Manuel J. A. Eugster & Friedrich Leisch

Exploratory Analysis of Benchmark Experiments – An Interactive Approach

Technical Report Number 056, 2009
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Exploratory Analysis of Benchmark Experiments – An Interactive Approach

Manuel J. A. Eugster and Friedrich Leisch

The analysis of benchmark experiments consists in a large part of exploratory methods, especially visualizations. In Eugster et al. [2008] we presented a comprehensive toolbox including the *bench plot*. This plot visualizes the behavior of the algorithms on the individual drawn learning and test samples according to specific performance measures. In this paper we show an interactive version of the *bench plot* can easily uncover details and relations unseen with the static version.

1 Introduction

In statistical learning, benchmark experiments are empirical experiments with the aim of comparing and ranking algorithms with respect to certain performance measures. New benchmark experiments are published on almost a daily basis; it is the primary method of choice to evaluate new learning algorithms in most research fields with applications related to learning algorithms. In Hothorn et al. [2005] the authors lay down a general framework for benchmark experiments. They use the bootstrap as sampling scheme such that the resulting performance observations are independent and identically distributed (iid) and can be analyzed using standard statistical methods. Using the foundations specified by their general framework, we introduced a toolbox of exploratory and inferential methods for the analysis of such benchmark experiments in Eugster and Leisch [2008] and Eugster et al. [2008]. The toolbox provides a range of adapted basic plots and a newly developed specialized plot, the *bench plot*, which absolutely provide insights into a benchmark experiment – but first uses quickly showed the requirement of interactivity during the exploratory analysis.

Interactive data visualization has a long tradition within statistics and data analysis. Since the 1960s visualization systems for exploratory data analysis with interactivity are developed; see, e.g., Cook and Swayne [2007] for the history of statistical data visualization. Closely related to the visualization of benchmark experiments is *Exploratory Model Analysis* (EMA), which refers to methods and procedures for exploring the space of models. Unwin and Volinsky [2003] introduce an interactive approach using parallel coordinates for different types of useful plots; and Wickham [2007] extends this approach by describing different levels of data, computed after fitting the models.

This paper tries to build a bridge between the process of creating a set of models (the benchmark experiment) and further analyses of models using methods of EMA or further analyses of the data set using methods of EDA. In Section 2 we introduce the benchmark experiment space by describing the different components a benchmark experiment accumulates and how they are linked together. To illustrate our ideas an example is introduced which is used through the article. Section 3 shows how we can use this linkage to create a interactive analysis environment. We shortly describe the software we are using and mainly introduce an interactive version of the bench plot as central point for the "navigation through" the benchmark experiment space. Section 4 then illustrates the usage of the interactivity by answering exemplary questions which arise during the analysis of the exemplar benchmark experiments. The figures in this example have ben left as they appear on screen, i.e. there is no labeling of a plot as any such information can be obtained by directly querying the graphic. Anyway, it is hard to show the full powerfulness of interactivity on paper, but all things to have a go by one self are available from `http://www.statistik.lmue.de/~eugster/`. Section 5 concludes the article with an outlook for further developments.

2 Benchmark experiment space

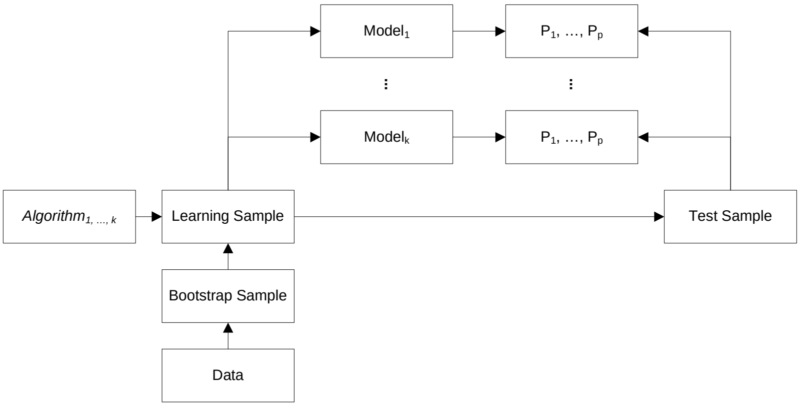


Figure 1: The relation of the components of the benchmark experiment space. The interactive visualization environment then enables the analysis of them for further analyses using methods of exploratory data (EDA) or model (EMA) analysis.

