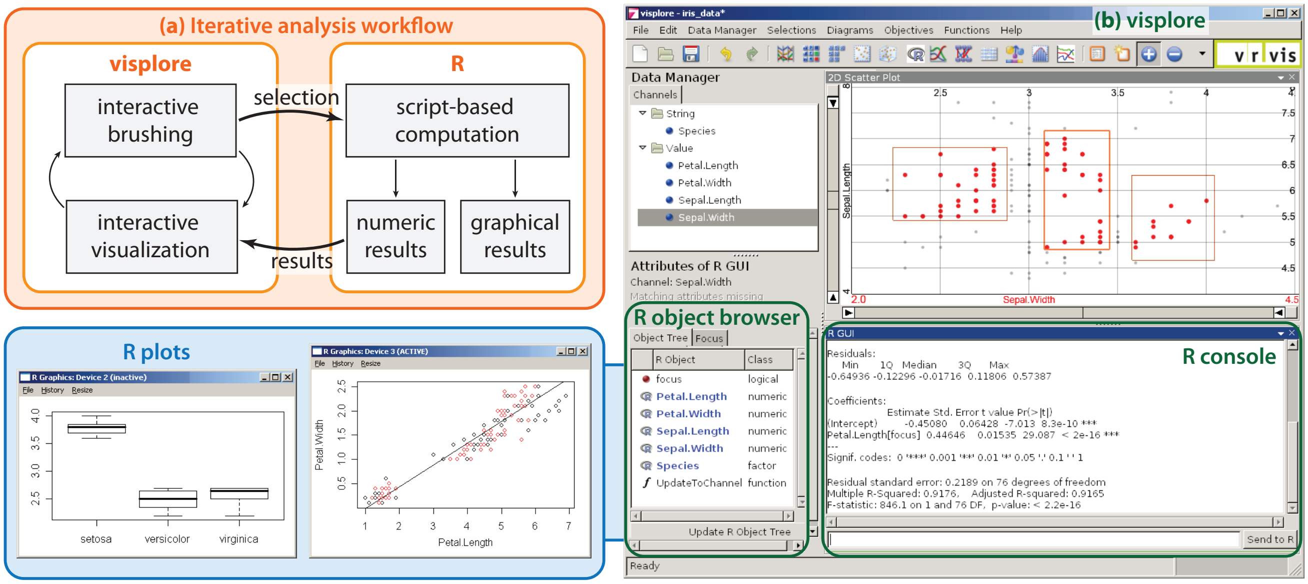
A Generic Model for the Integration of Interactive Visualization and Statistical Computing Using R

Johannes Kehrer[*]
VRVis Research Center,
Vienna, Austria

Roland N. Boubela[†]
Dept. of Statistics and Probability Theory,
Vienna University of Technology, Austria

Peter Filzmoser[‡]
Dept. of Statistics and Probability Theory,
Vienna University of Technology, Austria

Harald Piringer[§]
VRVis Research Center,
Vienna, Austria



Attributes of R GUI
Channel: Sepal.Width
Matching attributes missing

Object Tree | Focus

	R Object	Class
●	focus	logical
	Petal.Length	numeric
	Petal.Width	numeric
	Sepal.Length	numeric
	Sepal.Width	numeric
	Species	factor
f	UpdateToChannel	function

Update R Object Tree

```
Residuals:
    Min      1Q  Median      3Q     Max
-0.64936 -0.12296 -0.01716  0.11806  0.57387

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        -0.45080    0.06428  -7.013  8.3e-10 ***
Petal.Length[focus]  0.44646    0.01535  29.087  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2189 on 76 degrees of freedom
Multiple R-Squared: 0.9176,    Adjusted R-squared: 0.9165
F-statistic: 846.1 on 1 and 76 DF,  p-value: < 2.2e-16
```

Figure 1: (a) The integration of visplore and R enables an iterative analysis workflow. (b) The integrated *R object browser* shows all objects in the R workspace and allows synchronization between both environments. R commands and scripts can then be written using the *R console*.

Abstract

This poster describes general concepts of integrating the statistical computation package R into a coordinated multiple views framework. The integration is based on a cyclic analysis workflow. In this model, interactive selections are a key aspect to trigger and control computations in R. Dynamic updates of data columns are a generic mechanism to transfer computational results back to the interactive visualization. Further aspects include the integration of the R console and an R object browser as views in our system. We illustrate our approach by means of an interactive modeling process.

