



BioStat Prime

Table of contents

Introduction	5
Installation	7
Licensing Models	13
How to use BioStat Prime	14
File	25
Dataset	35
Aggregate	36
Compare Dataset	38
Expand	44
Find Duplicates	48
Group By	53
Merge	55
ReShape	63
Sampling	66
Sort	71
Subset	75
Transpose	81
Variable	83
Bin	84
Box Cox	86
Compute	91
Concatenate	98
Convert	99
Date order check	104
Delete Variable	107
Factor levels	108
ID Variable	120
Lag or Lead Variable	124
Missing Values	126
Rank Variable(s)	135
Recode Variables	138
Standardize Variable(s)	141
Transform Variable(s)	144

Analysis.....	146
Cluster	147
Correlations	151
Crosstab.....	153
Distribution Analysis.....	161
Factor analysis	173
Means	178
ANCOVA.....	179
ANOVA, 1 and 2 way	182
ANOVA, one-way with random blocks	184
ANOVA, one way with blocks	189
ANOVA, N way	191
ANOVA Repeated Measures, Long	194
ANOVA Repeated Measures, Wide	196
MANOVA	198
t-test, Independent	200
t-test, One Sample	203
t-test, Paired Samples	204
Missing Value	206
Moments	210
Non-Parametric	213
Proportions	229
Summary	236
Survival	243
Tables	249
Variance	251
Distribution	254
Chi-square test	255
Lognormal	260
Normal	261
Poisson	265
Graphics.....	268
Bar Chart	269
Box Plot	270
Contour Plot	271
AB 2D Contour Plot.....	273

Distribution Plot	274
HeatMap	279
Line Charts	281
Maps	285
Pie Charts	288
Scatter plots	290
Stem And Leaf	292
Strip Chart	293
Violin	294
Six Sigma	295
Six Sigma Overview	296
Cause and Effect	297
Pareto Chart	298
Loss Function Analysis	299
MSA (Measurement System Analysis)	300
Gage R&R-Measurement System Analysis	301
Attribute Agreement Analysis	303
Gage Bias Analysis	305
Process Capability	306
Shewhart Charts	309
Cusum Chart	313
EWMA Chart	314
MQCC Chart	316
Multi-Vari Chart	318
Model Fitting	319
Regression	320
Non-Linear Regression	342
Naive Bayes	346
SEM	347
Save Model to a file	348
Load Model from a file	349
Model Evaluation	350
Compare N Models	351
Confidence Interval	352
FIT	353
Outlier Test	356

Model scoring	357
Forecasting	358
Automated ARIMA (AR)	359
Exponential Smoothing (ES)	360
Holt Winters, Non-seasonal	361
Holt Winters, Seasonal	362
Plot Time Series, Separate OR Combined	363
Plot Time Series with Correlations	364
Design of Experiment	365
Create DoE Factor Details	370
Import Design Response	372
Export Design Response	373
Create Design	374
Inspect Design	383
Modify Design	387
Analyse Design	389
Features of BioStat Prime that enhance DOE	396
Triple dots	397
Advanced Users	406
Marketplace	409

Introduction

BioStat Prime is a user-friendly biostatistics software application. The BioStat Prime user interface feels easy to use and grasp on. It is intended to be used by both novices and more experienced users, regardless of their degree of statistical experience. The capabilities of BioStat Prime are close to those of the R Commander, which presently offers the greatest number of add-ons for statistics. It is incredibly user-friendly program for doing tasks. In this software, with a graphical user interface, users can complete nearly all biostatistics-related tests. User can use the software with R in the background, without them having to understand how the language functions.

- i** Additionally, the user can develop and run R language code by utilizing the software's advanced functionalities via the R console.

What makes BioStat Prime unique?



1. BioStat Prime empowers precision in Life Sciences through Statistical Insight.
2. It is a simple-to-use UI based platform that makes statistical analysis accessible to users without extensive programming knowledge.
3. The application can interact with multiple data sets, switching between them with a single mouse click.
4. The integration of R programming language, utilizing R's extensive statistical capabilities aids in performing the analysis using R without the need for direct coding.
5. The software has a free store called MARKETPLACE for adding functions to expand the functionality of BioStat Prime.
6. BioStat Prime includes graphical tools for data visualization, allowing users to create charts, graphs, and plots to better understand their data.

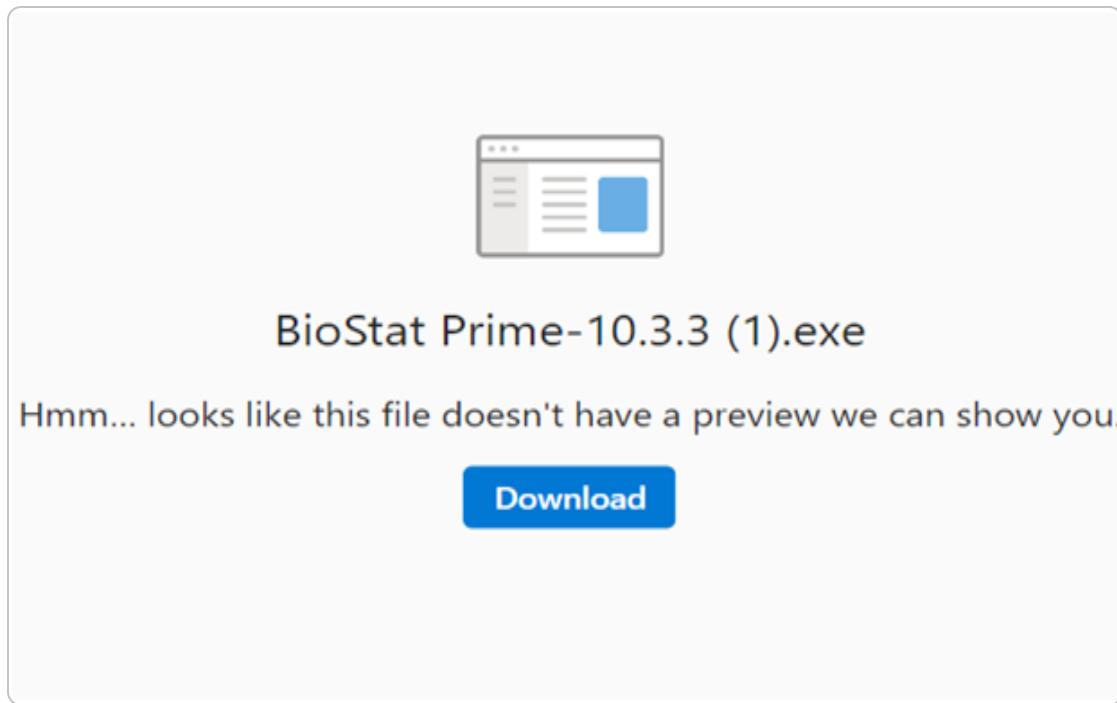
7. It also generates output and reports summarizing the results of statistical analysis. This facilitates easy interpretation and sharing of findings.
8. The console integration allows the user to add the R code chunks and run in console.

Installation

Windows

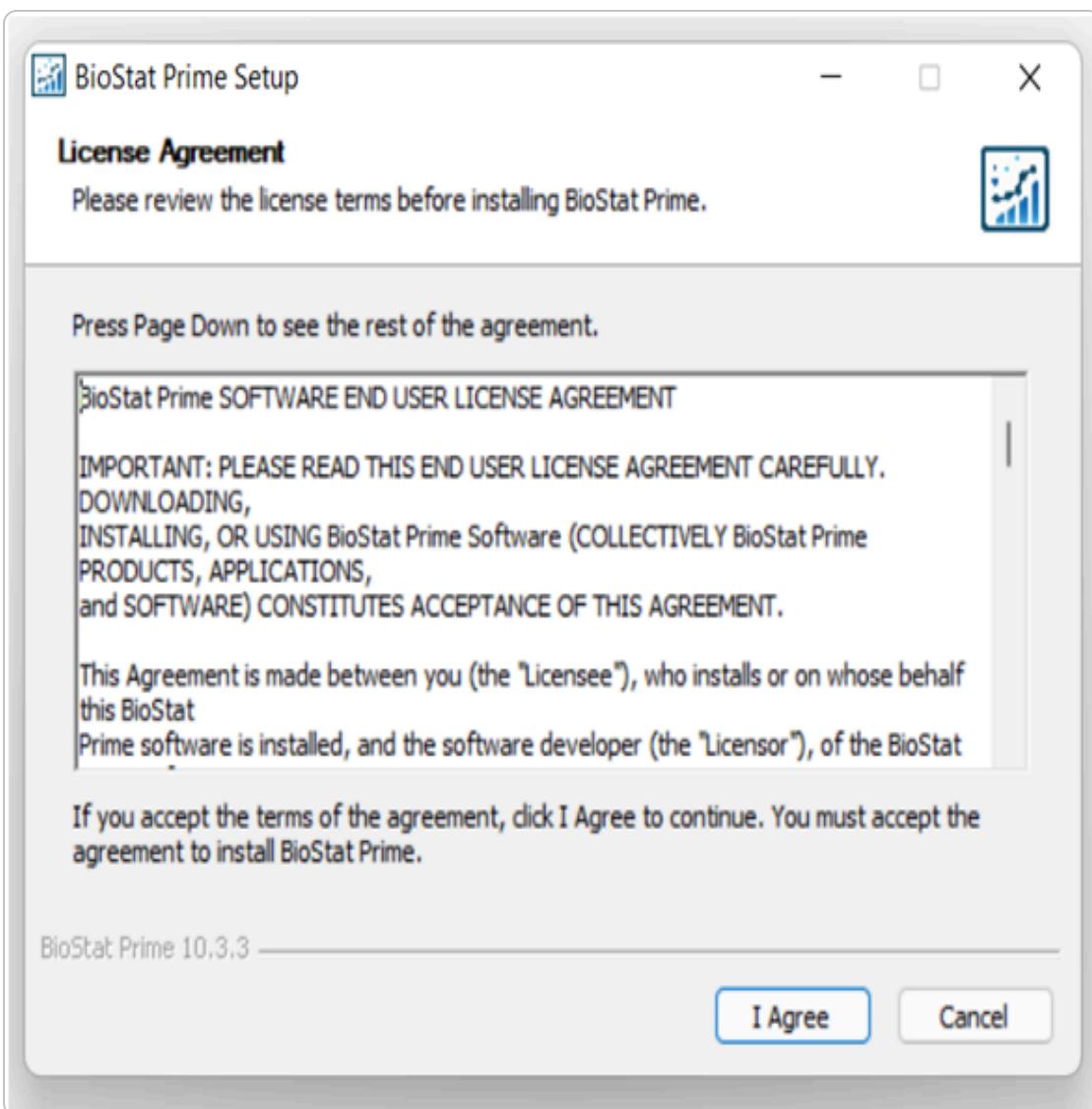
Installing BioStat Prime on Windows.

- Download the Windows installer of BioStat Prime from the given link.
- Double-click on download button to download the BioStat Prime on PC/Laptop.



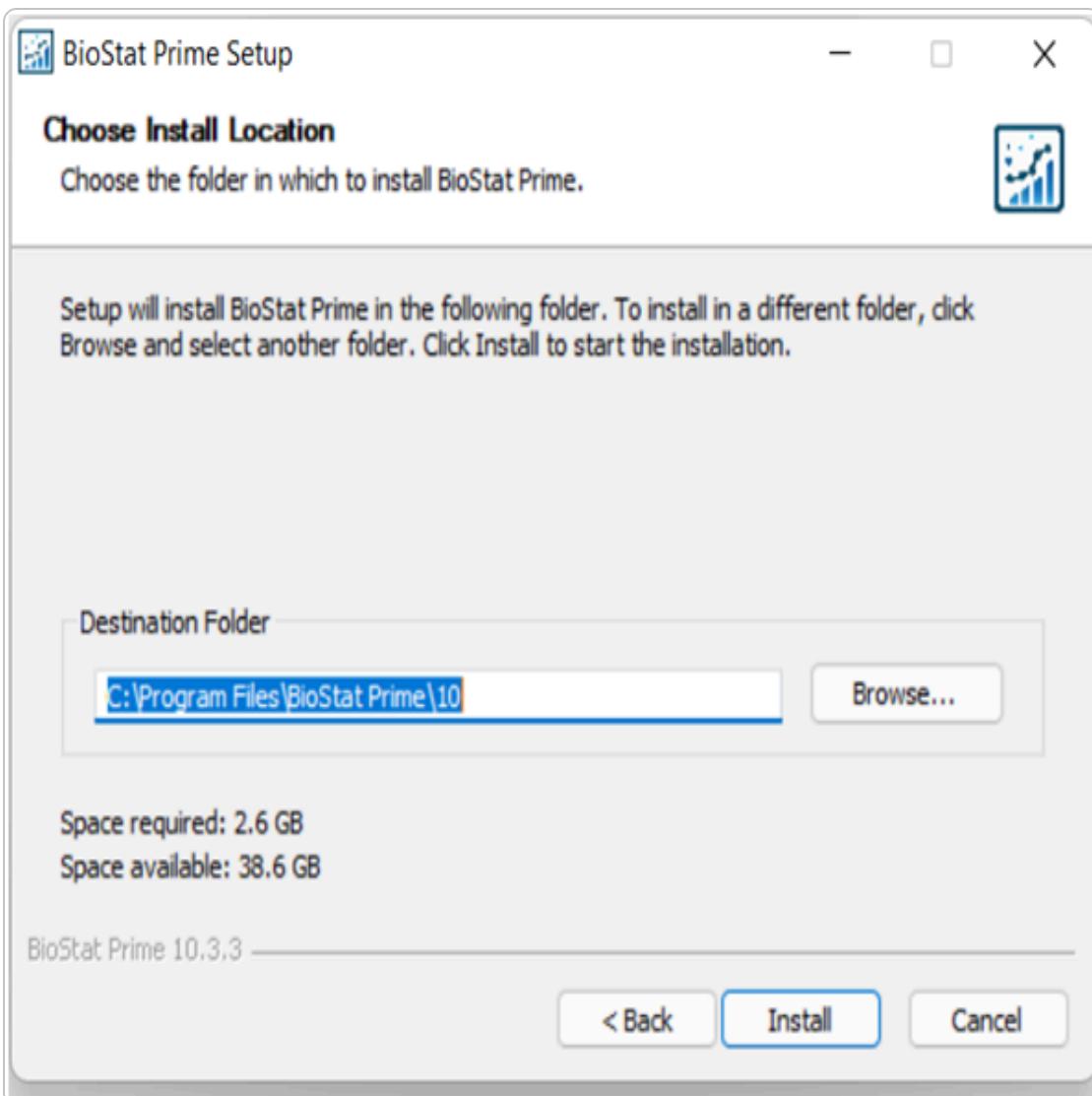
alt text

- Click on I agree to proceed to installation.



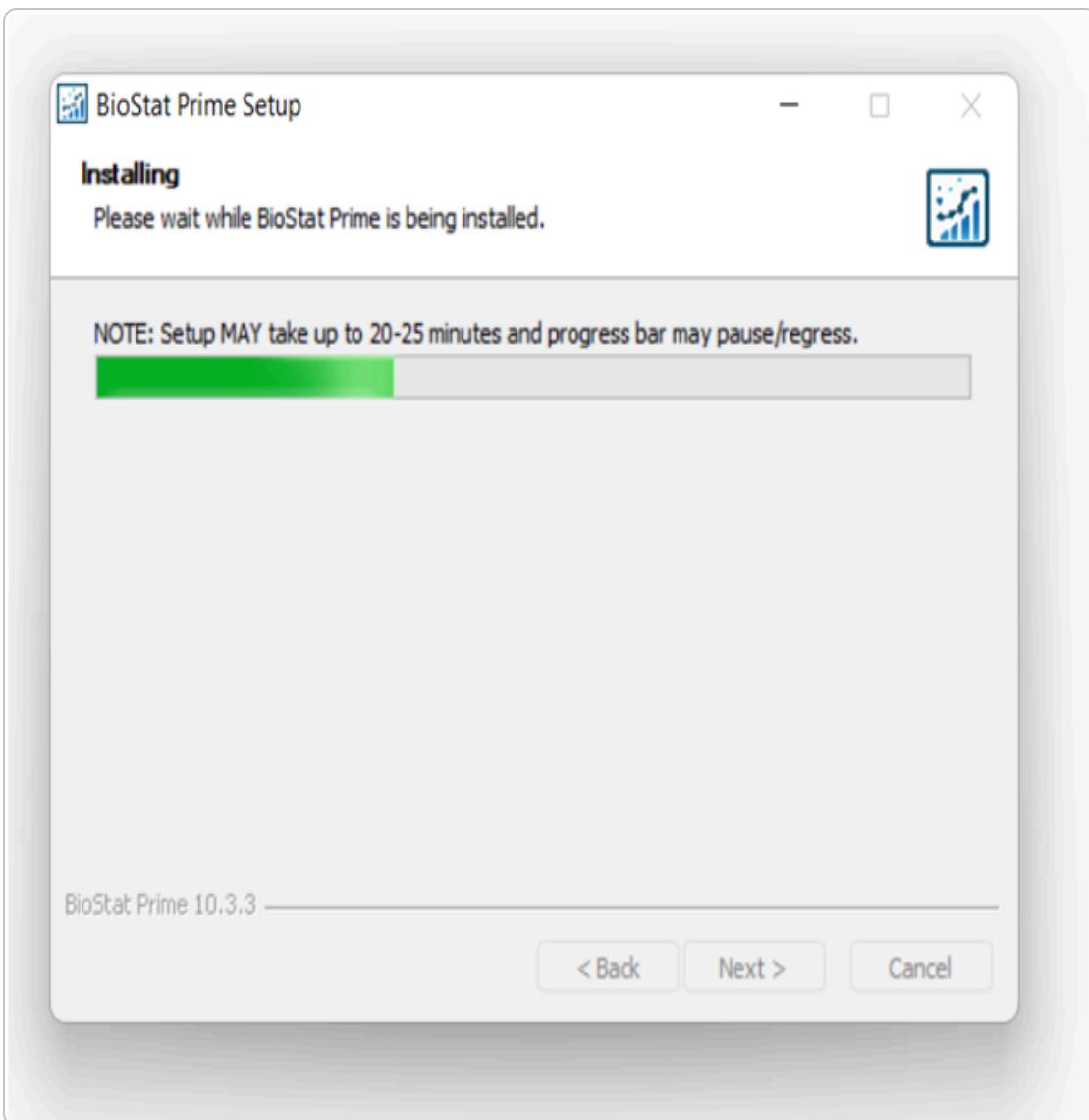
alt text

- Choose the directory where BioStat Prime will be installed and Click on Install to start the installation.



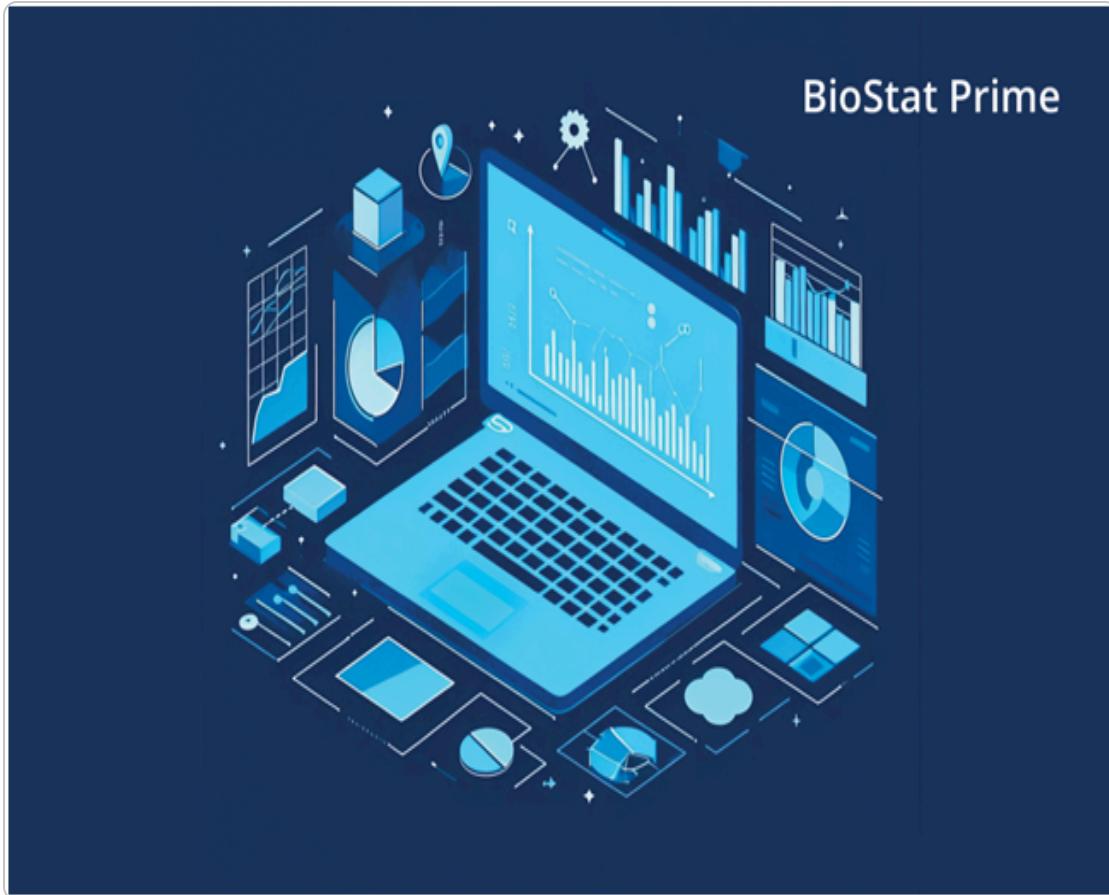
alt text

- Depending on user's machine specifications, the installation may take up to 25 minutes.



alt text

- Run BioStat Prime.



alt text

Mac OS

Installing BioStat Prime on Mac.

1. If user has Mac with Intel chip set, user needs to download and install BioStatPrime-v10- intel.dmg.
2. If user have Mac with M1 chip set, user needs to download and install BioStatPrime -v10- m1.dmg.
3. The BioStatPrime application is supported on macOS version Mojave i.e., 10.14.x and higher. If your macOS version is older, user needs to upgrade your OS to 10.14.x (Mojave) or higher.
4. Download the Mac installer of BioStat Prime from the given link.

5. From Downloads double-click the BioStatPrime-v10-intel.dmg or BioStatPrime - v10- M1.dmg that you downloaded.
6. Drag and drop BioStat Prime to your Applications.
7. Go to Applications and double click BioStat Prime.
8. Copy the Datasets_and_Demos, BioStatPrime _MarkDown, Docs, R_scripts and R_Markdown folders to a suitable location.
9. User can see the dialog below confirming that Apple has scanned our code, and no malicious code is found.

Licensing Models

Perpetual

Subscription

Grouped

Floating

Enterprise

How to use BioStat Prime

Here is a step-by-step tutor on how to use and explore the BioStat Prime software. Following this guide will ensure a smooth and effective use of the software.

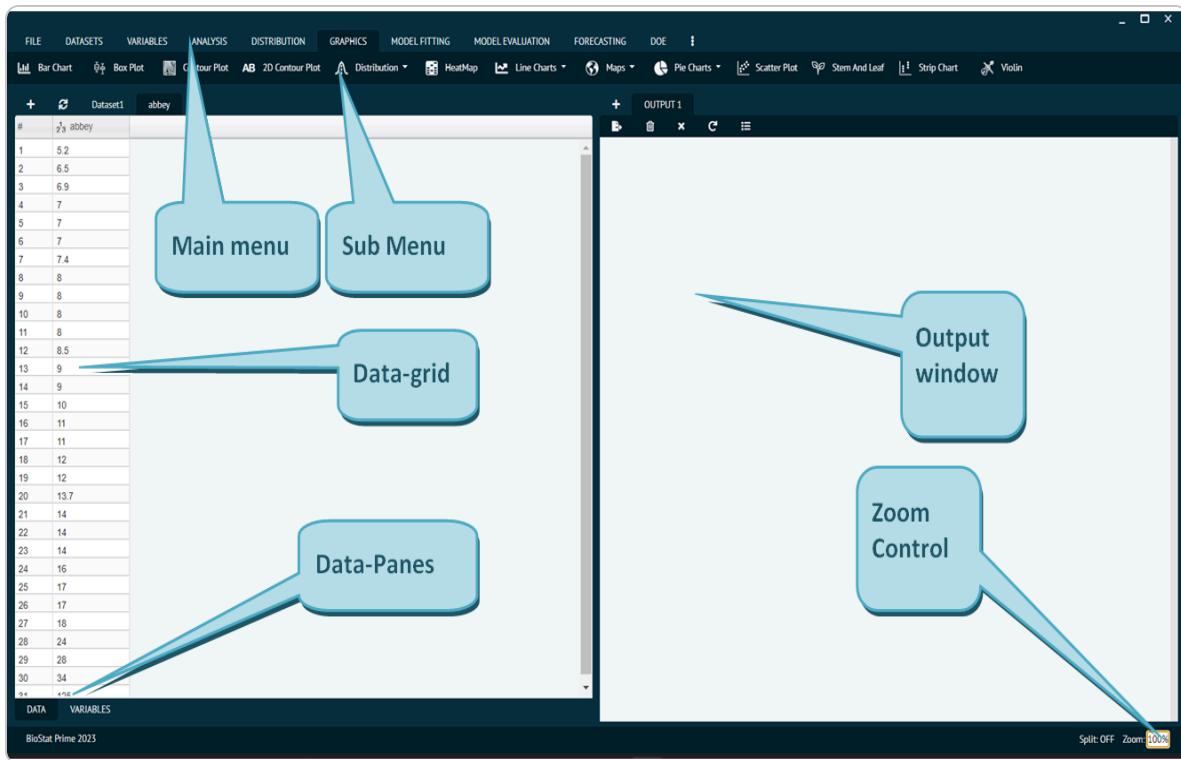
The UI of the software

As the software opens, user can see a blank unconstrained spreadsheet on left side and blank output on right side. **User has to fill this spreadsheet with some data and perform various tests.**

To populate the spreadsheet user can either type manually or user can **copy past from Excel or user can even load some inbuilt datasets.**

UI

The User Interface (UI) of BioStat Prime is divided into following sections; Main menu, Sub menu, Output window, Data-grid and R console ([Advanced Users](#)), Zoom control.



alt text

All the datasets that are imported into the software are displayed in the data-grid. The data-grid has two panes: **data pane** and **variable pane**.

i Both the panes are fully interactive.

! The main menu, at the top, comprises different functions that are responsible for data manipulation commands.

! Inside the main menu, is a sub menu that has various functions and tests that can be performed on the data and for all the functions in sub menu, if the user presses dropdown button, then related sub functions will appear in the drop-down.

! The result of analysis is displayed in output window.

- i** In BioStat Prime the user can work on multiple data pane windows and output windows.

Working with Data-grid

As the user starts to enter data inside the data-grid, he needs to make sure to specify which is which variable. Data-grid can contain variables of different types e.g. numeric, integer, logical, ordered factor, etc.

By making changes in variable pane user can have different levels of data grid columns. To make any change in the variable formatting the user need to switch to variable pane and the select the variable row to be changed with a right click.

In data pane user has access to various data types and in variable pane user has access to various variable types, imported from the dataset. All the research will be displayed in the output window.

- ⚠** Whenever user enters some data the output shows a comment stating Grid Edit.

The screenshot shows the R Commander interface with the 'VARIABLES' tab selected. A context menu is open over the 'carb' variable in the 'mtcars' dataset, listing options like 'Make Character' (which is highlighted in red) and 'Make Numeric'. A blue callout bubble points from the text 'Changing Variable format to character in variable pane' towards the 'Make Character' option.

FILE DATASETS VARIABLES ANALYSIS DISTRIBUTION GRAPHICS MODEL FITTING MODEL EVALUATION FORECAST

* Aggregate Compare Datasets Expand Find Duplicates Group By Merge ReShape

+ Dataset1 abby mtcars

#	Name	Class	Type	Measure	Levels
1	mpg	numeric	double	scale	
2	cyl	numeric	double	scale	
3	disp	numeric	double	scale	
4	hp	numeric	double	scale	
5	drat	numeric	double	scale	
6	wt	numeric	double	scale	
7	qsec	numeric	double	scale	
8	vs	numeric	double	scale	
9	am	numeric	double	scale	
10	gear	numeric	double	scale	
11	carb	numeric	double	scale	

Changing Variable format to character in variable pane

Add Factor Level
Make Factor
Make Ordered Factor
Make Character
Make Numeric
Insert New Date Variable
Insert New Date Variable at End
Insert New DateTime Variable
Insert New DateTime Variable at End
Insert New Numeric Variable
Insert New Numeric Variable at End
Insert New Character Variable
Insert New Character Variable at End
Insert New Factor Variable
Insert New Factor Variable at End
Delete Variable

DATA VARIABLES

alt text

This will open a pop-up window as shown above.

The screenshot shows a software interface for data analysis. At the top, there is a navigation bar with tabs: FILE, DATASETS (which is selected), VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, and MODEL FITTING. Below the navigation bar are several icons: Aggregate, Compare Datasets, Expand, Find Duplicates, Group By, and a Help icon.

The main area displays a table titled "mtcars". The table has columns: #, Name, Class, Type, Measure, and Levels. The rows show the following data:

#	Name	Class	Type	Measure	Levels
1	mpg	numeric	double	scale	
2	cyl	numeric	double	scale	
3	disp	numeric	double	scale	
4	hp	numeric	double	scale	
5	drat	numeric	double	scale	
6	wt	numeric	double	scale	
7	qsec	numeric	double	scale	
8	vs	character	character	scale	
9	am	numeric	double	scale	
10	gear	numeric	double	scale	
11	carb	numeric	double	scale	

A blue callout bubble points to the "vs" row in the table, specifically to the "Type" column which is listed as "character". The text inside the bubble reads: "Variable format changed to character in variable pane."

alt text

- i** User can change the formats of any column in variable section and that will be reflected in data-pane.

Dataset1 abby mtcars

	\bar{x}_3 drat	\bar{x}_3 wt	\bar{x}_3 qsec	\bar{x}_3 vs	\bar{x}_3 am	\bar{x}_3 gear	\bar{x}_3 carb
3.9	2.62	16.46	0	1	4	4	
3.9	2.875	17.02	0	1	4	4	
3.85	2.32	18.61	1	1	4	1	
3.08	3.215	19.44	abc	0	3	1	
3.15	3.44	17.02	0	0	3	2	
2.76	3.46	20.22	1	0	3	1	
3.21	3.57	15.84		0	3	4	
3.69	3.19	20		0	4	2	
3.92	3.15	22.9		0	4	2	
3.92	3.44	18.3		0	4	4	
3.92	3.44	18.9		0	4	4	
3.07	4.07	17.4		0	3	3	
3.07	3.73	17.6		0	3	3	
3.07	3.78	18		0	3	3	
2.93	5.25	17.98		0	3	4	
3	5.424	17.8		0	3	4	
3.23	5.345	17		0	3	4	
4.08	2.2	1		1	4	1	
				1	4	2	
				0	3	1	
				0	3	2	
3.15	3.435	17.3	0	0	3	2	
3.73	3.84	15.41	0	0	3	4	
3.08	3.845	17.05	0	0	3	2	
4.08	1.935	18.9	1	1	4	1	
4.43	2.14	16.7	0	1	5	2	
3.77	1.513	16.9	1	1	5	2	
4.22	3.17	14.5	0	1	5	4	
3.62	2.77	15.5	0	1	5	6	

Character can be added.

alt text

Dialog

On selecting any of the statistical function, a window will appear replacing the data-grid. This window is called Dialog.

The Dialog is where different variables are selected to perform some tests or analysis.



The variable from source side is sent to the target side by selecting it and

clicking the arrow button.

- i** To select multiple variables, user needs to hold Alt button on keyboard and select multiple source variables.

For each statistical function there are function specific options at the top of dialog window.

The top right corner of the dialog contains a few options like;

Execute button

Executes the dialog.

Syntax button

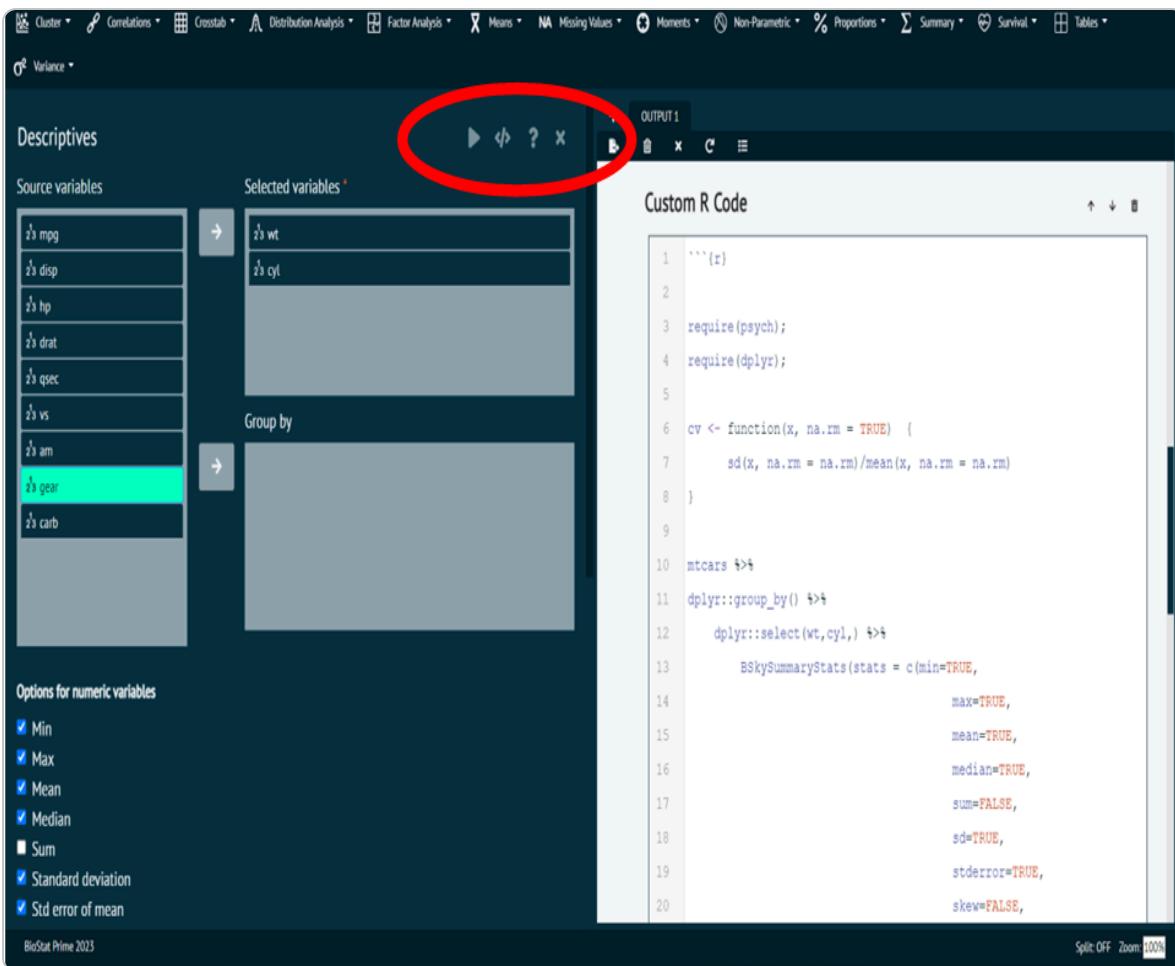
Displays the R syntax for respective dialog analysis.

?

Provides quick summary help.

Cross button

Used for closing dialog so that user can visualize the dataset.



alt text

Executing the Dialog

Once the input for analysis is fed into the dialog, the execution is performed by **execute dialog button** |> and the output is displayed in the output window.

A The box in the left column brings up the dialog again. It is like the history that tells us about the criteria we had chosen.

i But that is valid for the initial values only, once the dialog is executed then history will bring up only the initials values and not the values inserted later.

i Also, as soon as user edits the R syntax associated with the dialog, user removes the association between dialog and the output window because of

which the history is no more saved in the dialog and the output is executed as per the R syntax.

- ⚠ The arrow buttons in top right corner of output window aids the user in navigating between the different outputs by moving up and down.

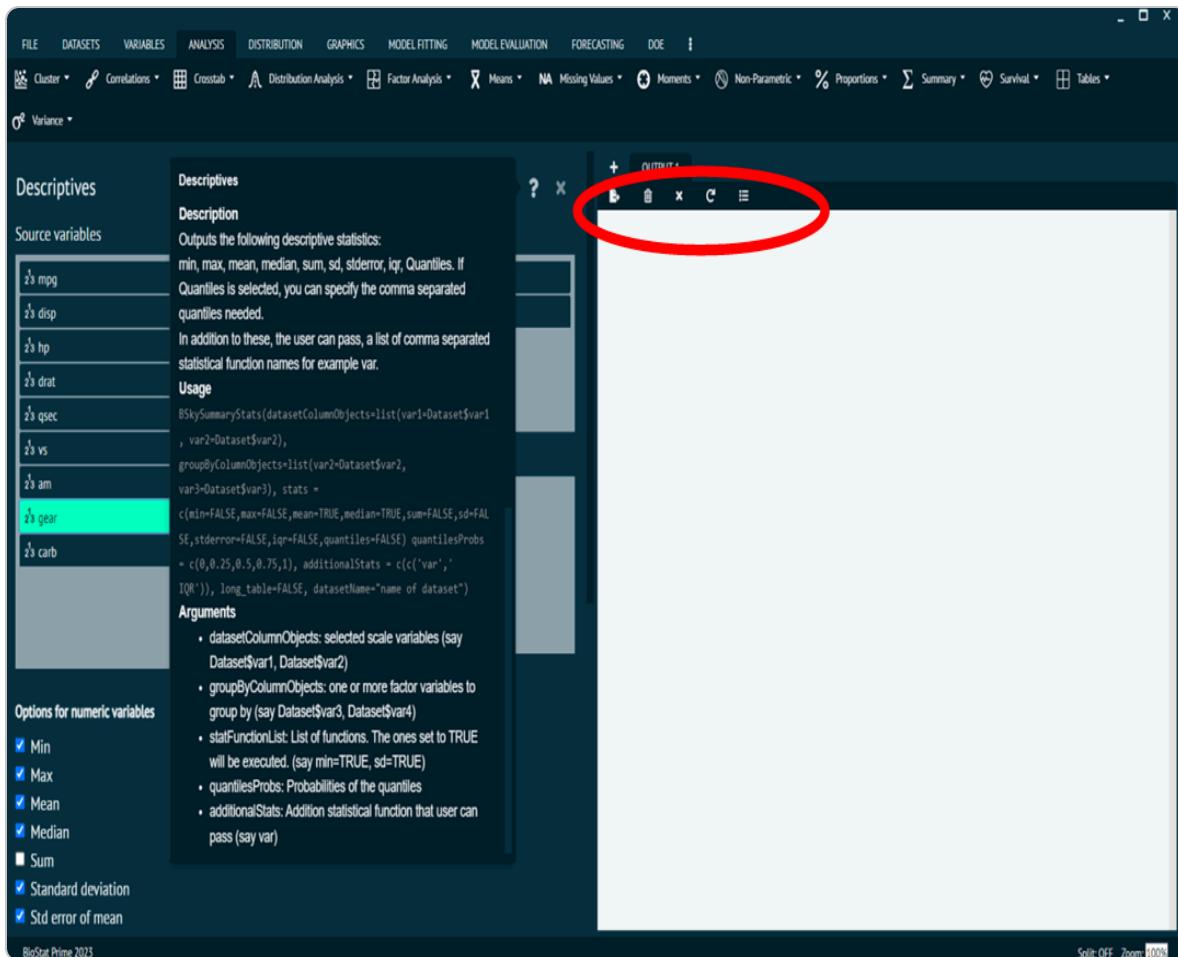
The screenshot shows the RStudio output window titled "OUTPUT 1". The main content is a "Descriptives" report. At the top left, there are icons for saving, deleting, and opening files. On the far right, there are three small buttons: an upward arrow, a downward arrow, and a square. Both the file icon and the downward arrow button are circled in red. Below the title, it says "Dataset Overview" with columns for "Dataset", "Variables", and "Observations". A single row shows "data" with 1 variable and 32 observations. Below this, under "Numerical Statistical Analysis by Variable", is a table with two columns: "stats" and "disp". The table rows are: min (71.1000), 1st Qu (120.8250), mean (230.7219), median (196.3000), 3rd Qu (326.0000), max (472.0000), sd (123.9387), std. error (21.9095), and cv (0.5372). At the bottom right of the window, it says "Split: OFF Zoom: 100%".

stats	disp
min	71.1000
1st Qu	120.8250
mean	230.7219
median	196.3000
3rd Qu	326.0000
max	472.0000
sd	123.9387
std. error	21.9095
cv	0.5372

alt text

Output Window

For each output of statistical analysis there are function specific options at the top of output window. From left to right.



alt text

Export

Used to save the output of the analysis by exporting it to the PC/Laptop with file type as R markdown, as HTML or as BioStat.

Delete

Used to clear the output of the analysis.

Close

Closes the output window (but atleast one output window should always be open).

Refresh

Used to restart the R console.



NOTE:



1. User can visualize dialog window and data-grid at the same time, for that the user needs to click, hold and drag the dialog window to a different position.



2. In order send the variables to target or destination box in dialog, the user needs to select the required variable and click the arrow button to send or un-send the source variable to target.

File

The file function is the First function in the main menu. It leads the user to following sub functions.



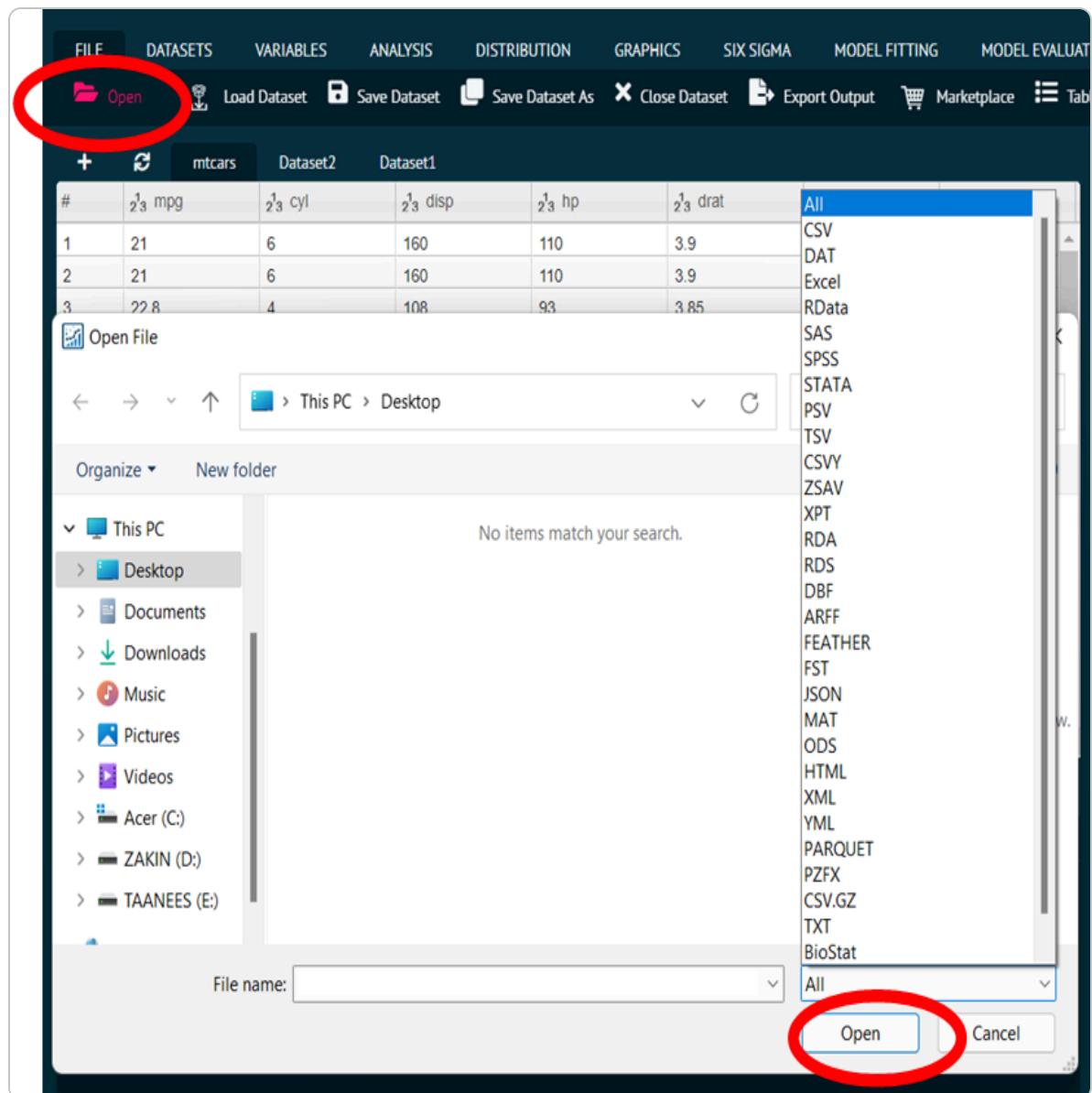
alt text

Open

Used to load external dataset from user's PC/Laptop supporting various formats including;

Formats Supported

CVS, DAT, Excel, RData, SAS, SPSS, STSTA, PSV, TSV, CSVY, ZSAV, XPT, RDA, RDS, DBF, ARFF, FEATHER, FST, JSON, MAT, ODS, HTML, XML, YML, PARQUET, PZFX, CSV.GZ, TEXT, BioStat.



alt text

Load Dataset

Loads the datasets that are internal to BioStat Prime, along with it is the feature of installing the associated R packages.

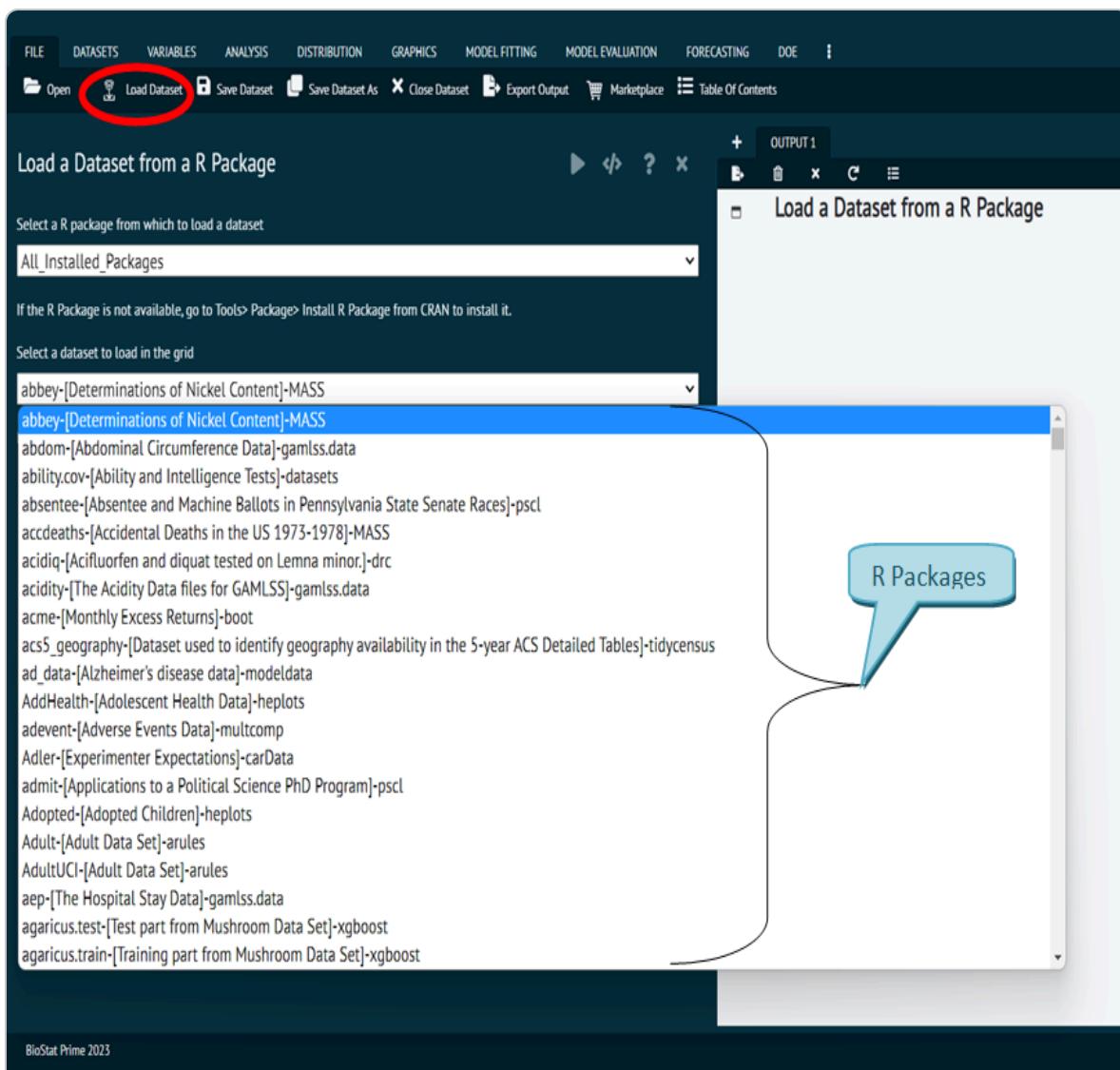
In order to load a dataset, user needs to do the following.

Steps

Select the dataset and R package from the dropdown in the load dataset dialog -> execute the dialog.

If the user does not see a dataset in the dropdown, the package selected does not contain the dataset.

⚠ Thus, to install the R Package required by the user, user needs to go to **Tools** -> **Package** -> **Install R Package from CRAN**.



alt text

Save Dataset

Saves the dataset the user had worked on into the following formats.

Formats Supported

R Object, Comma separated, Excel 2007-2010, SAS, SPSS, STATA, PSV, TSV, CSVY, ZSAV, XPT, RDA, RDS, DBF, ARFF, FEATHER, FST, JSON, MAT, ODS, HTML, XML, YML, PARQUET, PZFX.

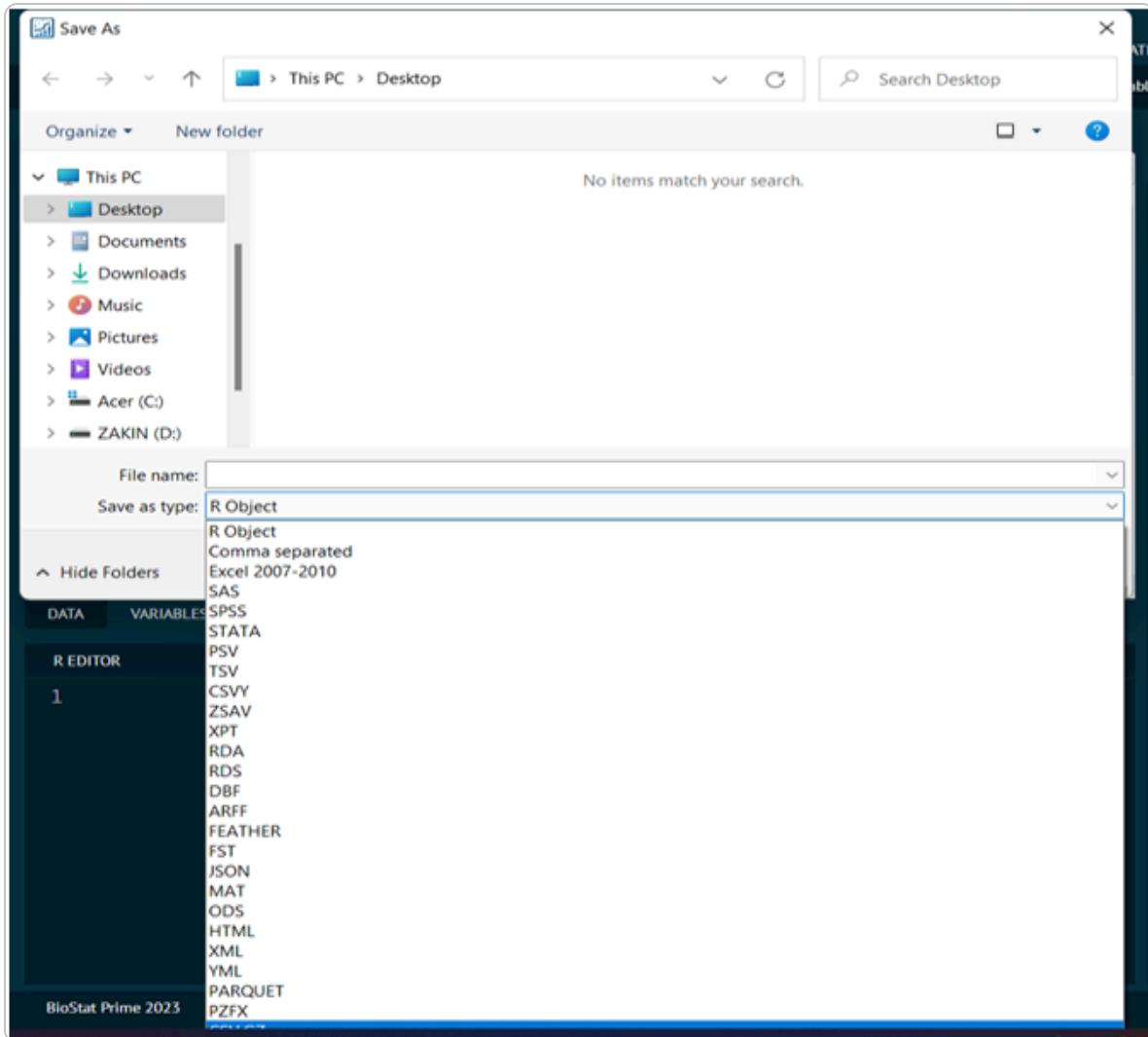
The user can decide as per the requirement which format to choose. Save seeks to update the current content of the previously saved file,

Save As

Save As aims to create a new folder or save an existing file to a new location with the same name or a different title. Formats used for save/save As the dataset are as follows;

Formats Supported

R Object, Comma separated, Excel 2007-2010, SAS, SPSS, STATA, PSV, TSV, CSVY, ZSAV, XPT, RDA, RDS, DBF, ARFF, FEATHER, FST, JSON, MAT, ODS, HTML, XML, YML, PARQUET, PZFX.



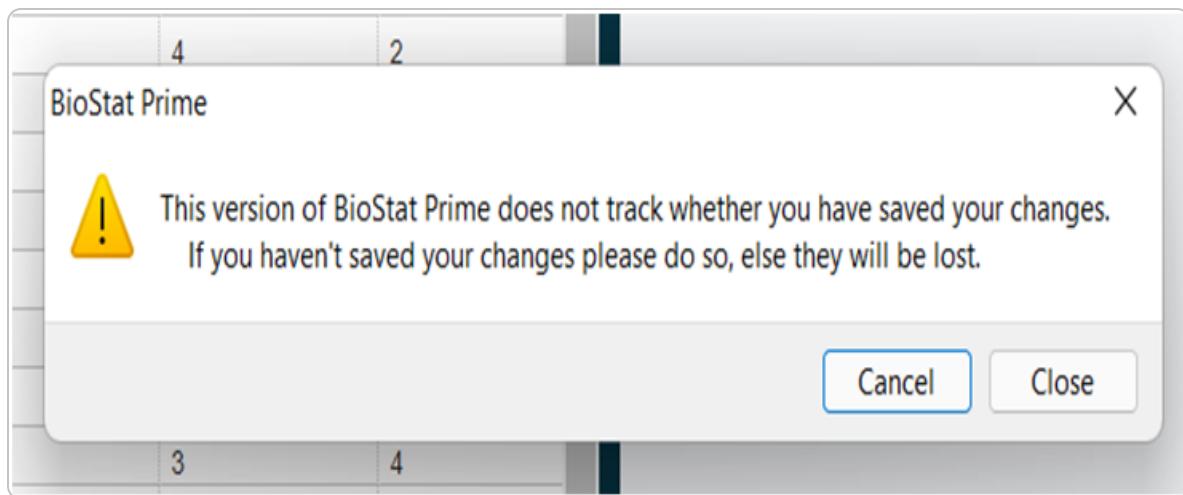
alt text

Close Dataset

Closes the dataset that user had been working on.



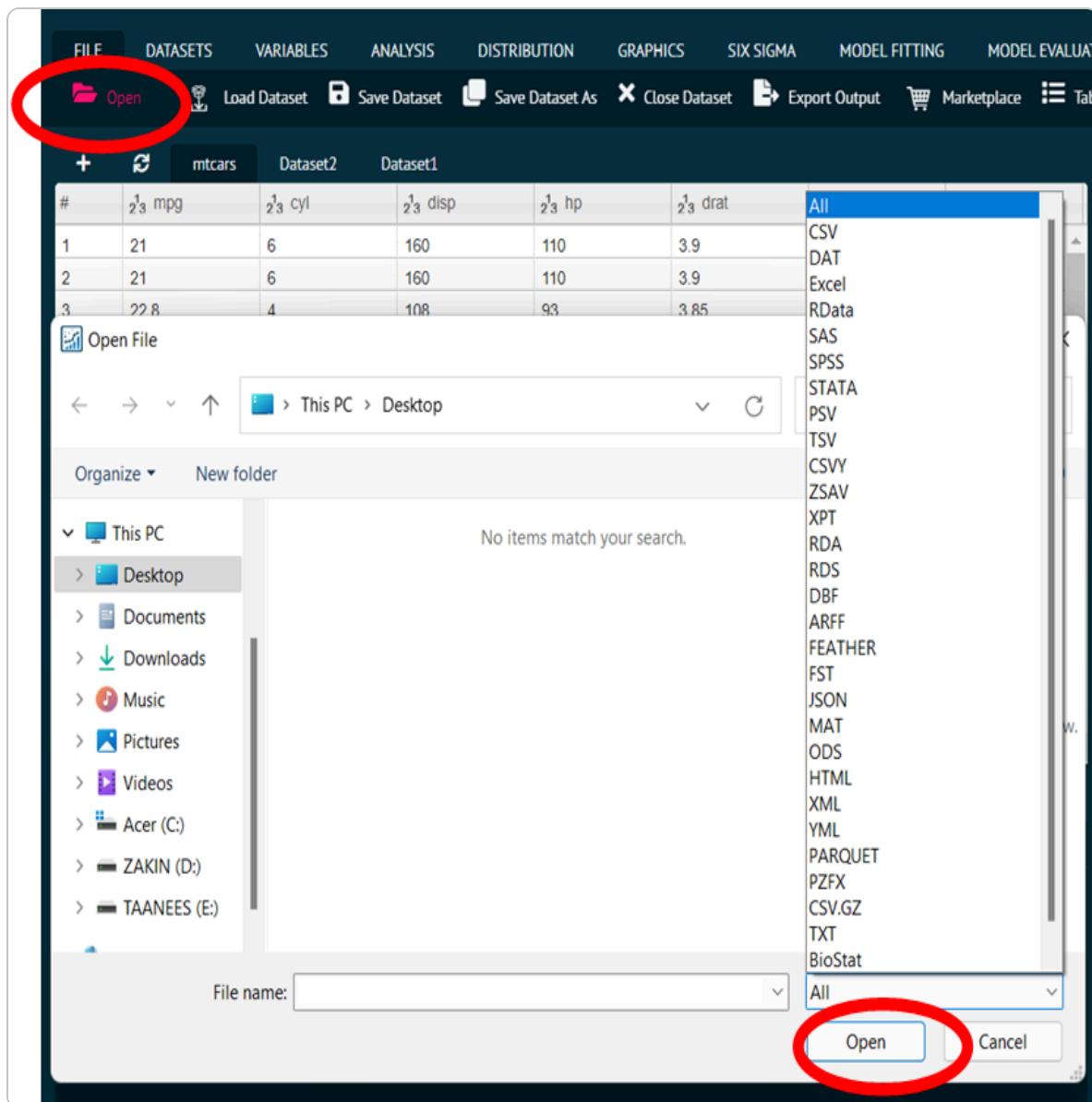
BioStat issues a warning to save the dataset before closing it.



alt text

Export output

Export output exports the output to be saved in user's system. It is used to save the output of the analysis by exporting it to the PC/Laptop with file type as R markdown, as HTML or as BioStat.



alt text

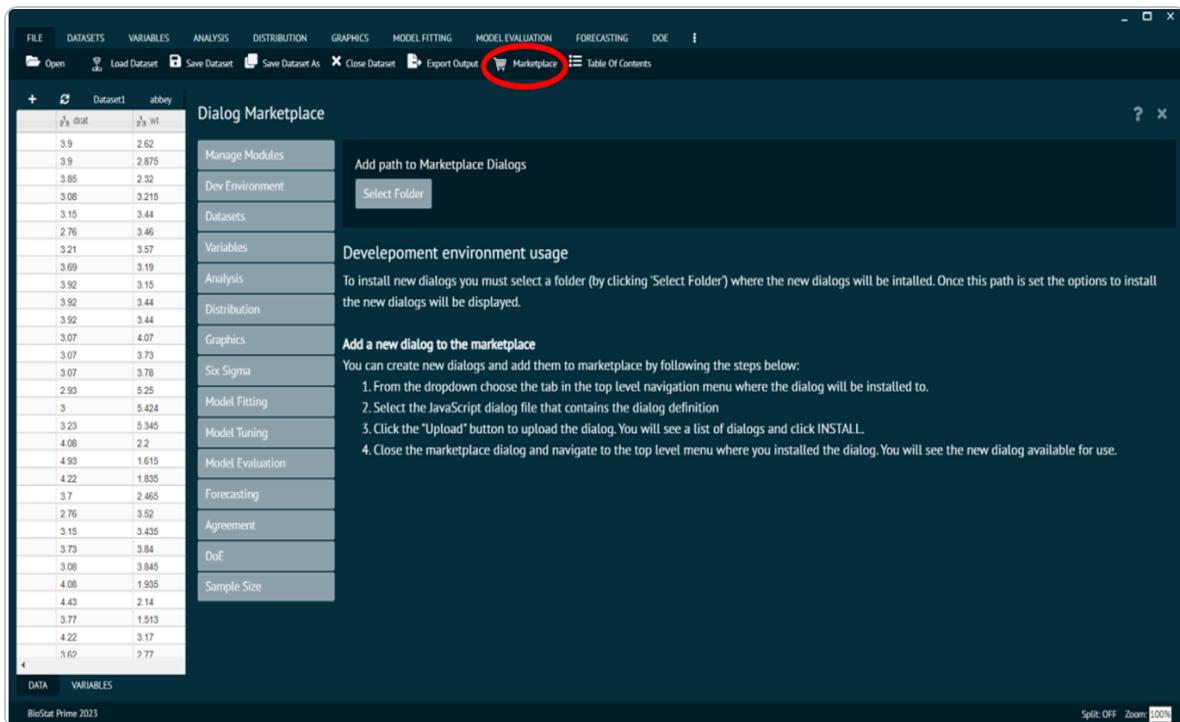
Marketplace

Marketplace ([Marketplace](#)) is one of the most popular feature of BioStat Prime as it enables the user to customise the application as per requirements and expands its functionality.

Marketplace is a free store for adding R libraries and functions to BioStat Prime to newer areas of statistics.

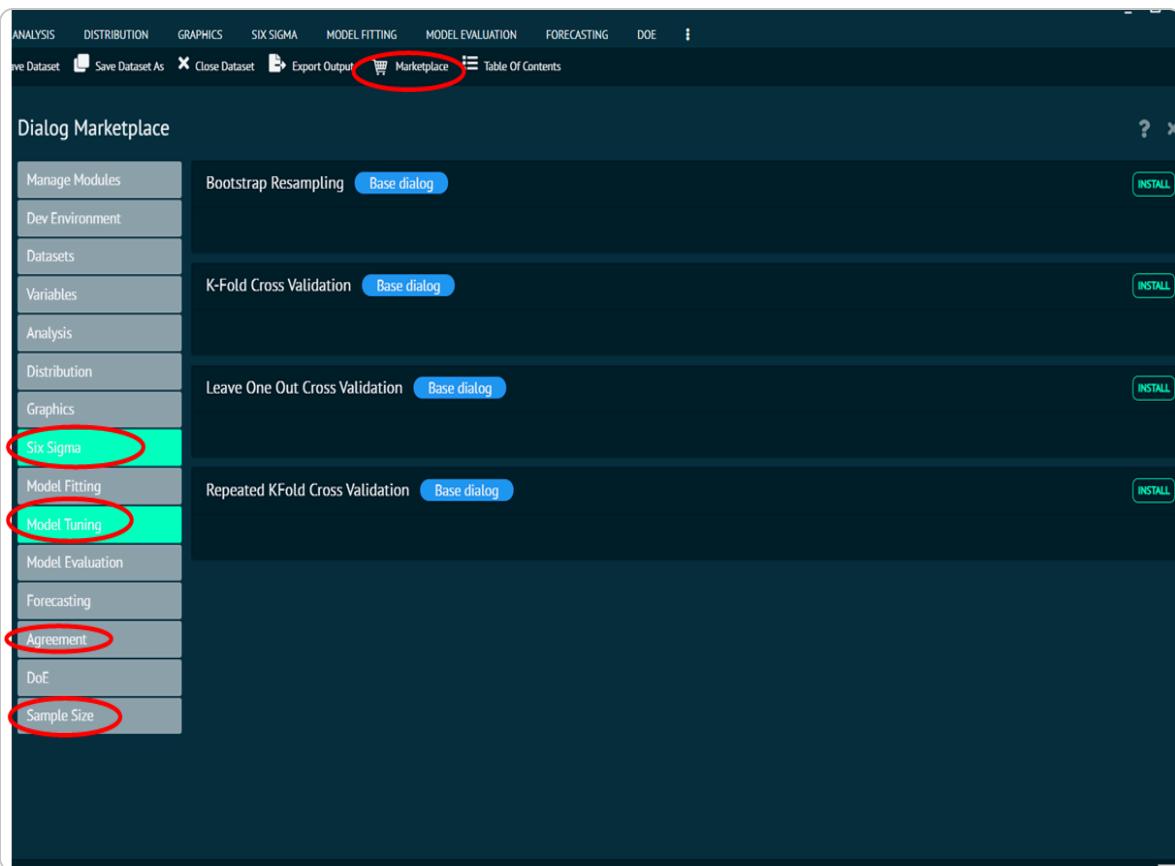
It Installs or hides R functions and packages. User can install various new libraries and dialogs like Six Sigma ([Six Sigma](#)), Agreement, Model Fitting ([Model Fitting](#)), Sample

Size and the associated libraries with them.



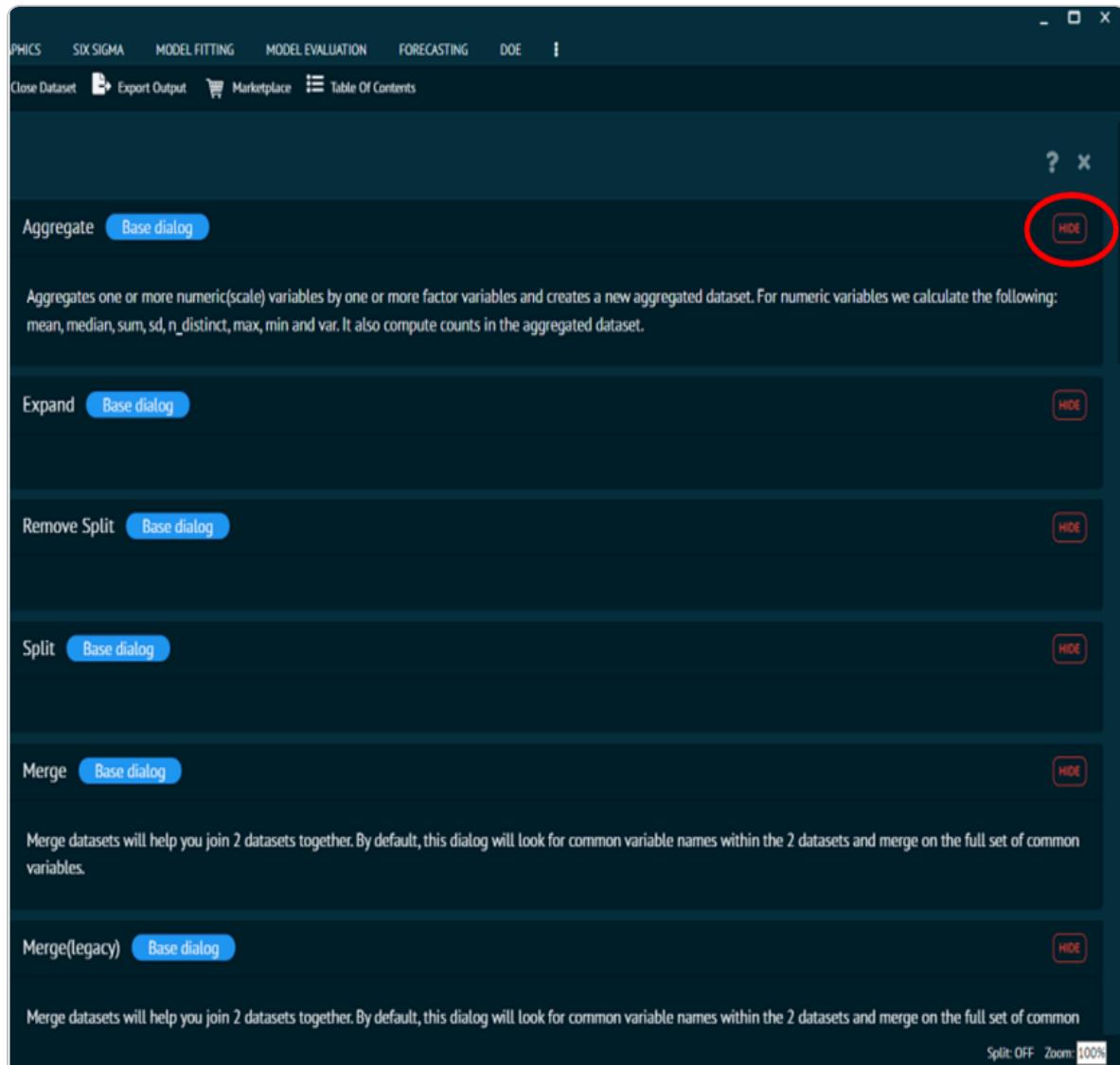
alt text

- i** The installation of various libraries increases the functionality of BioStat Prime, without any charges.



alt text

- ⓘ Also, the libraries of already installed dialogs (["Dialog" in "How to use BioStat Prime"](#)), can be hid by the user by clicking the hide button on right of the respective library.



Dataset

This section of the main menu gives access to the data manipulation commands for the sake of proper and customised analysis of the given dataset. It leads the user to sub functions like;

Aggregate ([Aggregate](#)), Compare Dataset ([Compare Dataset](#)), Expand ([Expand](#)), Find Duplicates ([Find Duplicates](#)), Group By ([Group By](#)), Merge ([Merge](#)), ReShape ([ReShape](#)), Sampling ([Sampling](#)), Sort ([Sort](#)), Subset ([Sampling](#)), Transpose ([Transpose](#)).

The screenshot shows a software application window with a dark-themed menu bar at the top. The menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION, FORECASTING, DOE, and a three-dot menu. Below the menu bar, there is a toolbar with icons for Aggregate, Compare Datasets, Expand, Find Duplicates, Group By, Merge, ReShape, Sampling, Sort, Subset, and Transpose. A table titled "Dataset1" is displayed, containing columns labeled X1 through X7. The first row has a header "#". An "OUTPUT 1" window is also visible on the right side of the interface.

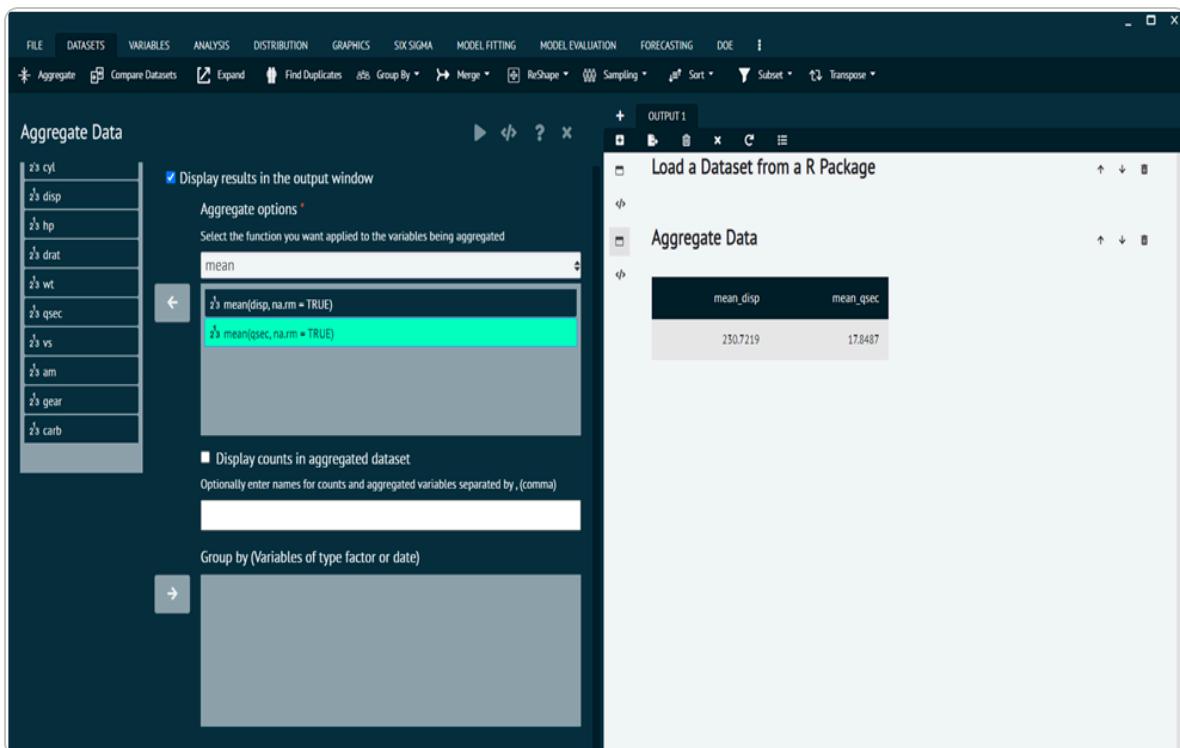
alt text

The above-mentioned functions are discussed in detail in the up-coming section.

Aggregate

Aggregates one or more numeric (scale) variables by one or more factor variables and creates a new aggregated dataset.

For numeric variables user can calculate the following: mean, median, sum, std deviation, n_distinct, max, min and var. It also computes the counts in the aggregated dataset.



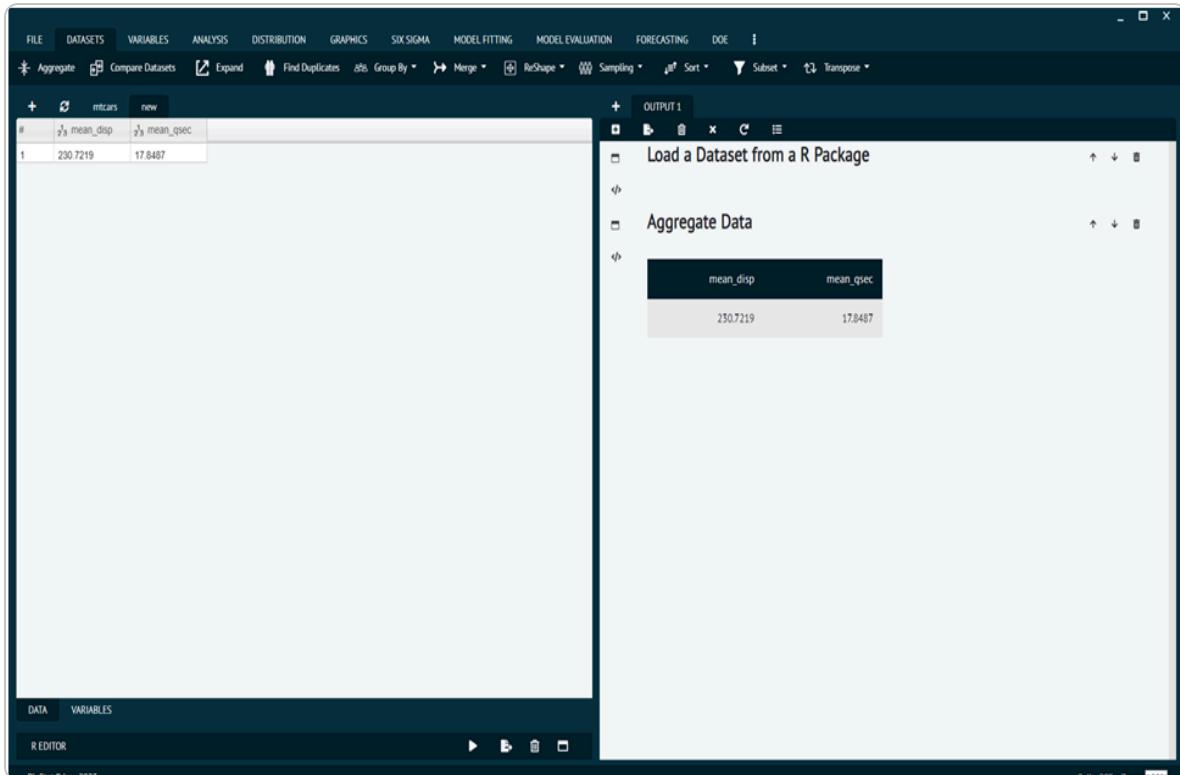
aggregate

To aggregate variables user needs to follow the steps given bellow.

Steps

Load the dataset -> click on the DATASET tab in main menu -> select AGGREGATE -> Once, the dialog appears select the functions to be applied to the variables being executed -> Execute the dialog.

Output of aggregate is given as.



alt text

The arguments used in executing the dialog are given as follows.

⚠ Arguments

1. var1: factor to group by
2. var2, var3: variable to aggregate
3. newvarmean: mean of var2 grouped by var1 in the aggregated dataset
4. newvarmedian: median of var3 grouped by var1 in the aggregated dataset

Compare Dataset

Compares two datasets and reports any differences between them. It will Compare the datasets that will help the user to compare two datasets.

- By default, the comparison is done row by row.

To compare datasets user needs to follow the steps given bellow.

Steps

Load the dataset -> click on the DATASET tab in main menu -> select Compare Dataset -> Once the dialog appears choose the datasets to be compared-> Execute the dialog.

Output of comparison is given as.