A benchmark experiment is described by the following components: (1) k candidate algorithms in which we are interested; (2) m data sets where we want to compare the algorithms on, in this paper it is assumed that $m = 1$; (3) p performance measures for the evaluation of the algorithms on the data sets; (4) the resampling method to draw B learning samples from each data set, i.e, sampling with replacement in this paper; (5)

and, in combination with the performance measures, appropriate methods to draw test samples, e.g., the out-of-bootstrap samples. The execution of the defined benchmark experiment leads to $k \times m \times B$ models, one for each combination of data set, drawn learning sample and algorithm. Then each of these models is evaluated according to the p performance measures. Thus, the benchmark experiment space we want to explore consists of data set, the bootstrap samples, the models and the performance measures. Figure 1 illustrates the relation of these components.

For the exploratory analysis there is one or more visual representation of each of these components and interaction concepts like selection, linkage and brushing can be used to highlight the relations.

2.1 Exemplar benchmark experiment

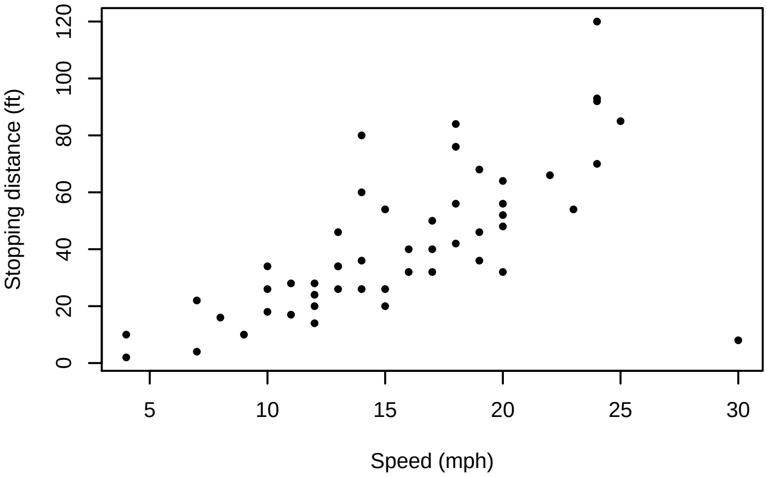


Figure 2: The *cars* data set [McNeil, 1977], which give the speed of cars and the distances taken to stop recorded in the 1920s. The original data set is extended with the data of a currently common car: 30 mph and 8 ft stopping distance.

The benchmark experiment we use for illustration is a regression problem and constructed as follows: The data set is *cars* [McNeil, 1977], which give the speed of cars and the distances taken to stop recorded in the 1920s. To make our demonstration more interesting, we accomplished an own experiment and recorded the speed and stopping distance of a currently common car: 30 mph and 8 ft stopping distance; see Figure 2 for the full data set. The candidate algorithms used (with corresponding `R` functions in parenthesis) are linear regression (`lm`), robust linear regression (`rlm`), cubic smoothing spline (`smooth.spline`, abbreviation in plots is `spl`) and local polynomial regression (`loess`) [all, e.g., in Venables and Ripley, 2002]. The performance measures are the prediction mean squared error and the computation time of the model fitting process. $B = 100$ bootstrap samples are drawn as learning samples with the corresponding out-of-bootstrap samples as test samples.

3 Interactive environment

The base software we rely on is the `R` statistical environment [R Development Core Team, 2008] with the `iPlots` package [Urbanek and Wichtrey, 2008], which provides high interaction statistical graphics directly accessible within `R`. This enables a mixture of command-line driven and interactive analysis which is highly useful for the analysis of benchmark experiments, as we will see later on. `iPlots` offers a wide variety of plots which all support interactive concepts, such as querying and linked highlighting. In this paper we explain only the concepts we need, for a full introduction we refer to Urbanek and Theus [2003]. The `icp` package[1] [Gouberman and Urbanek, 2008] extends `iPlots` and allows the creation of new, fully interactive plots within pure `R` code. We use this powerful environment and developed an interactive version of the bench plot as central point for the analysis of benchmark experiments and starting point for further analyses using EDA methods on the data sets and EMA methods on the models.

