Journal of Statistical Software

June 2012, Volume 49, Issue 8. *http://www.jstatsoft.org/*

Deducer: A Data Analysis GUI for R

Ian Fellows
University of California, Los Angeles

Abstract

While R has proven itself to be a powerful and flexible tool for data exploration and analysis, it lacks the ease of use present in other software such as SPSS and Minitab. An easy to use graphical user interface (GUI) can help new users accomplish tasks that would otherwise be out of their reach, and improves the efficiency of expert users by replacing fifty key strokes with five mouse clicks. With this in mind, **Deducer** presents dialogs that are understandable for the beginner, and yet contain all (or most) of the options that an experienced statistician, performing the same task, would want. An **Excel**-like spreadsheet is included for easy data viewing and editing. **Deducer** is based on Java's Swing GUI library and can be used on any common operating system. The GUI is independent of the specific R console and can easily be used by calling a text-based menu system. Graphical menus are provided for the **JGR** console and the Windows R GUI.

Keywords: GUI, R.

1. Introduction

R (R Development Core Team 2012) is a powerful statistical programming language that places the latest statistical techniques at one's fingertips through thousands of add-on packages available on the Comprehensive R Archive Network (CRAN) download servers. The price for all of this power is complexity. Because R analyses must be called as text commands, the user is required to find out the name of the function that will accomplish their task, and then remember that name along with the names of the variables to feed it, and its argument options. Perhaps more fundamentally, many users have never dealt with a program that requires them to type in commands that manipulate objects in the program.

For beginners, **Deducer** is designed to be an intuitive dialog based interface to common data manipulation and analysis tasks. It requires no initial knowledge of the R programming language. Data can be loaded, edited, and analyzed through graphical user interface (GUI) components without understanding the command functions. Output is sent to the console,

and custom R functions format the results into easy to read tables.

For expert R users, **Deducer** reduces the time necessary to construct a command, and minimizes the cognitive load of remembering infrequently used options. Otherwise, **Deducer** stays out of the way. The GUI is integrated into the regular R console, so using a mix of programming and GUI dialogs is natural.

2. Other R interfaces

There are a number of actively maintained GUI projects for R. Each of these packages are great efforts with their own advantages and disadvantages. Table 1 displays a comparison chart of the major R data analysis GUI efforts.

2.1. R Commander

R Commander (Fox 2005) is the most mature GUI project for R. Its dialogs cover an extensive set of analyses, plots, and data manipulation tasks. It also includes an easy to use plug-in system so its scope can be extended by interested programmers. The goal of the package is to facilitate student learning in basic statistics courses with the hope that they will eventually

	R Commander	**Rattle**	**RKWard**	**Deducer**
Environment and features				
Integrated into any console	no	no	no	yes
Cross-platform				
Windows	yes	yes	yes	yes
Mac	Non-native	Non-native	no	yes
Linux	yes	yes	yes	yes
Drag-and-drop	no	no	no	yes
HTML output	no	no	yes	no
Formatted output tables	no	no	yes	yes
Dialogs have memory	no	yes	yes	yes
GUI toolkit	Tcl/Tk	GTK	KDE	Java Swing
Functionality				
Univariate analysis	yes	yes	yes	yes
Longitudinal analysis	limited[1]	no	limited[1]	no
Plots	yes	limited[2]	yes	yes
Analysis visualizations	limited[2]	limited[2]	no	yes
Data manipulation	yes	yes	no	yes
Spreadsheet	yes	no	yes	yes
Extensibility				
Has plug-in architecture	yes	no	yes	yes
Extendable from within R	yes	no	no	yes
Number of plug-ins	32	0	0[3]	5 + 5[4]

Table 1: GUI project feature overview. (1) Time series only. (2) Only implemented for a few tasks. (3) Plug-ins loaded by default not included. (4) Five plug-ins on CRAN and five on R-Forge.

be "weaned off" of the GUI. **Deducer**, on the other hand, takes the view that a GUI can be useful for experts as well as students, provided it is designed with them in mind.

R Commander has a rich set of plug-in packages that greatly extend the scope of the statistical methods covered by the GUI. See Fox and Carvalho (2012) in this special volume for an example. As of this writing, there are 32 R packages on CRAN that extend or enhance R Commander. They range in subject matter from teaching to text mining, and provide a diverse functionality far outstripping any other R GUI project. That said, from the perspective of a professional data analyst, there are a number of design issues that make it sub-optimal for daily use. These (perhaps subjective) sub-optimalities are what lead to the development of **Deducer** as a separate project.

Console: R Commander requires the user to use a separate R console which manages the R commands and displays the output, this is a less than ideal set-up for advanced users who are comfortable with the command line. **Deducer** is integrated with the terminal, the usual Windows console, and **JGR**.

Task based dialogs: R Commander has separate dialogs for each R function it uses. For example, there are 6 menu items for loading data from a file (one for each loading function), whereas **Deducer** has one which picks the right function based on extension and user selection.

