STAT 6560
Graphical Methods

Spring Semester 2009

Project Work 2

Rong Xia

Utah State University

Department of Mathematics and Statistics

3900 Old Main Hill

Logan, UT 84322–3900

iplots

Interactive Graphics for R

Introduction

Interactive features are the building blocks for a graphical data analysis. Although interactive statistical graphics plays a major role in a data analysis process, it usually needs to be complemented with classical statistical methods or other data mining tools. However, existing interactive tools either make quite few efforts or are too cumbersome in incorporating statistics methods into graphics.

iplots is a **R** library which brings interactive graphics into the **R** enviornments. It is developed by Simon Urbanek and Tobias Wichtrey and its website is `http://www.iplots.org`. Using the R/SJave Interface, *iplots* offers a variety of customization options that can be modified and enhanced directly through the **R** interface. The current version of *iplots* offers scatterplot, histogram, barchart, box plots, mosaic plots, hammock plots, parallel coordinate plots and maps.

There is quite some similarity between *iplots* and *Mondrian*, such that many of the graphics from *Mondrian* can be created with *iplots* as well. However, the reader would need to learn the functions of the *iplots* packages as there is no simple point-and-click interface for creating plots as in *Mondrian*.):.

Compared with *GGobi*, *iplots* is more flexible in communicating with **R**, while *GGobi* is better at displaying 2D tour. The point-and-click interface for brushing is quite similar between *GGobi* and *iplots*. Readers familiar with one package should have no difficulty in using others.

Function Reference

- **Design Choices**

  iplots comes as an **R** library and it is invoked simply by the command *library(iplots)*. Plots in *iplots* use the same names as their static counterparts, except for the leading 'i'. E.g. to bring up an interactive histogram one simply types *ihist(data)*, where data is any arbitrary **R** vector. The most important plot options which specify colors, labels and scale information are supported transparently by *iplots*.

  iplots maintains a separate copy of all data that are displayed in the iPlots. This allows iPlots to opearate even after the underlying data has been deleted in **R**.

It also allows iPlots to use hot linking in all aspects, including update of plots on data changes. **iSets**, a kind of special data frame, is introduced in order to achieve these functions.

- **Plots**
  Here is a short listing of all functions related to iPlots implemented so far. For brevity reasons the parameter listed may not be complete and does not include default values.
  - *ibar(data, col)* : Interactive Bar Chart.
  - *ibox(x, y=NULL, ...)* : Interactive Box Plot.
  - *ihammock(vars, ...)* : Interactive Hammock Plot.
  - *ihist(x, breaks, col, bwidth,anchor, right, main, xlab, ylab,xlim,ylim)*
    Interactive Histogram Plot.
  - *imap(x, y=NULL, ...)* : Interactive Map.
  - *imosaic(vars, ..., type="observe")* : Interactive Mosaic Plot.
  - *ipcp(vars, ...)* : Interactive Parallel Coordinates Plot.
  - *iplot(x, y=NULL, xlab=NULL, ylab=NULL, ...)* : Interactive Scatterplot.
- **Manipulation of Plots and Data Sets**
  iplots support two methods for accessing individual plots. The first interface was defined to resemble the functions used in conjunction with graphics devices, such as:
  - *iplot.data(id=NULL)* : Retireve data from a plot.
  - *iplot.list*
    Interactive plots management functions, used to manage currently open iPlots.
  - *iplot.manip*
    iPlot manipulation functions, used to mainpulate iPlots.
  - *iplot.opt(..., plot=iplot.cur())* : Modify parameters of an interactive plot.
  - *iset, ivar*
    iSet and iVar - functions for managing data in iPlots.
  - *iset.list*
    iSet management functions, used to manage iSets.

The second means of accessing individual plots without changing the current plot is to use the plot object returned by plot creating commands.

- **Selection and Color Brushing**
  - *iset.col(col)* : Set color of cases in an iSet.
  - *iset.selected*
    These functions modify the selection or return the currently selected (highlighted) cases.

- **Toolkit**
  Every iPlot can have an arbitrary number of additional graphical objects attached, such as lines, rectangles, polygons or labels. Those objects, called iObjects.
  - *iabline*
    These functions add a straight line to the current iPlot.
  - *ilines(x, y, col=NULL, fill=NULL, visible=NULL, plot = iplot.cur())*
    Add connected lines or polygon to the current iPlot.
  - *iobj.list*
    Interactive objects (iObjects) management functions, used to manage iObjects of an iPlot.
  - *iobj.opt(o=iobj.cur(),...)* : Modify parameters of an iObject.
  - *itext(x, y=NULL, labels=seq(along=x), ax=NULL, ay=NULL, ..., plot = iplot.cur())* : Add text to the current iPlot.