The screenshot shows the BioStat Prime 2023 software interface. The main window title is "Compare Datasets". In the "Source Datasets" section, "mtcars" is selected as the first dataset and "abey" as the second dataset. A note states: "By default, the comparison is done row-by-row. See ID Options for more options." Under "Numeric Variable Tolerances", the "Unsigned numerical difference" option is selected. There is a field for "Max value of difference (percent should be 0-1)" with a value of 0.0. Under "Factor Variable Tolerances", the "Compare both underlying levels and labels" option is selected. Under "Character Variable Tolerances", the "Treat text as-is" option is selected. To the right, the "OUTPUT" window displays the results of the comparison:

version	arg	ncol	nrow	
1	x	mtcars	11	32
2	y	abey	1	31

Below this is the "Summary of overall comparison" table:

	statistic	value
1	Number of by-variables	0
2	Number of non-by variables in common	0
3	Number of variables compared	0
4	Number of variables in x but not y	11
5	Number of variables in y but not x	1
6	Number of variables compared with some values unequal	0
7	Number of variables compared with all values equal	0
8	Number of observations in common	31
9	Number of observations in x but not y	1

alt text

The various options present in this dialog are explained as under.

Numeric Variable Tolerance Options

Unsigned numerical difference (default)

Assesses whether 2 values are different by taking the absolute value of the difference and testing if it is larger than the max value of difference value

- ⚠ Example: age = 18.5 vs. age = 18.8 difference = $|18.5 - 18.8| = |-0.3| = 0.3$

Unsigned percent difference

Assesses whether 2 values are different by taking the absolute value of the percent difference and testing if it is larger than the max value of difference value

- ⚠ Example: age = 18.5 vs. age = 18.8 difference = $|18.5 - 18.8| / 18.8 = |-0.3| / 18.8 = 0.3 / 18.8 = 0.0160$

Max value of difference (blank by default)

If blank, values should be identical (as best detected by your system). Otherwise, enter a value > 0 that will be used to determine if the difference is large enough to be called different.

- ⚠ Example 1 with numerical difference: age = 18.5 vs. age = 18.8 and max value = 0.2 difference = $|18.5 - 18.8| = |-0.3| = 0.3$ since $0.3 > 0.2$, this would be flagged as different

- ⚠ Example 2 with numerical difference: age = 18.5 vs. age = 18.6 and max value = 0.2 difference = $|18.5 - 18.6| = |-0.1| = 0.1$ since $0.1 < 0.2$, this would not be flagged as different

- ⚠ Example 1 with percent difference: age = 18.5 vs. age = 18.8 and max value = 0.01 difference = $|18.5 - 18.8| / 18.8 = |-0.3| / 18.8 = 0.3 / 18.8 = 0.0160$ since $0.016 >$

0.01, this would be flagged as different

- ⚠ Example 2 with percent difference: age = 18.5 vs. age = 18.8 and max value = 0.01 difference = $|18.5 - 18.8| / 18.8 = |-0.3| / 18.8 = 0.1 / 18.8 = 0.0005$ since $0.0005 < 0.01$, this would not be flagged as different

Treat integer variables as numeric variables in comparisons

Should variables with class integer be compared to variables with class numeric? User may end up with variables of different classes when user reads in data from external sources (like Excel)

- ⚠ Example: age (integer) = c(18, 33, 45) vs. age (numeric) = c(18.6, 33.4, 45.1) If you want the values of these 2 variables compared between the data sets, check this box. By default, the system only compares numeric variables of the same class.

Factor Variable Tolerance Options

Compare both underlying levels and labels (default)

Compares both the stored values (1,2,3) and labels (mild, moderate, severe) between the variables

- ⚠ Example 1: disease (1 = mild, 2 = moderate, 3 = severe) vs. disease (1 = mild, 2 = severe) These 2 variables would be considered different because the 2 = moderate in 1st variable but 2 = severe in the 2nd variable
- ⚠ Example 2: disease (1 = mild, 2 = moderate, 3 = severe) vs. disease (1 = mild, 2 = moderate, 3 = sev) These 2 variables would be considered different because the 3 = severe in 1st variable but 3 = sev in the 2nd variable

Compare underlying levels only

Compares only the underlying levels (1,2,3) across factor variables

⚠ Example 1: disease (1 = mild, 2 = moderate, 3 = severe) vs. disease (1 = mild, 2 = severe) These 2 variables would not be considered different because the underlying values 1,2,3 in the 1st variable are the same as the values 1,2 are in the 2nd variable.

⚠ Example 2: disease (1 = mild, 2 = moderate, 3 = severe) vs. disease (1 = mild, 2 = moderate, 3 = sev) These 2 variables would be considered different because the 3 = severe in 1st variable but 3 = sev in the 2nd variable

Compare underlying labels only

Compares only the underlying labels (mild, moderate, severe) across factor variables

⚠ Example 1: disease (1 = mild, 2 = moderate, 3 = severe) vs. disease (1 = mild, 2 = severe) These 2 variables would not be considered different because the labels are the same

⚠ Example 2: disease (1 = mild, 2 = moderate, 3 = severe) vs. disease (1 = mild, 2 = moderate, 3 = sev) These 2 variables would be considered different because the 3 = severe in 1st variable but 3 = sev in the 2nd variable so the labels are different

Treat factor variables as character variables in comparisons

Checks if factors should be converted to character variables using their labels for the comparison. You may end up with discrepant classes if you read data from different sources.

⚠ Example: disease (factor with 1 = mild, 2 = moderate, 3 = severe) vs. disease (character with mild, moderate, severe) To compare these variables, check the box to convert the 1st variable to a character variable.

Character Variable Tolerance Options

Treat text as-is (default)

Text is compared exactly as presented including any differing spaces or upper/lowercase differences.

- ⚠ Example (note that . means a space): name = John vs. name = john These would be different since J is different from j

Ignore differences in upper/lowercase

Ignore case differences when doing the comparison

- ⚠ Example (note that . means a space): name = John vs. name = john These would not be different since J is now not different from j

Ignore differences in leading/trailing whitespace

Remove any leading/trailing whitespace before doing the comparison

- ⚠ Example (note that . means a space): name = john vs. name = john... By default, john is different from john... but selecting this option would make john = john... because the ... would get removed prior to the comparison

Ignore differences in both case and whitespace

Ignore both case and whitespace as described above

Variable Name Tolerance Options

Treat variable names as-is (default)

Upper/lowercase, spaces, dots, and underscores mean variables are different

- ⚠ Example: Variable = Age would not be compared to Variable = age using this option

Treat dots, underscores, and spaces equivalent in variable names

Ignore dots, underscores, and spaces in variable names

- ⚠ Example: Variable = Age.dx would be compared to Age_dx if you select this option. By default, they would not be treated as the same variable

Ignore upper/lowercase in variable names

Ignore differences in upper/lowercase in variable names

- ⚠ Example: Variable = Age would be compared to Variable = age using this option

Ignore case and treat dots, underscores, and spaces equivalent in variable names

Ignore differences in dots, underscores, spaces, and upper/lowercase as described above

- ⚠ Example: Variable = Age.dx would be compared to Variable = age_dx using this option

- ℹ Required R Packages: arsenal

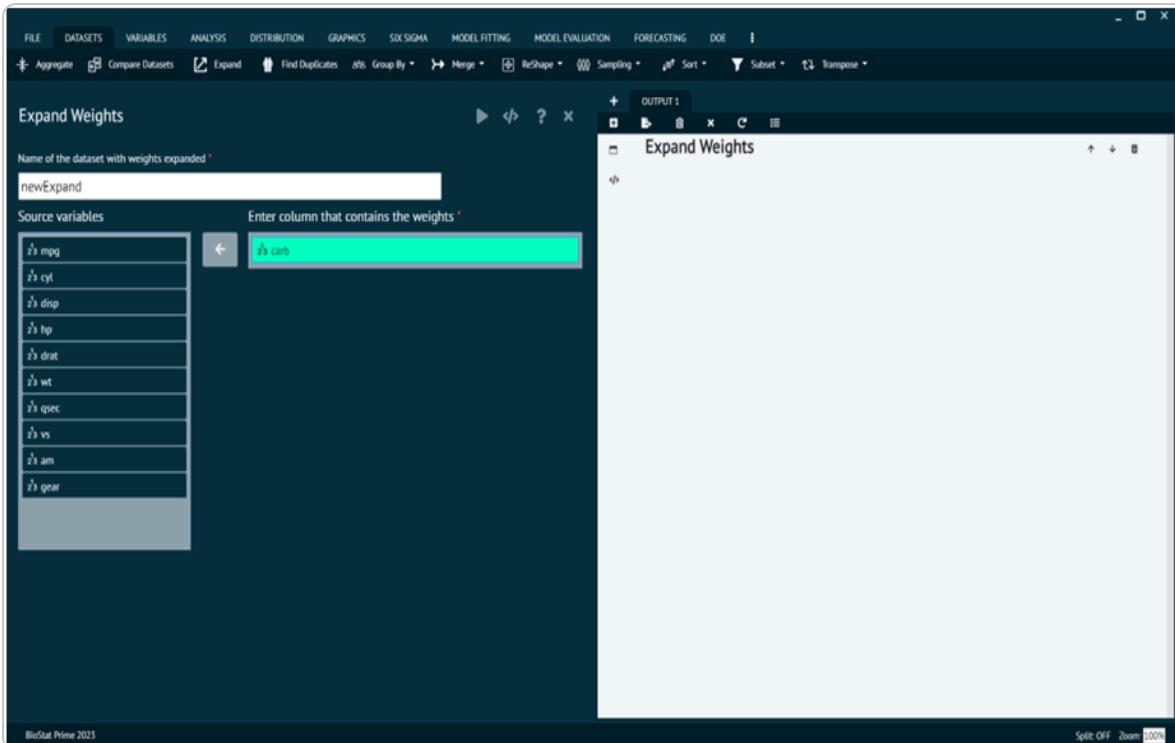
Expand

Creates a new dataset with rows expanded as per weights. In this dialog the weights refer to the dataset variable that contains the weights.

To expand weights user needs to follow the steps given below.

Steps

Load the dataset -> Click on the DATASET tab in main menu -> select EXPAND -> Once the dialog appears choose the Variable to be expanded -> Execute the dialog.



Before expanding weights.

The screenshot shows the BioStat Prime 2023 interface. On the left, the 'DATA' tab is active, displaying the 'mtcars' dataset. A red circle highlights the 'gear' column header. On the right, an 'OUTPUT 1' window titled 'Expand Weights' is open, showing the expanded weights for the 'gear' variable.

#	Name	Class	Type	Measure	Levels
1	mpg	numeric	double	scale	
2	cyl	numeric	double	scale	
3	disp	numeric	double	scale	
4	hp	numeric	double	scale	
5	drat	numeric	double	scale	
6	wt	numeric	double	scale	
7	qsec	numeric	double	scale	
8	vs	numeric	double	scale	
9	am	numeric	double	scale	
10	gear	numeric	double	scale	

alt text

The screenshot shows the BioStat Prime 2023 interface. On the left, the 'DATA' tab is active, displaying the 'mtcars' dataset. A red circle highlights the 'gear' column header. On the right, an 'OUTPUT 1' window titled 'Expand Weights' is open, showing the expanded weights for the 'gear' variable.

#	Name	Class	Type	Measure	Levels
1	mpg	numeric	double	scale	
2	cyl	numeric	double	scale	
3	disp	numeric	double	scale	
4	hp	numeric	double	scale	
5	drat	numeric	double	scale	
6	wt	numeric	double	scale	
7	qsec	numeric	double	scale	
8	vs	numeric	double	scale	
9	am	numeric	double	scale	
10	gear	numeric	double	scale	

alt text

After Expanding weights.

The screenshot shows the DataWeave application window. On the left, the 'mtcars' dataset is displayed as a table with columns: drat, wt, qsec, vs, am, gear, and carb. The 'carb' column has its last row circled in red. On the right, an 'OUTPUT 1' panel titled 'Expand Weights' is open, showing a single item labeled 'Expand Weights'.

alt text

This screenshot is similar to the one above, showing the 'mtcars' dataset in the main window and the 'Expand Weights' dialog in the output panel. In the 'mtcars' table, the entire row for 'carb' is circled in red.

alt text

The arguments used in executing the dialog are given as follows.

Arguments

1. Weights: The dataset variable that contains the weights.
2. data: The input data.frame or data.table.
3. newdata: The new dataset where the rows are replicated for the weights specified.

Find Duplicates

This dialog will find duplicates either by complete cases or by key variables.

Complete case duplicates are equal for every value for every variable.

Duplicates using key variables are duplicates defined only by equal values for specific variables, called "keys".

So, a duplicate row means the values are equal to a previous row.

i Duplicates are searched from the top to the bottom of the data set.

The screenshot shows the BioStat Prime 2023 software interface. The main window is titled 'Find Duplicates'. In the 'Source variables' list, 'mpg', 'disp', 'drat', 'wt', 'qsec', 'am', 'gear', and 'carb' are listed. In the 'Key Variables (optional)' list, 'cyl', 'hp', and 'vs' are selected. Under 'Create dataset with all rows associated with the duplicates', 'Dataset name' is set to 'allduprows'. Under 'Create dataset with original data and column indicating duplicates', 'Dataset name' is set to 'datadupvar' and 'Duplicate variable name' is set to 'duplicate'. Under 'Create dataset with all duplicates removed', 'Dataset name' is set to 'nodupdata'. The output window titled 'Find Duplicates' displays the results for the mtcars dataset. It shows 'KeyVariables' (cyl, hp, vs), 'Number of Rows, Duplicates, and Rows Associated with Duplicates for mtcars' (N: 32, Duplicates: 8, RowsAssociated.Duplicates: 15), and 'Frequency of Rows Associated with the Duplicates by Keys for mtcars' (cyl: 4, hp: 66, vs: 1, Freq: 2; cyl: 6, hp: 110, vs: 0, Freq: 2; cyl: 6, hp: 123, vs: 1, Freq: 2).

alt text

After finding the duplicates user can make 3 datasets, viz.

1. Dataset with all rows associated with the duplicates.

The screenshot shows a software interface with a menu bar at the top. The 'DATASETS' tab is selected. Below the menu, there are several icons: 'Aggregate', 'Compare Datasets', 'Expand', 'Find Duplicates', 'Group By', 'Merge', 'ReShape', and '...'. A red oval highlights the 'Find Duplicates' icon.

The main area displays a data grid. The columns are labeled '#', 'mtcars', 'allduprows', 'datadupvar', and 'nodupdata'. The 'allduprows' column contains binary values (0 or 1) indicating if each row has duplicates. The 'datadupvar' and 'nodupdata' columns are currently empty. The data grid contains 15 rows of car data from the 'mtcars' dataset, including columns for 'Cyl', 'hp', 'vs', 'mpg', 'disp', 'drat', and 'wt'.

At the bottom of the interface, there are tabs for 'DATA' (selected), 'VARIABLES', and 'R EDITOR'. To the right of the tabs are several small icons.

alt text

2. Dataset with original data and column indicating duplicates.

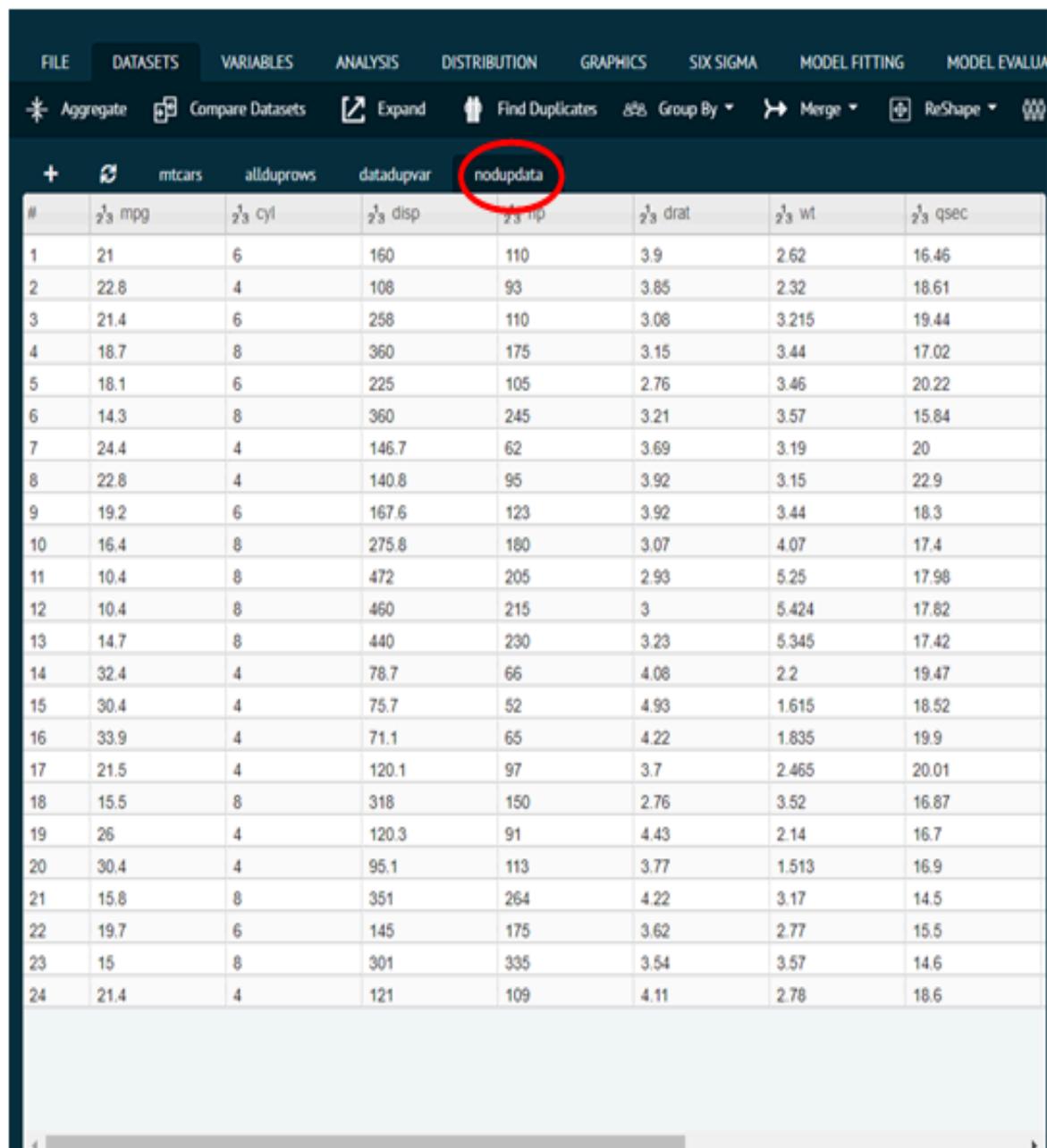
The screenshot shows the QMplus software interface with the following details:

- Toolbar:** FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION.
- Tool Buttons:** Aggregate, Compare Datasets, Expand, Find Duplicates, Group By, Merge, ReShape.
- Dataset View:** The dataset is titled "datadupvar". The columns are labeled #, mpg, cyl, disp, hp, drat, wt, and qsec. The rows are numbered from 1 to 28.
- Data:** The data is identical to the "mtcars" dataset, containing information about 28 different cars. The columns represent variables such as miles per gallon (mpg), number of cylinders (cyl), displacement (disp), horsepower (hp), ratio of weight to acceleration (drat), weight (wt), and time to go from 0 to 60 miles per hour (qsec).
- Bottom Navigation:** DATA, VARIABLES, R EDITOR, and various navigation icons.

A red circle highlights the dataset name "datadupvar" in the title bar, indicating that all duplicates have been removed.

alt text

3. Dataset with all duplicates removed.



The screenshot shows the SPSS software interface with the 'Datasets' tab selected. A dialog box titled 'Find Duplicates' is open. At the top of the dialog, there are several buttons: 'Aggregate', 'Compare Datasets', 'Expand', 'Find Duplicates' (which is highlighted with a red circle), 'Group By', 'Merge', 'ReShape', and 'View'. Below these buttons is a table with the following columns: '#', '_1 mpg', '_1 cyl', '_1 disp', '_1 hp', '_1 drat', '_1 wt', and '_1 qsec'. The table contains 24 rows of data from the mtcars dataset. The data includes various car specifications like engine displacement, horsepower, and weight.

#	_1 mpg	_1 cyl	_1 disp	_1 hp	_1 drat	_1 wt	_1 qsec
1	21	6	160	110	3.9	2.62	16.46
2	22.8	4	108	93	3.85	2.32	18.61
3	21.4	6	258	110	3.08	3.215	19.44
4	18.7	8	360	175	3.15	3.44	17.02
5	18.1	6	225	105	2.76	3.46	20.22
6	14.3	8	360	245	3.21	3.57	15.84
7	24.4	4	146.7	62	3.69	3.19	20
8	22.8	4	140.8	95	3.92	3.15	22.9
9	19.2	6	167.6	123	3.92	3.44	18.3
10	16.4	8	275.8	180	3.07	4.07	17.4
11	10.4	8	472	205	2.93	5.25	17.98
12	10.4	8	460	215	3	5.424	17.62
13	14.7	8	440	230	3.23	5.345	17.42
14	32.4	4	78.7	66	4.08	2.2	19.47
15	30.4	4	75.7	52	4.93	1.615	18.52
16	33.9	4	71.1	65	4.22	1.835	19.9
17	21.5	4	120.1	97	3.7	2.465	20.01
18	15.5	8	318	150	2.76	3.52	16.87
19	26	4	120.3	91	4.43	2.14	16.7
20	30.4	4	95.1	113	3.77	1.513	16.9
21	15.8	8	351	264	4.22	3.17	14.5
22	19.7	6	145	175	3.62	2.77	15.5
23	15	8	301	335	3.54	3.57	14.6
24	21.4	4	121	109	4.11	2.78	18.6

alt text

Summaries of the options in the Find Duplicates dialog is provided below.

Key Variables:

Specify optional key variables that define the duplicates.

- If no key variables are selected, complete case duplicates will be searched for.

Create dataset with all rows associated with the duplicates:

This will create a dataset of all duplicate rows and the first instance of each row corresponding to each duplicate. The output dataset will be sorted by all the variables in the complete duplicate case and by the key variables in the key variable case. The key variables will also be moved to the beginning of the output data set. The Dataset name field can be used to name this output data set.

Create dataset with original data and column indicating duplicates:

This will create a dataset including all the original data plus an additional column indicating the duplicate rows (0=not duplicate, 1=duplicate). The Dataset name field can be used to name this output data set. The Duplicate variable name field can be used to name this additional column.

Create dataset with all duplicates removed:

This will create a dataset that removes all the duplicate rows (either complete case or by key variables) where the duplicates are searched from top to bottom in the data set. This means all 2nd, 3rd, etc. instances of the rows will be removed. The Dataset name field can be used to name this output data set.

- i Required R Packages: dplyr, arsenal

Group By

This section of the dataset tab aids the user to split a loaded dataset and remove the split if a split is already set on dataset. It splits the data into groups based on the factors selected, once the dataset is split, the analysis user selects is performed independently for each split.

- A** For example if user runs a crosstabulation analysis or a hypothesis test, this analysis is performed independently for each split (the output of the analysis is also generated separately for each split).

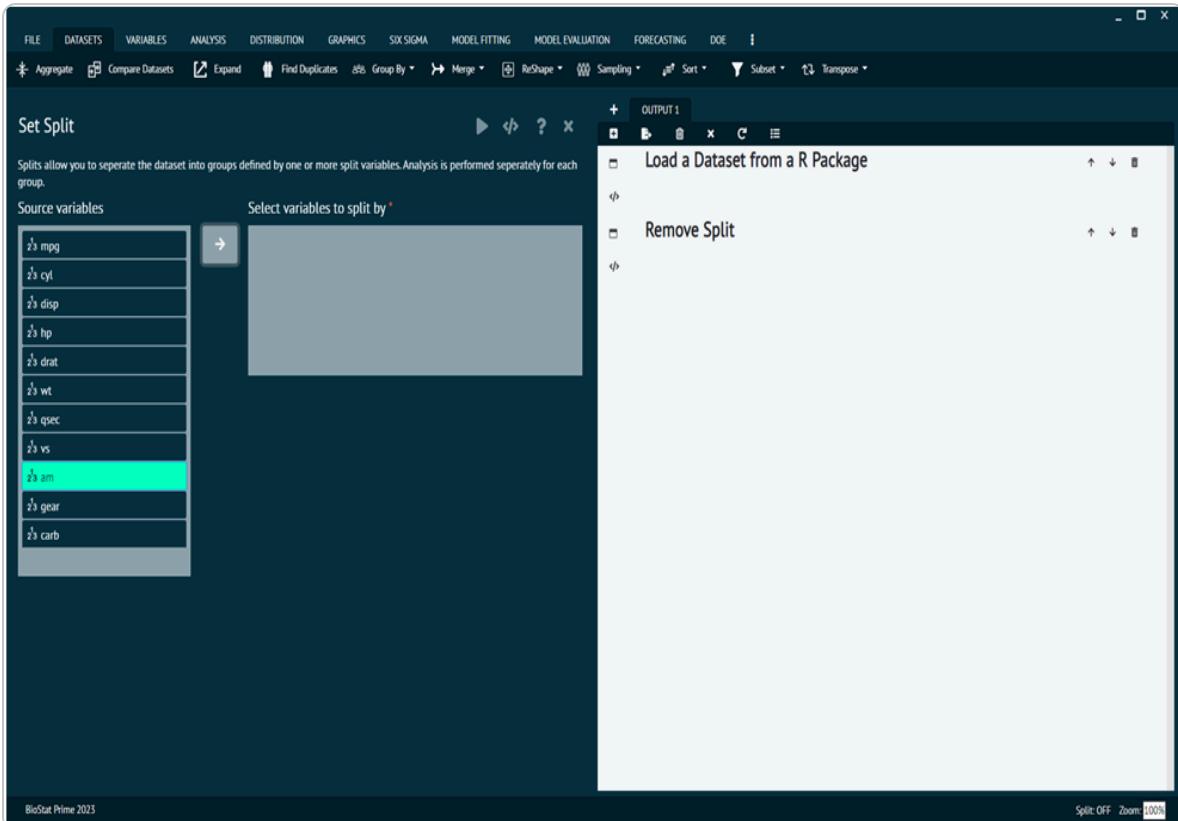
This tab has two options, viz.

Split

Splits allow you to separate the dataset into groups defined by one or more split variables. Analysis is performed separately for each group.

Remove Split

Removes the split (if a split is set on the dataset).



alt text

The arguments used in executing the dialog are given as follows.

⚠ Arguments

1. col.names: These are the column names/variable names that you want to split the dataset by, e.g. col.names =c("var1", "var2").
2. datasetnameorindex: this is the name of the index.
3. removeall.splits: TRUE splits are removed, FALSE splits are added.

Merge

Merge datasets will help user join 2 datasets together. User need to specify one or more variables in the active dataset and in the selected target dataset that you want the join to be performed on.

- i** The results will be saved in a new dataset.

A Merge Options

1. `inner_join`: return all rows from x where there are matching values in y, and all columns from x and y. If there are multiple matches between x and y, all combination of the matches are returned.
2. `left_join`: return all rows from x, and all columns from x and y. Rows in x with no match in y will have NA values in the new columns. If there are multiple matches between x and y, all combinations of the matches are returned.
3. `right_join`: return all rows from y, and all columns from x and y. Rows in y with no match in x will have NA values in the new columns. If there are multiple matches between x and y, all combinations of the matches are returned.
4. `full_join`: return all rows and all columns from both x and y. Where there are not matching values, returns NA for the one missing.
5. `semi_join`: Keep all rows in first dataset with a match in second dataset
6. `anti_join`: Keep all rows in first dataset without a match in second dataset

This section id dataset tab has 4 options that are explained as follows.

Merge

Merge datasets will help user to join 2 datasets together. By default, this dialog will look for common variable names within the 2 datasets and merge on the full set of common

variables. To perform this operation in BioStat Prime user needs to follow the steps given below.

Load the datasets -> click on the DATASET tab in main menu -> select MERGE -> select MERGE from the drop-down -> Once the dialog appears choose the Variables from each dataset -> add them to join the mapping -> Execute the dialog.

FILE DATASETS VARIABLES ANALYSIS DISTRIBUTION GRAPHICS SIX SIGMA MODEL FITTING MODEL EVALUAT

Aggregate Compare Datasets Expand Find Duplicates Group By Merge ReShape

Merge Datasets

Variables from the active (left) dataset: new

1b
23 mpg
23 cyl
23 disp
23 hp
23 drat
23 wt
23 qsec
23 vs
23 am
23 gear
23 carb
23 abhev

Select a dataset to join new with

mtcars

Variables from the selected (right) dataset: mtcars

23 mpg
23 cyl
23 disp
23 hp

Enter the name of the merged dataset.

ll

Join mapping*

Select a variable from the active dataset and a variable in the selected dataset and click Add

Add Delete

hp=mpg

Merge Options

- Left Join (Keep only rows in first (left) dataset)
- Right Join (Keep only rows in second (right) dataset)
- Inner Join (Keep rows common to both datasets)
- Full Join (Keep all rows in either dataset)
- Semi Join (Keep all rows in first (left) dataset with a match in second (right) dataset)
- Anti Join (Keep all rows in first (left) dataset without a match in second (right) dataset)

By default, .x and .y will be used as suffixes for common variables. If you want to change them, enter them here separated by a comma, e.g. 1,2. Note that any . will be replaced by a _ in the output data set

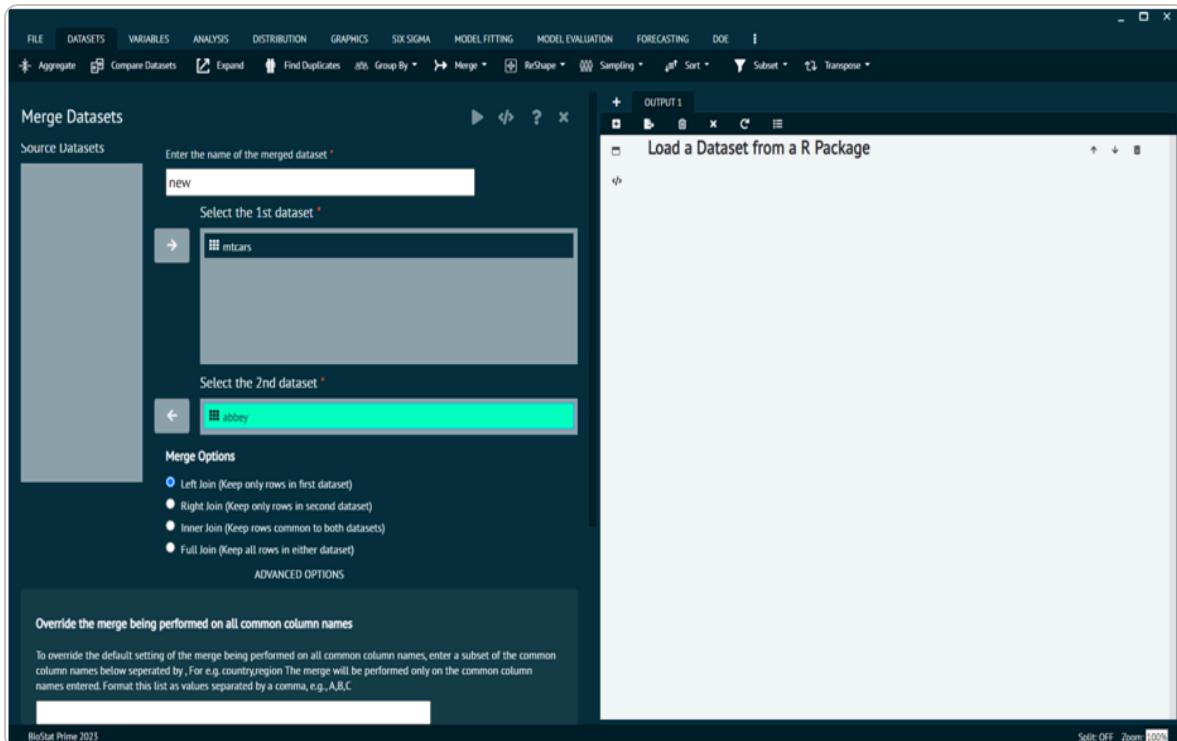
.x,.y

BioStat Prime 2023

alt text

- i** R Package Required: dplyr

Merge (legacy)



alt text

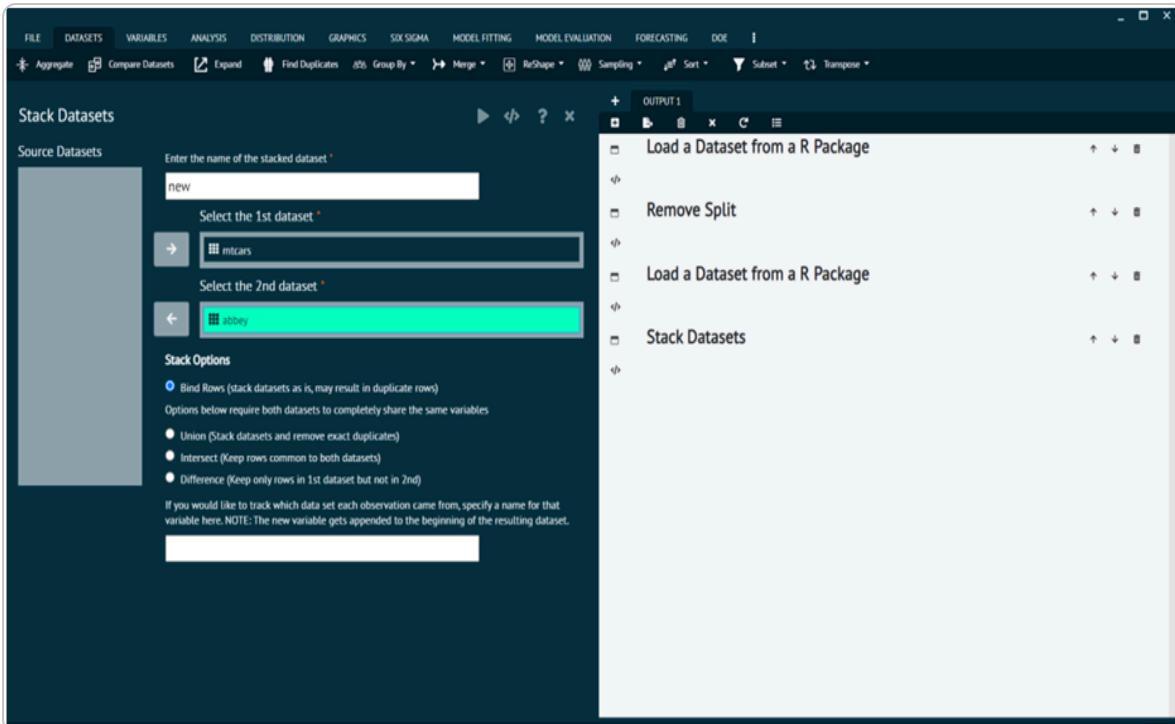
Stack

This dialog will help user to stack 2 datasets on top of each other. User can select one of the following options. Steps

1. Bind Rows: Stacks the 2 datasets exactly as they are. If a variable name is common to both datasets, values will fill in as user expects. If a dataset A contains a variable say var1 that is not present in the other dataset B, NA's will appear in variable var1 for all rows that correspond to dataset B. All options below require that both datasets share the same variables.
2. Union: stacks the datasets and removes duplicates
3. Intersect: keeps rows common to both
4. Difference: Keeps rows in 1st dataset, not in 2nd Depending on the option selected, the functions bind_rows, union, intersect and setdiff in the package

dplyr are called.

- i** User can optionally track which dataset the original observation came from. The dataset ID (1st/2nd) is appended to the beginning of the dataset that contains the results.



alt text

FILE DATASETS VARIABLES ANALYSIS DISTRIBUTION GRAPHICS SIX SIGMA MODEL FITTING MODEL EVALUATE

Aggregate Compare Datasets Expand Find Duplicates Group By Merge ReShape

+ mtcars abey new

1 wt	2 qsec	3 vs	4 am	5 gear	6 carb	7 abey
2.8	16	0	0	3	3	
2.4	17.88	0	0	3	4	
3.424	17.82	0	0	3	4	
3.345	17.42	0	0	3	4	
2.2	19.47	1	1	4	1	
1.615	18.52	1	1	4	2	
1.835	19.9	1	1	4	1	
2.465	20.01	1	0	3	1	
3.52	16.87	0	0	3	2	
3.435	17.3	0	0	3	2	
3.84	15.41	0	0	3	4	
3.845	17.05	0	0	3	2	
1.935	18.9	1	1	4	1	
2.14	16.7	0	1	5	2	
1.513	16.9	1	1	5	2	
3.17	14.5	0	1	5	4	
2.77	15.5	0	1	5	6	
3.57	14.6	0	1	5	8	
2.78	18.6	1	1	4	2	
						5.2
						6.5
						6.9
						7
						7
						7.4
						8
						8

alt text

Steps

Merge Update

Description Update merge updates a dataset with values from a second dataset based on exact variable name matching for observations with matching join

mapping variable values. You need to specify one or more variables in the active dataset and in the selected target dataset that you want the join to be performed on. The results will be saved in a new dataset.

Merge Options

Update variables in first (left) dataset with matches from second (right) dataset, insert non-matches:

This is a combination of updating variables in the left dataset for matches and creating new rows for unmatched rows.

Update variables in first (left) dataset with matches from second (right) dataset, ignore non-matches:

This only updates existing variables in the left datasets for matches. Unmatched rows are ignored.

Only insert non-matches in first (left) dataset:

This leaves intact all matching rows in the left dataset. Only non-matching rows from the right dataset are added to the left dataset.

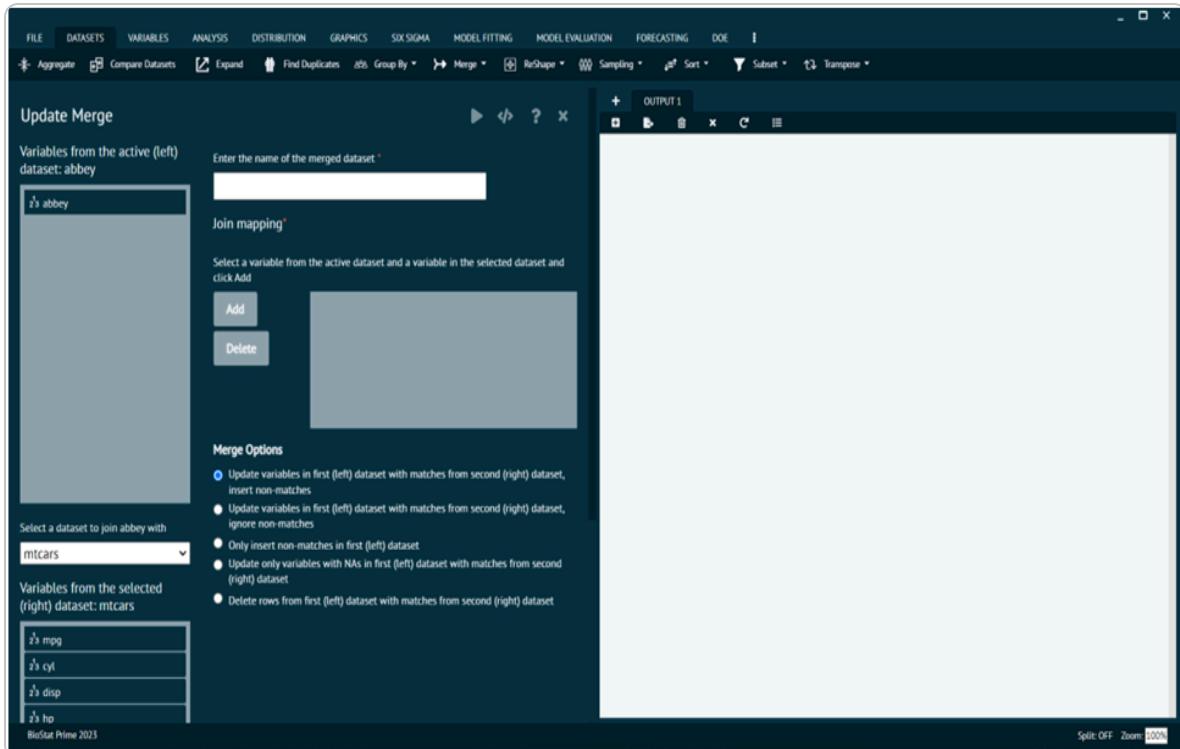
Update only variables with NAs in first (left) dataset with matches from second (right) dataset, ignore non-matches:

This updates rows that match, but only when the values in the left dataset are NA (i.e. are missing values).

Delete rows from first (left) dataset with matches from second (right) dataset:

This only deletes rows from the left dataset that match rows in the right dataset.

 R Packages Required: dplyr



alt text

ReShape

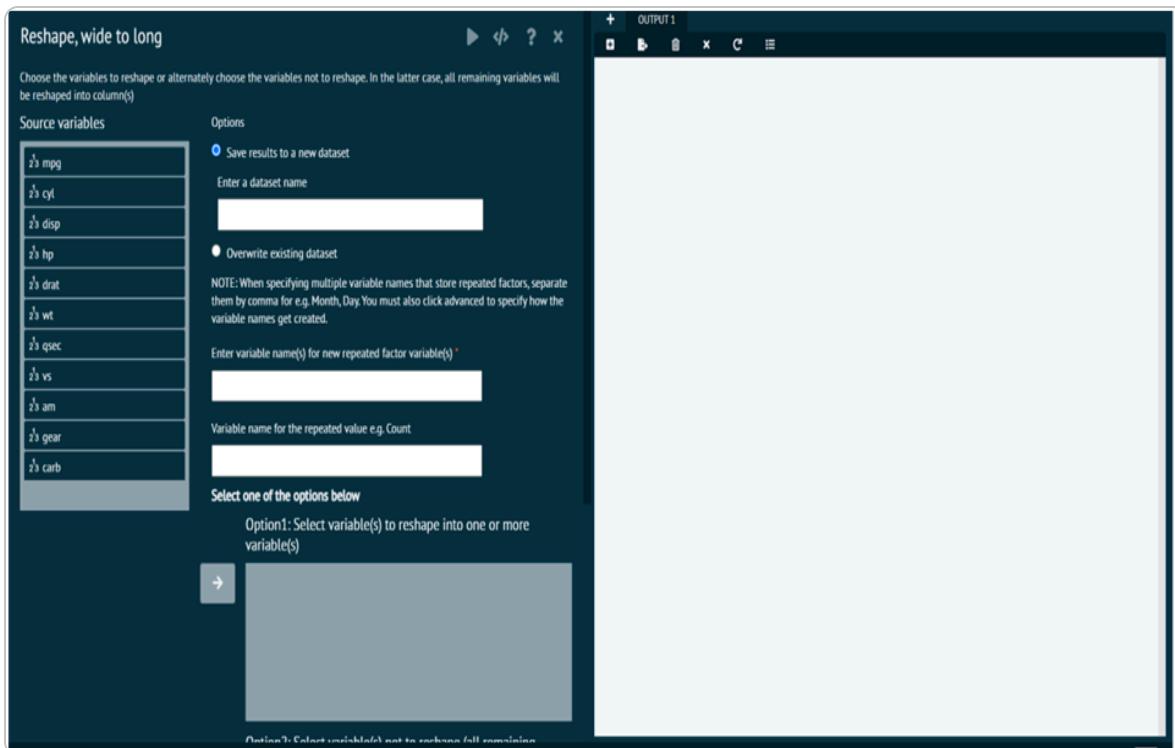
This section of the dataset tab aids the user to Reshape the loaded datasets. This tab has 2 options, viz.

Reshape wide to long

Reshape wide to long option takes a wide dataset and converts it to a long dataset by converting columns into key value pairs, Pivot_longer takes multiple columns and collapses into key-value pairs, duplicating all other columns as needed. User can use pivot_longer() when user notices that he has columns that are not variables.

User can choose the variables to reshape or alternately choose the variables not to reshape from wide dataset to long dataset. In the latter case, all remaining variables will be reshaped into column(s).

- i** When specifying multiple variables for the repeated factor(s) separate them by
,



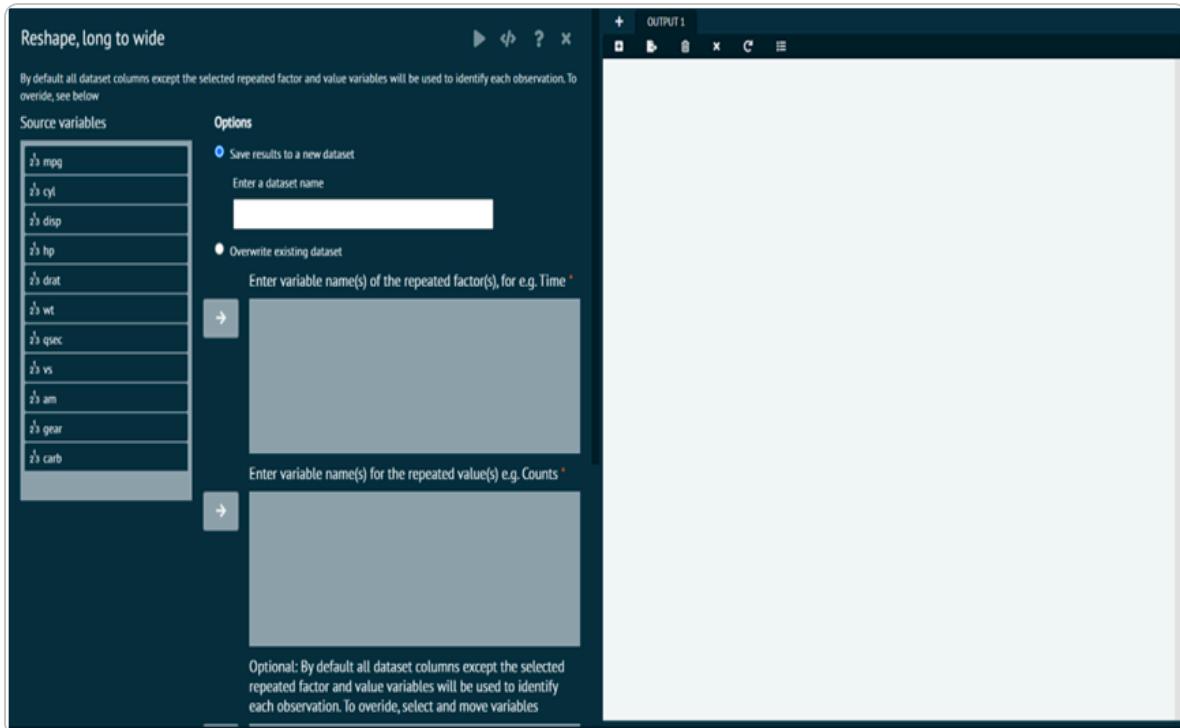
alt text

- R package Required: tidyverse

Reshape Long to Wide

User here chooses to reshape from longer dataset to wider dataset. This option takes a wide dataset and converts it to a long dataset by converting (widening) columns. Pivot_wider "lengthens" data, increasing the number of rows and decreasing the number of columns. User can use pivot_wider when user has variables/columns whose values need to be in rows.

- NOTE: When one repeated factor is specified, new variable names are prefixed with the name of the repeated factor. When multiple repeated factors are specified, they are prefixed by the name of the value variable
- By default, all dataset columns except the selected repeated factor and value variables will be used to identify each observation.



alt text

Sampling

Sample takes a sample of the specified size from the elements of x using either with or without replacement.

Random Split

If x has length 1, is numeric (in the sense of is.numeric) and $x \geq 1$, sampling via sample takes place from 1:x.

Random Split

Enter the name of the training dataset *

Enter the name of the test dataset *

Enter the split percentage 80

Should sampling be with replacements

Set seed * 12345

alt text

- i** x: Either a vector of one or more elements from which to choose, or a positive integer.

Sample Data

Sample Data takes a random sample of the rows from the existing dataset. Samples a % of rows or a specified number of rows with or without replacement. Saves the result to a new dataset or overwrites the existing dataset.

Sample Data

Dataset options

Save results to a new dataset
Enter a name of a dataset

Overwrite dataset

Sampling options

Specify the percentage of the dataset to sample
Enter the percentage

Specify the number of rows to select
Enter the number of rows

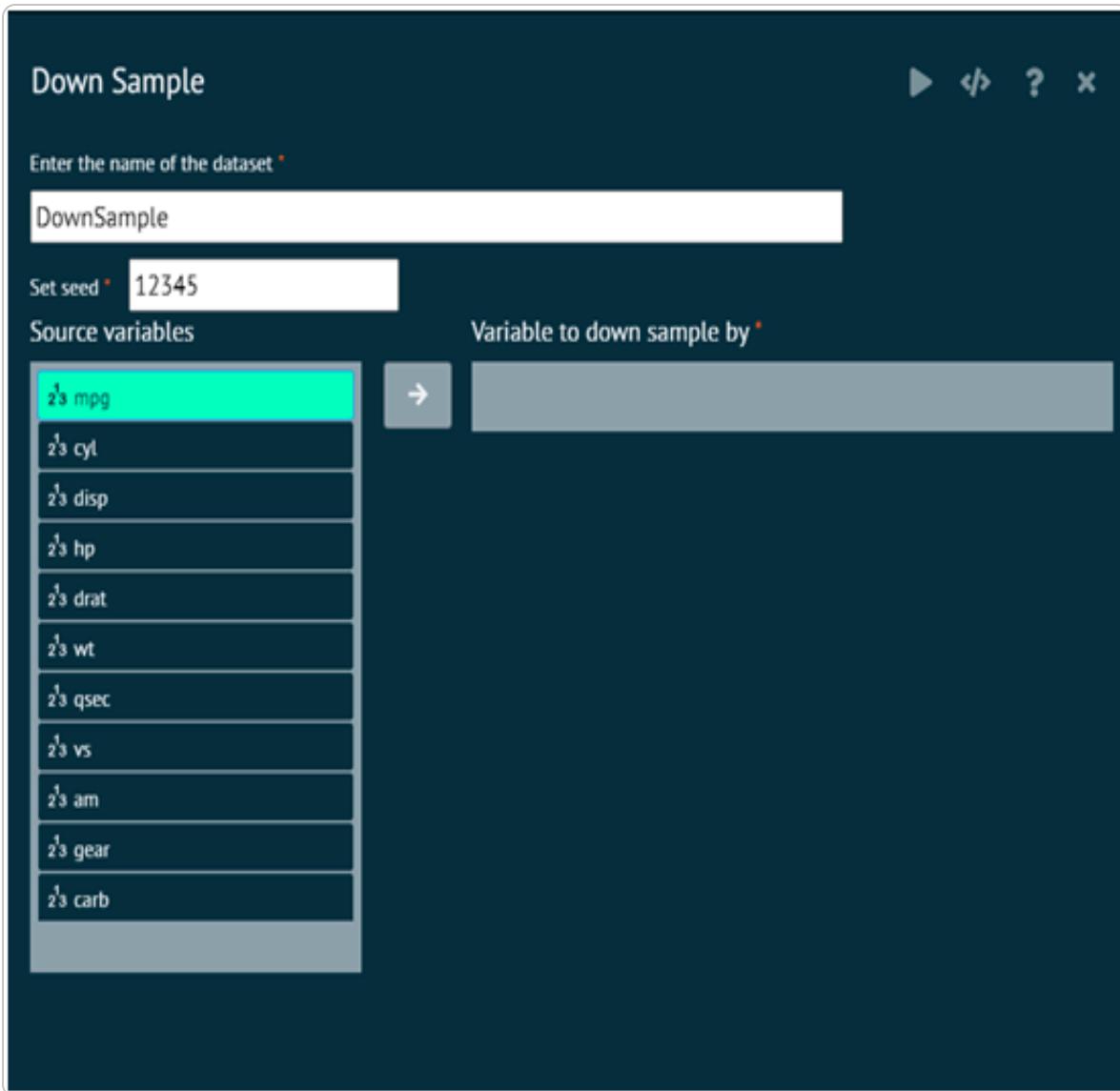
Sample with replacement
Optionally enter a variable name or formula for weights

Optionally set a seed for data reproducibility

alt text

Down Sample

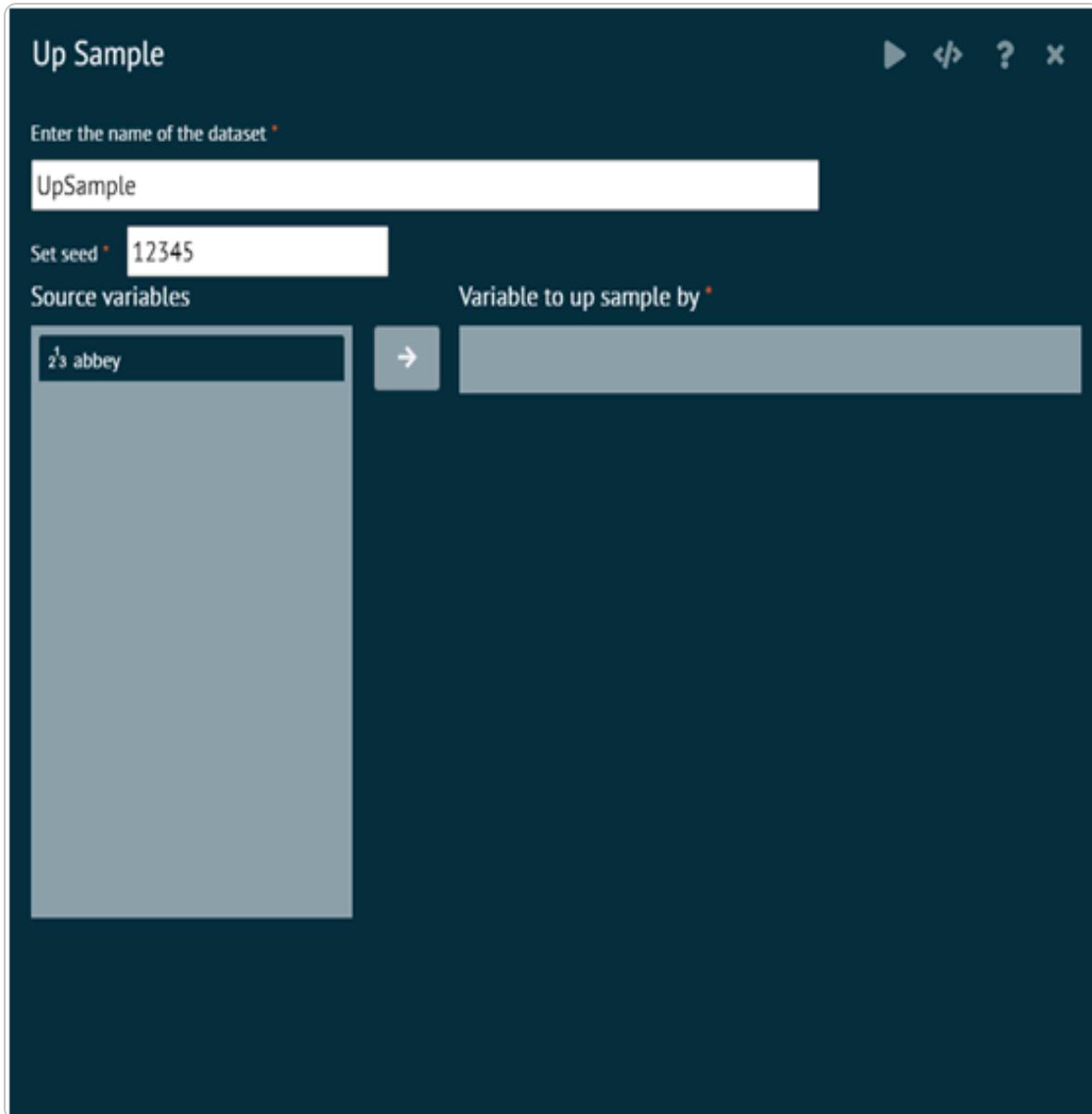
Down-Sampling Imbalanced Data. DownSample will randomly sample a data set so that all classes have the same frequency as the minority class.



alt text

Up Sample

Up-Sampling Imbalanced Data. upSample samples with replacement to make the class distributions equal.



alt text

Stratified Split

A series of test/training partitions are created using `createDataPartition` while `createResample` creates one or more bootstrap samples. `createFolds` splits the data into k groups while `createTimeSlices` creates cross-validation split for series data. `groupKFold` splits the data based on a grouping factor.

Stratified Split



Enter the name of the training dataset *

Enter the name of the test dataset *

Enter the split percentage * 80

Set seed 12345

Source variables

Variable to construct stratified samples from *

23 mpg
23 cyl
23 disp
23 hp
23 drat
23 wt
23 qsec
23 vs
23 am
23 gear
23 carb



alt text

Sort

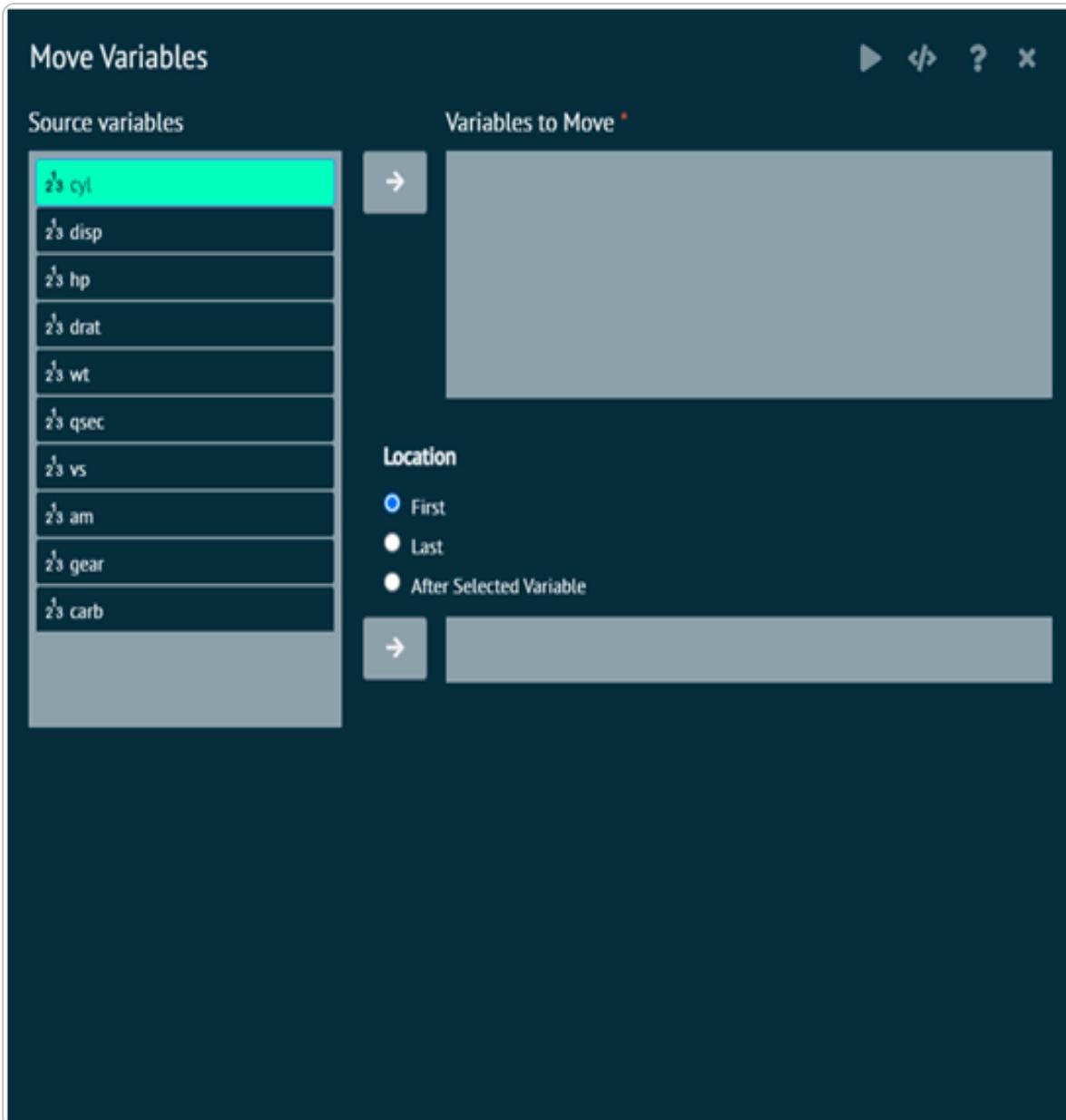
Move Variables

This will move variables to a specified location in the data set.

Variables to Move: Variables to move to a different location. They will be placed in the order specified in this box.

Location: Location in the data set to move the variables. First places the variables at the beginning of the data set. Last places the variables at the end of the data set. After Selected Variable places the variables after this variable in the data set.

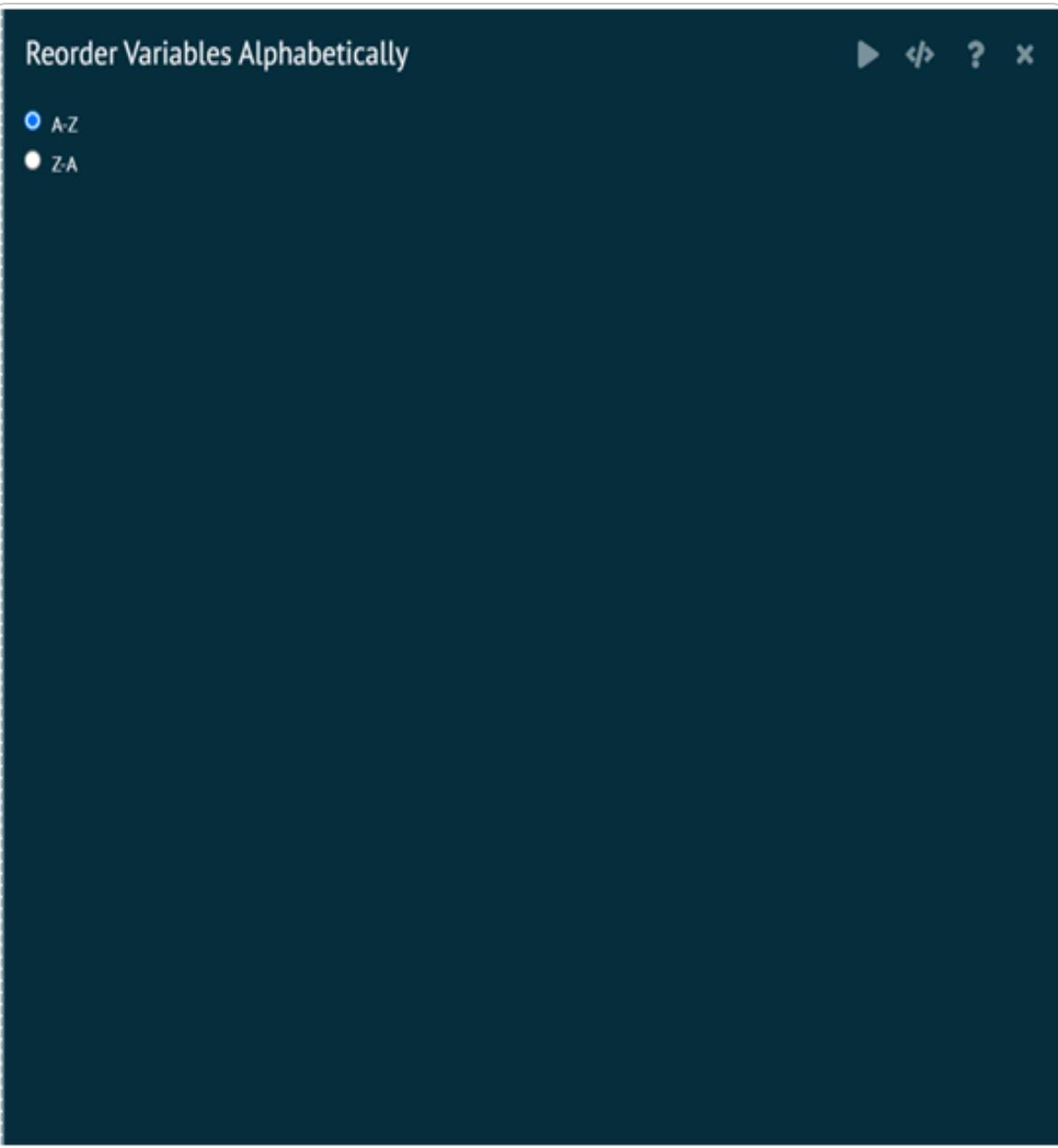
 Required R Packages: dplyr



alt text

Reorder Variables

Re-order variables in the dataset in alphabetical order. User uses the sort function to sort the names of the columns/variables in the dataset and the select function in the package dplyr to select the column names in the correct alphabetical order.



alt text

Sort Dataset

To sort a variable in descending order, user must select desc from the sort options and move the variable user wants to sort by.

Sort Dataset

▶ ⌂ ✕

Source variables

23 mpg
23 cyl
23 disp
23 hp
23 drat
23 wt
23 qsec
23 vs
23 am
23 gear
23 carb

Sort Options *

Specify a sort order, select asc for ascending, desc for descending

asc



**To sort a variable in descending order, you must select desc from the sort options and move the variable you want to sort by.

ONLY WHEN YOU SEE DESC(VARIABLE NAME) IN THE LIST IS THE VARIABLE SORTED IN DESCENDING ORDER

Show results in output

alt text

Subset

Subset

Subset datasets/dataframe. Returns a subset of the dataframe/dataset. User can specify the columns/variables that user wants in the smaller dataset. User can also specify selection criteria to be applied against each row of the dataframe.

Subset Dataset

You can choose to save the results in a new dataset or overwrite the existing dataset

Source variables	Options
mpg	<input checked="" type="radio"/> Save results to a new dataset Enter a dataset name <input type="text"/>
cyl	<input type="radio"/> Overwrite existing dataset
disp	<input type="radio"/> Display results in the output window
hp	<input type="checkbox"/> Select distinct cases
drat	<input type="checkbox"/> Remove unused factor levels
wt	Select variables to include in subsetted dataset *
qsec	<input type="button" value="→"/>
vs	
am	
gear	
carb	

Subsetting criteria is applied against each row, see examples below.

- 1: Select rows where var 1 is non empty and var2 is empty specify:
`is.na(var1) & is.na(var2)`
- 2: Select rows where var1 > 30 and var 2 is Male specify:
`var1>30 & var2=='Male'`
- 3: Complex and or criteria specify:
`(var1 !=10 & var2>20) | var3==40`
- 4: Pattern match (xxx) or an exact match (abc) specify:
`(grepl("xxx",var1) ==TRUE) | var1=="abc"`
- 5: Match a substring by position specify: `substr(var1,2,4) == "abc"`

alt text

Subset by Position

This section of Subset tab, subsets a dataset according to row position.

Specify New Dataset Name: Dataset name where the subsetted data will be stored

Variables to Sort By First: Variables used to sort the rows before any subsetting is undertaken. This only will affect options that select the number of rows, e.g. First/Last N Rows, First/Last Proportion of Rows, and Specify Row Numbers. It will always be in ascending order.

Groups to Subset Within: Specifying no variables will subset according to the row position of the entire dataset. Specifying variables will subset according to the row position within groups defined by all combinations of values for the specified variables.

Subset Type

First N Rows: Keeps the first N rows of the dataset overall or within groups

Last N Rows: Keeps the last N rows of the dataset overall or within groups

Rows with Lowest N Values for a Variable: Keeps the rows that have the lowest ordered values for a specified variable overall or within groups. For example, specifying 10 would keep the rows with the lowest 10 values for a variable.

Rows with Highest N Values for a Variable: Keeps the rows that have the highest ordered values for a specified variable overall or within groups. For example, specifying 10 would keep the rows with the highest 10 values for a variable.

First Proportion of Rows: Keeps the rows in the top proportion of the dataset overall or within groups. For example, specifying .10 would keep the top 10% of the dataset according to the total number of rows.

Last Proportion of Rows: Keeps the rows in the bottom proportion of the dataset overall or within groups. For example, specifying .10 would keep the bottom 10% of the dataset according to the total number of rows.

Rows within Lowest Percentile for a Variable: Keeps the rows in the lowest percentile for a specified variable, overall or within groups. For example, specifying .10 would keep the lowest 10th percentile for a variable (minimum to the 10th percentile).

Rows within Highest Percentile for a Variable: Keeps the rows in the highest percentile for a specified variable, overall or within groups. For example, specifying .10 would keep the highest 10th percentile for a variable (90th percentile to the maximum).

Specify Row Numbers: Keeps the exact numbered rows specified. For example, specifying 1,3,5 would keep the first, third, and fifth rows. Specifying 20:30 would keep rows 20 to 30. Specifying seq(2,10,by=2) would keep the even numbered rows up to the 10th row.

Include Tied Values: Specifies whether tied values should be included or not. For example, if you want the rows for the lowest 10 values of a variable and the 10th lowest value appears more than once, including the tied values will keep all rows that equal the duplicated value.

- Note that specifying negative values for N or the proportion removes the corresponding rows from the dataset. For example, specifying -10 for the First N Rows would remove the first 10 rows. Specifying -.10 for the First Proportion of Rows, would remove the first 10% of the rows.

- R Packages Required: dplyr



alt text

Subset by Logic

Returns a subset of the dataset. User can specify the columns/variables that user wants in the smaller dataset. User can also specify selection criteria to be applied against each row of the dataframe.

Save results to a new dataset: Specify a new dataset to store the subsetted data

Overwrite existing dataset: This saves the subsetted dataset to the existing dataset name

Display results in the output window: This prints the subsetted dataset in the output window only. The subsetted dataset is not saved in a dataset.

Selected distinct cases: This removes duplicates from the subsetted dataset. All variables have to be the same to be considered a duplicate.

Remove unused factor levels: This deletes factor levels that were excluded by the subset (i.e. no longer appear in the data).

Select variables to include in subsetted dataset: This allows the user to select specific columns they want to include in the dataset. If any are specified, then only the specified variables will show up in the subsetted dataset. If no variables are specified, then all variables will be kept.

Enter subsetting criteria: Specify variable logic that will be used to filter the rows of the dataset.



R Package :dplyr

Subset Dataset by Logic

▶ ⌂ ? ×

You can choose to save the results in a new dataset or overwrite the existing dataset

Source variables

z3 mpg
z3 cyl
z3 disp
z3 hp
z3 drat
z3 wt
z3 qsec
z3 vs
z3 am
z3 gear
z3 carb

Options

- Save results to a new dataset

Enter a dataset name

- Overwrite existing dataset
- Display results in the output window
- Select distinct cases
- Remove unused factor levels

Select variables to include in subsetted dataset



Subsetting criteria is applied against each row, see examples below.

1: Select rows where var 1 is non empty and var2 is empty specify:

`!is.na(var1) & is.na(var2)`

2: Select rows where var1 > 30 and var 2 is Male specify:

`var1>30 & var2=='Male'`

3: Complex and or criteria specify:

`(var1 !=10 & var2>20) | var3==40`

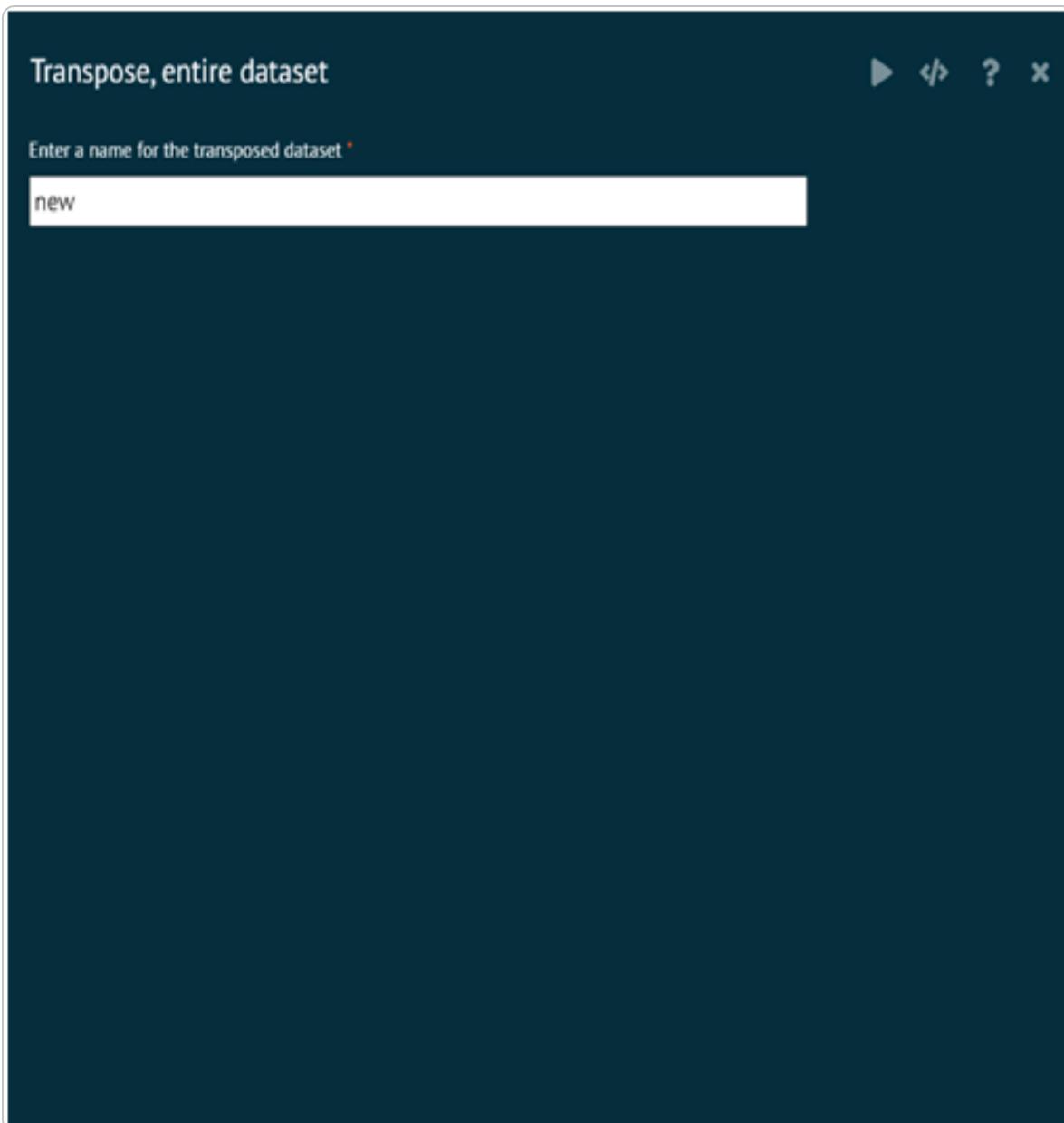
4: Pattern match (xxx) or an exact match (abc) specify:

alt text

Transpose

Transpose entire dataset

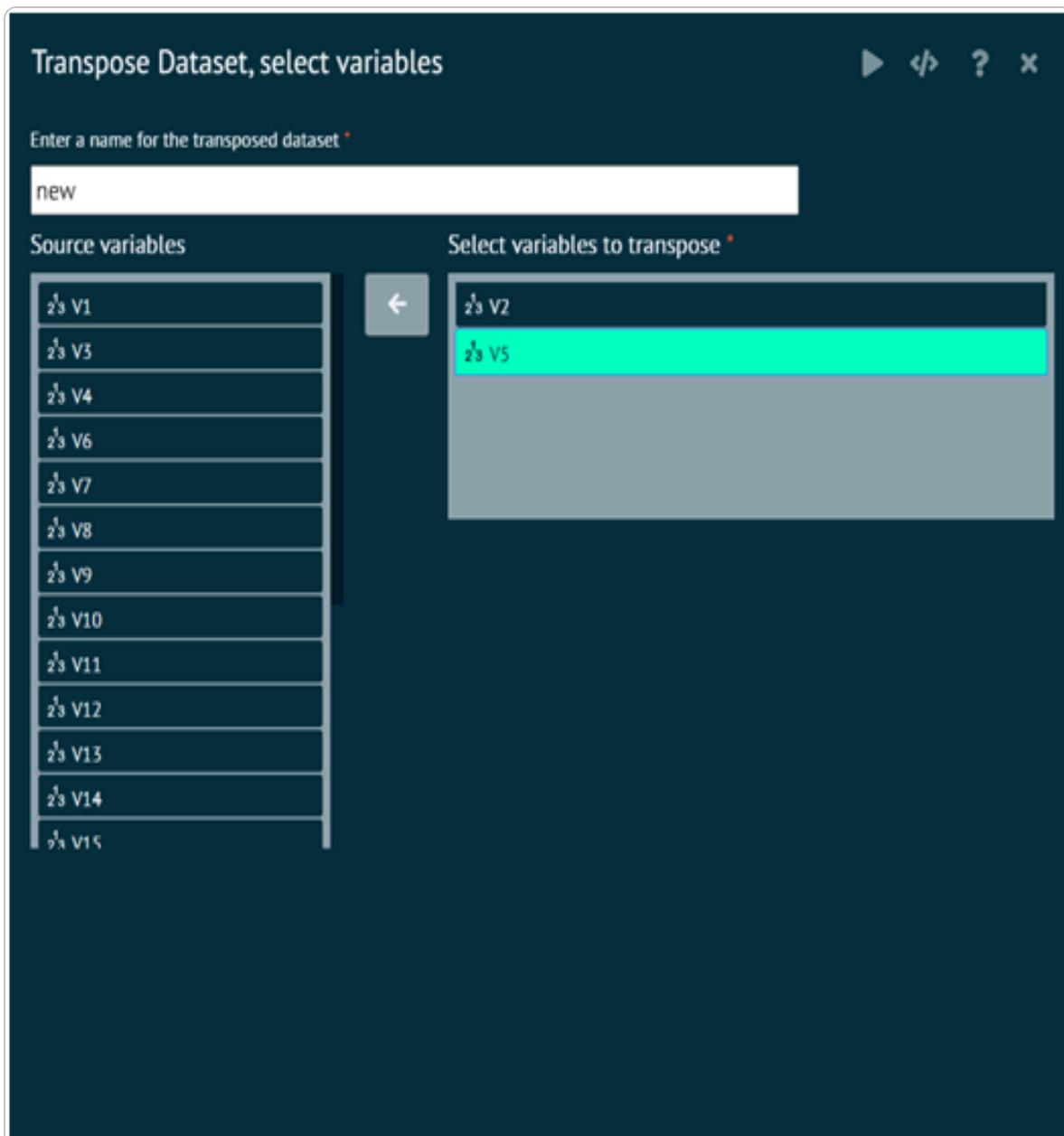
Invokes the transpose function in the base package that transposes the dataset. User have to specify the name of the dataset that stores the transposed dataset. The new transposed dataset is displayed in the grid.



alt text

Transpose dataset, select variables

Invokes the transpose function in the base package that transposes the variables selected and stores the results in the new dataset. User have to specify the name of the dataset that stores the transposed dataset. The new transposed dataset is displayed in the grid.



alt text

Variable

This section of the main menu gives access to the variable manipulation commands. It contains various operations that can be performed on the variables, i.e ;

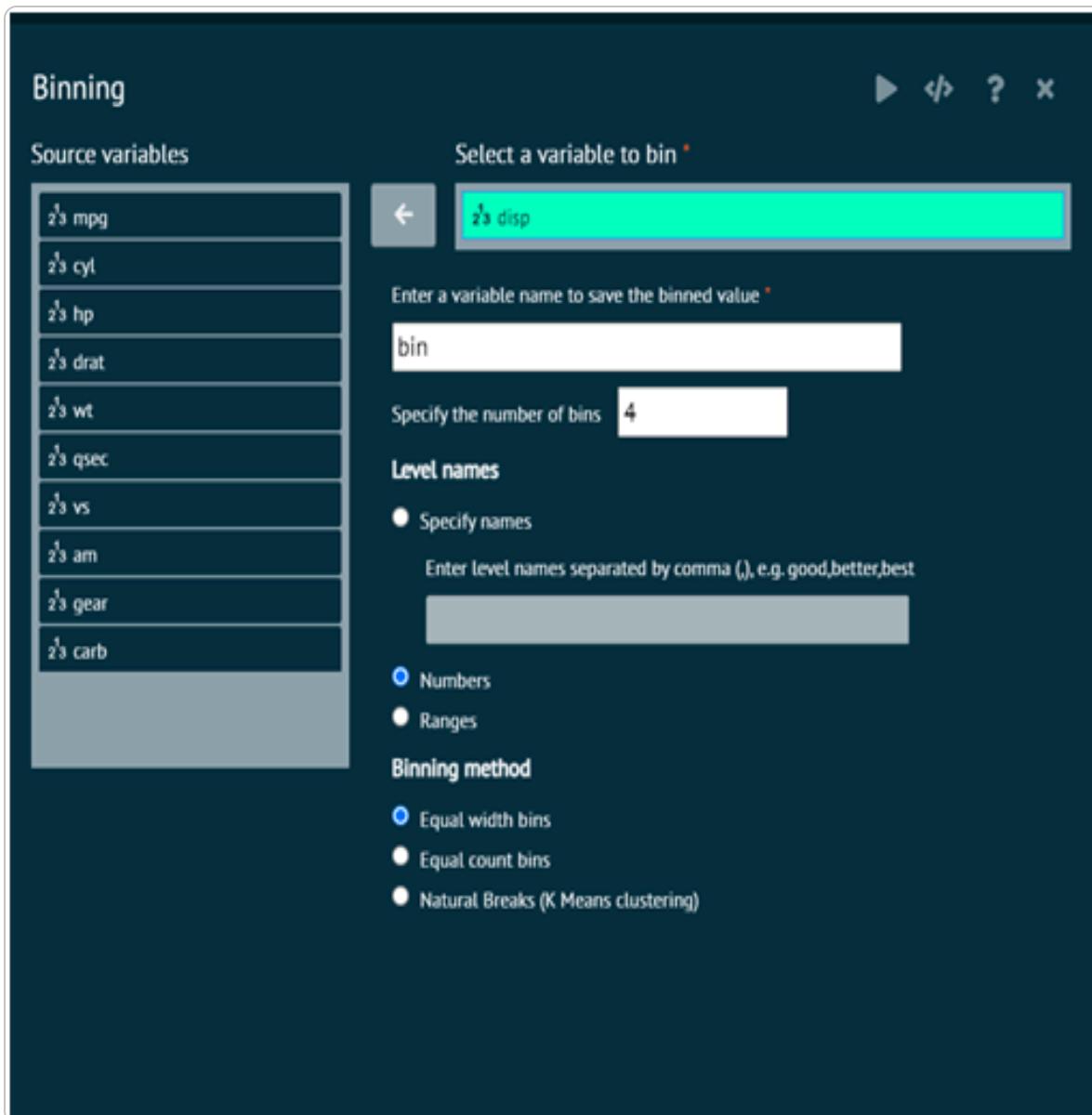
Bin, Box Cox, Compute, Concatenate, Convert, Date Order Check, Delete, Factor Levels, ID Variable, Lag or Lead Variable, Missing values, Rank, Recode, Standardize, Transform.

