Benchmarks Online

Research and Statistical Support
University of North Texas

RSS Matters

Link to the last RSS article here: Using S-Plus Graphics in Excel; SPSS News - Ed.

## Two User Interfaces to the R Statistical System: Rcgi and R Commander

By Dr. Rich Herrington, Research and Statistical Support Services Manager

This month we continue our exploration of window interfaces for R: R Commander - a simple drop down menu system; and Rcgi - a CGI based web interface to R. The following is an excerpt from the R website http://www.r-project.org:

> *R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R. R provides a wide variety of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity. One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control. R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs out of the box on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux). It also compiles and runs on Windows 9x/NT/2000 and MacOS* (from *Introduction*).
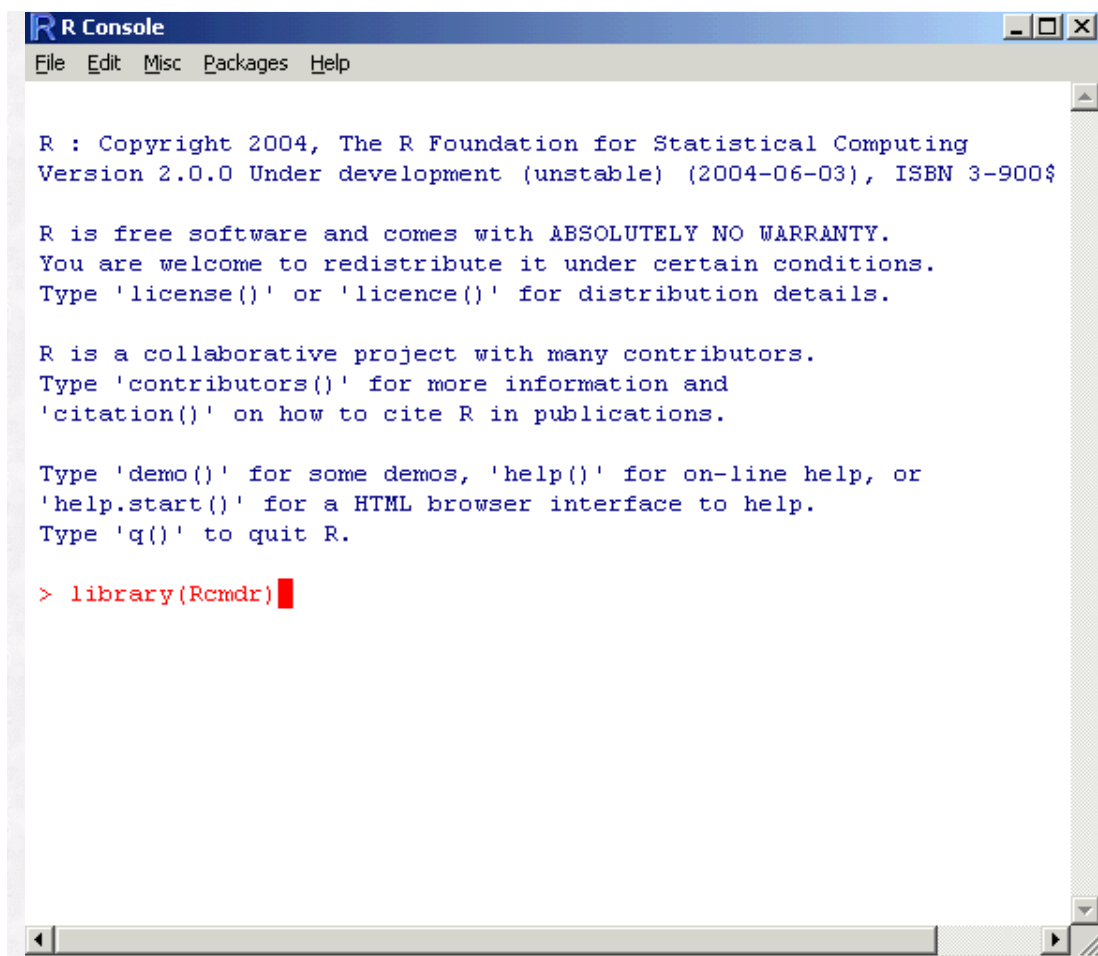
Starting R in the SDI Mode

At this point it is assumed that you have installed R and have an R icon on your desktop (see the earlier *Benchmarks Online* article on downloading and installing R). Additionally, it is assumed that you have installed all of the supporting R Commander (Rcmdr) packages. To use the Rcmdr library, we will want to configure the R system window to have a "single document interface" mode (sdi). Right mouse click the R icon on your desktop and select **properties**. Add " --sdi " to the Target field after the Rgui.exe statement (see below). Click **apply**, then **ok**. Double click the R icon on your desktop to start the R system.
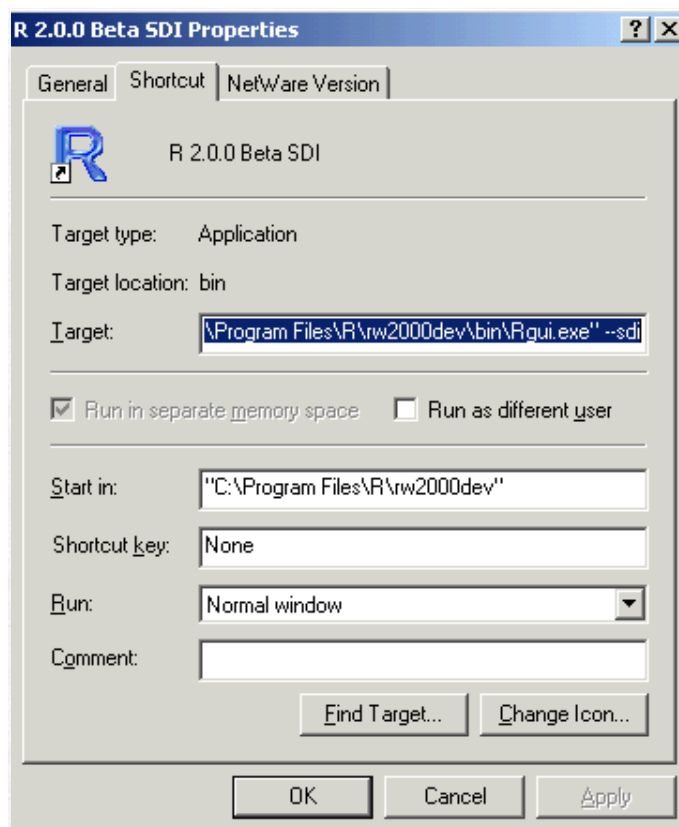
Loading the R Commander (Rcmdr) Package

