

# R-Instat Beta Version 0.4.0 Release Notes

*19th July 2017*

Welcome to the 0.3.1 beta release of R-Instat. This release replaces 0.3.0 as the current beta version. New changes in this version are noted below.

The main aim of this release is to get a group of testers using the software and providing feedback and bug reports to enable us to have a stable first release in July 2017. We will be officially launching R-Instat and the African Data Initiative (ADI) project at the World Statistics Congress in Marrakech, Morocco in the week of 16 July.

If you want to see what our original campaign was all about, take a look at the 3 minute video on [our campaign site](#).

As with the previous beta release, there are still incomplete feature and some bugs. As you explore the software we hope you be excited about it's potential, as well as seeing that we still have more to do!

It is worth keeping in mind the key target audiences for R-Instat. These are described in the original ADI campaign. We claimed this audience need software that is easy to use, free and open source and encourages good statistical practices. R-Instat is designed to exploit the power of the R statistical system, while keeping it easy to use through a dialog front end system.

Some target audiences work in one of the application areas for the tailored menus in R-Instat. The original Instat software included an extra menu for the analysis of climatic data and R-Instat does the same. A second tailored menu in R-Instat is related to the analysis of government procurement data. Both tailored menus have mechanisms to identify specific types of data that their audiences use and dialogs that make it easy to perform specific analyses of that data. We expect more tailored menus could be added in future versions for other areas. If you are in either of these groups, we hope you will start with this general document, before moving to a tailored menu.

We encourage users to also use R through RStudio. Copying the log into an RStudio script window is easily done and can help users manage the scripts used in their analysis. This can also help users to interact directly with the R code. We have had to deviate from 'standard' R code quite considerably to achieve the desired functionality of R-Instat. We have thought carefully about the R packages we use and the R code, including the package names, can be seen in the log. R users might also find this feature useful, for example, to

learn the syntax of the ggplot2 package, which R-Instat's graphics dialogs use.

An accompanying document, available here:

<http://www.africanmathsinitiative.net/blog/initiatives/african-data-initiative/> contains installation instructions. In this document, we focus on using R-Instat, with suggested examples. We also detail the bigger picture of the software and describe the known bugs and upcoming features. Finally, there is a section on how you can give us your feedback and get in contact.

The overall aim of the African Data Initiative project stretches beyond producing this software, but R-Instat is an important first stage for us. Thank you for joining us on our journey to improve statistics for everyone.

The ADI (R-Instat) Team

[R-Instat@AfricanMathsInitiative.net](mailto:R-Instat@AfricanMathsInitiative.net)

## Table of Contents

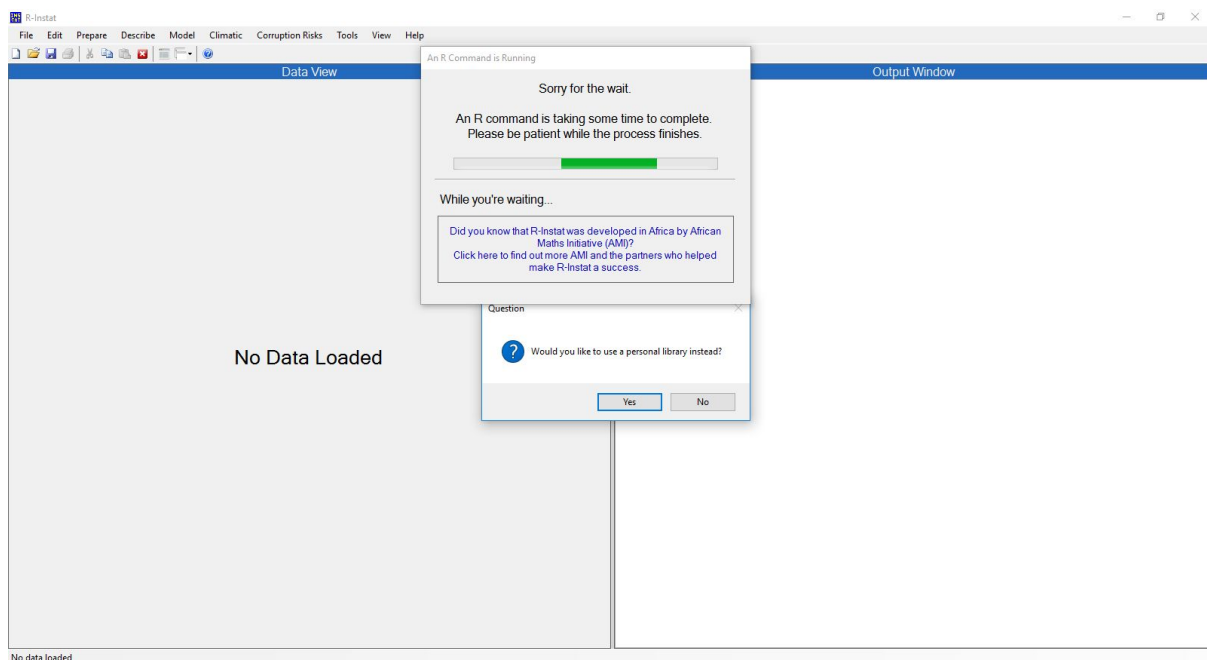
Running R-Instat for the first time	4
Exploring R-Instat	5
1. The Installation.	5
2. A first task - looking at a probability distribution	6
3. Making full use of what you have just done	7
4. Back to R-Instat	7
5. Housekeeping	8
6. Back to the Show Model Dialog	9
7. Saving your data	11
8. A Second Data Set	12
9. Some Graphs	13
10. A more ambitious analysis	14
11. Factors and Labels	15
12. Saving your data again	17
Changes Log	19
An R-Instat Snapshot	19
Known bugs, planned improvements, and upcoming features	21
Feedback and reporting bugs	23

# Running R-Instat for the first time

Once you have installed R-Instat, it is time to open it!

When R-Instat runs for the very first time, if you have not used R before, you may see the message box asking, “*Would you like to use a personal library instead?*”. Click **Yes** for the software to proceed to install the required R packages onto your computer into a folder in your documents.

If you do not see this message box (and no errors appear) then you can assume the R packages installed correctly as it is likely you already had the necessary folder structures.

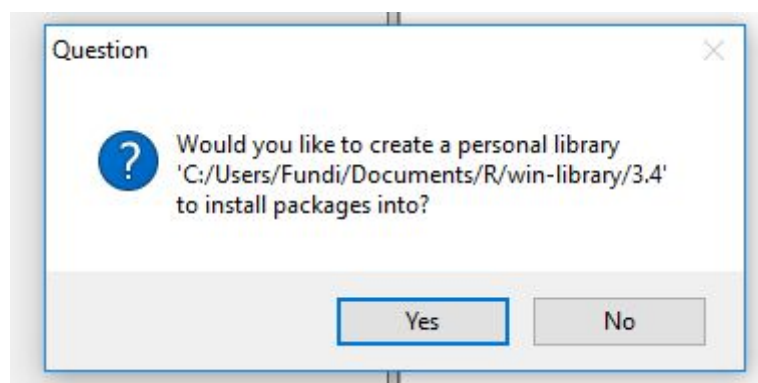


When any command is taking a bit longer to run (in this case installing packages), you will see the “Sorry for the wait” dialog box (below), just to let you know R-Instat is still running and hasn’t frozen. In future, we’d like to include a selection of “Did you know” and “Useful

tips” on rotation for this dialog.



After clicking yes, another message box may appear saying: “*Would you like to create a personal library*”. Please click **Yes** again to allow R to install packages to the specified folder.



The very first time you use R-Instat this may take several minutes, as many packages will be installed. Please be patient! After your first use, this will be much faster as packages will already be installed.

Once the waiting dialog has disappeared you are ready to start using R-Instat!

## Exploring R-Instat

Here is our suggestion of an initial run-through of R-Instat.

### 1. The Installation.

We hope it went smoothly. Please tell us. Currently R-Instat is only for Windows machines.

Mac and Linux users could use a virtual machine. We plan to make R-Instat cross platform in the future.