The above-mentioned functions are discussed in detail in the up-coming section.

Bin

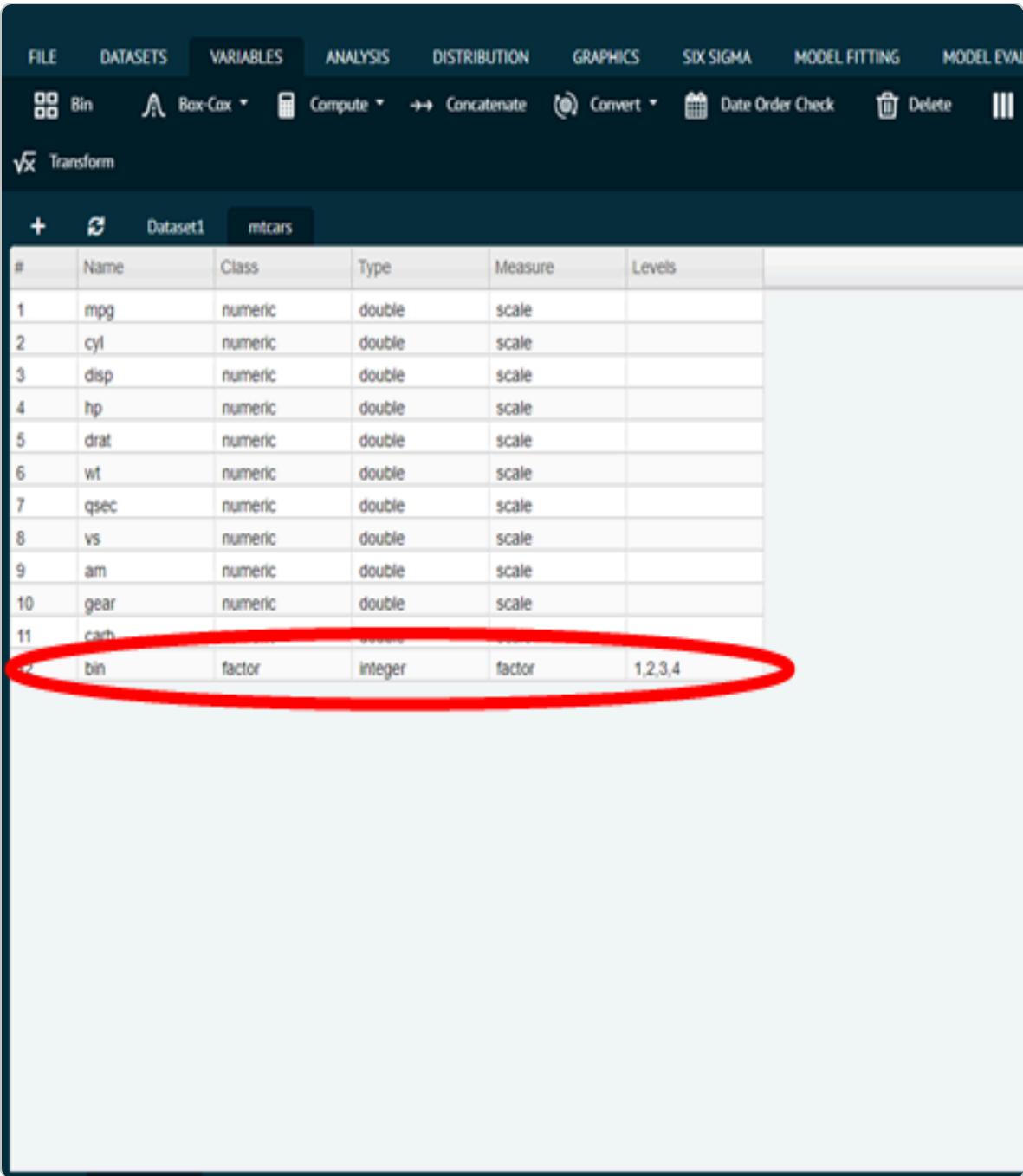
Bin function of variable menu selects a variable from the selected dataset to perform binning operation on it. As a result of which the binned value is stored in another variable whose name is determined by the user.

Variable selected to be binned.



alt text

Variable binned.



The screenshot shows a software interface with a menu bar at the top. The menu items include FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, and MODEL EVAL. Below the menu is a toolbar with various icons: Bin, Box-Cox, Compute, Concatenate, Convert, Date Order Check, Delete, and a three-line icon. A search bar labeled 'Transform' is positioned above the main table area. The main area contains a table with the following data:

#	Name	Class	Type	Measure	Levels
1	mpg	numeric	double	scale	
2	cyl	numeric	double	scale	
3	disp	numeric	double	scale	
4	hp	numeric	double	scale	
5	drat	numeric	double	scale	
6	wt	numeric	double	scale	
7	qsec	numeric	double	scale	
8	vs	numeric	double	scale	
9	am	numeric	double	scale	
10	gear	numeric	double	scale	
11	cach	numeric	double	scale	
>	bin	factor	integer	factor	1,2,3,4

alt text

This tab creates a factor dissecting the range of a numeric variable into bins of equal width, (roughly) equal frequency, or at "natural" cut points (determined by K-means clustering)

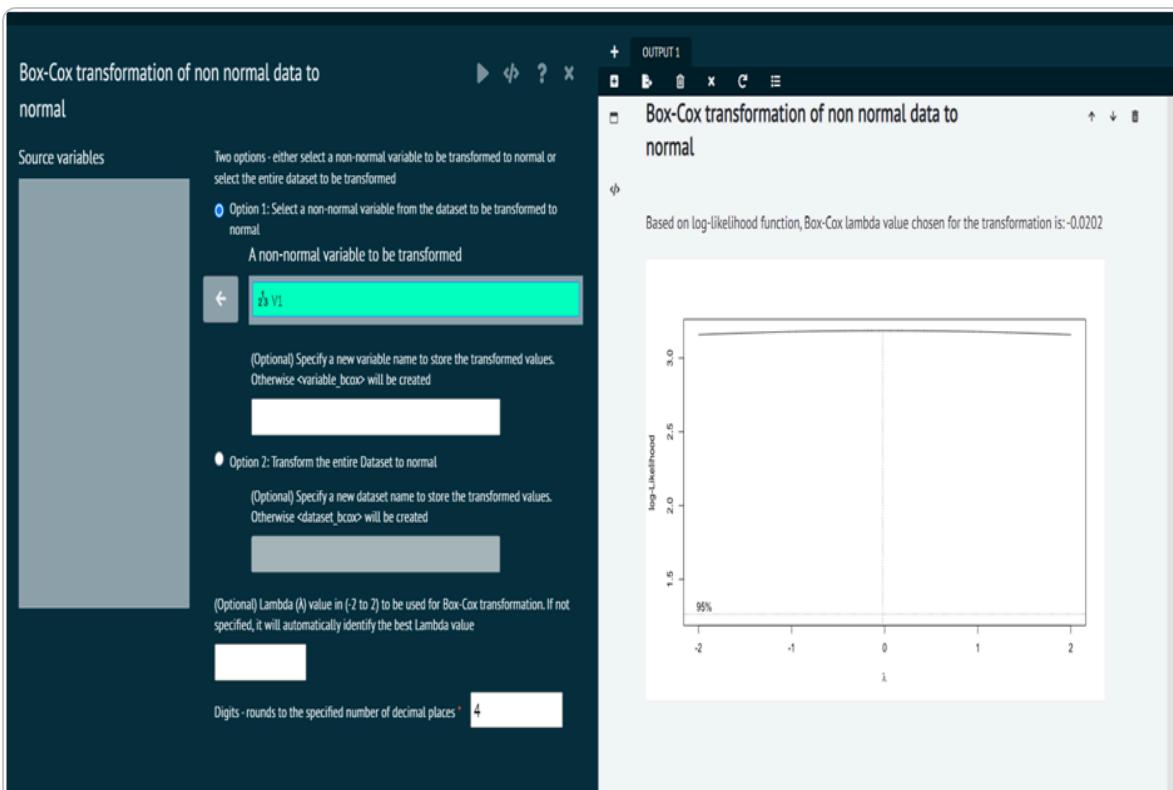
Box Cox

This section of variable menu aids the user to perform 4 different functions on variables of a given dataset, i.e; box cox transformation inspect lambda, add/remove lambda, reverse box cox.

Box-Cox Transformation of data of non-normal data to normal

It is a function used to transform non-normal variable to normal with MASS::boxcox Box-Cox transformation cannot be performed on negative values

- For the detail help - use R help(boxcox, package = MASS)



alt text

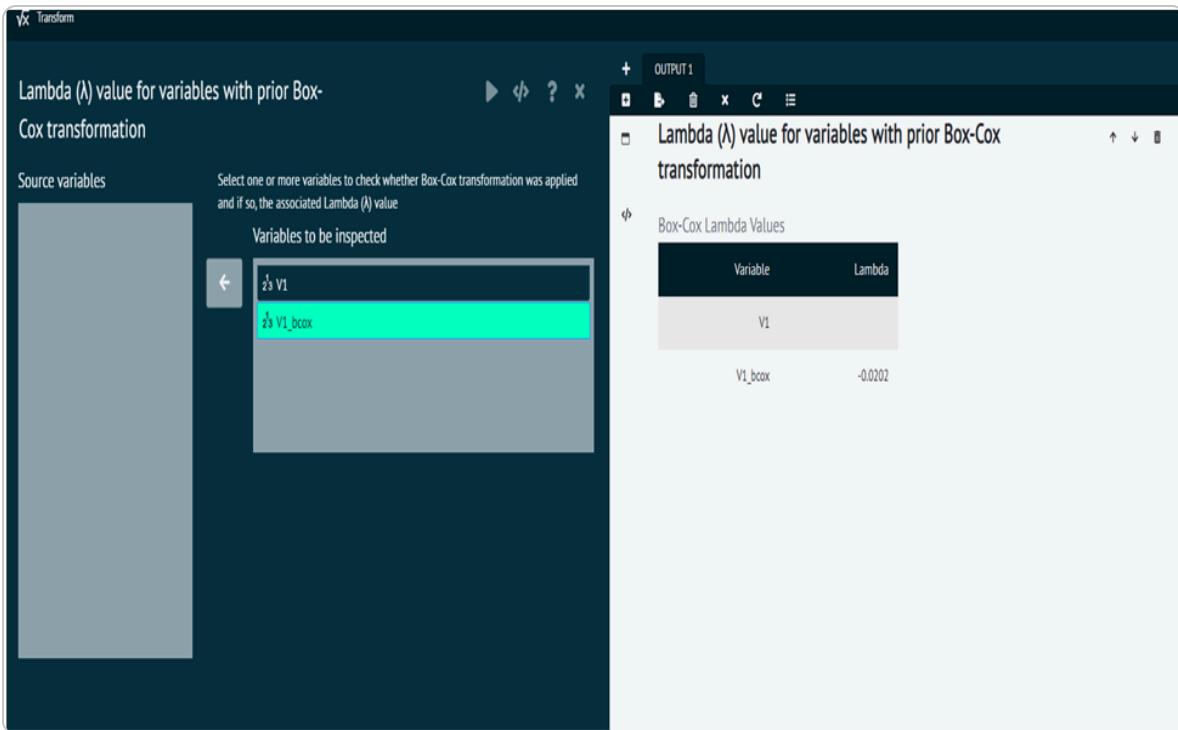
- ⚠** Lambda (λ) values associated with familiar Box-Cox transformations

1. $\lambda = 2$: square transformation (x^2)
2. $\lambda = 1$: no transformation; returns the original data (x)
3. $\lambda = 0.50$: square root transformation (\sqrt{x})
4. $\lambda = 0.33$: cube root transformation
5. $\lambda = 0.25$: fourth root transformation
6. $\lambda = 0$: natural log transformation ($\ln(x)$)
7. $\lambda = -0.50$: reciprocal square root transformation ($1/\sqrt{x}$)
8. $\lambda = -1$: reciprocal (inverse) transformation ($1/x$)
9. $\lambda = -2$: reciprocal square transformation ($1/x^2$)

Inspect Lambda

Checks for the associated Lambda (λ) value, if any, for the selected variables with prior Box-Cox transformation

- i** For the detail help on Box-Cox or Lambda (λ) – use R help(boxcox, package = MASS)



alt text

⚠ Lambda (λ) values associated with familiar Box-Cox transformations

1. $\lambda = 2$: square transformation (x^2)
2. $\lambda = 1$: no transformation; returns the original data (x)
3. $\lambda = 0.50$: square root transformation ($\text{sqrt}(x)$)
4. $\lambda = 0.33$: cube root transformation
5. $\lambda = 0.25$: fourth root transformation
6. $\lambda = 0$: natural log transformation ($\log(x)$)
7. $\lambda = -0.50$: reciprocal square root transformation ($1/\text{sqrt}(x)$)
8. $\lambda = -1$: reciprocal (inverse) transformation ($1/x$)
9. $\lambda = -2$: reciprocal square transformation ($1/x^2$)

Add/Remove Lambda

This dialog is provided for convenience if the Lambda (λ) associated with the variable needs to be recorded correctly or adjusted. The correct Lambda (λ) value is important as it will be used if inverse Box-Cox is needed

The screenshot shows the 'Add/Replace/Remove Lambda (\lambda) for a variable' dialog and its corresponding 'OUTPUT 1' pane.

Dialog Content:

- Source variables:** A list containing 'V1'.
- Select a variable:** A dropdown menu showing 'V1_bcox'.
- Option 1:** 'Add/Replace the Lambda (\lambda) value if the Box-Cox transformation was applied previously'.
 - Specify a Lambda (\lambda) value in (-2 to 2) to be recorded for the selected variable.
- Option 2:** 'Remove the Lambda (\lambda) value if it was mistakenly added to the variable'.

OUTPUT 1 Pane:

- Lambda (\lambda) value for variables with prior Box-Cox transformation:**

Variable	Lambda
V1	-0.0202
V1_bcox	-0.0202
- Add/Replace/Remove Lambda (\lambda) for a variable with prior Box-Cox transformation:**

Original Lambda value:-0.0202020202020201 changed to: 2 for V1_bcox

alt text

Inverse Box-Cox

Transform back (inverse) from a prior Box-Cox transformed value using the specified lambda or the lambda associated with the variable selected

The screenshot shows a software interface with a data table on the left and a transformation dialog on the right.

Data Table:

#	V1	V1_boxx	ne
1	12	2.4236	3.4641
2	16	2.6964	4

Transformation Dialog:

Title: Inverse Box-Cox transformation (convert back to non-transformed value)

Source variables: `$ V1_boxx`

Option 1: Select a variable to be converted back from a prior Box-Cox transformation. A dropdown menu shows `$ V1`.

Option 2: Type in a numeric value to be converted back from Box-Cox transformation. A text input field contains `5`.

Lambda (λ) value: A text input field contains `2`.

Digits - rounds to the specified number of decimal places: A text input field contains `4`.

alt text

Compute

Compute aids the user to compute the variable rows and save the output in a new row.

Applying Function to all Rows

Applies a function across all rows of the selected variables (columns) in a dataset. User can use the select function and the pipe (%>%) operator from the dplyr package to select the variables whose rows we will apply a function to. (These variables are piped into the apply function)

The screenshot shows the Compute interface with the following details:

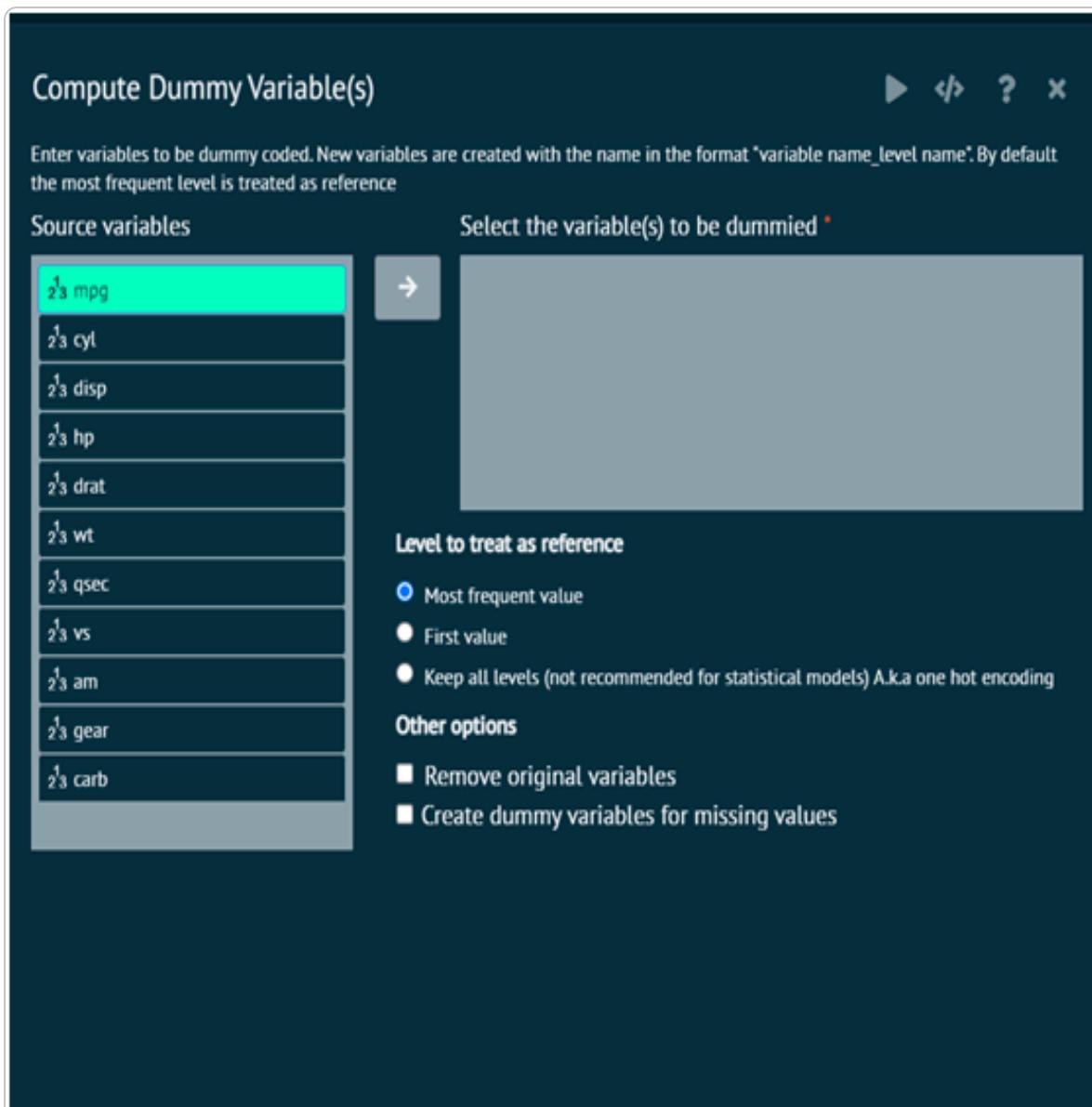
- Title:** Applying a function to all rows of selected variable(s).
- Description:** Create a new variable or overwrite an existing variable by applying a function to all row values of the selected variable(s).
- Source variables:** A list of variables from the mtcars dataset: mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, carb.
- Enter a new variable/Overwrite an existing variable:** An input field for defining the new variable name.
- Select variable(s):** A button to select multiple variables.
- Select an operation to apply:** A dropdown menu showing available operations: mean, median, min, max, sd. The "mean" option is highlighted.

alt text

- i** Computed values are stored directly in Dataset Package : dplyr

Dummy code

In this section variables entered are dummy coded. New variables are created with the name in the format "variable name_level name".

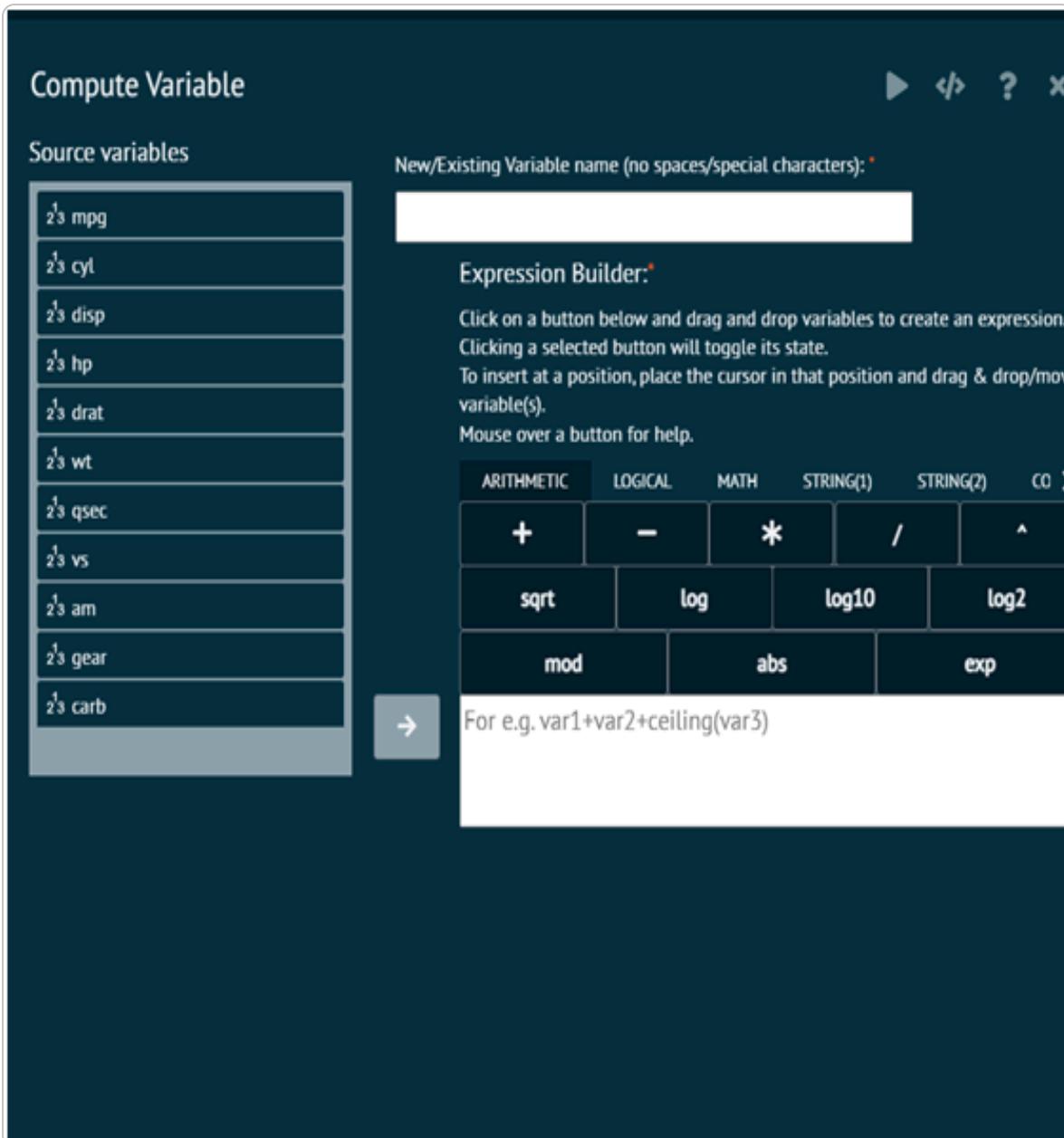


alt text

- i** By default the most frequent level is treated as reference

Compute Variable

Computes an expression and stores the result in a variable/column of a dataframe/dataset.



alt text

The arguments used in executing the dialog are given as follows.

⚠ Arguments

1. DatasetX: dataframe/dataset name.
2. var1: The new/existing column in the dataset/dataframe that needs to be computed
3. Expression: An expression in the form variable1 =variable2+variable3

Conditional Compute

Conditional Compute

Source variables

23 mpg
23 cyl
23 disp
23 hp
23 drat
23 wt
23 qsec
23 vs
23 am
23 gear
23 carb

New/Existing Variable name (no spaces/special characters):

Specify a condition e.g. var1 > 10*

Click on a button below and drag and drop variables to create an expression. Clicking a selected button will toggle its state. To insert at a position, place the cursor in that position and drag & drop/move variable(s). Mouse over a button for help.

ARITHMETIC	LOGICAL	MATH	STRING(1)	STRING(2)	C >
+	-	*	/	^	
sqrt	log	log10	log2		
mod	abs		exp		

For eg. gpa == 4

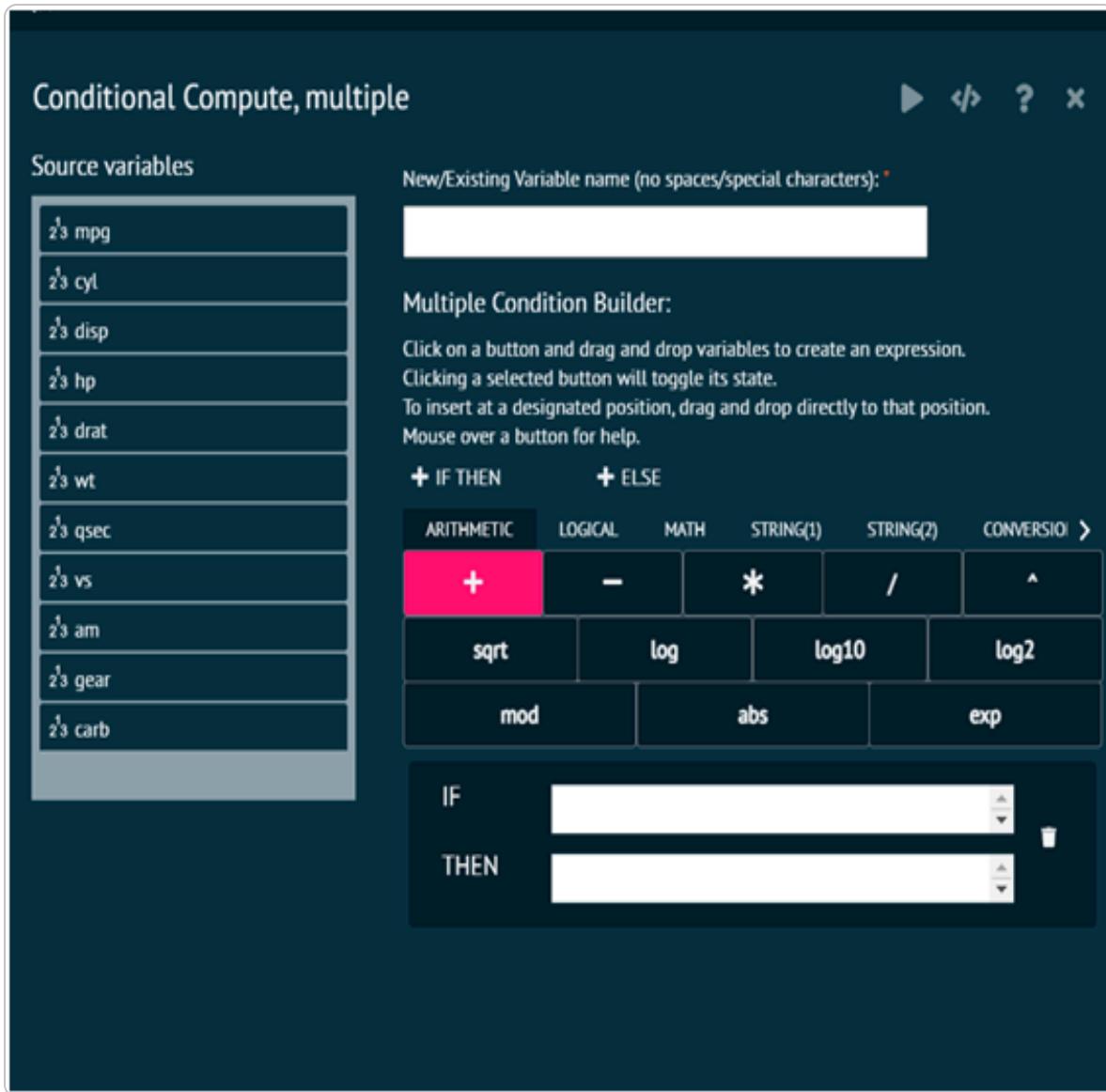
Value when condition is TRUE*

Click on a button below and drag and drop variables to create an expression. Clicking a selected button will toggle its state. To insert at a position, place the cursor in that position and drag & drop/move variable(s). Mouse over a button for help.

ARITHMETIC	LOGICAL	MATH	STRING(1)	STRING(2)	C >
+	-	*	/	^	

alt text

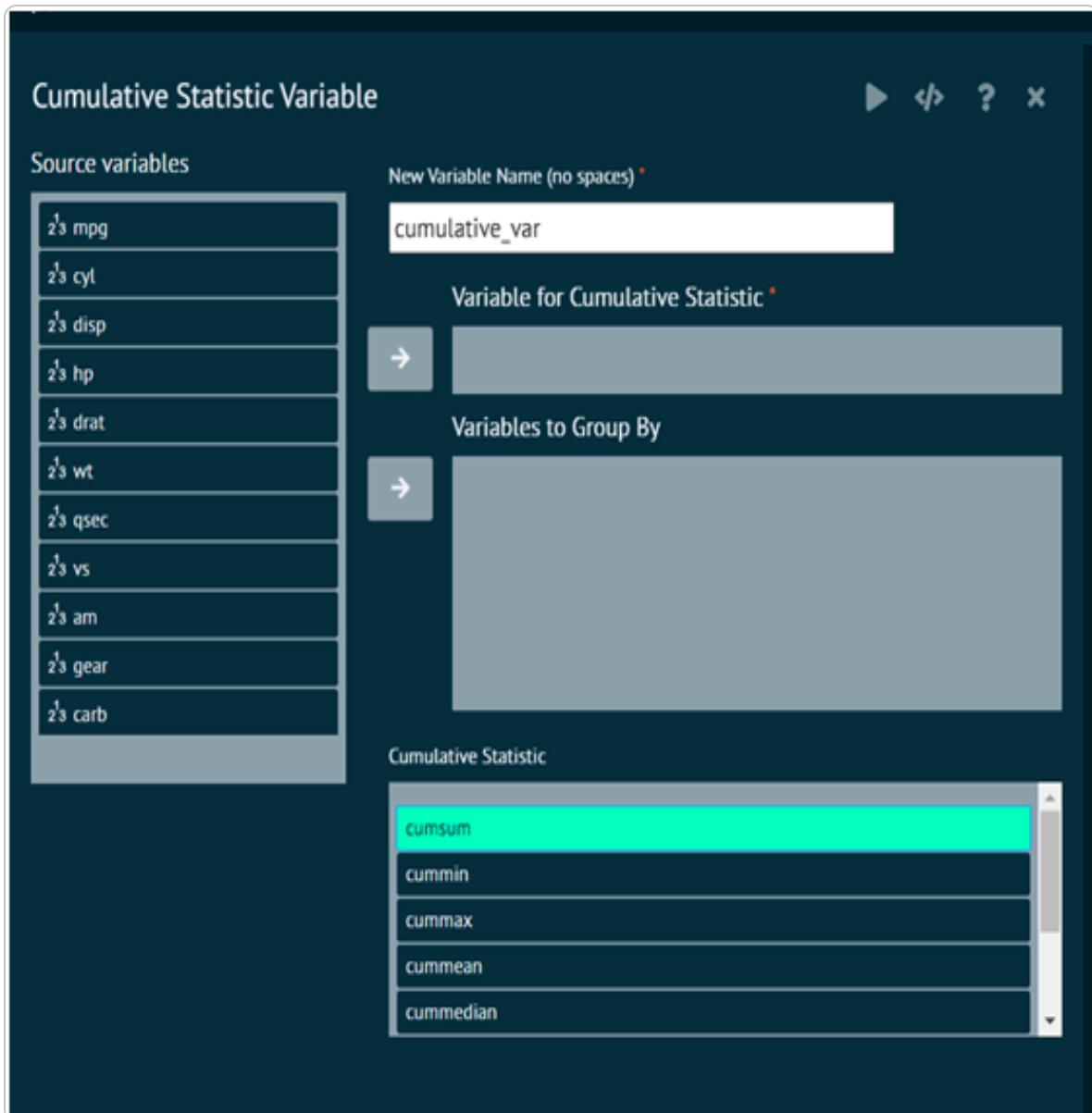
Conditional Compute Multiple



alt text

Cumulative Statistics Variable

This dialog creates a new variable that stores the cumulative value of a chosen statistic as you go down the rows in the current order of the dataset. User can optionally compute this cumulative value within one or more groups.



alt text

New Variable Name: Name of variable that will store the cumulative values

Variable for Cumulative Statistic: Variable for which the cumulative values will be computed. Must be numeric.

Variables to Group By: Optional variables to compute the cumulative statistic within.

Cumulative Statistic: Which statistic will be used for the cumulative statistic.

cumsum

cumulative sum

cummin

cumulative minimum

cummax

cumulative maximum

cummean

cumulative mean

cummedian

cumulative median

cumgmean

cumulative geometric mean

cumhmean

cumulative harmonic mean

cumvar

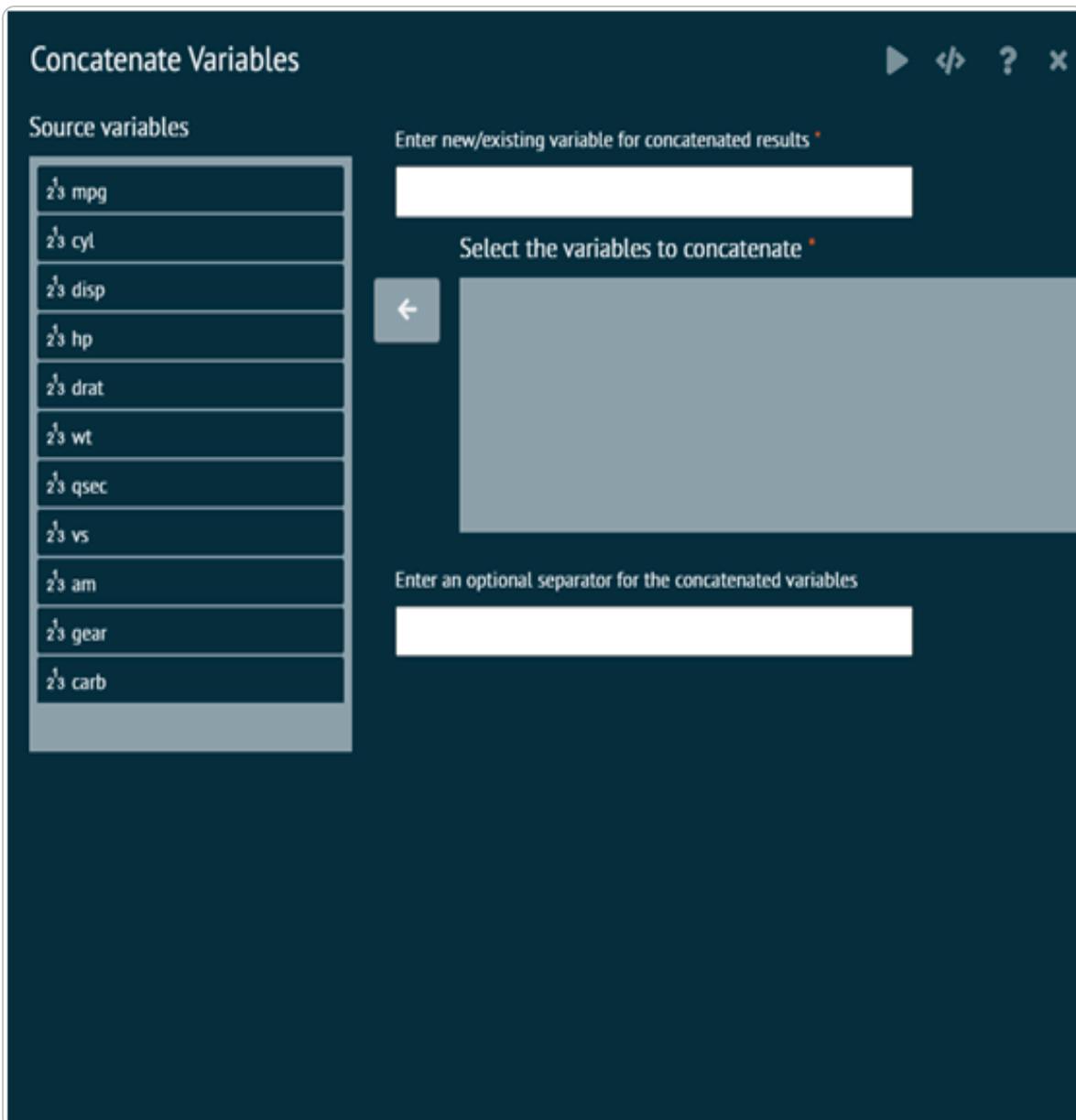
cumulative variance



Required R Packages: `cumstats`, `dplyr`

Concatenate

Create a factor dissecting the range of a numeric variable into bins of equal width, (roughly) equal frequency, or at "natural" cut points (determined by K-means clustering)



alt text

Convert

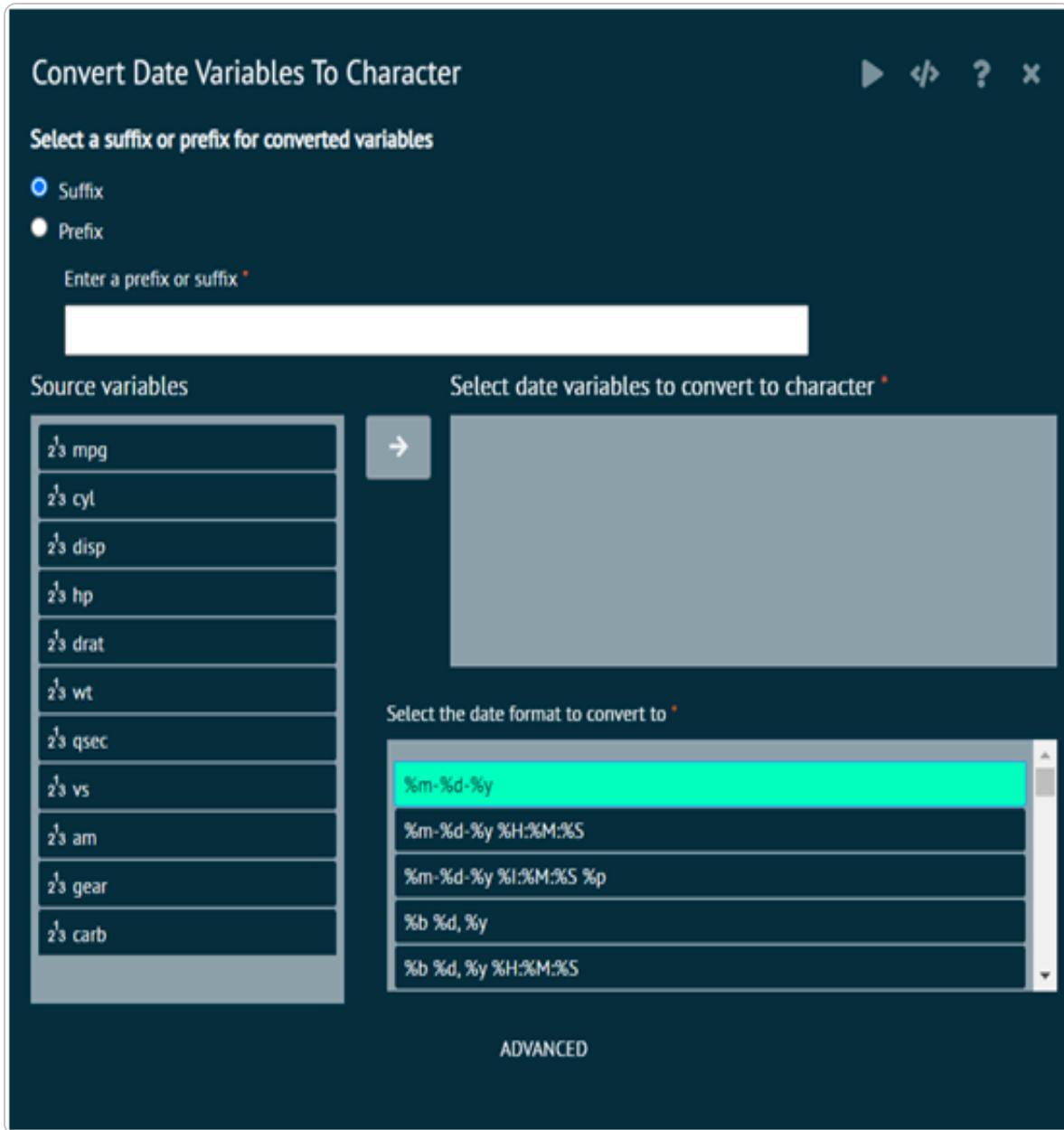
Convert section of variable menu aids the user to convert a character variable to date, to factor, to ordered factor and vice versa.

Date to Character

Converts date (posixct and date class) to character -to control the format in which the date is displayed. User can specify as input the format in which the string should be generated i.e. year/month/Day or month-dat=year etc.

The function above internally calls strftime in the base package.

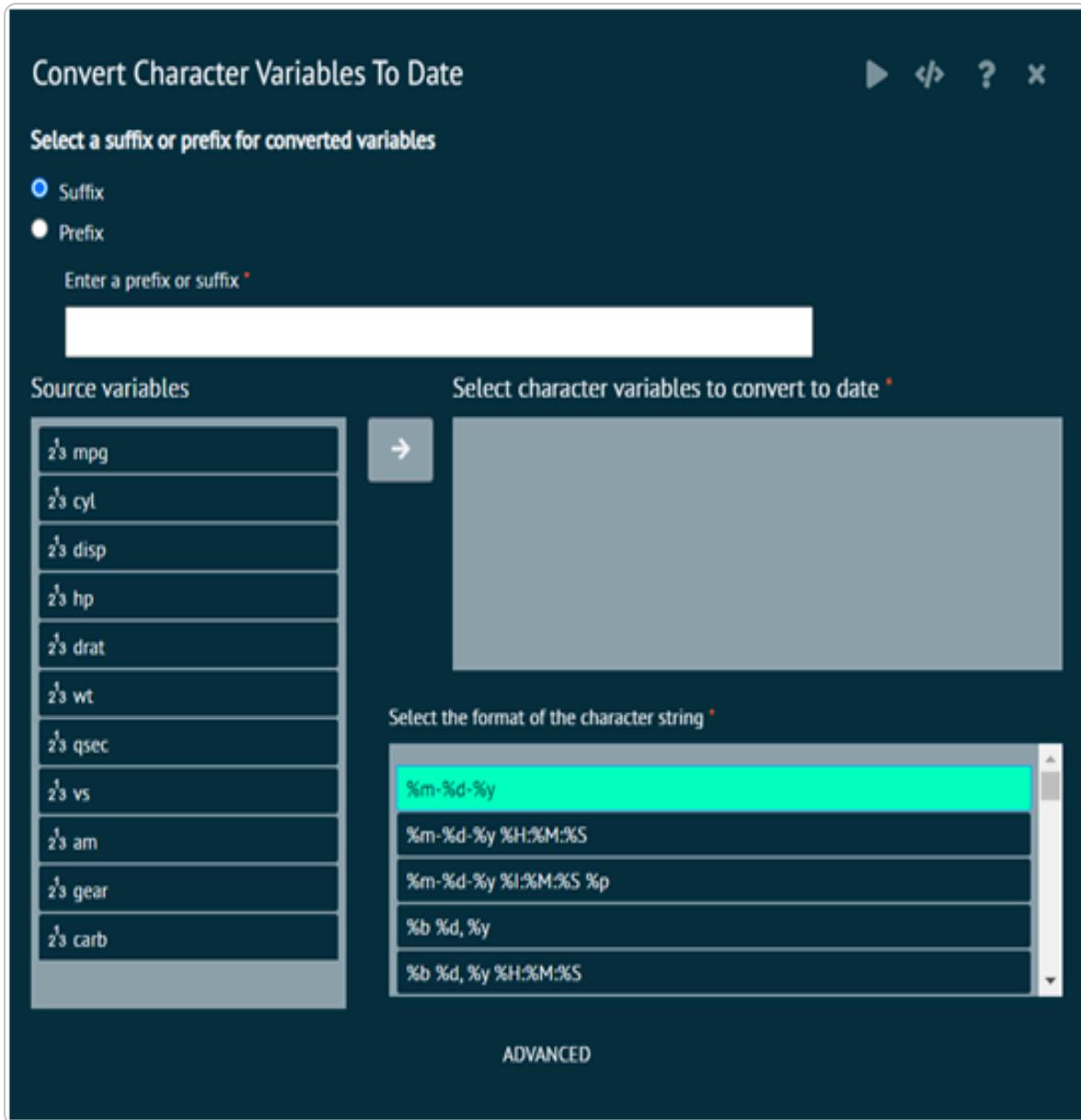
- i** BioStat Prime has extended strftime to support multiple variables.



alt text

Character to Date

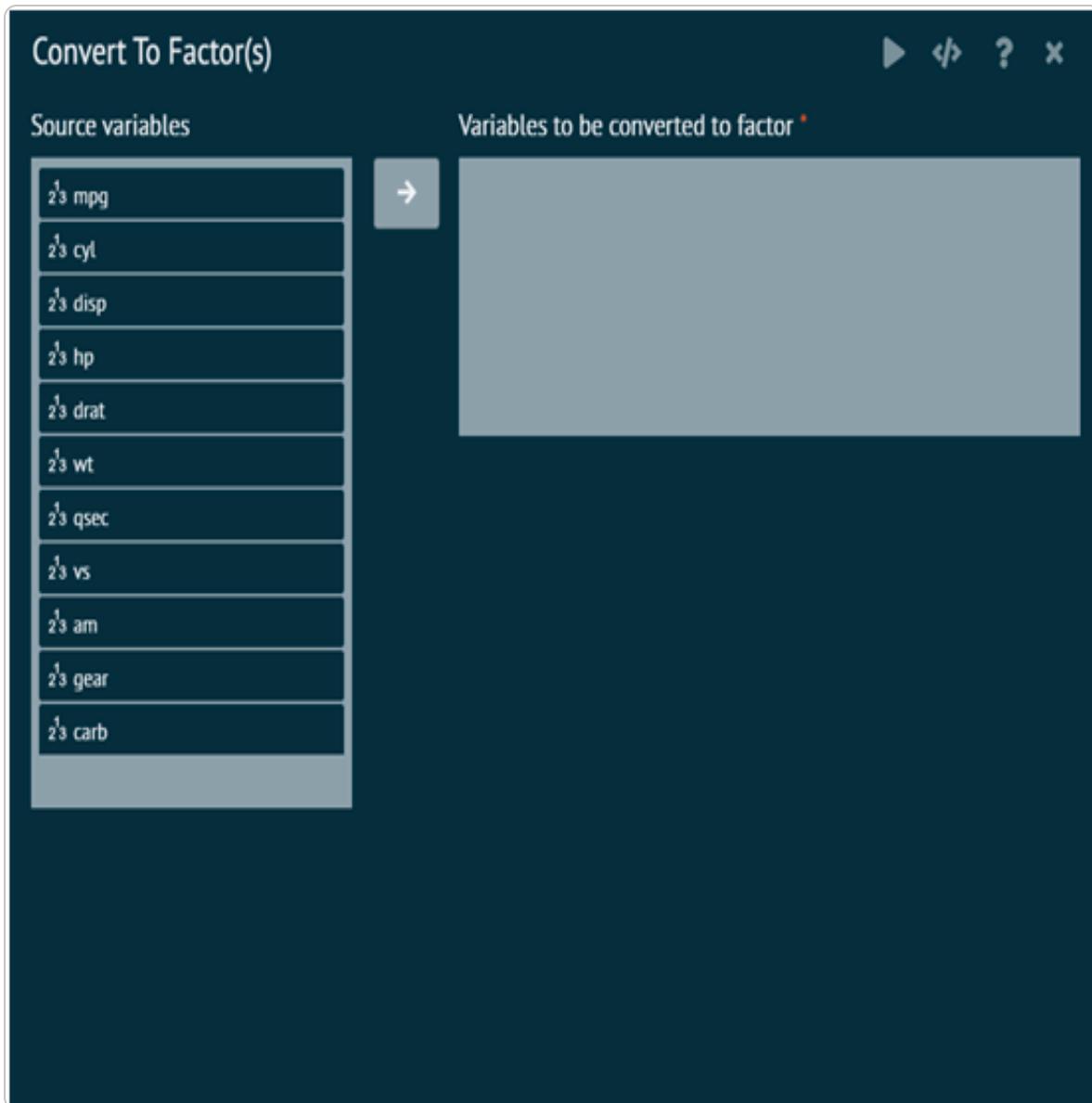
Converts a character to a date (POSIXct class). User needs to specify the format of the date stored in a character string.



alt text

Convert to Factor

The function `factor` is used to encode a vector as a factor (the terms ‘category’ and ‘enumerated type’ are also used for factors). If argument `ordered` is `TRUE`, the factor levels are assumed to be ordered. For compatibility with S there is also a function `ordered`. `is.factor`, `is.ordered`, `as.factor` and `as.ordered` are the membership and coercion functions for these classes.



alt text

Convert to ordered factor/ordinal

The function `factor` is used to encode a vector as a factor (the terms ‘category’ and ‘enumerated type’ are also used for factors). If argument `ordered` is `TRUE`, the factor levels are assumed to be ordered. For compatibility with S there is also a function `ordered`. `is.factor`, `is.ordered`, `as.factor` and `as.ordered` are the membership and coercion functions for these classes.

Convert To Ordered Factor(s)/Ordinal

▶ ⌂ ? ×

Source variables

23 mpg
23 cyl
23 disp
23 hp
23 drat
23 wt
23 qsec
23 vs
23 am
23 gear
23 carb

Variables to be converted to ordered factor/ordinal *

→

alt text

Date order check

This creates a list of rows in the active dataset where date variable values are not in a specified order. This helps identify potential date variable errors when dates or times are needed for an analysis.

For example, if three date columns are supposed to be in the order of date1 < date2 < date3, this dialog will print all observations where the values of those variables do not follow that order.

- i** Missing date values are allowed in the specified variables and will not be used for any comparisons.



alt text

The arguments used in executing the dialog are given as follows.

Date Variables (specify earliest to latest; same class; at least 2)

Specify at least 2 date variables in the order of earliest to latest. These can be any date class (POSIXct, Date), but all variables specified must be the same date class. If not, an error will result.

Comparison

Specify the comparison operator used to compare the date values. "<" means less than and "<=" means less than or equal to. If "<" is chosen, then dates that are equal will be flagged as errors. If "<=" is chosen, then dates that are equal will not be flagged as errors.

Row Identification Variables (optional)

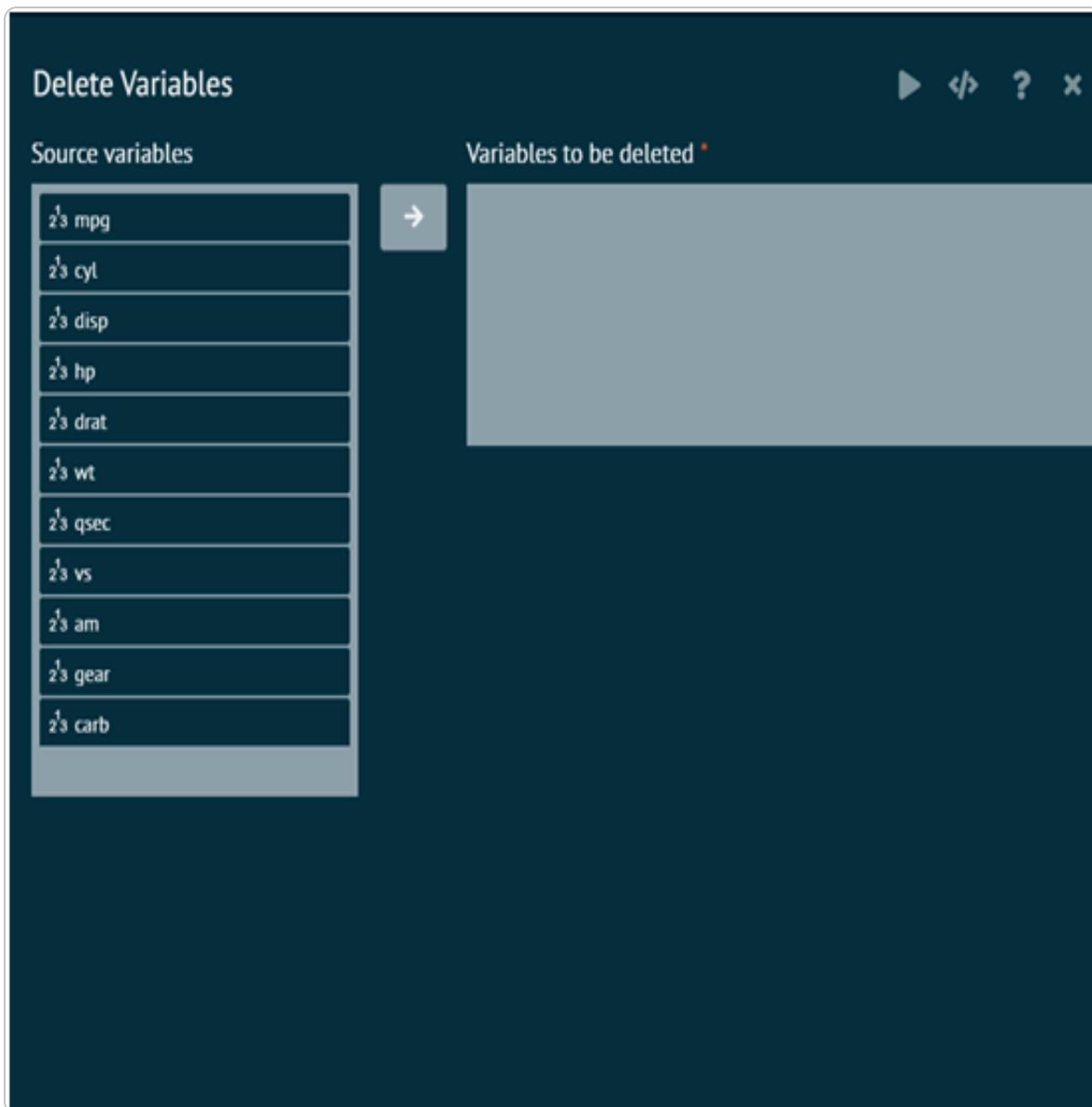
Specify one or more variables that may be useful to identify the rows. For example, subject identification number. These will be included in the list of errors. If no variables are specified, the row number of the dataset will be the only identifier.

Create dataset with date error variable

This will create a separate data set with the original data and a variable indicating whether each observation has a date order error (coded as 1=date order error and 0=no date order error). The Dataset name is the desired name of this data set and Date error variable name is the desired name of the date order error variable in this data set.

Delete Variable

Removes missing values/NA from dataset/dataframe. Creates new/Overwrites existing dataset by removing rows with one or more missing values for the columns/variable names selected



alt text

Factor levels

Add new levels

Adds additional levels to a factor. Add new levels to one or more factor variable(s). The results can be into existing variables (overwriting) or creating new variables by specifying a prefix/suffix.

New variables will be created with the prefix/suffix appended to existing names.

Add New Levels

Add new levels to one or more factor variable(s). Save results to existing variables or create new variables by specifying a prefix/suffix. New variables will be created with the prefix/suffix appended to existing names.

Source variables

23 mpg
23 cyl
23 disp
23 hp
23 drat
23 wt
23 qsec
23 vs
23 am
23 gear
23 carb

Factor variables to add new levels to *

→

Enter new levels enclosed in quotes and separated by a comma for e.g.
"level1","level2","level3"

Save new levels to new variable(s) or overwrite existing variable(s)

Specify a suffix (A new variable will be created with the suffix)

Enter a suffix

Specify a prefix (A new variable will be created with the prefix)

Enter a prefix

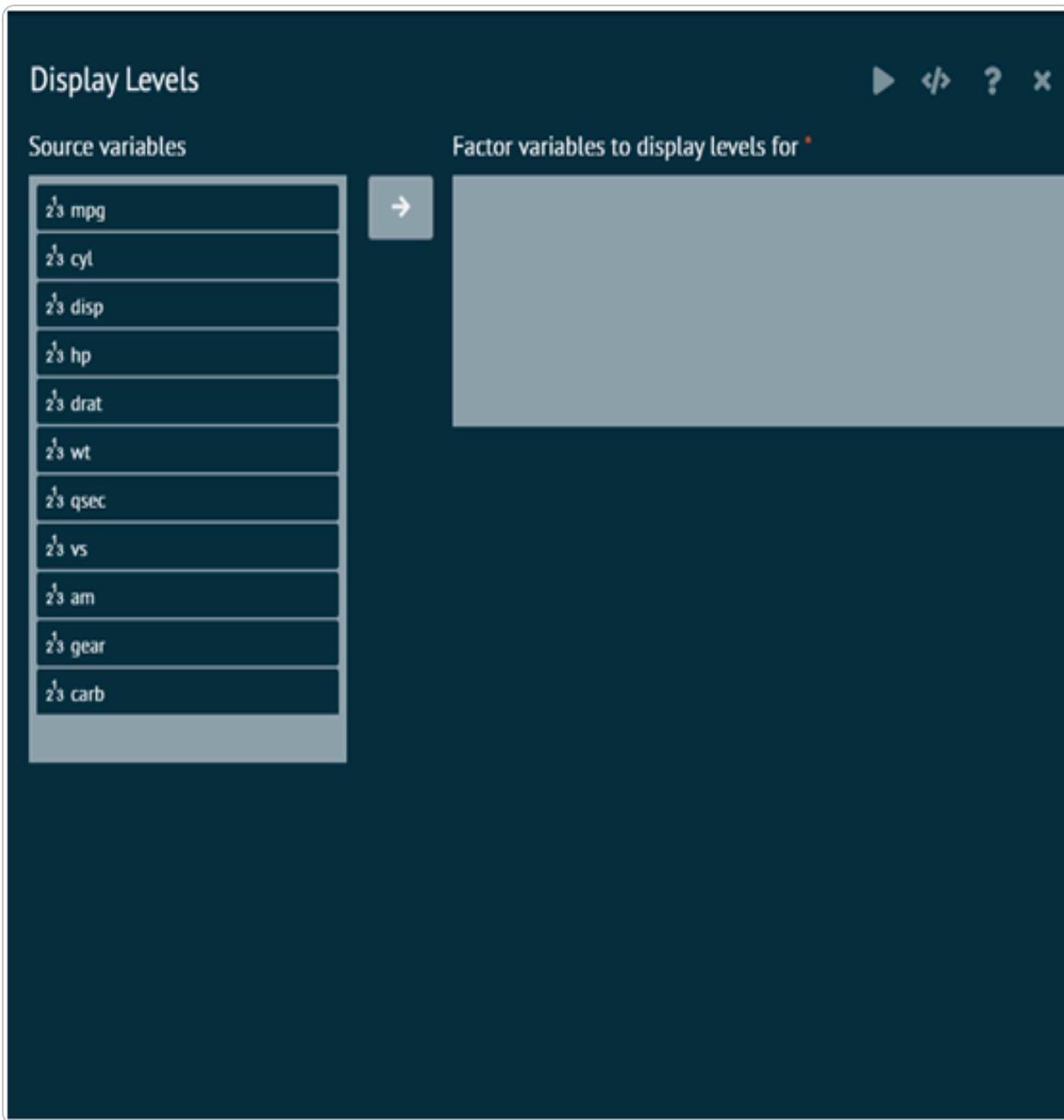
Overwrite existing variables

alt text

- DON'T ENCLOSE LEVELS IN DOUBLE QUOTES OR SINGLE QUOTES.
- THERE CANNOT BE SPACES IN THE LEVEL NAMES.
- ENTER LEVELS SEPARATED BY COMMAS IN THE FORMAT LEVEL1,LEVEL2, LEVEL3

Display levels

Applies the levels function in base to the selected variables in the dataset. Users select the function in dplyr to pipe the variables to map the function that applies the levels function to each variable.



alt text

Drop used levels

Enter the factor variable(s) to drop unused levels for. User can specify unused levels to drop by entering them or select to drop all unused levels for the variable(s) selected. If the dataset variable has a NA value(s) in the data, then that level is NOT dropped.

Drop Unused Levels

Enter the factor variables to drop levels for. You can specify unused levels to drop by entering them or select to drop all unused levels for the variable(s) selected. If the dataset variable has a NA value(s) in the data, then that level is NOT dropped.

Source variables

13 mpg
13 cyl
13 disp
13 hp
13 drat
13 wt
13 qsec
13 vs
13 am
13 gear
13 carb

Factor variables to drop levels for *

Method to use

- Drop all unused levels
- Specify levels to drop

Enter levels to drop separated by comma, for e.g. level1,level2,level3 NOTE:
Don't use spaces as separators between the levels

Save new levels to new variable(s) or overwrite existing variable(s)

- Specify a suffix (A new variable will be created with the suffix)

Enter a suffix

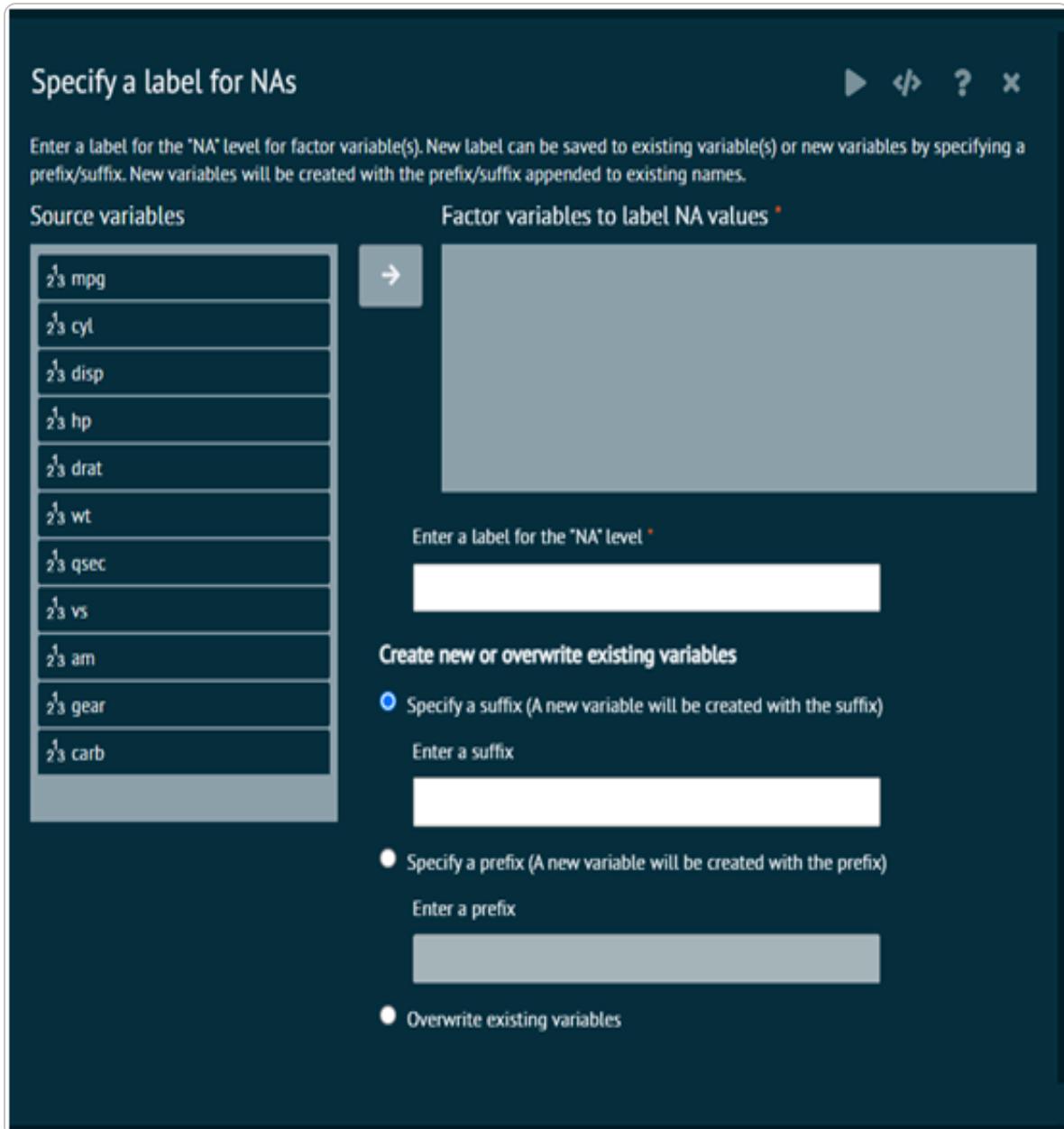
Specify a prefix (A new variable will be created with the prefix)

Enter a prefix

alt text

Specify a label for NAs

Enter a label for the "NA" level for factor variable(s). New label can be saved to existing variable(s) (we overwrite existing variables) or new label for NA can be saved to new variables by specifying a prefix/suffix. New variables will be created with the prefix/suffix appended to existing names. This gives missing value an explicit factor level, ensuring that they appear in summaries and on plots.



alt text

Lump the least or most common factor levels

Lump together the least or the most common factor levels into the "other" level. The default name of the new category containing the lumped levels is "other". Specifying weights is optional. User can overwrite existing variable(s) with the lumped levels or save the results to new variable(s)

Lump the least or most common factor levels



The default name of the new category containing the lumped levels is "other". Specifying weights is optional. You can overwrite existing variable(s) with the lumped levels or save the results to new variable(s)

Source variables

2/3 mpg
2/3 cyl
2/3 disp
2/3 hp
2/3 drat
2/3 wt
2/3 qsec
2/3 vs
2/3 am
2/3 gear
2/3 carb

Select variables to lump sparse levels for *



Name for the lumped level *

other

Method to use

- Lump together least frequent levels into "other" while ensuring that "other" is the smallest level
- Keep most common (+n)/least common (-n) categories

Enter the number of categories 0

- Keep categories that appear at least (+ prop)/at most (- prop) proportion of the time

Enter the proportion 0.1

Variable weights



alt text

Specify levels to keep or replace by other

Enter the factor levels to keep or drop. When keep is selected, remaining levels will be replaced by "Other". When drop is selected, dropped levels will be replaced by "Other"

Specify levels to keep or replace by other

Enter the factor levels to keep or replace by other. When levels to keep is selected, remaining levels will be replaced by "Other". When replace is selected, specified levels will be replaced by "Other".

Source variables	Factor variables to reorder *
z3 mpg	
z3 cyl	
z3 disp	
z3 hp	
z3 drat	
z3 wt	
z3 qsec	
z3 vs	
z3 am	
z3 gear	
z3 carb	

Level name used for "Other" values *

Method to use

Enter levels to keep separated by , remaining levels will be replaced by "Other" e.g. level1,level2,level3

Keep levels

Enter levels to replace by "Other" for e.g. level1,level2,level3

Drop levels

Save results to new variable(s) or overwrite existing variable(s)

● Configuration: z3 mpg,z3 cyl,z3 disp,z3 hp,z3 drat,z3 wt,z3 qsec,z3 vs,z3 am,z3 gear,z3 carb

alt text

Reorder Factor levels by Count

Re-order variables in the dataset in alphabetical order. BisStat Prime use the sort function to sort the names of the columns/variables in the dataset and the select function in the package dplyr to select the column names in the correct alphabetical order

Reorder Factor Levels by Count

Select the factor variables to reorder by count. You can overwrite existing variables or create new variables by specifying a prefix/suffix. New variables will be created with the prefix/suffix appended to existing names.

Source variables

23 mpg
23 cyl
23 disp
23 hp
23 drat
23 wt
23 qsec
23 vs
23 am
23 gear
23 carb

Select factor variables to re-order *

Specify an order

In the descending order by count
 In the ascending order by count

Make an ordered factor (ordinal)

Save new levels to new variable(s) or overwrite existing variable(s)

Specify a suffix (A new variable will be created with the suffix)
Enter a suffix

Specify a prefix (A new variable will be created with the prefix)
Enter a prefix

alt text

Reorder Factor levels by another variable

Reorder factor levels by sorting along another variable. Factor levels are reordered based on an arithmetic function i.e. mean, median, sum of the values in another variable. Select the factor variable to reorder, select a numeric variable to compute the mean, median or sum. This is computed for each level of the factor variable. The levels are then ordered based on this calculation.

- i The results can be saved into the existing variable(s) or user can create new variables by specifying a prefix/suffix.
- i New variables will be created with the prefix/suffix appended to existing names.

Reorder Factor Levels by Another Variable

Reorder factor levels based on an arithmetic function i.e. mean, median, sum of the values in another variable. Select the factor variable to reorder, select a numeric variable to compute the mean, median or sum. This is computed for each level of the factor variable. The levels are then ordered based on this calculation. You can overwrite existing variables or create new variables by specifying a prefix/suffix. New variables will be created with the prefix/suffix appended to existing names.

Source variables

23 mpg
23 cyl
23 disp
23 hp
23 drat
23 wt
23 qsec
23 vs
23 am
23 gear
23 carb

Select factor variable to re-order *

Variable to order by *

Select a function to order by *

- mean
- median
- sum
- min
- max

Specify an order

Descending

Ascending

Save results to a new variable or overwrite existing variable

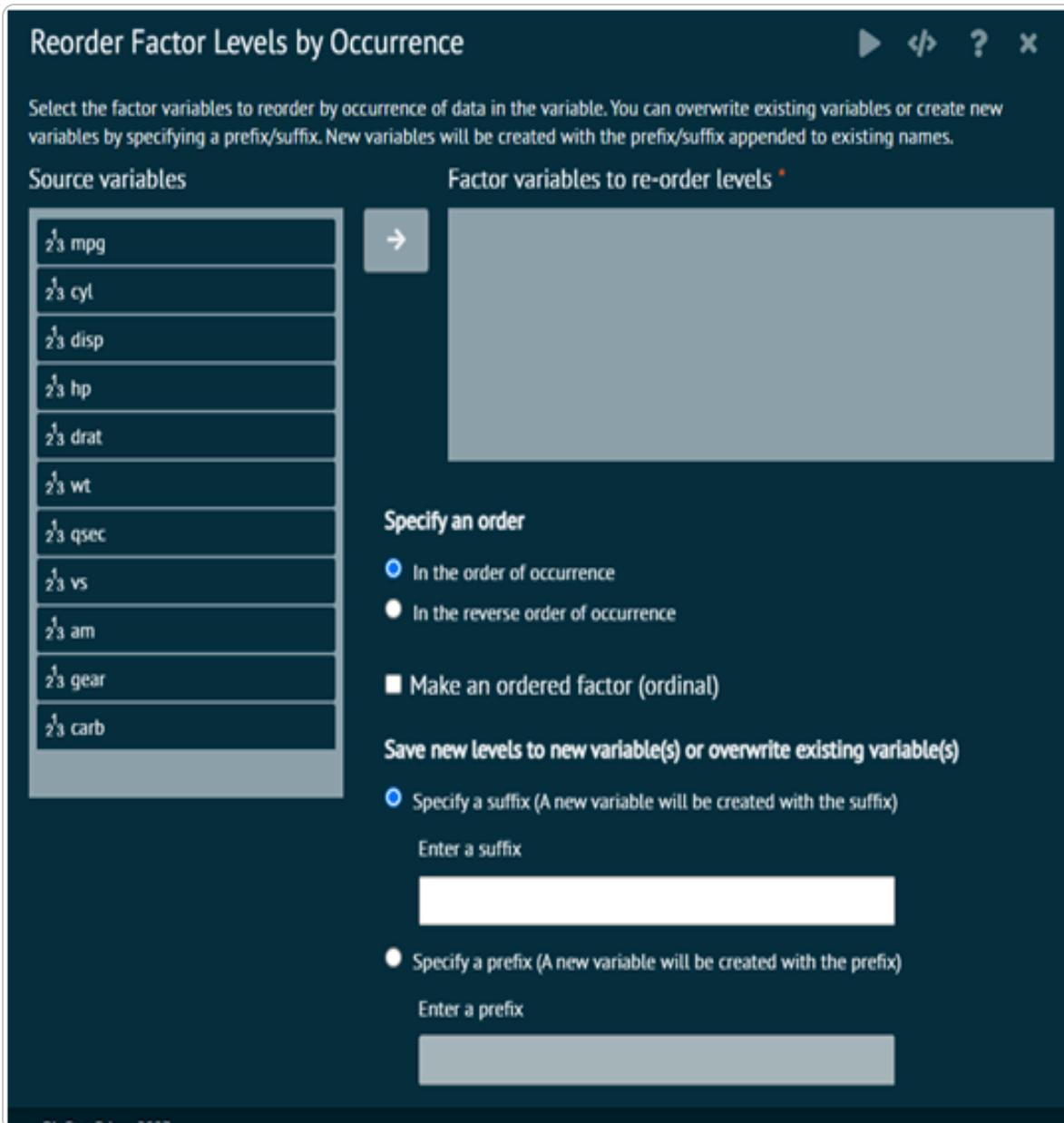
Specify a suffix (A new variable will be created with the suffix)

Enter a suffix

alt text

Reorder by occurrence

Reorder factors levels by first appearance (occurrence). See reorder by count for ordering by count/frequency.

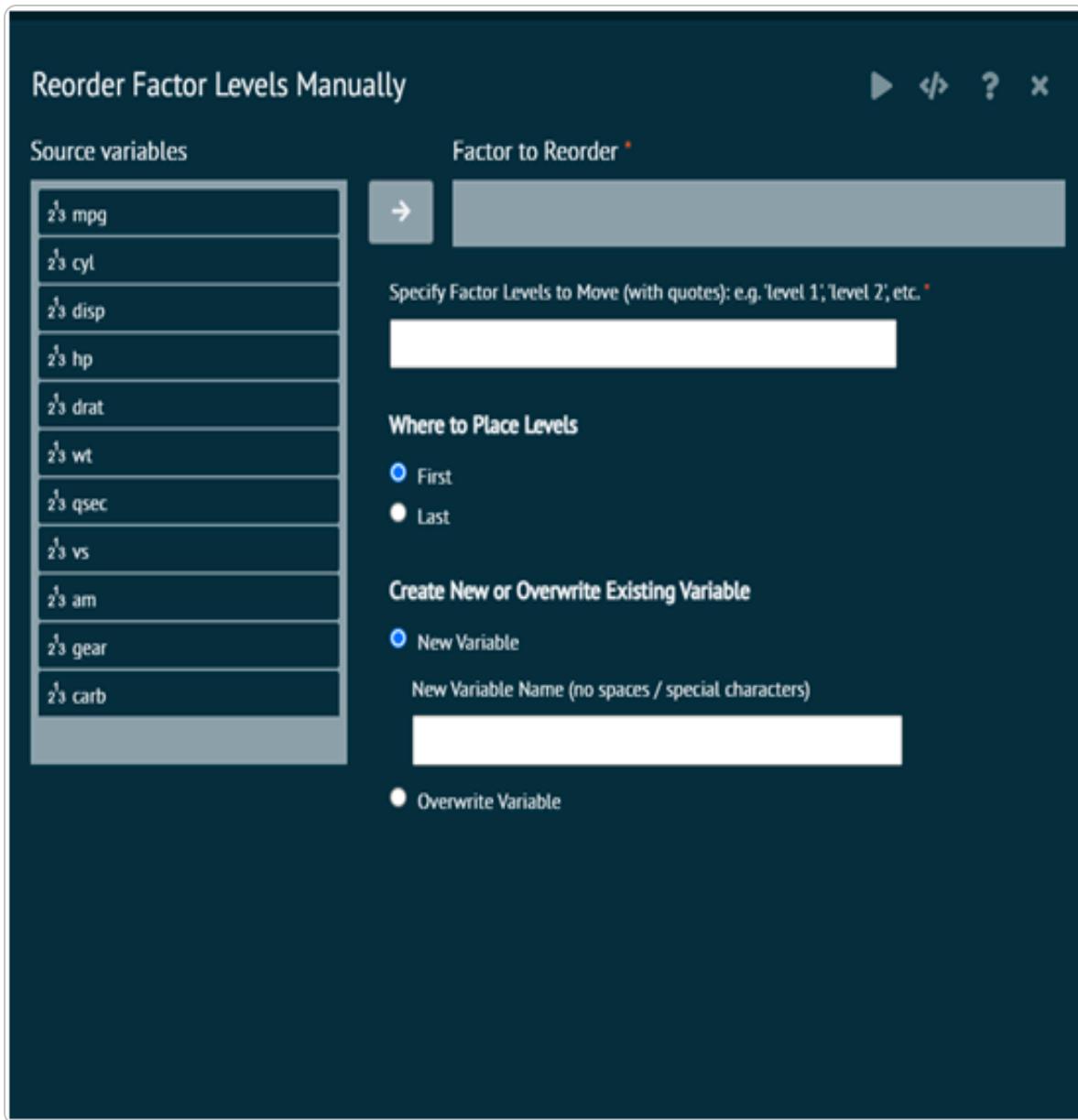


alt text

Reorder Manually

This is used to specify one or more factor levels that user wants to place first or last in the sort order. This can be useful for models, as the first factor level becomes the reference

group for parameter estimates when using reference cell coding. They can also be useful in plotting as the sort order is used to display the categories.



alt text

The arguments used in executing the dialog are given as follows.

