted at the end of the time axis, and are adjusted along the $y-$axis, to fit exactly to the series they have been generated from. Figure 4 and 5 show examples for hierarchical plotting. Onto the base series of Figure 4 a trend estimate and a prediction have been plotted, whereas the estimate for the seasonal component is plotted below the series.
Again it is possible to tidy up only those series, which have been selected before, facilitating the user to superpose different trend estimates or forecasts for comparison.

3.3 Breakpoints

Detecting breakpoints in time series is of great interest. Although a subjective assignment of breakpoints could offer sensible results, a support by numerical routines is usually very helpful. Three methods for determing breakpoints are included:

- Manually, supported by domain knowledge. E.g. the German reunification in 1989 will affect many time series, but for each series an exact breakpoint can be specified, depending on the reunification of the underlying sector.
- Lombard's (1983) method based on rank statistics,
- Carlstein's (1988) nonparametric estimator, which is based on the Kolmogorov-Smirnov-test.

Figure 5 shows how breakpoints are plotted. The output of a breakpoint routine is an ordinary binary time series which has the value *Min(Series)* if no breakpoint was detected, and *Max(Series)*, for each breakpoint. These series are plotted as needle plots, i.e. each value is connected with the x-axis by a perpendicular line. Using the hierarchical plotting option delivers a plot as shown in Figure 5.

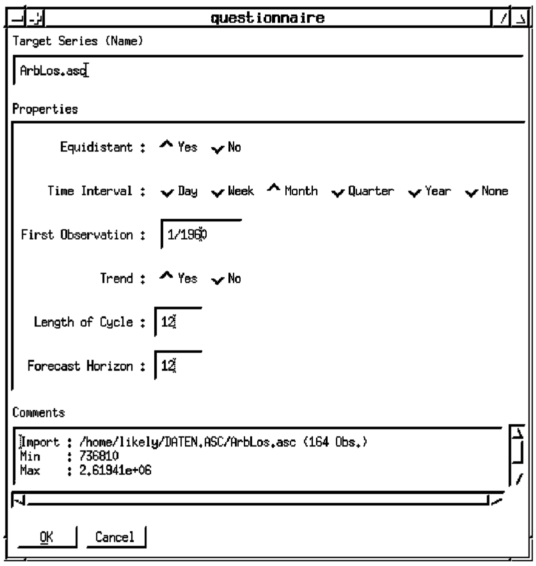

Figure 3: The questionnaire to query meta information of the series

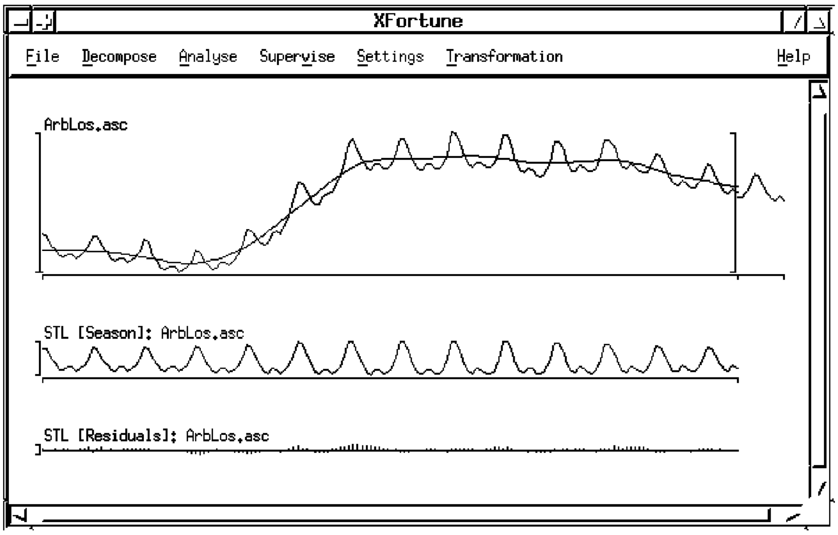

Figure 4: A FORTUNE scenario with hierarchical plotting selected

3.4 Decompositions

Many time series are dominated by a trend component as well as a seasonal component. A lot of techniques have been proposed for decomposition. An often used technique is the **X11**(-ARIMA) method (cf. Scott (1992)), which is as popular as it is hard to understand.