Pretty GUI toolkit: R Commander uses the Tcl/Tk toolkit to create the interface. Though cross-platform, the widgets are not very nice looking and can deviate from the normal platform look and feel. This is especially true on the Mac, where the non-native X Windows system is used. **Deducer** uses the Java, which is also cross-platform, but supports a more integrated look and feel with more advanced features such as drag and drop.

Dialog memory: One of the constantly vexing things about R Commander is that the dialogs do not remember what options were selected the last time they were used. Data analysis is an iterative process, it is very rare that the user specifies exactly the right set of options the first time.

2.2. Rattle

Rattle (R Analytical Tool To Learn Easily, Williams 2009) is another GUI based on the Gnome graphics system and is focused on data mining rather than classical statistics. As a result, it facilitates analyses in areas such as neural networks, and support vector machines, but provides no way to analyze contingency tables. Like R Commander, **Rattle** creates a new console window which controls the commands and logs the resulting output. **Rattle** is limited to working with one dataset and one outcome variable at a time.

2.3. RKWard

RKWard (Rödiger, Friedrichsmeier, Kapat, and Michalke 2012) is, perhaps, the most similar project to **Deducer** in terms of its goals. It features a spreadsheet data editor and a comprehensive R console replacement. Its data analysis features are accessed through menu items, and results are available as neatly formatted HTML tables. Each bit of GUI output is

accompanied by a `Run again` button, which brings up the dialog in the state that was used to generate the results. This facilitates reproducible research, and is a great feature.

Unfortunately, **RKWard** is in its infancy when it comes to non-Linux systems. There are a number of known issues on Windows, and no binary version is available for the Mac.

3. Getting oriented

3.1. Installation

Deducer is compatible with most R consoles, but is best integrated with the **JGR** console (Helbig, Theus, and Urbanek 2005). Because **JGR** is Java-based, **rJava** (Urbanek 2011) takes care of the synchronization issues between **JGR**'s read-eval-print loop and the dialogs' R calls, so the data viewer and dialogs do not need to block the console while open. On Mac OS X, **Deducer** must be used from within **JGR**.

Installation is fairly standard as it is available from the CRAN at `http://CRAN.R-project.org/package=Deducer`. `install.packages("Deducer")` downloads the latest version from CRAN and installs it. **Deducer** can then be loaded with `library("Deducer")`. If **Deducer** is called from the Windows R GUI, three new menus appear in the menu bar (`Deducer`, `Data` and `Analysis`).

If **Deducer** is called from the terminal, one can navigate its menus via a text based menu system accessed by calling `deducer()`. Specific dialogs can be brought up by passing its name to the deducer function. For example `deducer("Open Data")` will bring up the `Open Data` dialog with no need to traverse the menu system.

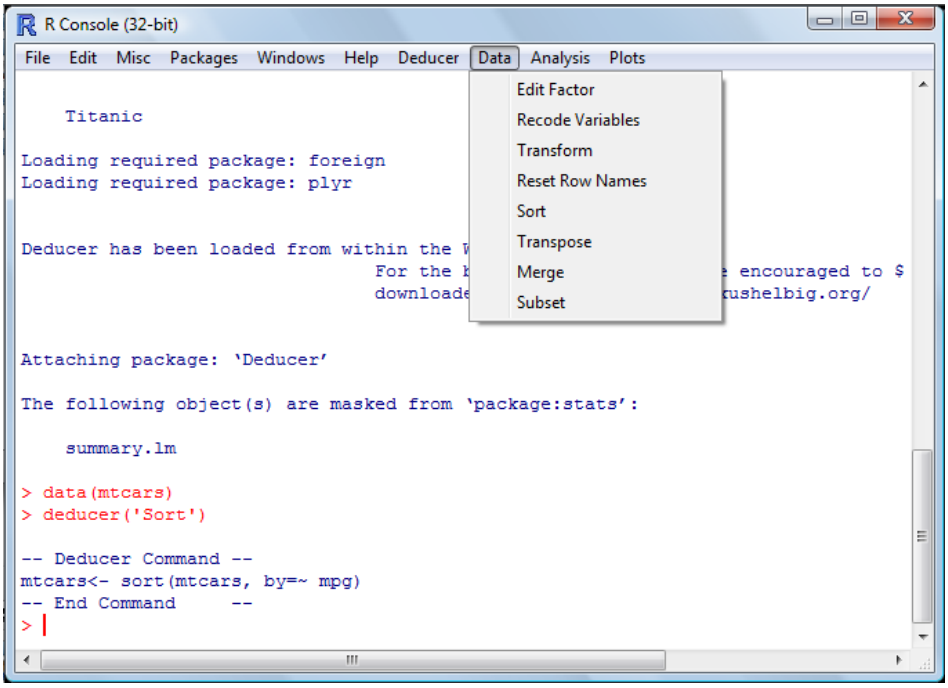

Figure 1: Deducer in the windows GUI.