Factor to Reorder

factor user wants to re-ordered

Specify Factor Levels to Move (with quotes)

These are the factor levels user wants to reorder. Only existing levels will be reordered. If you specify a non-existent level, a warning will be output, but any existing levels will be ordered in the way user specified. View the levels in the Variables tab of the data grid to see the current levels and sort order or go to Variables > Factor Levels > Display. Note that specifying all existing factor levels will reorder all levels, regardless of whether user selects "First" or "Last" for level placement.

Where to place levels

Selecting "First" will place the specified levels first in the sort order. Selecting "Last" will place the specified levels last in the sort order.

Create new or overwrite existing variable

Controls whether user wants to create a new variable with a new name or overwrite the existing variable. The new variable name cannot contain special characters like #, \$, %, &, (,), =, etc. Underscores, "_", are allowed.

Examples

Assume user has a four level factor with labels "a", "b", "c", "d" with a sort order of "a", "b", "c", "d" (first to last). Specifying "d" as first in the sort order would create a factor with a sort order of "d", "a", "b", "c". Specifying "b", "a" as last in the sort order would create a factor with a sort order of "c", "d", "b", "a". Specifying "b", "c", "d", "a" (i.e. all levels) would create a factor with a sort order of "b", "c", "d", "a".



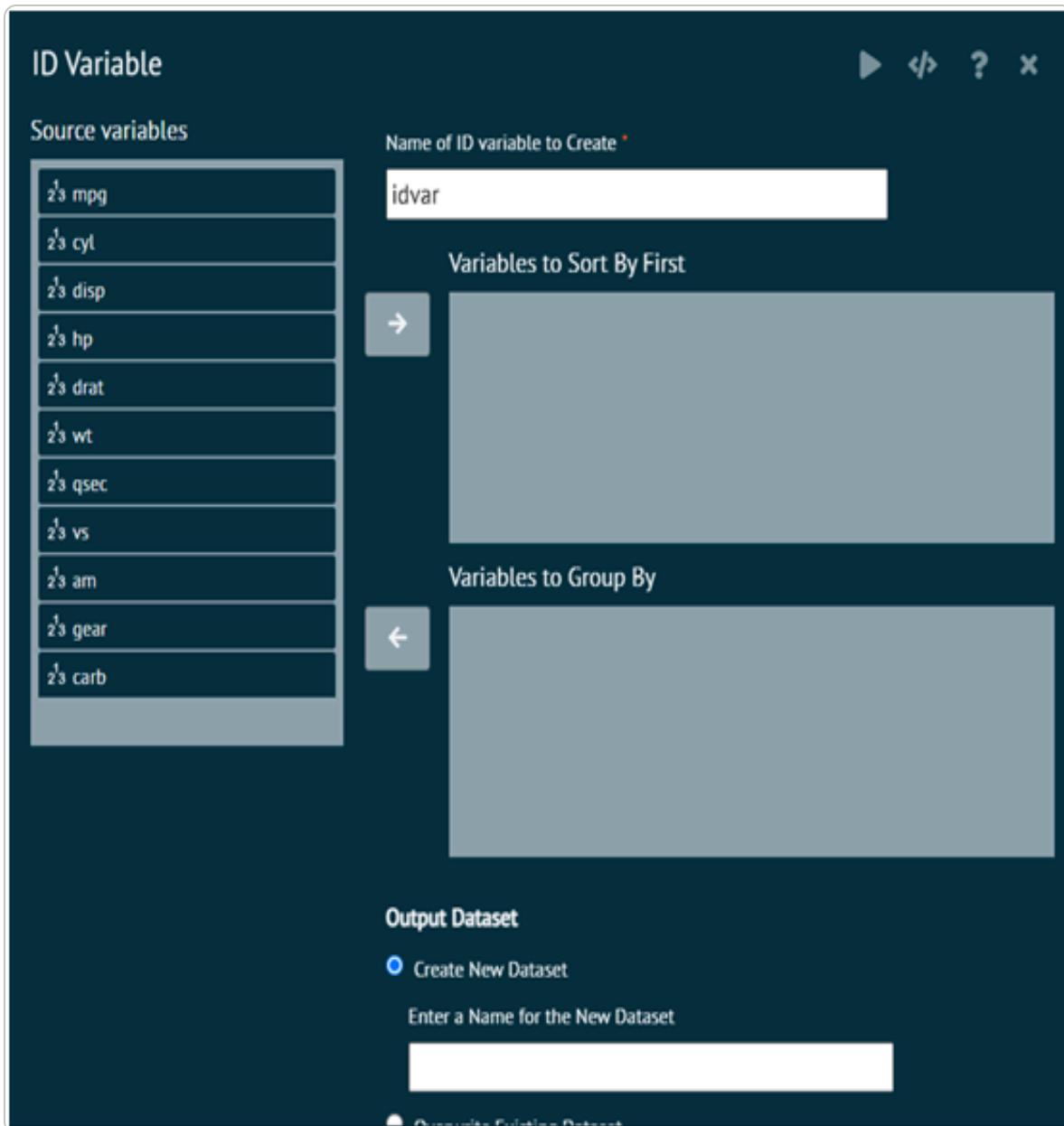
Required R Packages: dplyr,forcats

ID Variable

Creates a numeric row identification (ID) variable in either the current dataset or in a separate copy of the current dataset. The ID variable created will consist of the numeric values 1, 2, 3, etc., in that order, from top to bottom in the dataset.

Can optionally specify variables to sort by or group by before the identification variable is created. The variables to sort by and group by can be the same or different variables.

- ⚠** If no grouping variables are specified, the overall row number of the dataset will be assigned to the ID variable.



alt text

The arguments used in executing the dialog are given as follows.

Name of ID variable to create

Specify the desired name of the ID variable in the output dataset.

Variables to sort by first (optional)

Specify variables to sort by before groups are defined or the ID variable is created

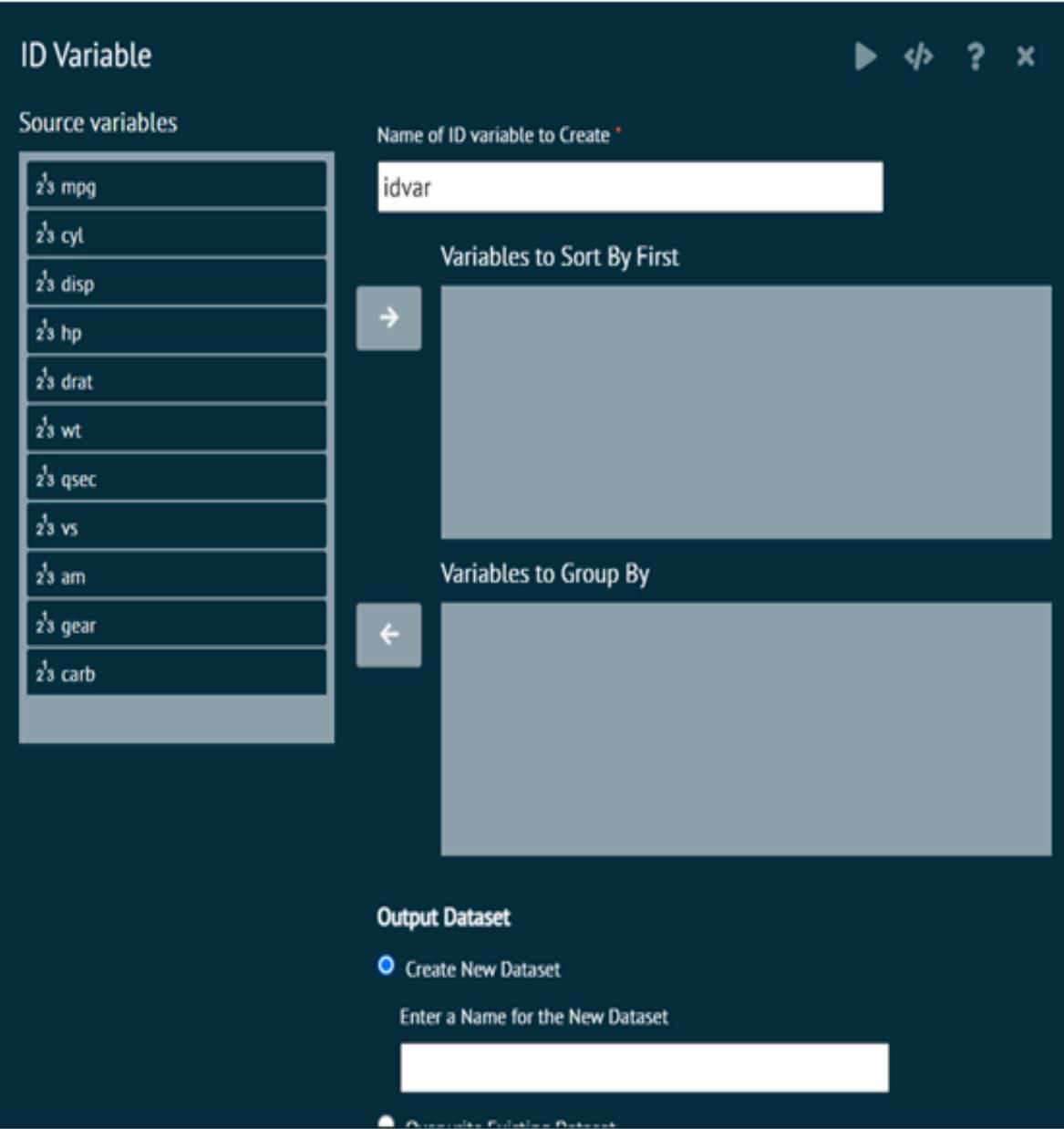
Variables to Group By (optional)

Specify variables whose levels will define the separate assignment of ID values. For example, grouping by gender will create values of 1, 2, 3, etc. separately for males and females, in order of appearance in the data set.

Output Dataset

Specify whether to create a new dataset or overwrite the current dataset

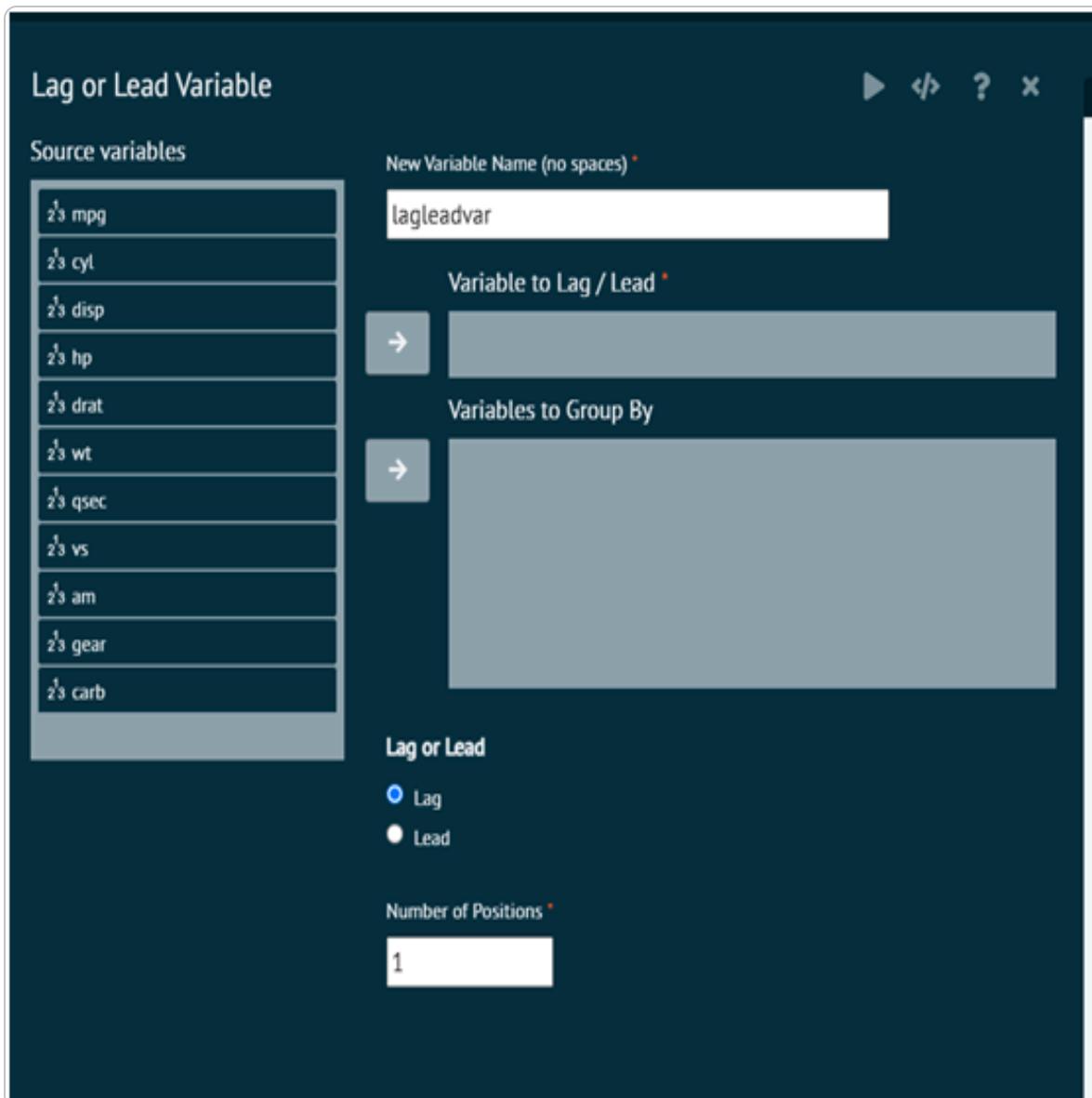
- i** Required R Packages: tidyverse



alt text

Lag or Lead Variable

This creates a new variable that finds the previous (lag) or next (lead) value in an existing variable based on the row position.



alt text

The arguments used in executing the dialog are given as follows.

```
New Variable Name
```

Variable name to store the lagged or leading values

Variable to Lag / Lead

Specify the existing variable to extract the lagged or leading values from

Variables to Group By (optional)

Specify the variables to group by. If variables are specified here, the lagged and lead values will be obtained only within groups defined by these variables. If no variables are specified here, the lagged and leading values will be obtained based on the entire column specified in Variable to Lag / Lead. Typically, values should be sorted by the grouping variables prior to doing a lag or lead.

Lag or Lead

Choose whether user wants to find the previous (lag) or next value (lead)

Number of Positions

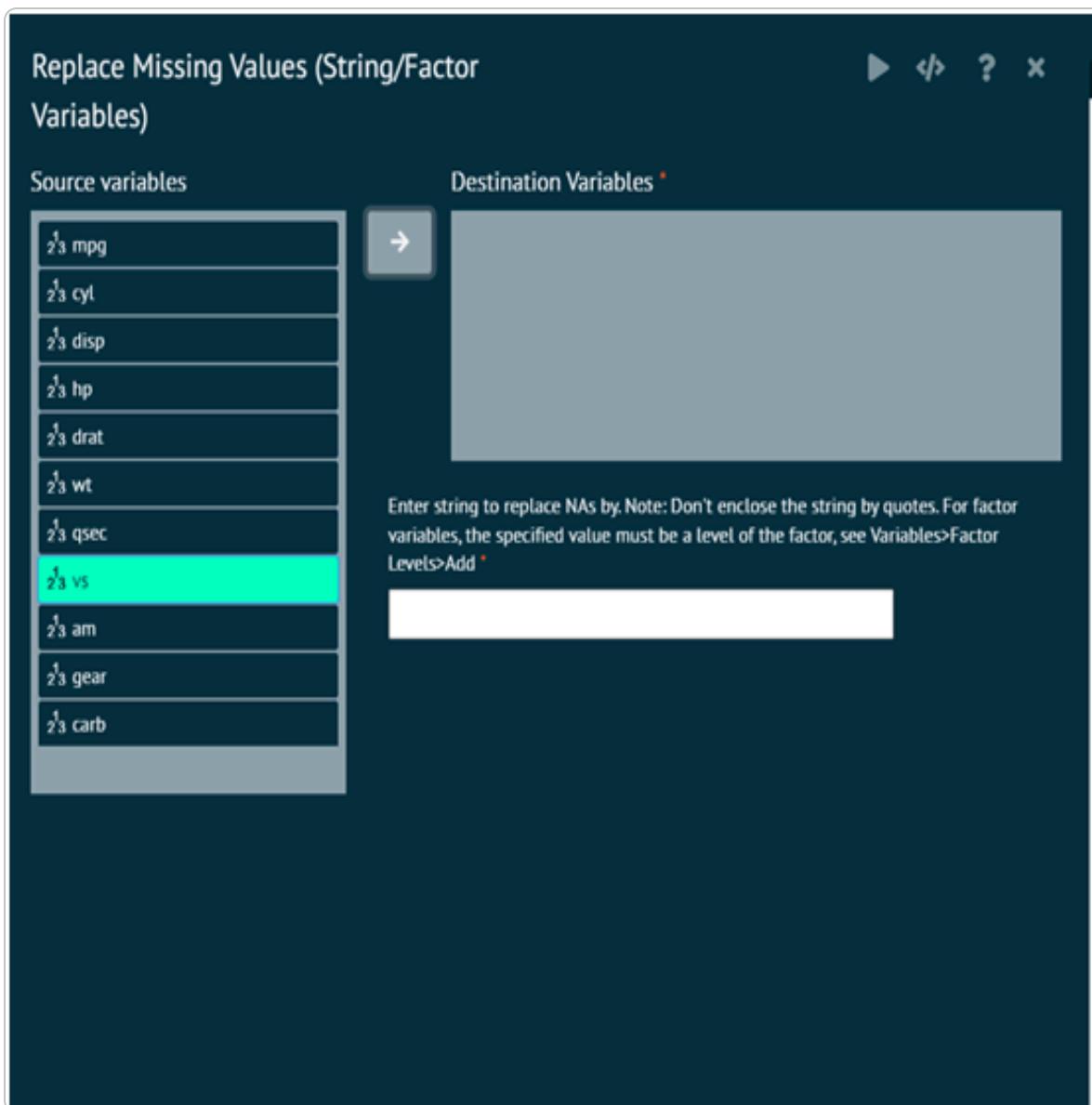
Specify the number of positions to lag or lead by. For example, a lagged value of 1 would extract the previous value and a lagged value of 2 would extract the value 2 positions previous.

- Required R Packages: dplyr

Missing Values

Character/Factor

Replace missing values in the variables selected by the specified value. When using the dialog, user doesn't have to enclose the string in double quotes



alt text

Fill Values Downward or Upward

This dialog fills in missing values in dataset columns by using the previous entry in each column. This can be useful in cases where values are not repeated, but recorded each time they change. Typically, this means the dataset is sorted in a meaningful way. The variables where values are filled in will be overwritten.

The arguments used in executing the dialog are given as follows.

Variables to Fill In Values

Specify variables for which missing values will be filled in

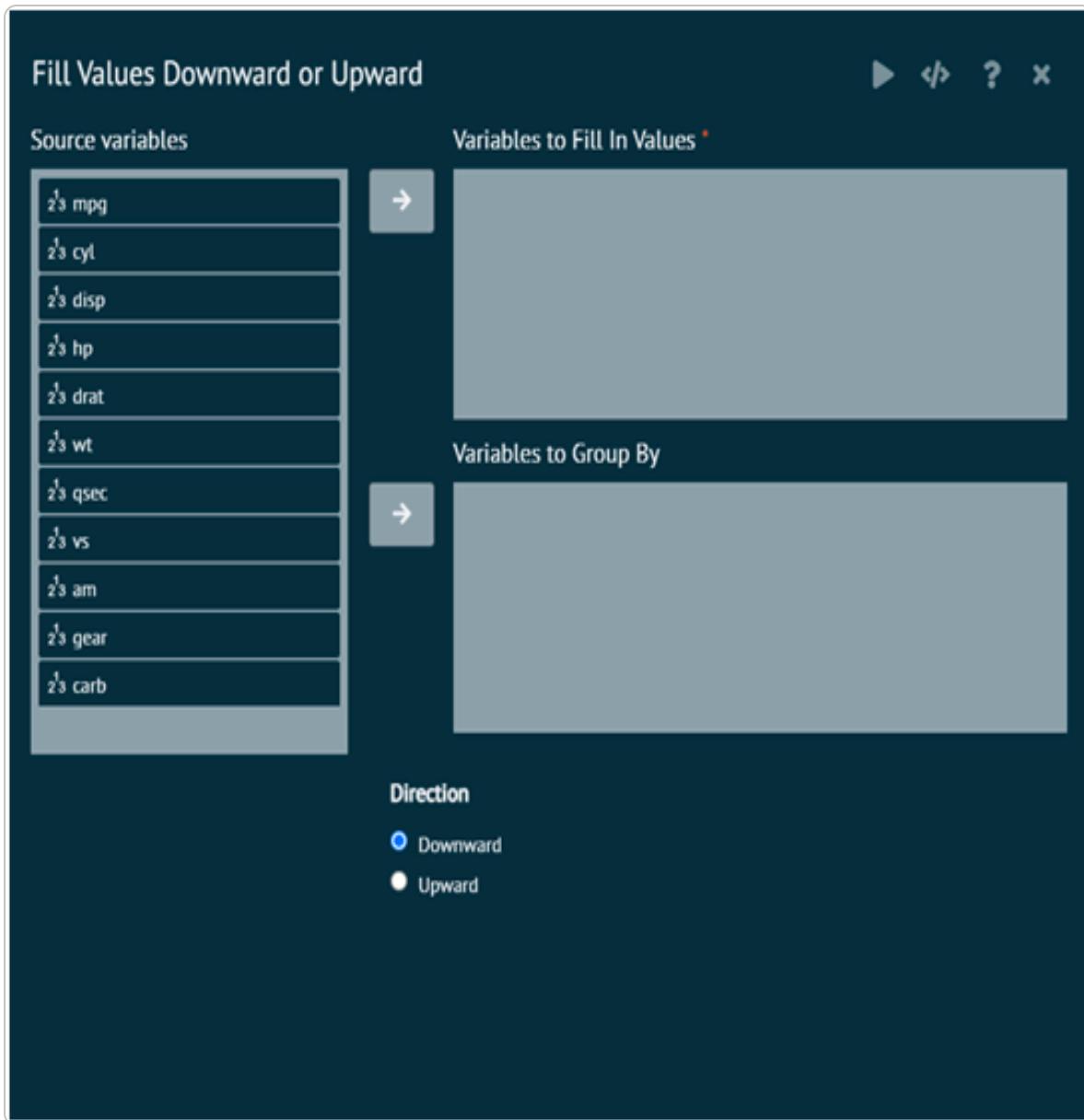
Variables to Group By

Specify variables that group rows together. Missing values will be filled in within groups defined by these variables. For example, grouping by a subject identifier would fill in values within subjects.

Direction

Specify the direction for which the values will be filled in.

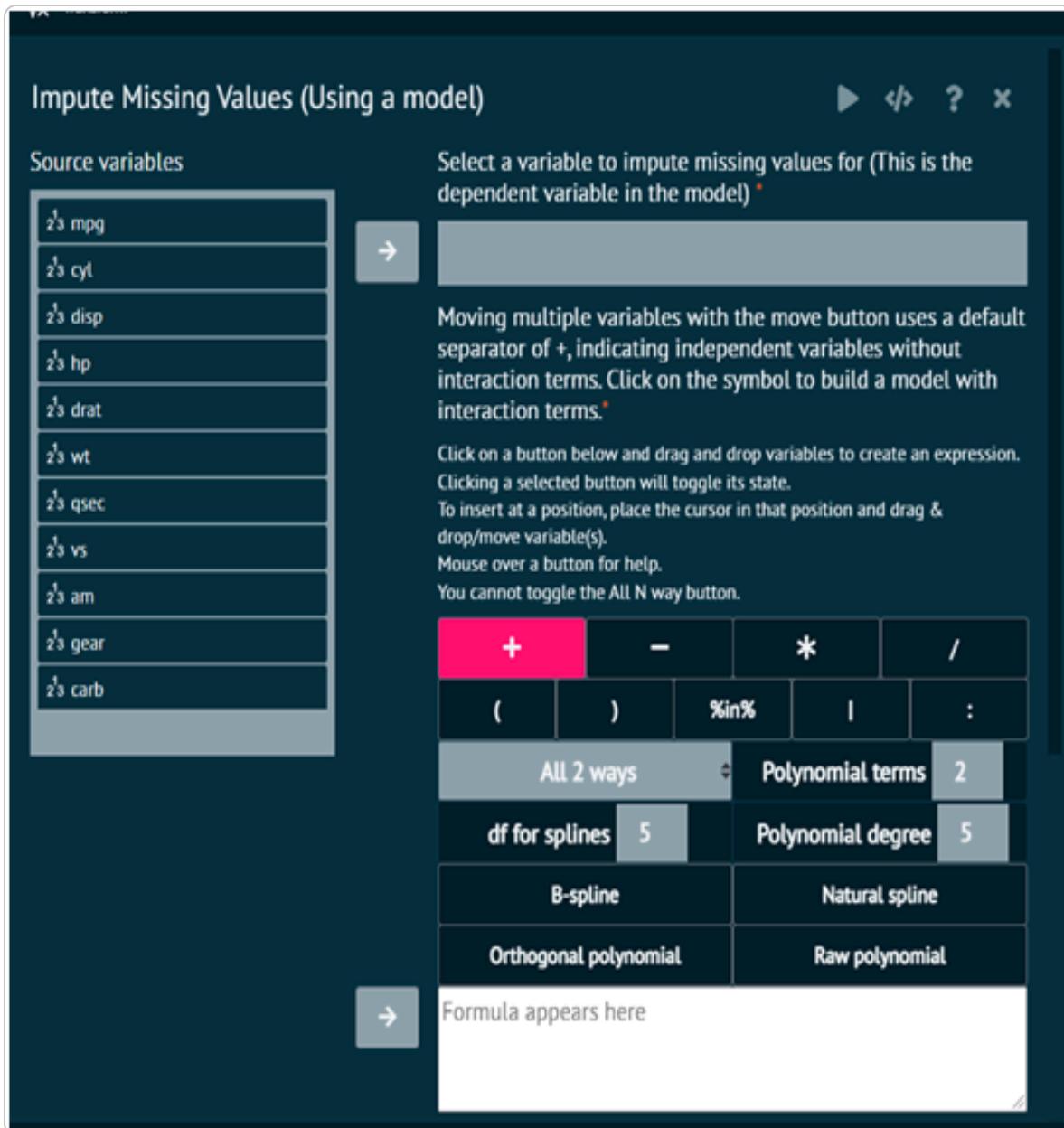
- i** R Packages Required: tidyverse



alt text

Model Imputation

BioStat Prime first constructs a model using the variable to impute values for as the dependent variable. It then uses the constructed model to predict values and replace missing values in the dependent variable by the predicted values.



alt text

The simputation package offers a number of commonly used single imputation methods, each with a similar simple interface. The following imputation methodology is supported.

- A**
- linear regression • robust linear regression • ridge/elasticnet/lasso regression
- CART models (decision trees) • Random forest • Multivariate imputation •
- Imputation based on the expectation-maximization algorithm • missForest (iterative random forest imputation) • Donor imputation (including various donor pool specifications) • k-nearest neighbour (based on gower's distance) •
- sequential hotdeck (LOCF, NOCB) • random hotdeck • Predictive mean

matching • Model based (optionally add [non]parametric random residual) •
Other (groupwise) median imputation (optional random residual)

- ⚠ Proxy imputation: copy another variable or use a simple transformation to compute imputed values.

Numeric

Replace missing values in variables selected by the operation selected i.e. median, mean, min, max

Replace Misssing values (Numeric variables)

Missing values (NAs) in the variables selected are replaced by applying a function i.e. median, mean, min, max or the value specified.

Source variables

13 mpg
13 cyl
13 disp
13 hp
13 drat
13 wt
13 qsec
13 vs
13 am
13 gear
13 carb

Select Variables to Replace Missing Values for *

Select a function or specify a value to replace NAs

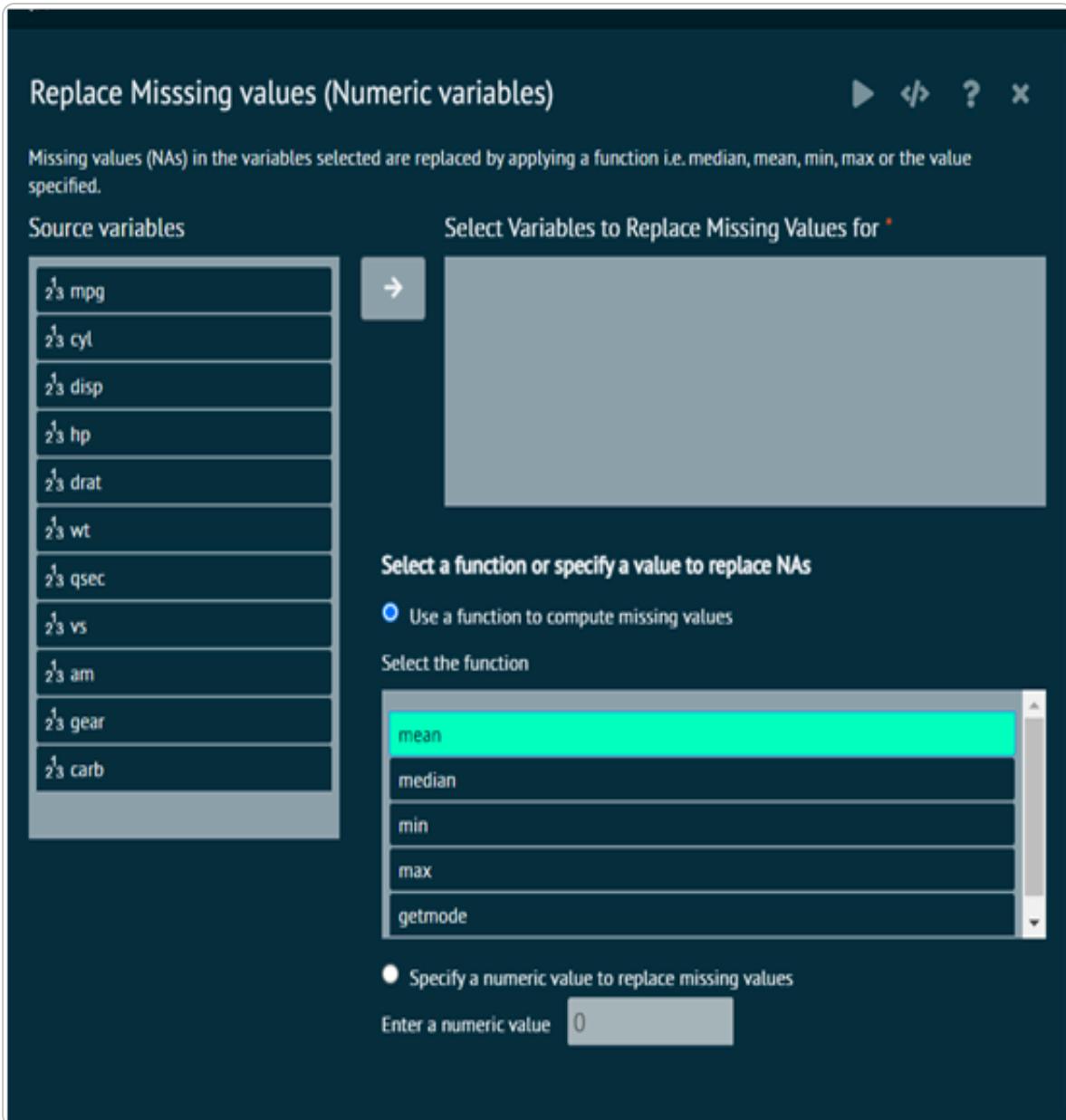
Use a function to compute missing values

Select the function

mean
median
min
max
getnode

Specify a numeric value to replace missing values

Enter a numeric value 0



alt text

⚠ Arguments

var

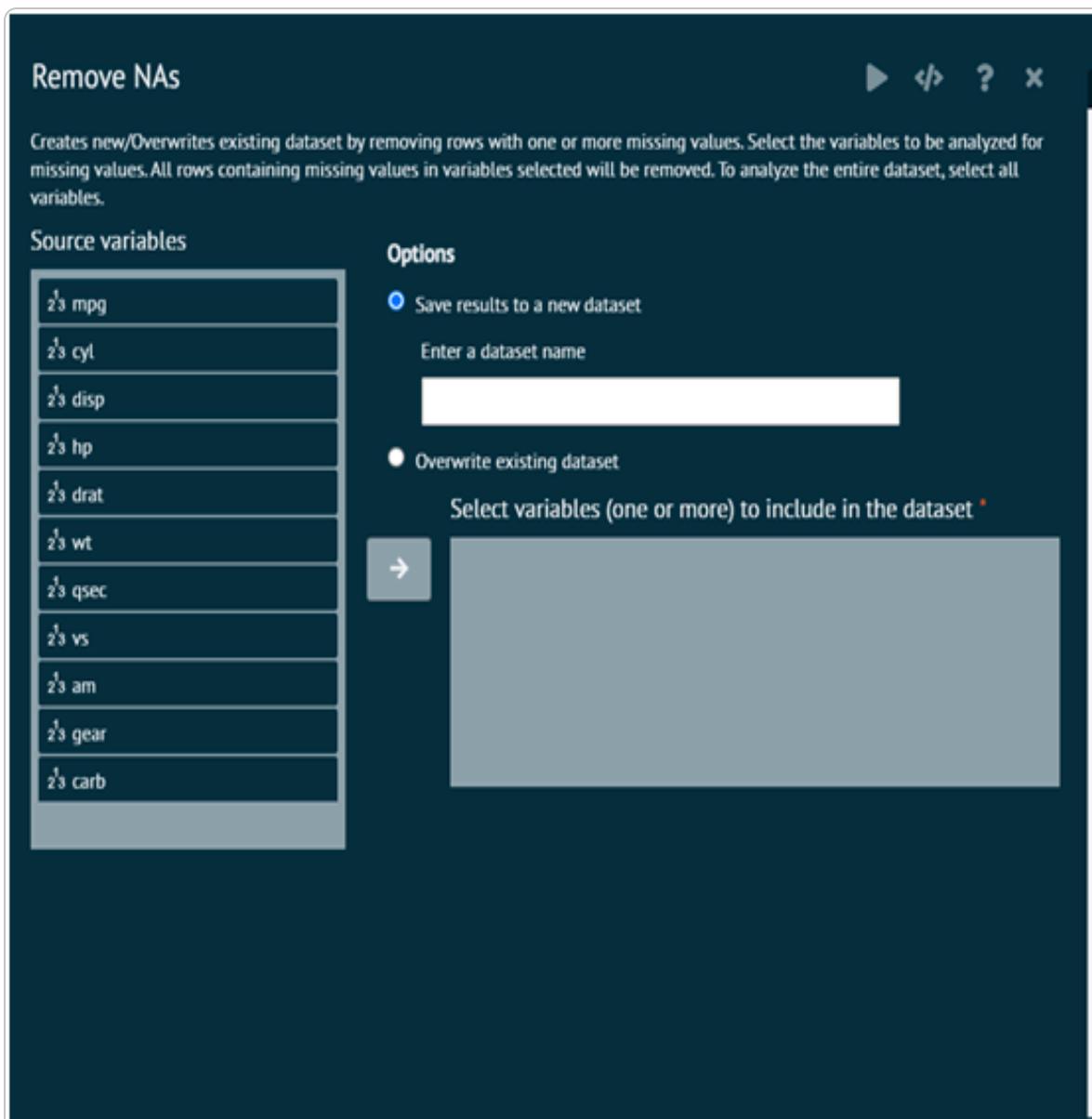
Character string representing the numeric variable with missing values (na), for e.g.
var = c('sales')

Dataset

The dataset that contains the variable var

Remove NAs

Remove missing values/NA from dataset/dataframe Creates new/Overwrites existing dataset by removing rows with one or more missing values for the columns/variable names selected



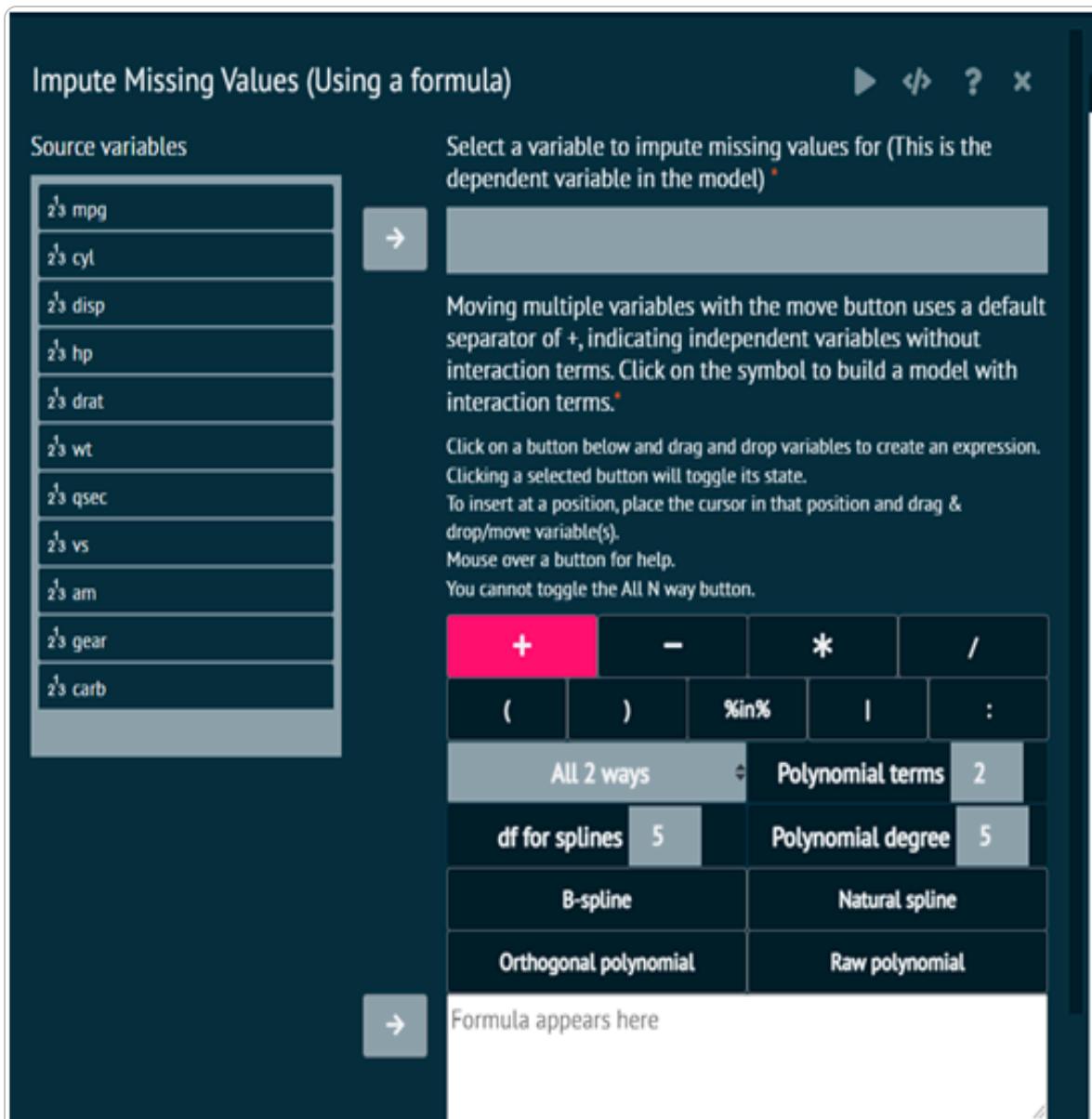
alt text

Arguments

object: an R object.

Impute Missing Values using a formula

Construct a formula to replace missing values. For example user builds a regression model to develop estimates for the missing values, once the equation is generated, user can plug the equation into the dialog and only the missing values in the variable selected will be computed.



alt text

Arguments

var

The name of the variable in dataset where missing values are to be replaced for e.g.
`var=c("sales")`. The variable must be of class numeric

Dataset

The dataset/dataframe that contains the variable var

Expression

The expression used to replace the missing value, in the example above its `var2*4+1.32`

Rank Variable(s)

RANKS WILL BE STORED IN NEW VARIABLES WITH THE PREFIX OR SUFFIX SPECIFIED

Six variations on ranking functions, mimicking the ranking functions described in SQL2003. They are currently implemented using the built in rank function, and are provided mainly as a convenience when converting between R and SQL.

All ranking functions map smallest inputs to smallest outputs.

- ⓘ Use `desc()` to reverse the direction.

Rank Variable(s)

▶ ⌂ ? ×

Enter a suffix or prefix for the new ranked variables

Suffix

Prefix

Enter a suffix/prefix *

Source variables

23 mpg
23 cyl
23 disp
23 hp
23 drat
23 wt
23 qsec
23 vs
23 am
23 gear
23 carb

Select the variable(s) to rank *



Optionaly select variable(s) to rank values within

→

Specify a ranking function

Select a ranking function, click on help for additional information *

BioStat Prime 2023

alt text



Arguments

1. x: A vector of values to rank. Missing values are left as is. If you want to treat them as the smallest or largest values, replace with Inf or -Inf before ranking.

2. n: number of groups to split up into.

⚠ Details

row_number()

equivalent to rank(ties.method = "first")

min_rank()

equivalent to rank(ties.method = "min")

dense_rank()

like min_rank(), but with no gaps between ranks

percent_rank()

a number between 0 and 1 computed by rescaling min_rank to [0, 1]

cume_dist()

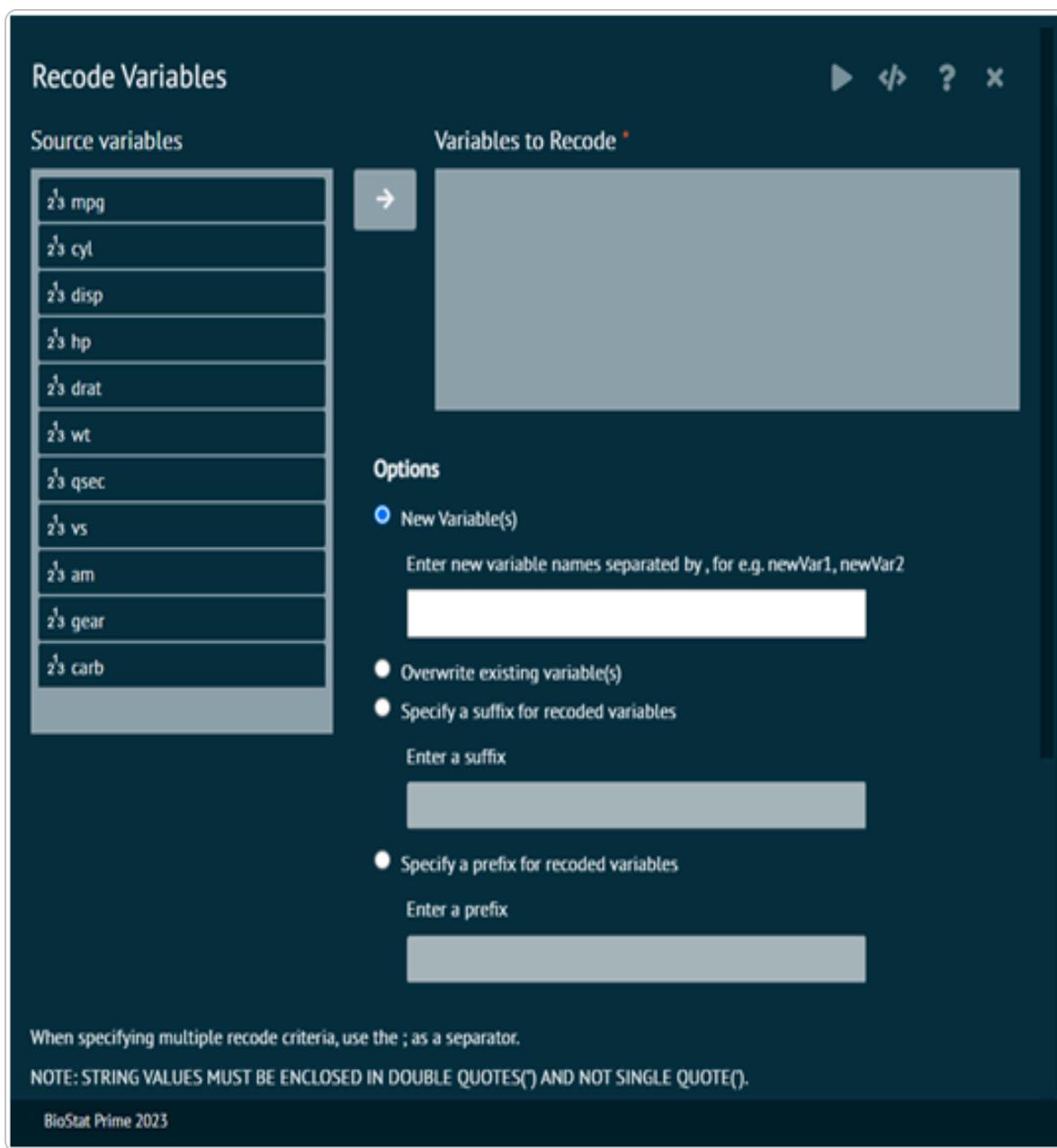
a cumulative distribution function. Proportion of all values less than or equal to the current rank.

ntile()

a rough rank, which breaks the input vector into n buckets.

Recode Variables

Recodes one or more a numeric vector, character vector, or factors according to recode specifications. User can store the results by overwriting existing variables, specifying new variable names to store recoded values or choosing to store the recoded values in new variables with a suitable prefix or suffix. the prefix or suffix will be applied to the existing variable name.



alt text

Arguments

colNames

A character vector containing one or more variables in the dataset to recode

newColNames

A character vector containing the names of the new columns.

OldNewVals

A character string of recode specifications in the form oldval1,newval1,
oldval2,newval2

NewCol

A Boolean indicating whether recoded values are stored in new variables (TRUE) or
existing variables are overwritten(FALSE).

prefixOrSuffix

Specify if user wants to store the recoded values in new variables prefixed or
suffixed with the name user specifies. Enter prefix or suffix.

prefixOrSuffixString

Enter a string to use as a prefix or suffix to the existing variable name. Recoded
values will be stored in these variables.

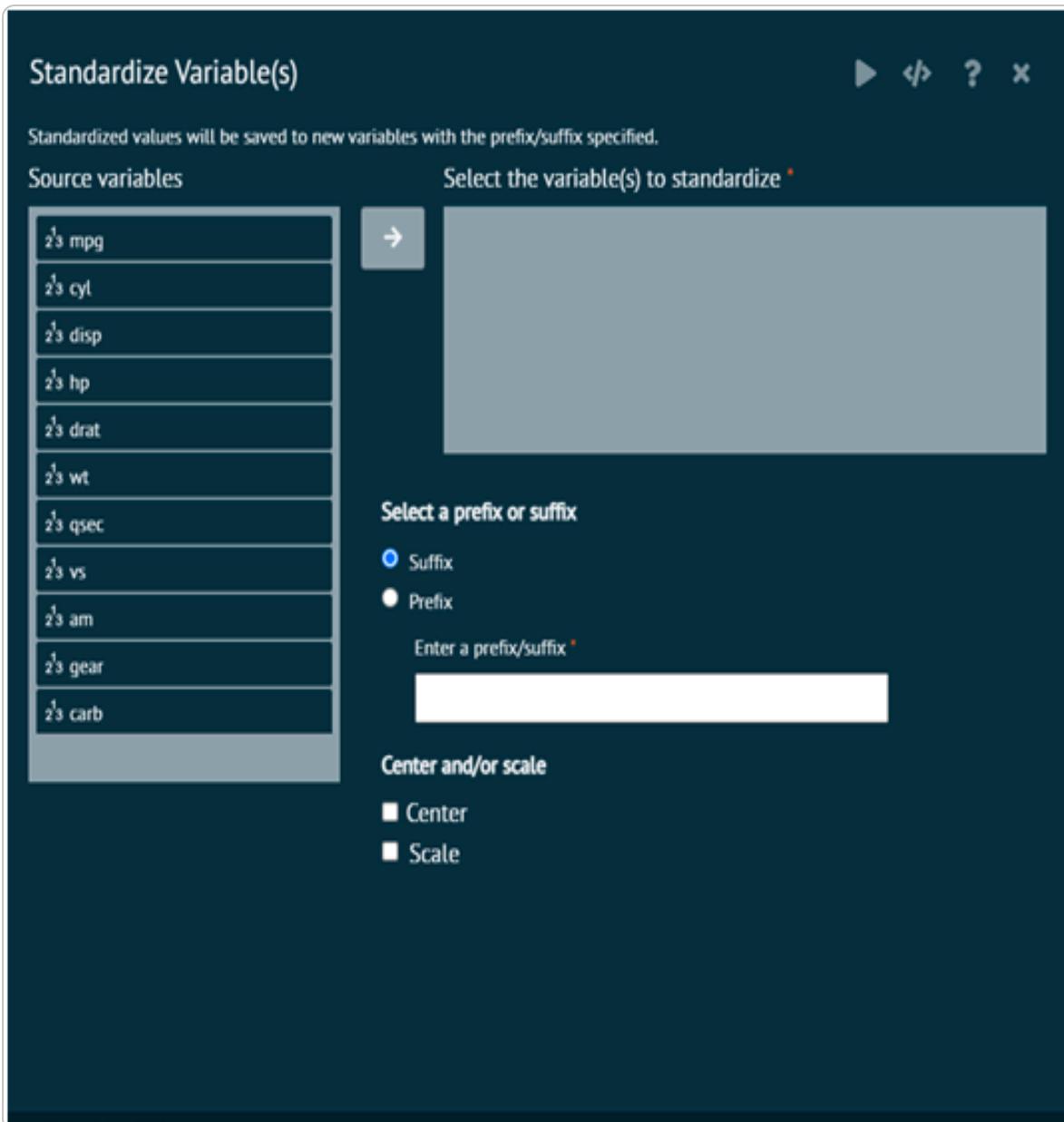
dataSetNameOrIndex

The dataset/dataframe name

- i** Note: BioStat Prime will not convert from numeric to factor. When a numeric is recoded, it will remain a numeric, when a factor variable is recoded it will remain a factor.

Standardize Variable(s)

Standardizes variables (z scores). The standardized values are stored in new variables with either the prefix or suffix of the original variables. The option is provided to center and/or scale.



alt text

⚠ Arguments

vars

One or more variables to standardize. Only numeric variables (not factors) supported.

center

If center is TRUE then centering is done by subtracting the column means (omitting NAs) of x from their corresponding columns, and if center is FALSE, no centering is done.

scale

If scale is TRUE then scaling is done by dividing the (centered) columns of x by their standard deviations if center is TRUE, and the root mean square otherwise. If scale is FALSE, no scaling is done.

stringToPrefixOrSuffix

A character string that specifies the prefix or suffix to use for the new standardized variables(i.e. new columns in the dataset).

prefixOrSuffix

specify if user wants a prefix or a suffix

datasetname

The dataset/dataframe name

- i** Note: BioStat Prime will not convert from numeric to factor. When a numeric is recoded, it will remain a numeric, when a factor variable is recoded it will

remain a factor.

Transform Variable(s)

Use the drop down to select the operation (log, log10,as.numeric...) to transform the selected variables. User can overwrite existing variables or create new variables by specifying a prefix/suffix.

Transform Variable(s) ▶ ⌂ ? ×

Use the drop down to select the operation (log, log10,as.numeric...) to transform the selected variables. You can overwrite existing variables or create new variables by specifying a prefix/suffix.

Source variables

- 13 mpg
- 13 cyl
- 13 disp
- 13 hp
- 13 drat
- 13 wt
- 13 qsec
- 13 vs
- 13 am
- 13 gear
- 13 carb

Select the variable(s) to transform *

Select an operation to apply *

- log10
- log
- log2
- abs
- ceiling

Create new or overwrite existing variables

Specify a suffix

Enter a suffix

Specify a prefix

Enter a prefix

alt text

Arguments

1. var: The variable to be transformed
2. Dataset: The dataset that contains the variable var

Analysis

It's one of the functions in main menu that facilitates the data analysis and statistical calculations. The main feature of this tab is that it provides a wide range of tests and statistical functions to the user.

The statistical techniques focus on the **design, analysis, and interpretation of data related to biology, medicine, public health, and other health sciences.**

It involves the application of statistical methods and techniques to address research questions and draw meaningful conclusions from data in these fields.



BioStat Prime aids the researchers in these techniques via different functions that are present in the Analysis tab of the main menu.

The functions in the analysis tab are explained in the next section.

Cluster

In statistics, clustering is a technique used to group similar data points into clusters or groups based on certain criteria.

⚠ The goal of clustering is to identify patterns or structures within a dataset by grouping data points that are similar to each other than to those in other clusters.

There are various clustering algorithms, each with its own approach to defining similarity and forming clusters.

BioStat Prime comes up with a platform to perform the algorithms to aid users in their analysis.

Hierarchical Clustering

This sub menu provides Hierarchical cluster analysis on a set of dissimilarities and methods for analyzing it.

Hierarchical clustering builds a hierarchy of clusters, creating a tree-like structure (dendrogram) that shows the relationships between clusters at different levels.

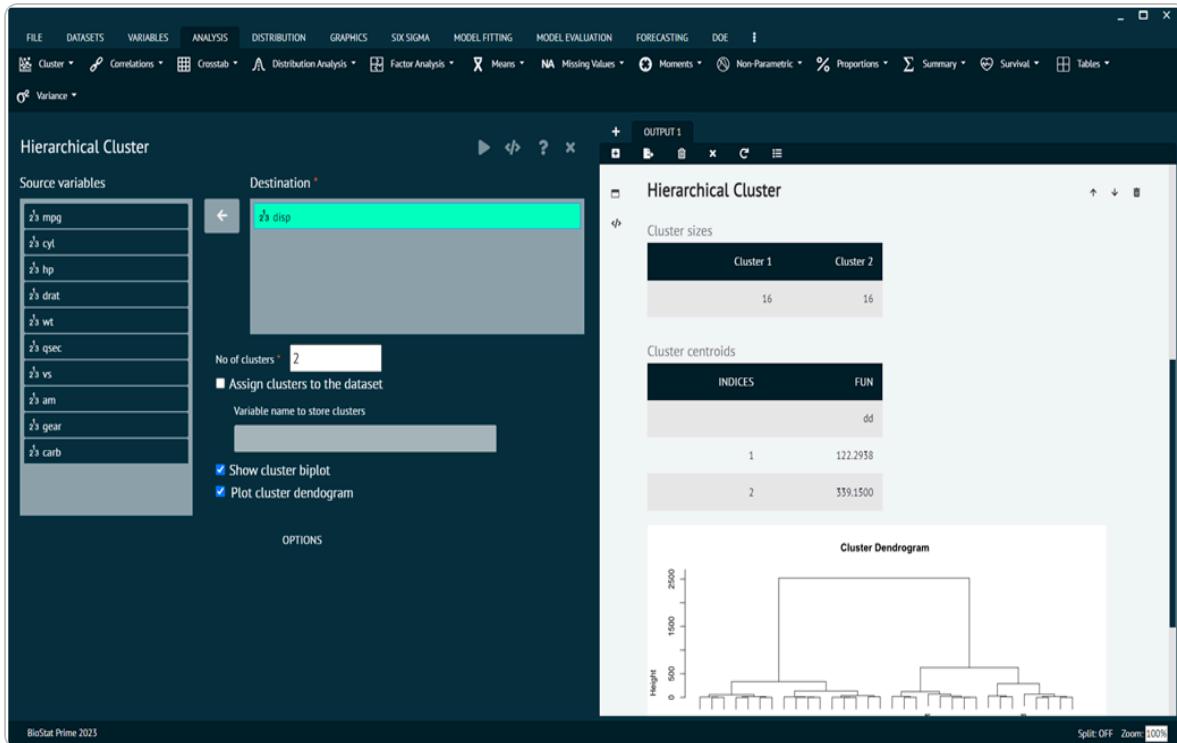
To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset → Click on the analysis tab in main menu → Select CLUSTER button → Select Hierarchical Cluster → This leads to the analysis technique in the dialog → Select the source variable → Write no. of clusters values → Execute the dialog.

The result of the analysis will be visible in the output. Users can also decide whether to **assign cluster values to dataset, plot cluster dendrogram, show cluster bi plot.**

The options tab at the bottom leads the user to further methods and metrics that the user can choose according to the requirements.



alt text

⚠ Arguments

1. **varsToCluster:** The variables to analyze
2. **method:** the agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).
3. **noOfClusters:** The number of clusters desired
4. **plotDendogram:** Plot a dendrogram True or false
5. **assignClusterToDataset:** Save the cluster assignments to the dataset
6. **label:** name for the new variable that stores the cluster assignments
7. **plotBiplot:** plot Biplot TRUE or FALSE

K-Means Clustering

This sub menu performs K-means clustering.

K-Means is a popular partition clustering algorithm that aims to partition data into K clusters. It iteratively assigns data points to clusters and updates cluster centroids until convergence.

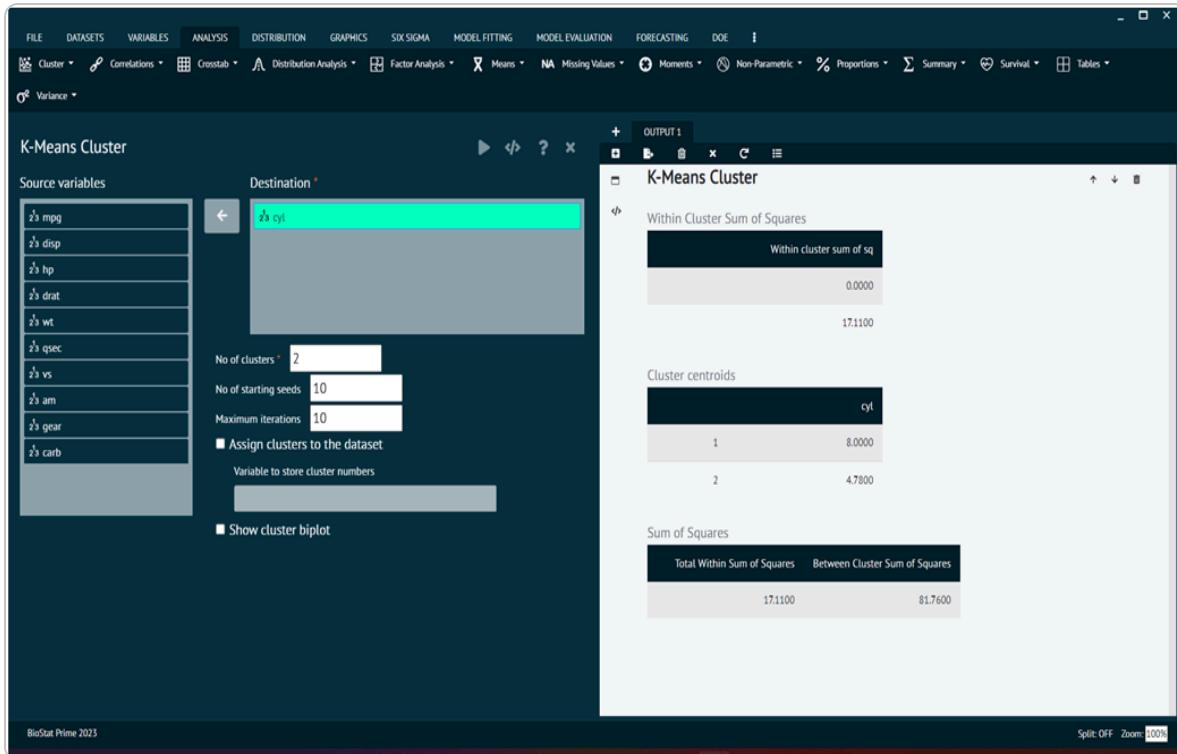
- ⚠ Partition clustering divides the data into non-overlapping clusters in a single step.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select CLUSTER button -> Select K-Means Cluster -> This leads to the analysis technique in the dialog -> Select the source variable -> Write no. of clusters values -> Execute the dialog.

The result of the analysis will be visible in the output. User can also decide whether to assign cluster values to dataset, show cluster bi plot, no. of starting seeds, maximum iterations.



alt text

⚠ Arguments

1. **vars:** The variables to analyze in a vector of form `c('var1','var2'...)`
2. **centers :**either the number of clusters, say `k`, or a set of initial (distinct) cluster centers. If a number, a random set of (distinct) rows in `x` is chosen as the initial centers.
3. **iter.max:** the maximum number of iterations allowed.
4. **num.seeds:** The number of different starting random seeds to use. Each random seed results in a different k-means solution.
5. **storeClusterInDataset:** Save the cluster assignments to the dataset
6. **varNameForCluster:** The variable names for the assigned clusters
7. **dataset:** The dataset to analyze

Correlations

Correlation in statistics refers to the statistical relationship or association between two or more variables. The goal of correlation analysis is to measure the strength and direction of a linear relationship between variables. It quantifies how changes in one variable are associated with changes in another variable.

BioSat Prime provides the user with the functionality to access this relationship by virtue of **Pearson, Spearman test**.

Pearson, Spearman Correlation

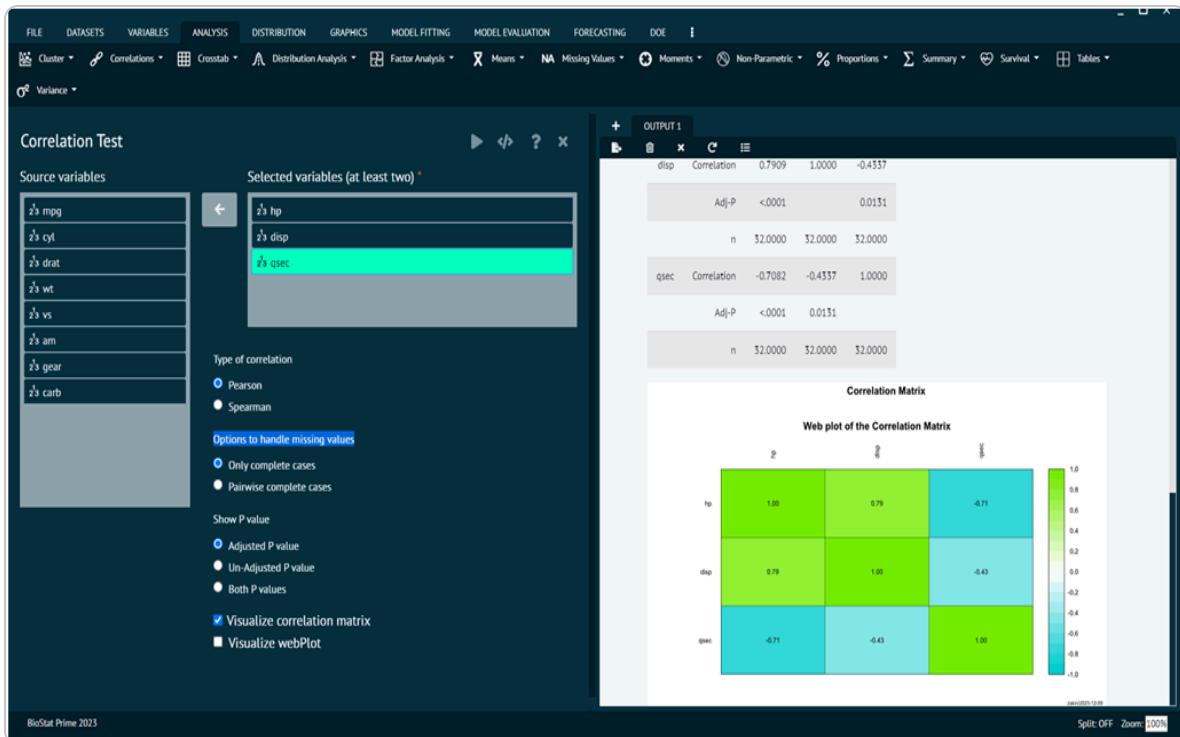
The Pearson correlation test and the Spearman correlation test are statistical methods used to assess the strength and direction of the relationship between two variables. However, they differ in terms of the types of relationships they can detect and the assumptions they make about the data.

⚠ While the Pearson correlation assesses linear relationships between continuous variables, the Spearman correlation is a non-parametric measure that assesses monotonic relationships, making it more robust in certain situations, especially when dealing with non-normally distributed or ordinal data.

⚠ The choice between them depends on the nature of the data and the type of relationship user wants to explore.

This function uses the `rcorr` function in the `Hmisc` package to compute matrices of Pearson or Spearman correlations along with the pair wise `p-values` among the correlations. The `p-values` are corrected for multiple inference using `Holm's method` (see `p.adjust`).

⚠ Observations are filtered for missing data, and only complete observations are used.



alt text

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select correlation -> Select the option namely pearson, seaman -> This leads to the analysis technique in the dialog -> Select the type of correlation in dialog -> Adjust the P value via selecting proper options -> Choose options to handle missing values -> Execute the dialog.

The result of the analysis will be visible in the output. User can also visualise the output by opting for **visualise option in dialog**.

- i** Note that stronger the colour in the output stronger is the correlation.

Crosstab

In statistics, a crosstab, short for "**cross-tabulation**" is a table that displays the relationships between two or more categorical variables. It provides a summary of the distribution of one variable in relation to another.

Crosstabs are particularly useful for analyzing and visualizing the association or dependency between categorical variables. Crosstabs are used when both variables under consideration are categorical. Categorical variables have distinct categories or groups with no inherent order.

The **chi-square test** of independence is often used in conjunction with crosstabs to determine whether there is a statistically significant association between the variables. This test assesses whether the observed frequencies in the cells are significantly different from what would be expected if the variables were independent.

BioStat Prime lays out 3 options in its Crosstab tab, i.e.

Crosstab

The main purpose of a crosstab is to show the frequency distribution of one variable across the levels of another variable.

This sub menu creates crosstab with row, column and layer variables. When multiple row and column variables are specified, BioStat Prime generates a separate cross table for each pair of row and column variables.

 Additionally, the following are displayed

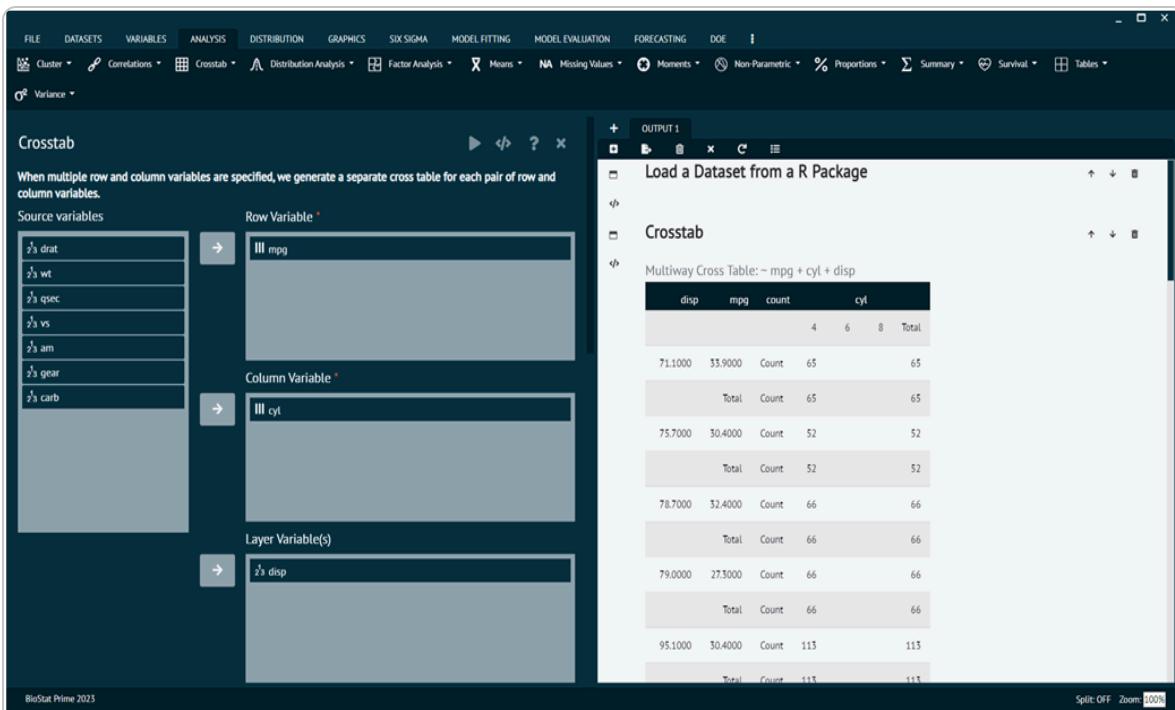
1. Expected counts
2. Row and column percentages
3. Unstandardized, standardized and adjusted residuals
4. Chisq with odds ratio, McNemar and Fisher statistics

⚠ NOTE: BioStat Prime automatically remove all rows where every

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select Crosstab -> The Crosstab contains 3 options, select the first one namely crosstab -> This leads to the crosstab analysis technique in the dialog -> Select the row and column variables -> Execute the dialog.



alt text

The result of the analysis will be visible in the output.

- i** When multiple row and column variables are specified, a separate cross table for each pair of row and column variables is generated.

Crosstab List

This sub menu creates frequency tables in a list format for combinations of one or more variables. Every combination of values across all specified variables will be tabled, with their observed frequencies. The specified variables can be any class, including **numeric**, **continuous variables**.

While this can be used for summary frequencies and percentages, a major use is checking data for inconsistencies.

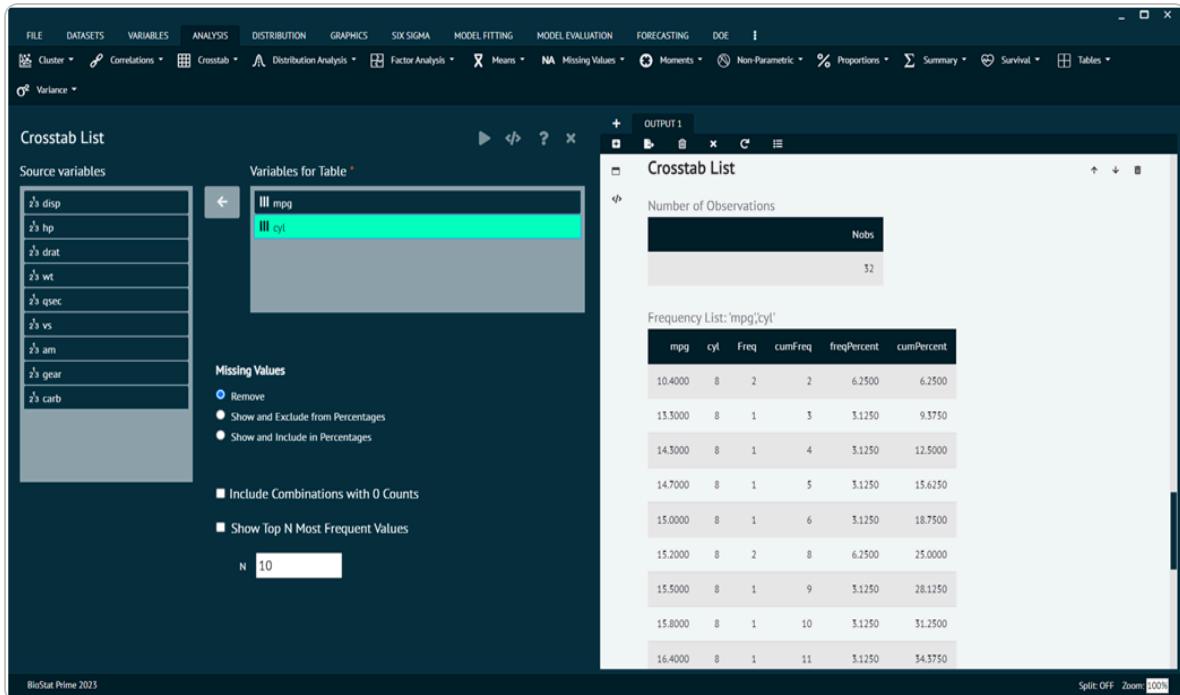
- i** Care should be taken about how many variables to cross-classify and how many possibilities can result, as some tables may take longer to produce.

In addition to raw counts, crosstabs often include percentages. These can be row percentages (percentage within each row) or column percentages (percentage within each column).

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> click on the analysis tab in main menu -> Select Crosstab -> The Crosstab contains 3 options, select the second one namely crosstab list -> This leads to the crosstablist analysis technique in the dialog -> Select variables for the table -> Execute the dialog.



alt text

User can also opt for other options at the bottom related to **frequencies**, **include combinations with 0 counts**, **show top N most frequent values**.

The arguments used in executing the dialog are given as follows.

Variables for Table

Variables to be included in the table, which can be any class. The table will be sorted according to the order of variables in this list. This means if variables A and B are the specified order, then the table will be sorted by levels of A, then levels of B within A.

Missing Values

Remove: Variable value combinations that have NA's will be excluded from the table.

Show and Exclude from Percentages: Variable value combinations that have NA's will be included in the table, but will not be included in percentage computations.

Show and Include in Percentages: Variable value combinations that have NA's will be included in the table and be included in percentage computations.

Include Combinations with 0 Counts

Whether to include variable value combinations that don't exist in the dataset. For example, if variables A and B both have observed values of 1, 2, and 3, but (A, B) combination (1, 3) isn't observed in the data, this option would include a row for the (1, 3) combination with a frequency of 0.

Show Top N Most Frequent Values

If checked, this would create a separate table with the top N most frequent variable combinations. N: How many variable combinations to show for the top N table.

- i R Packages Required: `arsenal`

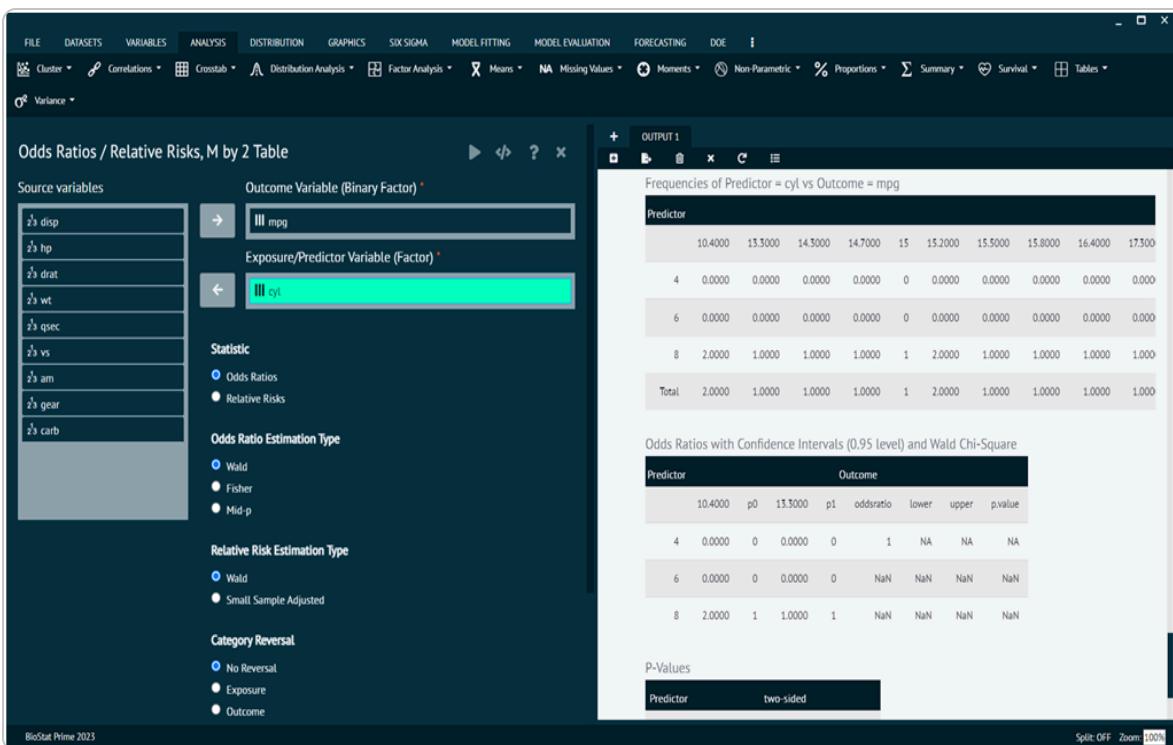
Odds Ratio/ Relative Risks, M by 2 Table

When working with a crosstab (contingency table) that involves categorical variables, measures such as odds ratios, relative risks, and the chi-square test (often referred to as the "chi-square test of independence") are commonly used to assess associations between variables.

- ! The odds ratio is a measure of association between two binary variables. It is commonly used in case-control studies or situations where the outcome is dichotomous. The odds ratio indicates the odds of an event occurring in one group relative to the odds in another group.
- ! The relative risk (RR) is another measure of association, commonly used in cohort studies or situations where the outcome is binary. The relative risk indicates the risk of an event occurring in one group relative to the risk in another group.

A The chi-square test is used to assess whether there is a statistically significant association between two categorical variables. The test involves comparing the observed frequencies in a contingency table with the frequencies that would be expected under the assumption that the variables are independent . A **significant chi-square test suggests that the variables are associated.**

i The choice between odds ratio and relative risk depends on the study design and the nature of the data. Chi-square test is used to determine whether observed associations are statistically significant.



alt text

To analyze all three of them in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select Crosstab -> The Crosstab contains 3 options, select the third one namely Odds Ratio/ Relative

Risks, M by 2 Table -> This leads to the crosstablist analysis technique in the dialog -
> Select variables for the table -> Execute the dialog.

This Sub menu is used to compare probabilities of having a "disease" for one group relative to another, in a ratio of odds (odds ratio) form or probability ratio (relative risk) form.

i The odds of "disease" is defined as the probability of "disease" divided by the probability of "no disease".

! A contingency table of outcome frequencies in each group is provided.