A modern competitor to X11 is the **STL** routine proposed by Cleveland et al. (1990). STL makes extensive use of the **lowess** smoother (cf. Chambers et al. (1983)). Both techniques, STL and lowess, are implemented in FORTUNE. Since the user must specify all meta information of a series, all STL parameters can be set automatically, following Cleveland's proposals. Thus the user does not need to supply any further parameters to start a decomposition. STL's optional robust loop is always performed.

Edmundson (1990) proposed a judgemental decomposition method, where the user specifies a trend component manually by a polygon. The difference between the specified trend and the series is a mixture of the seasonal component and the irregular component, which is plotted seasonwise into one coordinate system. Thus all seasons overplot. Now the user may specify another polygon for the shape of the seasons, which is assumed to be constant over time.

Whenever the user specifies a knot of a component, the dependent components are updated immediately. Enabling a setting, moving and deleting of knots enables the user to work iteratively, until he is satisfied by the chosen decomposition.

Edmundson's decomposition including trend and seasons with unlimited number of knots, was first ever implemented inside FORTUNE.

From these components a forecast is generated by simple extrapolation.

3.5 Forecasting

One possibility of generating forecasts is to use trend-season-decompositions. Other possibility are methods especially designed for forecasting. FORTUNE offers two forecasting procedures:

- Parzen's ARARMA (cf. Parzen (1982))
- Exponential Smoothing (cf. Schlittgen & Streitberg (1987) pp 380–387)

  Within the exponential smoothing the

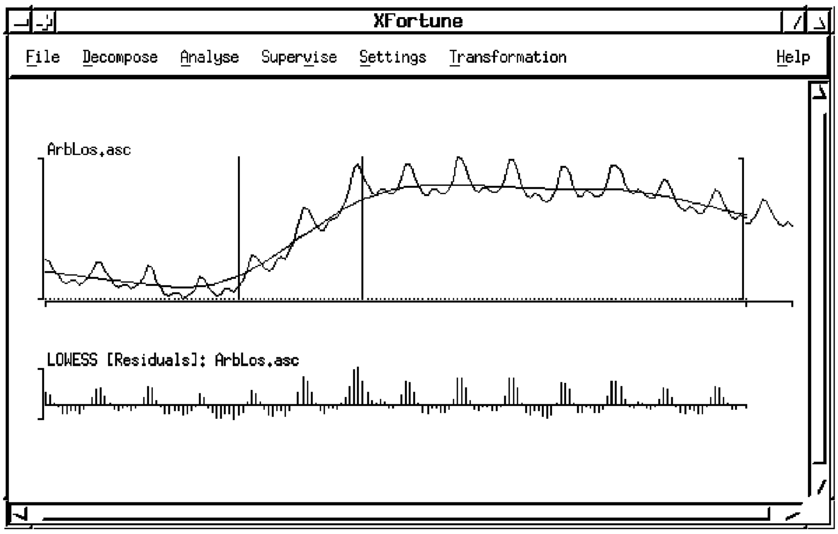

Figure 5: Two changepoints, a trend estimate and a forecast, superposed onto the base time series