3.1 Benchmark experiment plot

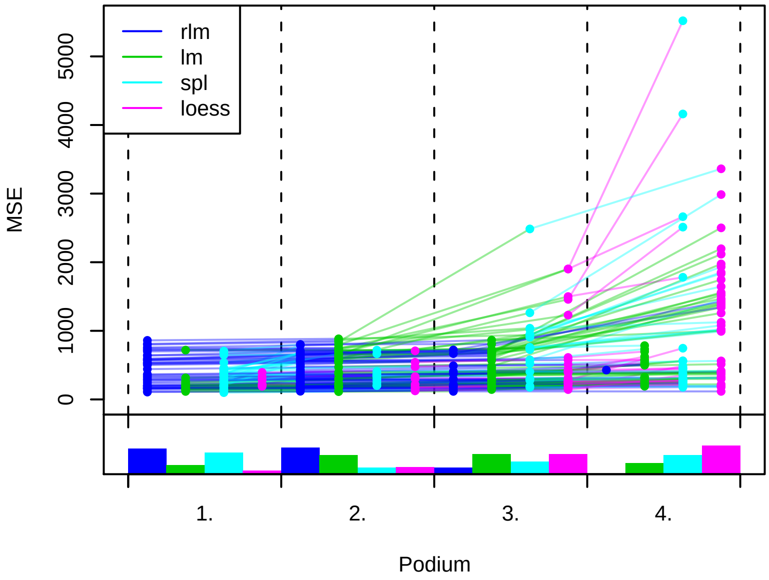


Figure 3: Static version of the bench plot preserved by the analysis of the exemplar benchmark experiment; it shows the predication mean squared error.

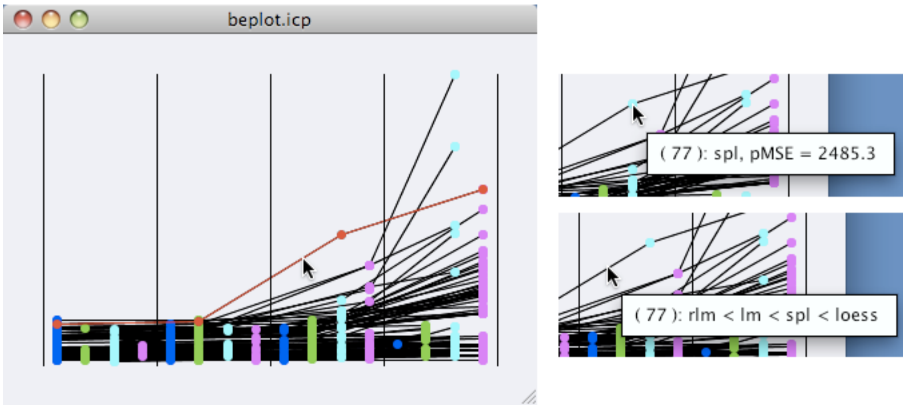
The idea of the bench plot is to show the behavior of the algorithms on the individual drawn samples according to specific performance measure; it is a visual representation of one performance measure and the bootstrap samples of the benchmark experiment space. For the motivation and a comprehensive description we refer to Eugster and Leisch

[1] We extended the package with new functionality which is not yet integrated into the official version, but available from the website mentioned in the introducing section.

[2008]. To present an example, Figure 3 shows a static version created for our exemplar benchmark experiment. The x-axis of the plot is a podium with as many places as there are algorithms, the y-axis is the performance measure. For each bootstrap sample the algorithms are ordered according to the performance measure, whereby ties are broken randomly. Then, for each rank a dot plot on the corresponding podium place is drawn and, as there dependencies between the "dots", we connect dots which belong together. To overcome the overdrawing of the dots a bar plot is shown for each podium place at the bottom.

3.2 Interactive benchmark experiment plot

The interactive version of the plot only consists of the upper part of the static version, as the bar plots are just another visualization of the same data and can be shown in a separate barplot linked with the bench plot. There are two graphic elements representing interesting content: (1) a *dot* represents a model by performance measure; (2) a *line* represents a bootstrap sample. An interaction, like a mouse click, on one of them fires an event which calls a `R` function. In these event-handler functions one has all possibilities, here we use them to make things with other benchmark experiment space components which are not represented in the bench plot. This functionality of R-event-handler functions is added by us and not available in the official `icp` package.