Once installed and opened you should see the screen that looks like this:

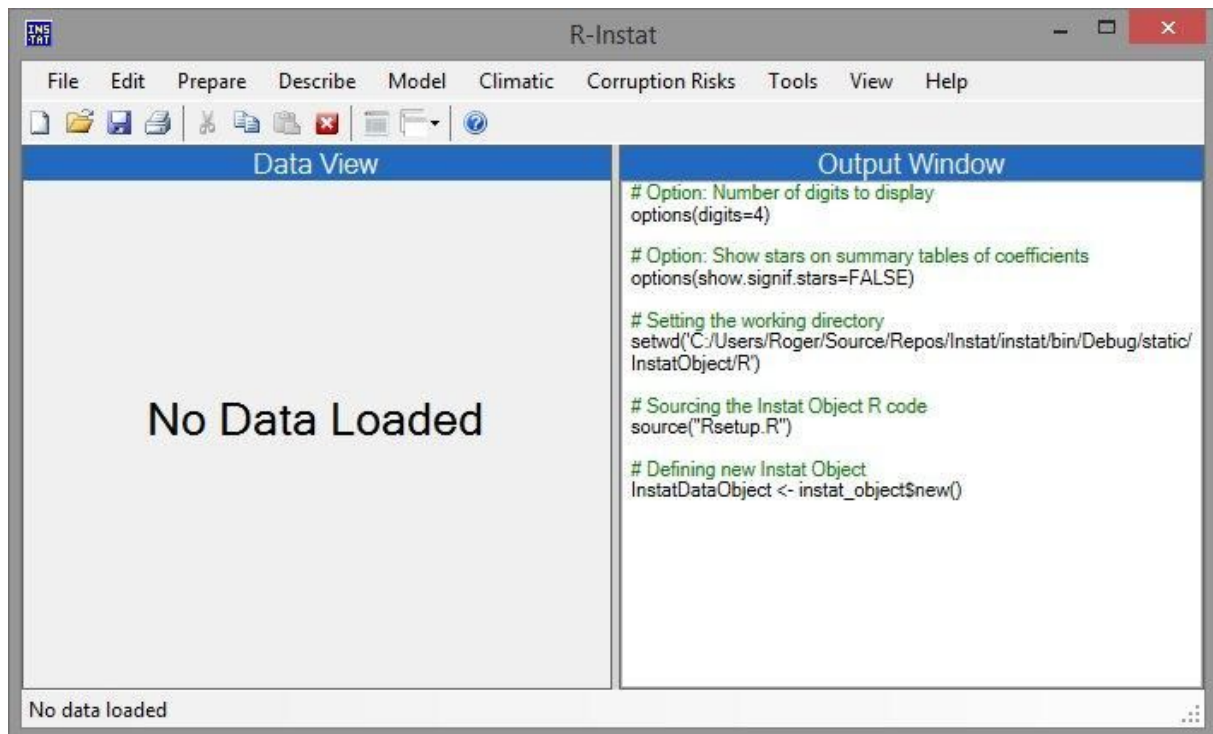
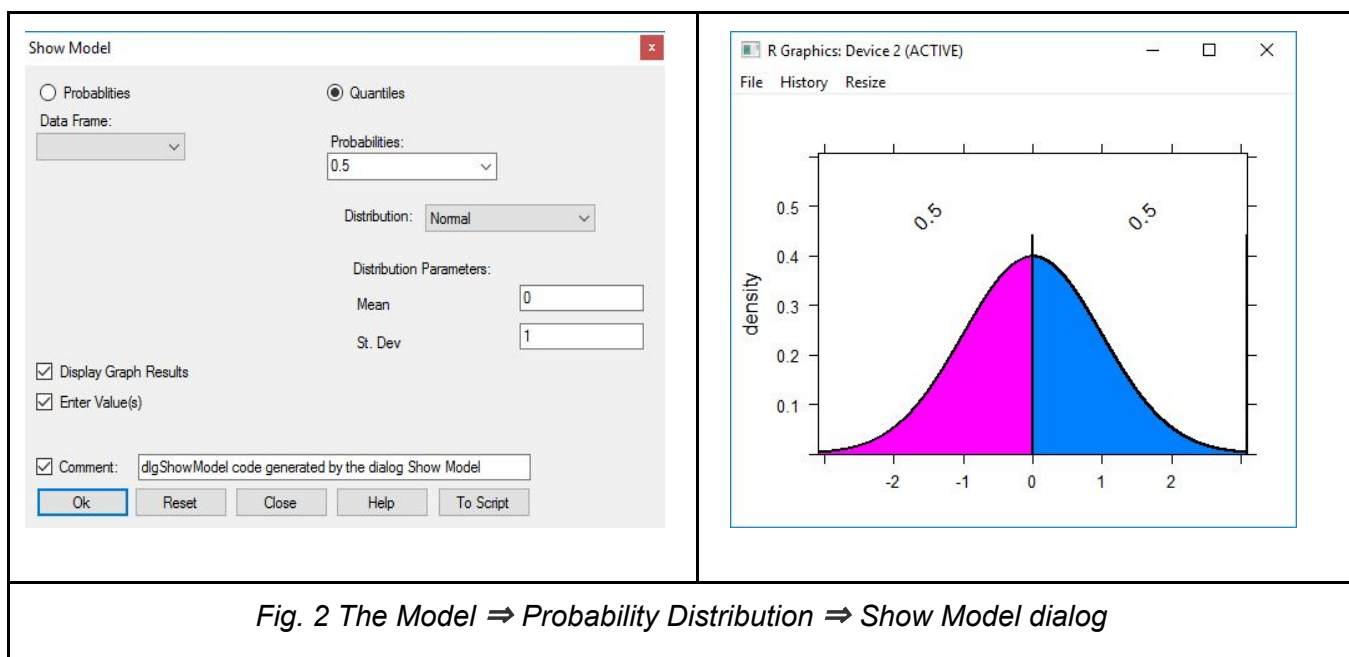


Fig. 1: R-Instat main Interface

## 2. A first task - looking at a probability distribution

Almost everything you do in R-Instat will need some data. But we can start with a dialog that can be used without. Go to **Model** ⇒ **Probability Distribution** ⇒ **Show Model**.

Open the dialog and just press **OK**, or (even more exciting!), use the drop down where it currently says *0.5 to choose something else*, as shown in Fig. 2.

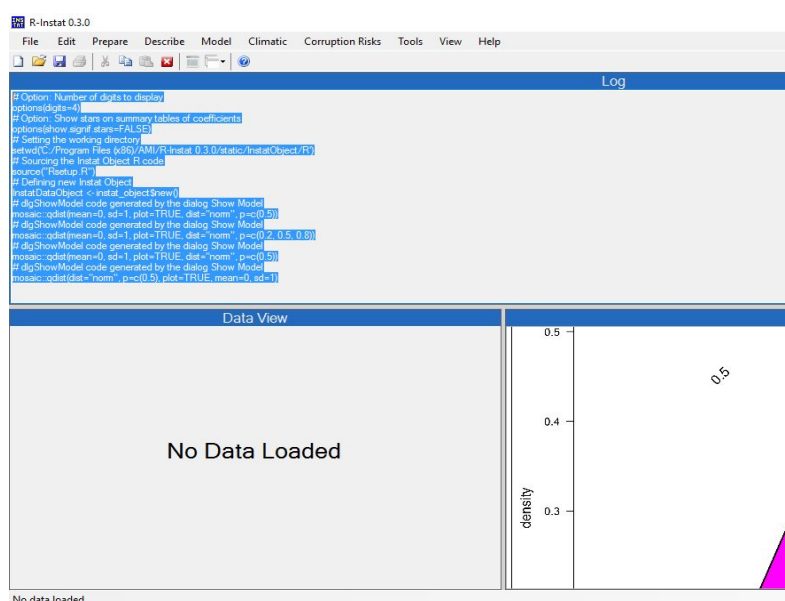


When you press **OK** you will see a graph in the Output Window.

### 3. Making full use of what you have just done

We are aiming for R-Instat to be accessible for beginners to statistics packages - but we assume that, as a tester, you may have some familiarity with R already. If not, then read, but omit this next task.

Go to **View ⇒ Log Window**. Copy the contents from the top to at least the first line that says **mosaic** (as shown in the Fig 3). Then open R (or an R environment like RStudio) and paste the contents. You should get the same graph.



We are keen to encourage more experienced users to consider starting an analysis in R-Instat and finishing in R if they need to.

For you, as testers, we also hope that if (or when!) things go wrong, you could try running the last commands in R, or send us the R code that was generated and didn't do what you wanted.

## 4. Back to R-Instat

Back in R-Instat we continue briefly with this same dialog. First get some data to use, by going to **File ⇒ New** accepting the defaults and clicking OK, Fig. 4. You should now have a small empty data frame as shown below.

Go to **Prepare ⇒ Column: Generate ⇒ Regular Sequence** and generate the sequence from 0.05 to 0.95 in steps of 0.1 as shown in Fig. 4.