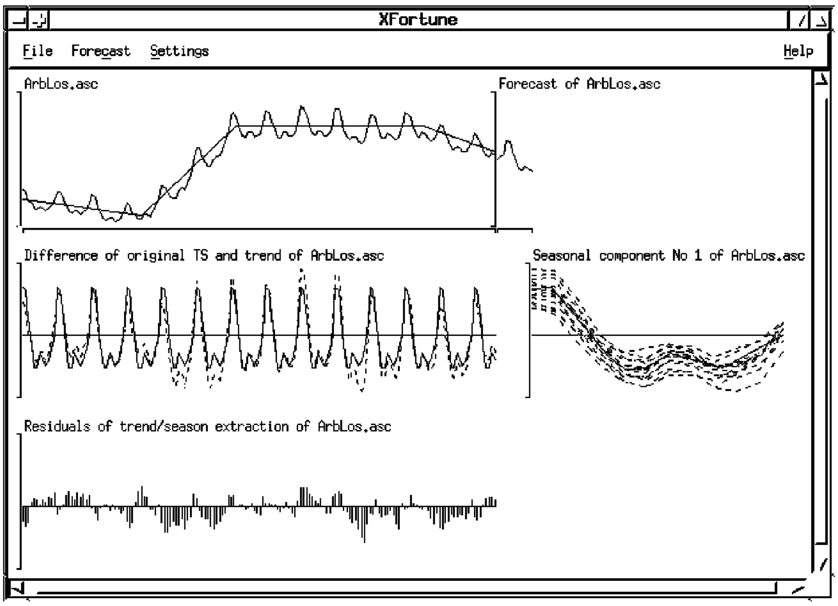

Figure 6: The interactive implementation of Edmundson's decomposition.

– Simple

– Holt (1957), Winters (1960)

approaches are both available for additive and multiplicative models.

Besides these forecasting procedures, it is also possible to specify forecast values manually. These values may simply be entered by the user, and can then be used as any other series in this session.

3.6 Transformations

Decompositions and forecasts are methods for generating new series. Another is given by the transformation tool. The transformation tool provides the following calculations on time series: Simple arithmetic calculations ($y' := a + b \cdot y^c$, $a, b, c \in \mathbb{R}$), $n-$th differences to lag l, Box-Cox transformation (the default parameter θ is chosen to satisfy approximately $\sigma_t = c \cdot \mu_t^{1-\theta}, \quad \forall t \in T$ and $c, \theta \in \mathbb{R}$), standardization to mean $\mu = 0$ and variance $\sigma^2 = 1$, spectrum (incl. various smoothing windows), autocorrelation, partial autocorrelation and binary operations $(+, -, *, /)$ on two series. In Figure 7 a sample application of a simple binary operation on two series is shown. The upper two series show female, resp. male unemployment figures. Both series show two peaks per season; the winter peak due to lower activities in building-trade, and a summer peak due to the drop outs, and graduates from highschool. While the ratio eliminates the summer peak, which is obviously not affected by gender, the winter peak gets more significant. In addition it can be seen, that rising unemloyment first affects men.

4 The RPC-Interface to XploRe4

Extendability and interactivity in graphical software seem contradictory. FORTUNE uses a tradeoff to solve this dilemma. Whereas it is not possible to extend the interactive functions in FORTUNE, the user can incorporate new numeric functions, e.g. decompositions or forecasts, written in XploRe4. This is done by using XploRe4 as a RPC-server. Figure 8 shows the overall structure of an RPC-call. To achieve a flexible interface to XploRe4, FORTUNE offers a menu item, which invokes a dialog box, that accepts any XploRe4 command. After the command is issued, the RPC request starts. The *RPC-send* sends the command, as well