1 Introduction and Motivation

Visualization and statistics both facilitate the understanding of complex data characteristics. Traditional statistical tools use static visualizations mainly for presentation purposes (confirmatory analysis). Visual analysis, in contrast, combines computational means with powerful interaction concepts such as linking and brushing. The statistical functionality, however, typically has to be implemented either from scratch or by adapting open-source algorithms. In contrast, this poster demonstrates concepts to tightly integrate statistical computing based on the environment *R* [4] in an existing framework for visual analysis, called *visplore* [3]. Such an integration provides on-demand access to a vast amount of statistical methods and graphics, including recent developments in computational analysis. Benefits of the integration include rapid prototyping of semiautomated analytical approaches as well as on-demand data transformations (e.g., normalizing the data, applying a box-cox transformation, computing robust statistical moments).

Currently, only few approaches tightly integrate statistical software and interactive visualization in a generic way. The visual analysis framework Mondrian [5], for example, can load data from an R workspace using a GUI interface. Other approaches integrate linking and brushing facilities into R graphics [6]. As a third alternative, two stand-alone environments can be coupled. The R package *rggobi*, for instance, links R and GGobi [1] in a way that results from both applications can be combined. The analysis is mainly steered from R, for example, by creating GGobi plots or animations via the command-line. Our work is inspired by the latter kind of approaches and enables a highly interactive loop between visualization and computing, which is mainly controlled via brushing. In contrast to other systems, visplore provides rich visual feedback during interaction by showing intermediate results and by using multi-threading [3]. The proposed concepts are generic and can potentially be applied to other visual analysis frameworks as well.

2 Integrating R into visplore

This section describes the concepts for our integration of R into our framework of coordinated multiple views. A key mechanism is the dynamic exchange of both selection information and data attributes between the two environments. We can directly access internal memory structures in R using its API [4], which enables a fast

[*] e-mail: kehrer@vrvis.at
[†] e-mail: roland.boubela@tuwien.ac.at
[‡] e-mail: p.filzmoser@tuwien.ac.at
[§] e-mail: hp@vrvis.at

Published in Proc. IEEE Conf. Visual Analytics Science and Technology (VAST 2012), pp. 133-134, 2012.

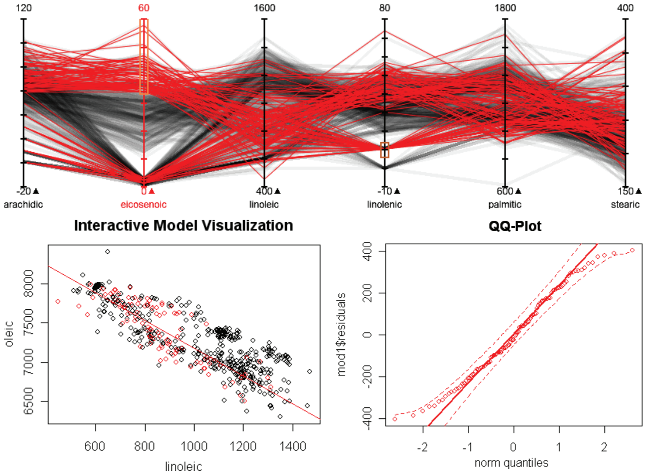


Figure 2: Interactive creation and evaluation of a regression model.

data transfer. Data columns in visplore can then be made available for R as vectors, where they can be accessed, e.g., within scripts. Selections in visplore result from brushing in linked views and are represented as logical vectors in R. The user can choose whether the selection information should be synchronized manually or automatically, e.g., when altering a brush.

The described data exchange enables an iterative analysis workflow, which is illustrated in Fig. 1a. The user brushes interesting subsets of the data in views such as scatterplots or parallel coordinates. R scripts can then be executed on-demand or automatically whenever the selection information changes. Using R's indexing feature, computations can optionally be restricted to the selected subset of the data, e.g., `subset = dataset[focus]`. Alternatively, the index vector can be used as a parameter to an R function. Instead of computing statistics of the whole data, the user can then study local summaries or statistical models created only from the brushed subset of the data (e.g., after deselectiong outliers). The computed results can be automatically transferred back to visplore, where they can be visualized and explored further like any other data attribute. Also, the data can be analyzed using statistical graphics within R, which are dynamically updated when the selection changes.