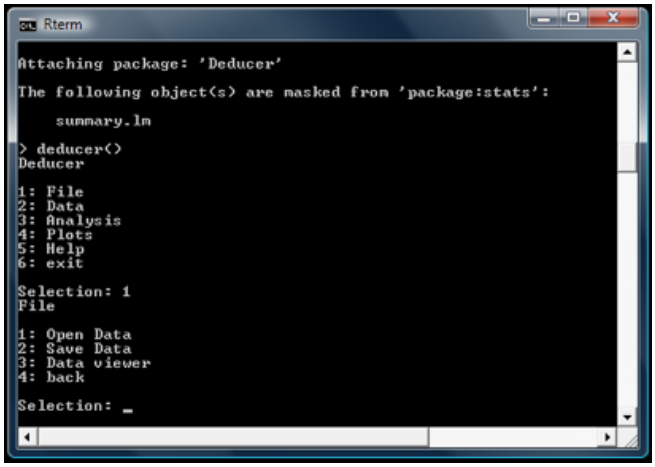

Figure 2: **Deducer** in the terminal.

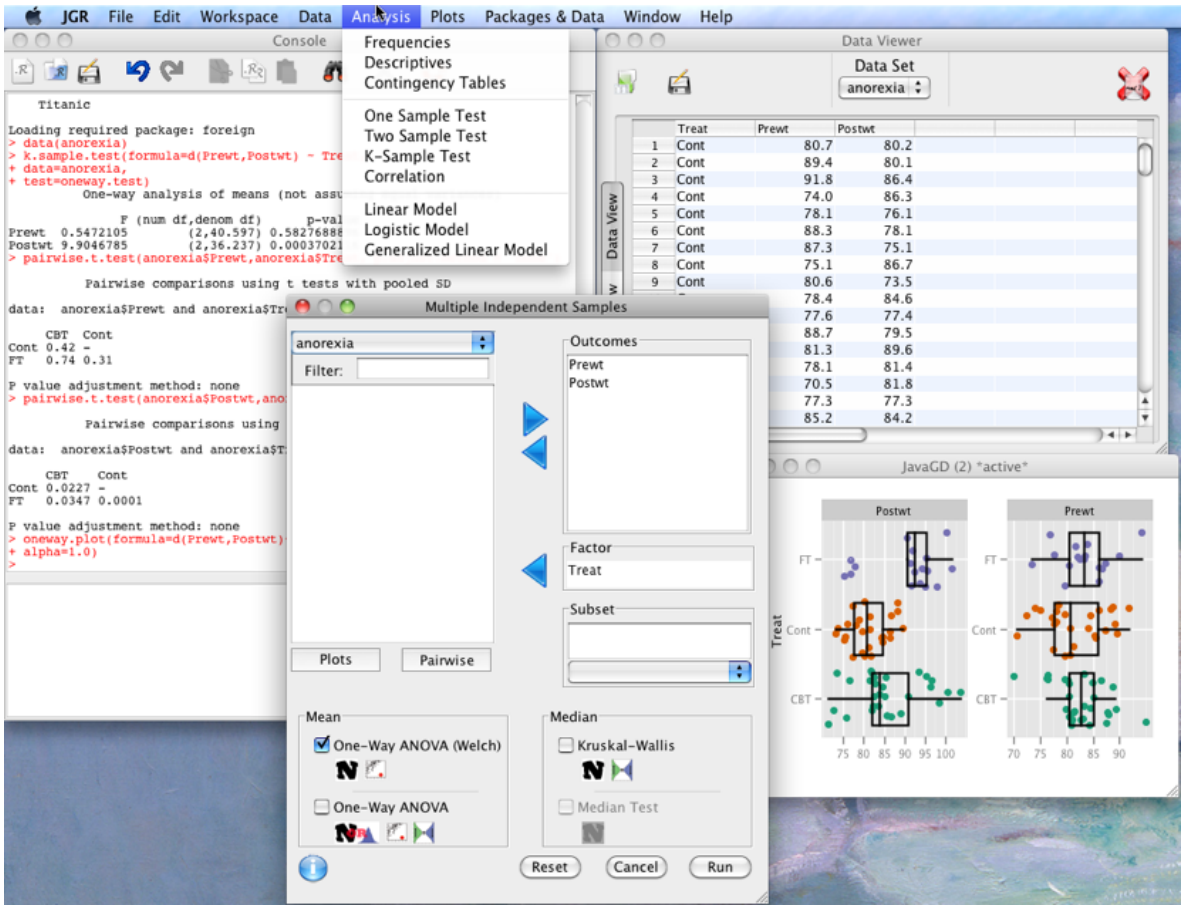

Figure 3: **Deducer** in **JGR**.