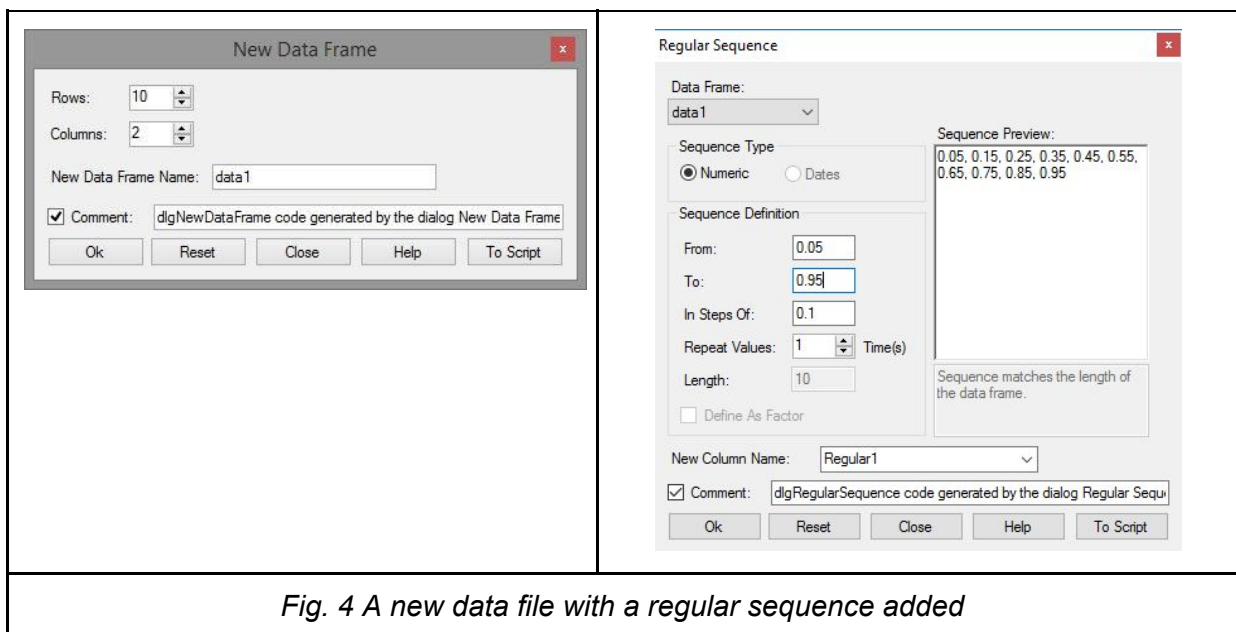


Fig. 4 A new data file with a regular sequence added

## 5. Housekeeping

Before getting another graph, it would be good to tidy the small data frame we have. First the name of the data frame. Go to the bottom of the screen and **Right Click** on the name **data1**. You should get a simple menu as shown in Fig. 5. Click on **Rename** and it opens a dialog. Type the new name as **Test Prob** and press **OK**. It should tell you that names cannot have spaces as shown below. So make it **TestProb** instead and press **OK**.



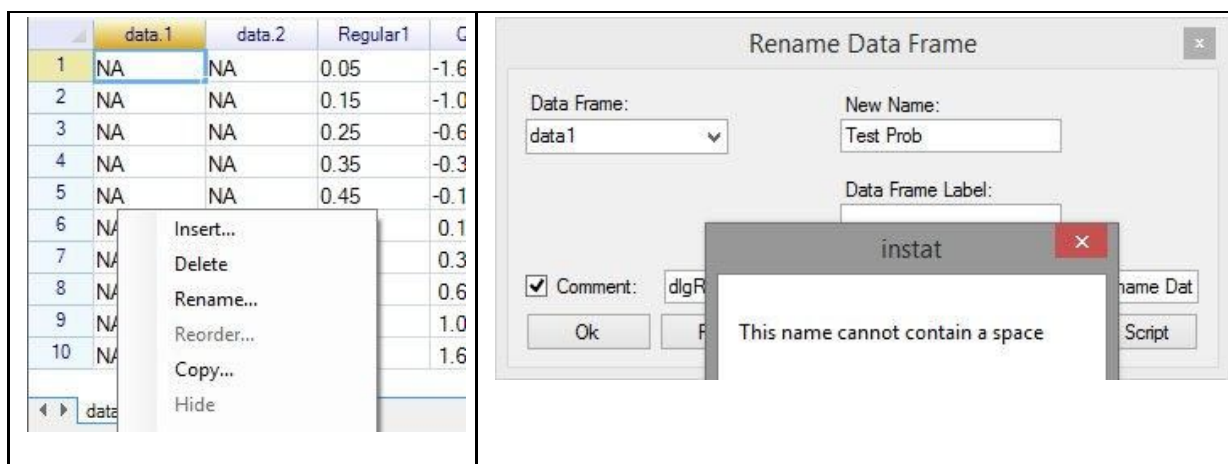


Fig. 5 Renaming a data frame

We also made a small mistake in Fig. 4 above, by generating the data into a new column, rather than overwriting the columns we had already. Now we also get rid of the extra columns. So go to where it says data.1 and data.2. Select them both and **Right-Click** to give the menu shown in Fig. 5a. From this menu accept the option to **Delete Column(s)**.

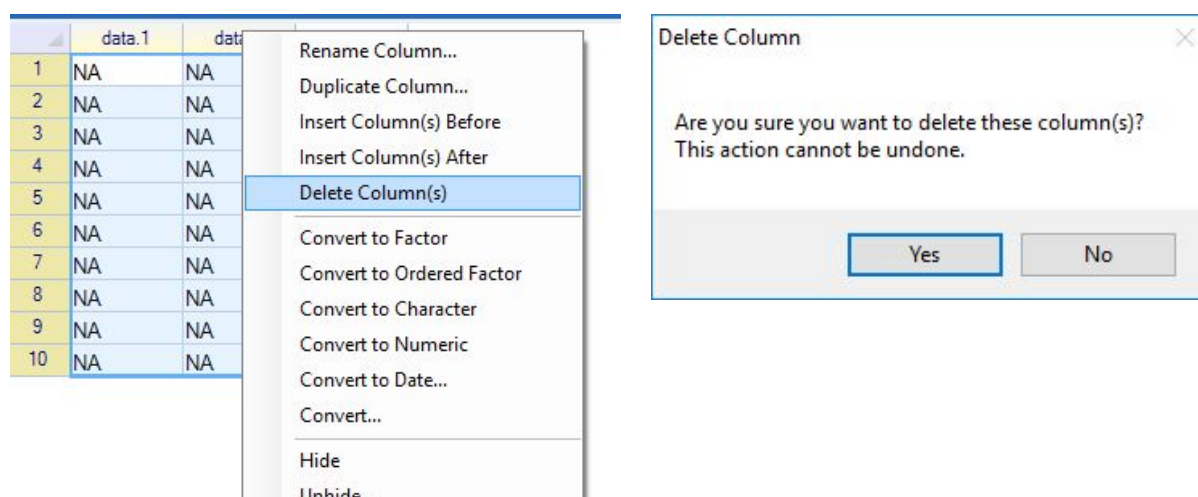


Fig. 5a Deleting unwanted columns

R-Instat checks that you do want to delete these columns - which you do, so click on **Yes**. The message in Fig. 5a that **“This action cannot be undone”** may give you the false impression that **Undo** exists elsewhere. We hope it will, in the future, but it is not there yet.

## 6. Back to the Show Model Dialog

There is an easy way to return to a recent dialog. Go to the toolbar as shown in Fig. 6 and you see the last dialogs that were used.

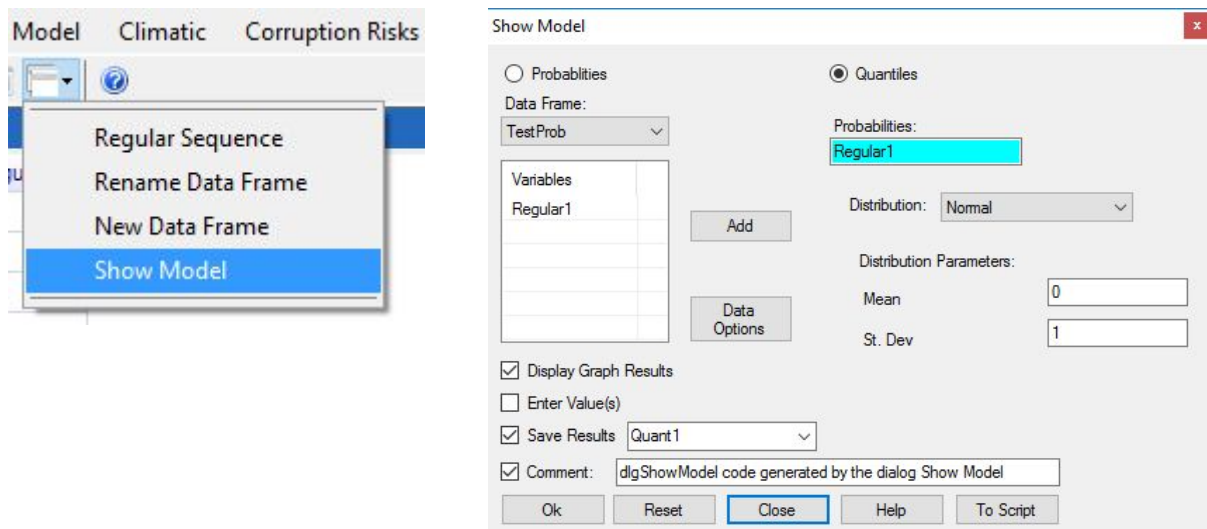


Fig. 6 Reviewing the recent dialogues to return to Show Model

Return to the **Show Model** dialog.