For convenience reasons, we incorporate the standard *R console* as a view in visplore, which allows the user to write R commands and execute them using the R language interpreter (e.g., applying data transformations). Additionally, an integrated *R object browser* shows the existing objects in the R workspace (including variables and results from computations). This browser also enables the creation or synchronization with corresponding data columns in visplore. Operations such as deleting, editing or renaming of R objects can be done via a context menu.

3 Interactive Creation of Regression Models

We demonstrate the advantages of combining interactive visualization and statistical computing via R. The analyzed data stems from a survey on the classification of Italian olive oils based on their composition of different fatty acids [2].

The analysis starts with a setup that shows the different fatty acids in parallel coordinates (Fig. 2). We write an R script that creates a linear regression model (`lm`) based on the selected data subset (`focus`) of the fatty acids oleic and linoleic. The two data attributes are shown in an R scatterplot depicting the trend line for the selected data (red). Additionally, we check whether the residuals of the regression model are distributed normally using a *QQ plot* [7]. The R script is executed whenever the selection in visplore changes, which updates the regression model and the R graphics:

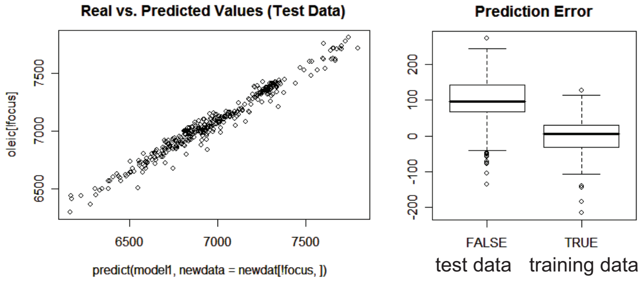


Figure 3: Visual analysis of a multivariate regression model, which is created from the selected data (training data) and compared to the real values of the unselected data (test data).

```
dev.off()
model1 <- lm(oleic ~ linoleic, subset = focus)
plot(oleic ~ linoleic, col = focus + 1)
abline(model1, col="red")   # trend line of model
x11()
qq.plot(model1$residuals, "norm", main="QQ-Plot")
```

As a next step, a multivariate regression model is created based on the brushed subset (training data) of four fatty acids. In Fig. 3, the predictions of the created model are then compared to the real values of the regressor, which are not selected (test data). Box plots are used to compare the prediction error on the test data with the error on the training data. The discrepancy between the two plots indicates a model fit with low prediction quality. Finally, summary statistics such as the residuals of the fitted model are transferred back to visplore, where they are investigated via brushing.

4 Conclusions

Adding the computational features of R to a powerful visualization framework like visplore creates a comprehensive toolbox for data analysts. The interaction loop between visplore and R enables the user to not only explore the data, but also to interactively create and evaluate statistical models, combining dynamic graphics with R modeling tools. Since visplore uses the API to directly access R's internal memory structures, the integration scales to millions of data values. During interaction, only the selection information and the resulting attributes need to be synchronized. When R is busy due to large data or complex computations, multi-threading ensures that visplore remains responsive [3]. In future work, we want to tighter integrate R graphics into the visplore system. New views could visualize typical R objects like regression or classification models in order to further enhance the interactive modeling process. We consider our proposed model generic enough to be applicable to other visual analysis frameworks as well.

References

[1] D. Cook and D. F. Swayne. *Interactive and Dynamic Graphics for Data Analysis With R and GGobi*. Springer, 2007.

[2] M. Forina, C. Armanio, S. Lanteri, and E. Tiscornia. Classification of olive oils from their fatty acid composition. In *Food Research and Data Analysis*, pages 189–214, 1983.

[3] H. Piringer, C. Tominski, P. Muigg, and W. Berger. A multi-threading architecture to support interactive visual exploration. *IEEE Trans. Visualization and Computer Graphics*, 15(6):1113–1120, 2009.

[4] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, 2011.

[5] M. Theus and S. Urbanek. *Interactive Graphics for Data Analysis: Principles and Examples*. Chapman & Hall, 2008.

[6] S. Urbanek. iPlots eXtreme: Next-generation interactive graphics design and implementation of modern interactive graphics. *Comput. Stat.*, 26(3):381–393, 2011.

[7] M. B. Wilk and R. Gnanadesikan. Probability plotting methods for the analysis of data. *Biometrika*, 55(1):1–17, 1968.