! In addition, a table of odds ratios or relative risks for each group relative to the reference group with confidence intervals and a Wald Chi-Square p-value is included.

! Lastly, a table of p-values is shown with mid-p exact, Fisher's exact, and Wald Chi-Square versions, comparing each group to the reference group.

The arguments used in executing the dialog are given as follows.

Outcome Variable

Binary "disease" (yes/no) variable of interest. By default, the highest category in the sort order is defined as "disease yes".

Exposure/Predictor Variable

Groups to compare. Can have more than 2 groups.

Statistic

Which statistic to compute

Odds Ratio / Relative Risk Estimation Type

Wald (unconditional maximum likelihood), Fisher (conditional maximum likelihood),
Mid-p (median unbiased method), or Small Sample Adjusted

Distribution Analysis

In statistics, a distribution refers to the set of all possible values and their corresponding probabilities or frequencies for a given variable. Understanding the distribution of data is fundamental in statistical analysis. Different statistical tests can be employed to assess whether a given dataset follows a specific distribution, such as the normal distribution.

BioStat Prime brings forth some normality distribution tests under the distribution sub menu in analysis tab of main menu. The distribution tab comprises 7 normality test that are discussed below in detail.

- Users must keep in mind that normality tests are sensitive to sample size, and with large sample sizes, even small departures from normality may lead to rejecting the null hypothesis.

- ⚠ It's essential to consider the context of your data and the specific requirements of user's analysis when interpreting the results of normality tests.

Anderson-Darling Normality Test

The Anderson-Darling test is one such test, and it is specifically used for testing the goodness of fit of a sample to a specified distribution, often the normal distribution.

- The Anderson-Darling test is more sensitive to deviations in the tails of the distribution compared to other normality tests like the Shapiro-Wilk test.

To analyse it in BioStat Prime user must follow the steps as given.

Style

Load the dataset -> Click on the analysis tab in main menu -> Select Distribution tab -> The Distribution tab contains 7 options, select the first one namely Anderson-

Darling Normality test -> This leads to analysis technique in the dialog -> Select variables to target -> Execute the dialog.

The screenshot shows the BioStat Prime 2023 software interface. On the left, the 'Odds Ratios / Relative Risks, M by 2 Table' dialog is open. It has several sections: 'Source variables' (containing 'disp', 'hp', 'drat', 'wt', 'qsec', 'vs', 'am', 'gear', 'carb'), 'Outcome Variable (Binary Factor)' set to 'mpg', 'Exposure/Predictor Variable (Factor)' set to 'cyl', 'Statistic' (radio buttons for 'Odds Ratios' and 'Relative Risks' with 'Relative Risks' selected), 'Odds Ratio Estimation Type' (radio buttons for 'Wald', 'Fisher', and 'Mid-p' with 'Fisher' selected), 'Relative Risk Estimation Type' (radio buttons for 'Wald' and 'Small Sample Adjusted' with 'Wald' selected), and 'Category Reversal' (radio buttons for 'No Reversal', 'Exposure', and 'Outcome' with 'No Reversal' selected). On the right, the 'OUTPUT 1' window displays two tables. The first table, titled 'Frequencies of Predictor = cyl vs Outcome = mpg', shows counts for predictor values 4, 6, and 8 across outcome values 10.4000, 13.3000, 14.3000, 14.7000, 15, 15.2000, 15.5000, 15.8000, 16.4000, and 17.3000. The second table, titled 'Odds Ratios with Confidence Intervals (0.95 level) and Wald Chi-Square', shows odds ratios for predictor values 4, 6, and 8 against outcome values 10.4000, 13.3000, 14.3000, 14.7000, 15, 15.2000, 15.5000, 15.8000, 16.4000, and 17.3000. The P-Values table at the bottom indicates a two-sided test.

alt text

Kolmogorov-Smirnov Normality Test

The Kolmogorov-Smirnov (K-S) test for normality is a non-parametric test used to determine whether a sample comes from a normal distribution. It is based on the **cumulative distribution function (CDF)** of the normal distribution and involves comparing the observed cumulative distribution of the data with the expected cumulative distribution of a normal distribution.

- i** The Kolmogorov-Smirnov test is sensitive to departures from normality in both the center and the tails of the distribution.

- i** When the sample size is small, the test may have limited power to detect deviations from normality.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

- Load the dataset -> Click on the analysis tab in main menu -> select Distribution tab -> The Distribution tab contains 7 options, select the second one namely Kolmogorov-Smirnov Normality test -> This leads to analysis technique in the dialog -> Select variables to target -> Execute the dialog.

The screenshot shows the BioStat Prime software interface. The main window title is "Kolmogorov-Smirnov Test of Normality". On the left, under "Source variables", the "disp" item is highlighted with a green background. On the right, under "Target variables", the "wt" item is also highlighted with a green background. The output window displays the test results for the variable "wt". It shows a warning message about ties being present in the data. The "Test Result" section provides the test statistic "D" (0.9349) and the p-value (0 ***). Below this, there is additional information about the test method (One-sample Kolmogorov-Smirnov test), alternative hypothesis (two-sided), and p-value (< 2.2e-16). A note at the bottom of the output window states: "Warning: ties should not be present for the Kolmogorov-Smirnov test".

alt text

Shapiro-Wilk Normality Test

The Shapiro-Wilk test is a statistical test used to assess whether a sample comes from a normally distributed population. It is commonly used for **testing the assumption of normality** in statistical analyses.



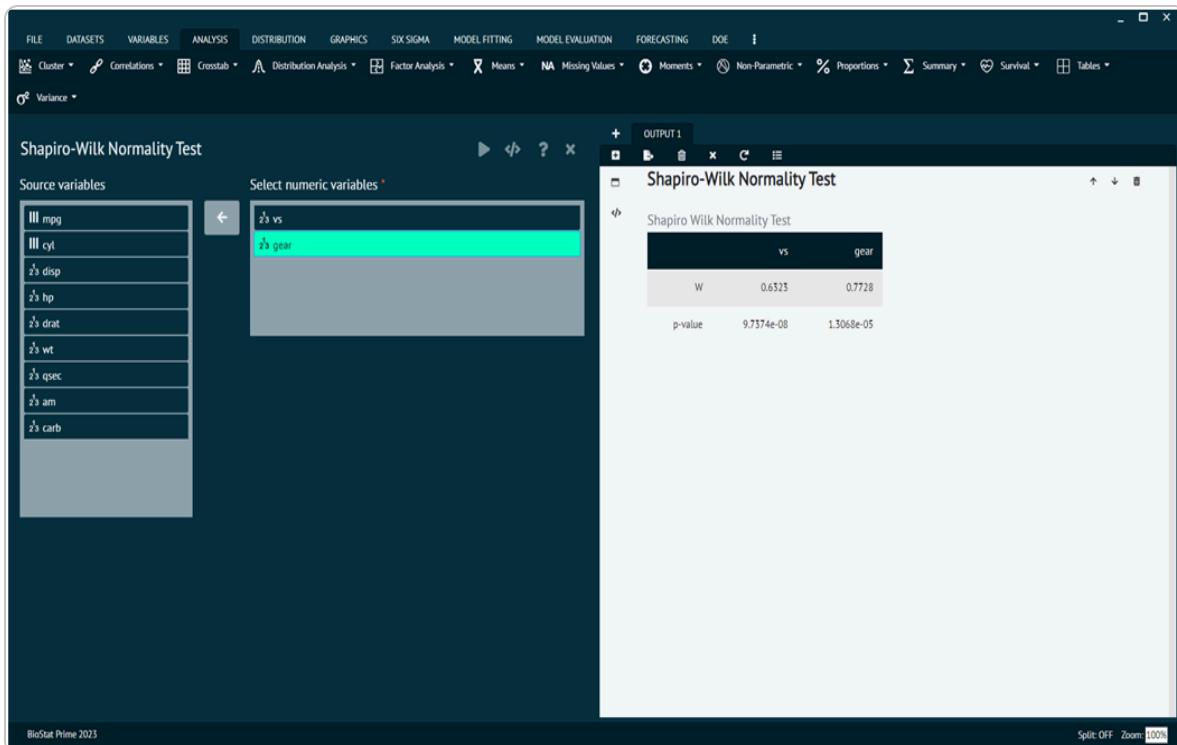
The test is particularly useful when dealing with smaller sample sizes.

⚠ The Shapiro-Wilk test is sensitive to deviations from normality, especially in the tails of the distribution.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select Distribution tab -> The Distribution tab contains 7 options, select the third one namely Shapiro-Wilk Normality Test -> This leads to the analysis technique in the dialog -> Select variables to target -> Execute the dialog.



alt text

Distribution Fit

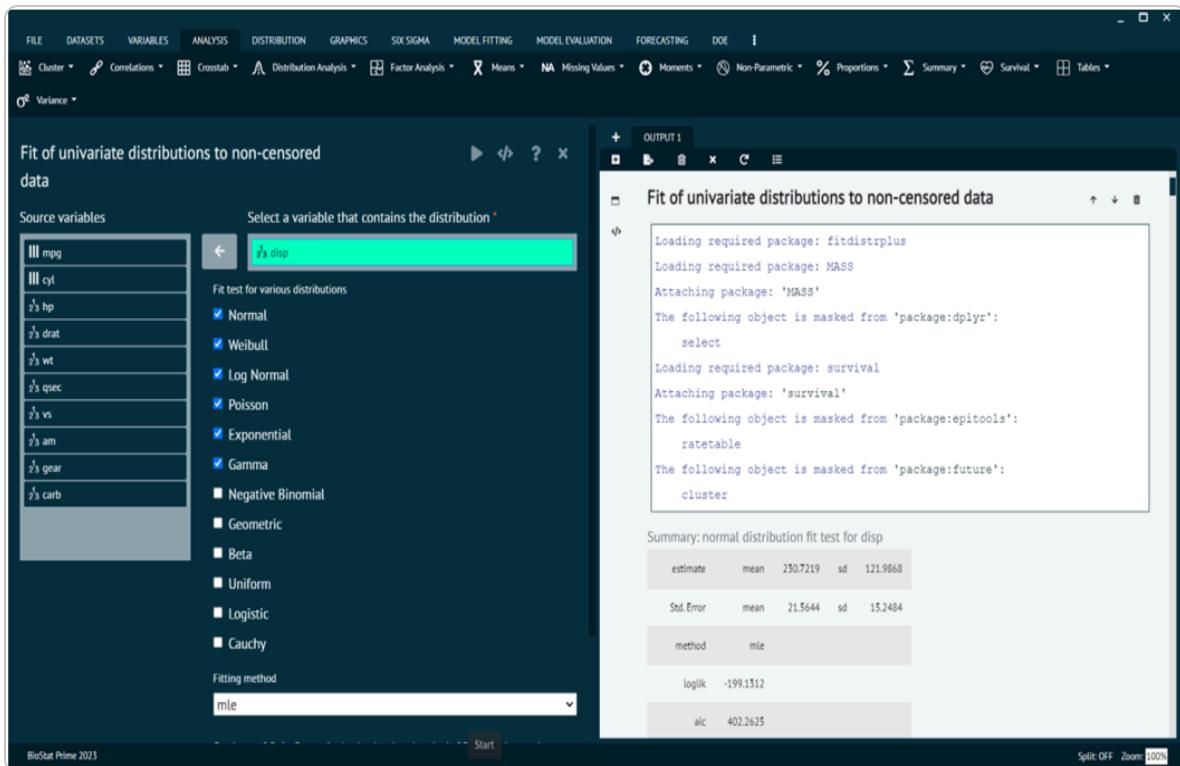
Distribution fitting is a statistical technique used to model and describe the distribution of a dataset by finding the probability distribution that best fits the observed data.

A The goal is to identify a parametric distribution (such as normal, exponential, gamma, etc.) that provides a good representation of the data.

To analyse it in BioStat Prime user must follow the steps as given.

Step

Load the dataset -> Click on the analysis tab in main menu -> Select Distribution tab -> The Distribution tab contains 7 options, select the fifth one namely Distribution Fit -> This leads to the analysis technique in the dialog -> Select variables to target -> Fit the test for various distributions -> Execute the dialog.



alt text

The various distributions are visible in the output window.

The four possible fitting methods are described below:

mle

When method="mle" (default) Maximum likelihood estimation consists in maximizing the log-likelihood. A numerical optimization is carried out in mledist via optim to find the best values (see mledist for details).

mme

When method="mme" Moment matching estimation consists in equalizing theoretical and empirical moments. Estimated values of the distribution parameters are computed by a closed-form formula for the following distributions : "norm", "Inorm", "pois", "exp", "gamma", "nbinom", "geom", "beta", "unif" and "logis". Otherwise the theoretical and the empirical moments are matched numerically, by minimization of the sum of squared differences between observed and theoretical moments. In this last case, further arguments are needed in the call to fitdist: order and memp (see mmedist for details).

qme

When method = "qme" Quantile matching estimation consists in equalizing theoretical and empirical quantile. A numerical optimization is carried out in qmedist via optim to minimize of the sum of squared differences between observed and theoretical quantiles. The use of this method requires an additional argument probs, defined as the numeric vector of the probabilities for which the quantile(s) is(are) to be matched (see qmedist for details).

mge

When method = "mge" Maximum goodness-of-fit estimation consists in maximizing a goodness-of-fit statistics. A numerical optimization is carried out in mgelist via optim to minimize the goodness-of-fit distance. The use of this method requires an additional argument gof coding for the goodness-of-fit distance chosen. One can use the classical Cramer-von Mises distance ("CvM"), the classical Kolmogorov-Smirnov distance ("KS"), the classical Anderson-Darling distance ("AD") which gives more weight to the tails of the distribution, or one of the variants of this last distance

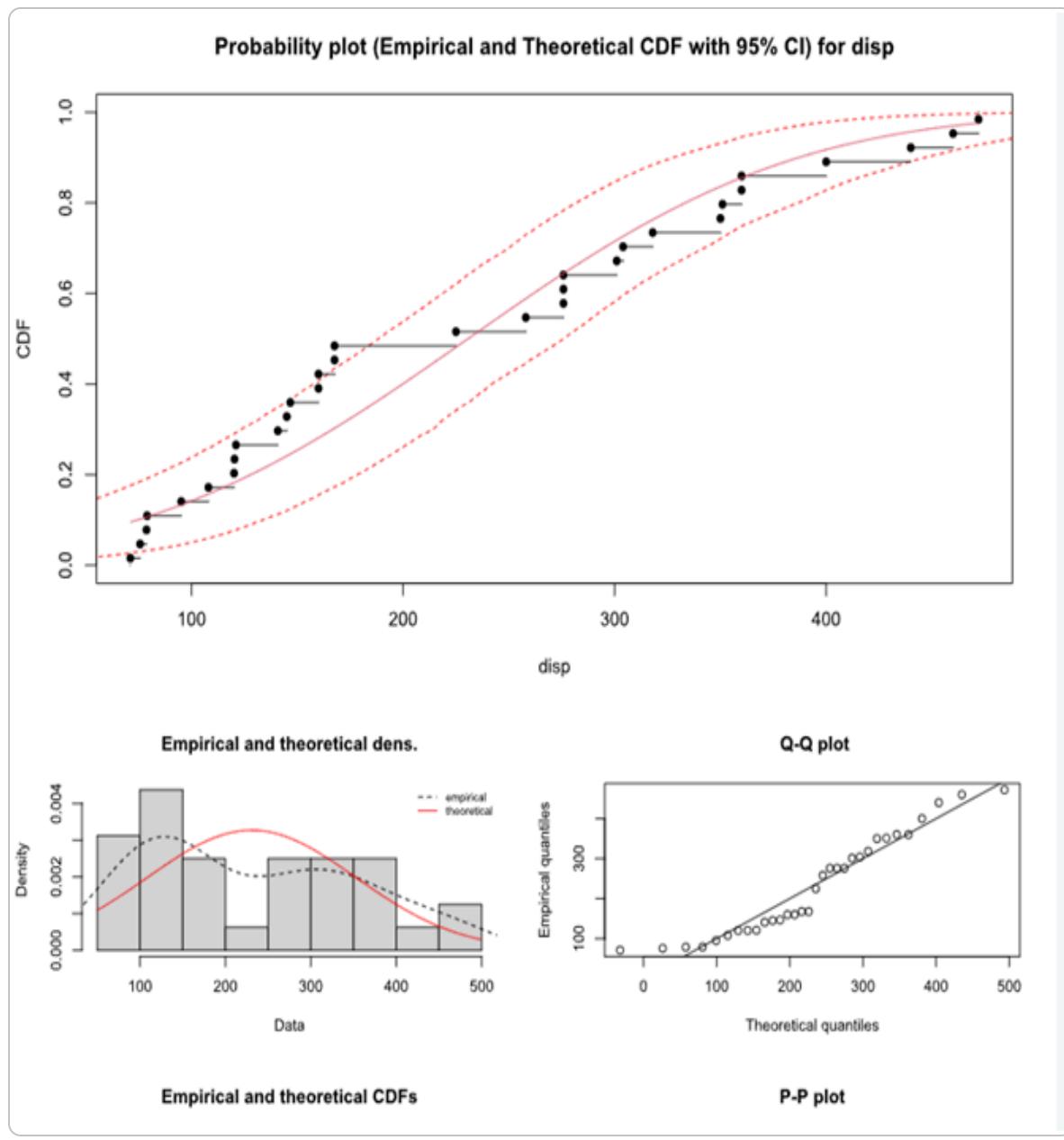
proposed by Luceno (2006) (see mgelist for more details). This method is not suitable for discrete distributions.

mse

When method = "mse" Maximum goodness-of-fit estimation consists in maximizing the average log spacing. A numerical optimization is carried out in msedist via optim.

- ⚠ convergence is an integer code for the convergence of optim/constrOptim defined as below or defined by the user in the user-supplied optimization function. 0 indicates successful convergence. 1 indicates that the iteration limit of optim has been reached. 10 indicates degeneracy of the Nelder-Mead simplex. 100 indicates that optim encountered an internal error.
- ⚠ Goodness-of-fit statistics are computed by gofstat(). The Chi-squared statistic is computed using cells defined by the argument chisqbreaks or cells automatically defined from data, in order to reach roughly the same number of observations per cell, roughly equal to the argument meancount, or slightly more if there are some ties.
- ⚠ For continuous distributions, Kolmogorov-Smirnov, Cramer-von Mises and Anderson-Darling and statistics are also computed, as defined by Stephens (1986).

Statistics of importance are Cramer-von Mises, Anderson-Darling and Kolmogorov statistics for continuous distributions and Chi-squared statistics for discrete ones ("binom", "nbinom", "geom", "hyper" and "pois")



Distribution Fit with Gamlss

The Gamlss package in R is used for **fitting Generalized Additive Models for Location, Scale, and Shape (GAMLSS)**.

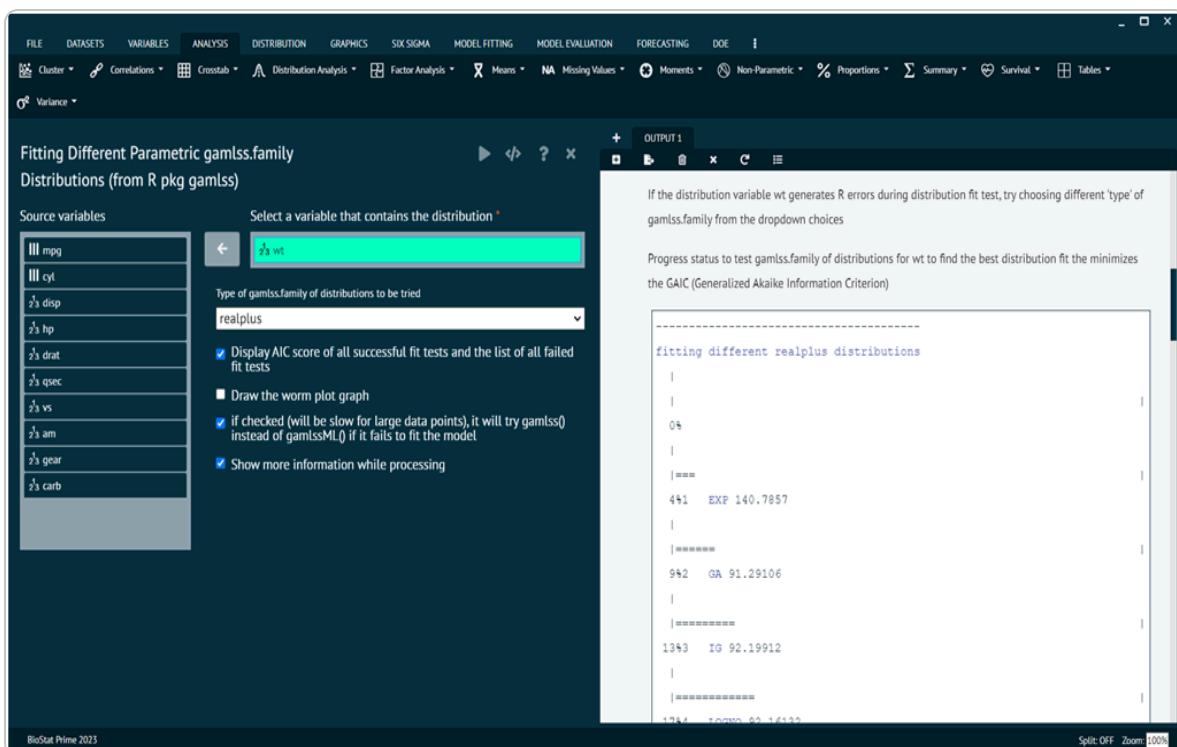
GAMLSS is a flexible framework for modeling distributions and is capable of handling a wide range of distributional shapes.

BioStat Prime utilizes this package of R to aids user to fit different parametric `gamlss.family` distributions from R pkg `gamlss`.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select Distribution tab -> The Distribution tab contains 7 options, select the sixth one namely Distribution Fit with Gamlss -> This leads to the analysis technique in the dialog -> Select variables that contains distribution -> Check the options at the bottom as per the preference -> Execute the dialog.



alt text

The function `fitDist()` is using the function `gamlssML()` to fit all relevant parametric `gamlss.family` distributions, specified by the argument `type`, to a single data vector (with no explanatory variables). The final marginal distribution is the one selected by the generalised Akaike information criterion with penalty `k`. The default is `k=2` i.e AIC which means that the "best" distribution is selected according to the classic AIC. `k` can be set to anything, such as `log(n)` for the BIC (not provided on the dialog at this time)

The following are the different type argument:

realAll

All the gamlss.family (not provided on the dialog at this time) continuous distributions defined on the real line, i.e. realline and the real positive line i.e. realplus

realline

The gamlss.family continuous distributions : "NO", "GU", "RG", "LO", "NET", "TF", "TF2", "PE", "PE2", "SN1", "SN2", "exGAUS", "SHASH", "SHASHo", "SHASHo2", "EGB2", "JSU", "JSUo", "SEP1", "SEP2", "SEP3", "SEP4", "ST1", "ST2", "ST3", "ST4", "ST5", "SST", "GT"

realplus

The gamlss.family continuous distributions in the positive real line: "EXP", "GA", "IG", "LOGNO", "LOGNO2", "WEI", "WEI2", "WEI3", "IGAMMA", "PARETO2", "PARETO2o", "GP", "BCCG", "BCCGo", "exGAUS", "GG", "GIG", "LNO", "BCTo", "BCT", "BCPEo", "BCPE", "GB2"

real0to1

The gamlss.family continuous distributions from 0 to 1: "BE", "BEo", "BEINFO", "BEINF1", "BEOI", "BEZI", "BEINF", "GB1""

counts

The gamlss.family distributions for counts: "PO", "GEOM", "GEOMo", "LG", "YULE", "ZIPF", "WARING", "GPO", "DPO", "BNB", "NBF", "NBI", "NBII", "PIG", "ZIP", "ZIP2", "ZAP", "ZALG", "DEL", "ZAZIPF", "SI", "SICHEL", "ZANBI", "ZAPIG", "ZINBI", "ZIPIG", "ZINBF", "ZABNB", "ZASICHEL", "ZINBF", "ZIBNB", "ZISICHEL"

binom

The gamm family distributions for binomial type data :"BI", "BB", "DB", "ZIBI", "ZIBB", "ZABI", "ZABB"

Distribution Analysis Cullen and Frey Graph

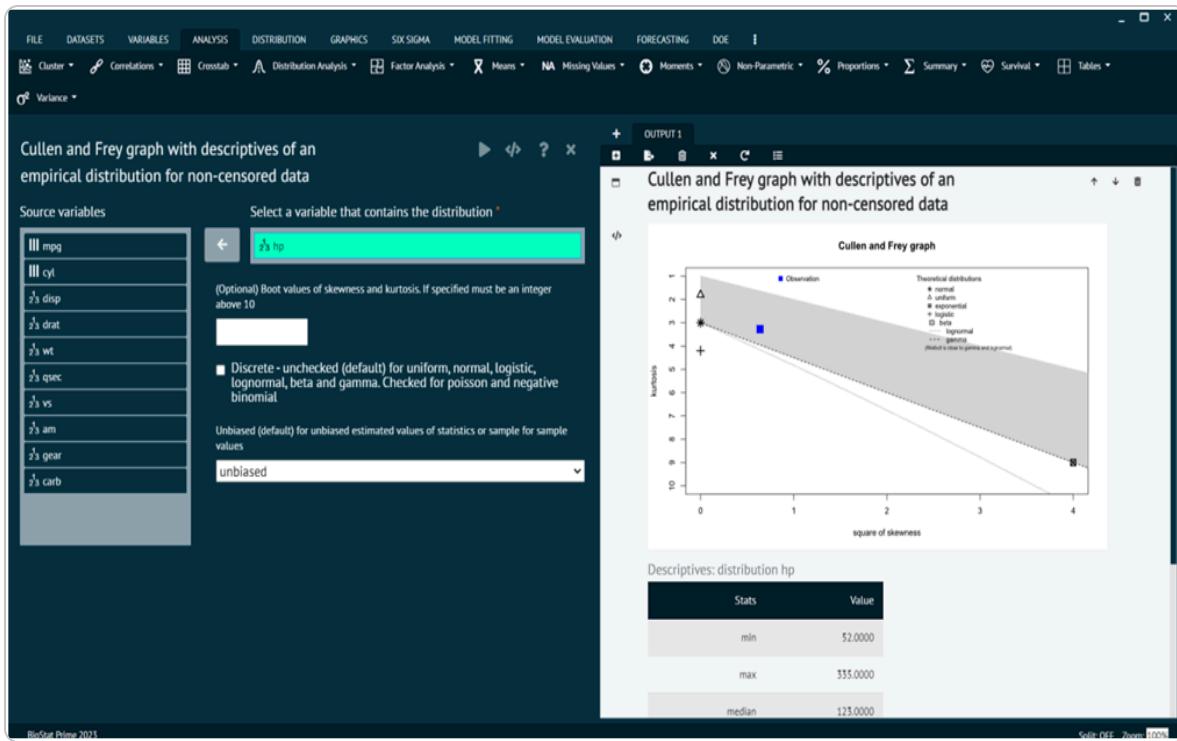
The Cullen and Frey graph, also known as the Cullen and Frey graph for skewness and kurtosis, is a graphical method for assessing the skewness and kurtosis of a dataset. It's a visual tool that helps you quickly inspect the departure from normality in terms of skewness and kurtosis.

BioStat Prime aids users to derive Cullen and frey graph with descriptive of an empirical distribution of a non-censored data.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select Distribution tab -> The Distribution tab contains 7 options, select the seventh one namely Distribution Analysis Cullen and Frey Graph -> This leads to the analysis technique in the dialog -> Select variables that contains distribution -> Check the options at the bottom as per the preference -> Execute the dialog.



alt text

Factor analysis

Factor analysis is a statistical technique used to identify and analyze underlying factors or latent variables that explain the observed correlations among a set of variables.

The goal of factor analysis is to reduce the dimensionality of the data by identifying a smaller number of latent factors that explain the observed correlations among variables. This can simplify the interpretation of complex datasets and help identify underlying patterns or structures.

Factor analysis can be conducted using various statistical software packages, and BioStat Prime utilized R packages to conduct factor analysis. BioStat Prime brings forth 2 ways of factor analysis, viz.

1. Factor

2. Principal Component Analysis.

Principal Component Analysis (PCA) and Factor Analysis (FA) are both techniques used in multivariate analysis to uncover patterns and relationships in high-dimensional data. However, they serve different purposes, and it's important to distinguish between them.

- ⚠ • PCA can be viewed as a special case of factor analysis where all the variance in the data is treated as common (shared) variance.

- ⚠ • Factor Analysis is more focused on capturing shared variance due to latent factors and specific (unique) variance associated with each variable.

- ⚠ • In PCA, the principal components are linear combinations of the original variables and are not interpreted in terms of underlying constructs or factors.

While PCA and Factor Analysis share similarities, their primary objectives differ. PCA is primarily a variance-driven technique for dimensionality reduction, while Factor Analysis is a model-based technique for understanding the underlying structure of the data in

terms of latent factors. The choice between them depends on the research question and the nature of the data.

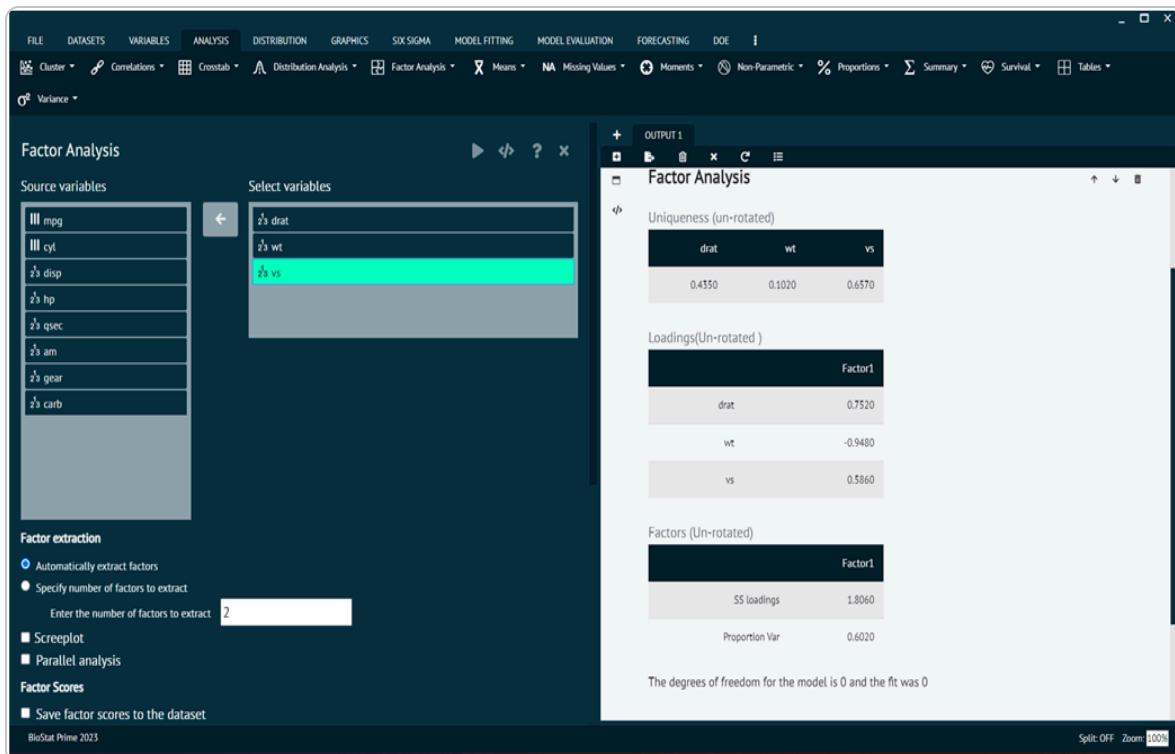
Factor Analysis

Perform maximum-likelihood factor analysis on a covariance matrix or data matrix and generates a screeplot.

To analyse Factor Analysis in BioStat Prime user must follow the steps as given.

Steps

Load the dataset --> Click on the analysis tab in main menu --> Select factor analysis tab --> Select Factor --> Once the dialog appears choose the items to be included --> Specify no. of factors to be extracted, user can also save factors and take a scree plot --> Execute the dialog.



alt text

For further information the user can explore model tuning and model evaluation options for the same.

The following are the different type argument:

vars

One or more numeric variables to extract factors from.

autoextraction

Automatically determine the number factors or extract specific numbers of factors.

screeplot

If TRUE generates a screeplot.

rotation

determine the type of rotation and takes one of the values (none, quartimax, geominT, varimax, oblimin, simplimax, promax, geominQ and bentlerQ)

saveScores

saves the factor scores in the dataset

dataset

The dataset from which the 'vars' have been picked.

Principal Component Analysis

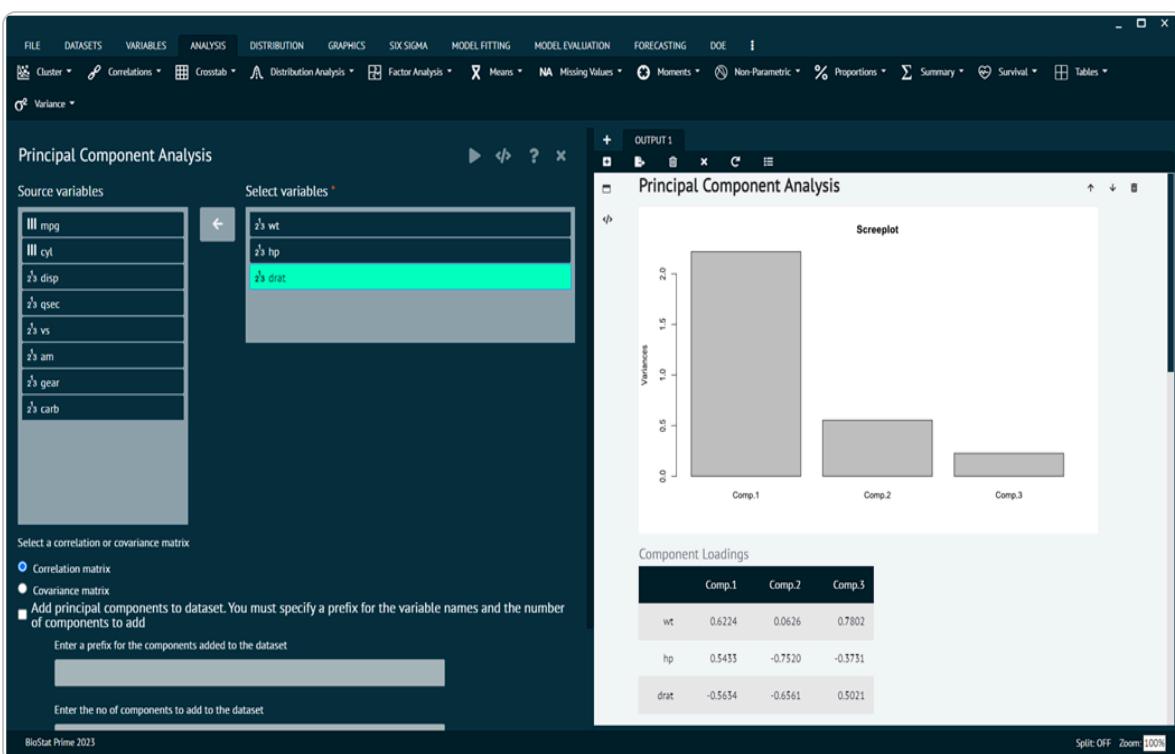
Performs a principal components analysis on the given numeric data matrix and returns the results as an object of class princomp.

To analyse Principal Component Analysis in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select factor analysis tab -> Select Principal Component Analysis -> Once the dialog appears choose the items to be included -> Specify no. of factors to be extracted, user can also save factors and take a scree plot -> Execute the dialog.

For further information the user can explore model tuning and model evaluation options for the same.



alt text

The following are the different type argument:

vars

The variables in a character vector to extract components from

cor

A boolean that specifies whether the calculation should use a correlation or covariance matrix

componentsToRetain

A numeric that Specifies the number of components to retain in the dataset. A new variable is created in the dataset for each component invoked

generateScreeplot

Generates a screeplot

prefixForComponents

Prefix to use when saving the components to a dataset

dataset

The name of the dataset as a string

Means

This section of analysis tab comes up with ways of performing the **analysis of Covariance (ANCOVA)**, **analysis of variance (ANOVA)** and **T-tests**. Each sub function of the Means tab is discussed in detail in up-coming section.

ANCOVA

ANCOVA

ANCOVA stands for **Analysis of Covariance**. It is a statistical technique that combines the principles of **analysis of variance (ANOVA)** with **regression**.

ANCOVA is used to compare group means while statistically controlling for the effects of other continuous variables that are not of primary interest, referred to as covariates.

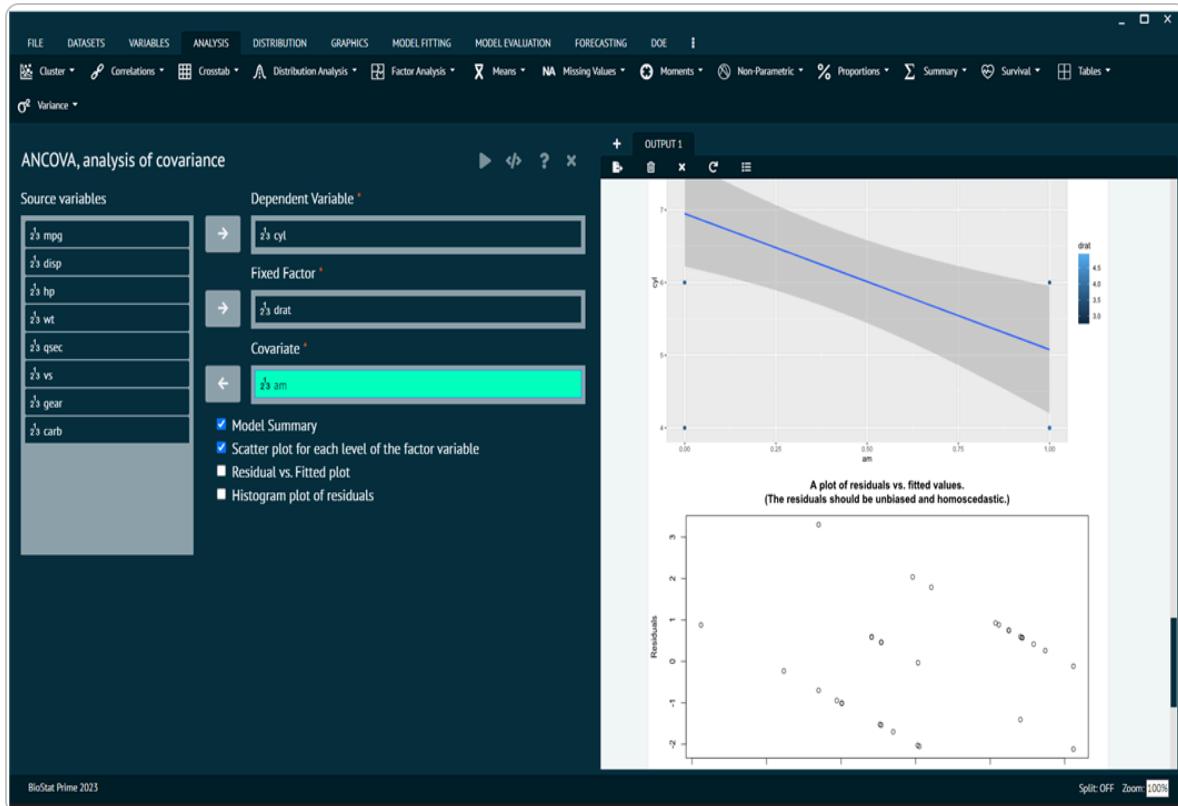
A The main idea behind ANCOVA is to remove the variance associated with the covariates from the dependent variable, allowing for a more accurate assessment of the group differences in the variable of interest.

i This is particularly useful when there is reason to believe that the covariates are related to the dependent variable, and you want to account for this relationship in the analysis.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select means -> The means tab leads to the ANCOVA analysis technique in the dialog -> In the dialog select the variable and options according to the requirement -> Execute the dialog.



alt text

The output of the analysis is shown in the output window. The user can also opt for Model Summary, Scatter plot for each level of the factor variable, Residual vs. Fitted plot, Histogram plot of residuals visualise the output as a plot.

Analysis of covariance (ANCOVA) combines features of both ANOVA and regression. It augments the ANOVA model with one or more additional quantitative variables, called **covariates**, which are related to the response variable. The covariates are included to reduce the variance in the error terms and provide more precise measurement of the treatment effects.



ANCOVA is used to test the main and interaction effects of the factors, while controlling for the effects of the covariate.

BioStat Prime first generate an Anova table with the interaction term. The goal is to examine whether the interaction term is not significant i.e. the slopes of the dependent variable against the covariate for each level of the fixed factor is not different. BioStat Prime uses the Anova package in the car package to generate this Anova table.

BioStat Prime then regenerate the Anova table controlling for the interaction term to determine whether the intercepts of the dependent variable against the covariate for each level of the fixed factor are different.

BioStat Prime provides the option to generating a scatter plot for of dependent variable against the covariate variable for each level of the fixed factor.

BioStat Prime provides the option to plot the residuals vs. fitted plot For the model where we have controlled the interaction term. The residuals should be unbiased and homoscedastic.

BioStat Prime provides the option to generate a histogram for the residuals for model where we have controlled the interaction term. (Distribution should be approx normal).

BioStat Prime gives user the option to summarize the model

Arguments

formula

an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.

data

an optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the variables in the model. If not found in `data`, the variables are taken from `environment(formula)`, typically the environment from which `lm` is called.

ANOVA, 1 and 2 way

ANOVA, or Analysis of Variance, is a statistical method used to analyze the differences among group means in a sample.

There are two main types of ANOVA: **one-way ANOVA** and **two-way ANOVA**.

One-Way ANOVA is used when there is one independent variable (factor) with more than two levels (groups).

Two-Way ANOVA is an extension of One-Way ANOVA and is used when there are two independent variables (factors).

i The aov function in R is commonly used for performing ANOVA.

To analyse it in BioStat Prime user must follow the steps as given.

Steps Load the dataset -> Click on the analysis tab in main menu -> Select means -> The means tab leads to the ANOVA,1 and 2 way analysis technique in the dialog -> In the dialog select the variable and options according to the requirement -> Execute the dialog.

mpg	n	mean	median	min	max	sd	variance
10.4000	2	2.9650	2.9650	2.9300	3.0000	0.0495	0.0024
13.3000	1	3.7300	3.7300	3.7300	3.7300	NA	NA
14.3000	1	3.2100	3.2100	3.2100	3.2100	NA	NA
14.7000	1	3.2300	3.2300	3.2300	3.2300	NA	NA
15.0000	1	3.5400	3.5400	3.5400	3.5400	NA	NA
15.2000	2	3.1100	3.1100	3.0700	3.1500	0.0566	0.0032
15.5000	1	2.7600	2.7600	2.7600	2.7600	NA	NA
15.8000	1	4.2200	4.2200	4.2200	4.2200	NA	NA
16.4000	1	3.0700	3.0700	3.0700	3.0700	NA	NA
17.0000	1	3.0700	3.0700	3.0700	3.0700	NA	NA
17.8000	1	3.9200	3.9200	3.9200	3.9200	NA	NA
18.1000	1	2.7600	2.7600	2.7600	2.7600	NA	NA
18.7000	1	3.1500	3.1500	3.1500	3.1500	NA	NA

alt text

This function fits an analysis of variance model along with data summaries, displays type I,II,III sum of squares, displays marginal means and contrasts (using marginal means). Model is built with and without interaction effects.

- Optionally performs Levene's test for homogeneity of variance across groups and plots graphs.

ANOVA, one-way with random blocks

In analysis of variance (ANOVA), the one-way ANOVA with random blocks is a variation of the traditional one-way ANOVA that incorporates the concept of random blocks. This design is often used when there is a potential source of variability in the experiment that is not of primary interest but needs to be controlled for.

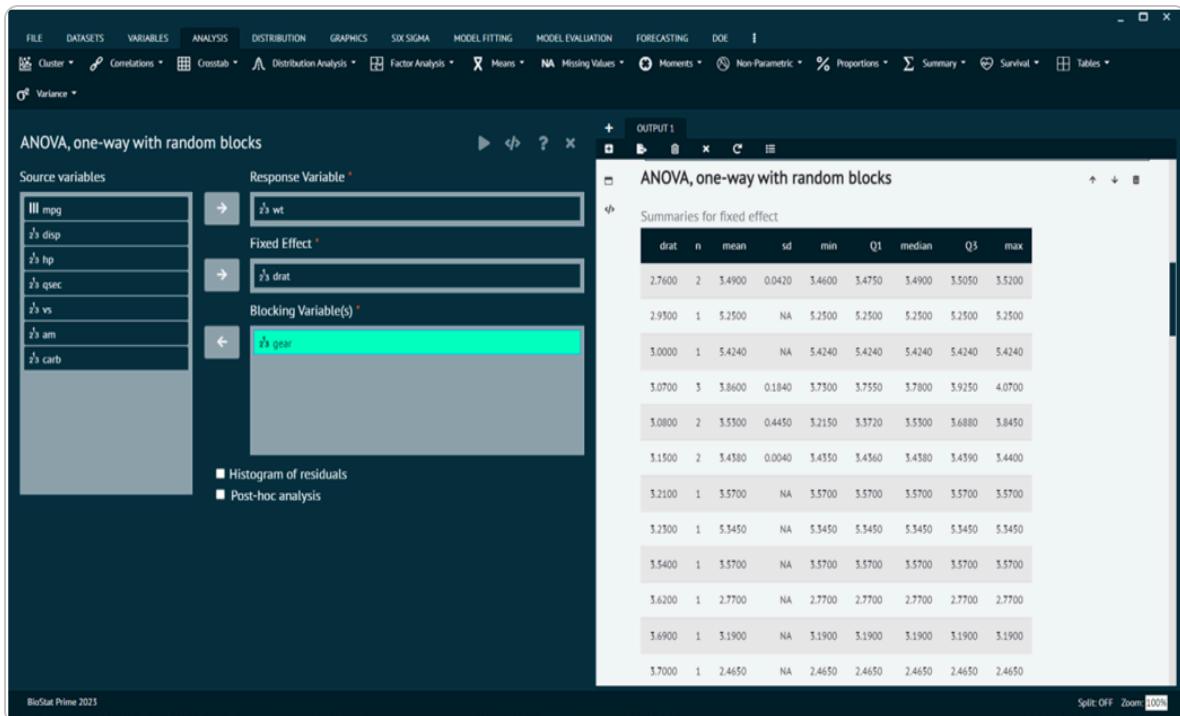
- ⚠** In the context of ANOVA, blocks refer to groups or conditions that are not of primary interest but introduce variability. These blocks are considered random because their levels are randomly selected from a larger population. The inclusion of random blocks helps to control for the potential impact of these extraneous factors.

Fits a linear mixed-effects model (LMM) to data, via REML or maximum likelihood

To analyse it in BioStat Prime user must follow the steps as given.

Steps

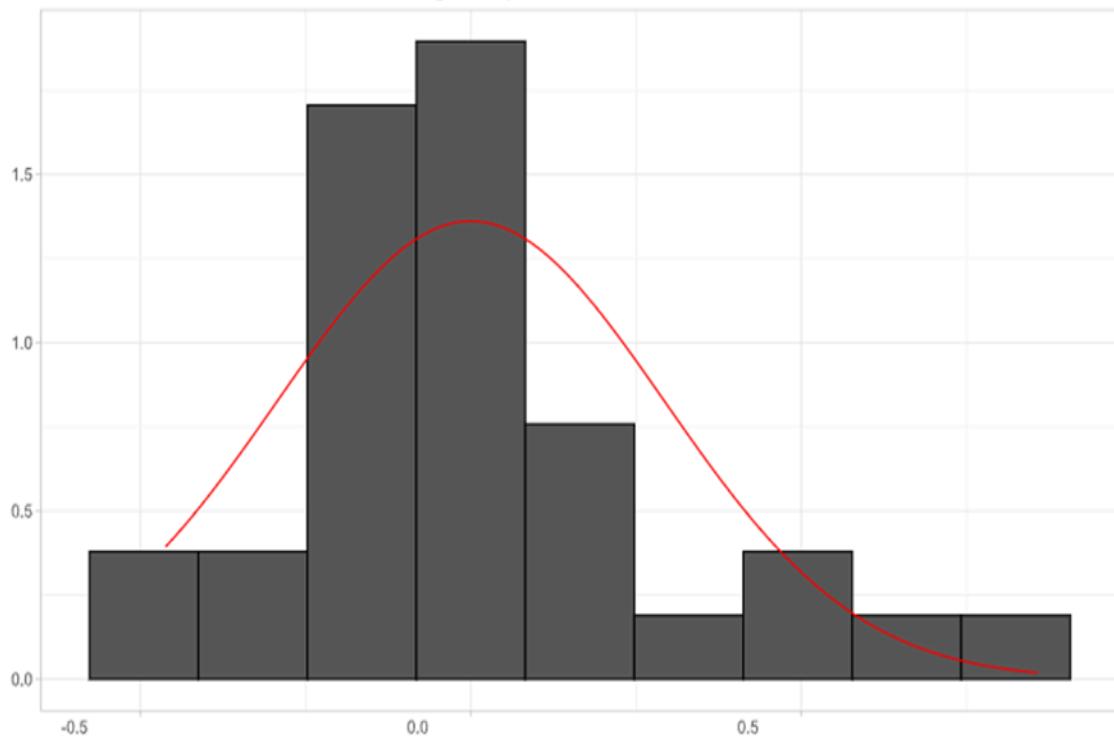
Load the dataset -> Click on the analysis tab in main menu -> Select means -> The means tab leads to the ANOVA, one-way with random blocks analysis technique in the dialog -> In the dialog select the variable and options according to the requirement -> Execute the dialog.

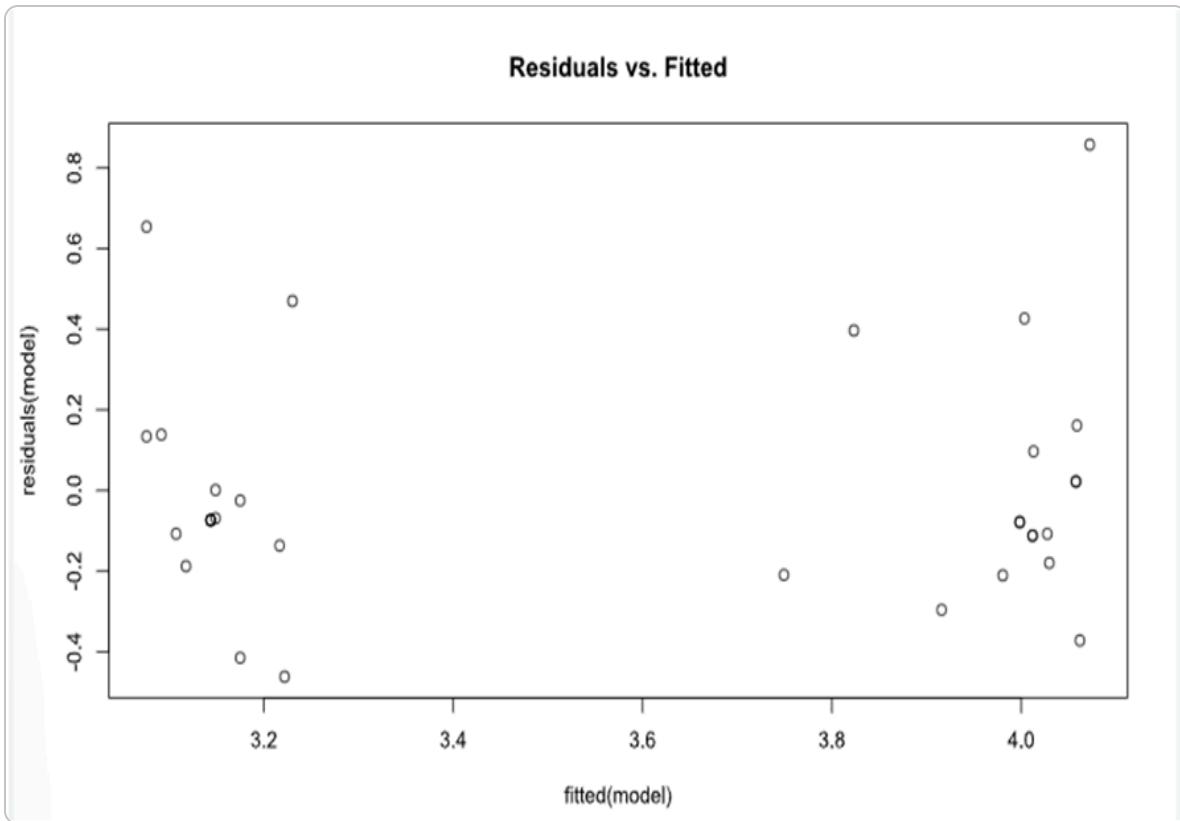


alt text

The output of the analysis is shown in the output window. The user can also opt for Histogram of residuals, Post-hoc analysis.

Histogram plot of model residuals





alt text

Arguments

formula

a two-sided linear formula object describing both the fixed-effects and random-effects part of the model, with the response on the left of a `~` operator and the terms, separated by `+` operators, on the right. Random-effects terms are distinguished by vertical bars (`|`) separating expressions for design matrices from grouping factors. Two vertical bars (`||`) can be used to specify multiple uncorrelated random effects for the same grouping variable. (Because of the way it is implemented, the `||`-syntax works only for design matrices containing numeric (continuous) predictors; to fit models with independent categorical effects, see `dummy` or the `lmer_alt` function from the `afex` package.)

data

an optional data frame containing the variables named in formula. By default the variables are taken from the environment from which lmer is called. While data is optional, the package authors strongly recommend its use, especially when later applying methods such as update and drop1 to the fitted model (such methods are not guaranteed to work properly if data is omitted). If data is omitted, variables will be taken from the environment of formula (if specified as a formula) or from the parent frame (if specified as a character vector).

REML

logical scalar - Should the estimates be chosen to optimize the REML criterion (as opposed to the log-likelihood)? na.action: a function that indicates what should happen when the data contain NAs. The default action (na.omit, inherited from the 'factory fresh' value ofgetOption("na.action")) strips any observations with any missing values in any variables.

ANOVA, one way with blocks

The "one-way" ANOVA refers to a scenario where there is **one independent variable (factor)** that categorizes the data into different groups, and you are interested in comparing the means of these groups to determine if there are any statistically significant differences.

The term "with blocks" in ANOVA typically refers to a **design that includes the concept of blocking**. Blocking is used when there are known sources of variability that are not of primary interest but should be taken into account to increase the precision of the experiment.



Blocks are used to create more homogeneous groups within which the experimental units are similar.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select means -> The means tab leads to the ANOVA, one way with blocks analysis technique in the dialog -> In the dialog select the variable and options according to the requirement -> Execute the dialog.

The screenshot shows the BioStat Prime 2023 software interface. The top menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION, FORECASTING, DOE, and various statistical tools like Cluster, Correlations, Crosstab, Distribution Analysis, Factor Analysis, Means, Missing Values, Moments, Non-Parametric, Proportions, Summary, Survival, and Tables. A dropdown menu for 'Variance' is open.

The main window title is "ANOVA, one way with blocks". On the left, under "Source variables", there is a list of car attributes: disp, hp, drat, wt, vs, am, gear, carb. On the right, under "Response variable (one)", "Fixed effect", and "Blocking variable(s)", the variable "cyl" is selected. Below these fields are two buttons: "Histogram of residuals" and "Post-hoc analysis".

The "OUTPUT1" tab is active, displaying R code and its output. The R code includes loading the FSA package, running fisherR(), and attaching the package. It also notes that the 'bootCase' object is masked from the 'car' package. The output shows summaries for fixed effects and blocking variables, including tables for cyl and mpg.

cyl	n	mean	sd	min	Q1	median	Q3	max
4	11	19.1570	1.6820	16.7000	18.5600	18.9000	19.9500	22.9000
6	7	17.9770	1.7070	15.5000	16.7400	18.3000	19.1700	20.2200
8	14	16.7720	1.1960	14.5000	16.0980	17.1750	17.5550	18.0000

cyl	mpg	n	mean	sd	min	Q1	median	Q3	max
8	10.4000	2	17.9000	0.1130	17.8200	17.8600	17.9000	17.9400	17.9800
8	15.3000	1	15.4100	NA	15.4100	15.4100	15.4100	15.4100	15.4100

alt text

In the context of a one-way ANOVA with blocks, user would have one main factor (e.g., a treatment or condition), and the blocks would be another variable that is not the primary focus of user's study but is thought to contribute to variability.

The idea is to account for the variability due to the blocks so that user can better detect differences related to the main factor.

ANOVA, N way

Fits an analysis of variance model, displays type I,II,III sum of squares, displays marginal means and contrasts (using marginal means).

- ⚠ Optionally performs Levene's test for homogeneity of variance across groups and plots graphs.

- ℹ Levene's test is run for all the main effects

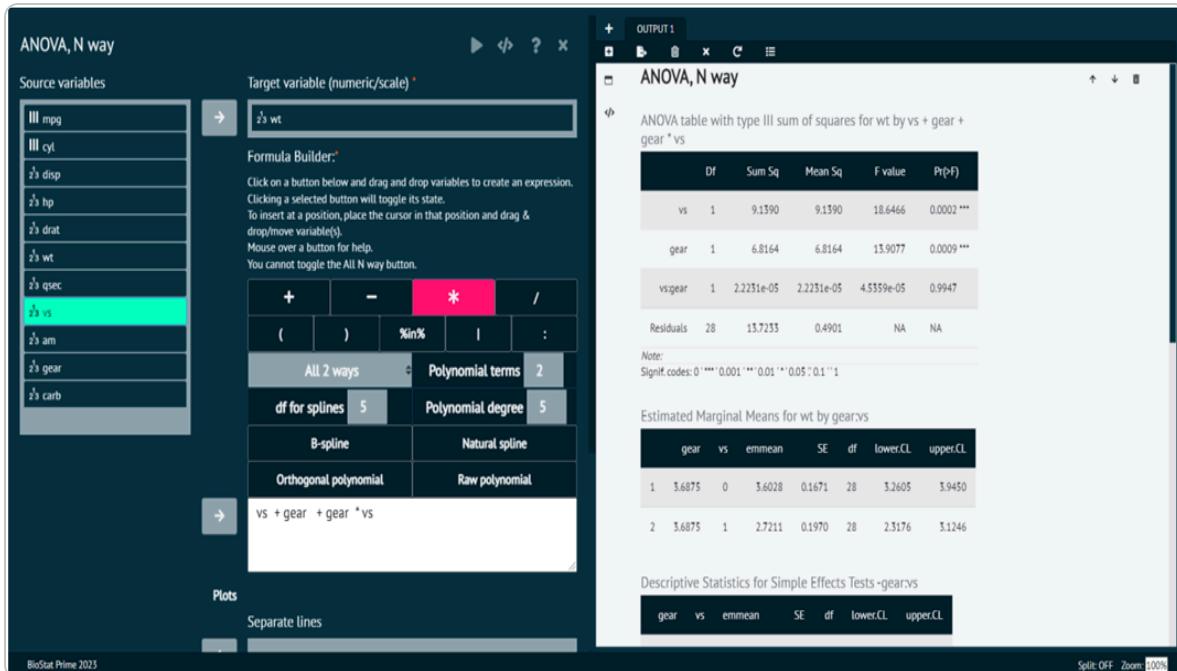
In statistics, an N-way ANOVA (Analysis of Variance) with blocks refers to a statistical analysis that involves multiple independent variables or factors (more than two). The term "blocks" in this context refers to a **way of handling potential sources of variability** that are not the primary focus of the study but need to be accounted for to improve the precision of the analysis.

N-way ANOVA indicates that there are multiple independent variables (factors). For example, in a 2-way ANOVA, there are two independent variables, and in an N-way ANOVA, there are more than two. Each independent variable can have multiple levels or categories.

- ⚠ N-way ANOVA with blocks involves analyzing the effects of multiple independent variables on a dependent variable while taking into account the potential impact of blocking variables.

To analyse it in BioStat Prime user must follow the steps as given. Steps

Load the dataset -> Click on the analysis tab in main menu -> Select means ☐The means tab leads to the ANOVA, N way analysis technique in the dialog -> In the dialog select the target variable and create a formula according to the requirement -> Execute the dialog.



alt text

⚠ NOTE:

1. To get all marginal means and post-hocs user needs to construct a formula with the main effects and all the interaction terms in the model. So if you are attempting to analyze a 3 way interaction, user needs to specify
A + B + C + A:B + B:C + A:C + A:B:C
If instead you specify ABC, user will get the complete ANOVA table, user will NOT get the estimated marginal means and post-hocs for all the interactions.
2. Estimated marginal means AND POST-HOCS are computed for all main effects and the SPECIFIED INTERACTIONS

The user can build a formula in the formula builder by following the steps given below.

⚠

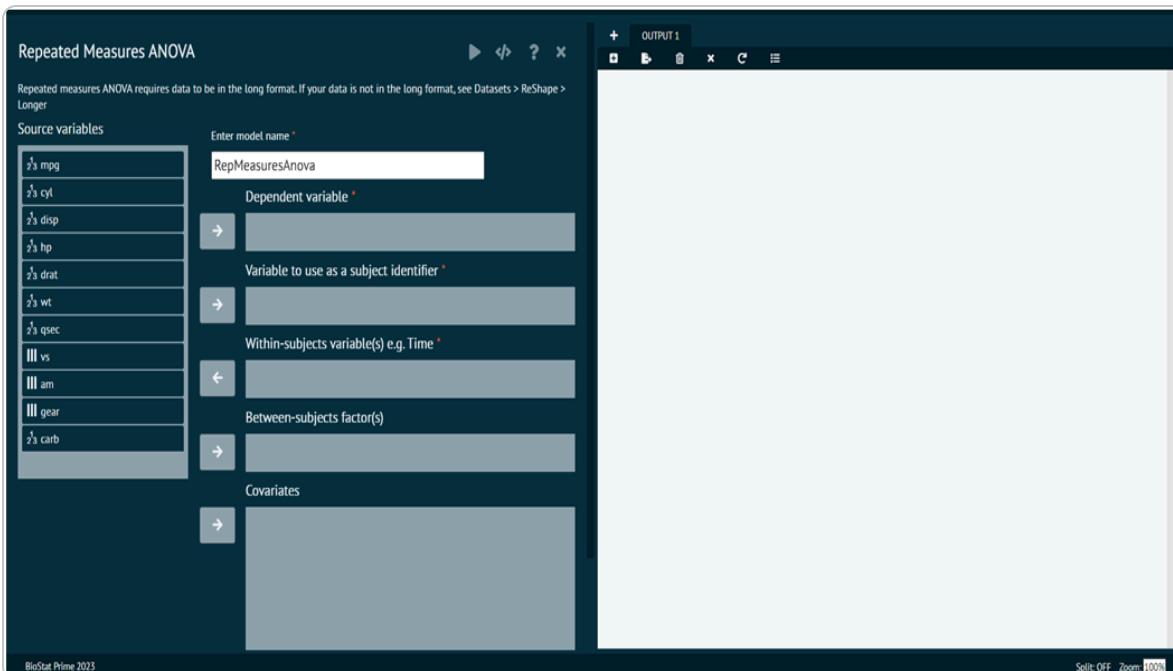
1. Click on a button in formula builder and drag and drop variables to create an expression.

2. Clicking a selected button will toggle its state.
3. To insert at a position, place the cursor in that position and drag & drop/move variable(s).
4. Mouse over a button for help.
5. User cannot toggle the All N way button.

ANOVA Repeated Measures, Long

With repeated measures ANOVA F statistics are computed for each within subjects factor, between subject factor and the interaction term for mixed ANOVA

- Info BioStat Prime currently support a single within subject and between subject factor, the between subject factor is optional.



alt text

⚠ Arguments

data

A data.frame containing the data. Mandatory

dv

character vector (of length 1) indicating the column containing the dependent variable in data.

between

character vector indicating the between-subject(s) factor(s)/column(s) in data. Default is NULL indicating no between-subjects factors.

within

character vector indicating the within-subject(s)(or repeated-measures) factor(s)/column(s) in data. Default is NULL indicating no within-subjects factors.

covariate

character vector indicating the between-subject(s) covariate(s) (i.e., column(s)) in data. Default is NULL indicating no covariates.

- Please note that factorize needs to be set to FALSE in case the covariate is numeric and should be treated as such.

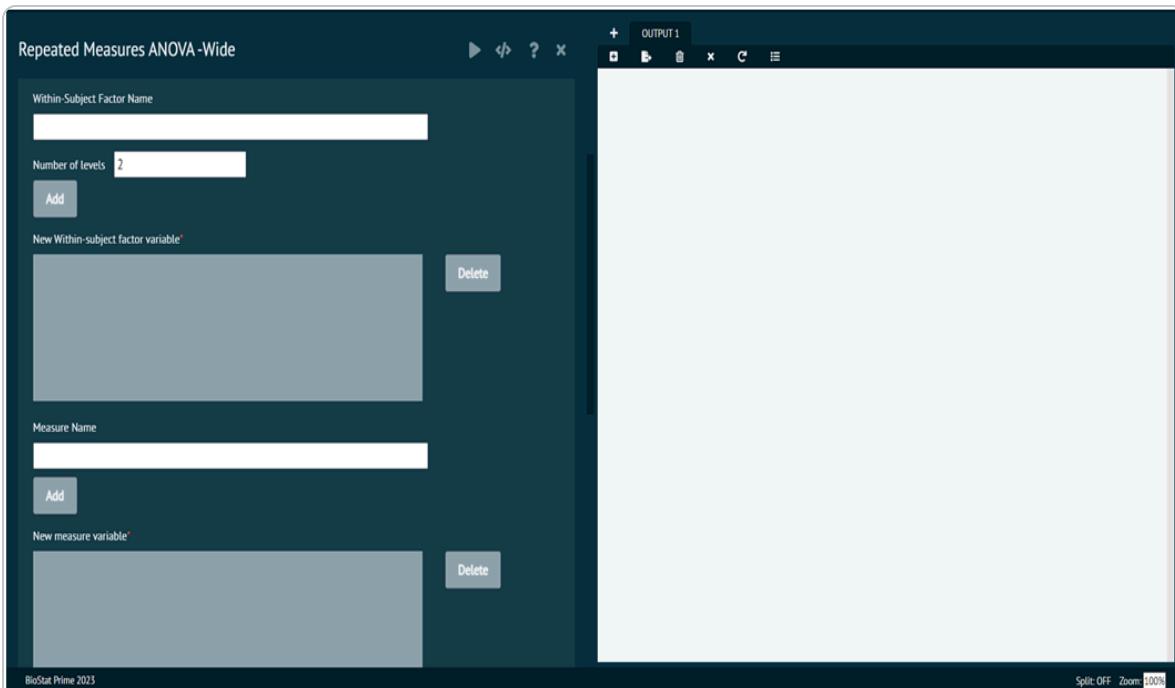
anovatable

list of further arguments passed to function producing the ANOVA table.

ANOVA Repeated Measures, Wide

With repeated measures ANOVA F statistics are computed for each within subjects factor, between subject factor and the interaction term for mixed ANOVA

- Info icon: BioStat Prime currently support a single within subject and between subject factor, the between subject factor is optional.



alt text

⚠ NOTE:

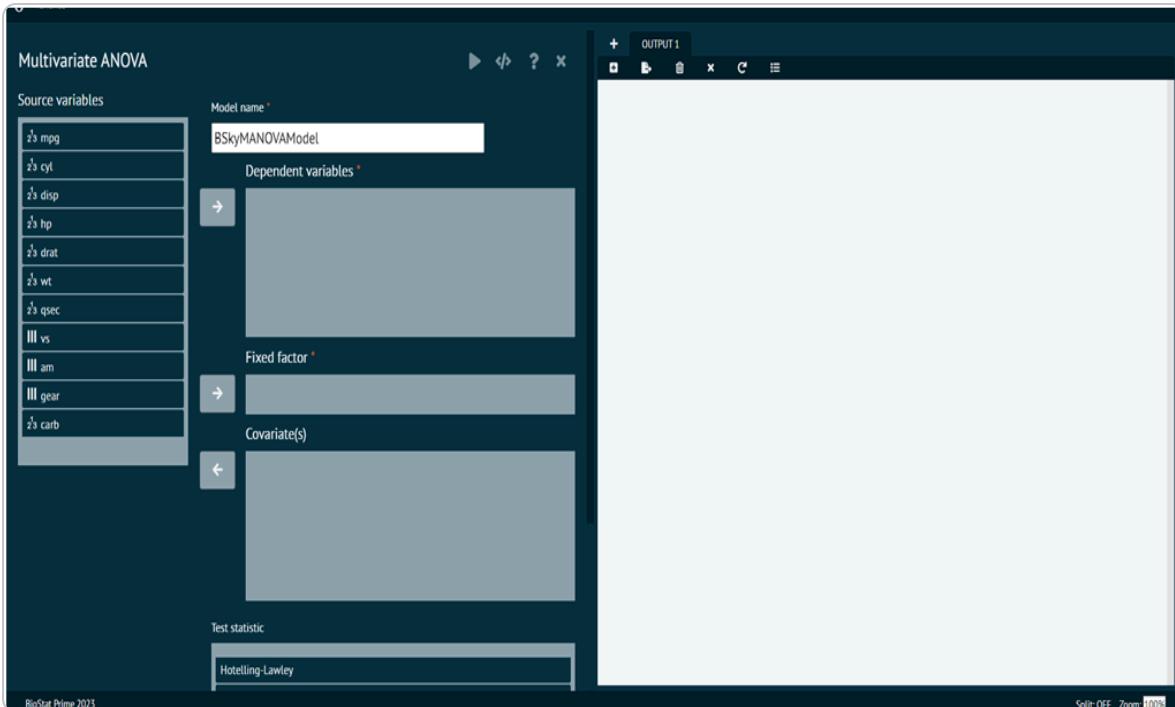
1. BioStat Prime needs to reshape the data when running a repeated measures ANOVA on a wide dataset
2. BioStat Prime supports multiple repeated measures for a single variable e.g. Blood Sugar measured at pretest, posttest and at a followup visit
3. User needs to specify a repeated factor name e.g. Blood Sugar and the number of levels. BioStat Prime will create a factor variable e.g. named Blood Sugar with levels created from the names of the variables containing the

repeated measures e.g. the levels of the factor will be pretest, posttest and followup

4. User needs to specify a measure name e.g. Value. BioStat Prime will create a variable e.g. Value with all the Blood Sugar values corresponding to the pretest, posttest and followup for each subject.
5. BioStat Prime supports a single between-subject and within-subject factor variable.
6. Future versions will support multiple measures as well as multiple between subject and within subject factor variables.
7. By default each row of the dataset corresponds to a unique subject, user can also specify a variable for the subject ID.

MANOVA

Class "manova" differs from class "aov" in selecting a different summary method. Function **manova** calls aov and then add class "manova" to the result object for each stratum.



alt text

Multivariate ANOVA

Omnibus multivariate tests and corresponding F and p values are provided. Follow up univariate tests are also provided.



NOTE: BioStat Prime currently support a single independent factor

- NOTE: BioStat Prime don't display the confidence interval in the plot of means as there are unnecessary warnings displayed. We are working with the author of the gplots package to rectify this.

t-test, Independent

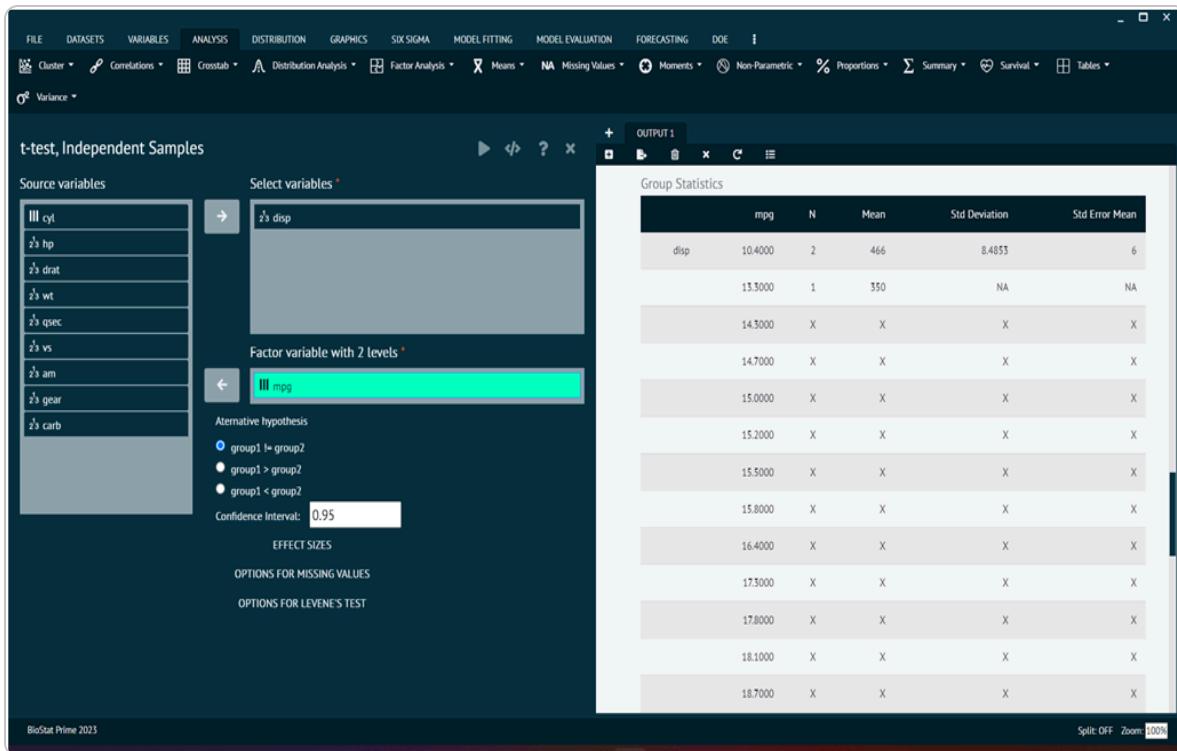
An independent samples t-test is a statistical test used to compare the means of two independent groups to determine if there is a significant difference between them. It's commonly employed when user has two separate groups of observations, and user wants to assess whether the means of these groups are statistically different from each other.

Performs a one sample t-tests against the two groups formed by a factor variable (with two levels). Displays results for equal variances **TRUE** and **FALSE**. For equal variances the pooled variance is used otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used. Internally calls `t.test` in the `stats` package for every selected variable.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select means -> The means tab leads to the t-test, Independent analysis technique in the dialog -> In the dialog select the variable and options according to the requirement -> Execute the dialog.



alt text

⚠ Arguments

varNamesOrVarGlobalIndices

selected scale variables (say var1, var2)

group

a factor variable with two levels (say var3)

conf.level

a numeric value (say 0.95) .

missing

missing values are handled on a per variable basis (missing =0) or list wise across all variables (missing=1).

datasetNameOrDatasetGlobalIndex

Name of the dataset (say Dataset) from which var1, var2 and var3 are selected.

alternative

a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.

t-test, One Sample

A one-sample t-test is a statistical test used to determine if the mean of a single sample is significantly different from a known or hypothesized population mean. It's commonly used when you have a sample of data and want to assess whether the sample mean is consistent with a specific population value.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select means -> The means tab leads to the t-test, one sample analysis technique in the dialog -> In the dialog select the variable and options according to the requirement -> Execute the dialog.

The screenshot shows the BioStat Prime software interface. The main window displays the 't-test, One Sample' dialog. On the left, under 'Source variables', the 'mpg' variable is selected. In the center, a 'Select variables' dropdown shows 'qsec' highlighted. Below this, the 'Alternative hypothesis' section is set to 'Population mean = mu'. The 'Test value (mu)' field contains '0'. The 'Confidence interval:' field contains '0.95'. At the bottom, sections for 'EFFECT SIZES' and 'OPTIONS FOR MISSING VALUES' are visible. To the right, the 'OUTPUT1' tab is active, showing the results of the t-test. The output includes a note about independent sample T test on 'qsec', a table of 'One Sample Statistics' with 'N' as 32, 'Mean' as 17.8487, 'Std Deviation' as 1.7869, and 'Std Error Mean' as 0.3159. The 'One Sample t-test' table shows 'Test Value = 0', 'confidence: 0.95', 'df' as 31, 'Sig.(2-tail)' as 7.7905e-33, 'mean difference' as 17.8487, 'lower' as 17.2045, and 'upper' as 18.4950. A note at the bottom states 'Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ''1''. The bottom status bar indicates 'BioStat Prime 2023' and 'Split: OFF Zoom: 100%'. The overall background is dark blue.

alt text

t-test, Paired Samples

A paired samples t-test (also known as a dependent samples t-test or a matched-pairs t-test) is a statistical test used to determine if there is a significant difference between the means of two related groups.

- ⚠** The key characteristic of this test is that it is applied to paired observations, where each observation in one group is directly related to an observation in the other group.

Performs one sample t-tests on selected variables. Optionally computes effect size indices for standardized differences: Cohen's d and Hedges' g (This function returns the population estimate.)