Untick the **Enter Values** checkbox and **tick** the **Save Results** checkbox (see Fig. 6).

Select your **TestProb** data from the drop down in the top left.

Select the column called **Regular1** and click **Add** to put it in the receiver on the right.

Then press **OK**. This generates a new graph, and also put the corresponding deviates into a new column in the data frame, Fig. 6a.

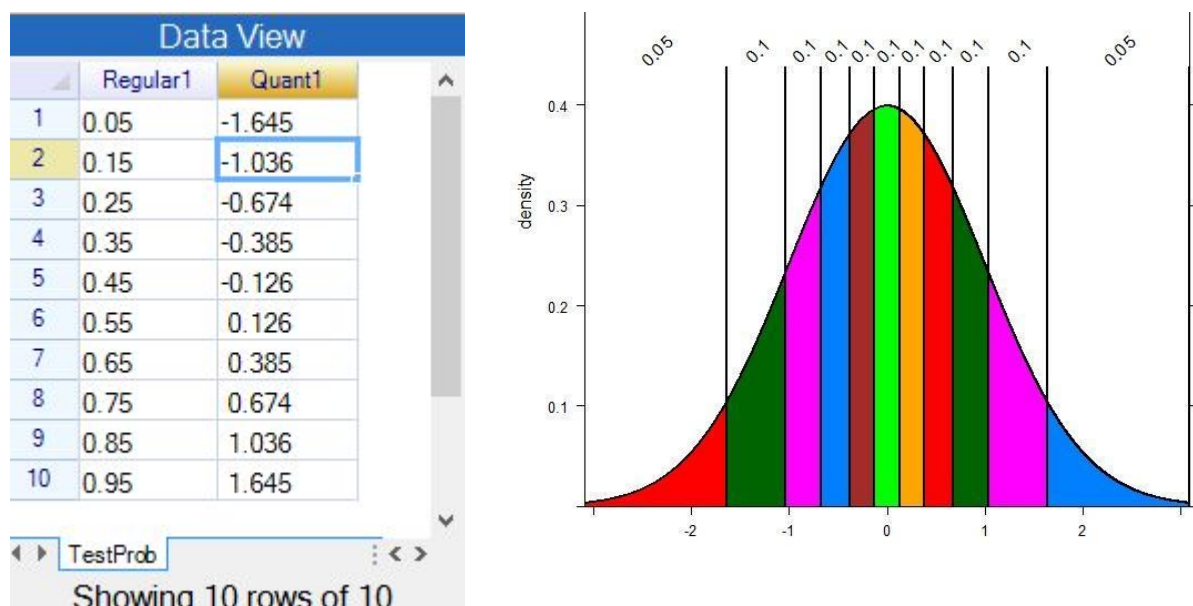


Fig. 6a Results from the second use of the Show Model dialogue

From the data in Fig. 6a, you may wish to have the results in the **Quant1** column with a

different number of decimals. This is a good excuse to show another feature of R-Instat.

Go to the **View Menu**, Fig. 6b and click on **Column Metadata**. Another window will open, Fig. 6b. This window has one row for each column of data, which shows the properties of the columns. One column in the metadata gives the number of significant figures for the data in each column. Change the 3 to 5 for the **Quant1** row and then press enter so it takes effect. Then change it to 2.

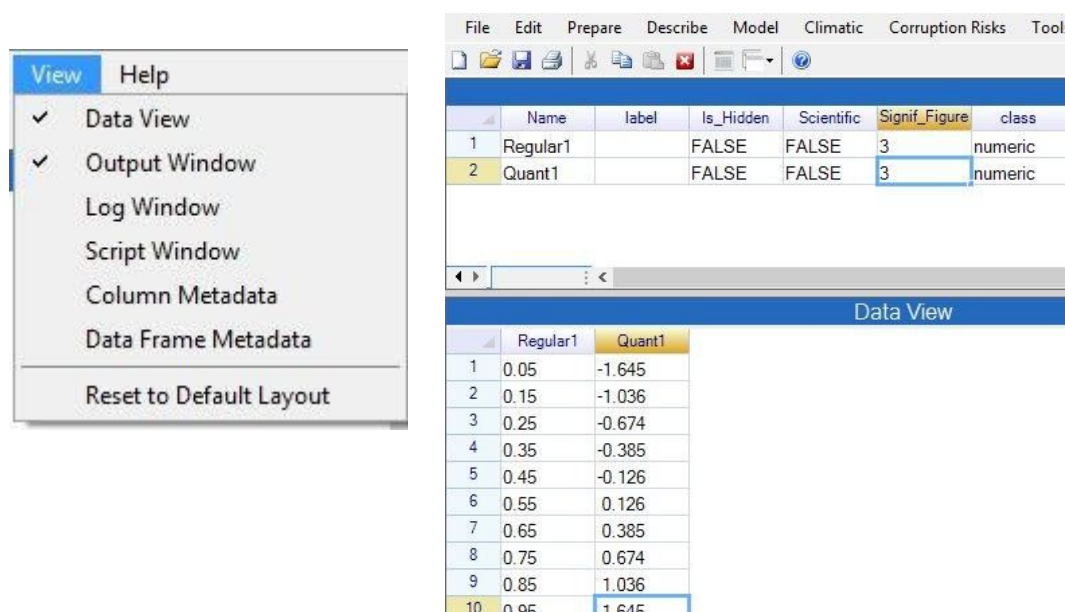


Fig. 6b Column metadata

## 7. Saving your data

Before using a different data set we choose to save these data, so you could resume later. This uses the **File ⇒ Save As** dialogue, Fig. 7.

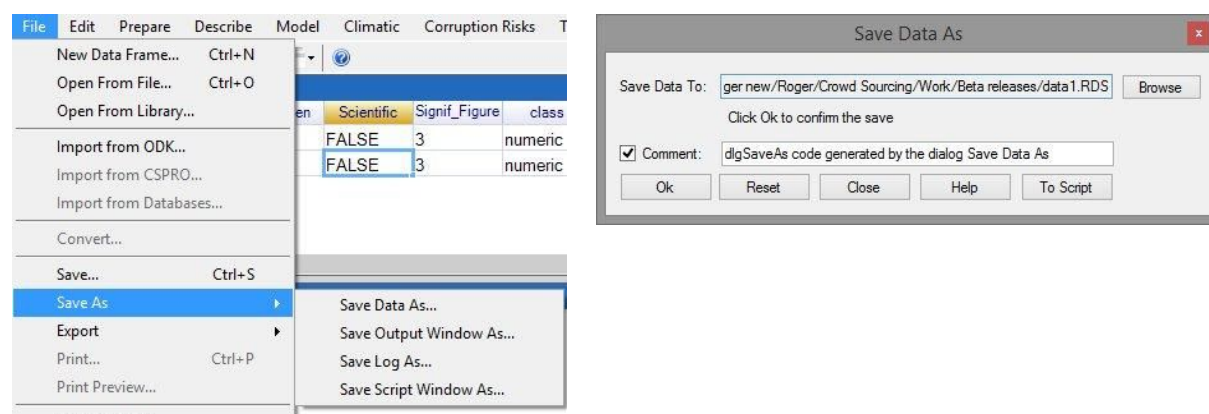


Fig. 7 Saving a data file

Fig. 7 shows you could choose to save the data file, or the log file, or the contents of the

output window. Here choose just to **Save Data As** in Fig. 7 (this include the meta-data).

In the resulting dialogue, also shown in Fig. 7, click on the Browse button and choose where you want to save your data files. Give it the name data1. It will add the RDS extension, to signify it is saved as an R data file.

Then click on OK, Fig. 7 to confirm the saving.

## 8. A Second Data Set

Go to **File ⇒ Open From Library**. Click on the **From Package** dropdown and choose **ggplot2**. Choose the first example, **diamonds** as shown in Fig. 8. You should see that a second **Help** button is now enabled, just below the list of datasets. Click on that button to get further information about the dataset. This help is shown in a window in a browser. (It is the example dataset used by Hadley Wickham, the author of ggplot2, for many of the examples in his own documentation.)