File type	Extension	R function
R workspace	`*.rda` and `*.rdata`	`load()`
R object	`*.robj`	`dget()`
Comma seperated	`*.csv`	`read.table()`
Text file	`*.txt`	`read.table()`
SPSS	`*.sav`	`read.spss()`
SAS export	`*.xpt`	`read.xport()`
DBase	`*.dbf`	`read.dbf()`
Stata	`*.dta`	`read.dta()`
Systat	`*.sys` and `*.syd`	`read.systat()`
ARFF	`*.arff`	`read.arff()`
Epiinfo	`*.rec`	`read.eipinfo()`
Minitab	`*.mtp`	`read.mtp()`
S data dump	`*.s3`	`read.S()`

Table 2: `Open Data`: supported formats.

Finally, **Deducer** can be used within **JGR**. This is the recommended environment and descriptions from this point forward will refer to the **JGR** console, though all examples can be executed easily in other environments.

Extensive documentation and video tutorials are available through the online manual (Fellows 2012) which is located at `http://www.Deducer.org/manual.html`. The information button in the lower left hand corner of each dialog provides access to the part of the manual relevant to that particular task.

3.2. Loading and saving data frames

The `Open Data` *dialog*

JGR and **Deducer** support a simple yet versatile dialog for bringing in data from a file. The `Open Data` dialog facilitates the use of the `read.table` function for text delimited data, and uses the foreign package to load data written in other statistical programs such as SPSS or Minitab. The specific function needed is determined by the file extension, though this can be overridden by selecting a file format from the `Format` combo box. The R name that this data is loaded to can be changed by entering a value into the `Set Name` field in the lower left hand corner. The types shown in Table 2 are supported.

If the file is not a text file, or a comma-separated file, it will be loaded into the workspace. Otherwise, a supplemental dialog will be displayed showing a preview of the data file to make sure that all settings are correct.

- The `Record Separator` option allows the user to specify what character separates values in the data set.
- The `Quote` option lets the user define whether the data file defines character variables by surrounding them with quotation marks.
- The `Header` option specifies whether the first row contains the variable names.

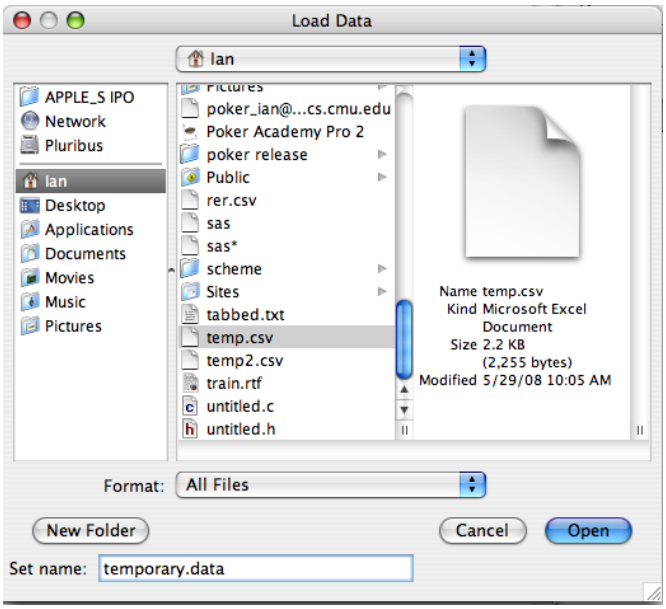

Figure 4: The `Open Data` dialog.

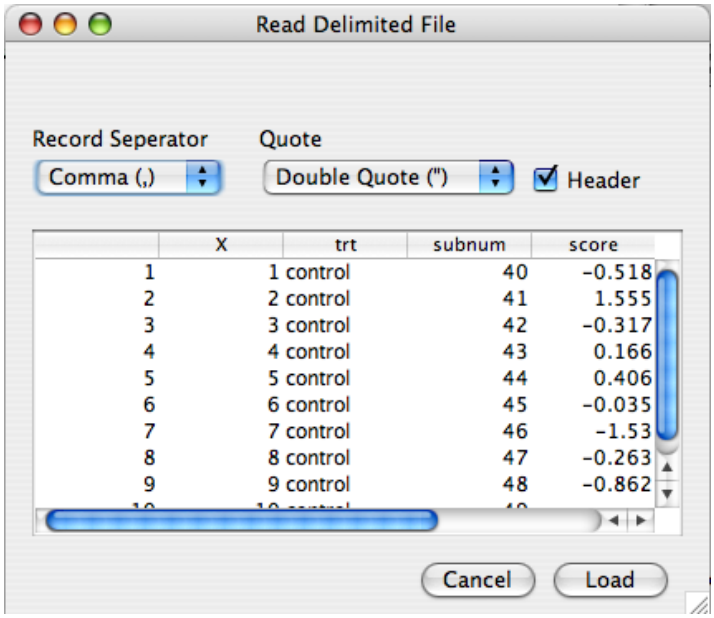

Figure 5: Loading a delimited text file.