To load the Rcmdr package, select **Packages** from the main menu bar in your **main R console window** and select **Load Package**. **Select Rcmdr** in the list. The Rcmdr window system will appear. **Alternatively**, you can **manually load the Rcmdr package** window from the main R console window by typing the "library" function, with the appropriate library, into the command prompt:

After selecting the Rcmdr package in the package selection window OR by loading the package manually, the following menu system will appear:

Some Fundamentals of the Rcmdr Window System

The Rcmdr window system can be resized by "grabbing" one of the corners of the window with the mouse (hover over the corner until you get a "double-headed" arrow" then press your left mouse button and drag the mouse). The upper window in the Rcmdr window is a script window where commands can be typed and then be submitted for processing. Once these commands have been typed in, the commands can be highlighted and the "Submit" button can be clicked. Alternatively, a **right-mouse click on the highlighted script brings up a small menu where the script can be submitted for processing.** Additionally, this menu can be used to find, select all, delete, copy and paste scripts that are in this window. Type the following script in the R script window; highlight the script; right-mouse click and "Submit" the script:
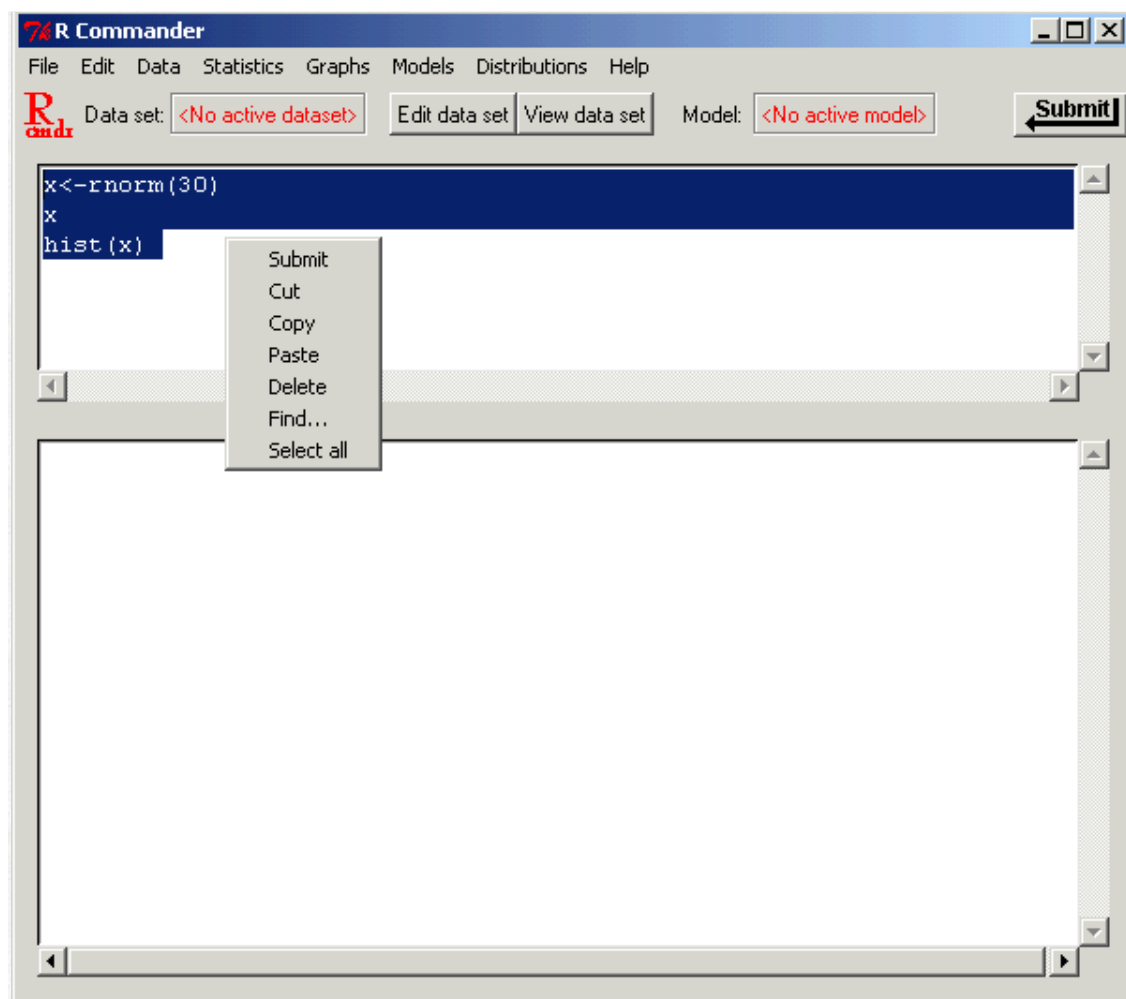
Submitting the R script above, generates 30 random normal numbers; places them in the object "x"; displays the object "x"; and then generates a frequency distribution of the numbers (histogram). The bottom window of the Rcmdr generates the output from your R commands. You can scroll through this output; copy, cut, and paste the output.

```
R Commander                                                    _ |□| X
File  Edit  Data  Statistics  Graphs  Models  Distributions  Help

R cmdr   Data set: <No active dataset>   Edit data set | View data set   Model: <No active model>        Submit

x<-rnorm(30)
x
hist(x)



> x<-rnorm(30)
> x
  [1]  -1.64463896  -0.04398871   0.83432024   1.51068900  -1.90670113   0.7
  [7]   1.05309295   0.47228434   0.50326244  -0.57903772  -0.34461794   0.3
 [13]   1.90405213  -1.17491388  -1.00386514  -0.05670504   0.08162636   1.9
 [19]  -0.18278839   2.15320076   1.25403735  -0.47775569   1.59295251  -0.9
 [25]  -0.81395899  -0.55241804  -1.10287473   0.15682651   0.52454880  -0.2
> hist(x)
```

The following graph is also generated:

Histogram of x
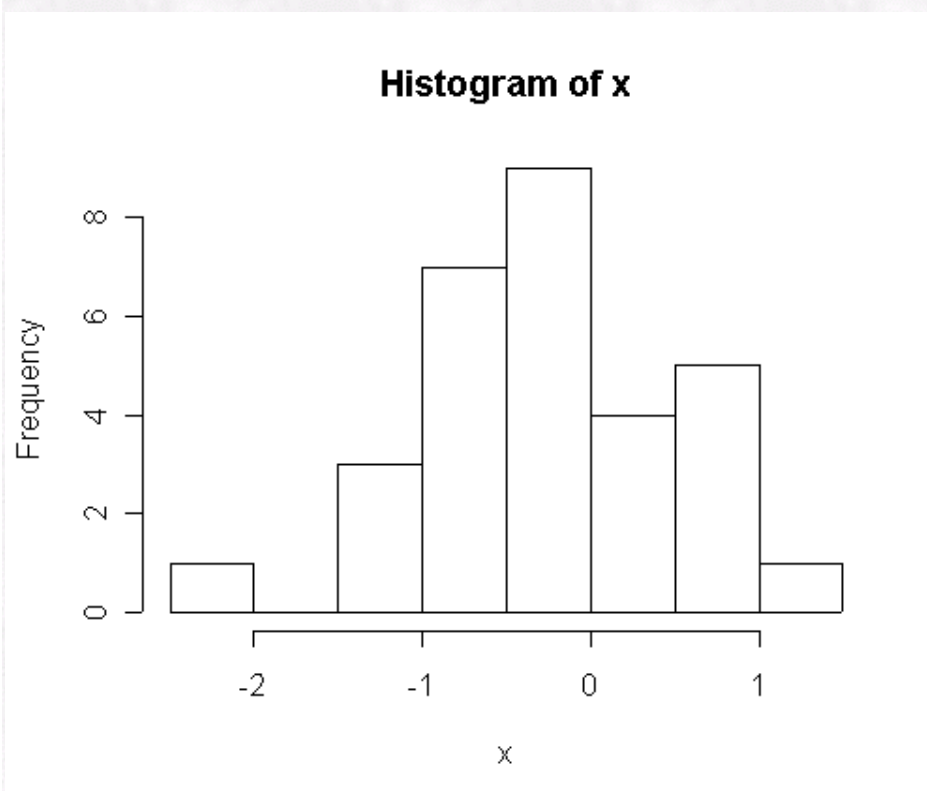
Additionally, using the drop down menu system in Rcmdr generates a script that appears in the script

window. This convenience allows one the benefit of the drop down menu system, but also allows one to learn the R command language set as well. Additionally, commands that are generated by the window system can be saved to a script file for additional processing at a later time. For an example, lets generate a normal density function from the menu system (click "ok" to accept the default normal density parameters):
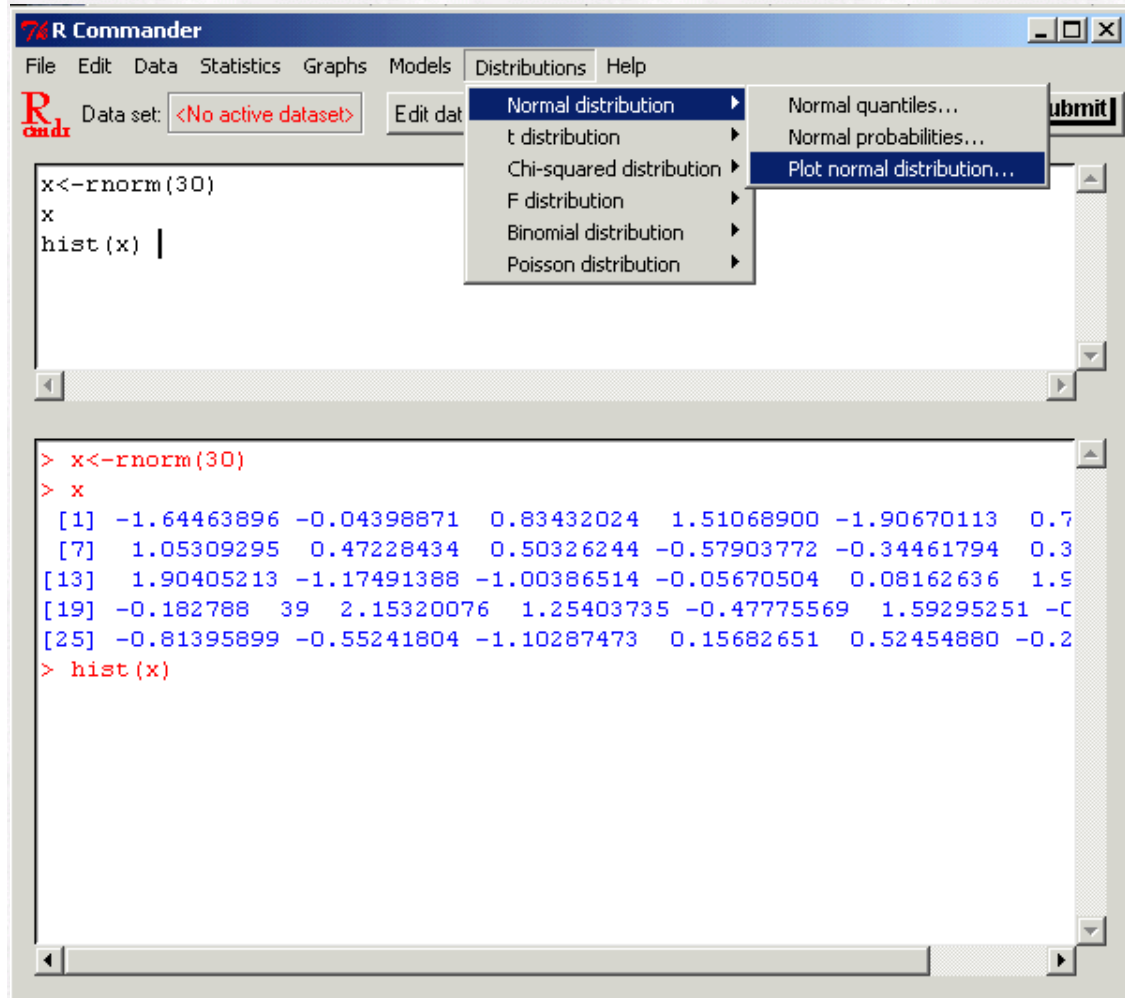
The following R script is generated in your script window; and the following is generated in your output window:

```
%R Commander                                                      _ □ ×
File   Edit   Data   Statistics   Graphs   Models   Distributions   Help

R        Data set: <No active dataset>    Edit data set  View data set    Model: <No active model>        Submit
cmdr

x
hist(x)  .x <- seq(-3.291, 3.291, length=100)
plot(.x, dnorm(.x, mean=0, sd=1), xlab="x", ylab="Density", main=expr
abline(h=0, col="gray")
remove(.x)


> x
 [1] -1.64463896 -0.04398871  0.83432024  1.51068900 -1.90670113  0.7
 [7]  1.05309295  0.47228434  0.50326244 -0.57903772 -0.34461794  0.3
[13]  1.90405213 -1.17491388 -1.00386514 -0.05670504  0.08162636  1.5
[19] -0.182788   39  2.15320076  1.25403735 -0.47775569  1.59295251 -0
[25] -0.81395899 -0.55241804 -1.10287473  0.15682651  0.52454880 -0.2
> hist(x)

> .x <- seq(-3.291, 3.291, length=100)

> plot(.x, dnorm(.x, mean=0, sd=1), xlab="x", ylab="Density", main=ex

> abline(h=0, col="gray")

> remove(.x)
```