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select means -> The means tab leads to the t-test, paired samples analysis technique in the dialog -> In the dialog select the variable and options according to the requirement -> Execute the dialog.

t-test, Paired Samples

Source variables

- mpg
- cyl
- disp
- hp
- drat
- qsec
- am
- gear
- carb

First numeric variable: `vs`

Second numeric variable: `wt`

Alternative hypothesis:

- Difference != mu
- Difference > mu
- Difference < mu
- Assume equal variance

Null hypothesis (mu): 0

Confidence level: 0.95

EFFECT SIZES

OUTPUT 1

t-test, Paired Samples

Summary Statistics

vars	n	mean	sd	median	trimmed	mad	min	max	range	skew
vs	32	0.4375	0.5040	0.0000	0.4231	0.0000	1.0000	1.0000	0.2403	
wt	2	3.2172	0.9785	3.3250	3.1527	0.7672	1.5130	5.4240	3.9110	0.4231

Paired t-test

Null Value Considered: 0				
		sample estimate	confidence: 0.95	confidence: 0.95
t	df	p-value	mean of the differences	lower
-11.8572	31	4.7267e-13 ***	-2.7798	-3.2579
				-2.3016

Note:
Signif. codes: 0 '***' 0.001 '** 0.01 '*' 0.05 '.' 0.1 ' ' 1

Additional Details

Additional Comments	
Test Method Performed	Paired t-test
Alternative	two.sided

BioStat Prime 2023

Split OFF Zoom 100%

alt text

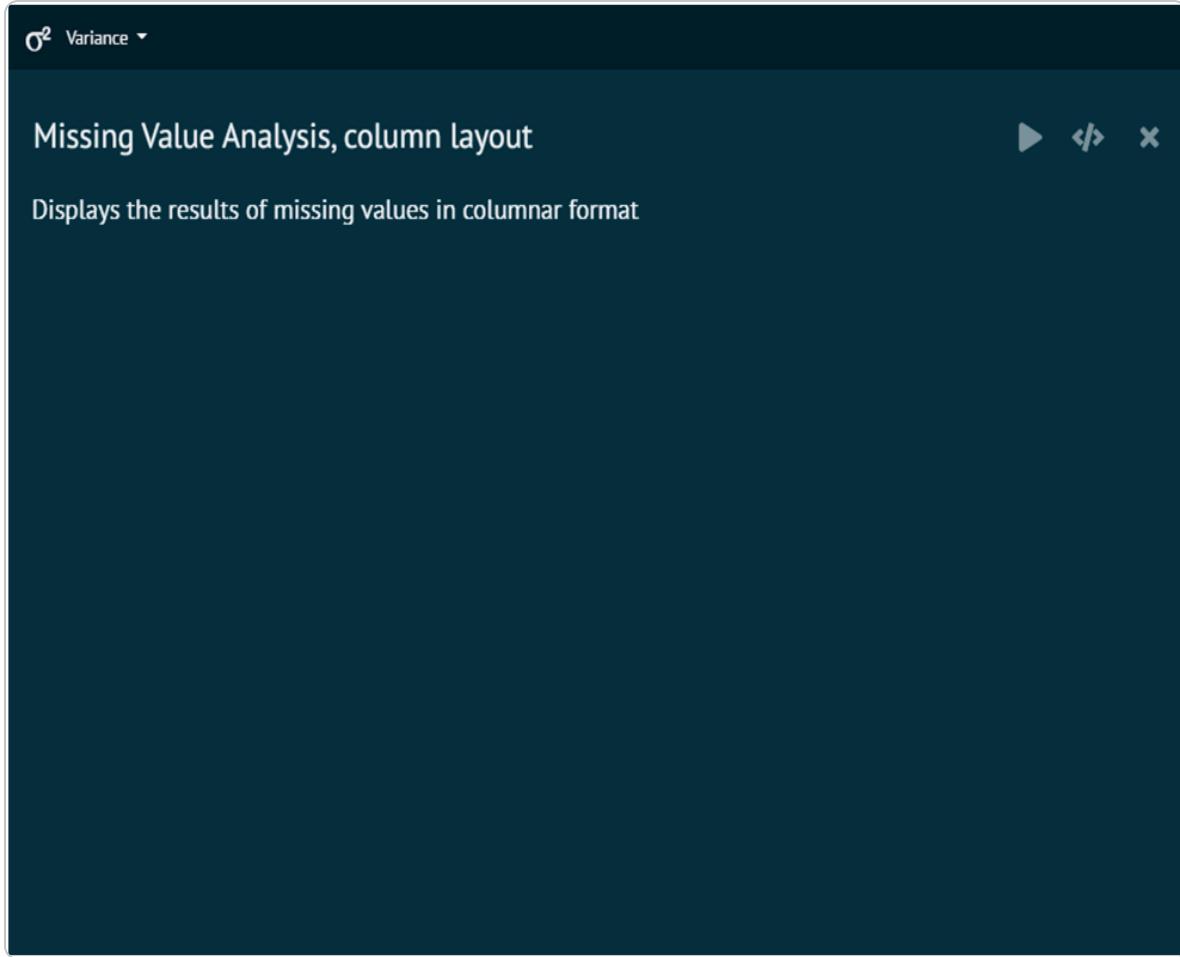
Missing Value

Column layout

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select missing values -> The missing values tab leads to column layout in the dialog -> Execute the dialog.



alt text

Row layout

Missing value analysis is an essential step in data preprocessing, helping you understand and handle missing data in your dataset. The "row layout" in this context suggests that user is examining missing values on a row-wise basis, looking at how missing values are distributed across individual rows in user's dataset.

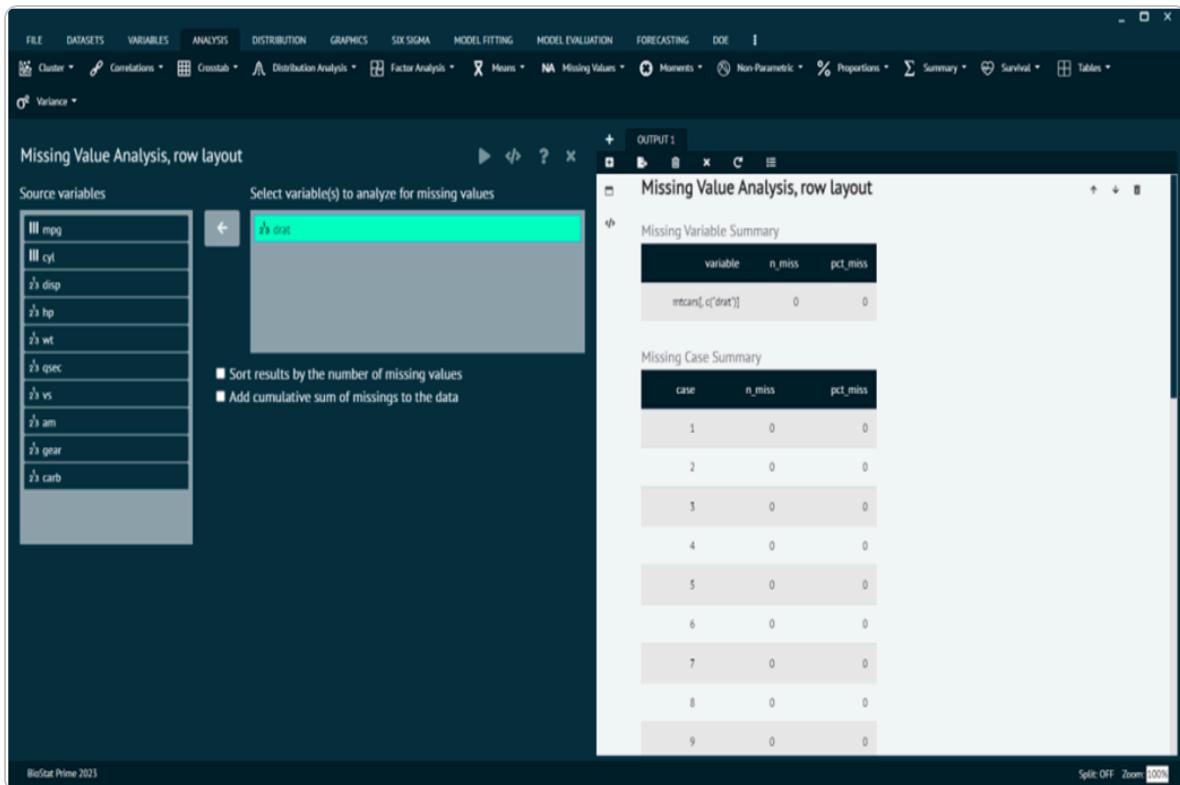
Analyzes missing values and displays results in rows, displays summary information of missing values at the variable level and lists the number of missing values on each row for variables being analyzed.

Provides a summary for each variable of the number, percent missings, and cumulative sum of missings of the order of the variables. By default, it orders by the most missings in each variable.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select missing values -> The missing values tab leads to row layout in the dialog -> In the dialog select the variable and options according to the requirement -> Execute the dialog.



alt text

⚠ Arguments

data

a dataframe

order

a logical indicating whether to order the result by n_miss. Defaults to TRUE. If FALSE, order of variables is the order input.

add_cumsum

logical indicating whether or not to add the cumulative sum of missings to the data. This can be useful when exploring patterns of nonresponse. These are calculated as

the cumulative sum of the missings in the variables as they are first presented to the function.

Moments

D'Agostino skewness test

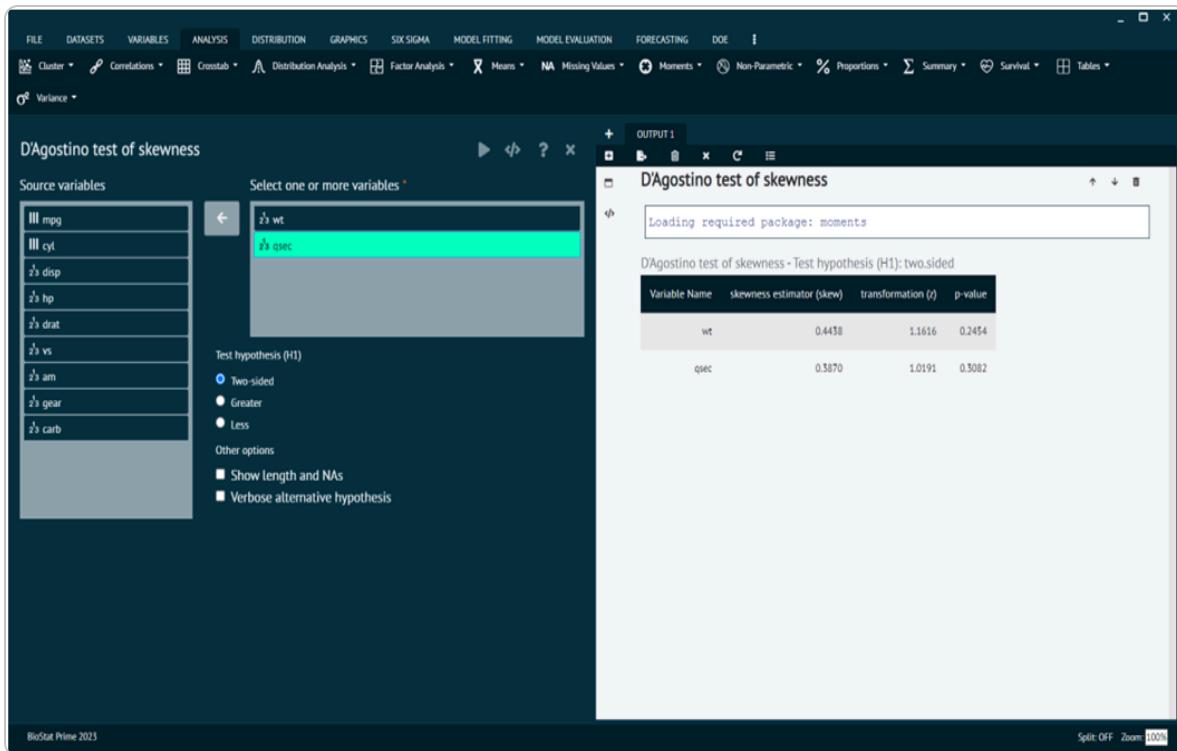
The D'Agostino skewness test is a statistical test used to assess whether the skewness of a sample differs from what would be expected in a normal distribution. Skewness measures the asymmetry of a distribution, indicating whether the data is skewed to the left or right. The D'Agostino skewness test is one of the omnibus tests for normality, alongside tests like the Shapiro-Wilk test and the Anderson-Darling test. These tests are designed to check whether a given sample comes from a normally distributed population.

Performs D'Agostino test for skewness in normally distributed data.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select Moments -> The moments tab leads to D'Agostino skewness test in the dialog -> In the dialog select the variable and options according to the requirement -> Execute the dialog.



alt text

⚠ Arguments

x

a numeric vector of data values.

y

a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". User can specify just the initial letter.

⚠ Under the hypothesis of normality, data should be symmetrical (i.e. skewness should be equal to zero).

⚠ This test has such null hypothesis and is useful to detect a significant skewness in normally distributed data.

Non-Parametric

Chi-Square test

The chi-square test is a statistical test used to determine if there is a significant association between two categorical variables. It is a non-parametric test, meaning it makes no assumptions about the distribution of the data. The test is applicable when the variables are categorical and the data can be presented in a contingency table.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select Non-Parametric -> The non-parametric tab leads to Chi-Square test -> In the dialog select the variable and options according to the requirement -> Execute the dialog.

The screenshot shows the BioStat Prime software interface. The main window title is "Chi-squared Test". On the left, a list of source variables is shown: cyl, disp, hp, drat, wt, qsec, vs, am, gear, carb. On the right, under "Selected variables", "mpg" and "cyl" are listed. Below this, a note states: "Test against equal proportions or enter proportions to test against. If your variable is gender, leave this control blank to test for equal proportions. To test for 20% females, 80% males, enter 0.2,0.8. Enter a proportion for every level. Proportions must total to 1." The output window titled "OUTPUT1" displays the results of the Chi-squared test for "cyl". The "Test Result" table shows X-squared = 3.9375, df = 24, and p-value = 1.0000. The "Additional Details" section notes the test method performed is "Chi-squared test for given probabilities". The "Frequencies for variable cyl" table shows observed values (4, 6, 8) and expected values (11, 7, 14) for categories 1, 2, and 3 respectively, along with residuals. A second "Chi-squared test for given probabilities" table is also present in the output.

alt text

Arguments

x

a numeric vector or matrix. x and y can also both be factors.

y

a numeric vector; ignored if x is a matrix. If x is a factor, y should be a factor of the same length.

correct

a logical indicating whether to apply continuity correction when computing the test statistic for 2 by 2 tables: one half is subtracted from all $|O - E|$ differences; however, the correction will not be bigger than the differences themselves. No correction is done if simulate.p.value = TRUE.

p

a vector of probabilities of the same length of x. An error is given if any entry of p is negative.

rescale.p

a logical scalar; if TRUE then p is rescaled (if necessary) to sum to 1. If rescale.p is FALSE, and p does not sum to 1, an error is given.

simulate.p.value

a logical indicating whether to compute p-values by Monte Carlo simulation.

B

an integer specifying the number of replicates used in the Monte Carlo test.

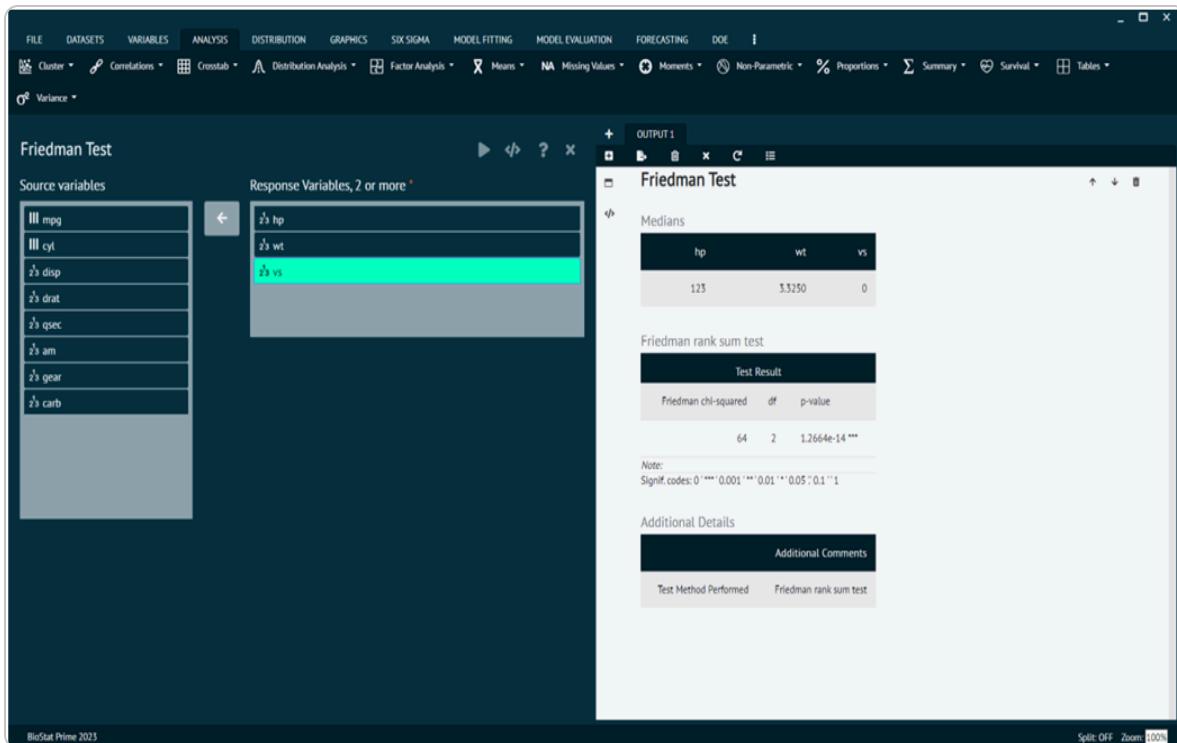
Friedman Test

The Friedman test is a non-parametric statistical test used to detect differences in treatment effects among multiple related groups. It is an extension of the Wilcoxon signed-rank test for more than two related samples. The Friedman test is particularly suitable when the data are not normally distributed or when the assumptions of a repeated measures ANOVA are not met.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select Non-Parametric -> The non-parametric tab leads to Friedman Test -> In the dialog select the variable and options according to the requirement -> Execute the dialog.



alt text

! Arguments

y

either a numeric vector of data values, or a data matrix.

groups

a vector giving the group for the corresponding elements of y if this is a vector; ignored if y is a matrix. If not a factor object, it is coerced to one.

blocks

a vector giving the block for the corresponding elements of y if this is a vector; ignored if y is a matrix. If not a factor object, it is coerced to one.

formula

a formula of the form $a \sim b | c$, where a , b and c give the data values and corresponding groups and blocks, respectively.

data

an optional matrix or data frame (or similar: see `model.frame`) containing the variables in the formula formula. By default the variables are taken from `environment(formula)`.

subset

an optional vector specifying a subset of observations to be used.

na.action

a function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

Kruskal-Wallis Rank Sum Test

The Kruskal-Wallis test is a non-parametric statistical test used to determine if there are any statistically significant differences between the medians of three or more independent groups.

To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> Click on the analysis tab in main menu -> Select Non-Parametric -> The non-parametric tab leads to Kruskal-Wallis Rank Sum Test -> In the dialog select the variable and options according to the requirement -> Execute the dialog.

Kruskal-Wallis Rank Sum Test

Source variables: cyl, disp, hp, drat, wt, qsec, vs, am, carb

Response variable: gear

Factor variable: mpg

Estimation method: Monte Carlo (10000 simulations)

Multiple comparison adjustment: holm

Options for handling ties: mid-ranks

OUTPUT 1

Kruskal-Wallis Rank Sum Test

Loading required package: coin

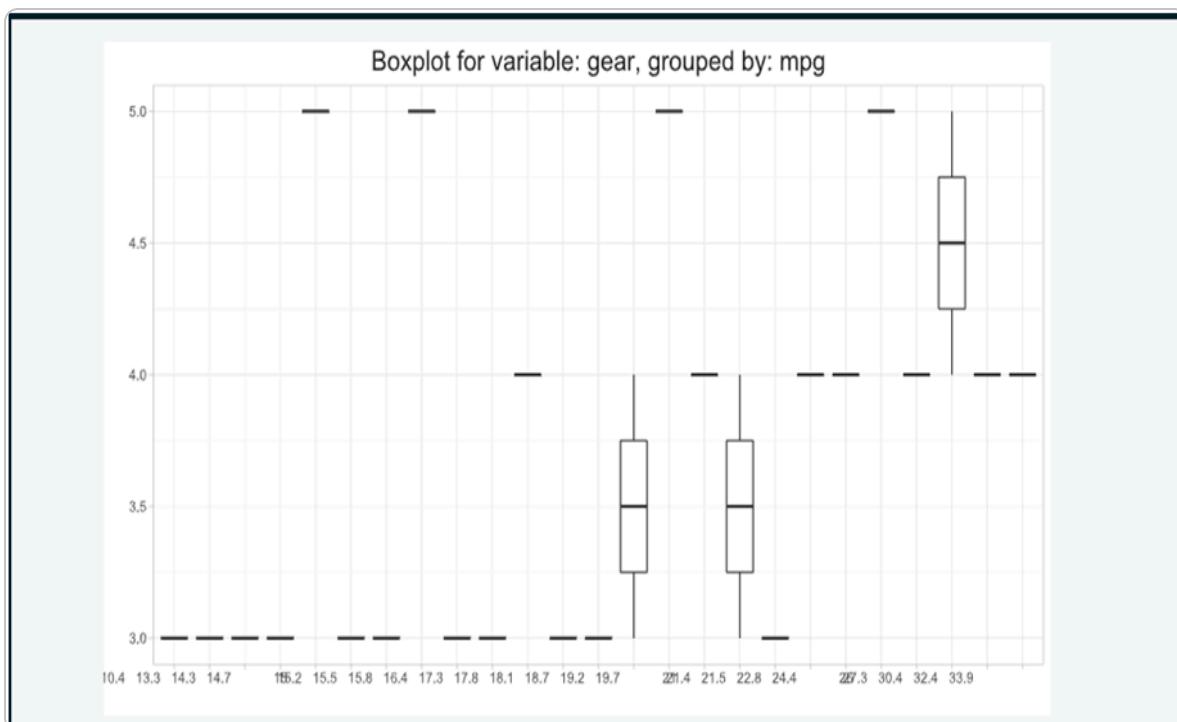
Summary Statistics

TargetVariable	mpg	count	min	Quantile_1st_25	mean	median	Quantile_3rd_75	max
gear	10.4000	2	3	3.0000	3.0000	3.0000	3.0000	3
gear	13.3000	1	3	3.0000	3.0000	3.0000	3.0000	3
gear	14.3000	1	3	3.0000	3.0000	3.0000	3.0000	3
gear	14.7000	1	3	3.0000	3.0000	3.0000	3.0000	3
gear	15.0000	1	5	5.0000	5.0000	5.0000	5.0000	5
gear	15.2000	2	3	3.0000	3.0000	3.0000	3.0000	3
gear	15.5000	1	3	3.0000	3.0000	3.0000	3.0000	3
gear	15.8000	1	5	5.0000	5.0000	5.0000	5.0000	5
gear	16.4000	1	3	3.0000	3.0000	3.0000	3.0000	3
gear	17.3000	1	3	3.0000	3.0000	3.0000	3.0000	3
gear	17.8000	1	4	4.0000	4.0000	4.0000	4.0000	4

Split OFF Zoom 100%

alt text

Box plot for variable.



alt text

Arguments

Arguments x

a numeric vector of data values, or a list of numeric data vectors. Non-numeric elements of a list will be coerced, with a warning.

g

a vector or factor object giving the group for the corresponding elements of x. Ignored with a warning if x is a list.

formula

a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.

data

an optional matrix or data frame (or similar: see `model.frame`) containing the variables in the formula formula. By default the variables are taken from `environment(formula)`.

subset

an optional vector specifying a subset of observations to be used.

na.action

a function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

Wilcoxon Test, independent samples

The Wilcoxon rank-sum test, also known as the Mann-Whitney U test, is a non-parametric statistical test used to determine whether there is a significant difference between two independent groups. It is often used when the assumptions of the t-test are not met, especially when the data are not normally distributed or when the measurement scale is ordinal.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select Non-Parametric -> The non-parametric tab leads to Wilcoxon Test, independent samples -> In the dialog select the variable and options according to the requirement -> Execute the dialog.

Wilcoxon Test, independent samples

Source variables: mpg, disp, hp, drat, wt, qsec, am, gear, carb

Response Variable (one): cyl vs

Factor (one) with only two levels: cyl

Alternative Hypothesis: Group1 < Group2 < mu

Test Method: Exact

Null hypothesis (mu): 0

Confidence interval: 0.95

Output:

Summary Statistics:

cyl	median
vs	
4	1
3	0

Wilcoxon rank sum test with continuity correction

Null Value Considered: 0				
W	p-value	difference in location	lower	upper
198.5000	0.0001***	1	0.9999	1.0000

Note: Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1

Additional Details:

Test Method Performed: Wilcoxon rank sum test with continuity correction

Alternative: two.sided

alt text

Arguments

x

numeric vector of data values. Non-finite (e.g., infinite or missing) values will be omitted.

y

an optional numeric vector of data values: as with x non-finite values will be omitted.

alternative

a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.

mu

a number specifying an optional parameter used to form the null hypothesis. See 'Details'.

paired

a logical indicating whether you want a paired test.

exact

a logical indicating whether an exact p-value should be computed.

correct

a logical indicating whether to apply continuity correction in the normal approximation for the p-value.

conf.int

a logical indicating whether a confidence interval should be computed.

conf.level

confidence level of the interval.

formula

a formula of the form `lhs ~ rhs` where `lhs` is a numeric variable giving the data values and `rhs` a factor with two levels giving the corresponding groups.

data

an optional matrix or data frame (or similar: see `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.

subset

an optional vector specifying a subset of observations to be used.

na.action

a function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

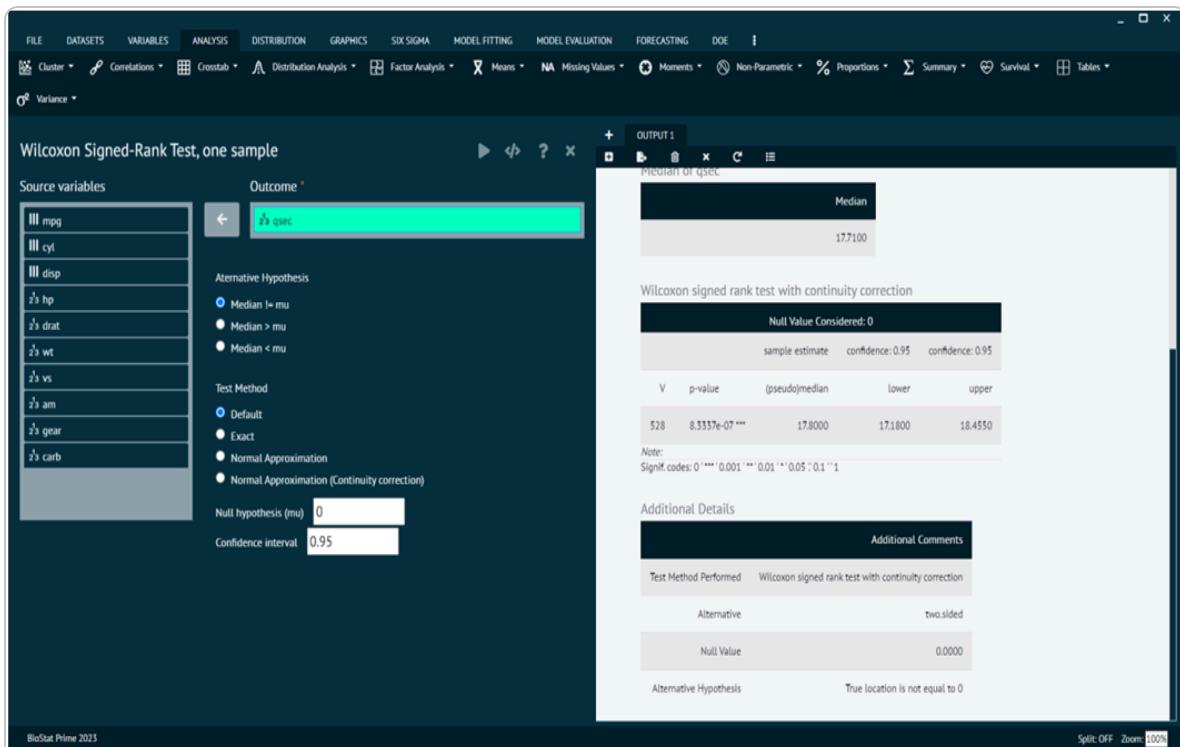
Wilcoxon Signed-Rank Test, one sample

The Wilcoxon signed-rank test is a non-parametric statistical test used to assess whether the median of a single sample is different from a specified value (often a hypothesized median). It's particularly useful when the data are not normally distributed or when the measurement scale is ordinal.

To analyse it in BioStat Prime user must follow the steps as given.

Step

Load the dataset -> Click on the analysis tab in main menu -> Select Non-Parametric -> The non-parametric tab leads to Wilcoxon Signed-Rank Test, one sample -> In the dialog select the variable and options according to the requirement -> Execute the dialog.



alt text

⚠ Arguments

X

numeric vector of data values. Non-finite (e.g., infinite or missing) values will be omitted.

y

an optional numeric vector of data values: as with x non-finite values will be omitted.

alternative

a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.

mu

a number specifying an optional parameter used to form the null hypothesis. See 'Details'.

paired

a logical indicating whether you want a paired test.

exact

a logical indicating whether an exact p-value should be computed.

correct

a logical indicating whether to apply continuity correction in the normal approximation for the p-value.

conf.int

a logical indicating whether a confidence interval should be computed.

conf.level

confidence level of the interval.

formula

a formula of the form `lhs ~ rhs` where `lhs` is a numeric variable giving the data values and `rhs` a factor with two levels giving the corresponding groups.

data

an optional matrix or data frame (or similar: see `model.frame`) containing the variables in the formula formula. By default the variables are taken from `environment(formula)`.

subset

an optional vector specifying a subset of observations to be used.

na.action

a function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

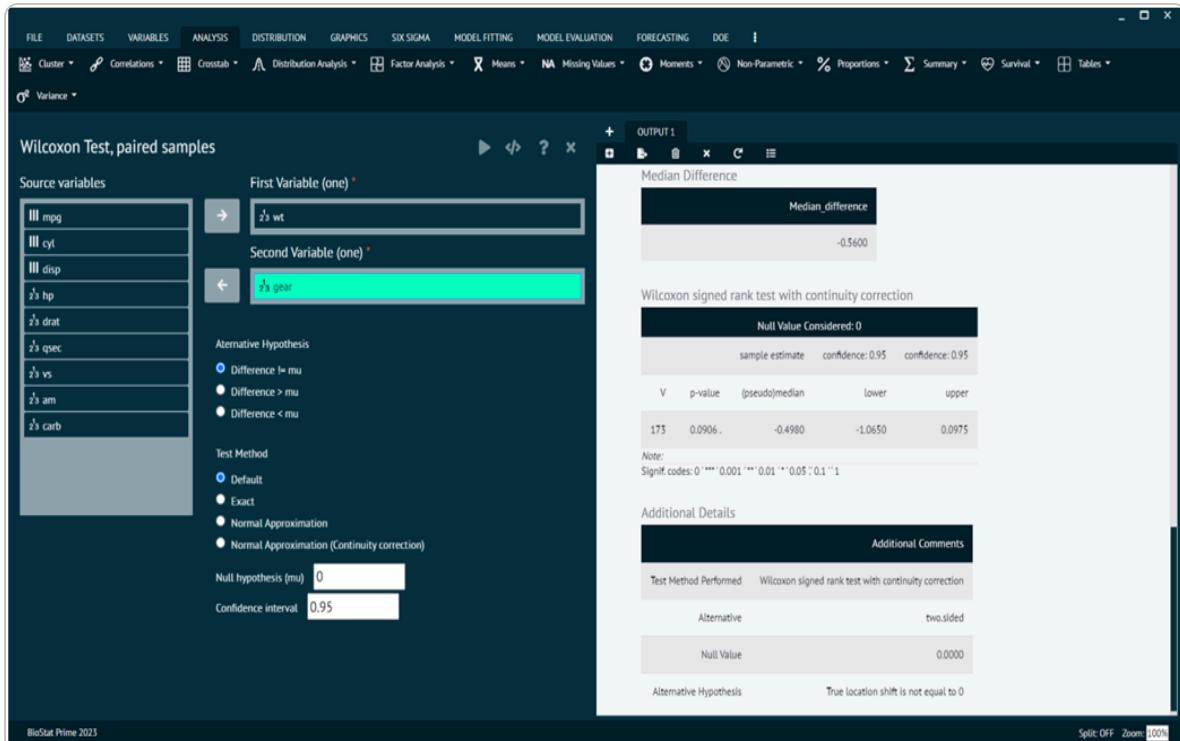
Wilcoxon Test, Paired samples

The Wilcoxon signed-rank test for paired samples is a non-parametric statistical test used to determine if there is a significant difference between the medians of two related groups. It is an alternative to the paired t-test when the assumption of normality is not met, or when dealing with ordinal or non-normally distributed data.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select Non-Parametric -> The non-parametric tab leads to Wilcoxon Test, paired samples -> In the dialog select the variable and options according to the requirement -> Execute the dialog.



alt text

⚠ Arguments

x

numeric vector of data values. Non-finite (e.g., infinite or missing) values will be omitted.

y

an optional numeric vector of data values: as with `x` non-finite values will be omitted.

alternative

a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". User can specify just the initial letter.

mu

a number specifying an optional parameter used to form the null hypothesis. See 'Details'.

paired

a logical indicating whether user want a paired test.

exact

a logical indicating whether an exact p-value should be computed.

correct

a logical indicating whether to apply continuity correction in the normal approximation for the p-value.

conf.int

a logical indicating whether a confidence interval should be computed.

conf.level

confidence level of the interval.

formula

a formula of the form `lhs ~ rhs` where `lhs` is a numeric variable giving the data values and `rhs` a factor with two levels giving the corresponding groups.

data

an optional matrix or data frame (or similar: see `model.frame`) containing the variables in the formula formula. By default the variables are taken from `environment(formula)`.

subset

an optional vector specifying a subset of observations to be used.

na.action

a function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

Proportions

Two Sample Proportion Test

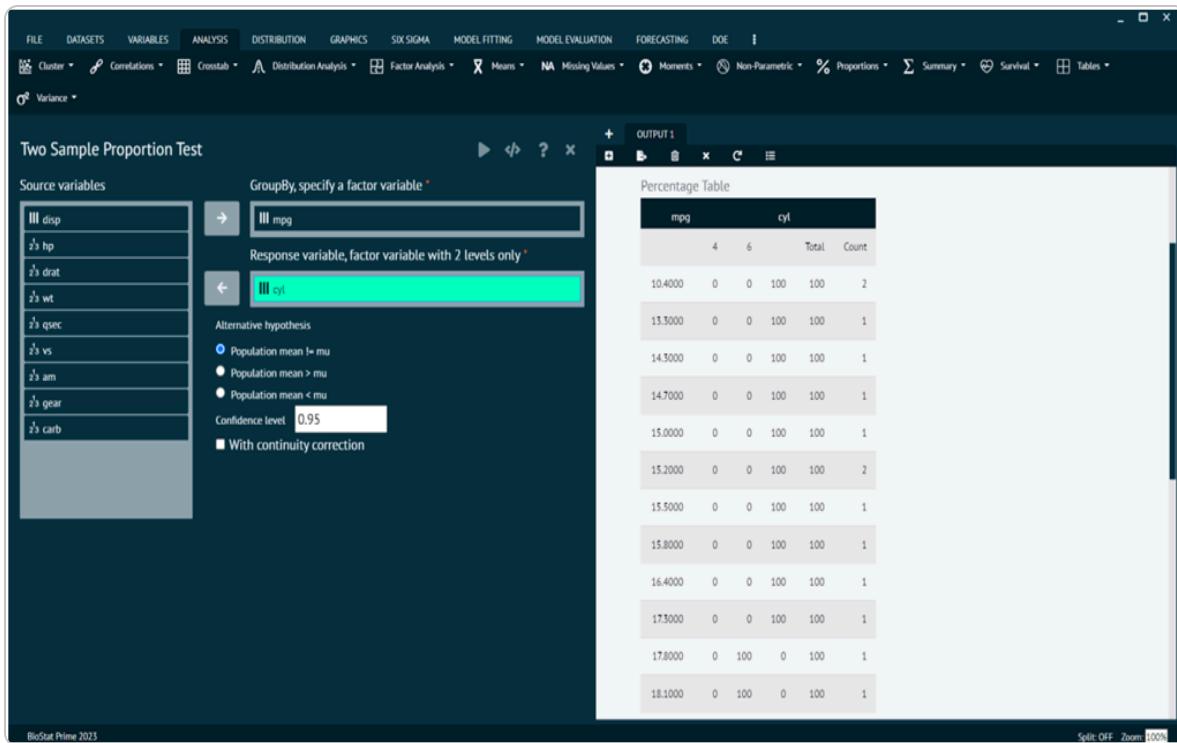
A two-sample proportion test is a statistical method used to compare the proportions of two independent groups. This test is often applied when you have two sets of binary data, and you want to determine if there is a significant difference between the proportions of success (or presence of an attribute) in the two groups.

⚠ prop.test can be used for testing the null that the proportions (probabilities of success) in several groups are the same, or that they equal certain given values.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select proportions -> The proportions tab leads to Two Sample Proportion Test -> In the dialog select the variable and options according to the requirement -> Execute the dialog.



alt text

⚠ Arguments

x

a vector of counts of successes, a one-dimensional table with two entries, or a two-dimensional table (or matrix) with 2 columns, giving the counts of successes and failures, respectively.

n

a vector of counts of trials; ignored if x is a matrix or a table.

p

a vector of probabilities of success. The length of p must be the same as the number of groups specified by x, and its elements must be greater than 0 and less

than 1.

alternative

a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". User can specify just the initial letter. Only used for testing the null that a single proportion equals a given value, or that two proportions are equal; ignored otherwise.

conf.level

confidence level of the returned confidence interval. Must be a single number between 0 and 1. Only used when testing the null that a single proportion equals a given value, or that two proportions are equal; ignored otherwise.

correct

a logical indicating whether Yates' continuity correction should be applied where possible.

Single Sample Exact Binomial Test

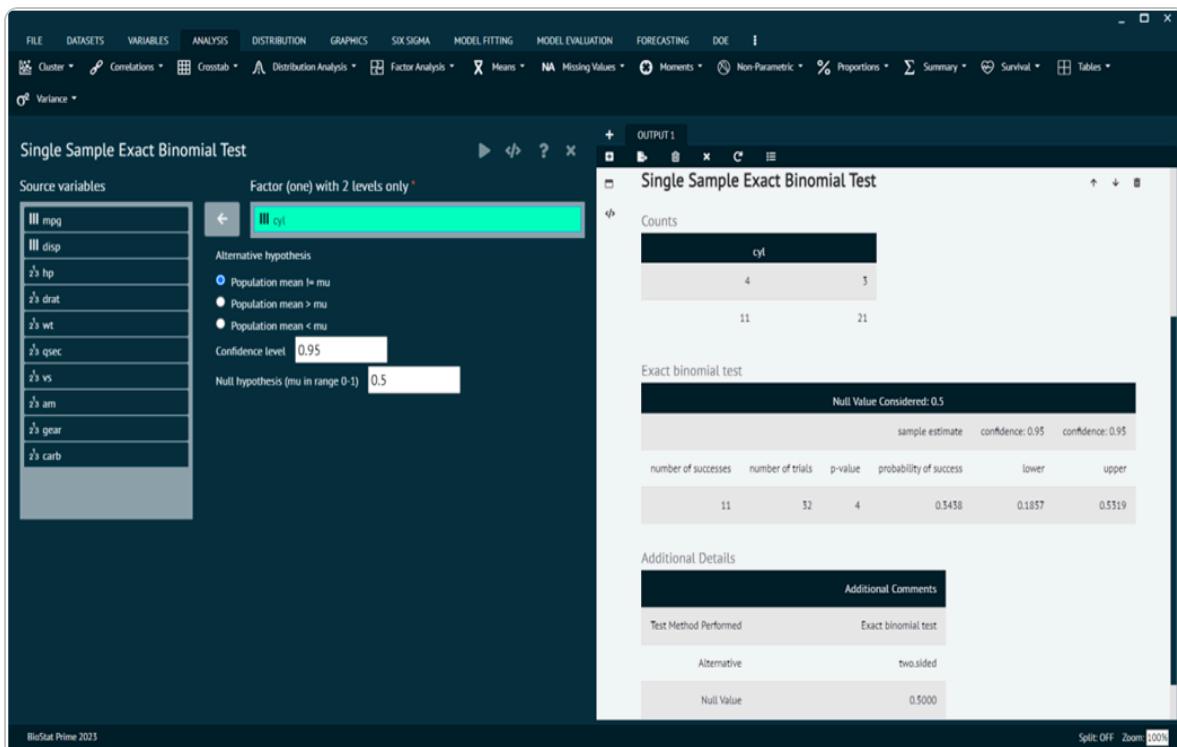
The single sample exact binomial test is a statistical test used to assess whether the observed proportion of successes in a binary outcome significantly differs from a hypothesized proportion. It is appropriate when you have a single group or sample with binary data, and you want to test if the observed proportion is consistent with a specific value.

- ⚠️ Performs an exact test of a simple null hypothesis about the probability of success in a Bernoulli experiment.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select proportions -> The proportions tab leads to Single Sample Exact Binomial Test -> In the dialog select the variable and options according to the requirement -> Execute the dialog.



alt text

⚠ Arguments

x

number of successes, or a vector of length 2 giving the numbers of successes and failures, respectively.

n

number of trials; ignored if x has length 2.

p

hypothesized probability of success.

alternative

indicates the alternative hypothesis and must be one of "two.sided", "greater" or "less". You can specify just the initial letter.

conf.level

confidence level for the returned confidence interval.

Single Sample Proportion Test

The single sample proportion test is a statistical test used to determine whether the observed proportion of successes in a binary outcome significantly differs from a hypothesized proportion. This test is particularly useful when you have a single group or sample with binary data, and you want to evaluate whether the sample proportion is consistent with a specified value.

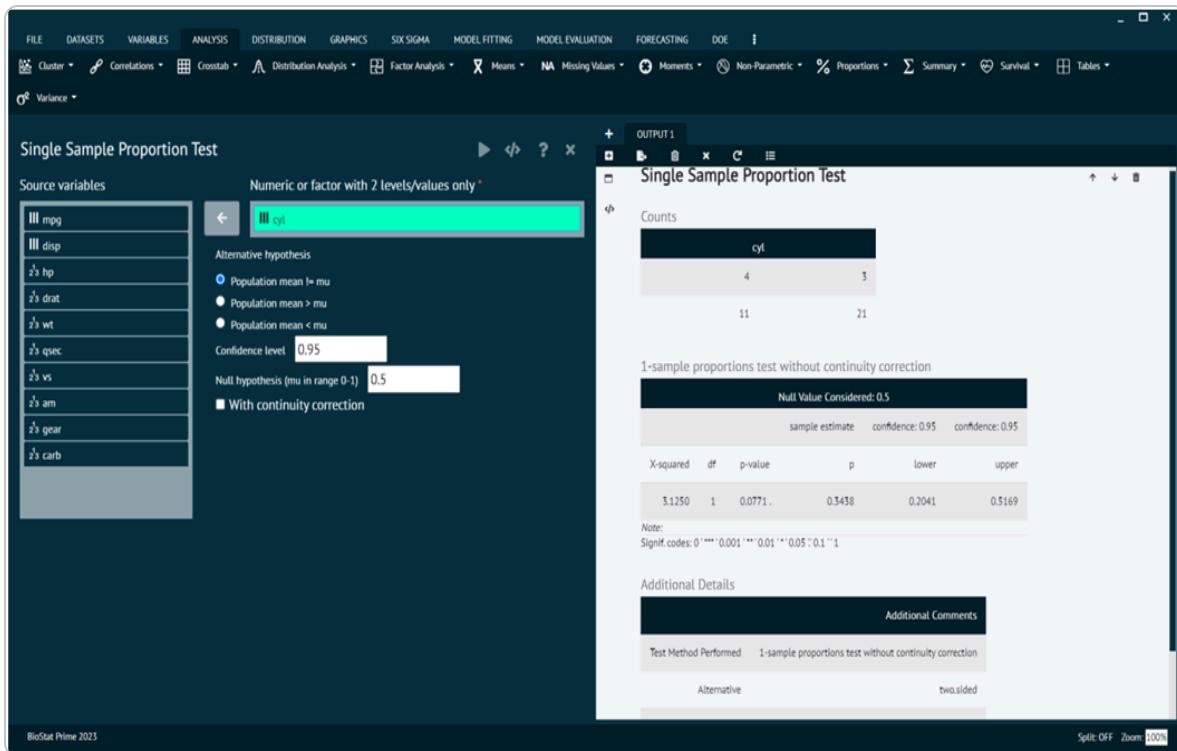


prop.test can be used for testing the null that the proportions (probabilities of success) in several groups are the same, or that they equal certain given values.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select proportions -> The proportions tab leads to Single Sample Proportion Test -> In the dialog select the variable and options according to the requirement -> Execute the dialog.



alt text

⚠ Arguments

x

a vector of counts of successes, a one-dimensional table with two entries, or a two-dimensional table (or matrix) with 2 columns, giving the counts of successes and failures, respectively.

n

a vector of counts of trials; ignored if x is a matrix or a table.

p

a vector of probabilities of success. The length of p must be the same as the number of groups specified by x, and its elements must be greater than 0 and less

than 1.

alternative

a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". User can specify just the initial letter. Only used for testing the null that a single proportion equals a given value, or that two proportions are equal; ignored otherwise.

conf.level

confidence level of the returned confidence interval. Must be a single number between 0 and 1. Only used when testing the null that a single proportion equals a given value, or that two proportions are equal; ignored otherwise.

correct

a logical indicating whether Yates' continuity correction should be applied where possible.

Summary

Descriptive Statistics

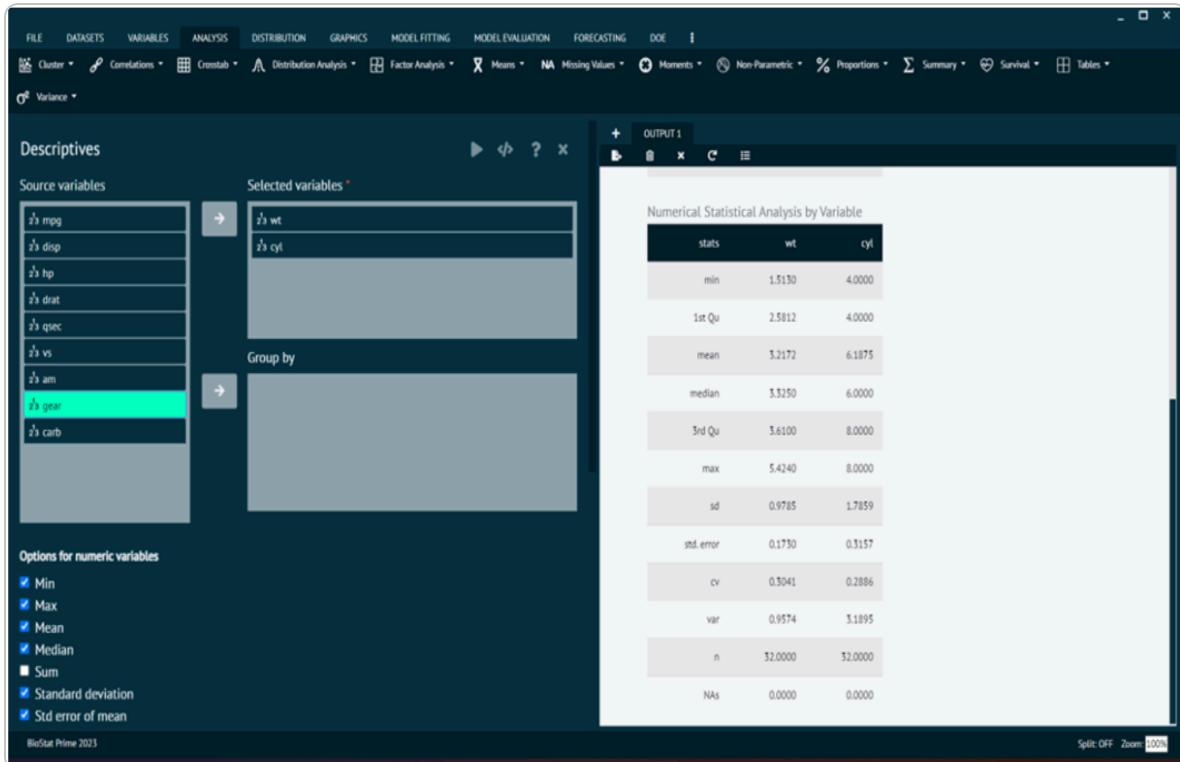
Descriptive statistics are used to summarize and describe a dataset, providing a clear and concise overview of its main characteristics. There are several types of descriptive statistics commonly used, including Measures of central tendency are statistical measures that describe the centre or typical value of a dataset. They provide insight into where the "average" or "middle" of the data lies.

To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the analysis tab in main menu -> Select summary.

The summary tab contains an option namely descriptive that contains all the descriptive statistic analysis techniques. Once the descriptive techniques are chosen and variables are targeted then, user needs to execute the dialog to see the analysis in output window.



alt text

In Descriptive function of summary tab, user can opt for options like MIN, MAX, MEAN, MEDIAN, SUM, STANDARD DEVIATION, STD ERROR MEAN as per the requirement.

Furthermore, other functions can also be applied on the dataset like explore dataset, explore variables, frequencies.

! Outputs the following descriptive statistics: min, max, mean, median, sum, sd, stderror, iqr, Quantiles. If Quantiles is selected, you can specify the comma separated quantiles needed.

i In addition to these, the user can pass, a list of comma separated statistical function names for example var.

! Arguments

datasetColumnObjects

selected scale variables (say *Datasetvar1*, *Datasetvar2*)

groupByColumnObjects

one or more factor variables to group by (say *Datasetvar3*, *Datasetvar4*)

statFunctionList

List of functions. The ones set to TRUE will be executed. (say min=TRUE, sd=TRUE)

quantilesProbs

Probabilities of the quantiles

additionalStats

Addition statistical function that user can pass (say var)

datasetName

Name of the dataset from which datasetColumnObjects and groupByColumnObjects are chosen

long_table

Long table option is introduced to accommodate analysis done on a large number of variables. Choosing the long format controls the width of the output table making it easy to view results without having to scroll right on the output window.

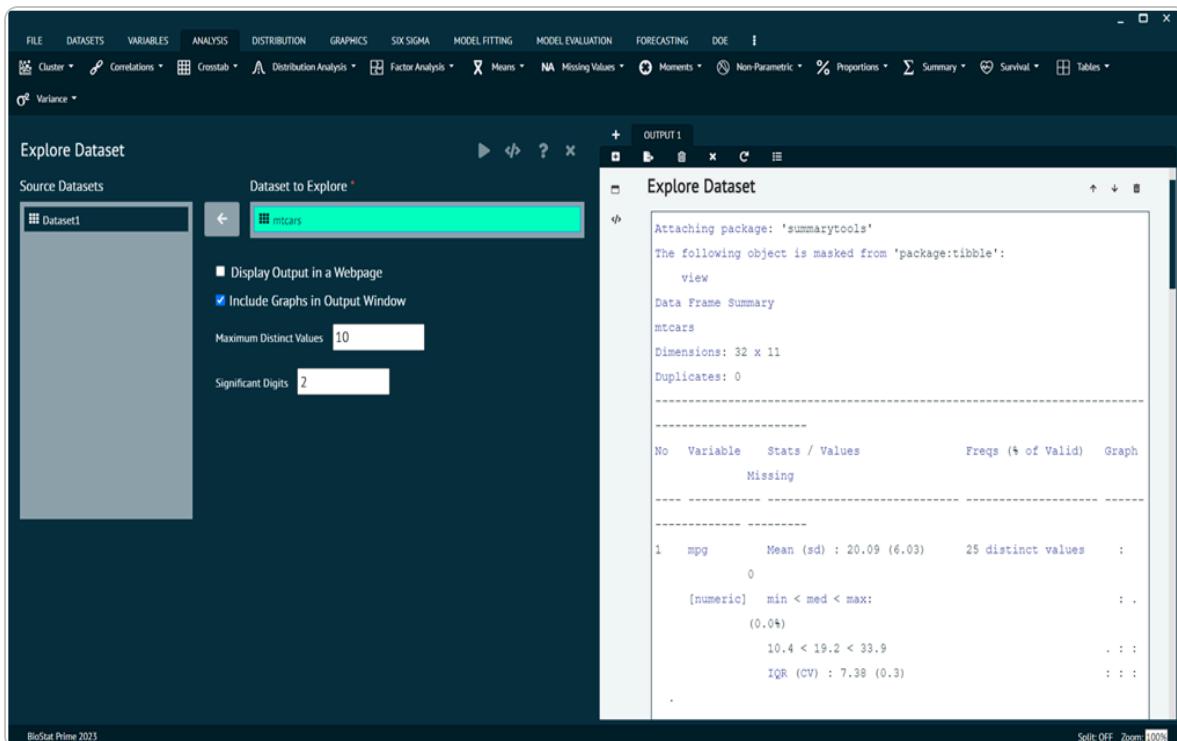
Explore Dataset

This section of summary tab gives user a chance to explore the dataset. The picture below shows the way user can opt for required dataset and explore it.

This creates a table describing a dataset. Descriptions include the dataset name, number of observations, number of variables, number of duplicate records, variable names, variable classes, variable summary statistics, and graphs.

i This tool is meant more for data exploration and cleaning purposes, rather than data analysis purposes.

i A text version of the table is displayed by default, but a pretty html version can optionally be displayed in the default web browser.



alt text

⚠ Arguments

Dataset to Explore

Dataset that you want to describe

Display Output in a Webpage

Check if you want to display a pretty version of the table in the default web browser. This version will have graphs included.

Include Graphs in Output Window

Check if you want to include text versions of the graphs in the BioStat output window

Maximum Distinct Values

The maximum number of values to display frequencies for. If a variable has more distinct values than this number, the remaining frequencies will be reported as a whole category, along with the number of additional distinct values. For character variables, the most frequent values are displayed, so this also controls how many to show in that case. The default is 10.

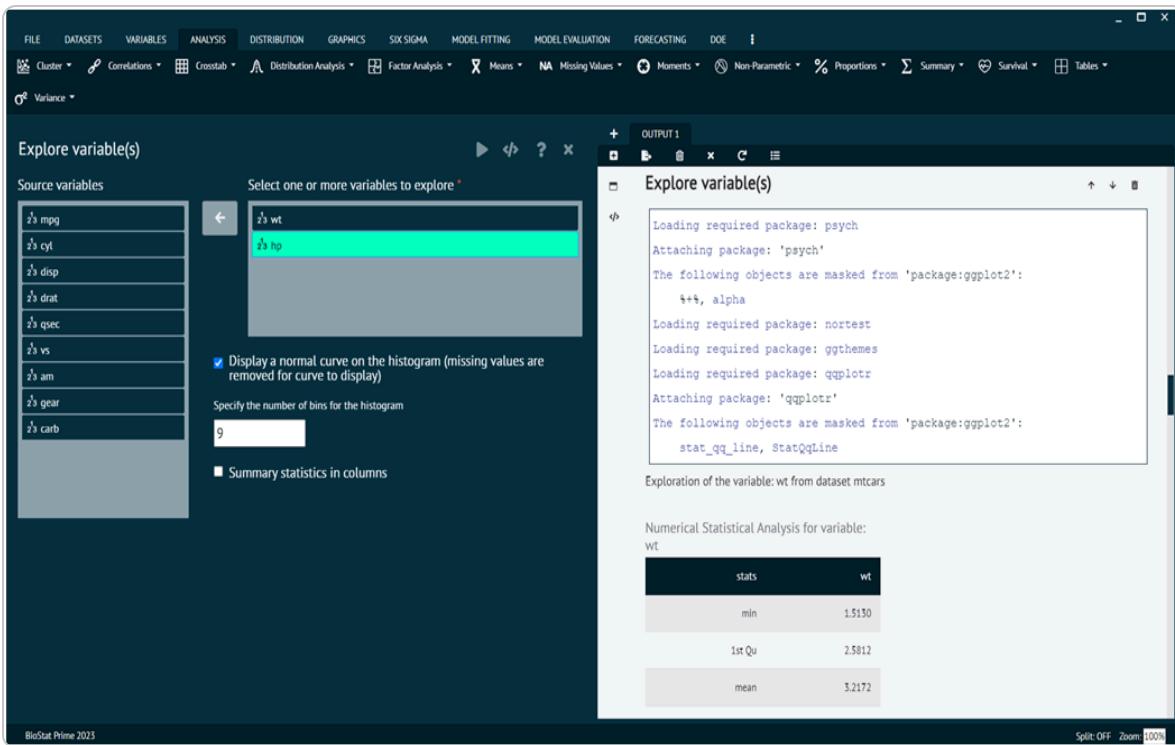
Explore Variables

This section of summary tab gives user a chance to explore the variables of loaded dataset. The picture below shows the way user can choose required variables and execute the dialog to explore it.

Outputs the following descriptive statistics and plots: min, max, mean, median, modes, sum, sd, cv (coefficient of variance), var, stderror, skew, kurtosi, mad, iqr, and quartiles.

i In addition, 95% confidence interval for mean and sd are computed.

i Histogram and QQ plots are displayed.



alt text

Frequency

This section of summary tab gives user a chance to evaluate the frequencies of different variables of loaded dataset. The picture below shows the way user can choose required variables and execute the dialog to evaluate the frequency of selected variable.

Generates the frequencies for every unique value in one or more variables or column names selected.

The screenshot shows the BioStat Prime 2023 software interface. The top menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION, FORECASTING, DOE, and various statistical tools like Cluster, Correlations, Crosstab, Distribution Analysis, Factor Analysis, Means, Missing Values, Moments, Non-Parametric, Proportions, Summary, Survival, and Tables. A dropdown menu for 'Variance' is open.

The main window displays a 'Frequency Table' for the 'hp' variable. On the left, under 'Source variables', the list includes mpg, cyl, disp, drat, wt, qsec, vs, am, gear, and carb. The 'wt' variable is highlighted with a green background. An arrow button points from the source list to the 'Select variables' list on the right, which contains 'hp'. The output panel shows the 'Frequency Table for hp' with the following data:

hp	Frequency	Percent	CumPercent	Valid Percent	Valid CumPercent
110	3	9.3750	43.7500	9.3750	43.7500
175	3	9.3750	68.7500	9.3750	68.7500
180	3	9.3750	78.1250	9.3750	78.1250
66	2	6.2500	15.6250	6.2500	15.6250
123	2	6.2500	53.1250	6.2500	53.1250
150	2	6.2500	59.3750	6.2500	59.3750
245	2	6.2500	93.7500	6.2500	93.7500
52	1	3.1250	3.1250	3.1250	3.1250
62	1	3.1250	6.2500	3.1250	6.2500
65	1	3.1250	9.3750	3.1250	9.3750
91	1	3.1250	18.7500	3.1250	18.7500
93	1	3.1250	21.8750	3.1250	21.8750
95	1	3.1250	25.0000	3.1250	25.0000

alt text

Survival

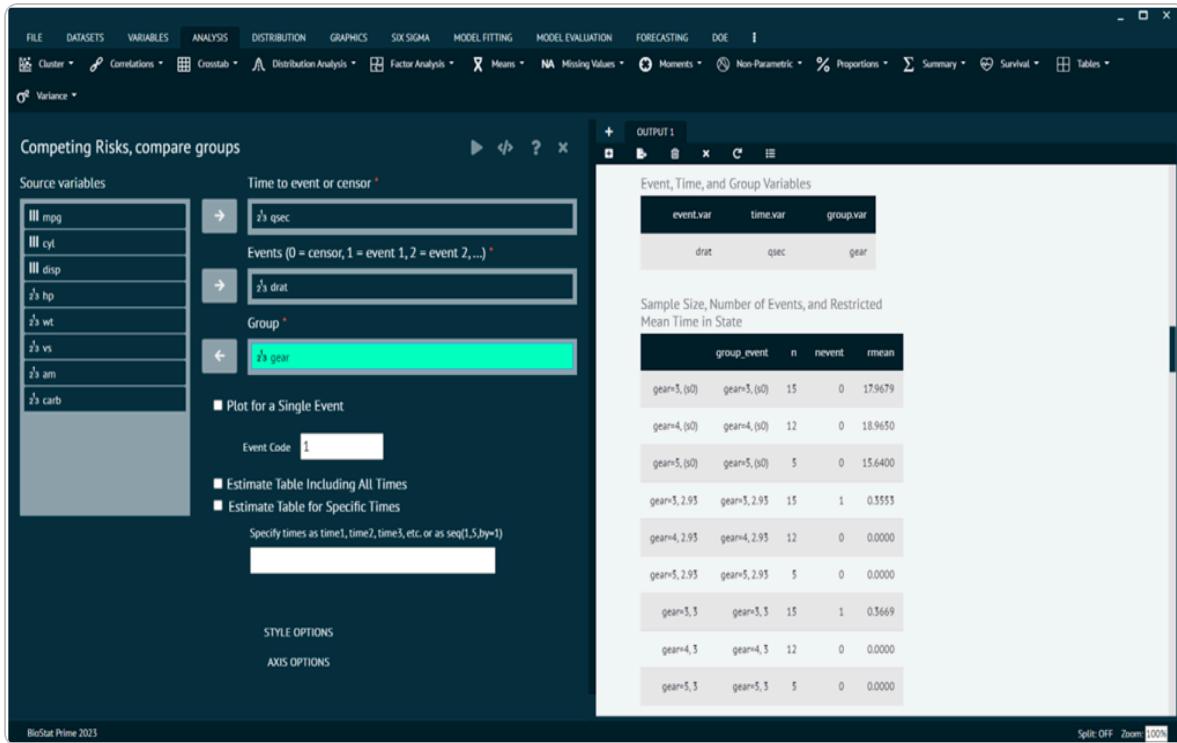
In statistics, "survival" refers to the analysis of time until an event of interest occurs. The primary goal of survival analysis is to estimate the time until an event happens and to understand the factors that may influence the time to event. Survival analysis is particularly relevant when dealing with time-to-event data, and it provides a powerful tool for studying and modeling the timing of various events of interest in different fields. It allows researchers to make predictions about the probability of events occurring over time and to compare survival experiences between different groups.

Competing Risks, Compare groups

In survival analysis, when dealing with multiple events or outcomes that are considered as competing risks, one needs to account for the fact that an individual or subject may experience one type of event, preventing the occurrence of another. The concept of competing risks arises when there are multiple possible failure events, and one is interested in understanding the probabilities and risks associated with each event.

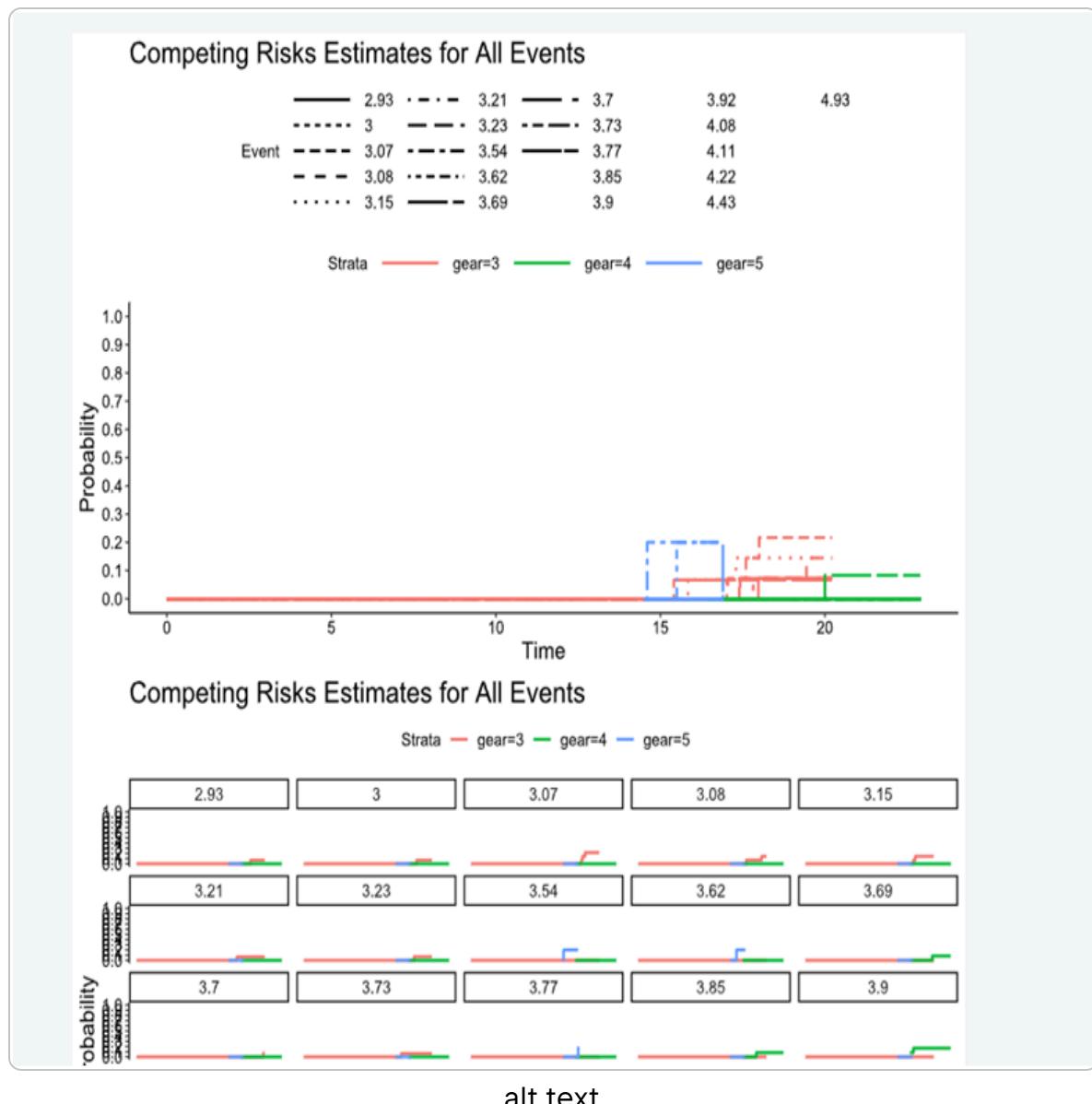
To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> Click on the analysis tab in main menu -> Select survival -> The survival tab leads to Competing Risks, Compare groups -> In the dialog select the variable and options according to the requirement -> Execute the dialog.



alt text

Competing Risks Estimate for all events in the output window.



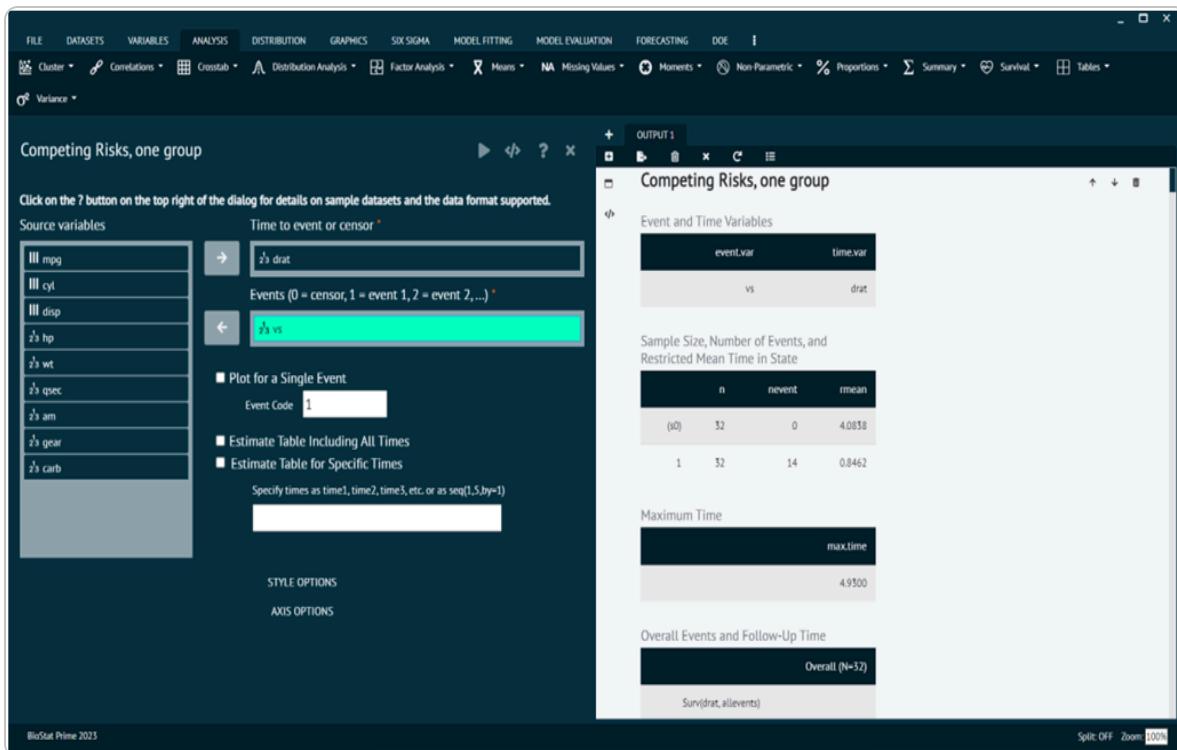
Competing Risks, One group

When dealing with competing risks in a single group, one is interested in understanding the probabilities and risks associated with different types of events that may occur, but one does not have a distinct comparison group. The analysis will focus on estimating and comparing the cumulative incidence functions for the competing events within the same group.

To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> Click on the analysis tab in main menu -> Select survival -> The survival tab leads to Competing Risks, one group -> In the dialog select the variable and

options according to the requirement -> Execute the dialog.



alt text

Kaplan-Meier Estimation, compare groups

Kaplan-Meier Estimation, One group

Kaplan-Meier estimation is a non-parametric method used in survival analysis to estimate the probability of an event (e.g., survival) occurring at a given time. It is often applied when studying the time until an event of interest, such as the failure of a system, the onset of a disease, or the occurrence of a specific event in a study.

To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> Click on the analysis tab in main menu -> Select survival -> The survival tab leads to Kaplan-Meier Estimation, one group -> In the dialog select the variable and options according to the requirement -> Execute the dialog.

Kaplan-Meier Estimation, One Group

Click on the ? button on the top right of the dialog for details on sample datasets and the data format supported.

Source variables

Time to event or censor *

Event (1 = event, 0 = censor) *

Plot Type

Survival

Failure

Estimate Table Including All Times

Estimate Table for Specific Times

Specify times as time1, time2, time3, etc. or as seq(1,5,by=1)

STYLE OPTIONS

AXIS OPTIONS

BioStat Prime 2023

OUTPUT 1

Kaplan-Meier Estimation, One Group

Survival Summary: Surv(gear,am)

Overall (N=32)	
Surv(gear, am)	
- N	32.000
- Events	15.000
- Median Survival	5.0000
- Median Follow-Up	4.0000

Restricted Mean and Median Survival Times

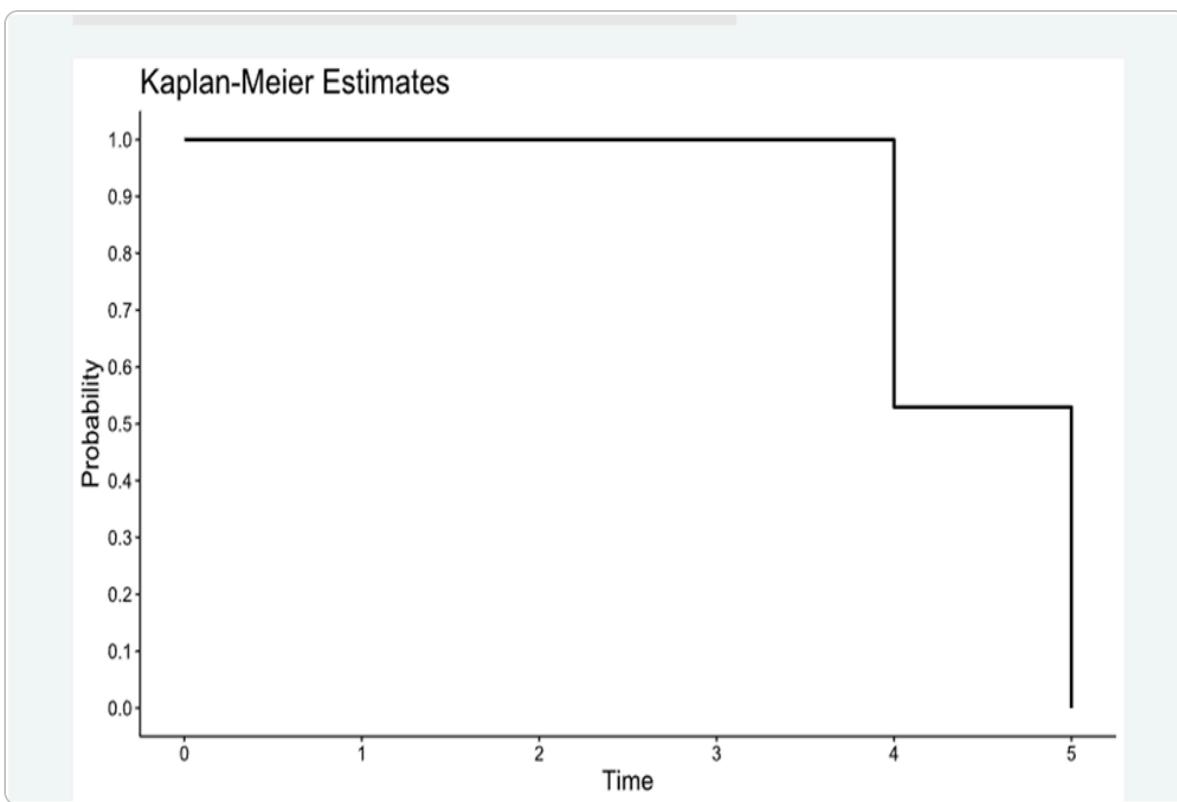
rmean	rmeanstd.error	median	conf.low	conf.high
4.5294	0.1211	5	4	NA

Kaplan-Meier Estimates

The Kaplan-Meier survival curve plot shows the estimated survival probability over time. The y-axis is labeled 'Probability' and ranges from 0.0 to 1.0. The x-axis is labeled 'Time' and ranges from 0 to 5. The curve starts at 1.0 at time 0 and remains constant until approximately time 3.8. At time 3.8, there is a vertical drop to about 0.55. From time 3.8 to 4.5, the probability remains at 0.55. At time 4.5, there is another vertical drop to 0.0. The curve stays at 0.0 until time 5. The plot area has a light gray background with white grid lines.

alt text

Kaplan-Meier Estimates in the output.



alt text

Tables

Tables in statistics are a common way to organize and present data for easy interpretation. Different types of tables are used depending on the nature of the data and the specific goals of the analysis.

Table, Advanced

The examples for this category are ANOVA Table, Regression Coefficients Table, Survival Analysis Table etc.

The screenshot shows the BioStat Prime 2023 software interface. On the left, under 'Source variables', there is a list of variables: cyl, hp, wt, am, carb, vs, gear, and spec. A 'Groups to Compare' section contains '2's drat'. A note states: 'NOTE: At least one variable must be specified in at least one of the below: ANOVA Test, Kruskal-Wallis Test, Median Test, Pearson's Chi-Square Test, Fisher's Exact Test, Ordinal Trend Test, or No Test'. Below this are sections for 'Variables for ANOVA Test' (containing '2's vs'), 'Variables for Kruskal-Wallis Test' (containing '2's gear'), and 'Variables for Median Test' (containing '2's spec'). On the right, the 'OUTPUT 1' tab is active, showing 'Variable Summaries' for the 'vs' group. The table includes columns for -Mean, Median, Q1, Q3, SD, N, and various statistical tests like 2.76 (N=2), 2.95 (N=1), 3 (N=1), 3.07 (N=3), 3.08 (N=2), 3.15 (N=2), 3.21 (N=1), 3.23 (N=1), and 3.54 (N=1). Similar tables are shown for 'gear' and 'spec' groups.

alt text

Table, Basic

The examples for this category are Frequency Distribution Table, Summary Statistics Table, Contingency Table (Cross-tabulation) etc.

FILE DATASETS VARIABLES ANALYSIS DISTRIBUTION GRAPHICS SIX SIGMA MODEL FITTING MODEL EVALUATION FORECASTING DOE

σ^2 Variance

Table, Basic

Source variables

- mpg
- cyl
- disp
- wt
- qsec
- gear

Variables to Summarize *

- drat
- hp
- vs

Groups to Compare (optional)

- am

Strata (optional)

- carb

Table Title

Variable Summaries

carb 0 (N=19) 1 (N=15)

	drat	hp
- Mean (SD)	3.180 (0.478)	4.058 (0.153)
- Median (Q1, Q3)	3.080 (2.920, 3.590)	4.080 (4.022, 4.115)

vs

	drat	vs
- Mean (SD)	104.000 (6.557)	72.500 (13.675)
- Median (Q1, Q3)	105.000 (101.000, 107.500)	66.000 (65.750, 72.750)

2 drat

	drat	vs
- Mean (SD)	3.292 (0.429)	4.310 (0.493)
- Median (Q1, Q3)	3.150 (3.098, 3.555)	4.270 (4.025, 4.555)

Biostat Prime 2023

Split OFF Zoom 100%

alt text

Variance

In statistics, variance is a measure of the dispersion or spread of a set of values. It quantifies how much individual data points in a dataset differ from the mean (average) of the dataset. A low variance indicates that the values tend to be close to the mean, while a high variance indicates that the values are more spread out.

Bartlett's Test

Bartlett's test is a statistical test used to assess whether the variances of two or more groups are equal. It is commonly employed when conducting analysis of variance (ANOVA) to determine whether there are significant differences in the variances between groups. The test is sensitive to departures from normality.

To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> Click on the analysis tab in main menu -> Select variance -> The variance tab leads to Bartlett's test -> In the dialog select the variable and options according to the requirement -> Execute the dialog.

The screenshot shows the BioStat Prime 2023 software interface. The main window title is "Bartlett's Test". On the left, under "Source variables", are listed: cyl, disp, hp, drat, wt, qsec, am. On the right, under "Response variable", is "carb". Below that, under "Numeric or factors variables", are listed: mpg, vs, gear. To the right of the dialog, the "OUTPUT1" tab is selected, displaying a table titled "Variance". The table has columns: gear, vs, mpg, stats.var. The data rows are:

gear	vs	mpg	stats.var
3	0	10.4000	0.0000
		13.3000	NA
		14.3000	NA
		14.7000	NA
		15.0000	NA
		15.2000	0.5000
		15.5000	NA
		15.8000	NA
		16.4000	NA
		17.3000	NA
		17.8000	NA
		18.1000	NA

alt text

Levene's Test

Levene's test is a statistical test used to assess whether the variances of two or more groups are equal. Like Bartlett's test, Levene's test is commonly used in analysis of variance (ANOVA) to evaluate the assumption of homogeneity of variances (homoscedasticity). It is less sensitive to departures from normality compared to Bartlett's test and is often considered a robust alternative.

To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> Click on the analysis tab in main menu -> Select variance -> The variance tab leads to Levene's test -> In the dialog select the variable and options according to the requirement -> Execute the dialog.

The screenshot shows the BioStat Prime software interface. The main window displays the 'Levene's Test' dialog. On the left, under 'Source variables', there is a list of variables: cyl, disp, hp, wt, qsec, vs, am, gear, carb. A 'Response Variable (one)' dropdown is set to 'drat'. A 'Factor Variable' dropdown is set to 'mpg'. Under 'Center', the 'Median' option is selected. To the right, the 'OUTPUT' tab is active, showing the 'Levene's Test' results. The 'Summary Statistics' table includes columns: mpg, count_drat, variance_drat, sd_drat, std_err_drat, min_drat, Quantile_1st_drat, and mean_drat. The data rows show values for mpg ranging from 10.4000 to 18.1000. The 'Split OFF Zoom 100%' status bar is visible at the bottom right.

alt text

Variance Test, F-test

The F-test is a statistical test used to compare variances between two or more groups. It is often employed in the context of analysis of variance (ANOVA) to assess whether the variances of different groups are equal. The F-test follows an F-distribution, and the null hypothesis is that the variances are equal across groups.