The values of these options are intelligently guessed automatically. As the user changes the option values, the results are previewed in the preview table. This allows the user to load data without having to first open it in a supplemental program (such as **Notepad**) to determine its format.

The R function called by this dialog will depend on the type of data loaded. The example above produces the following `read.table` call:

File type	Extension	R function
R workspace	`*.rda` and `*.rdata`	`save()`
R object	`*.robj`	`dput()`
Comma seperated	`*.csv`	`write.table()`
Tab delimited	`*.txt`	`write.table()`
DBase	`*.dbf`	`write.dbf()`
Stata	`*.dta`	`write.dta()`
ARFF	`*.arff`	`write.arff()`

Table 3: `Save Data`: supported formats.

```
temporary.data <- read.table("/Users/Ian/temp.csv", header = TRUE,
  sep = ",",  quote = "\"")
```

The `Save Data` *dialog*

Saving data to a file is simple, no matter what format the data needs to be in. Using the `Save Data` dialog in the `File` menu, data can be saved in any of te formats shown in Table 3.

3.3. The `Data Viewer`

The `Data Viewer` provides an easy to use, spreadsheet-like environment to view and edit data. Copy and pasting is supported, compatible with **Excel** 2003/2007, so data can be moved from **Excel** to R by simply copying it to the `Data Viewer`. Contextual menus are used to insert, delete and copy rows and columns.

The `Data Viewer` consists of two tabs, one containing the data table, and the other displaying variable information. The `Data View` shows the data frame values, which can be copied,

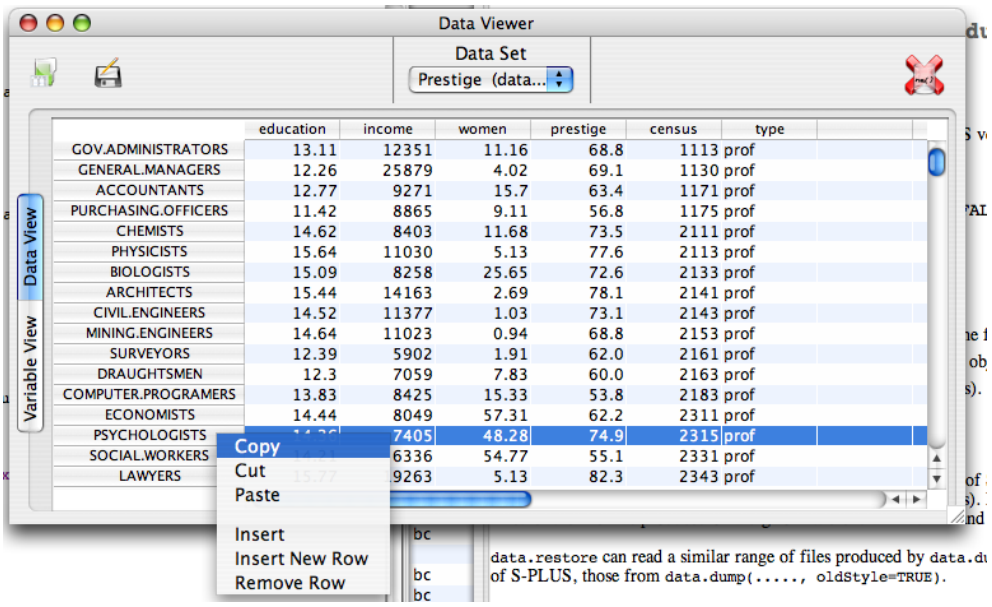

Figure 6: The `Data View`.

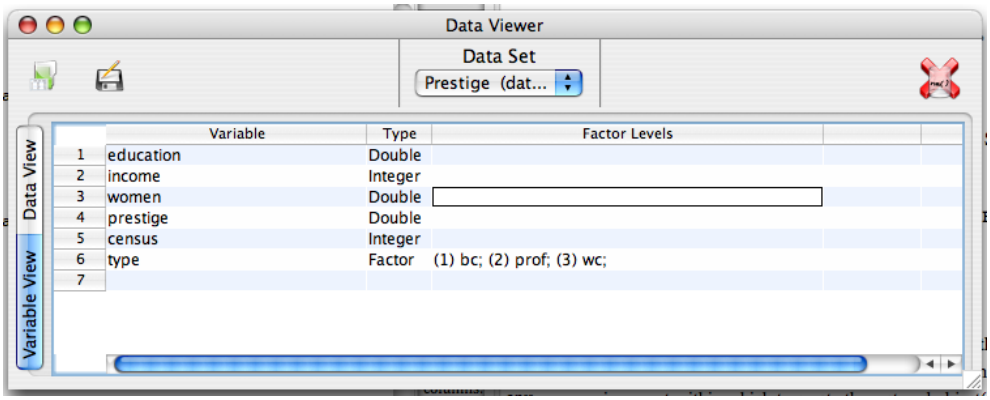

Figure 7: The `Variable View`.

pasted, or edited in a manner similar to **Excel**. Right clicking (or command clicking on the Mac) on the row or column headers allows the user to insert, copy, or delete columns and rows. In the `Variable View` the properties of each variable in the data frame can be edited. The variable column represents the variable name. The type column determines the storage type. Variables can be converted to and from character, double (numeric), integers, logical (yes/no), factor, date or time. The levels of factors are displayed in the `Factor Levels` column, and can be edited by clicking on the appropriate cell, bringing up the `Factor Editor`.