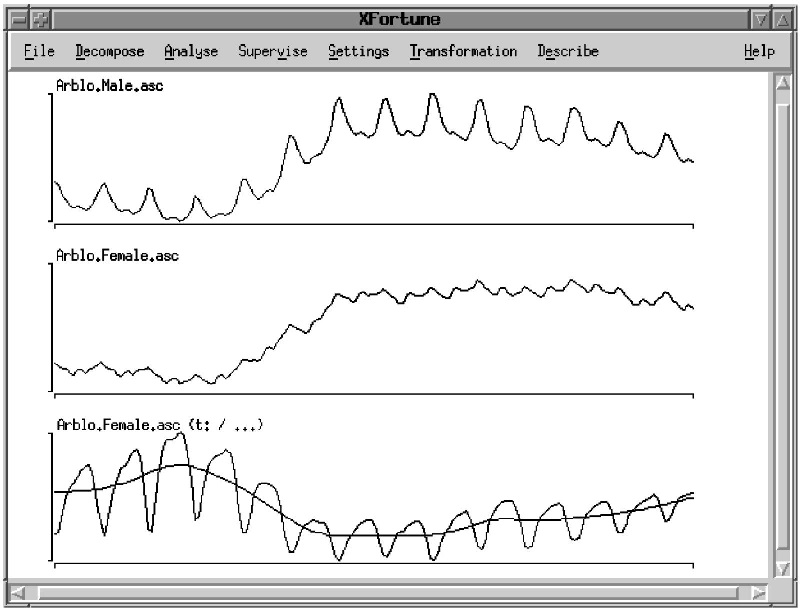

Figure 7: Calculating the ratio of female and male unemployment figures.

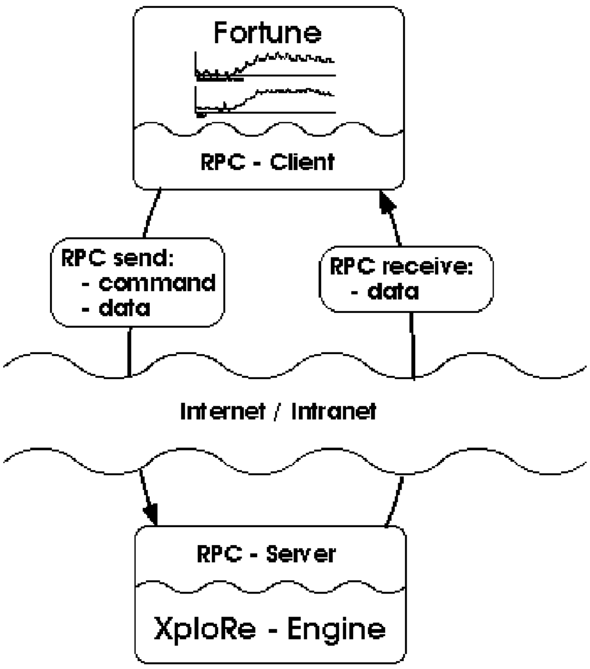

Figure 8: The RPC-interface between FORTUNE and XploRe4.

as the currently selected time series to the XploRe4 RPC-server. The timeaxis and the data of the selected series are available in the XploRe4 environment as the variables `t` resp. `x`. Although the transformation calculator already provides an FFT, the corresponding XploRe4 commands are:

```
 data = x ~ 0 * x
trans = fft(data)
    x = sqrt( sum( trans^2, 2))
    t = 1:rows(x) / rows(x)/2
```

If the commands are executed successfully, the series in `t` and `x` are returned to FORTUNE and then can be used as any other series in the environment. Besides transformations and decompositions of already existing series, the interface can be ideally used as a generator for patterned and random series.

Since it is not likely for every FORTUNE user to have a copy of XploRe4, the default XploRe4-server that is connected to, is located at the Humboldt University in Berlin.

5 Conclusion

FORTUNE proves to be a very valuable tool for the interactive exploration of time series. It enables the user to quickly generate components, forecasts etc. Interrogations, overlaying and hierarchical plotting allow an easy comparison and judgement of the achieved results, which reach far beyond the possibilities of static systems. The RPC-interface to XploRe4 extends the functionality of FORTUNE. Thus it is easily possible to make use of user specific routines.

FORTUNE is freeware, and can be downloaded at http://www1.math.uni-augsburg.de/~theus/Fortune.html, or http://www.wiwi.hu-berlin.de/~koetter/
Fortune.html, which also provide additional information concerning the packages. Currently binaries for Solaris 1, Solaris 2 and PC-Linux are available for downloading.