FILE **DATASETS** **VARIABLES** **ANALYSIS** **DISTRIBUTION** **GRAPHICS** **SIX SIGMA** **MODEL FITTING** **MODEL EVALUATION** **FORECASTING** **DOE** **⋮**

σ² Variance

Variance Test, F-test

Source variables

- mpg
- disp
- hp
- wt
- qsec
- vs
- am
- gear
- carb

Response variable *

drat

Factor variable, with only two levels *

cyl

Alternative hypothesis

- Difference != 1
- Difference > 1
- Difference < 1

Confidence level 0.95

OUTPUT 1

Variance Test, F-test

cyl	var
4	0.1356
3	0.1878

F test to compare two variances

Null Value Considered: 1		sample estimate	confidence: 0.95	confidence: 0.95		
F	num df	denom df	p-value	ratio of variances	lower	upper
0.7114	10	20	0.5913	0.7114	0.2565	2.4319

Additional Details

Additional Comments	
Test Method Performed	F test to compare two variances
Alternative	two.sided

Statistica 2021

alt text

Distribution

A statistical distribution, or probability distribution, describes how values are distributed for a field. In other words, the statistical distribution shows which values are common and uncommon. BioStat Prime provides various tests under Distribution tab in main menu like, Chi Square test, Lognormal, Normal, Poisson.

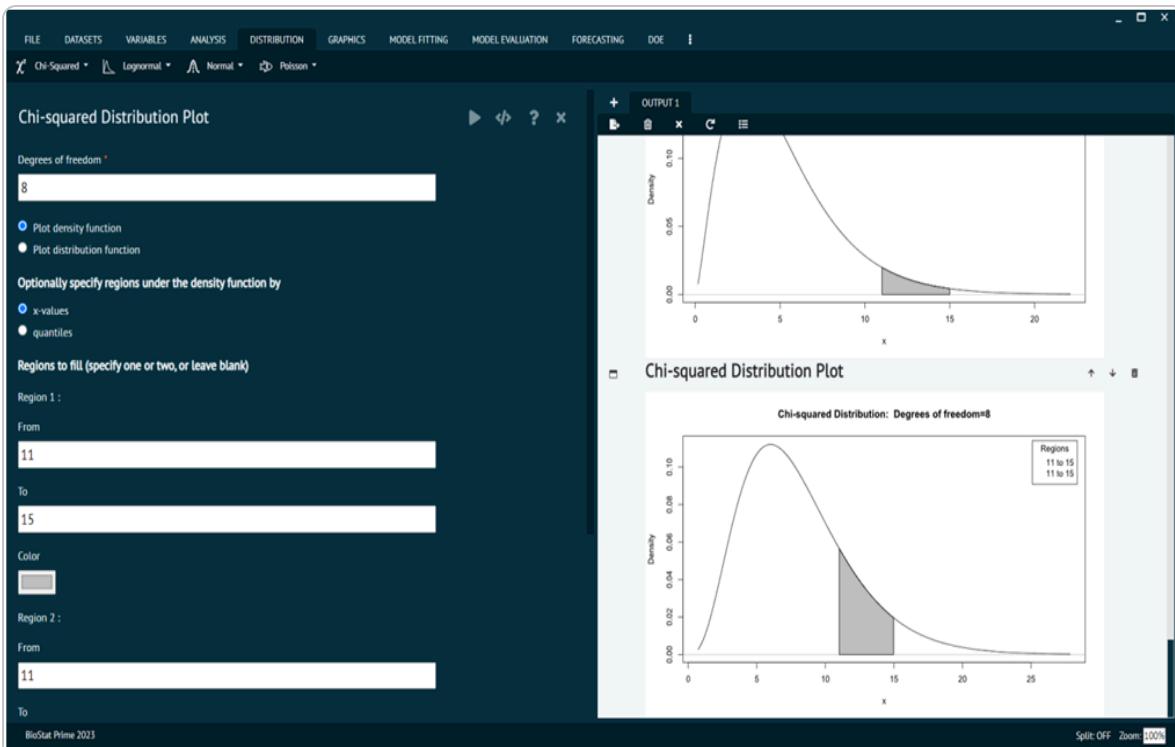
Chi-square test

The chi-square test, also known as the χ^2 test (chi-squared test), is a statistical test used to determine if there is a significant association or independence between two categorical variables in a contingency table. Chi-square statistic is calculated from the contingency table to assess the extent of the association. It measures the difference between the observed frequencies (counts) and the expected frequencies (counts) under the assumption of independence. The formula for calculating the chi-square statistic depends on the table's dimensions but generally involves comparing each observed frequency to its expected value and summing up these differences.

Chi-square Distribution plot

To analyse it in BioStat user must follow the steps as given.

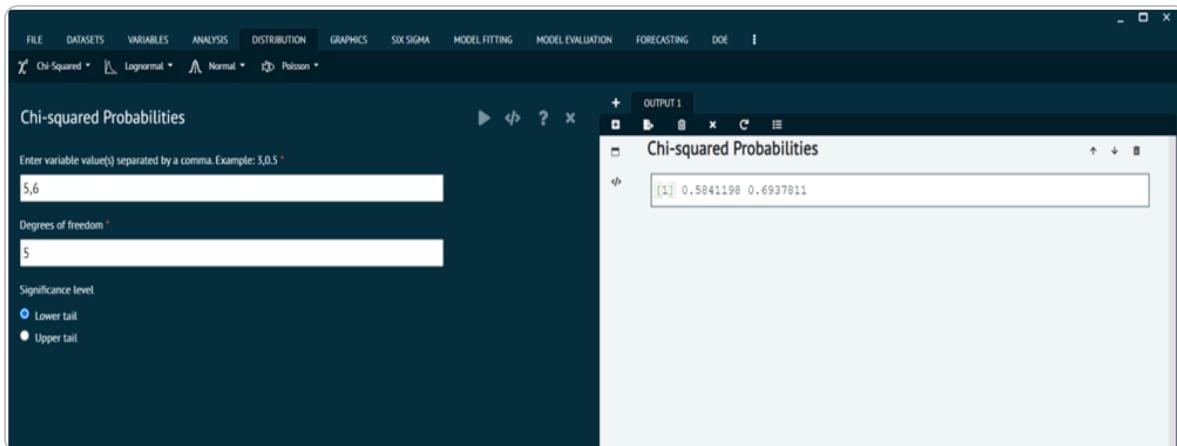
Load the dataset -> Click on the Distribution tab in main menu -> Select Chi square test -> This leads to analysis technique Chi-square Distribution plot in the dialog -> In the dialog window select the options according to the requirements then execute -> The output will be represented in output window.



Chi-square Probabilities

To analyse it in BioStat user must follow the steps as given.

Load the dataset -> Click on the Distribution tab in main menu -> Select Chi square test -> This leads to analysis technique Chi-square Probabilities in the dialog -> In the dialog window select the options according to the requirements then execute -> The output will be represented in output window.

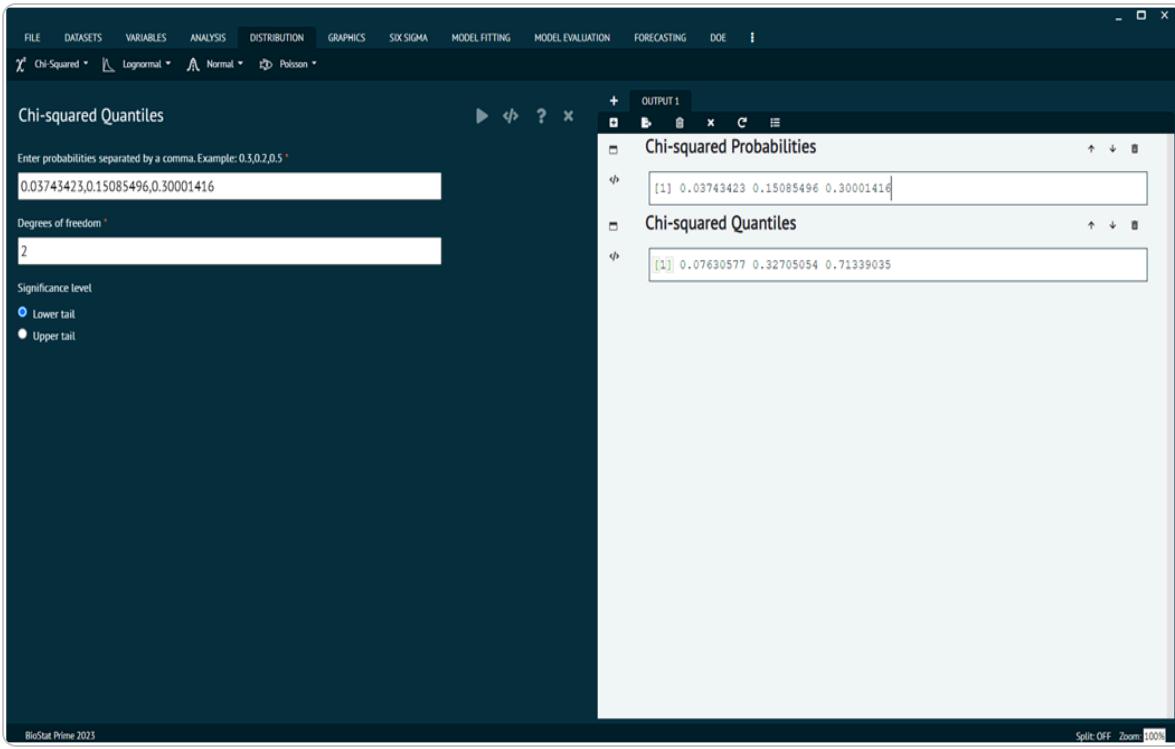


alt text

Chi-square Quantiles

To analyse it in BioStat user must follow the steps as given.

Load the dataset -> Click on the Distribution tab in main menu -> Select Chi square test -> This leads to analysis technique Chi-square Quantiles in the dialog -> In the dialog window select the options according to the requirements then execute -> The output will be represented in output window.



alt text

Sample from Chi-square Distribution

To analyse it in BioStat user must follow the steps as given.

Load the dataset -> Click on the Distribution tab in main menu -> Select Chi square test
this leads to analysis technique **Sample from Chi-square Distribution in the dialog -> In the dialog window select the options according to the requirements then execute -> The output will be represented in output window.**

The screenshot shows the BioStat Prime 2023 software interface. The top menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION, FORECASTING, DOE, and HELP. The DISTRIBUTION tab is selected. A toolbar below the menu contains icons for Chi-Squared, Lognormal, Normal, and Poisson distributions.

The main workspace displays a dataset named "ChiSquaredSamples" containing 29 rows of numerical values. The columns are labeled "#", "1", and "2".

A dialog box titled "Sample from Chi-squared Distribution" is open on the right side of the screen. It contains the following fields:

- Enter name for dataset: ChiSquaredSamples
- Degrees of freedom: 3
- Number of samples (rows): 100
- Number of observations (columns): 1
- Seed: 12345

Under "Add to dataset", there are three checkboxes:

- Sample means
- Sample sums
- Sample standard deviations

The bottom of the interface shows tabs for DATA and VARIABLES, and an R EDITOR tab. The status bar at the bottom indicates "BioStat Prime 2023".

alt text

OUTPUT 1

Sample from Chi-squared Distribution

We don't calculate sample mean, sum or standard deviation when there is a single row or column

	obs1
sample1	3.3425
sample2	3.6706
sample3	1.7874
sample4	2.0465
sample5	9.3165
sample6	2.1787
sample7	1.4721
sample8	6.7020
sample9	1.2030
sample10	2.8099
sample11	3.1757

Split: OFF Zoom: 100%

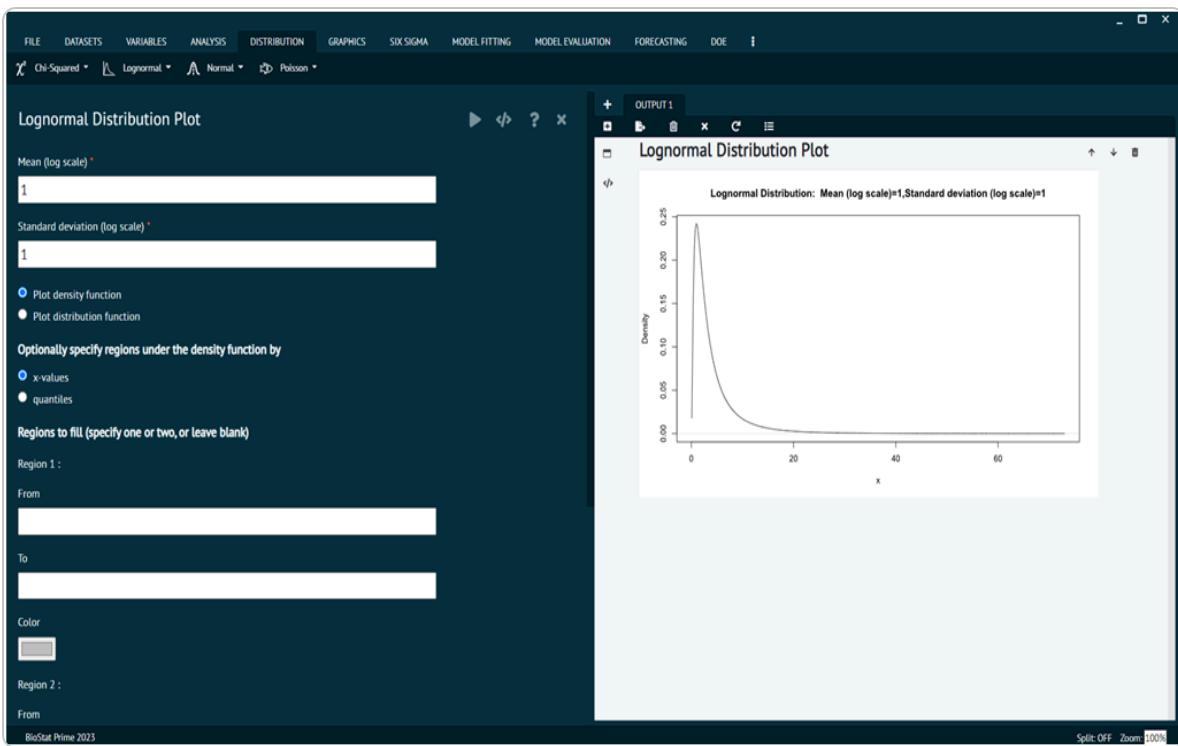
alt text

Lognormal

The lognormal distribution is a continuous probability distribution of a random variable whose logarithm is normally distributed. In other words, if X follows a lognormal distribution, then $Y = \ln(X)$ follows a normal (Gaussian) distribution. The lognormal distribution is often used to model the distribution of random variables that are the product of many independent and identically distributed random variables.

To analyse it in BioStat user must follow the steps as given.

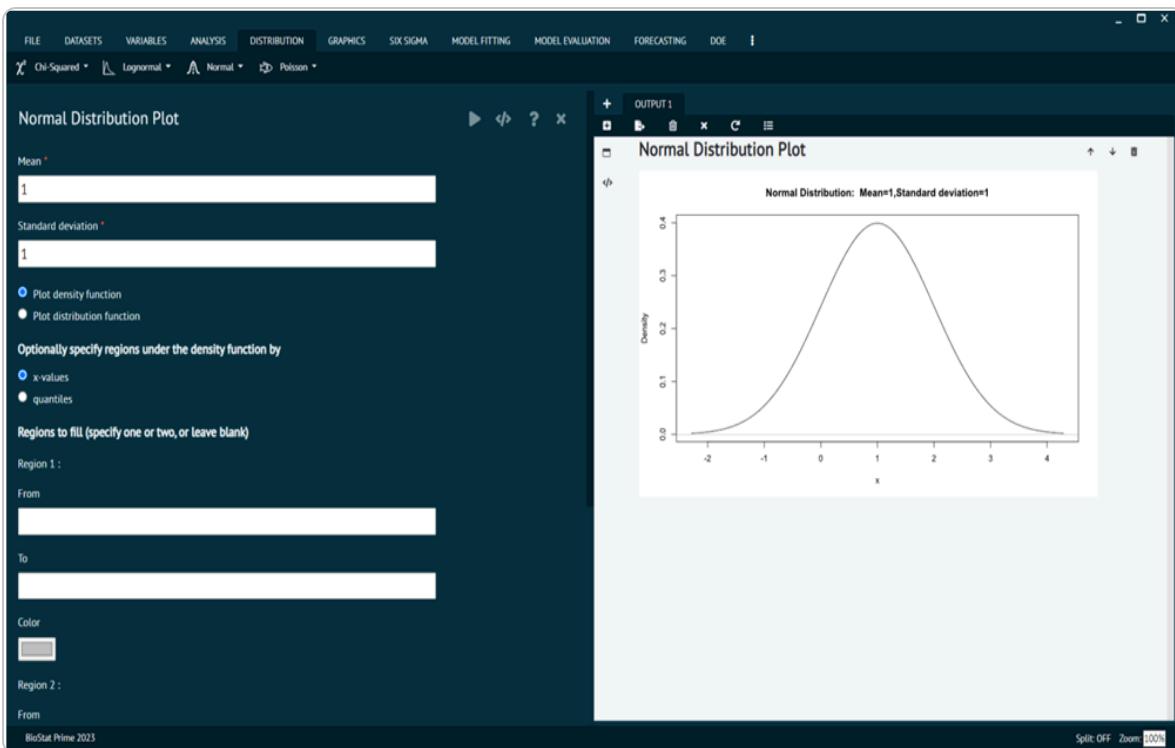
Load the dataset -> Click on the Distribution tab in main menu -> Select Lognormal -> This leads to analysis techniques in the dialog -> In the dialog window select the options according to the requirements then execute -> The output will be represented in output window.



alt text

Normal

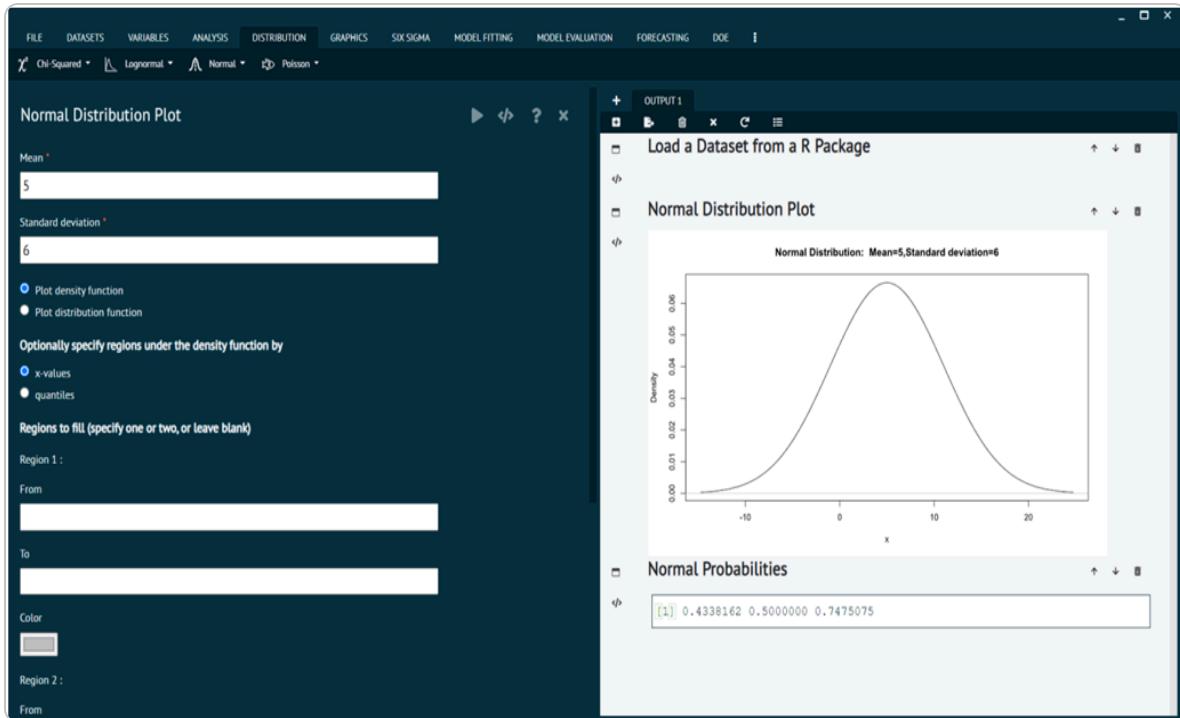
In statistics, "normal" typically refers to the normal distribution, also known as the Gaussian distribution. It's a continuous probability distribution that is symmetric around its mean, forming a bell-shaped curve. Many natural phenomena, such as heights, weight, and IQ scores, tend to follow a normal distribution. The normal distribution is characterized by two parameters: the mean (μ) and the standard deviation (σ). The mean determines the center of the distribution, while the standard deviation determines the spread or dispersion of the data points around the mean.



alt text

Normal Probabilities

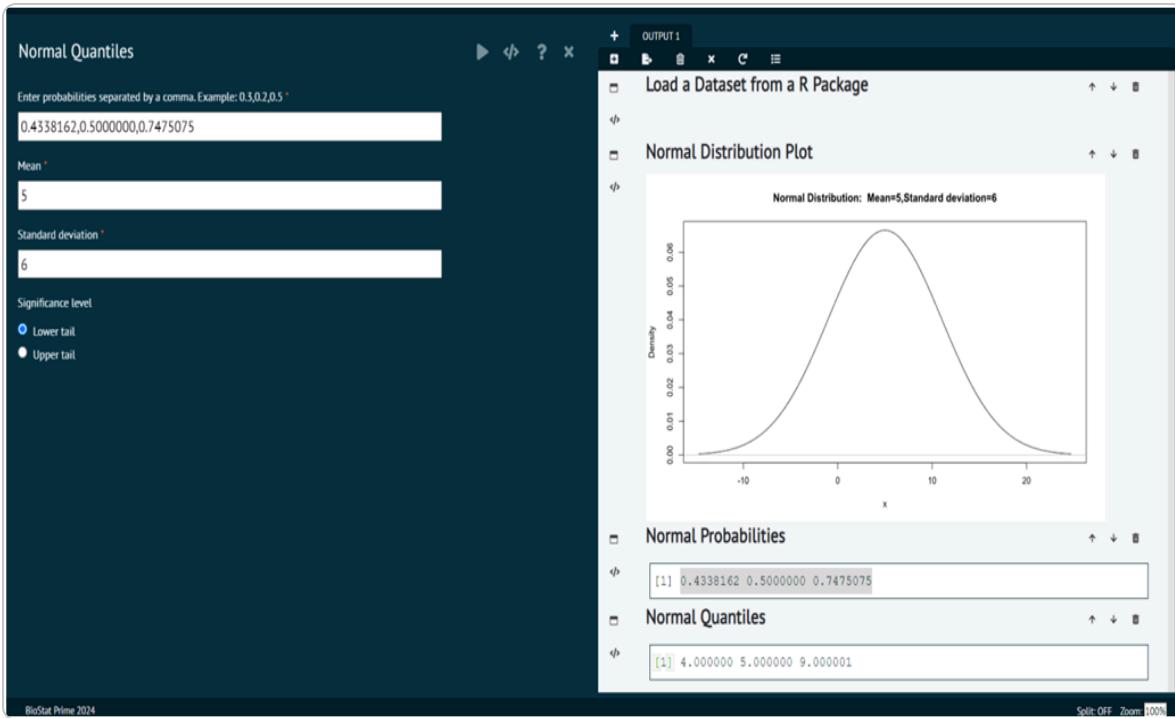
Normal probabilities refer to the probabilities associated with the normal distribution. These probabilities describe the likelihood of observing certain values or ranges of values within a normal distribution.



alt text

Normal Quantiles

Normal Quantiles refer to the values that divide a normal distribution into intervals with equal probabilities. These Quantiles are often used in statistical analysis for constructing confidence intervals, hypothesis testing, and understanding the distribution of data.



alt text

Sample from Normal Distribution

Sampling from a normal distribution allows you to create synthetic data or simulate random variables that follow a normal distribution, which is useful for various statistical analyses and simulations.

The screenshot shows the BioStat Prime 2024 software interface. At the top, there is a menu bar with FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION, FORECASTING, DOE, and a Help icon. Below the menu, there are several dropdown menus: Chi-Squared, Lognormal, Normal, and Poisson.

The main workspace contains three tabs: Dataset1 (selected), mtcars, and NormalSamples. The Dataset1 tab displays a table with one row:

#	Name	Class	Type	Measure	Levels
1	obs1	numeric	double	scale	

Below the table are buttons for DATA and VARIABLES, and an R EDITOR button.

To the right, an OUTPUT 1 window is open with the title "Sample from Normal Distribution". It contains a message: "We don't calculate sample mean, sum or standard deviation when there is a single row or column". Below this, a table titled "Samples from Normal Distribution" lists 11 rows:

	obs1
sample1	8.5132
sample2	9.2568
sample3	4.5442
sample4	2.2790
sample5	8.6355
sample6	-5.9077
sample7	8.7806
sample8	3.3429
sample9	3.2950
sample10	-0.5159
sample11	4.3025

At the bottom of the window, there are buttons for Split, OFF, Zoom, and 100%.

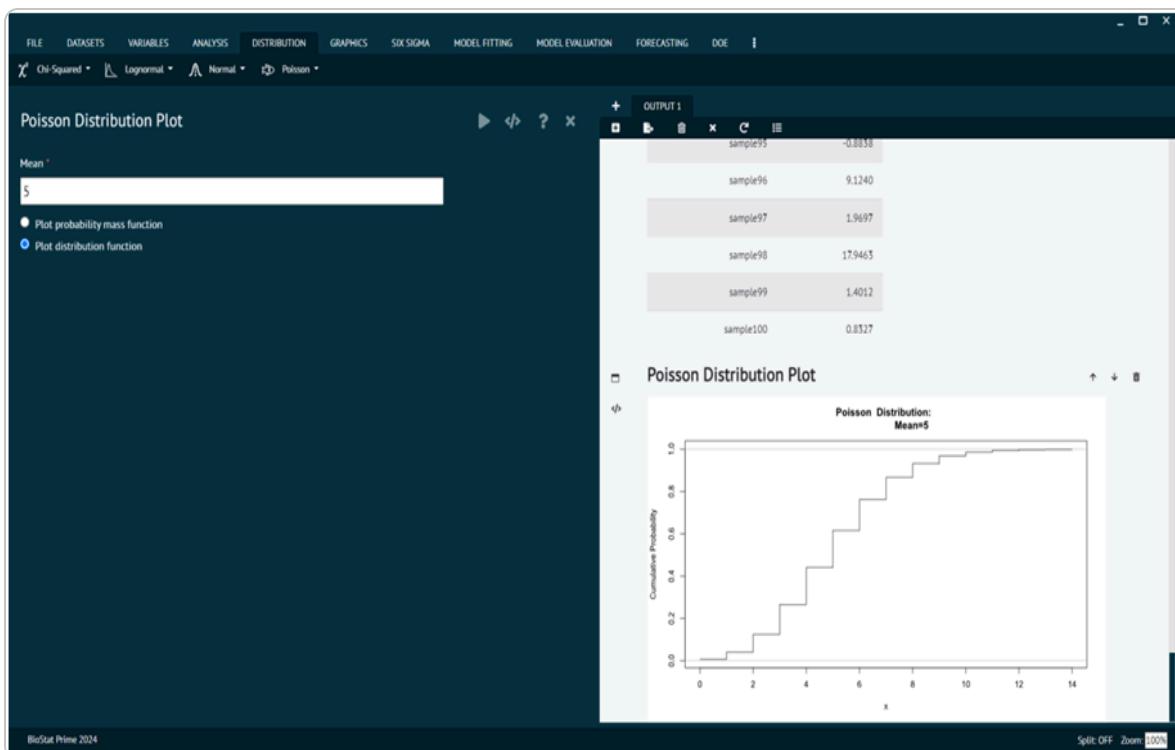
alt text

Poisson

The Poisson distribution is a probability distribution that describes the number of events that occur in a fixed interval of time or space, given a known average rate of occurrence, and assuming that the events occur independently of each other. It serves as a fundamental tool in statistical inference, hypothesis testing, and making predictions about future events based on past observations.

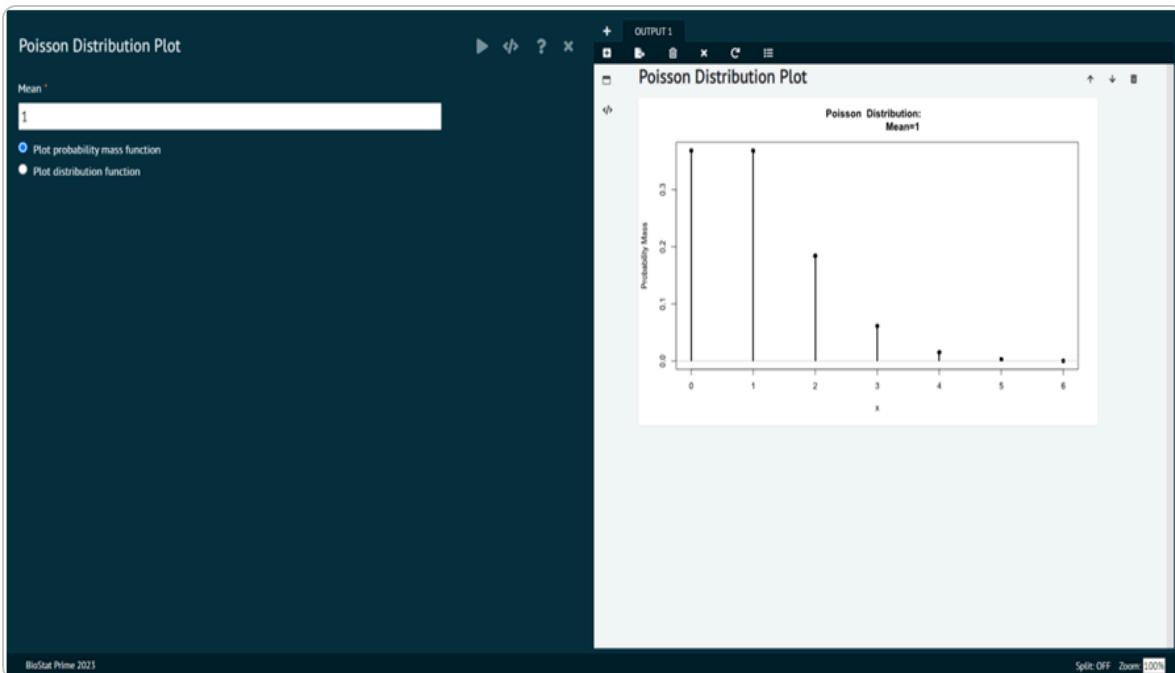
Poisson Distribution Plot

A Poisson distribution plot visually represents the probability distribution of a discrete random variable that represents the number of events occurring in a fixed interval of time or space, given a known average rate of occurrence. Plot distribution function



alt text

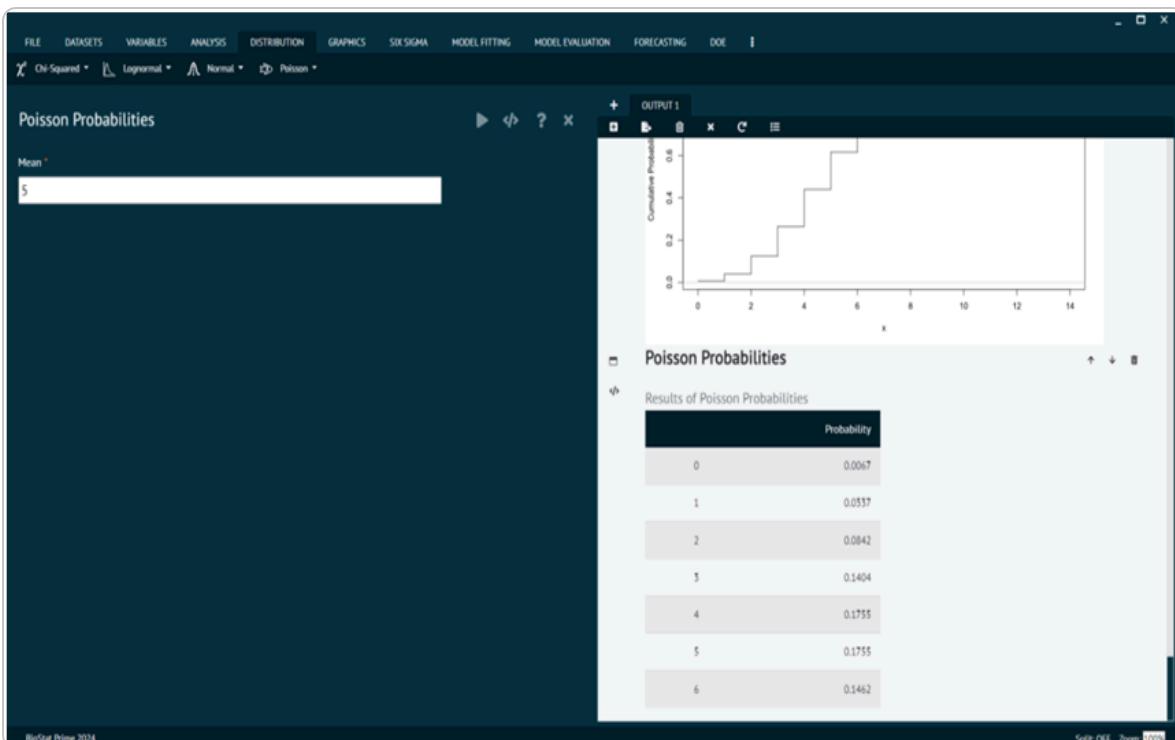
Plot probability mass function



alt text

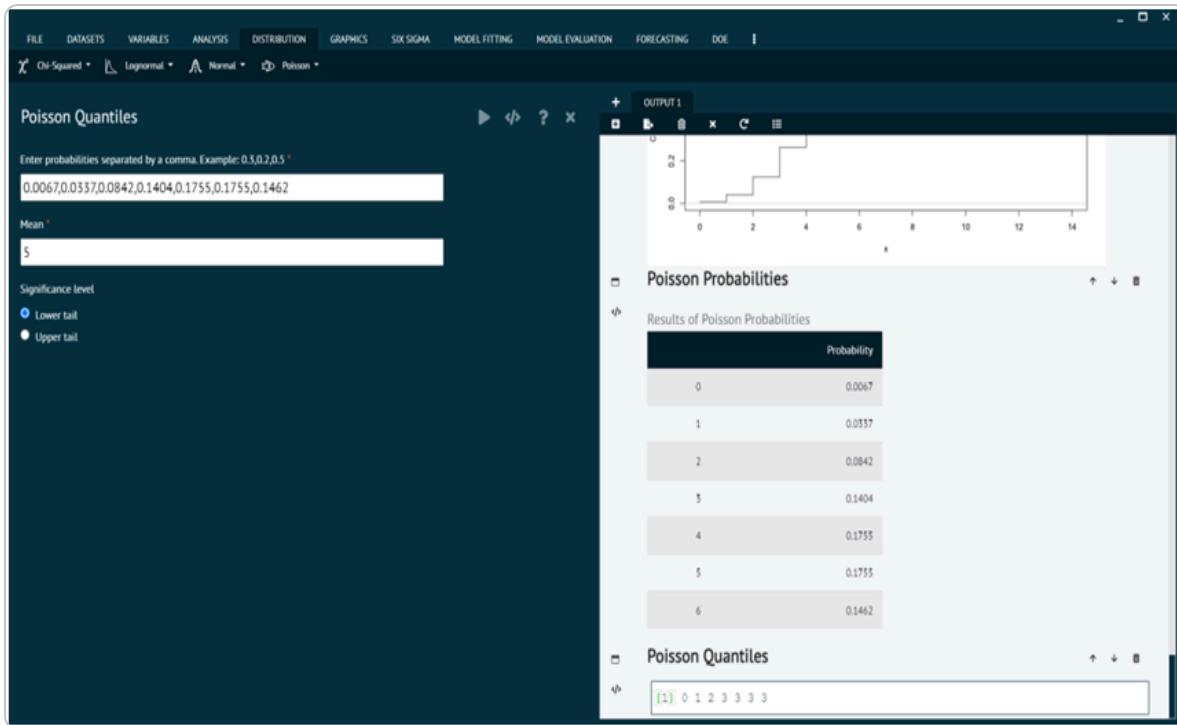
Poisson Probabilities

Poisson's probabilities refer to the probabilities associated with the Poisson distribution.



alt text

Poisson Quantiles



alt text

Graphics

BioStat Prime provide users a variety of high-quality graphs and charts by utilizing the full potential of R language at the backend. R language is known for presenting the best data visualizing plots and BioStat Prime has taken advantage of that and put forth a section called Graphics in its main menu that not only has options for data visualization but also offers customization options for graph appearance, labels, and annotations. Some examples are.

Bar Chart, Box Plot, Contour Plot, AB 2D Contour Plot, Distribution, HeatMap, Line Charts, Maps, Pie Charts, Scatter Plot, Stem and Leaf, Strip Chart, Violin.



alt text

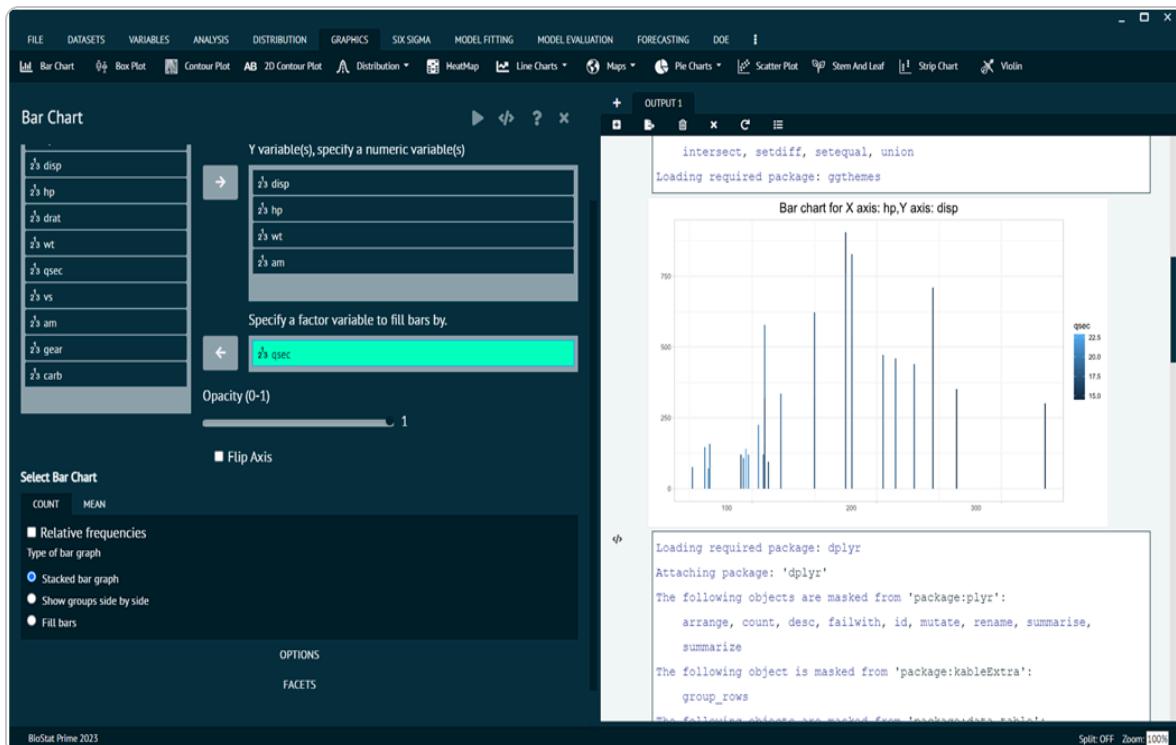
Bar Chart

For representing any dataset in terms of Bar Chart.

Load the dataset that needs to be visualized -> Go to Graphics -> Bar Chart -> Put in the values for variables -> Choose additional options (like variable to fill the bars, opacity, count, etc.) as per the user's requirement -> Execute the dialog.

The Options tab and Facets tab at the bottom can be utilized to add more features to the bar chart in the output.

The picture below shows the bar chart for a loaded dataset and the dialog for the same.

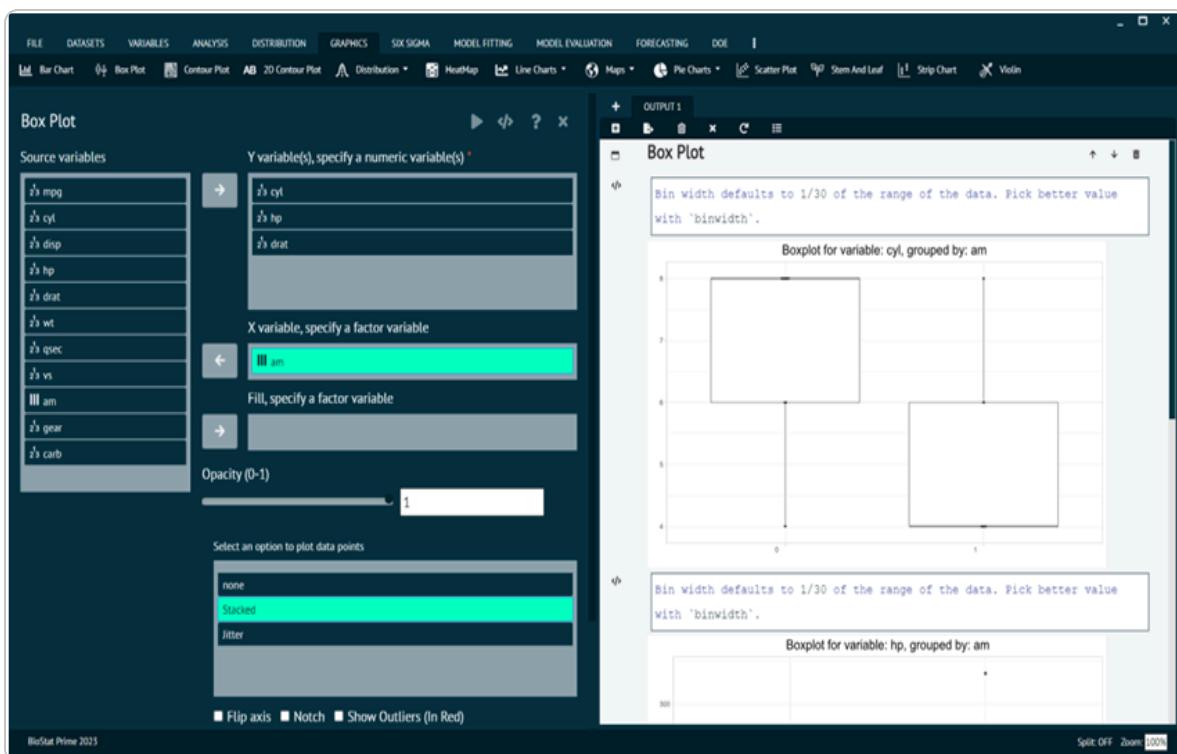


Box Plot

For representing any dataset in terms of Box Plot.

Load the dataset that needs to be visualized -> Go to Graphics -> Box Plot -> Put in the values for variables -> Choose additional options (like opacity, data points, flip axis, etc.) as per the user's requirement -> Execute the dialog.

User can choose multiple numeric values for Y to have a plot for each value of Y with respect to fixed value of X. Also, the value of X needs to be a factor variable. The picture below shows the box plot for a loaded dataset and the dialog for the same.

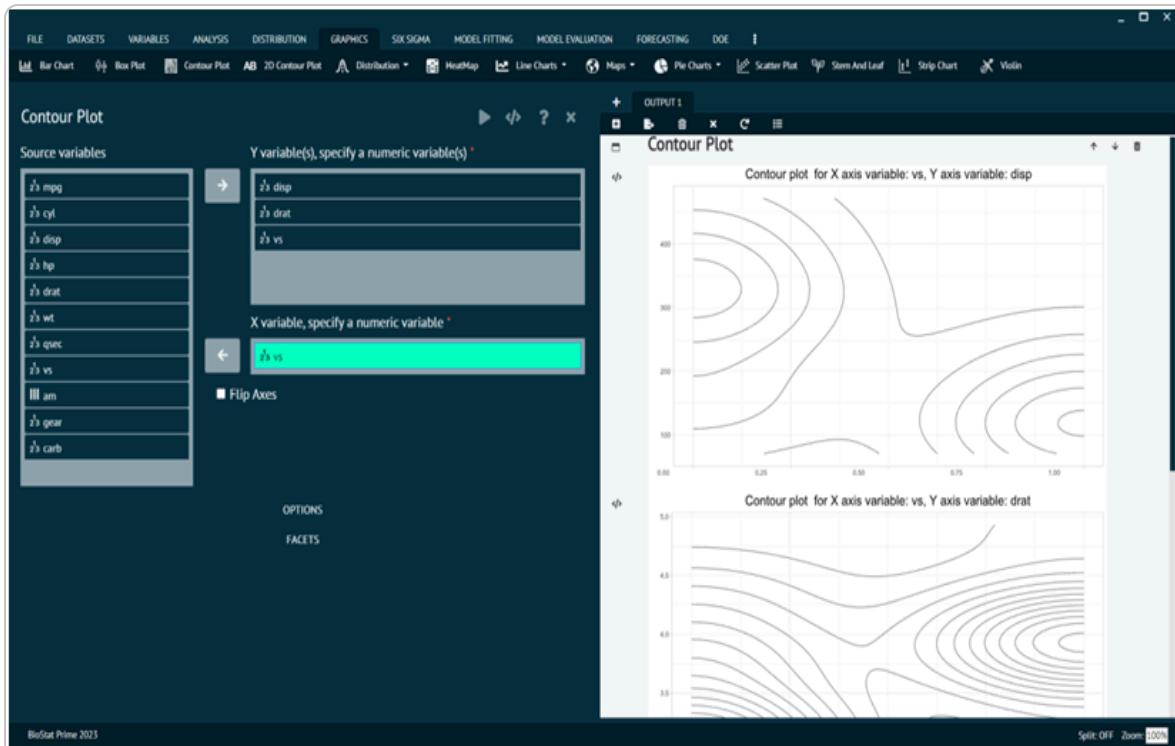


alt text

Contour Plot

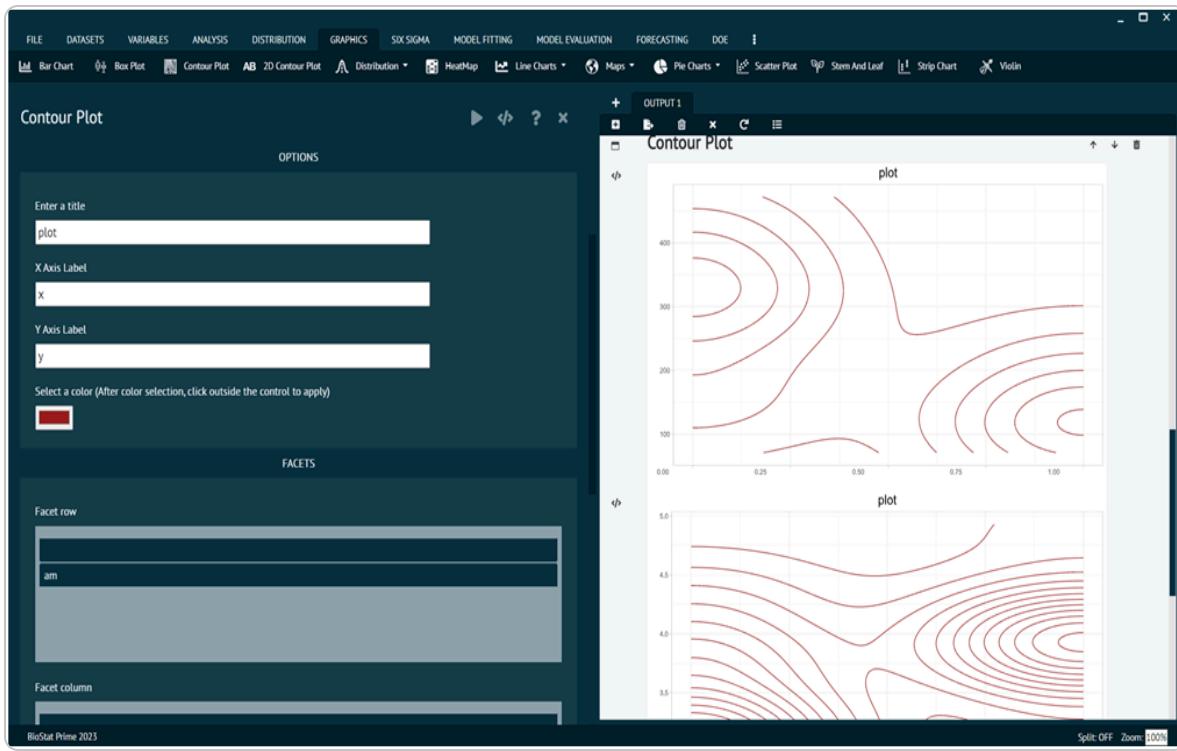
For representing any dataset in terms of Contour Plot.

Load the dataset that needs to be visualized -> Go to Graphics -> Contour Plot -> Put in the values for variables -> Choose additional options (like opacity, data points, flip axis, etc.) as per the user's requirement -> Execute the dialog.



alt text

User can choose multiple numeric values for Y to have a plot for each value of Y with respect to fixed numeric value of X. The Options tab and Facets tab at the bottom can be utilized to add more features to the output as shown below.



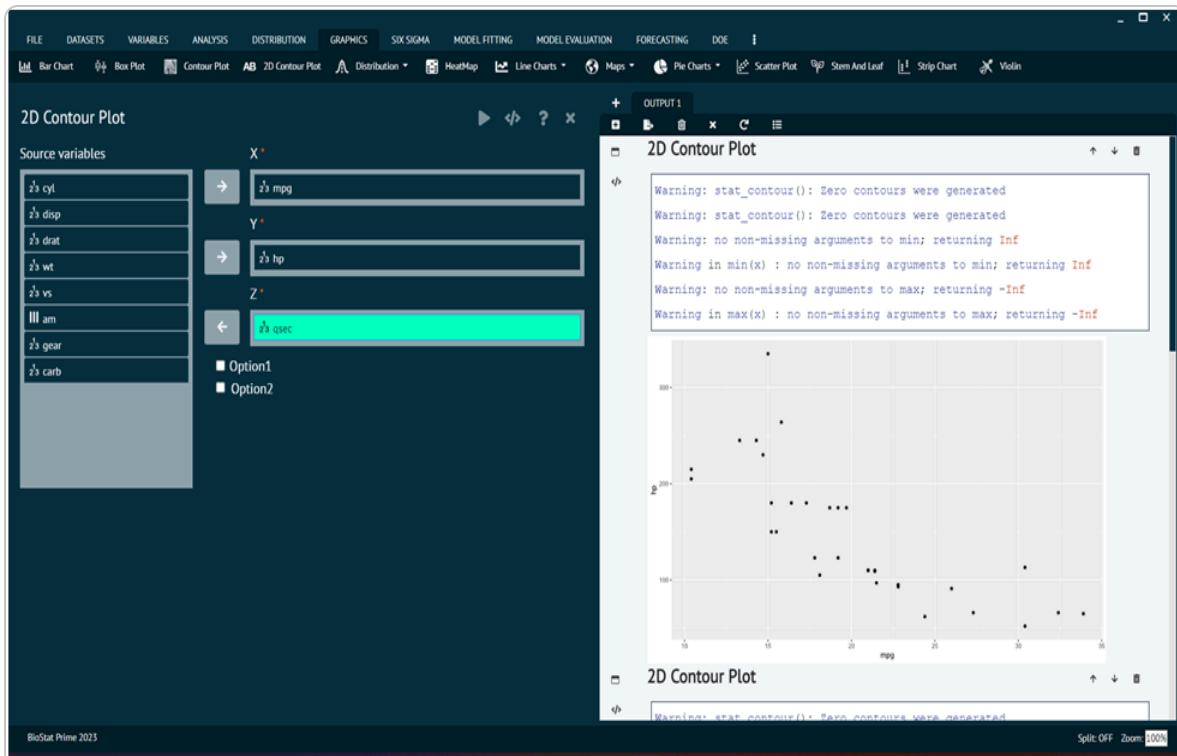
alt text

AB 2D Contour Plot

For representing any dataset in terms of AB 2D Contour Plot.

Load the dataset that needs to be visualized -> Go to Graphics -> AB 2D Contour Plot -> Put in the values for variables -> Execute the dialog.

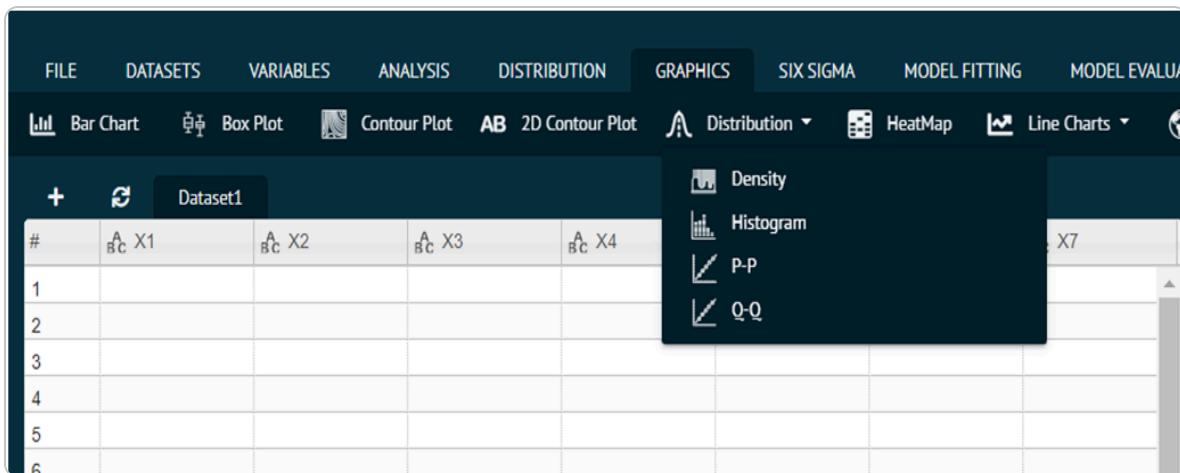
The output of the 2D Contour Plot of a sample dataset can be seen in the picture below.



alt text

Distribution Plot

The distribution tab of graphics menu contains 4 options of data visualization i.e., Density, Histogram, P-P plot, Q-Q plot.



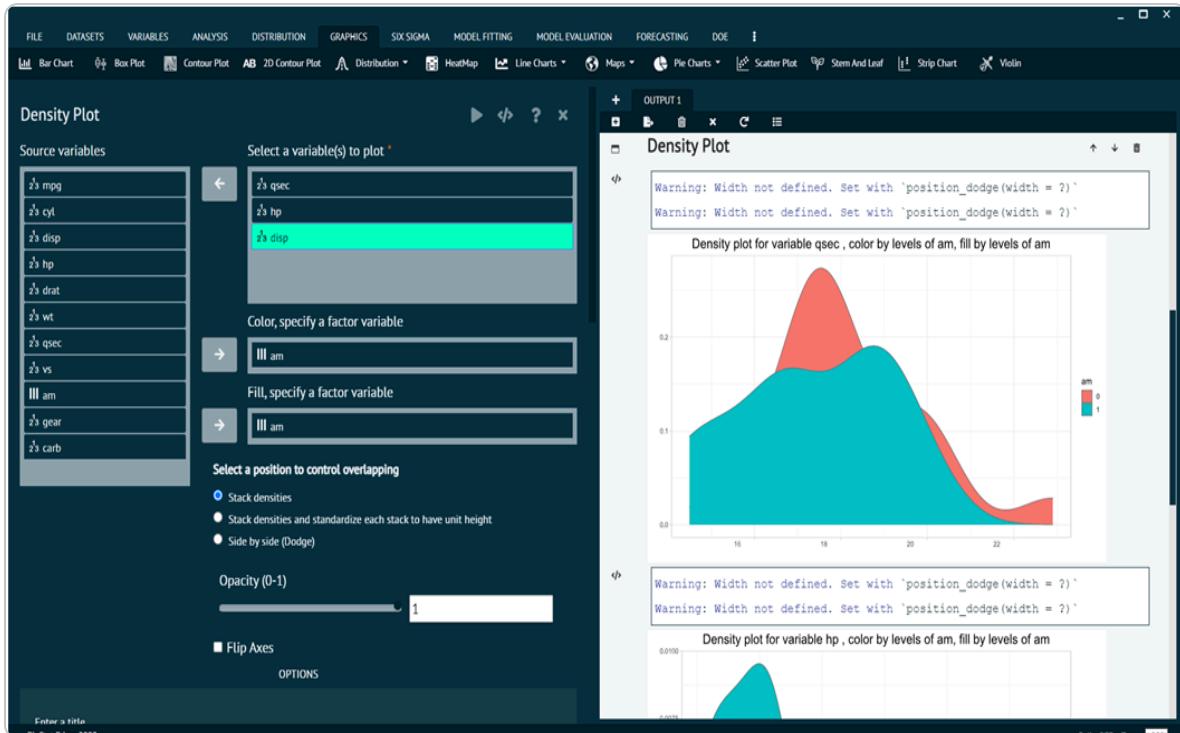
alt text

The function of each option is discussed below.

Density

For representing any dataset in terms of Density plot.

Load the dataset that needs to be visualized -> Go to Graphics -> Distribution -> Density -> Put in the values for variables -> Execute the dialog.



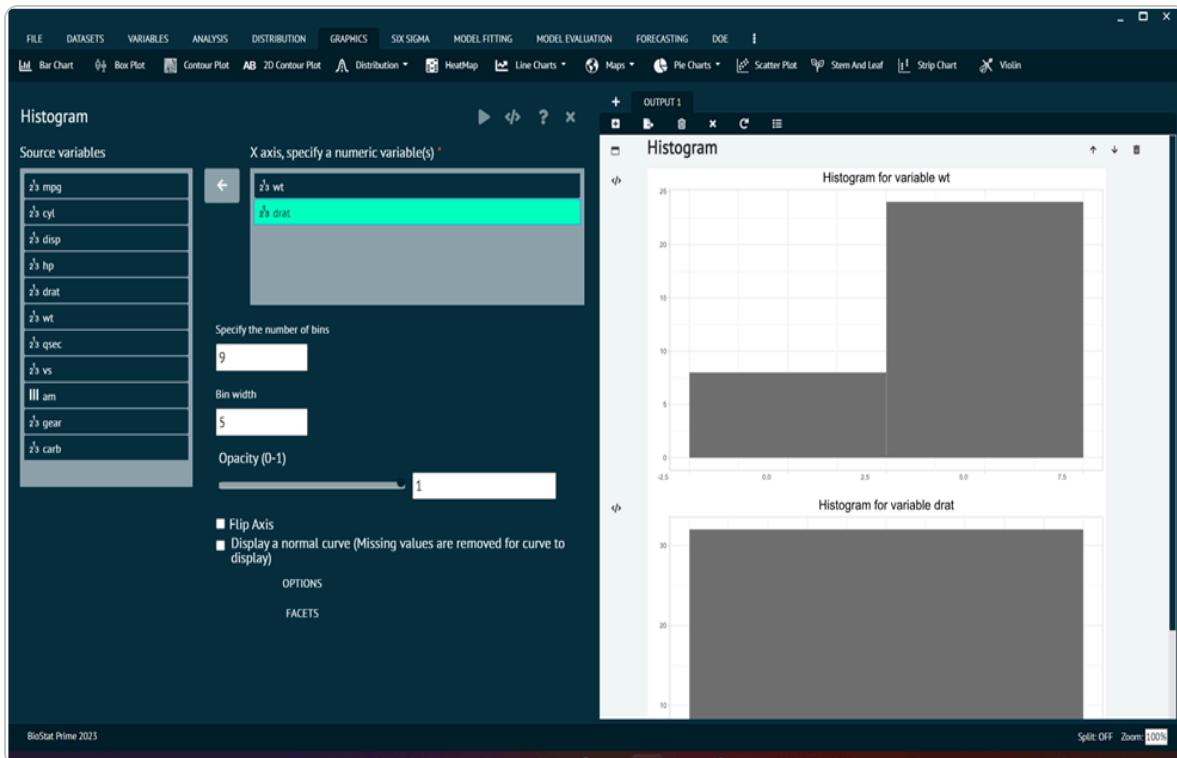
alt text

The output of the Density Plot of a sample dataset can be seen in the picture above. The Options tab and Facets tab at the bottom can be utilized to add more features to the output. User can also select the position to control overlapping, flip axes and opacity of the output.

Histogram

For representing any dataset in terms of Histogram.

Load the dataset that needs to be visualized -> Go to Graphics -> Distribution -> Histogram -> Put in the values for variables -> Execute the dialog.



alt text

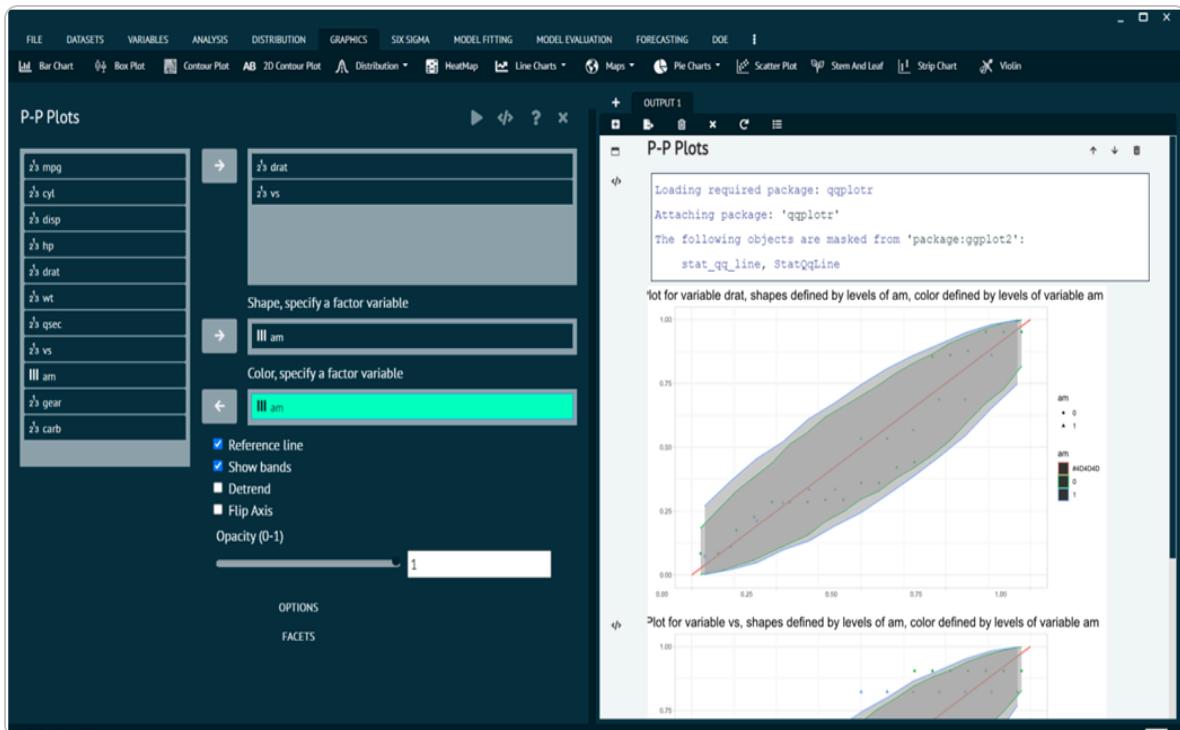
The output of the Histogram of a sample dataset can be seen in the picture above. The Options tab and Facets tab at the bottom can be utilized to add more features to the output. User can also control opacity, flip axes and display normal curve of the output.

PP Plot

For representing any dataset in terms of PP Plot.

Load the dataset that needs to be visualized -> Go to Graphics -> Distribution -> PP -> Put in the values for variables -> Execute the dialog.

The output of the PP Plots of a sample dataset can be seen in the picture below. The Options tab and Facets tab at the bottom can be utilized to add more features to the output. User can also control opacity, flip axes and display reference line or bands or detrend in the output.



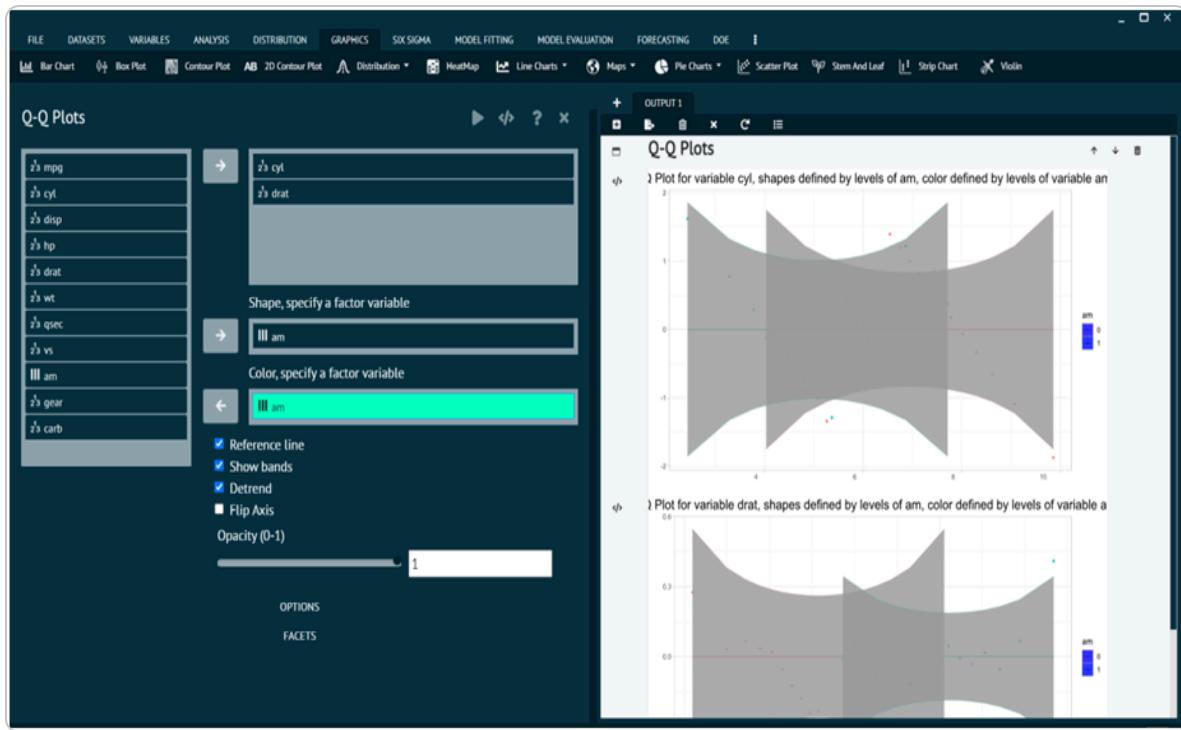
alt text

QQ Plot

For representing any dataset in terms of QQPlot.

Load the dataset that needs to be visualized -> Go to Graphics -> Distribution -> PP -> Put in the values for variables -> Execute the dialog.

The output of the QQ Plots of a sample dataset can be seen in the picture below. The Options tab and Facets tab at the bottom can be utilized to add more features to the output. User can also control opacity, flip axes and display reference line or bands or detrend in the output.

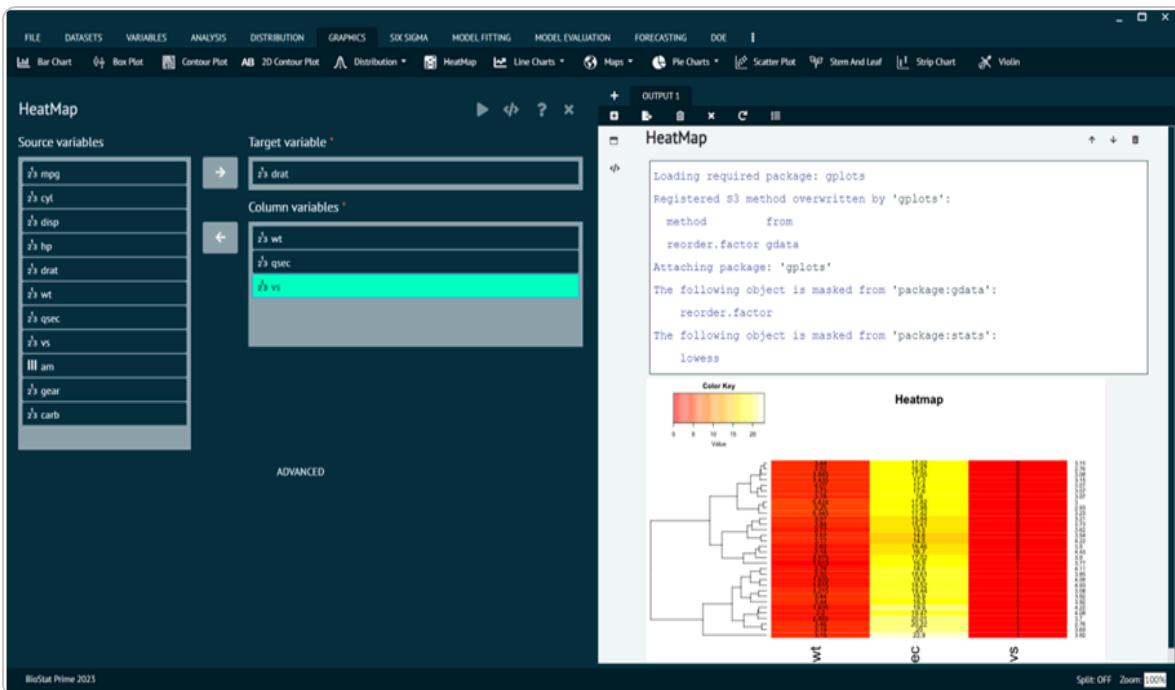


alt text

HeatMap

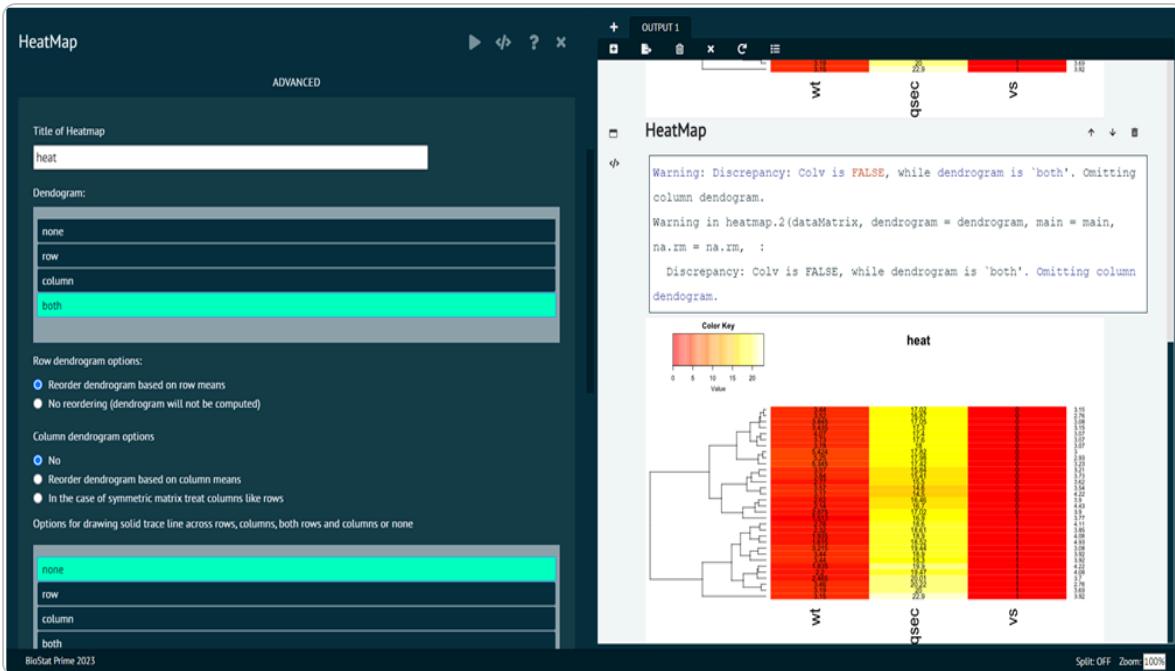
For representing any dataset in terms of HeatMap

Load the dataset that needs to be visualized -> Go to Graphics -> HeatMap -> Put in the values for variables -> Execute the dialog.



alt text

The advanced tab at the bottom leads to some advanced features of the as shown in the picture below.



alt text

Line Charts

The Line Charts tab of graphics menu contains 4 options of data visualization i.e., Frequency Chart, Line Chart, Plot of Means, Two Y Axis.

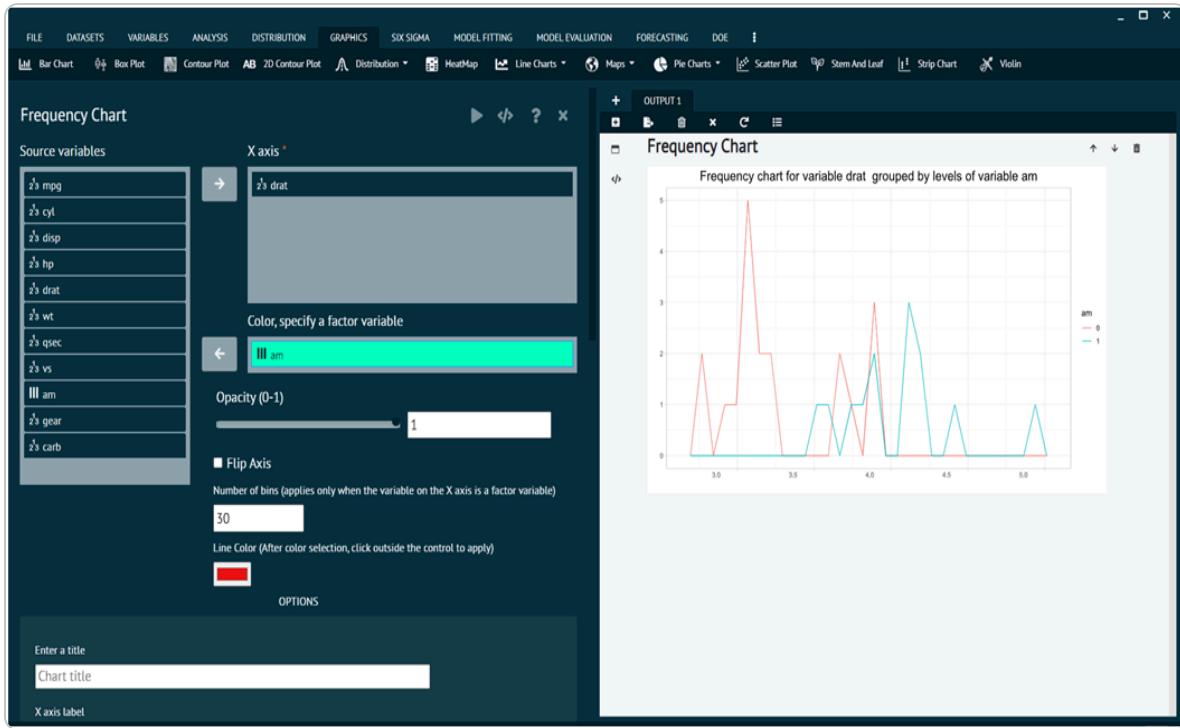


alt text

Frequency Chart

For representing any dataset in terms of Frequency Chart.

Load the dataset that needs to be visualized -> Go to Graphics -> Line Chart \square
Frequencies -> Put in the values for variables -> Execute the dialog.



alt text

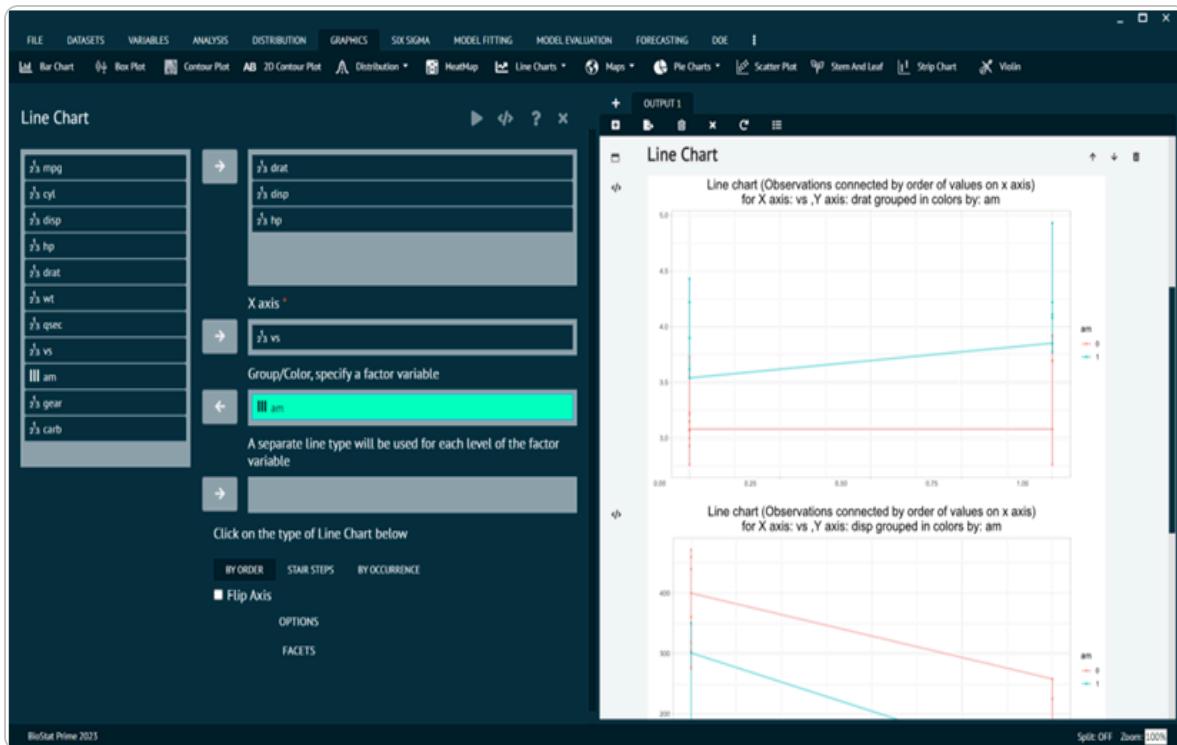
The output of the Frequency Chart of a sample dataset can be seen in the picture above. The Options tab and Facets tab at the bottom can be utilized to add more features to the output. User can also control opacity, flip axes, no. of bins and line colour of the output.

Line Chart

For representing any dataset in terms of Line Chart.

Load the dataset that needs to be visualized -> Go to Graphics -> Line Charts □ Line Chart -> Put in the values for variables -> Execute the dialog.

The output of the Line Chart of a sample dataset can be seen in the picture below. The Options tab and Facets tab at the bottom can be utilized to add more features to the output. User can also flip axes, type of line chart in the output.