4. The dialogs

The dialogs are located in the `Data` and `Analysis` menus. Unlike some other statistical GUIs, **Deducer**'s dialogs are conceptually organized by task. That is, each dialog represents a specific task that a user might sit down and wish to accomplish. For example, the two-sample test dialog includes the t test, as well as the Wilcoxon rank sum. Even though statistically, these tests belong to two separate families (parametric and non-parametric), from a task-based view, they both answer the same type of question: Are the central tendencies of the two independent samples different?

In addition to GUI dialogs, **Deducer** introduces quite a few new R functions. Some of these implement statistical algorithms for which no built in R function was available (e.g., permutation t tests, data frame sorting and mid p values for exact tests), others serve to provide a simple unified interface to a number of statistical procedures, and present their results in an easy to read format (e.g., `descriptive.table` and `two.sample.test`). There are also a few convenience and utility functions included. One that is used in many of the analysis dialogs is '`d`', which is a keystroke saving wrapper function for `data.frame`.

Detailed instructions for using the dialogs will not be presented, instead, brief functional descriptions and example output will be given. For the interested reader, the online manual (`http://www.Deducer.org/manual.html`) provides detailed descriptions of each GUI dialog, as well as several analysis examples.

4.1. Data manipulation

Before any analysis, the data needs to be manipulated into a form amenable to analysis. **De-**

ducer provides several dialogs that can be helpful in altering the data frame and its variables.

`Edit Factor`**:** This dialog can be used to add to, or remove levels of a factor, their order, and the type of contrast used (i.e., treatment, sum, Helmert, or polynomial).

`Recode Variables`**:** This dialog uses a custom R function (`recode.variables`) to transform specific values, or ranges of values, of a variable to new values.

`Transform`**:** Provides various variable transformations, including log, square root, standardization, binning, and Box-Cox.

`Reset Row Names`**:** Changes the row names of a data frame to integers from 1 to the number of rows.

`Sort`**:** Sorts a data frame by any number of variables of any type. Uses a new S3 method for the class `data.frame` for easy data frame sorting.

`Merge`**:** Combines two data frames into one, based on common identifiers. Warnings are given if the identifiers do not uniquely identify observations.

`Transpose`**:** Transposes the rows and columns of a data frame

`Subset`**:** Creates a subset of a data frame based on a logical expression.

4.2. Analysis

The `Analysis` menu contains dialogs providing descriptive information about variables, as well as inference functions which output p values, confidence intervals, effect sizes, and analysis visualizations. One design principle adhered to was the belief that all analyses should be accompanied by a visualization helping users understand the quantities being tested, and model fit.

`Frequencies`

Frequency tables provide descriptive information for categorical and ordinal variables. They display the number of cases that fall into each category of a specific variable, and calculate percentages. A new R function `frequencies` is provided which presents the results in easy to understand tables.

Features: Counts and percentages.

`Descriptives`

Calculates descriptive statistics for a set of variables, possibly stratified by another set. It supports a number of built-in statistics (e.g., Mean and Standard Deviation), and provides facilities for custom functions to be specified. This dialog uses a new R function `descriptive.table` which tabulates the statistics into nicely formatted tables.

Features: Mean, standard deviation, variance, max, min, percentiles, skew, and kurtosis.

Contingency Tables

Contingency tables (sometimes called crosstabs) are used to summarize and analyze the joint distribution of two variables, possibly stratified by a third. While R has powerful functions that can be applied to table counts, they spread out over a number of packages, and do not provide easy to read output. **Deducer** takes a unified approach to contingency tables.

First, a `contingency.tables` object is created, formed with any number of row or column variables, as well as one stratifying variable. When printed, the tables are formatted and displayed, with optional row and column percentages. Tests can then be added to the object, which causes them to be applied to each table. On printing the `contingency.tables` object, the tests are then formatted into tables. The user need not be aware of this R level structure. They can simply navigate the dialog and view the printed table.

Features: Percentages, expected counts, residuals, χ^2 test, likelihood ratio test, Fisher's exact test, Mantel-Haenszel, Kedall's τ, Spearman's ρ, Kruskal-Wallis, Monte Carlo exact tests, as well as mid p values.

One Sample Test

Performs one sample distribution tests. The new `one.sample.test` function takes any number of variables, and performs a statistical test on each of them. The results of these tests are formatted into a table, making it easy to investigate multiple variables at the same time. Similarly, the `onesample.plot` function creates a plot with either histogram, or box-jitter plot summaries, with separate panels for each variable.