(77): spl, pMSE = 2485.3

(77): rlm < lm < spl < loess

Figure 4: Possible interactions for the interactive bench plot. Mouse-click (left window) on a line highlights the line and all other linked objects. `Control`-mouse-over: a dot (right upper window clip) displays information about the model; a line (right lower window clip) information about the benchmark run.

All in all, the user interface of the plot strictly follows the user interface convention defined by `iPlots`, the interactions are performed using the same keyboard shortcuts. The most important interactions with their default behavior are:

Mouse-click: Highlight the object at the mouse position and all other linked objects, see Figure 4, left window. The event-handler function is empty as there is no common functionality which is usable for all benchmark scenarios.

Control-**mouse-over:** Show a tooltip with information returned by the event-handler function.

- The default information for a *dot* is the name and the performance value of the algorithm; Figure 4, upper-right window clip.
- The default information for a *line* is the bootstrap identifier and the order of the algorithms, as this is sometimes hard to determine with all the lines; Figure 4, lower-right window clip.

Other functionality is accessible using the context menu, e.g., hide the lines or step for- and backward through different versions of randomly broken ties.

In the next section we show by a example how the interactive bench plot with its functionality can contribute to an easy and fast analysis of the benchmark experiment space.

4 Interactive analysis

A common exploratory analysis of a benchmark experiment consists of the usage of various methods shown, e.g., in Eugster et al. [2008]. Among others, the bench plot of the prediction error, shown in Figure 3, is produced and this plot raises, for example, the questions *"which learning sample was used to train the worst* `loess` *model?"* (highlighted in Figure 4) or *"why is this single blue dot there at the fourth place?"*. It is not easily answerable with the static version, but with the interactive version of the plot.

First, we have to decide which additional visual representations of which benchmark experiment space elements are helpful for this specific analysis: (1) we visualize the raw data using a scatterplot. For higher-dimensional problems one can use a scatterplot matrix to show the raw data or use projection methods; (2) the bootstrap samples are visualized using a barchart with color brushing linked to the scatterplot; (3) and the models are represented as lines within the scatterplot. Of course, other representations are possible and one has to decide for each posing of a question which visualizations of which elements are suitable. The analysis is now a mixture of interactions and console-typed commands to translate the interactions between the different visualizations[2].

Figure 5 shows the splitting of the data for the benchmark run highlighted in Figure 4 together with the corresponding models. The barchart displays the frequency of the observations in the learning sample; the observations with zero appearance build the test sample. We see that the "modern-car" observation is not used for learning and the performance of the algorithms are in the upper level of the performance range. Is this coherence always true? Figure 6 shows the bench plot with all benchmark runs highlighted where the "modern-car" observation is not in the learning set – the assumed coherence applies.

The highlighting also uncovers that in these cases the smoothing spline algorithm performs best – this is an interesting and further investigable fact. We refine the highlighting to the cases where the smoothing spline algorithm has a lower performance than

[2] Most of this could be automated using the click-event-handler function.

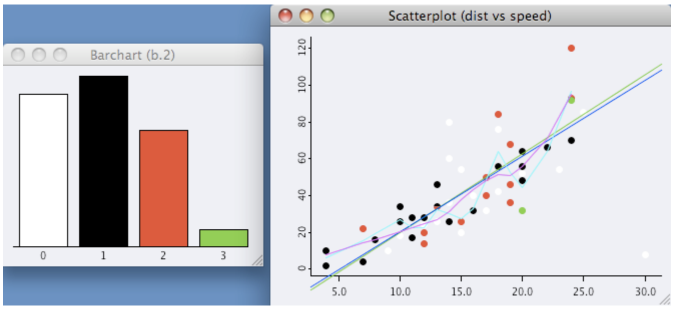


Figure 5: Visualizations of benchmark experiment space elements for the run highlighted in Figure 4: the raw data using a scatterplot; the bootstrap samples with a barplot linked to the scatterplot and showing the frequency of observations; the models as lines within the scatterplot.