- **Event Handling**
  iplots offer some basic functions to make interaction between iPlots and **R** more flexible. These functions can be used to build animations and small interaction loops in **R**.
  - *ievent.wait()*
    Interactive events, provides a way of building interactive event loops in **R**.

Example

The primary goal of the study is to estimate the average housing rent prices in Munich taking into account the size, location and amenities of the dwelling.

We first look at the response variable "rent" over the "size" of the housing in a scatterplot. To better interpret, we use differnet colors for different "number of rooms" in the

house, and overlay the regression line on the scatterplot. There seems to be a strong positive correlation of "rent" and "size". See **Figure 1** for details.

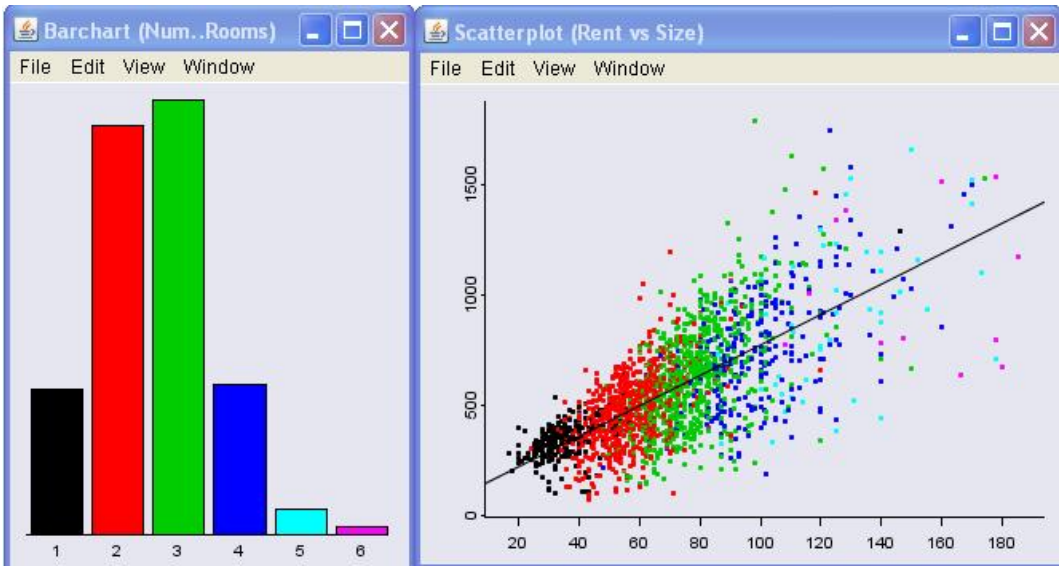

Figure 1: Scatterplot for the rent data

To further explore the data, we also highlight the observations with "rent> 855" and look at the barplot for the year of the house. We find out that both newly built and old houses tend to be more expensive, as displayed in **Figure 2**.

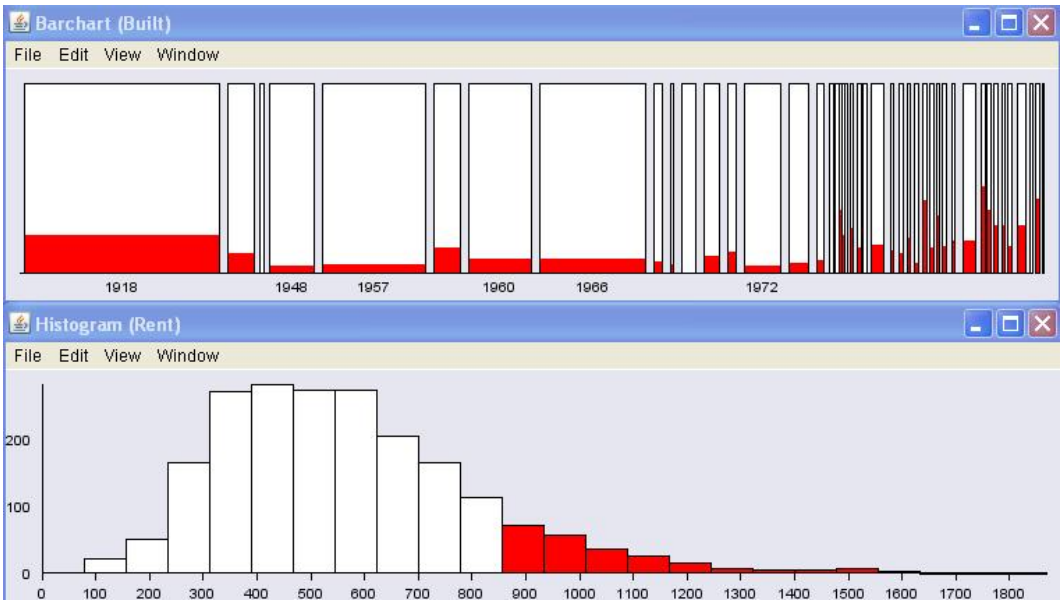

Figure 2: Histogram for Rent and Barplot for Years

These parallel boxplots show that neighborhood factors and amenities factors such as "Warm Water", "Central Heating", "Bath" and "Plus Kitchen" also play a role in determine the price of the rent, although these effects are not so siginificant. See **Figure 3** for details.