Notice that when your graphic image is generated that a history of plots are recorded. Click on **"History"** then **"Recording"**. This should be checked as it is turned on when Rcmdr starts. You can use your "PAGE-UP" and "PAGE-DOWN" keys on the keyboard to flip back and forth through the various plots you have generated.
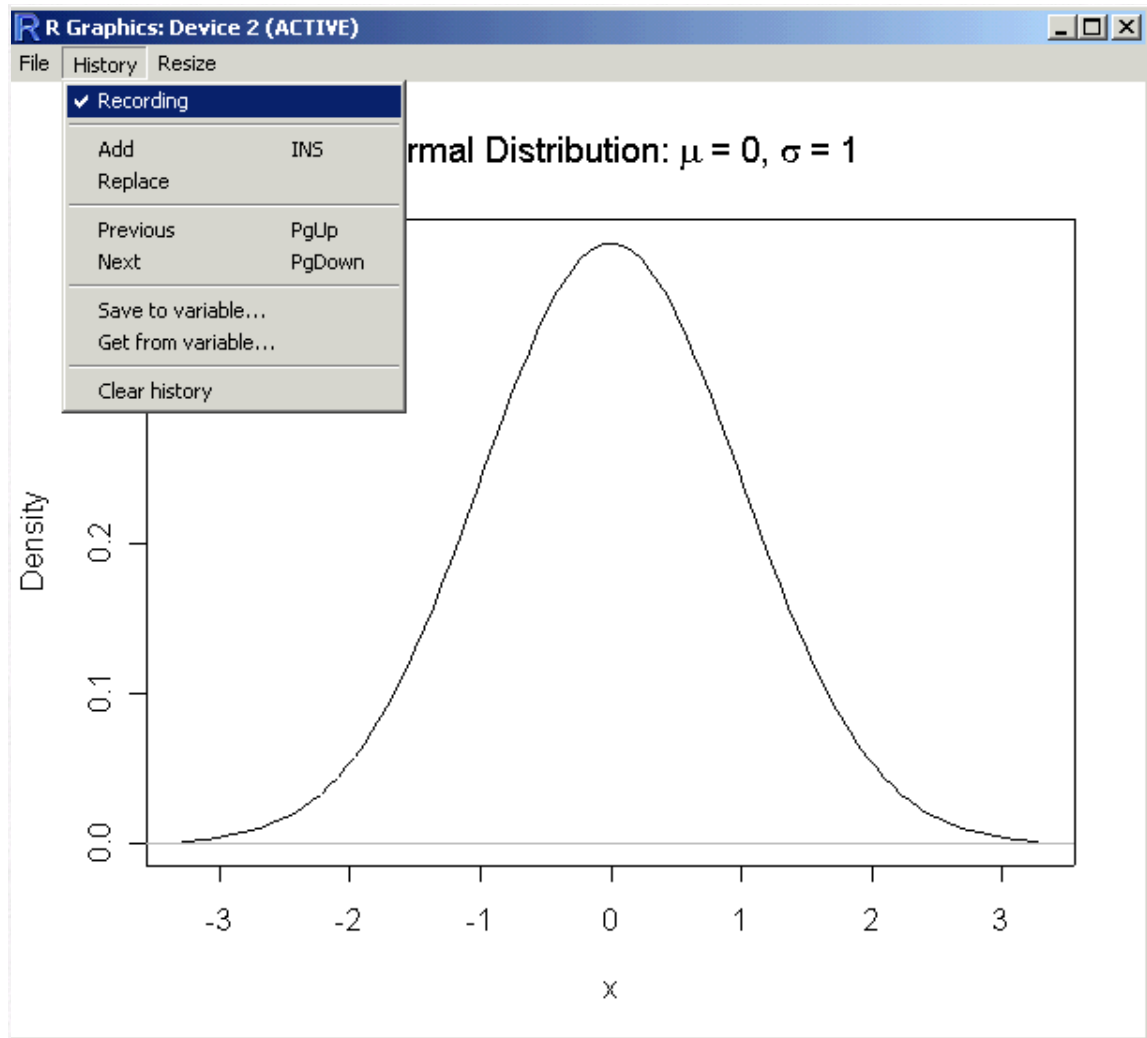
On the **File** menu option of your R graphics device you will notice that you can copy your image to your clipboard for insertion into other Windows programs (i.e. Microsoft Word). Also you can print the image out as well.
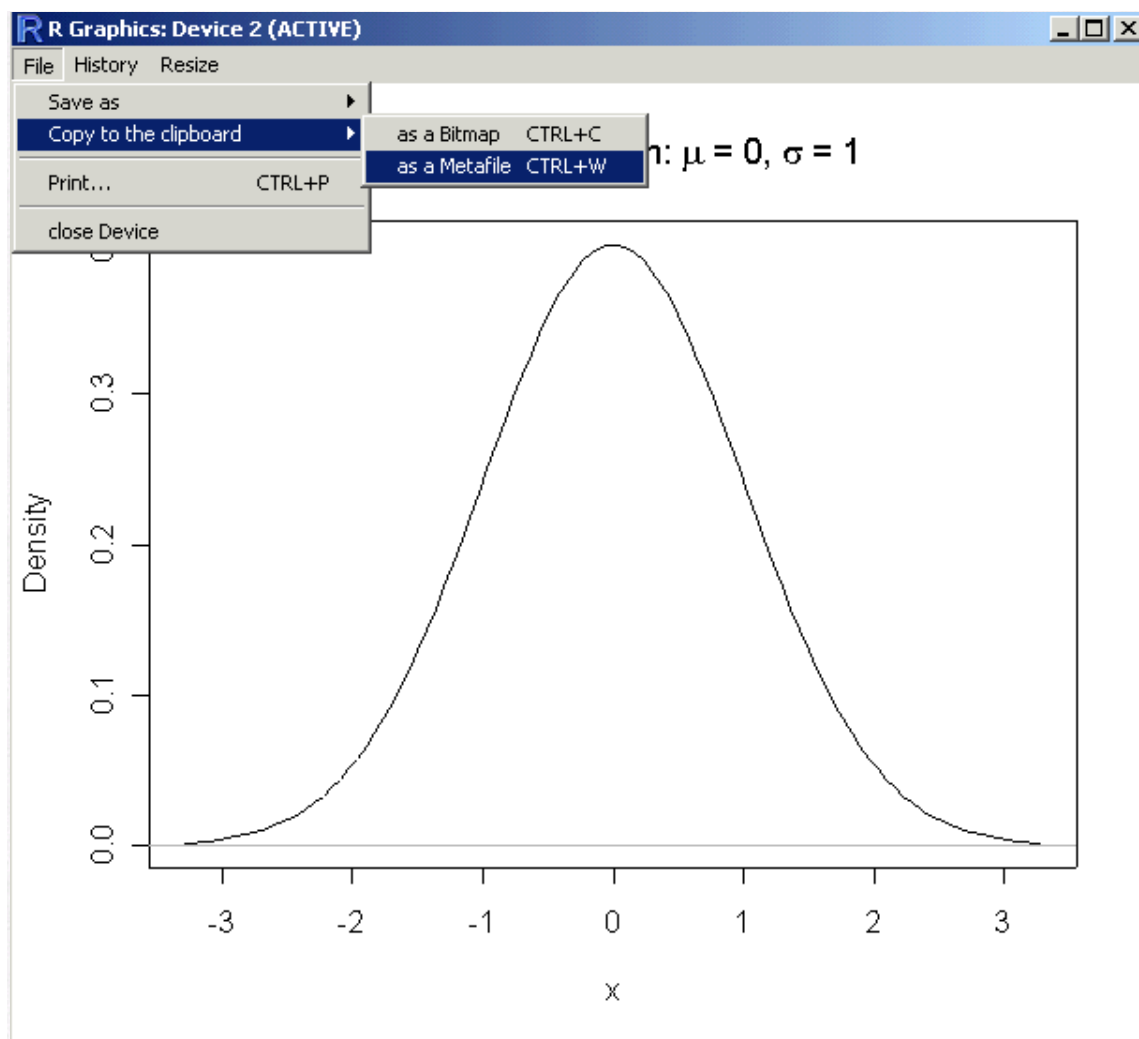
Running R scripts from your Rcmdr window in your Rcgi Web Interface and Vice-Versa