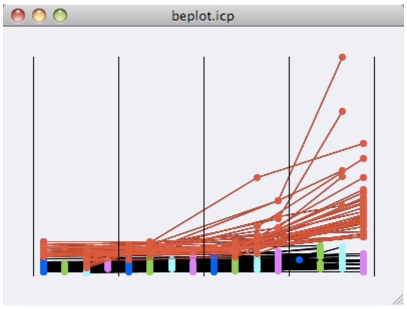
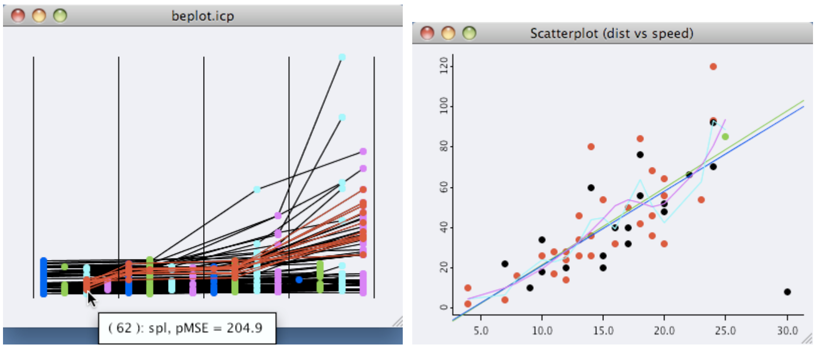


Figure 6: All benchmark runs highlighted where the “modern-car” observation is not used in the learning sample; they are all in the upper level of the performance range.

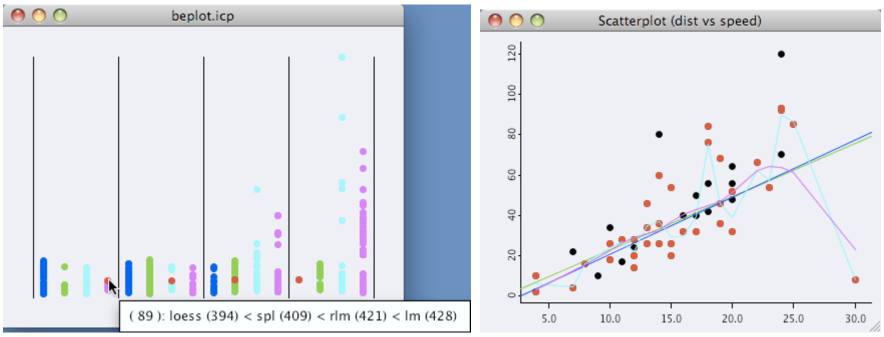
all other algorithms at the first place (Figure 7, left window). Run 62 is the one with the lowest performance of this selection and we take a look at the bootstrap sample (red and green displays the learning sample, black the test sample) and models (Figure 7, right window). As the green dot is in the training sample, the smoothing spline model ends with a descending line and therefore has a good prediction for the “modern-car” observation: 19.1 versus 95.1 (`lm`), 98.2 (`rlm`) and 192.5 (`loess`). The obvious assumption is that the green dot is in each learning sample of the highlighted cases, which turns out to be true.

Another interesting benchmark run is the one where the robust linear regression (blue) is on the last place. The low performance indicates that this is a run where all algorithms perform well and maybe the algorithm is there because of the randomly broken ties. As this dot or line is hard to grab we hide the lines. The highlighted dots look like they are in a row, so we redefine the *Control*-mouse-over-dot event-handler function to display the order of the algorithms with the rounded performance measures – we come to know that there is a small difference (Figure 8, left window). The bootstrap sample visualization



(62): spl, pMSE = 204.9

Figure 7: All benchmark runs highlighted where the “modern-car” observation is not used in the learning sample and the smoothing spline algorithm has a lower performance than all other algorithms at the first place (left window). And the details for the run with lowest performance of this selection (right window).

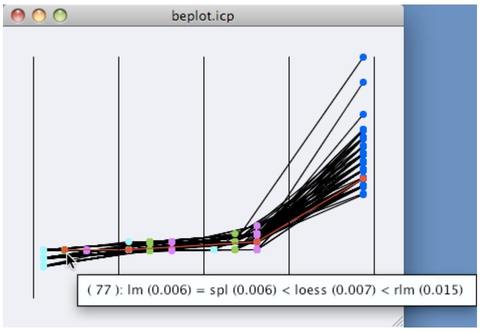


(89): loess (394) < spl (409) < rlm (421) < lm (428)

Figure 8: The bench plot with hided lines and information about the run where robust linear regression is on the last place (left window); and the details (right window).