Features: One sample t test, Shapiro-Wilk test, histogram, box/jitter plot.

K Sample Test

Investigates differences in distribution between two or more samples. The new function `k.sample.test` is introduced, which applies any statistical test to the specified outcome variables, and formats the results into a readable table.

Features: ANOVA, Welsh ANOVA, Kruskal-Wallis, box-jitter plots, pairwise comparisons.

Correlation

This dialog creates arrays of correlations between any number of variables. A new function, `cor.matrix`, creates a table of correlations. To visualize these relationships, three new plotting functions are provided. `qscatter_array` and `ggcorplot` create scatter plots for each correlation value in the table. `plot.cor.matrix` displays circles whose areas are proportional to the sizes of the correlations.

Features: Scatter plot matrices along with Pearson's, Spearman's and Kendall's correlations.

4.3. Models

Deducer has an advanced system for constructing, analyzing and visualizing models. The GUI helps the user specify the model formula, which can include interaction and nested

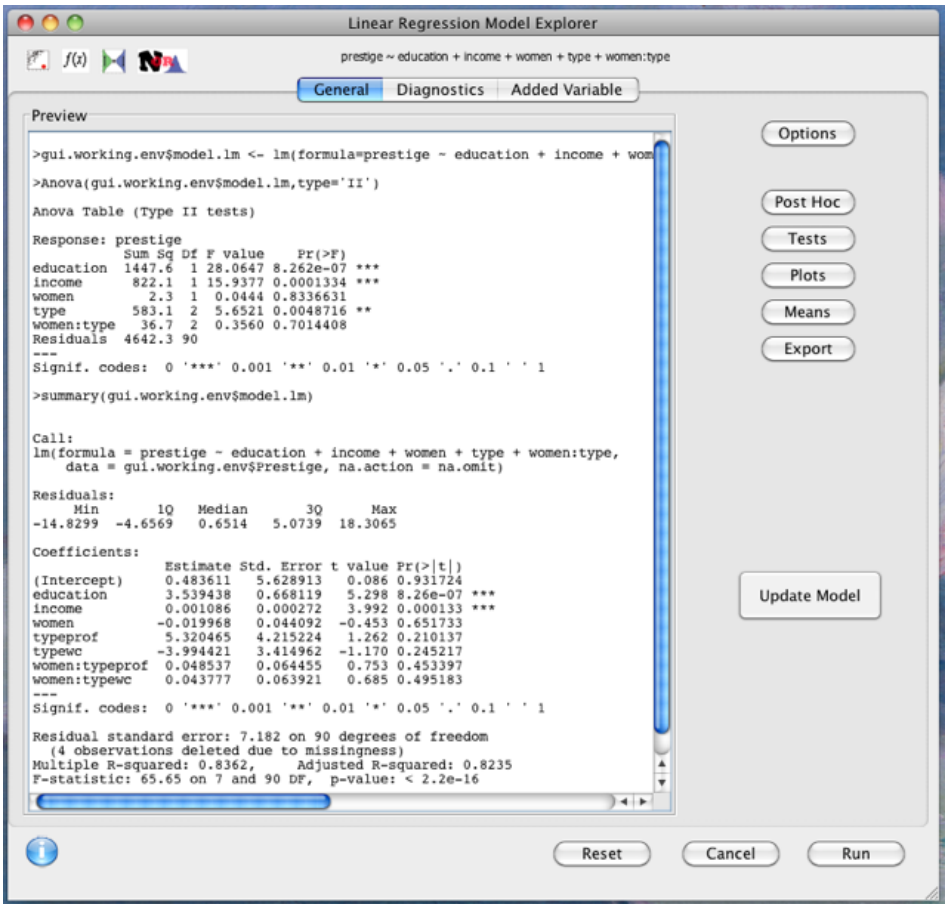

Figure 8: Model explorer.

terms. Next it presents the user with the model explorer (displayed in Figure 8), which shows a preview of their results, along with options for model testing, assessing model fit, and result visualization. All generalized linear models are supported, including linear and logistic regression models. For several options the **car** package (Fox and Weisberg 2011) is leveraged.

The tabs at the top of the dialog in Figure 8 present the users with plots to asses the assumptions of the linear model. The icons in the top left outline the major assumptions being made. From left to right these are: No outliers, linearity, equal variance (homoskedasticity), and either normality, or a sufficiently large sample size. The buttons on the right bring up a wide array of options for investigating the regression model, including most of those that a professional analyst would need.

`Options:` Common model summaries, including ANOVA (type II or III) tables, parameter summaries, heteroskedasticity-consistent covariance matrices (linear model), variance inflation factors, observation influence, and parameter correlations.

`Post Hoc:` Post-hoc tests/intervals with 10 types of contrasts and 12 types of multiple comparison corrections.

`Tests:` Linear hypothesis tests.