Now we want to Start a web browser at the following link:

http://rss.acs.unt.edu:/cgi-bin/R/Rprog

Paste the R script from your R Commander window into the R Web interface:

Program Input

Clear the R program

```
x<-rnorm(30)
x
hist(x)
.x <- seq(-3.291, 3.291, length=100)
plot(.x, dnorm(.x, mean=0, sd=1), xlab="x", ylab="Density", main=expression
(paste("Normal Distribution: ", mu, " = 0, ", sigma, " = 1")), type="l")
abline(h=0, col="gray")
remove(.x)
```

Run the R Program   Display Graphic (GIF)

The **"Clear the R program"** button will clear the text from the CGI window; The **"Display Graphic (GIF)"** button will display any images that have been generated. Running the R script produces the following text:

Program Output

```
*** Reference 12385:20040616115431


R : Copyright 2004, The R Foundation for Statistical Computing
Version 1.9.0  (2004-04-12), ISBN 3-900051-00-3

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for a HTML browser interface to help.
Type 'q()' to quit R.

> postscript("/tmp/R/out/12385.Rps",horizontal=FALSE)
> x<-rnorm(30)
> x
 [1]  2.208836075 -0.075356146  1.505779711 -0.843985008 -0.015290103
 [6] -1.586712326 -0.445050519 -0.139256333 -1.333271165 -0.154997830
[11]  0.021461790 -0.168944954  0.006298734  0.092021108 -0.002851723
[16] -1.674924965 -0.227965484  0.016842216  2.233093633  0.747183530
[21] -0.274493061 -1.101367206 -0.444823478 -0.298343350 -1.270845969
[26]  0.155130381  0.939933519 -0.346617798  0.525746462 -1.444021370
> hist(x)
> .x <- seq(-3.291, 3.291, length=100)
> plot(.x, dnorm(.x, mean=0, sd=1), xlab="x", ylab="Density", main=expression(pa
> abline(h=0, col="gray")
> remove(.x)
>
>
```

Graphical Output: High Quality PostScript (fast) or Low Quality GIF (slow)