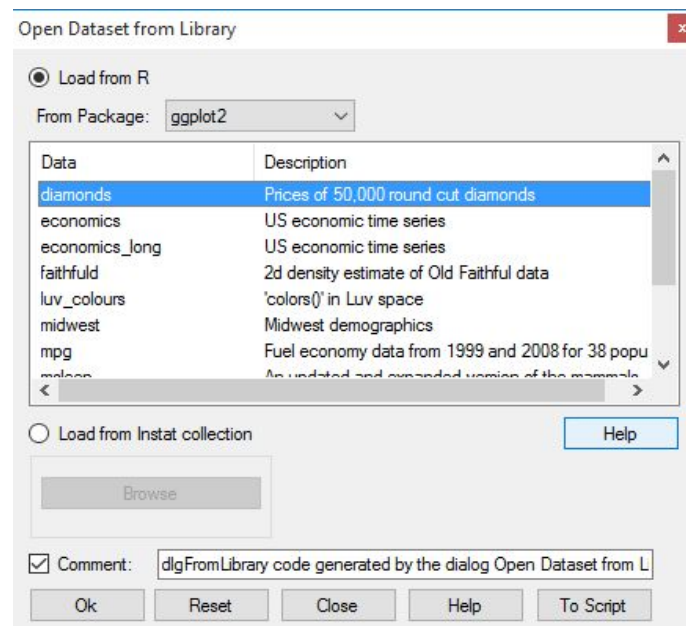


Fig. 8. Using a library dataset

Column Metadata

	Name	label	Is_Hidden	Scientific	Signif_Figure	class
1	carat		FALSE	FALSE	3	numeric
2	cut		FALSE	FALSE	NA	ordered.factor
3	color		FALSE	FALSE	NA	ordered.factor
4	clarity		FALSE	FALSE	NA	ordered.factor
5	depth		FALSE	FALSE	3	numeric
6	table		FALSE	FALSE	3	numeric
7	price		FALSE	FALSE	3	integer
8	x		FALSE	FALSE	3	numeric

Data Frame Metadata

	data_name	label	Is_Calculate	class	Row_Count	Column_Count
1	data1		FALSE	data.frame	10	2
2	diamonds		FALSE	data.frame	53940	10

Data View

	carat	cut (o.f)	color (o.f)	clarity (o.f)	depth	table	price	x	y	z
1	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
2	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
3	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
4	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
5	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75
6	0.24	Very Good	J	VVS2	62.8	57.0	336	3.94	3.96	2.48
7	0.24	Very Good	I	VVS1	62.3	57.0	336	3.95	3.98	2.47
8	0.26	Very Good	H	SI1	61.9	55.0	337	4.07	4.11	2.53
9	0.22	Fair	E	VS2	65.1	61.0	337	3.87	3.78	2.49
10	0.23	Very Good	H	VS1	59.4	61.0	338	4.00	4.05	2.39

Output Window

```
[1] FALSE
attr(,"Signif_Figures")
[1] 3

# Right click menu: Delete Column(s)
InstatDataObject
$remove_columns_in_data(cols=c
("data.1","data.2"), data_name="data1")

# code generated by the dialog
utils::data(X=diamonds,
package="ggplot2")

# dfgFromLibrary code generated by th
dialog Open Dataset from Library
diamonds <- as.data.frame
(x=diamonds)
InstatDataObject$import_data
(data_tables=list(diamonds=diamonds))
```

Showing 1000 rows of 53940

Fig. 8 The diamonds data

Now return to the dialog, *select the **diamonds** dataset again* and press **OK**. The dataset should open as a second data frame in R-Instat. If you scroll to the bottom you will see it appears to have just 1000 rows.

However, now return to the **View** menu, Fig. 6 and choose **Data Frame Metadata** and you will see, Fig. 8 that the data frame has 10 columns and 53940 rows. The data you see is just a window onto part of that data frame which is stored in full in the background.

Fig. 8 also shows the column metadata which provides details on each of the columns in this data frame. In Fig. 8 the windows for both the Data View and the Column metadata are tabbed, so you can easily see the data or the metadata for each of the two data frames.

## 9. Some Graphs

Go to **Describe ⇒ One Variable ⇒ Graph**. **Right-click** in the variables selector and choose the option to **Add All**. (Or you can just select all the columns and then click on **Add**.)



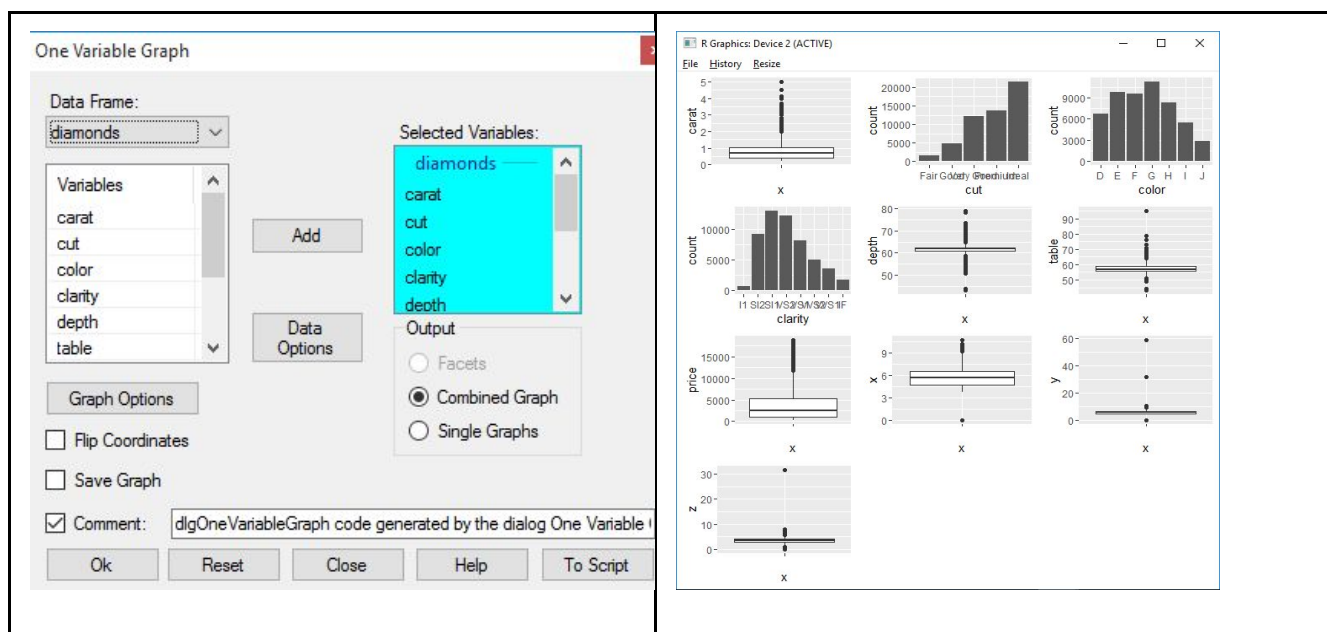


Fig. 9 Using the one-variable graph dialogue

In the dialog in Fig. 9 the radio button changes from **Facets** to **Combine Graph**. That is because the selected variables are of different data types. Some columns are categorical while others are numeric. Press **OK** to give the results also shown in Fig. 9.

If you choose just numeric columns, or just categorical, in the dialogue in Fig. 9, then you can use facets - where the y-scale is usually identical for each graph. You can also use the **Graph Options** button, see Fig. 9, to give different graphs than box plots and bar charts.

Try also **Describe**  $\Rightarrow$  **One Variable**  $\Rightarrow$  **Summarise** to provide numerical summaries that parallel the graphs given above, see Fig. 9a.