`Plots:` Model visualizations, including effect plots, and ROC (receiver-operator characteristic) plots (logistic model).

`Means:` Post-hoc means with confidence intervals.

`Export:` Handles R object creation, including residuals (regular, standardized and studentized), predicted values, Cook's distance, DFBETA, hat, covariance ratio, DFFITS.

4.4. Plots

There are two ways to create plots within the GUI. The first is to use the plot builder, which is a graphical interpretation of the grammar of graphics (Wilkinson 1999) as implemented by the **ggplot2** package (Wickham 2009). It is a flexible tool, which can create virtually any type of plot. Secondly, **Deducer** also interfaces with the **iplots** package (Urbanek and Wichtrey 2011) to provide interactive graphics for data exploration.

5. Extensibility

An important aspect of **Deducer** is its ability to be extended by third-party extensions. Once a developer has created a GUI for a particular method, all that is nessisary is to add it to the menu system. For example, if a developer has created a factor analysis GUI that is started when the R code `dialog$run()` is executed, the following code will add it to the menu system.

```
deducer.addMenu("Example")
deducer.addMenuItem("Factor Analysis",,"dialog$run()","Example")
if(.windowsGUI){
        winMenuAdd("Example")
        winMenuAddItem("Example", "Factor Analysis",
                                    "deducer('Factor Analysis')")
}else if(.jgr){
        jgr.addMenu("Example")
        jgr.addMenuItem("Example", "Factor Analysis",
                                    "deducer('Factor Analysis')")
}
```

The first two lines of the code adds a new menu named `Example` to the text menus system, and then creates a menu item to run the factor analysis. Next, if we are in **JGR** or the Windows R GUI, then we can add to the GUI menu system.

A developer can use any programming language or GUI toolkit they feel comfortable with, but the recommended toolkit is Java Swing. **Deducer** provides a number of GUI components specifically designed for R and data analysis. A full list of these components along with documentation is available on the website.

6. Conclusion

Deducer's dialogs and R functions make common data analysis tasks in R easier to both perform and interpret. No programming or scripting is required, which opens R to a large

audience of users who would otherwise be locked out of experiencing R's power. The breadth of analyses covers most of what would be taught in introductory, and intermediate, applied statistics courses, thus making it appropriate for use in the classroom.

Undoubtedly, there are some omissions from the system, for example only univariate analyses are currently implemented. As **Deducer** becomes more mature, its breadth will continue to grow. Because statistics is a large and constantly growing subject, there will always be areas that **Deducer** does not provide a GUI for. However, as **Deducer** is extensible through a plug-in system, any interested party can create a package implementing a GUI for their favorite analysis, which can then be accessed through **Deducer**'s menus.

References

Fellows I (2012). ***Deducer**: An R Graphical User Interface (GUI) for Everyone*. Version 2012-01-05, URL http://www.Deducer.org/manual.html.

Fox J (2005). "The R Commander: A Basic-Statistics Graphical User Interface to R." *Journal of Statistical Software*, **14**(9), 1–42. URL http://www.jstatsoft.org/v14/i09/.

Fox J, Carvalho MS (2012). "The **RcmdrPlugin.survival** Package: Extending the R Commander Interface to Survival Analysis." *Journal of Statistical Software*, **49**(7), 1–32. URL http://www.jstatsoft.org/v49/i07/.

Fox J, Weisberg S (2011). *An R Companion to Applied Regression*. 2nd edition. Sage Publications, Thousand Oaks.

Helbig M, Theus M, Urbanek S (2005). "**JGR**: A Java GUI for R." *The Computing and Graphics Newsletter*, **16**(2), 9–12.

R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Rödiger S, Friedrichsmeier T, Kapat P, Michalke M (2012). "**RKWard**: A Comprehensive Graphical User Interface and Integrated Development Environment for Statistical Analysis with R." *Journal of Statistical Software*, **49**(9), 1–34. URL http://www.jstatsoft.org/v49/i09/.

Urbanek S (2011). ***rJava**: Low-Level R to Java Interface*. R package version 0.9-3, URL http://CRAN.R-project.org/package=rJava.

Urbanek S, Wichtrey T (2011). ***iPlots**: Interactive Graphics for R*. R package version 1.1-4, URL http://CRAN.R-project.org/package=iplots.

Wickham H (2009). ***ggplot2**: Elegant Graphics for Data Analysis*. Springer-Verlag, New York.

Wilkinson L (1999). *The Grammar of Graphics*. Springer-Verlag, New York.

Williams GJ (2009). "**Rattle**: A Data Mining GUI for R." *The R Journal*, **1**(2), 45–55. URL http://journal.R-project.org/.

Affiliation:

Ian Fellows
Department of Statistics
University of California, Los Angeles
Los Angeles, United States of America
E-mail: ifellows@stat.ucla.edu
URL: http://www.Deducer.org/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 49, Issue 8 *Submitted:* 2011-02-28
June 2012 *Accepted:* 2011-09-08