The R script that is processed is produced with the output below the R script. Viewing the graphics can be done by selecting the **"Low Quality GIF"** link at the bottom of the page OR the **"Display Graphic (GIF)** button at the top of the page. A graphic image(s) is generated within the browser:
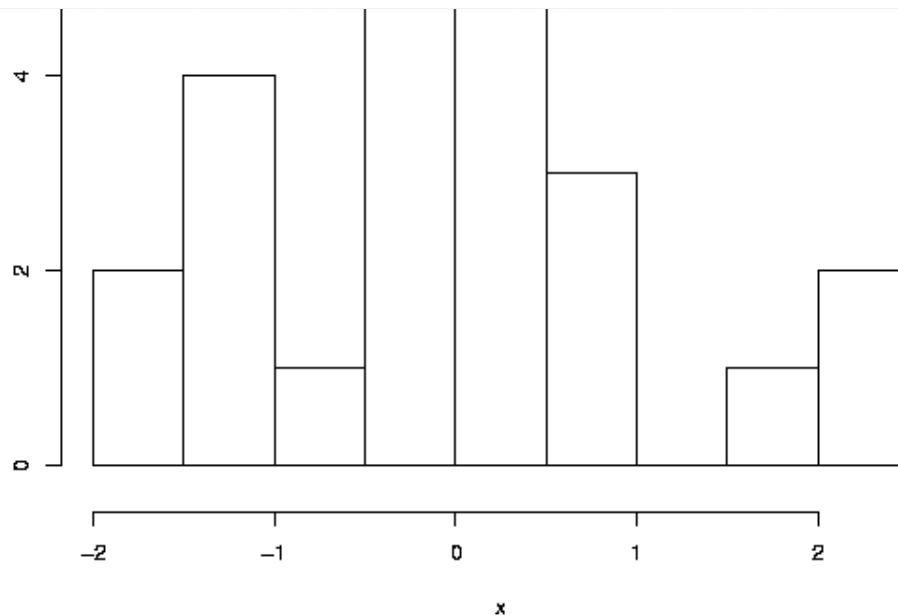
Normal Distribution: μ = 0, σ = 1
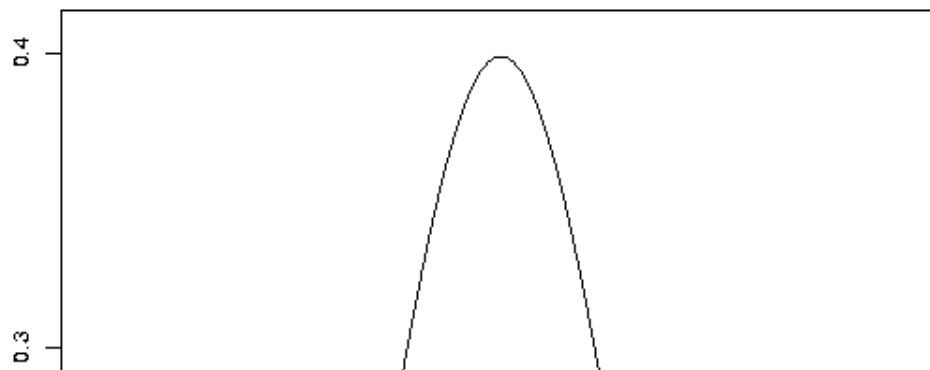
Here we see that the script produces two images. These images can be printed from your browser. Additionally, a right-mouse click on these images will allow you to copy to your clipboard for pasting into other programs (i.e. Microsoft Word), or for saving to a file. **To get back to the script window, press the Back Page on your web browser.** You can continue to click the "Back Page" on your browser to get to the point from which the script window was generated (such as from within an HTML document). **Note: you can right-mouse click on the "Display Graphic" button or the "Low Quality GIF" button and select the "Open in new window" option to open your graphics in a separate window from your script window.**

Creating a Data-frame to Use within R Commander

In R we can create a number of different data objects. We can create **scalars** (single numbers), **vectors** (concatenated scalars - a single column or single row of numbers), or a **matrix** (concatenated vectors - several adjacent columns with multiple rows). Additionally, data can be of **type "character" or "numeric".** Vectors and matrices must be all of one type or the other. For example:

```
R Commander                                                          _ □ ×
File  Edit  Data  Statistics  Graphs  Models  Distributions  Help

R      Data set: <No active dataset>     Edit data set | View data set    Model: <No active model> Submit
cmdr
```

```
x<-3
x
w<-c(3,6,10)
w
y<-matrix(c(3,6,10,12), ncol=2)
y
z<-matrix(c("rich", "bob", "alice", "stewart"), ncol=2)
z
```
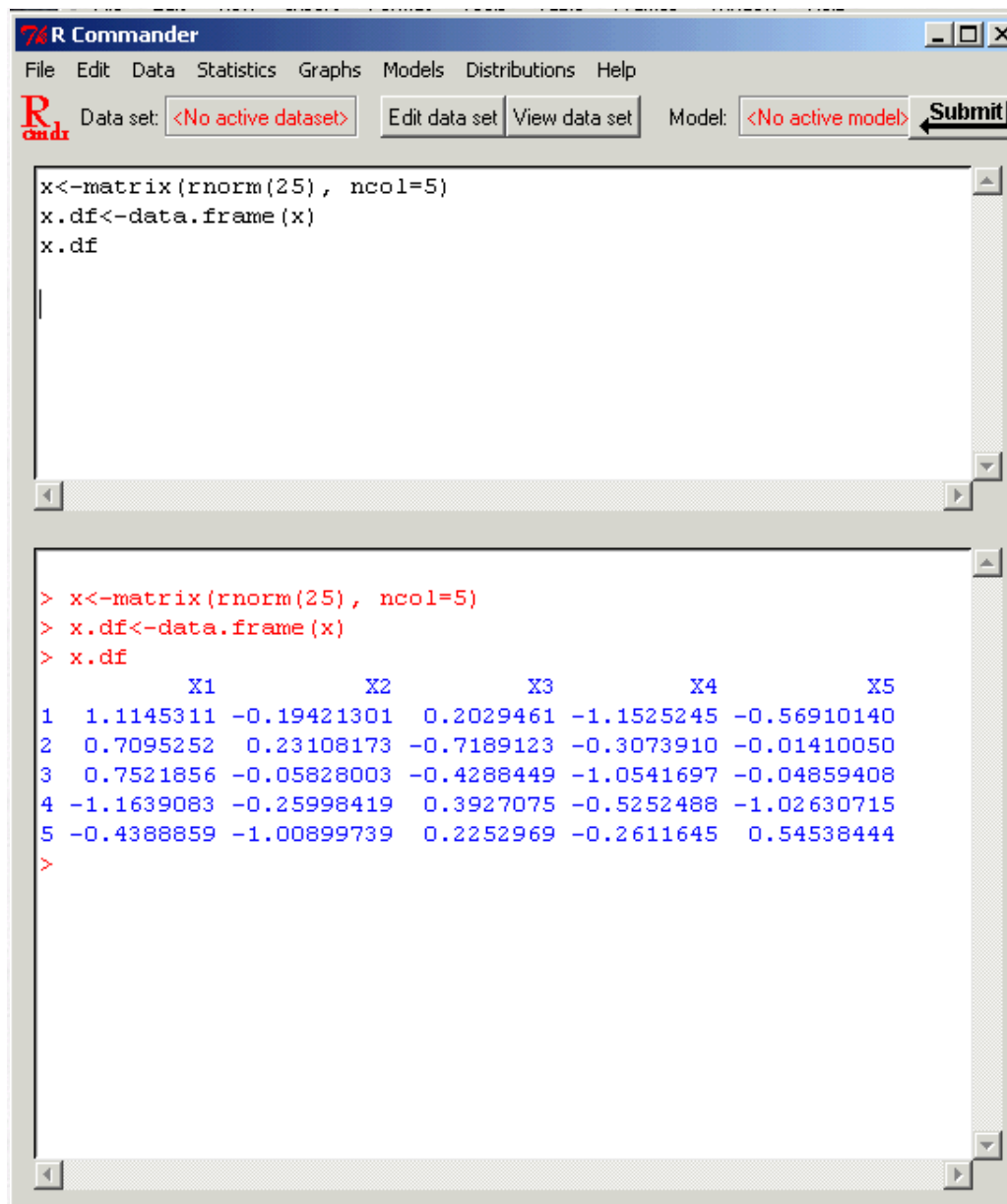
```
> x<-3
> x
[1] 3
> w<-c(3,6,10)
> w
[1]   3   6 10
> y<-matrix(c(3,6,10,12), ncol=2)
> y
     [,1] [,2]
[1,]    3   10
[2,]    6   12
> z<-matrix(c("rich", "bob", "alice", "stewart"), ncol=2)
> z
     [,1]    [,2]
[1,] "rich" "alice"
[2,] "bob"  "stewart"
```