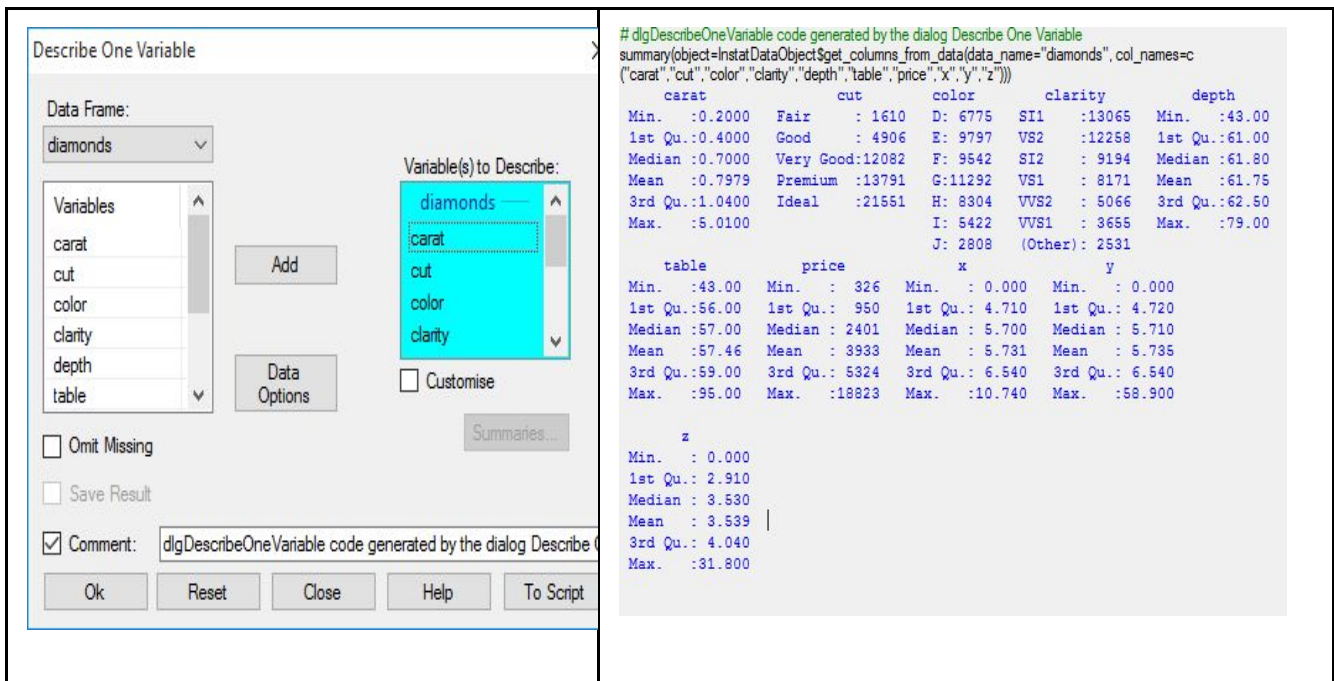


Fig. 9a Numerical summaries for each column of data

## 10. A more ambitious analysis

Go to the **Describe** ⇒ **Multivariate** ⇒ **Correlations** dialog. Use the **diamonds** data frame. Note that only numeric columns are available for this dialog.

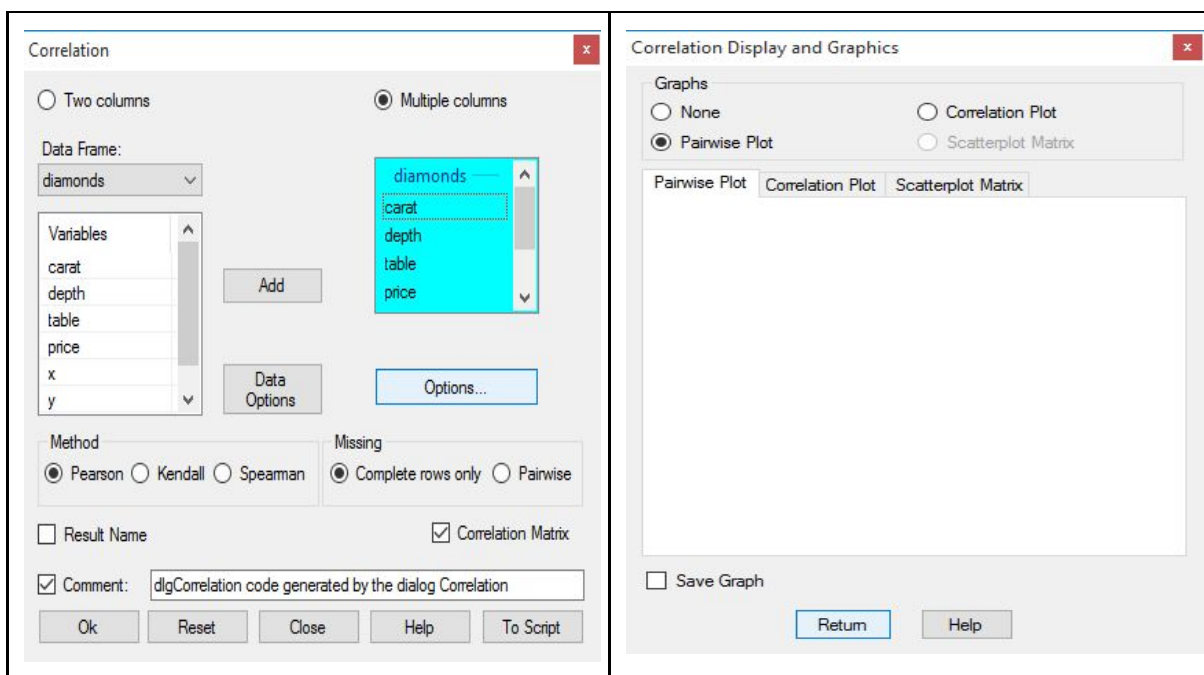


Fig. 10 The Correlations dialogue

Now it depends on the speed of your computer. If it is good, then copy over all the columns,

see Fig. 10. Otherwise just copy 3 more columns over. Then choose the **Options button** and select the **Pairwise Plot**. Then press **Return**.

When you press **OK** you see a graph below.

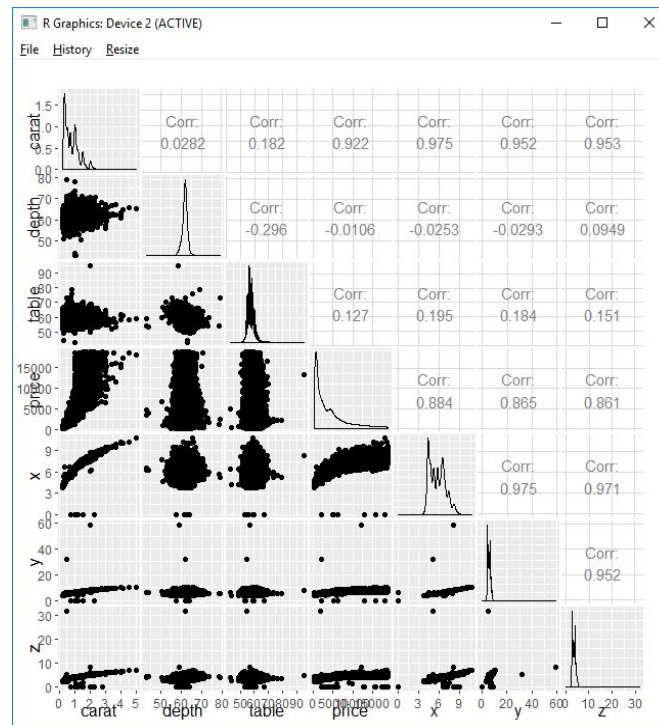


Fig. 10a A pairwise plot – or table?

We would, at this point, like you to reflect on whether Fig. 10 is a graph or a table. The fact that these ideas have merged is one reason we have chosen to distinguish between **Describe** and **Model** in our menus in R-Instat, rather than the more traditional **Graphics** and **Statistics**.

## 11. Factors and Labels

Go back to **File ⇒ Open From Library**. In the **From Package** field change to **sjmisc** there is just one dataset, called **efc**. This is a survey dataset imported from SPSS, Fig. 11.

There are 26 columns in this dataset and the column metadata in Fig. 11 shows that each of the variables also has a **label**. This label is the question to which the data gives the responses.

Another column in the metadata in Fig. 11 shows the **labels** that correspond to the numbers in the data. For example, column **c161sex** is the carer's gender. This is coded 1 and 2 in the data, and Fig. 11 shows that **1** corresponds to **Male**, while **2** corresponds to **Female**.

In R and hence R-Instat, columns like this, containing categorical data, are most often factor



columns.

Column Metadata							
	Name	label	Is_Hidden	Scientific	Signif_Figure	class	labels
10	c86cop5	does caregiving caus	FALSE	FALSE	3	numeric	Never = 1, Sometimes = 2, Often = 3, A
11	c87cop6	does caregiving caus	FALSE	FALSE	3	numeric	Never = 1, Sometimes = 2, Often = 3, A
12	c88cop7	do you feel trapped in	FALSE	FALSE	3	numeric	Never = 1, Sometimes = 2, Often = 3, A
13	c89cop8	do you feel supported	FALSE	FALSE	3	numeric	never = 1, sometimes = 2, often = 3, alv
14	c90cop9	do you feel caregiving	FALSE	FALSE	3	numeric	never = 1, sometimes = 2, often = 3, alv
15	c160age	carer' age	FALSE	FALSE	3	numeric	NA
16	c161sex	carer's gender	FALSE	FALSE	3	numeric	Male = 1, Female = 2
17	c172code	carer's level of educat	FALSE	FALSE	3	numeric	low level of education = 1, intermediate

Data View										
	c84cop3	c85cop4	c86cop5	c87cop6	c88cop7	c89cop8	c90cop9	c160age	c161sex	c172code
1	2	2	1	1	2	3	3	56	2	2
2	3	3	4	1	3	2	2	54	2	2
3	1	4	1	1	1	4	3	80	1	1
4	3	1	1	1	1	2	4	69	1	2
5	1	2	2	2	1	4	4	47	2	2
6	3	3	3	2	2	1	1	56	1	2
7	4	1	1	2	4	1	4	61	2	2
8	2	1	1	1	2	3	3	67	2	2
9	3	2	2	1	3	1	3	59	2	NA
10	1	2	1	1	1	1	3	49	2	2

Fig. 11 A dataset imported from SPSS

In the data put the cursor at the top of the column called **c161sex**. Then **Right click** to get the menu shown earlier in Fig. 5. Click on **Convert to Factor**. You should see that the labels are attached, and an **(f)** appears after the column name, Fig. 11a. For the next column, **level of education**, repeat to make this a factor and also make the following column, **current employment** also a factor.