(Figure 8, right window) shows that the “modern-car” observation is in the learning sample and, as we can see, it has a high impact (leverage factor) to the smoothing spline and local polynomial regression model.

During the execution of the benchmark experiment we also clocked the computation time of the model fitting process as another performance measure. This can be interesting if algorithms perform similar according to the “main” performance measure, e.g., the mean squared prediction error. For the analysis we display the time information in a further bench plot, which is linked with the bench plot containing the prediction error information. This allows us to consider more than one performance measure at a time. Figure 9 shows the window with the same case highlighted as in Figure 4; `lm` and `spl` consume equivalent computation time, `loess` nearly equivalent and `rlm` twice as much. The plot shows the interesting aspect, that there is not much difference between `lm` and `loess`, even `loess` estimates a lot of simple linear models, so the implementation seems highly optimized.



(77): lm (0.006) = spl (0.006) < loess (0.007) < rlm (0.015)

Figure 9: Bench plot of the second performance measure, the computation time of the model fitting process, linked with the bench plot showing the predication error (selection from Figure 4).

5 Conclusion

This paper showed how to integrate interactivity into the process of the exploratory analysis of benchmark experiments. The interactive version of the bench plot allows a rapid exploration of the results of benchmark experiments and, with the concept of linkage, of the whole benchmark experiment space. The idea is to use this as a "kind of pre-processing" to further analyses with exploratory data and model analysis procedures and methods.

The method presented in this paper is usable for benchmark experiments with one data set, but they can consist of more. We introduced some visualizations for these kind of experiments in Eugster et al. [2008], where the idea of "behind each plot primitive is a statistical object" applies too. Therefore, interactivity could be added in a meaningful way and would be helpful during the exploratory analysis.

Acknowledgments

Simon Urbanek for getting me on the way with `iPlots+icp` by implementing a first prototype of the bench plot within an hour. Sebastian Kaiser for a lot of useful discussion.

References

Dianne Cook and Deborah F. Swayne. *Interactive and Dynamic Graphics for Data Analysis: With R and GGobi*. Springer-Verlag, 2007. ISBN 978-0-387-71761-6.

Manuel J. A. Eugster and Friedrich Leisch. Bench plot and mixed effects models: First steps toward a comprehensive benchmark analysis toolbox. In Paula Brito, editor, *Compstat 2008—Proceedings in Computational Statistics*, pages 299–306. Physica Verlag, Heidelberg, Germany, 2008. ISBN 978-3-7908-2083-6.

Manuel J. A. Eugster, Torsten Hothorn, and Friedrich Leisch. Exploratory and inferential analysis of benchmark experiments. Technical Report 30, Institut für Statistik, Ludwig-Maximilians-Universität München, Germany, 2008. URL http://epub.ub.uni-muenchen.de/4134.

Alexander Gouberman and Simon Urbanek. *icp: Interactive Custom Plots - customizable interactive graphics for R*, 2008. URL http://www.rosuda.org/iPlots/. R package version 1.1-0.

Torsten Hothorn, Friedrich Leisch, Achim Zeileis, and Kurt Hornik. The design and analysis of benchmark experiments. *Journal of Computational and Graphical Statistics*, 14(3):675–699, 2005.

Donald R. McNeil. *Interactive Data Analysis*. Wiley, New York, 1977.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. URL http://www.R-project.org. ISBN 3-900051-07-0.

Antony Unwin and Chris Volinsky. Parallel coordinates for exploratory modelling analysis. *Computational Statistics & Data Analysis*, 43, 2003.

Simon Urbanek and Martin Theus. iplots: High interaction graphics for r. In Kurt Hornik, Friedrich Leisch, and Achim Zeileis, editors, *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*, 2003.

Simon Urbanek and Tobias Wichtrey. *iplots: iPlots - interactive graphics for R*, 2008. URL http://www.iPlots.org/. R package version 1.1-3.

William Venables and Brian Ripley. *Modern Applied Statistics with S*. Springer-Verlag, fourth edition, 2002.

Hadley Wickham. Exploratory model analysis with r and ggobi. *JSM 2007*, 2007.