We could not mix character type "rich" and the numeric type "3" in a vector or matrix. We need a **"data-frame"** for this purpose. The data-frame is what is we will refer to a rectangular data matrix where columns are "variables" or measurements on each respondent and rows are the distinct respondents. For example, a column could consist of names, another column might consist of salaries. "Names" would be a character variable and "Salary" would be numeric variable in the same data matrix. R Commander will need all data that will appear in the drop down dialog boxes to be in the form of a data-frame so that variable information can be utilized in the menu system. To create a simulated data-frame we can use the following script:

Notice that the data.frame function generates default variable headings: X1, X2, X3, X4, X5. These would represent variables for respondents. The numbers 1,2,3,4,5 would represent different respondents. Next, notice that the **"Data Set"** icon is grayed out and there is "**<No active dataset>"**. Click the "Data Set" icon button and a drop down menu list will appear. Select the "x.df" object:

Notice that there is now an "Active data set" called x.df.

```
76 R Commander                                          _ □ ×

File   Edit   Data   Statistics   Graphs   Models   Distributions   Help

R      Data set: x.df    Edit data set  View data set   Model:  <No active model>      Submit
cmdr
```

```
x<-matrix(rnorm(25), ncol=5)
x.df<-data.frame(x)
x.df

attach(x.df)
```
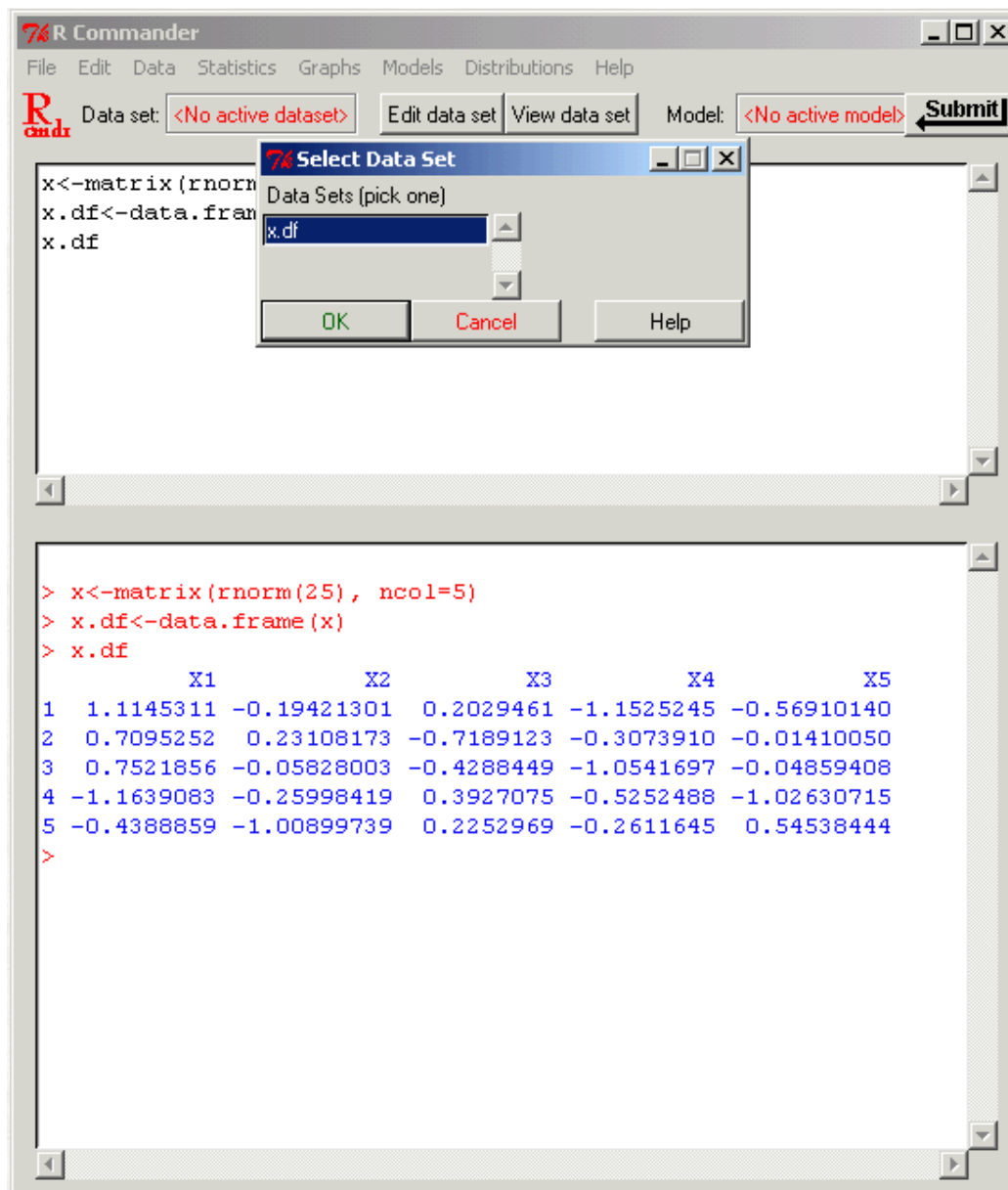
```
> x<-matrix(rnorm(25), ncol=5)
> x.df<-data.frame(x)
> x.df
          X1          X2         X3         X4          X5
1  1.1145311 -0.19421301  0.2029461 -1.1525245 -0.56910140
2  0.7095252  0.23108173 -0.7189123 -0.3073910 -0.01410050
3  0.7521856 -0.05828003 -0.4288449 -1.0541697 -0.04859408
4 -1.1639083 -0.25998419  0.3927075 -0.5252488 -1.02630715
5 -0.4388859 -1.00899739  0.2252969 -0.2611645  0.54538444
>

> attach(x.df)
```