c90cop9	c160age	c161sex (f)	c172code (f)	c175empl (f)
3	56	Female	intermediate level of education	yes
2	54	Female	intermediate level of education	yes
3	80	Male	low level of education	no
4	69	Male	intermediate level of education	no
4	47	Female	intermediate level of education	no
1	56	Male	intermediate level of education	yes
4	61	Female	intermediate level of education	no
3	67	Female	intermediate level of education	no
3	59	Female	NA	no
3	49	Female	intermediate level of education	no

Showing 908 rows of 908

Prepare Describe Model Climatic Corruption Ris

Data Frame  
Check Data  
Column: Calculate  
Column: Generate  
**Column: Factor**  
Column: Text  
Column: Date  
Column: Reshape  
Keys and Links  
Data Object  
R Objects

Column Meta  
Is\_Hidden  
FALSE  
Convert To Factor...  
Recode Numeric...  
Recode Factor...  
Combine Factors...  
Dummy Variables...  
Levels/Labels...  
View Labels...  
Reorder Levels...  
Reference Level...  
Unused Levels...  
Contrasts...  
Factor Data Frame...

htot	neg_c_7	pos_v_4
12	12	
20	11	
11	13	
10	15	
12	15	

Fig. 11a Factors in R-Instat

To look at these factor columns in more detail use **Prepare ⇒ Column: Factor ⇒ Levels/Labels**, Fig. 11b. First look at the levels of **e172code**, which has 3 levels. Next, change to examine **e175empl**. Click the checkbox called **Include Levels**. This shows that the labelling information from the SPSS-type labels, namely 0 and 1 is still available as part of the factor.

Labels/Levels

Data Frame: efc

Variables  
e16sex  
c172code  
c175empl

Add

Data Options

Factor Selected: c172code

Ord	Label	Freq
1	low level of education	180
2	intermediate level of education	506
3	high level of education	156

Add Level

Add Levels

☒ Comment: dlgLabels code generated by the dialog Labels/Levels

Ok Reset Close Help To Script

Labels/Levels

Data Frame: efc

Variables  
e16sex  
c172code  
c175empl

Add

Data Options

Factor Selected: c175empl

Ord	Label	Level	Freq
1	no	0	518
2	yes	1	384

Add Level

Add Levels

☒ Comment: dlgLabels code generated by the dialog Labels/Levels

Ok Reset Close Help To Script

Fig. 11b Factor levels and labels in R-Instat

Explanation: Label and labels and then level and levels – how confusing!

Many statistics package distinguish between a **variable label** and the **value labels**.

Each column or variable in a data frame has a name, e.g. **c172code** in Fig.11. It may also have a **variable label**, which is **carer's level of education** for this particular variable.

This column has numeric codes 1, 2, 3 and these have **Value labels** shown in Fig. 11b.

In R these columns which contain categorical data are called factors. The number of categories is usually referred to as the number of factor levels, so c172code has 3 levels. We usually want to display the associated value labels, e.g. **low level of education**, rather than 1. But sometimes we may want to know the corresponding numerical values also and (confusingly) these are also called the levels! So, when in Fig. 11b we asked to include the levels of the factor we were asking to look also at the numeric codes.

## 12. Saving your data again

If you have followed this exploration then you now have 3 unconnected data frames in R-Instat. An analysis will often have multiple data frames, but usually they will be connected, e.g. from a multilevel survey, or in situations with primary data together with another data frame containing summary values. Unconnected data would usually be in separate analyses.

The most common situation is to save the current data so the analysis can be resumed on a later occasion.

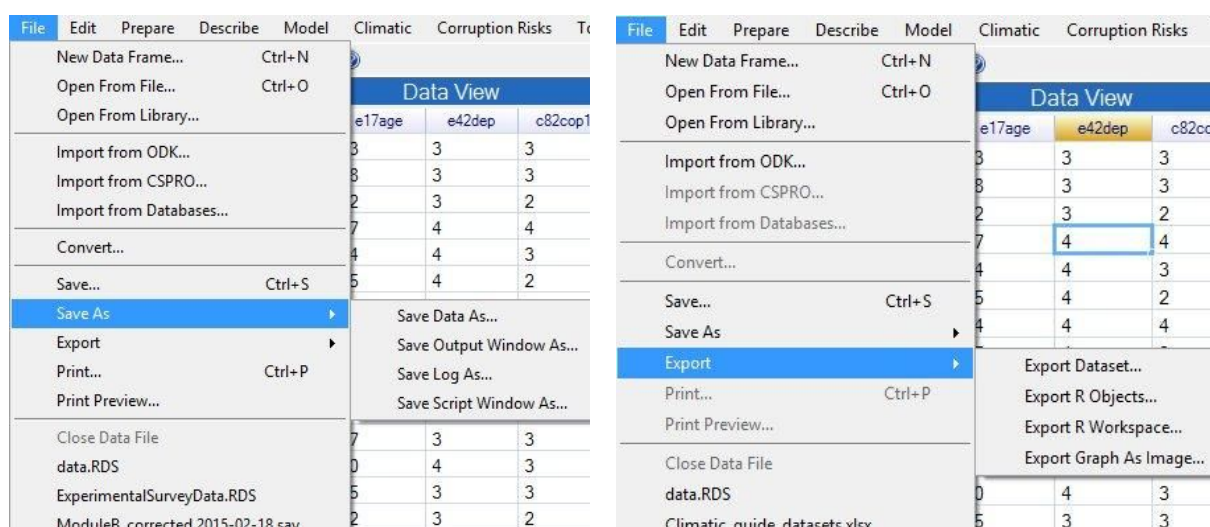


Fig. 12 Saving or exporting data and other aspects of an analysis

Within R-Instat the data frames plus the associated metadata are stored in an "Instat object". If you save graphs or models, then these are added, as further metadata, to their corresponding data frame and will also be saved.

You may also wish to save the contents of the output window, or the log file of R commands, using the menu options shown in Fig. 12.

An alternative is to export data, etc. This option is also shown in Fig. 12. The data from a single data frame can be saved, Fig. 12a, as an RDS file to use in R, or in a wide range of other formats such as Excel, csv, SPSS or Stata.

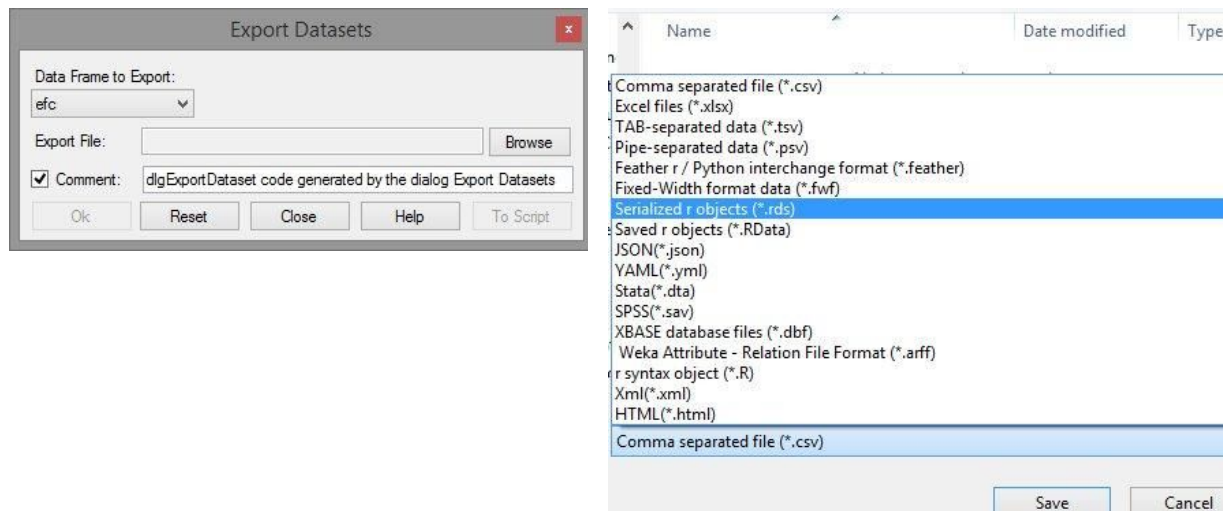


Fig. 12a Exporting data

## Changes Log

### Changes since version 0.3.0

1. Improved the efficiency in refreshing the data grid. There should be a noticeable speed improvement when working with large data sets.
2. Added a “maximum columns to display” option (parallel of max rows) which has reduced the chances of a major crash with wide data sets when set to 30 (the default) or below. The number of columns being displayed is now indicated at bottom of the data grid.
3. Implemented auto recovery options. A backup of the data is saved every five minutes in idle time. The log file is backed up after running every command. The internal log (i.e. including internal R code) is backed up after running every (internal) command. When R-Instat crashes unexpectedly these files may be retrievable through a new dialog that appears when restarting after a crash. Options include opening the backup data and re-running the backup log file.
4. A message appears when an R command has been running for more the two seconds to notify users that R-Instat is still running.
5. An initial implementation of some climatic dialogs (start of rains, spells) is in place.
6. Additional settings have been added to the Options dialog.
7. Import from file dialog now only displays preview for csv and xls/xlsx files (since these are the only file types rio can import with a row limit). Some general bug fixes on this dialog and it can now import text based .DAT files (h/t Savitri Wilson)
8. The Stack dialog can now stack multiple sets of columns.
9. The Merge dialog is simplified (additional options are on the sub dialog) and it can now merge a subset of the columns of each data frame
10. Simplified the design of the Open from Library dialog and added packages