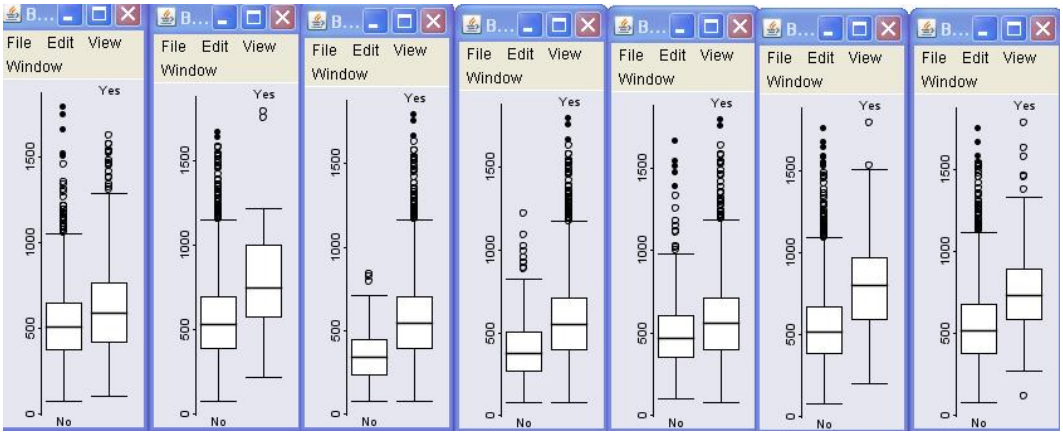

Figure 3: Parallel Boxplots for Neighborhood Factors and Amenities Factors

Finally we re-produce the scatterplot with a loop such that enables us to select a subset of the data and displays the smoother fitted to this subset, as shown in **Figure 4**.

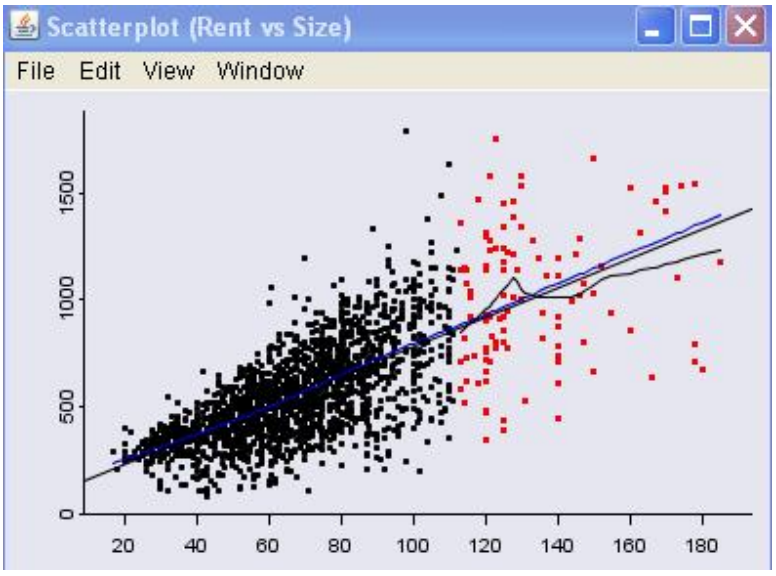

Figure 4: Interactive Scatterplot for Rent with Smoothers

Acknowledgements

This presentation is based on the following work from Martin Theus and Simon Ur-

banek: "iPlots: Interactive Graphics for R" (Theus & Urbanek (2004)), "Interactive Graphics for Data Analysis: Principles and Examples" (Theus & Urbanek (2008)) and the user manual for *iplots* package in **R**.

Links to Data Files and R-codes:

- `http://www.interactivegraphics.org/Datasets_files/rent.txt`
- `http://www.math.usu.edu/~symanzik/teaching/2009_stat6560/RDataAndScripts/xia_rong_project2_rent.R`

References

Theus, M. & Urbanek, S. (2004), 'iplots : Iteractive Graphics for R', *Statistical Computing and Graphics* **15**, 11–14.

Theus, M. & Urbanek, S. (2008), *Interactive Graphics for Data Analysis : Principles and Examples*, CRC Press, Boca Raton, FL.

`http://www.iplots.org`