This mechanism allows a user to switch between different data sets for viewing and editing. Simple editing and viewing in a spreadsheet fashion can be accomplished by selecting **"Edit data set" or "View data set"**:

NOTE: Make sure that you close the editing or viewing window before opening or moving to other windows!! There is a limitation of the spreadsheet program in that it doesn't work well with other windows that are concurrently open. For this reason, edit, enter or view your spreadsheet and finish this task before moving on to other window movements or dialog boxes. **Remember, in general you will need to have an active data set in order to use the menu systems - unless you are using the menu system to generate a graph or data (not ACTING on data).**

Using other Menus in R Commander

In general, the variables of the data-frame will appear in the men dialog boxes. Selecting these variables with the mouse will make them active in the dialog box. For example to generate a histogram:

Producing the following density plot (a histogram with a relative frequency on the y-axis):
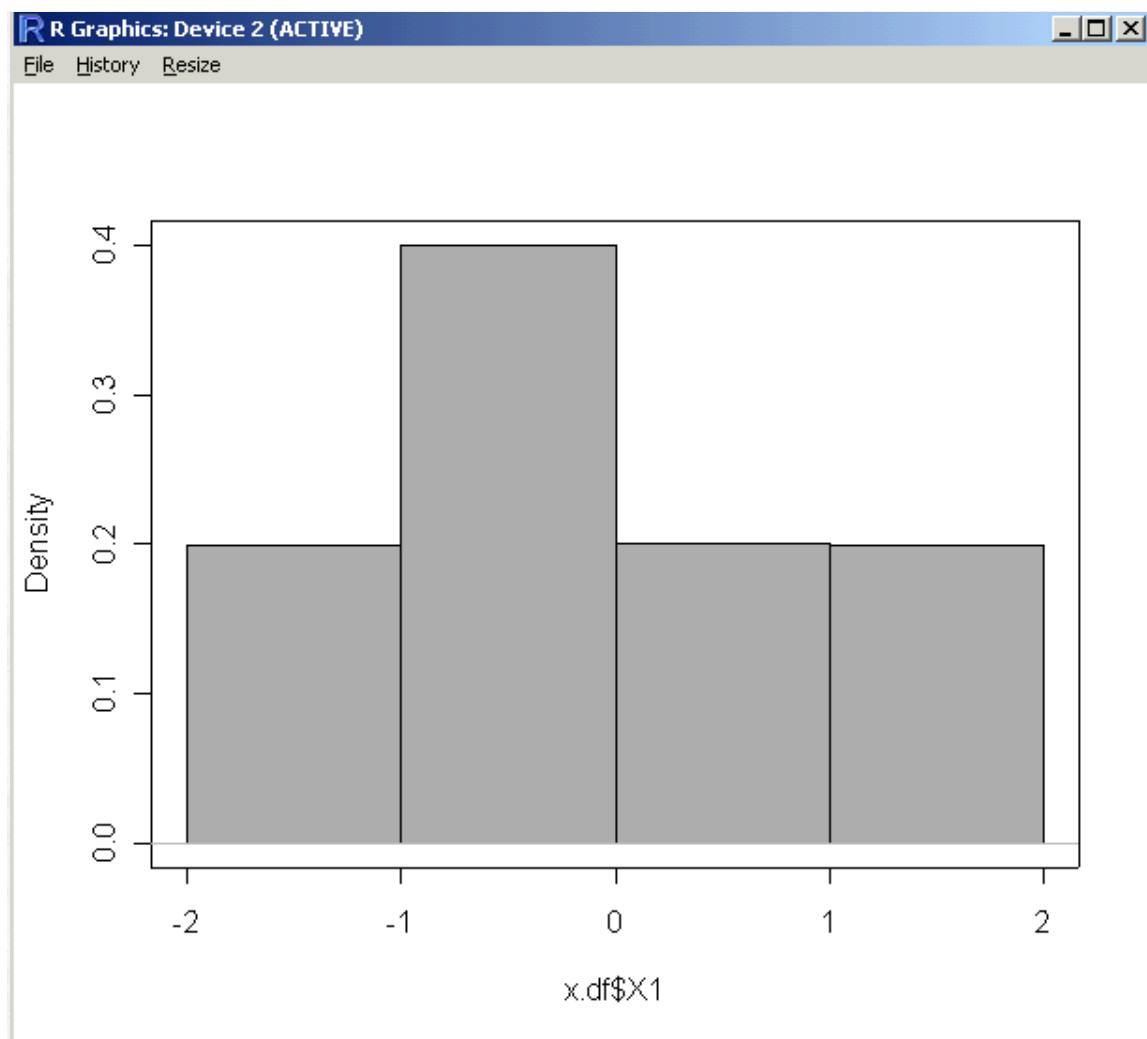
References

Fox, John (2003). The R Commander: A Basic-Statistics GUI for R. http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/

Return to top