Additional information on XploRe4 can be found at http://www.wiwi.hu-berlin.de.

6 Acknowledgement

The FORTUNE software was developed under the framework of the EUROSTAT LIKELY-project which was part of the DOSES-project. It was coordinated by Mike Talbot of the Scottish Agricultural Statistics Service and was divided into two parts:

- Cognitive Maps

  The cognitive map part again was divided into a group dealing with the methodology of graphical models, headed by Alex Gammerman and Colin Aitken and a group which designed the software tools for handling cognitive maps, headed by Antony R. Unwin, who supervised this group at Trinity College Dublin.

- Time Series Models

  The time series part was developed by a group at the statistics department of the University of Dortmund, which was headed by Friedhelm Eicker. The authors have been part of this group, and would like to thank Axel Benner, Gregor Rieken, Jürgen Symanzik and Andreas Prenneis for their most valuable contributions to the FORTUNE Software.

Since the Dortmund group no longer exists, the further development of the FORTUNE software stalled at least for the near future.

References

Becker, R.A., Clark, L.A., Lambert, D., (1994), Cave Plots: A Graphical Technique for Comparing Time Series, *Journal of Computational and Graphical Statistics*, Vol. 3, No. 3, pp 277–283.

Carlstein, E. (1988), Nonparametric change-point estimation, *Annals of Statistics*, Vol. 16, pp. 188–197.

Chambers, John M., Cleveland, William S., Kleiner Beat, Tukey, Paul A. (1983). *Graphical Methods for Data Analysis* Wadsworth, Belmont, California.

Cleveland, Robert B., Cleveland, William S., McRae, Jean E., Terpenning, Irma (1990). STL: A seasonal-trend decomposition procedure based on loess (incl. discussion). Journal of Official Statistics (Sweden), Vol. 6, pp 3–73.

Cleveland, William S. (1993). *Visualizing Data.* Hobart Press, Summit NJ.

Cook, D., Buja, A., Cabrera, J. & Hurley, C. (1995). Grand Tour and Projection Pursuit. *Journal of Computational and Graphical Statistics.* Vol. 4 No. 3, 155–171.

Edmundson, R.H., (1990). Decomposition: A strategy for judgemental forecasting. *Journal of Forecasting.* Vol. 9, pp 305–314.

Härdle, W., Klinke, S. & Turlach, B.A. (1995), *XploRe: An Interactive Statistical Computing Environment.* Springer, New York.

Holt, C. C. (1957). Forecasting seasonals and trends by exponentially weighted moving averages. *Carnegie Institute of Technology*, Pittsburgh, Pennsylvania.

Lombard, F. (1983). Asymptotic distributions of rank statistics in the change-point problem. *South African Statistical Journal*, Vol. 17, pp 83–105

McDougall, A.J. & Cook, D.H. (1994). *A User's Guide to XQz, (LaTeX version with Figures)*, Tech. Report 94–007, Dept. of Statistics, Rutgers University.

Parzen, Emanuel (1982), ARARMA models for time series analysis and forecasting, *Journal of Forecasting*, pp 67– 82.

Schlittgen, Rainer, Streitberg, Bernd H.J. (1987) *Zeitreihenanalyse*, 2. Auflage, Oldenbourg, München.

Scott, Stuart (1992). An extended review of the X11-ARIMA seasonal adjustment package. *International Journal of Forecasting*, Vol. 8, pp 627–633.

Unwin, Antony R., & Wills, G. (1988). *Eyeballing Time Series.* In Proceedings of the 1988 ASA Statistical Computing Section, pp 263–268.

Wilhelm, A. F. X., Unwin A. R. & Theus, M. (1996) Software for Interactive Statistical Graphics — A Review, In: *Advances in Statistical Software 5, Softstat 95*, ed. F. Faulbaum, Fischer, Stuttgart, pp 3–12.

Winters, P.R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Sciences*, Vol. 6, pp 324–342.