11. In general, sub dialogs open on the first tab after the main dialog has been reset
12. Dialog specific headers for selector box e.g. “factors” or “graphs” instead of always “variables” is implemented on some dialogs
13. Small bug fixes on: One variable compare models, One/two/three variable frequencies, Regular sequence, Delete rows/columns, Levels/Labels dialog

### Technical changes

14. Calls to R are now run on a separate thread to prevent the software hanging
15. RSyntax now contains a list of before and after codes to run and conditions have been added for core controls to check RSyntax
16. Backwards compatibility for options file from older beta version of R-Instat
17. To Script button now obtains correct code (when all functions contained in RSyntax)

## An R-Instat Snapshot

We provide a short description of where we are, menu by menu, and where we expect it to be in later versions.

- a) **File Menu** You should be able to import data from a wide range of software, thanks to the **rio** R package, which is included for import and export. As shown above, we have now resolved how to import value labels from SPSS and Stata.
- b) **Different files.** We are reasonably happy with the way long datasets are handled. In previous versions we had faced speed issues and serious timeout errors which crashed R-Instat when using wide data sets. This version sees a substantial speed improvement for wide data sets and the introduction of a maximum columns to display has significantly reduced the occurrence of timeout crashes.
- c) **Spreadsheet Operations.** We are fairly happy with the way the spreadsheet is working, including right click menus for users familiar with spreadsheet packages. The spreadsheet is just a window onto your data, the default is to display only the first 1000 rows and 30 columns. The speed of refreshing the grid has significantly improved in this version. This can be changed in the options. We would like to add **paste** facilities in the spreadsheet for future versions.
- d) **Prepare menu** This is reasonably complete. There are good facilities for calculating new columns, for dealing with numeric and factors and also some for manipulating text and date columns. Merging and stacking facilities have been enhanced for this version. The dialogs for filtering, appending and unstacking data are also there, but may be enhanced further. We can now cope with labelled data (from SPSS or Stata, as described above), but have yet to add facilities for multiple missing value codes. This will be in the first full release.
- e) **Prepare menu continued** We also have dialogs to manage keys and links between data

frames. We also have dialogs to manage R objects, e.g. graphs, models, and filters.

- f) **Describe menu.** You can summarise data quite well, and we now have some dialogs for tables with html output. We have spent a long time on graphics and have recently managed to implement facilities for the powerful **themes** component of ggplot2. This is still to be improved further as will other ggplot features, particularly adding to our list of available geoms.
- g) **Model menu.** There is still a lot to be added here. We are happy with the structure of the menu, but improvements are needed for the output from the fitting, the range of models that can be processed, and also the facilities for choosing and using the models.
- h) **Climatic menu.** A lot of work has been done on general facilities which relate to climatic data, such as handling dates and calculating and summarising columns. We have begun implementing tailored products, such as the inventory plot dialog, start of the rains and spells and many more are almost complete, such as extremes.
- i) **Corruption Risks menu.** This is for the analysis of procurement data to study corruption risks. It is currently tailored to the procurement data available in the R-Instat library. An expansion of these facilities will be available in future versions.
- j) **Help menu.** There is a lot to do here to have good help facilities. We will also add tooltips in dialogs. We have started to produce series of short videos to explain a range of aspects concerning the use of R-Instat which will be available to users.

Now you are "on your own"! We hope you will enjoy playing with R-Instat and look forward to your reflections and feedback.

## Known bugs, planned improvements, and upcoming features

Below is organised list of known bugs, planned improvements, and upcoming features. This is fairly comprehensive, but may not be complete. We hope this will help you to understand where we see the current state of the software and also where we think it's going as we get towards the first full release.

### *General*

1. The software only runs on Windows 7 and later. We are investigating whether it can be adapted for older Windows versions and eventually we want to implement it across operating systems, initially prioritising Linux. We will soon provide installation manual for these.

2. The software installs (offline) a large number of R packages. However, now only a handful of core packages are loaded on start-up, making start-up much faster. Our own R code will eventually become an R package, but currently it is in .R files that are sourced on opening R-Instat.
3. Tool tips will be added to almost all dialogs, which we see as an important component of the help.

### *File and Edit Menu, and Options (Tool Menu)*

4. Translations are not yet implemented and are disabled in the *Options* dialog but will be available in several languages in future versions. The first additional languages will be French and Spanish. This was an important part of our original campaign, and the structures are already in place to implement this.
5. Some options in *Tools > Options* may have no effect. These are known bugs. In general, this dialog is incomplete and will be redesigned.
6. The *Import* and *Export* dialogs use the *rio* R package. *Rio* uses standard R packages for importing/exporting but there is no guarantee it can always successfully read and write files. This is very dependent on the file. Importing from csv is currently the most robust file type. Additional options on this dialog will appear in future versions to accommodate variations in files where possible. Please send us files that do not import correctly.
7. The most recently used files list at the bottom of the file menu does not always display the correct recent files. This will be corrected in the next release.

### *Output and Display (View Menu)*

8. Messages and warnings from R are not displayed in the output window. We are investigating the possibility of including these in future versions.
9. HTML tables produced from table dialogs sometimes display in a web browser instead of/as well as the output window. We are investigating this issue.
10. Options to turn on/off different aspects of the output window will be added in *Options* e.g. turn off the display of R commands.
11. We are investigating alternatives to displaying the “no data loaded” message on open. A possible option is showing a 1 x 1 grid that automatically expands as data is entered.

### *Prepare Menu*

12. When a filter is applied, dialogs that add columns ignore the filter, and add for the



whole column. In future versions, we will have other options such as missing values outside the filtered rows.

13. Freezing columns is disabled until fully implemented.
14. The *Regular Sequence* dialog is known to be slow to load the first time, especially with long data sets, due to the preview window. This will be improved.
15. Row names are not preserved by some operations in the *Prepare* menu such as sorting. This is due to **dplyr** functions not preserving row names. Users are suggested to put unique identifiers in one of their own columns and not rely on row names. The row names dialog can assist with this.
16. The *Merge* dialog sometimes cannot join by a numeric column paired with an integer column. This is a known bug with the *dplyr* R package. As a work-around, integer columns can be converted to numeric first, using the *Convert* dialog, to enable the join.
17. An error message is given by the Calculate or Enter dialogues, when the vector to be added to the current data frame is shorter or longer than the number of rows of the data frame. This is to be corrected in the subsequent release.

## *Describe Menu and Graphics*

18. Scrolling the mouse wheel on graphs displayed in a separate window causes a major crash of the software.
19. Producing a graph through the general graphics dialog that is not valid can crash the software. This is caused by the link to R failing to display the graph. We are investigating how to handle these errors to prevent a full crash.
20. In future, the *Combine Graphs* dialog will allow graphs from different data frames to be combined.
21. Adding layers to a graph through Plot Options on Specific graph dialogs is not yet implemented.
22. The general graphics dialog contains relatively few geoms (layers) currently. More will be added soon. The general plot options dialog also has more options to be added e.g. coordinate options.

## *Model Menu*

23. In general, there are more dialogs to be added here. The current dialogs will also be improved for the next release.

## **Feedback and reporting bugs**

Thank you for taking the time to test R-Instat, we hope you enjoyed the experience! This is still a testing (beta) version and we are actively seeking feedback from testers to help us



improve R-Instat because a full launch in July 2017. We welcome any feedback, comments, questions, bug reports, criticisms and suggestions you may have about this software – small and large. There are several ways you can provide your feedback:

1. R-Instat is an open source software and the code is managed on [GitHub](#). Our [issues page](#) is a natural way for testers to report specific bugs or suggestions, as all developers will see this. It can access here: <https://github.com/africanmathsinitiative/Instat/issues>.

Click the green **New Issue** button on the right side to send your message. When reporting a bug, it's most helpful to us if you can be as specific as possible and detail how to reproduce the bug, pasting the R code and attaching data if possible. Note that our issues page is publicly visible.

2. For more general feedback, comments and questions, you may wish to contact us via email at [R-Instat@AfricanMathsInitiative.net](mailto:R-Instat@AfricanMathsInitiative.net).

R-Instat Team, African Data Initiative