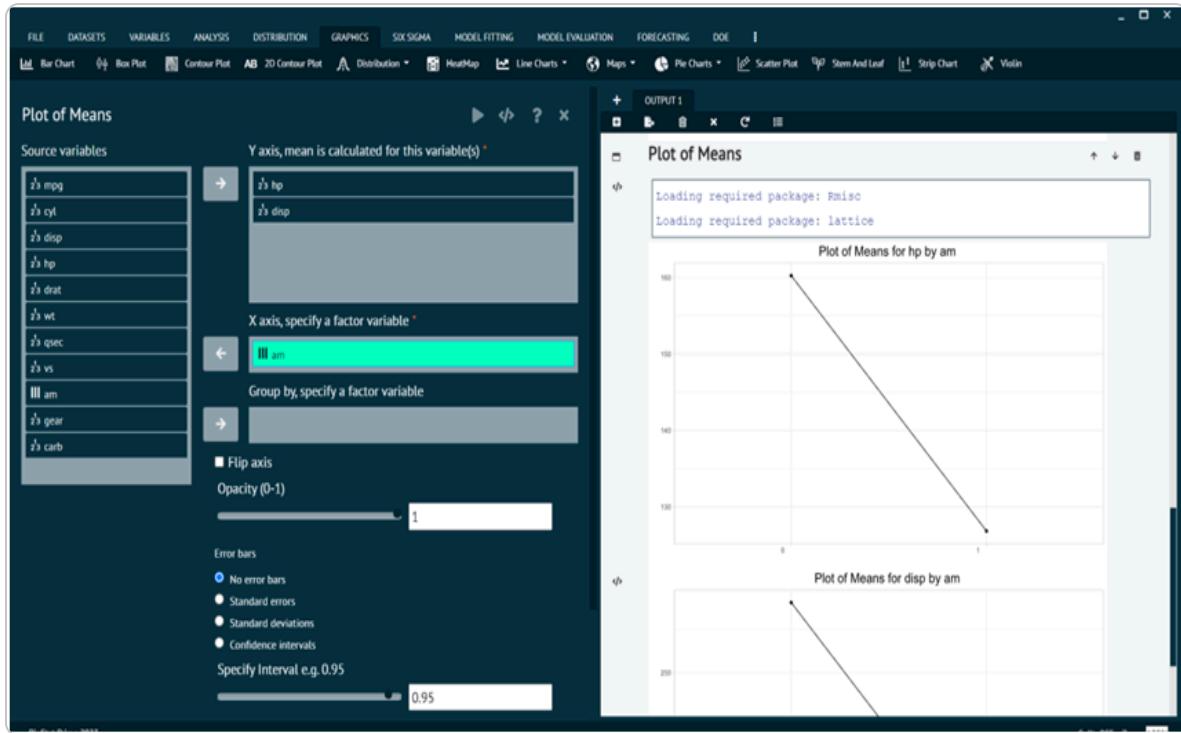
alt text

Plot of Means

For representing any dataset in terms of Plot of Means.

Load the dataset that needs to be visualized -> Go to Graphics -> Line Chart -> Plot ofMeans -> Put in the values for variables -> Execute the dialog.

The output of the Plot of Means a sample dataset can be seen in the picture below. User can also flip axes, Control opacity, error bars, specify intervals for the output.

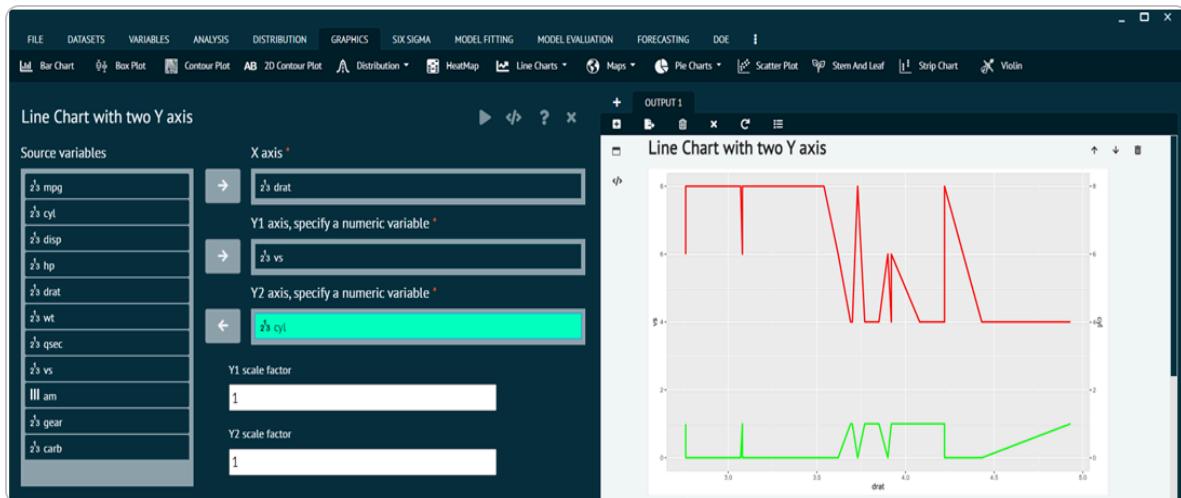


alt text

Two Y axis

For representing any dataset in terms of Line Chart with two Y axis.

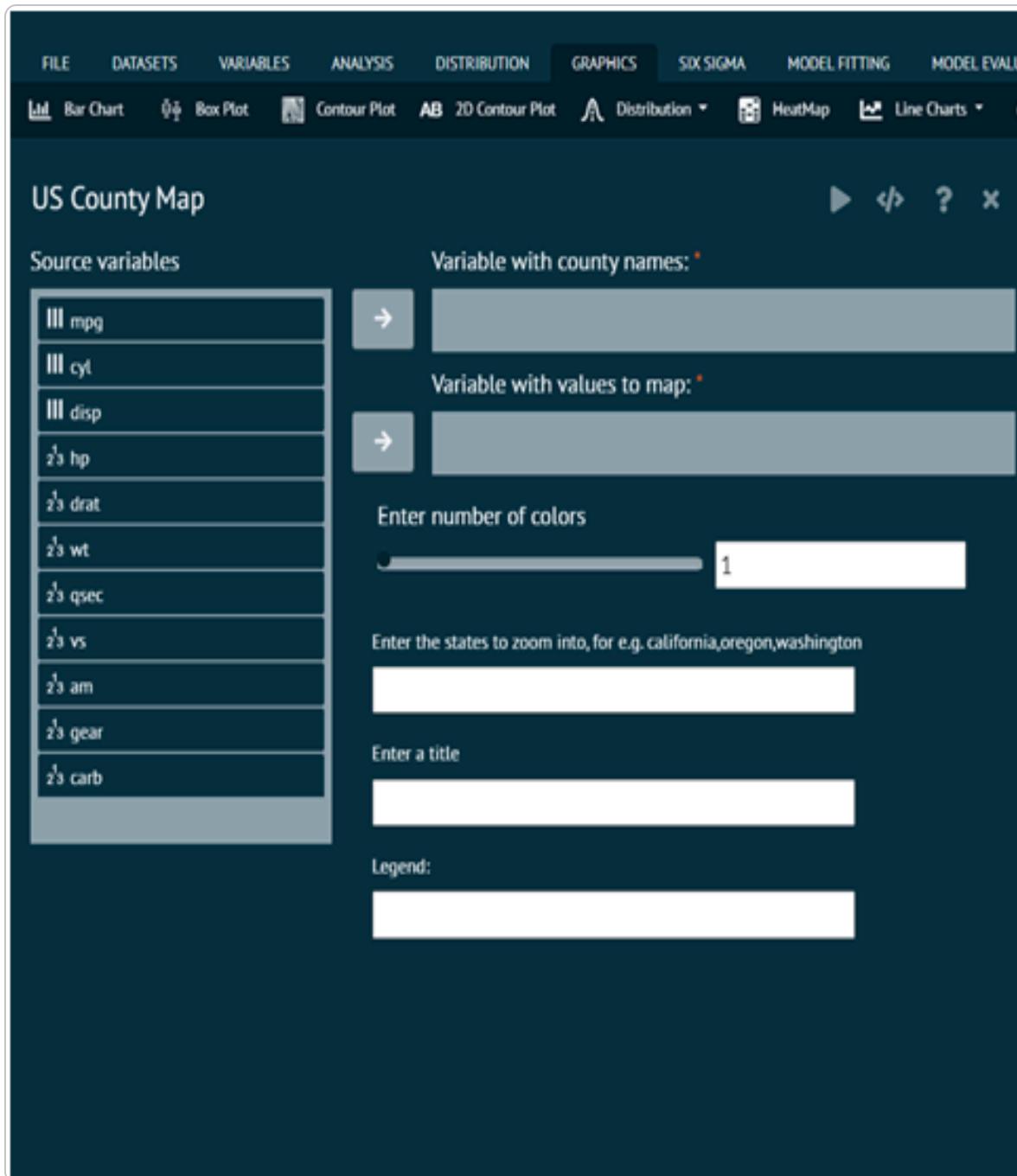
Load the dataset that needs to be visualized -> Go to Graphics -> Line Chart -> Two Y Axis -> Put in the values for variables -> Execute the dialog.



alt text

Maps

This section of graphics tab provides user the ability to visualize maps. Once the appropriate dataset is loaded, user can see a plot for US Country map, US State map, World Map.



alt text

FILE DATASETS VARIABLES ANALYSIS DISTRIBUTION GRAPHICS SIX SIGMA MODEL FITTING MODEL EVALUATION

Bar Chart Box Plot Contour Plot AB 2D Contour Plot Distribution HeatMap Line Charts

US State Map

Source variables

- mpg
- cyl
- disp
- hp
- drat
- wt
- qsec
- vs
- am
- gear
- carb

Variable with US State names:

Variable with values to map:

Enter number of colors: 1

Enter the states to zoom into, for e.g. california,oregon,washington

Enter a title

Legend:

alt text

FILE DATASETS VARIABLES ANALYSIS DISTRIBUTION GRAPHICS SIX SIGMA MODEL FITTING MODEL EVALUATION

Bar Chart Box Plot Contour Plot AB 2D Contour Plot Distribution HeatMap Line Charts

World Map

Source variables

- mpg
- cyl
- disp
- hp
- drat
- wt
- qsec
- vs
- am
- gear
- carb

Variable with country names: *

Variable with values to map: *

Enter number of colors 1

Enter the countries to zoom into, for e.g. united states of america, canada, mexico

Enter a title

Legend:

alt text

Pie Charts

The Pie Charts tab of graphics menu contains 2 options of data visualization i.e., Coxcomb plot, PIE Chart.

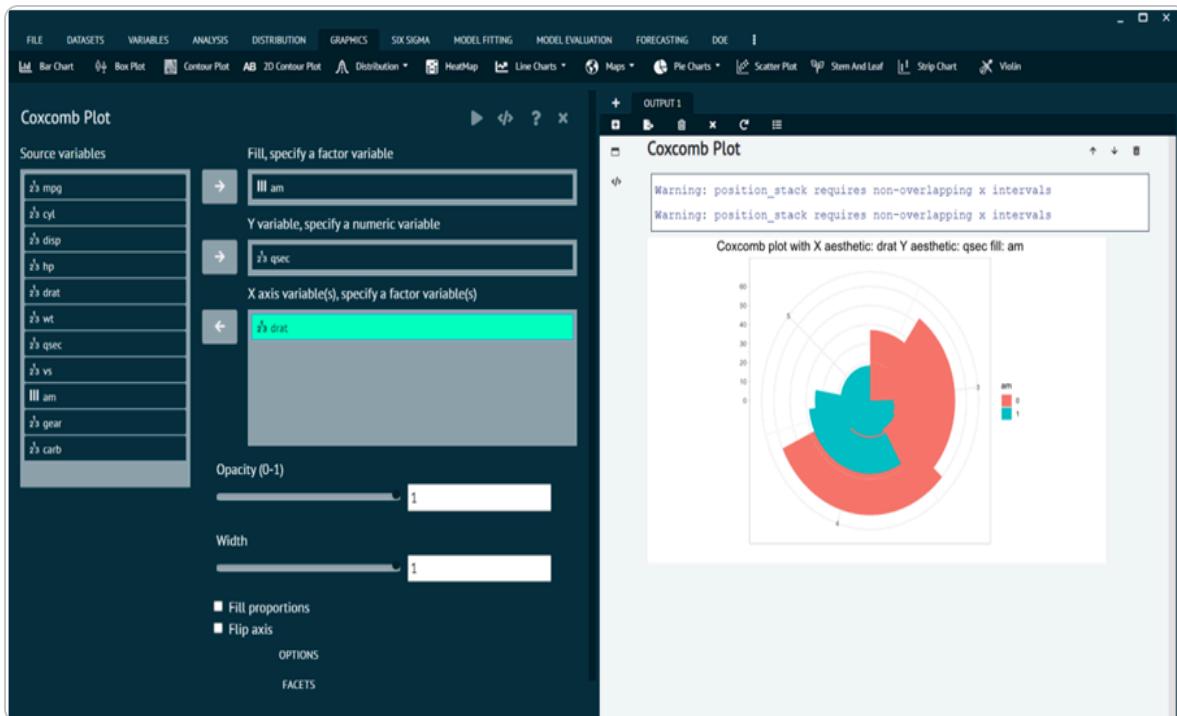
Coxcomb Plot

For representing any dataset in terms of Coxcomb Plot.

Load the dataset that needs to be visualized -> Go to Graphics -> Pie Charts -> Coxcomb Plot -> Put in the values for variables -> Execute the dialog.

The output of the Coxcomb Plot of a sample dataset can be seen in the picture below.

The Options tab and Facets tab at the bottom can be utilized to add more features to the output. User can also flip axis, fill proportions, control the opacity, width of the pie chart in the output.

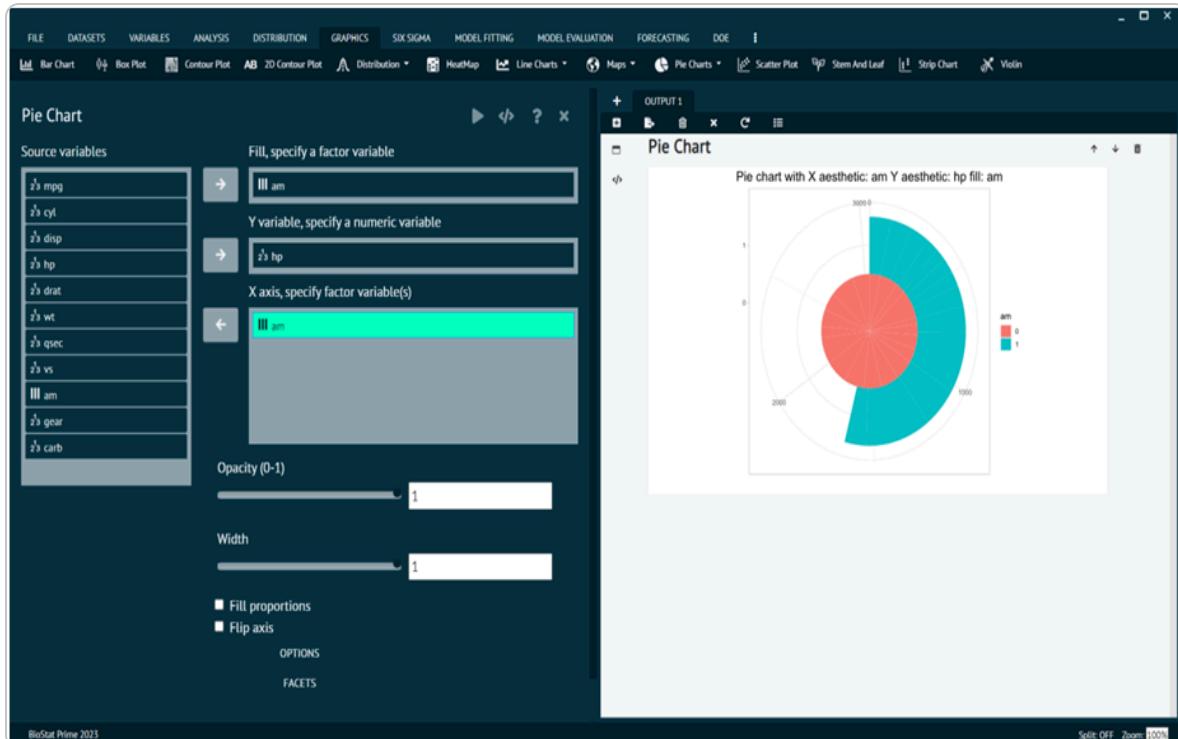


alt text

Pie Chart

For representing any dataset in terms of Pie Chart.

Load the dataset that needs to be visualized -> Go to Graphics -> Pie Charts -> Pie Chart -> Put in the values for variables -> Execute the dialog.



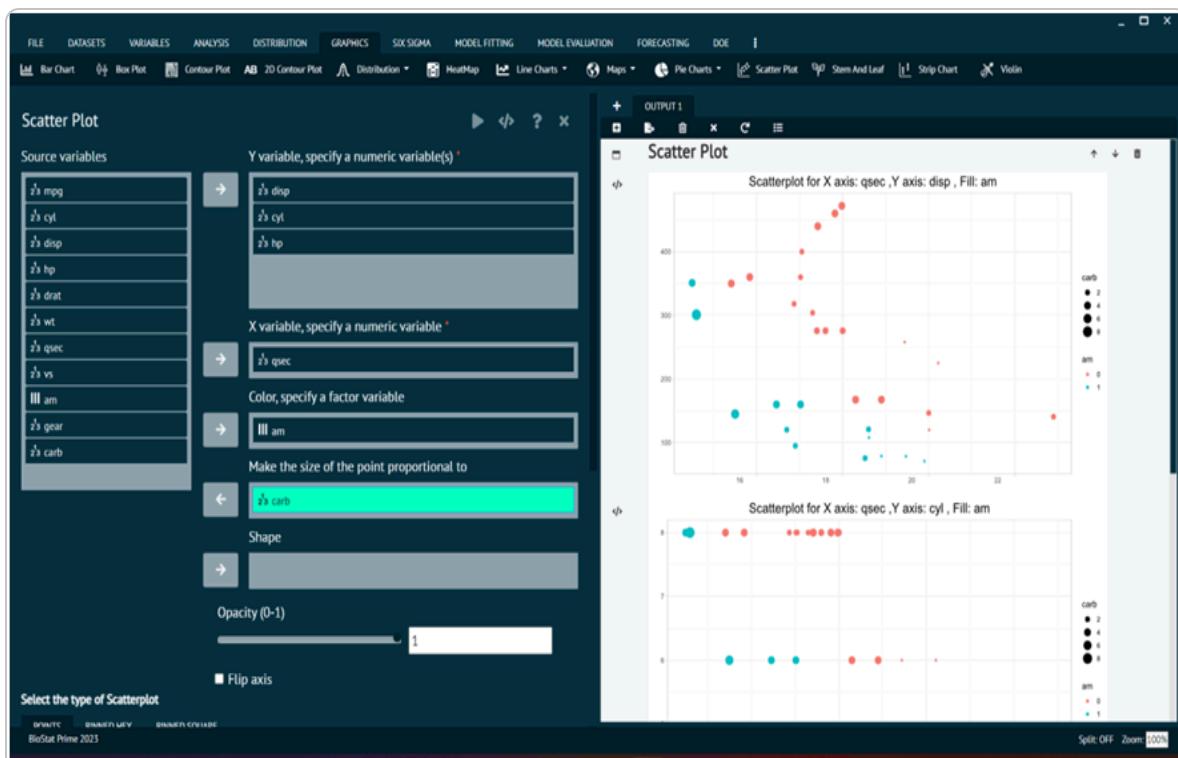
The output of the Pie Chart of a sample dataset can be seen in the picture above. The Options tab and Facets tab at the bottom can be utilized to add more features to the output. User can also flip axis, fill proportions, control the opacity, width of the pie chart in the output.

Scatter plots

For representing any dataset in terms of Scatter Plot.

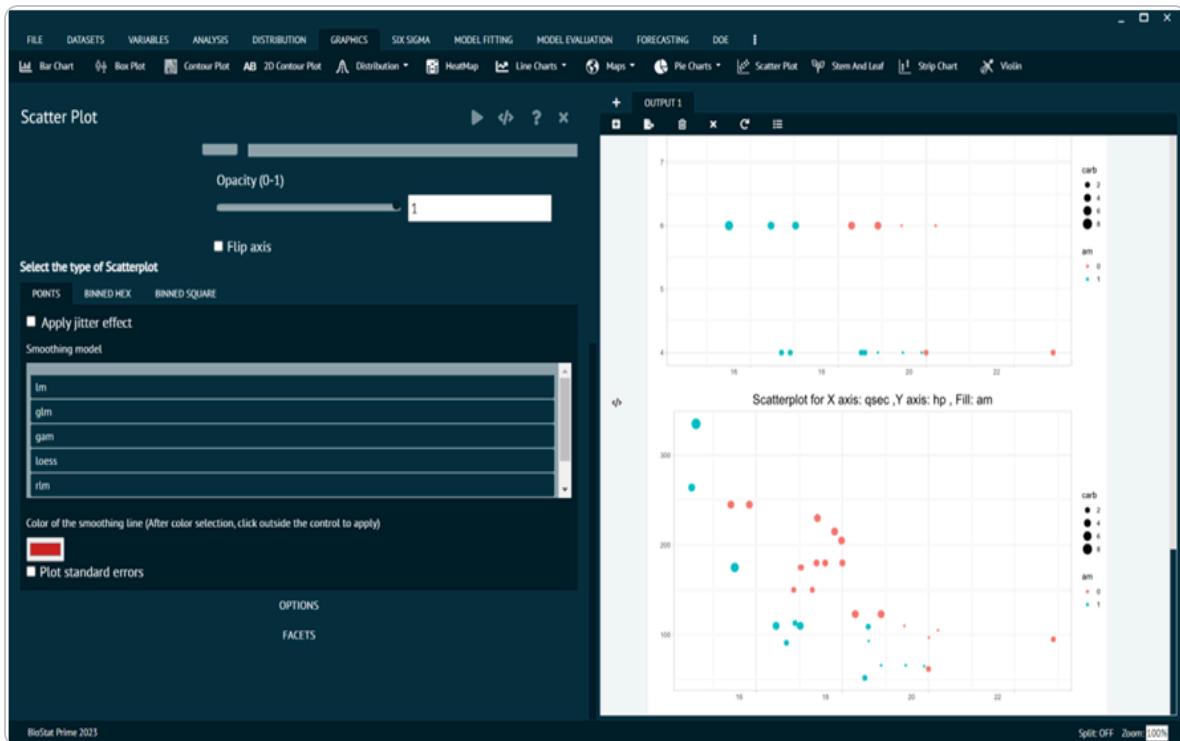
Load the dataset that needs to be visualized -> Go to Graphics -> Scatter Plots -> Put in the values for variables -> Execute the dialog.

The output of the Scatter Plot of a sample dataset can be seen in the picture below. User can also flip axis, control the opacity of the plot in the output.



alt text

The Select type of Scatter plot tab at the bottom can be utilized to add more features to the output as shown below.



alt text

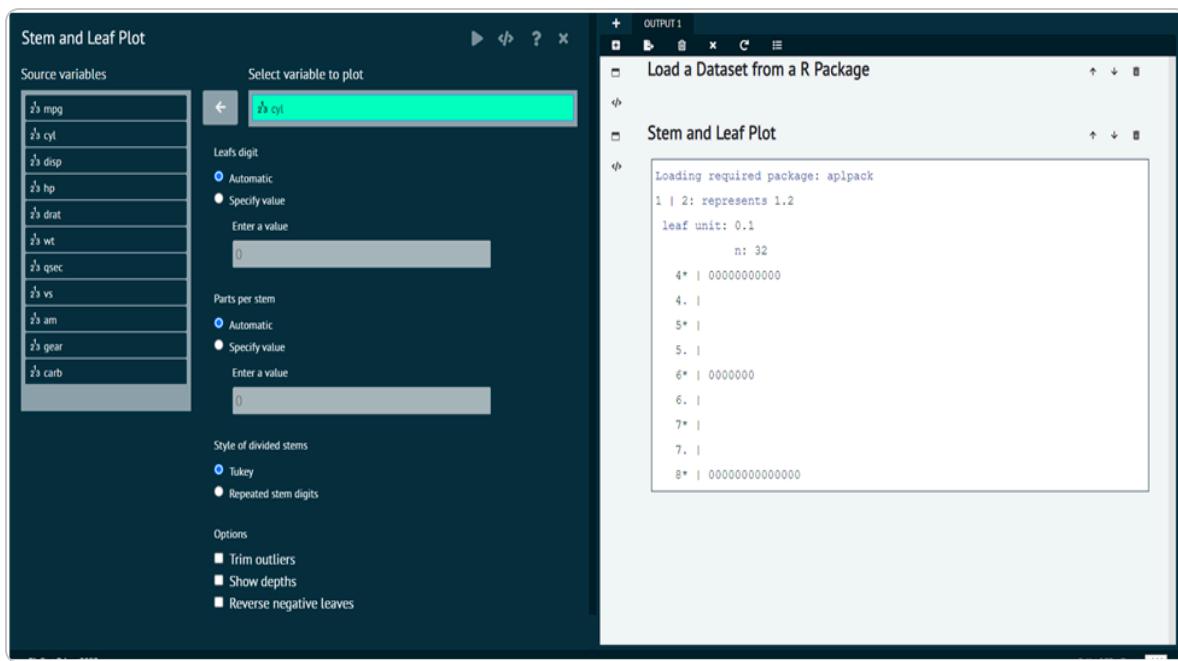
Stem And Leaf

For representing any dataset in terms of Stem and Leaf.

Load the dataset that needs to be visualized -> Go to Graphics → Stem and Leaf -> Put in the values for variables -> Execute the dialog.

The output of the Stem and Leaf of a sample dataset can be seen in the picture below.

User can also Trim the outlines, show depths, Reverse negative leaves.



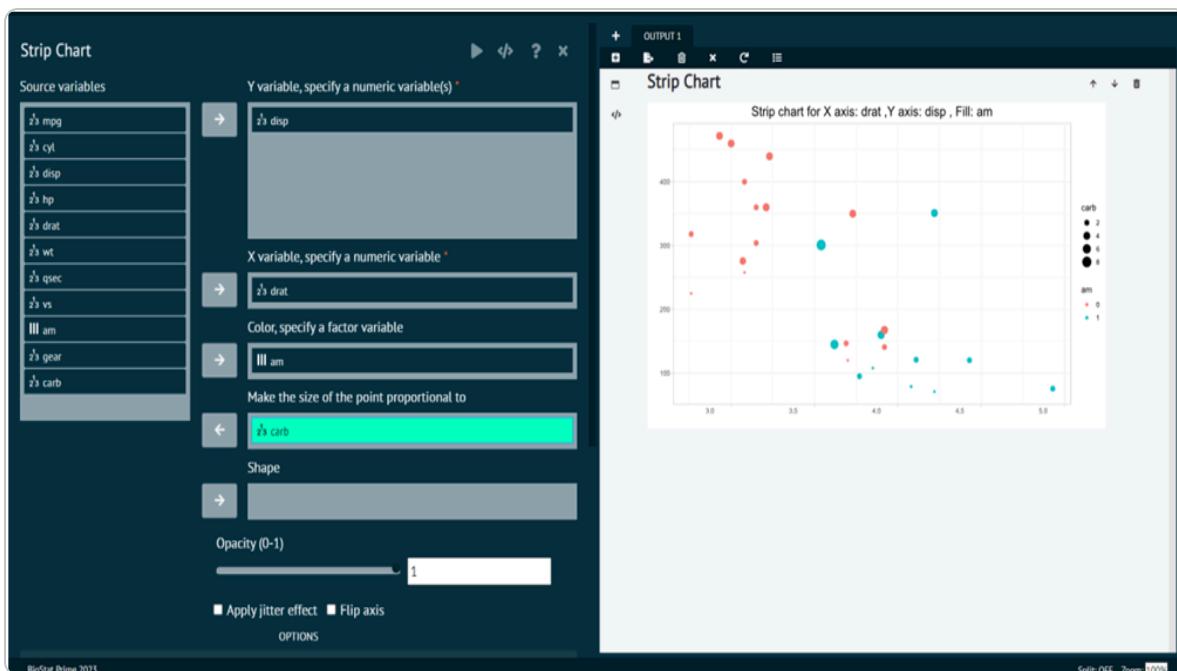
alt text

Strip Chart

For representing any dataset in terms of Scatter Plot.

Load the dataset that needs to be visualized -> Go to Graphics -> Strip Chart -> Put in the values for variables -> Execute the dialog.

The output of the Scatter Plot of a sample dataset can be seen in the picture below. User can also flip axis, apply jitter effect, control the opacity of the plot in the output.

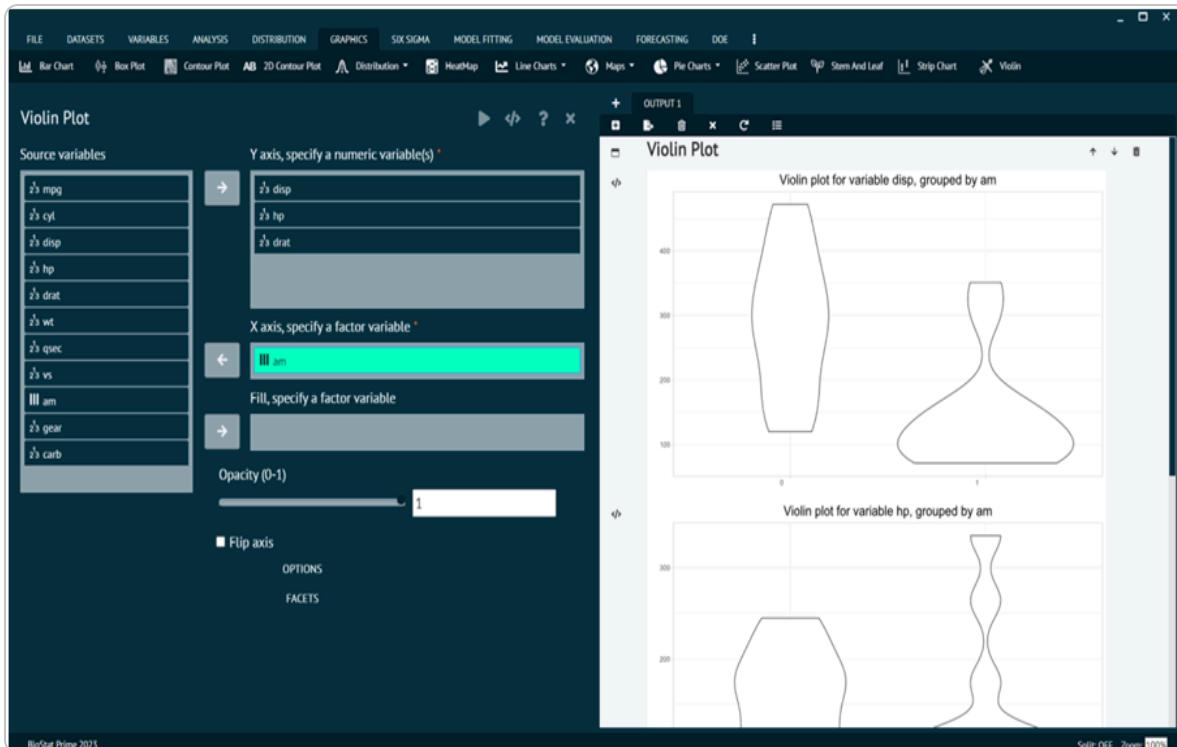


alt text

Violin

For representing any dataset in terms of Violin.

Load the dataset that needs to be visualized -> Go to Graphics -> Violin -> Put in the values for variables -> Execute the dialog.



alt text

Six Sigma

Six Sigma is a rigorous, focused and highly effective implementation of proven quality principles and techniques. Incorporating elements from the work of many quality pioneers, Six Sigma aims for virtually error free business performance. A very powerful feature of Six Sigma is the creation of an infrastructure to assure that performance improvement activities have the necessary resources.

Six Sigma Overview

Six Sigma Overview can be utilized by user to get a complete guide to Six Sigma. It guides the user to various resources that helps the user to understand Six Sigma to its full potential.

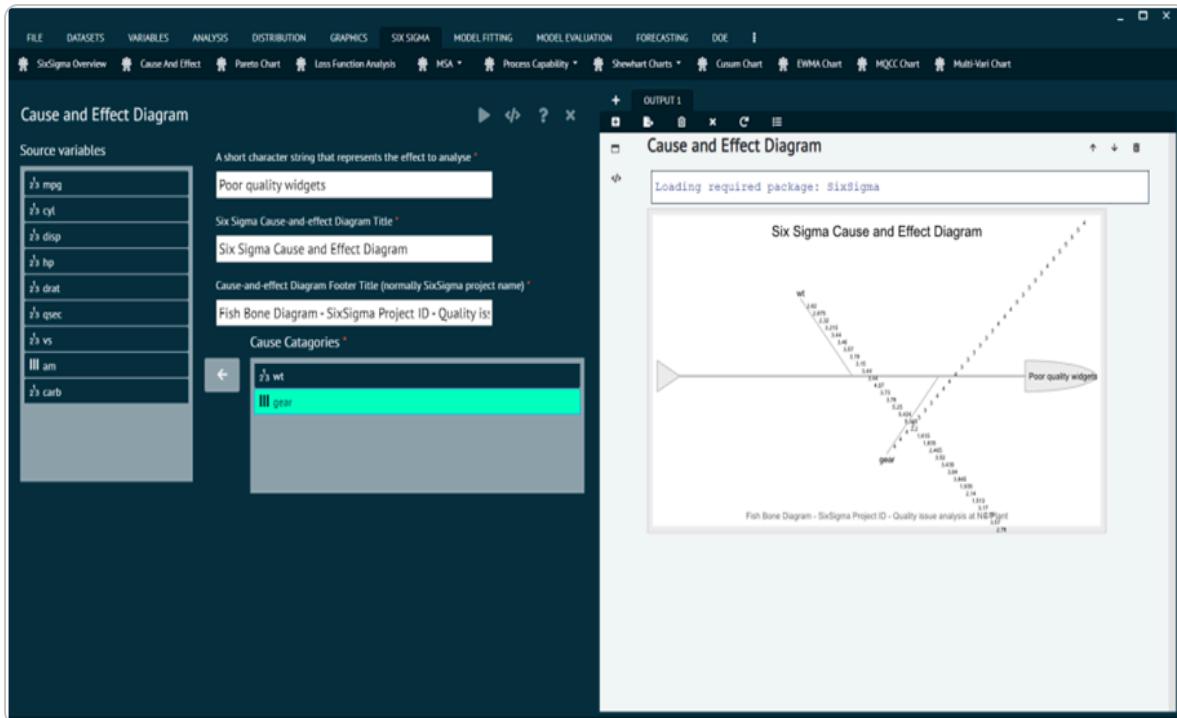
The screenshot shows the BioStat Prime 2023 software interface. At the top, there is a navigation bar with tabs: FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, and MODEL EVALUATION. The 'SIX SIGMA' tab is highlighted with a red circle. Below the navigation bar, there is a toolbar with icons for 'SixSigma Overview' (circled in red), 'Pareto Chart', and others. The main content area displays a 'Six Sigma reference tutorial by Thomas Pyzdek (Pyzdek Institute)'. This section includes a close button 'x'. Below this, there is a heading 'Pyzdek Institute's Six Sigma tutorial' followed by a bulleted list: '• What is Six Sigma and How Does It Work?'. Next is a heading 'R Package (QCC) - quality control charting tutorial' with a bulleted list: '• R quality control charting tutorial'. Finally, there is a heading 'Additional Six Sigma tutorial found on the internet (based on R SixSigma Package)' with a bulleted list: '• Part-1', '• Part-2', '• Part-3', '• Part-4', and '• Part-5'. At the bottom left of the content area, it says 'BioStat Prime 2023'.

alt text

Cause and Effect

To analyse Cause and Effect in BioStat user must follow the steps given below.

Load the dataset -> Click on the Six Sigma tab in main menu -> Select Cause and Effect -> This leads to analysis techniques in the dialog -> Select the cause categories from source variables -> Execute and visualise the output in output window.



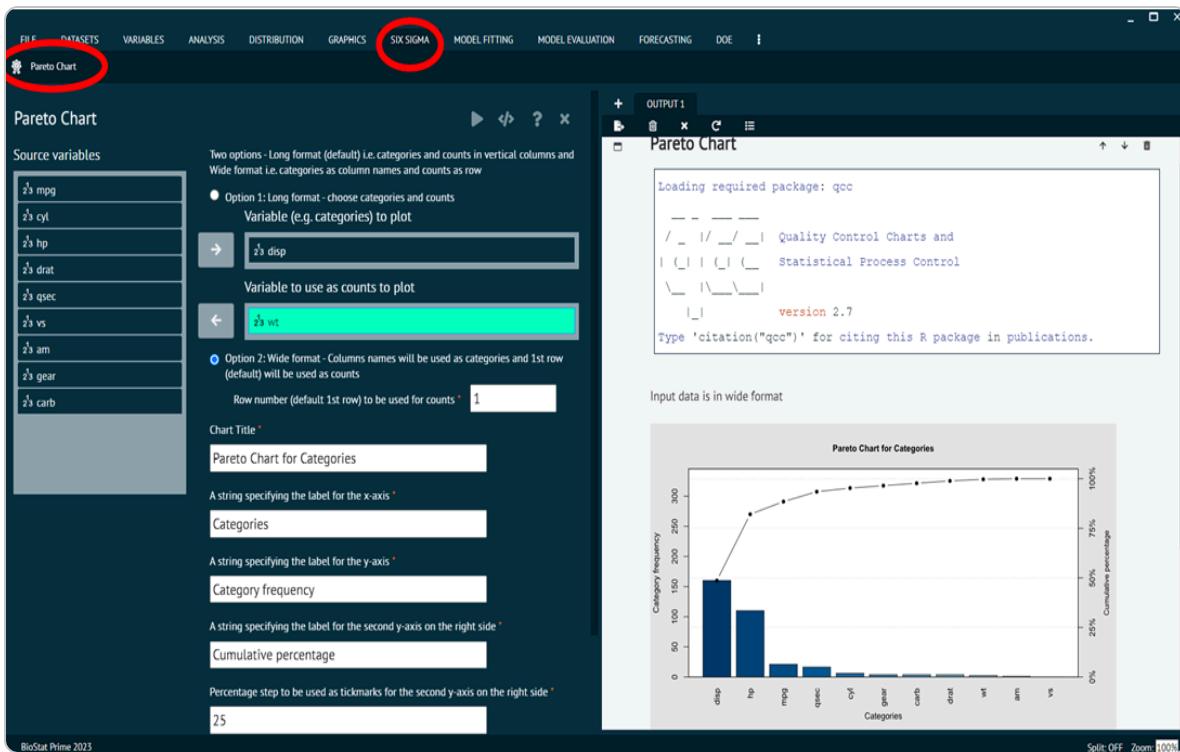
alt text

Pareto Chart

A Pareto chart is a specific type of chart used in statistics that combines both bar and line charts. Pareto chart is designed to highlight the most important factors among a set of variables. The chart is based on the Pareto principle, which states that, for many phenomena, roughly 80% of the effects come from 20% of the causes. In a Pareto chart, the bars represent individual categories or factors, and they are arranged in descending order from left to right. The cumulative percentage of the total is represented by a line.

To analyse Pareto Chart in BioStat user must follow the steps given below.

Load the dataset -> Click on the Six Sigma tab in main menu -> Select Pareto Chart -> This leads to analysis techniques in the dialog -> Selected the various options in the dialog according to the requirement -> Execute and visualise the output in output window.

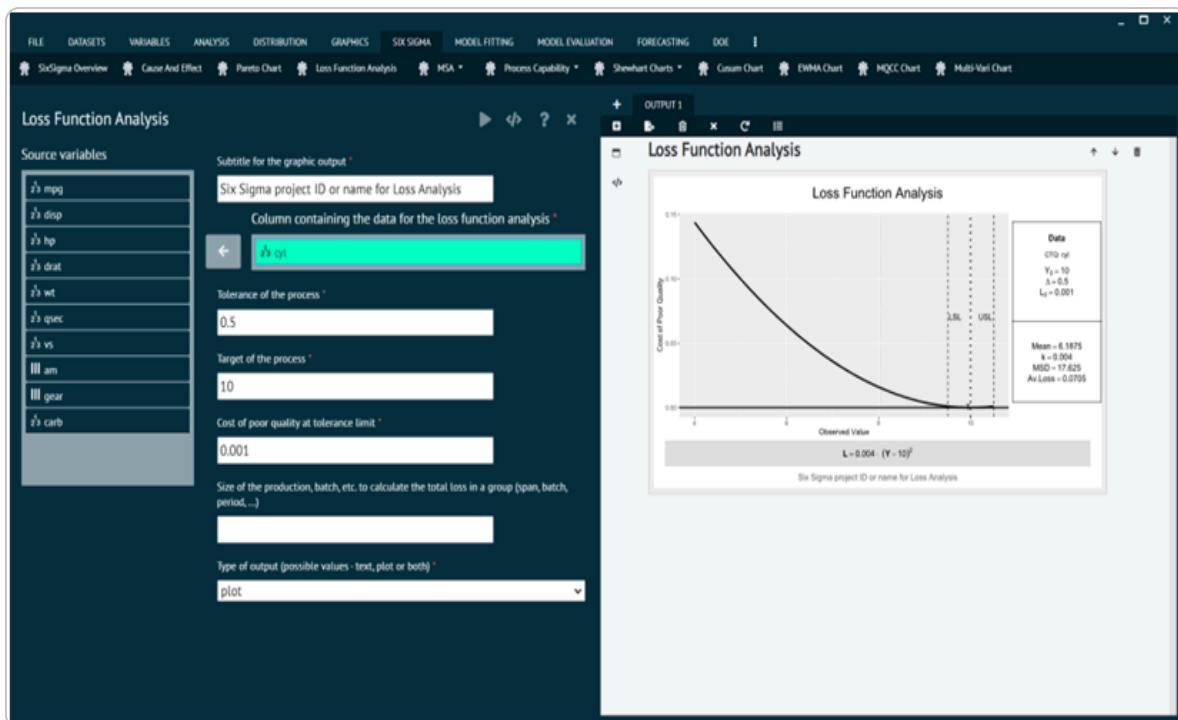


alt text

Loss Function Analysis

To analyse Loss Function Analysis in BioStat user must follow the steps given below.

Load the dataset -> Click on the Six Sigma tab in main menu -> Select Loss Function Analysis -> This leads to analysis techniques in the dialog -> Selected the various options in the dialog according to the requirement -> Execute and visualise the output in output window.



alt text

MSA (Measurement System Analysis)

Measurement System Analysis (MSA) is a statistical method used to assess and ensure the reliability and accuracy of a measurement system. The goal of MSA is to identify and quantify sources of variation within a measurement process. The results of MSA can be used to improve measurement processes, reduce variability, and enhance the overall quality of data in a particular system. It is a fundamental step in ensuring the reliability of data in various applications, ultimately contributing to improved decision-making and quality control.

Gage R&R-Measurement System Analysis

This method assesses the variation in measurements due to operators (appraisers) and equipment (gages). It helps distinguish between variability introduced by the measurement system and the actual variation in the process.

To analyse in Gage R&R-Measurement System Analysis BioStat user must follow the steps given below.

Load the dataset -> Click on the Six Sigma tab in main menu -> Select MSA -> Choose Gage R&R-Measurement System Analysis -> This leads to analysis techniques in the dialog -> selected the various options in the dialog according to the requirement -> Execute and visualise the output in output window.

Gage R&R - Measurement System Analysis

▶ ⌂ ? ×

Source variables

13 mpg
13 cyl
13 disp
13 hp
13 drat
13 wt
13 qsec
13 vs
13 am
13 gear
13 carb

Main title for the graphic output *

Six Sigma Gage R&R Study

Subtitle for the graphic output (e.g. the name of the SixSigma project) *

Six Sigma project ID or name for Gage R&R Study

Measured variable *



Part variable *



Appraiser (operators, machines, ...) variable *



LSL - numeric value of lower specification limit used with USL to calculate Study Variation as %Tolerance

USL - numeric value of upper specification limit used with LSL to calculate Study Variation as %Tolerance

Tolerance - numeric value for the tolerance - default (usl - lsl)

StDev - numeric value for number of std deviations to use in calculating Study

alt text

Attribute Agreement Analysis

Attribute Agreement Analysis is a specific method within Measurement System Analysis (MSA) that focuses on assessing the agreement or reliability of categorical or attribute data among different appraisers. This analysis is particularly useful when the measurement system involves subjective judgments or classifications, such as visual inspections, quality ratings, or pass/fail decisions. The primary objective of Attribute Agreement Analysis is to quantify the level of agreement or disagreement between different individuals or appraisers when making judgments about the same set of items. This helps identify sources of variability in the measurement process that may be attributed to the appraisers rather than the actual characteristics of the items being assessed.

To analyse in Attribute Agreement Analysis BioStat user must follow the steps given below.

Load the dataset -> Click on the Six Sigma tab in main menu -> Select MSA -> Choose Attribute Agreement Analysis -> This leads to analysis techniques in the dialog -> Selected the various options in the dialog according to the requirement -> Execute and visualise the output in output window.

The screenshot shows the BioStat software interface. The main window is titled "Attribute Agreement Analysis". On the left, there is a list of source variables: "z\\$ cyl", "z\\$ hp", "z\\$ drat", "z\\$ wt", "z\\$ qsec", "z\\$ vs", "z\\$ gear", and "z\\$ carb". A dropdown menu "Select the variable for sample/part" has "z\\$ mpg" selected. Another dropdown menu "Select the variable for appraiser/operator" has "z\\$ disp" selected. A third dropdown menu "Select the variable for attribute/response" has "III am" selected. An optional field "(Optional) Select the variable for reference/standard response" is empty. A confidence interval field "Confidence interval (alpha) between 0 to 1" contains "0.95". Below these fields is a note: "Leave blank if all the rows to be used. Otherwise specify the Rows to be used to analyze (e.g. specify as 1:25 or 1,4,5,7:12)". The right side of the screen shows an "OUTPUT 1" window titled "Within Appraiser Agreement". It displays a table with the following data:

Operator	Agreement	Inspected	%Agreement	0.95 CI (lower)	0.95 CI (upper)
71.1000	1	25	4	0.1012	20.3517
75.7000	1	25	4	0.1012	20.3517
78.7000	1	25	4	0.1012	20.3517
79.0000	1	25	4	0.1012	20.3517
95.1000	1	25	4	0.1012	20.3517
108.0000	1	25	4	0.1012	20.3517
120.1000	1	25	4	0.1012	20.3517
120.3000	1	25	4	0.1012	20.3517
121.0000	1	25	4	0.1012	20.3517
140.8000	1	25	4	0.1012	20.3517
145.0000	1	25	4	0.1012	20.3517
146.7000	1	25	4	0.1012	20.3517
160.0000	1	25	4	0.1012	20.3517
169.4000	1	25	4	0.1012	20.3517

alt text

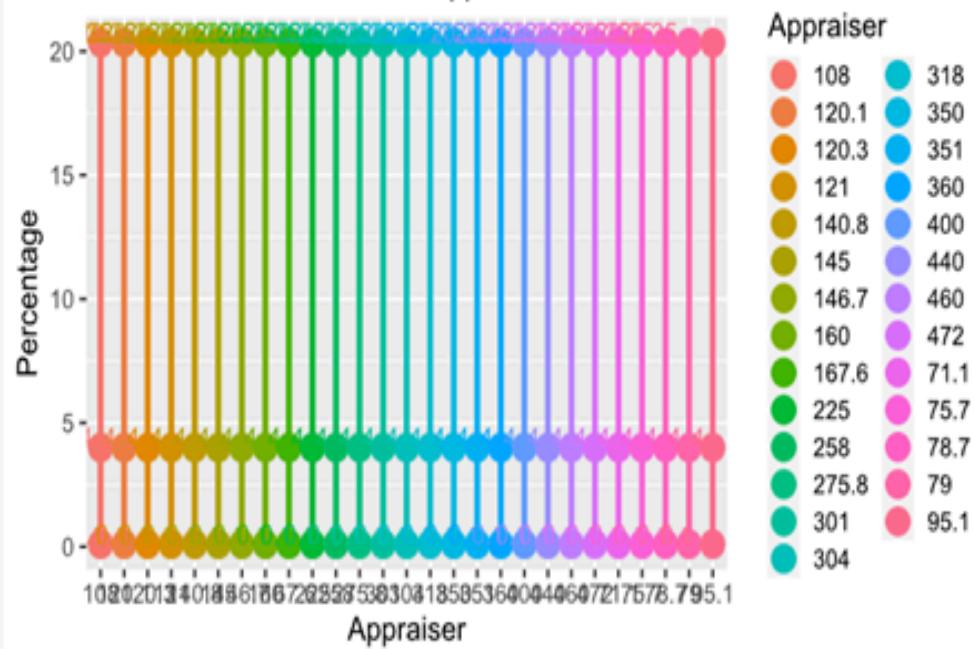
Between Appraiser Fleiss Kappa Statistic

Operator	Response	Kappa	SE Kappa	z	p.value
All	0	-0.0380	0.0110	-3.6030	0 ***
	1	-0.0380	0.0110	-3.6030	0 ***

Note:

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '

Confidence Intervals Within Appraisers



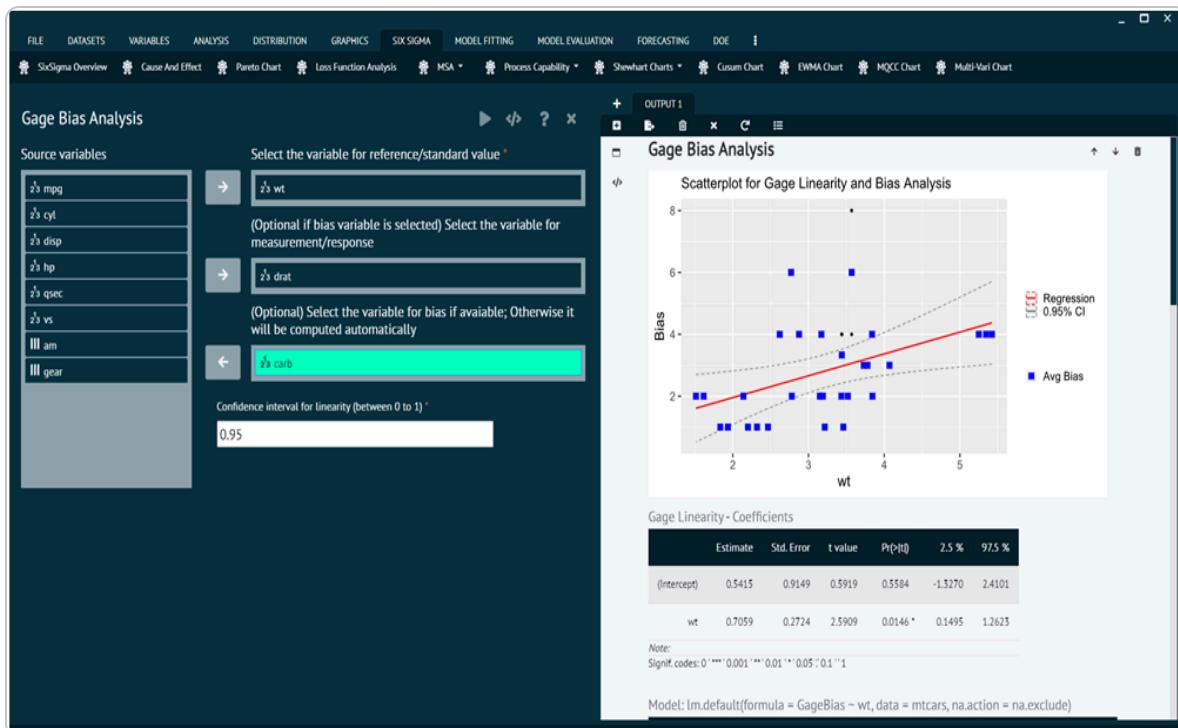
alt text

Gage Bias Analysis

Gage Bias Analysis is a component of Measurement System Analysis (MSA) that focuses on evaluating and quantifying the bias or systematic error within a measurement system. The bias refers to the tendency of a measurement system to consistently overestimate or underestimate the true value of the characteristic being measured. Gage Bias Analysis helps identify and understand this systematic error to improve the accuracy of measurements.

To analyse in Gage Bias Analysis BioStat user must follow the steps given below.

Load the dataset -> Click on the Six Sigma tab in main menu -> Select MSA -> Choose Gage Bias Analysis -> This leads to analysis techniques in the dialog -> Selected the various options in the dialog according to the requirement -> Execute and visualise the output in output window.



alt text

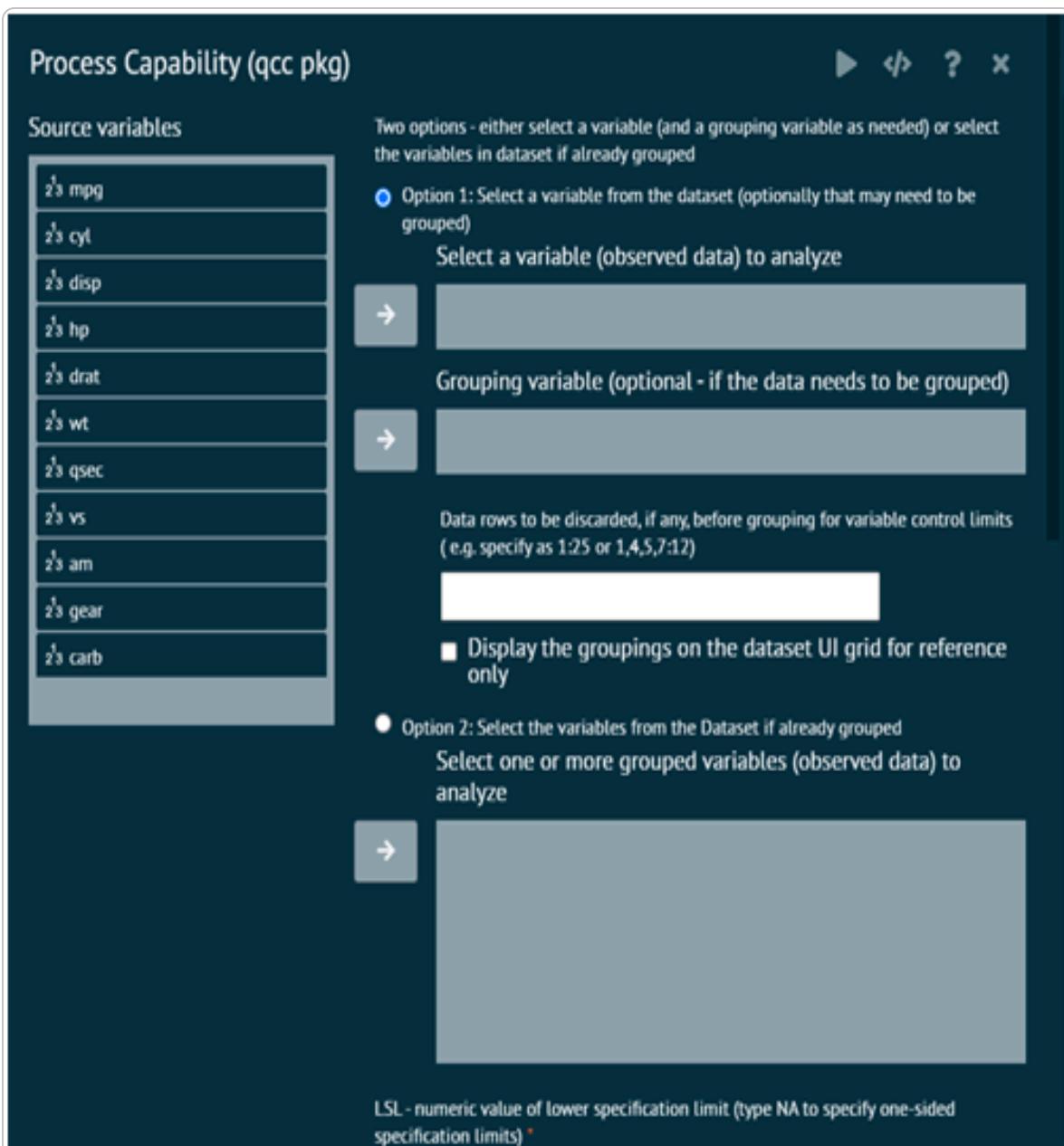
Process Capability

Process Capability is a statistical measure that assesses how well a process can produce products or deliver services within specified limits. It is a key concept in statistical quality control and is used to determine whether a process is capable of meeting predefined specifications. The main objective of process capability analysis is to understand the inherent variability of a process and compare it to the tolerance or specification limits.

Process Capability (Qcc Pkg)

To analyse in Process Capability (QccPkg) BioStat user must follow the steps given below.

Load the dataset -> Click on the Six Sigma tab in main menu -> Select Process Capability -> Choose Process Capability (Qcc Pkg) -> This leads to analysis techniques in the dialog -> selected the various options in the dialog according to the requirement -> Execute and visualise the output in output window.



alt text

Process Capability (SixSigma Pkg)

To analyse in Process Capability (SixSigma Pkg) BioStat user must follow the steps given below.

_ Load the dataset --> Click on the Six Sigma tab in main menu --> Select Process Capability --> Choose Process Capability (SixSigma Pkg) --> This leads to analysis techniques in the dialog --> Selected the various options in the dialog according to the requirement --> Execute and visualise the output in output window.

FILE DATASETS VARIABLES ANALYSIS DISTRIBUTION GRAPHICS SIX SIGMA MODEL FITTING MODEL EVALUATION FORECASTING DOE

SixSigma Overview Cause And Effect Pareto Chart Loss Function Analysis MSA Process Capability Shewhart Charts Cusum Chart EWMA Chart MQCC Chart Multi-Vari Chart

Process Capability Analysis

Source variables Variable with the data of the process performance: `~\$ hp`

LSL - numeric value of lower specification limit: `5`

USL - numeric value of upper specification limit: `5`

Compute a Confidence Interval
Alpha - Type I error (α) for the Confidence Interval: `0.05`

Show graphs and figures for the Process Capability Study
Target of the process:

Variable with the data of the long term process performance: `~\$ qsec`

Main title for the graphic output: Six Sigma Process Capability Analysis Study

Subtitle for the graphic output (e.g. the name of the Six Sigma project): Six Sigma project ID or name for Process Capa

Process Capability Analysis

Cp value without Confidence Intervals: Cp: 0

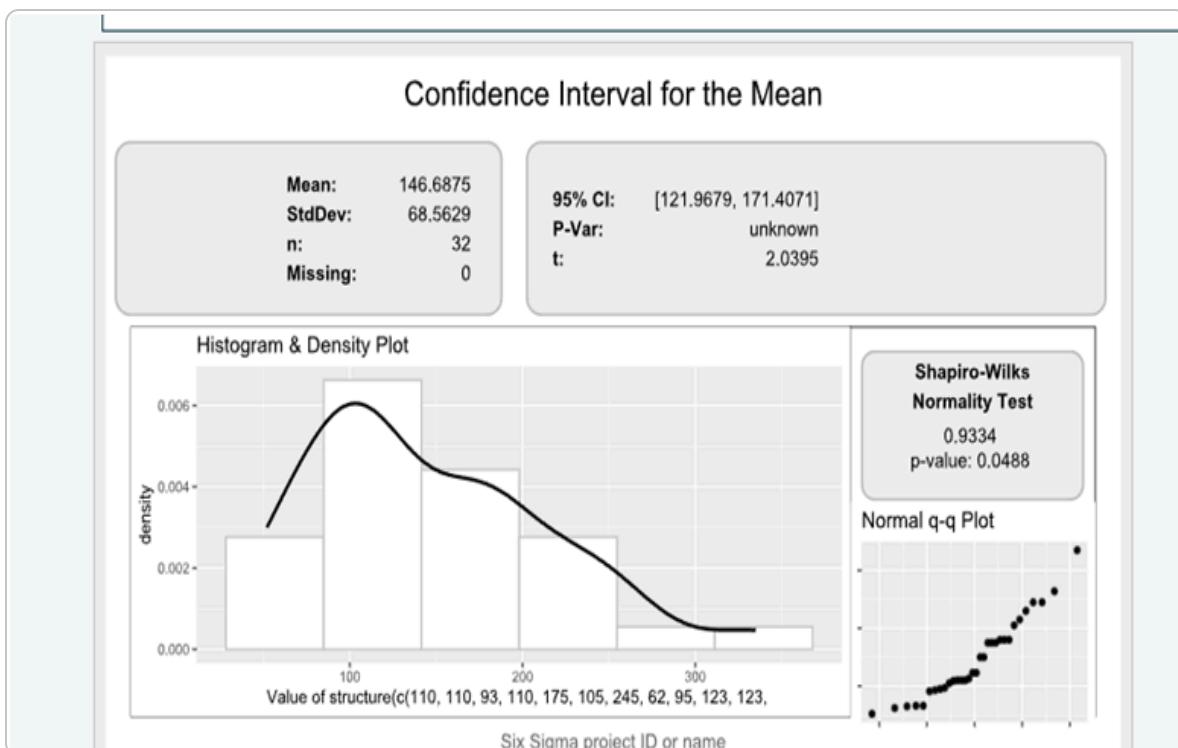
Cpk value without Confidence Intervals: Cpk: -0.6888

Cp Z value: Cp_Z: -2.0865

Mean = 146.6875; sd = 68.5629
95% Confidence Interval= 121.9679 to 171.4071
LL UL
121.9679 171.4071

Confidence Interval for the Mean

alt text



alt text

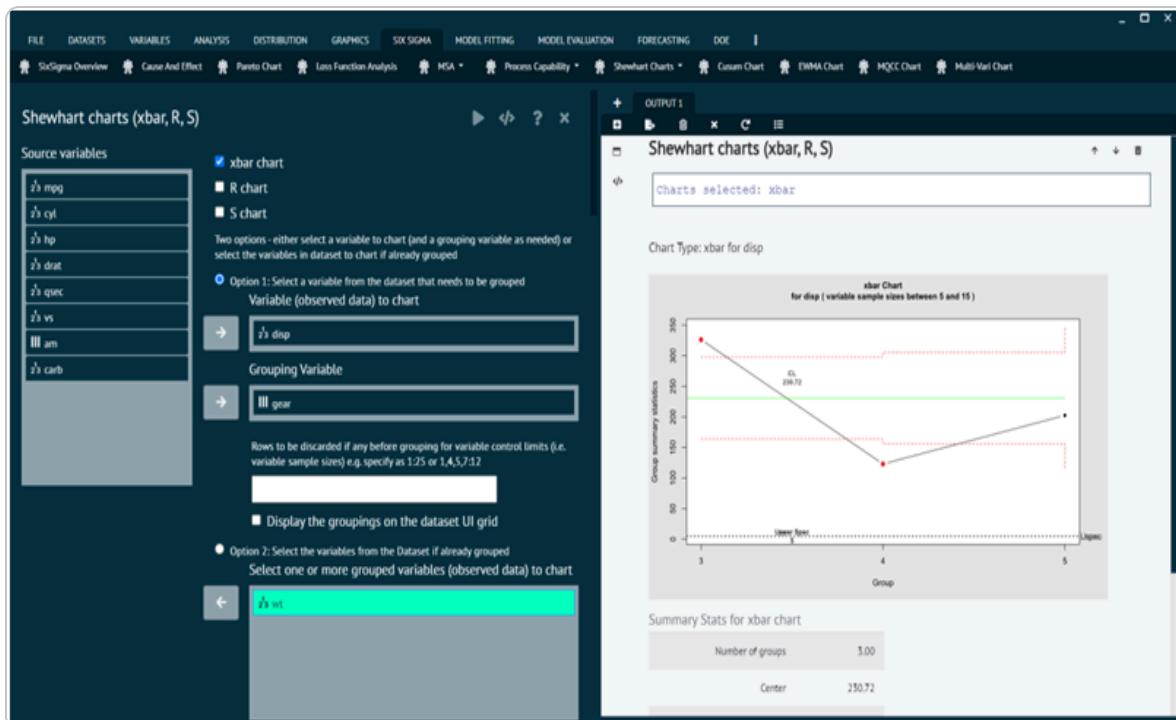
Shewhart Charts

These charts are widely used in quality control and statistical process monitoring to identify and address variations in a process. The primary goal of Shewhart Charts is to distinguish between normal process variation and variations that may indicate a need for corrective action. Shewhart Charts are fundamental in quality management and Six Sigma methodologies, providing a visual and statistical approach to process control. There are several types of Shewhart Charts, each designed to monitor different aspects of a process.

Shewhart Chart (Xbar,R,S)

To analyse in Shewhart Chart (Xbar,R,S) BioStat user must follow the steps given below.

Load the dataset -> Click on the Six Sigma tab in main menu -> Select Shewhart Charts -> Choose Shewhart Chart (Xbar,R,S) -> This leads to analysis techniques in the dialog -> Selected the various options in the dialog according to the requirement -> Execute and visualise the output in output window.



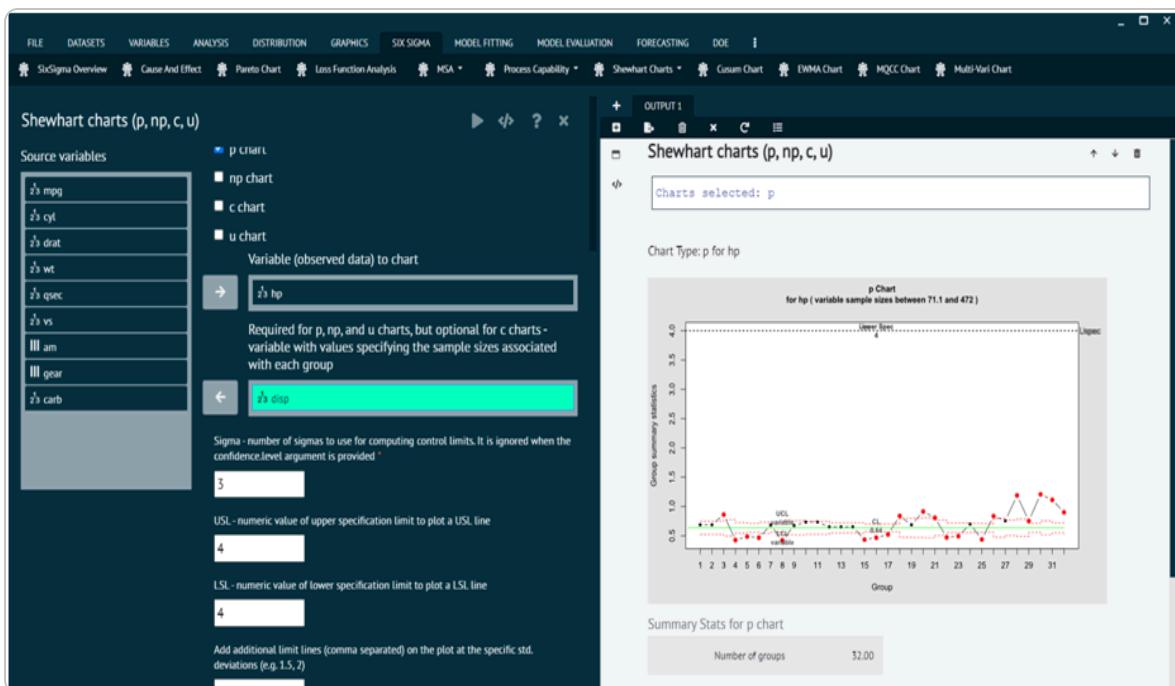
alt text

Shewhart Chart (P,NP,C,U)

Used for monitoring the proportion of nonconforming items in a sample (p-chart), the number of nonconforming items in a sample (np-chart), the count of nonconforming items in a subgroup (c-chart), and the number of nonconforming units per unit (u-chart).

To analyse in Shewhart Chart (P, NP, C, U) BioStat user must follow the steps given below.

Load the dataset -> Click on the Six Sigma tab in main menu -> Select Shewhart Charts -> Choose Shewhart Chart (P, NP, C, U) -> This leads to analysis techniques in the dialog -> Selected the various options in the dialog according to the requirement -> Execute and visualise the output in output window.



alt text

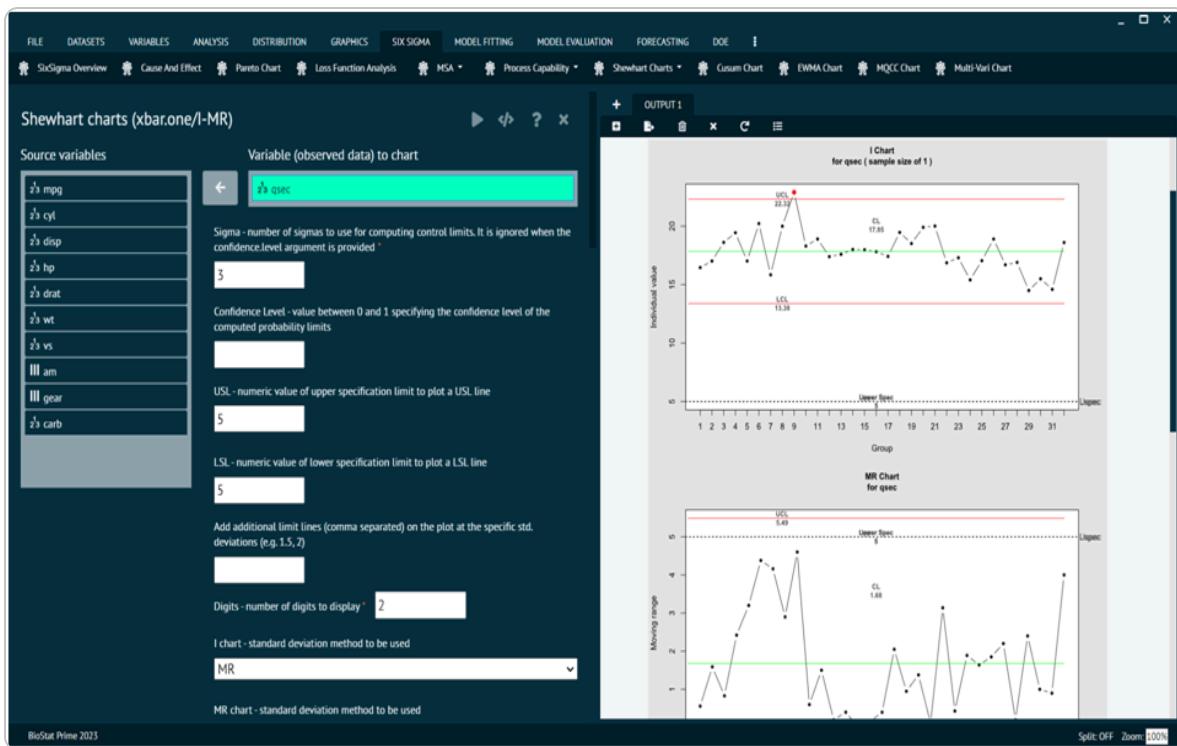
Shewhart Chart (Xbar.One/I-MR)

The X-bar and Individual Moving Range (X-bar.I-MR or X-bar.One) chart is a specific type of Shewhart control chart commonly used for monitoring the central tendency (average) and dispersion of a process over time. This type of chart is suitable when the data is collected in subgroups, and each subgroup consists of a small number of individual measurements.

To analyse in Shewhart Chart (Xbar.One/I-MR) BioStat user must follow the steps given

below.

Load the dataset -> Click on the Six Sigma tab in main menu -> Select Shewhart Charts -> Choose Shewhart Chart (Xbar.One/I-MR) -> This leads to analysis techniques in the dialog -> Selected the various options in the dialog according to the requirement -> Execute and visualise the output in output window.



alt text

Shewhart Chart (I-MR Between/Within)

The Individual and Moving Range (I-MR) Between/Within control chart is a specific type of Shewhart control chart used when the data collected is organized into subgroups, and each subgroup consists of measurements taken at different levels (or locations) and at different times. This type of chart is commonly used when assessing the variation between different levels and within each level of a process.

To analyse in Shewhart Chart (I-MR Between/Within) BioStat user must follow the steps given below.

Load the dataset -> Click on the Six Sigma tab in main menu -> Select Shewhart Charts -> Choose Shewhart Chart (I-MR Between/Within) -> This leads to analysis techniques in

the dialog -> Selected the various options in the dialog according to the requirement ->
Execute and visualise the output in output window.

Shewhart charts (I-MR Between/Within)

Source variables

23 mpg
23 cyl
23 disp
23 hp
23 drat
23 wt
23 qsec
23 vs
23 am
23 gear
23 carb

Two options - either select a variable to chart (and a grouping variable as needed) or select the variables in dataset to chart if already grouped

Option 1: Select a variable from the dataset that needs to be grouped

Variable (observed data) to chart

Grouping Variable

Rows to be discarded if any before grouping for variable control limits (i.e. variable sample sizes) e.g. specify as 1:25 or 1,4,5,7:12

Display the groupings on the dataset UI grid

Option 2: Select the variables from the Dataset if already grouped

Select one or more grouped variables (observed data) to chart

Show the MR chart of subgroup means

Standard deviation method to be used for the underlying 'Within' S chart

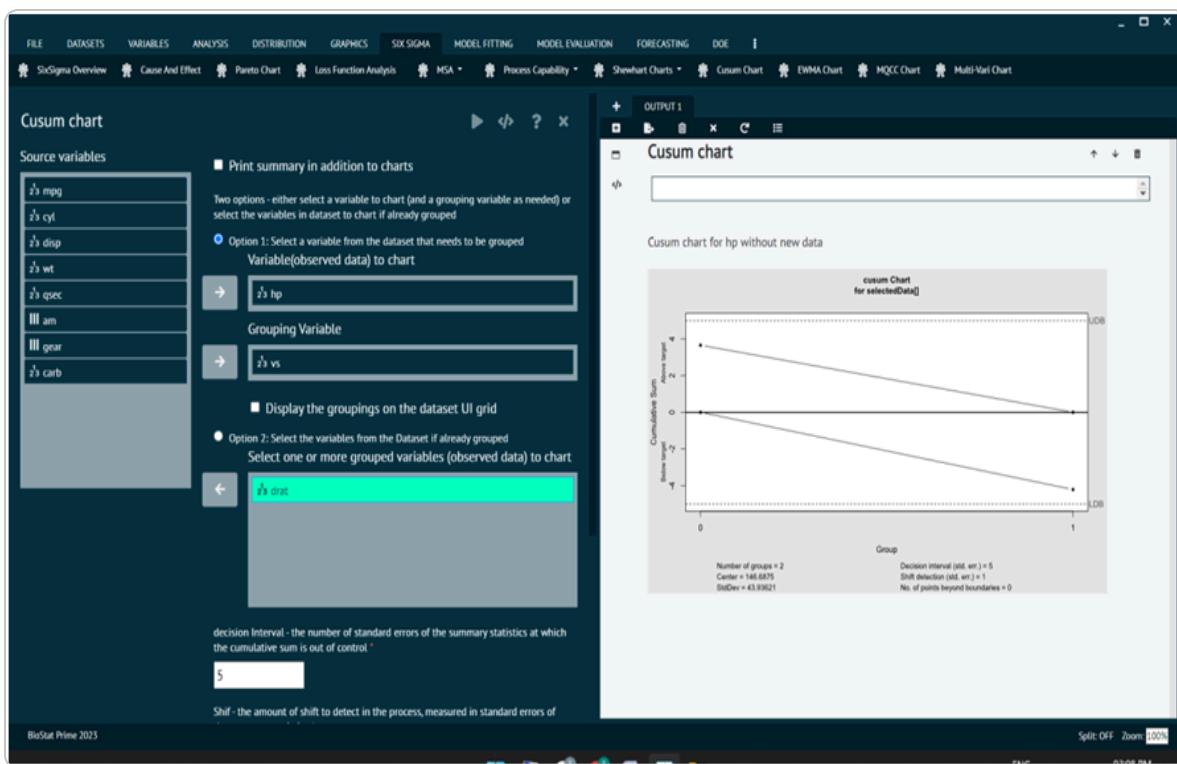
alt text

Cusum Chart

A Cumulative Sum (CUSUM) chart is a statistical control chart that is used in Six Sigma and other quality management methodologies to monitor the stability of a process over time. The CUSUM chart is particularly useful for detecting small, persistent shifts in the process mean.

To analyse Cumulative Sum (CUSUM) chart in BioStat user must follow the steps given below.

Load the dataset -> Click on the Six Sigma tab in main menu -> Select Cumulative Sum (CUSUM) chart -> This leads to analysis techniques in the dialog -> Selected the various options in the dialog according to the requirement -> Execute and visualise the output in output window.



alt text

EWMA Chart

An Exponentially Weighted Moving Average (EWMA) chart is a type of statistical control chart that is used to monitor the stability of a process over time. It is particularly useful when there is a need to give more weight to recent data points, making it sensitive to changes in the process mean.

To analyse EWMA chart in BioStat user must follow the steps given below.

Load the dataset -> Click on the Six Sigma tab in main menu -> Select EWMA chart -> This leads to analysis techniques in the dialog -> Selected the various options in the dialog according to the requirement -> Execute and visualise the output in output window.

EWMA chart



Source variables

<input type="checkbox"/> ¹ / ₃ mpg
<input type="checkbox"/> ¹ / ₃ cyl
<input type="checkbox"/> ¹ / ₃ disp
<input type="checkbox"/> ¹ / ₃ hp
<input type="checkbox"/> ¹ / ₃ drat
<input type="checkbox"/> ¹ / ₃ wt
<input type="checkbox"/> ¹ / ₃ qsec
<input type="checkbox"/> ¹ / ₃ vs
<input type="checkbox"/> ¹ / ₃ am
<input type="checkbox"/> ¹ / ₃ gear
<input type="checkbox"/> ¹ / ₃ carb

■ Print summary in addition to charts

Two options - either select a variable to chart (and a grouping variable as needed) or select the variables in dataset to chart if already grouped

○ Option 1: Select a variable from the dataset that needs to be grouped

Variable(observed data) to chart



Grouping Variable



■ Display the groupings on the dataset UI grid

● Option 2: Select the variables from the Dataset if already grouped

Select one or more grouped variables (observed data) to chart



Sigma - number of sigmas to use for computing control limits *

The smoothing parameter (between 0 and 1) *

alt text

MQCC Chart

To analyse MQCC chart in BioStat user must follow the steps given below.

Load the dataset -> Click on the Six Sigma tab in main menu -> Select MQCC chart -> This leads to analysis techniques in the dialog -> Selected the various options in the dialog according to the requirement -> Execute and visualise the output in output window.

Multivariate Quality Control Chart (MQCC)



Source variables

<input type="checkbox"/> <code>mpg</code>
<input type="checkbox"/> <code>cyl</code>
<input type="checkbox"/> <code>disp</code>
<input type="checkbox"/> <code>hp</code>
<input type="checkbox"/> <code>drat</code>
<input type="checkbox"/> <code>wt</code>
<input type="checkbox"/> <code>qsec</code>
<input type="checkbox"/> <code>vs</code>
<input type="checkbox"/> <code>am</code>
<input type="checkbox"/> <code>gear</code>
<input type="checkbox"/> <code>carb</code>

Print summary in addition to chart(s)

Data (select one or more variables) to chart *



(Optional) Select grouping Variable if subgroups are present



(Optional) exclude groups (if subgroups are numeric) from computation/charting (e.g. specify as 1:10 or comma seperated as 1,4,5,7:12)

(Optional) New Data - groups (if subgroups are numeric) to be used as New Data to chart (e.g. specify as 1:25 or 1,4,5,7:12) - new data to plot but not included in the limit computations

If control limits (Phase I) must be computed and plotted

If prediction limits (Phase II) must be computed and plotted

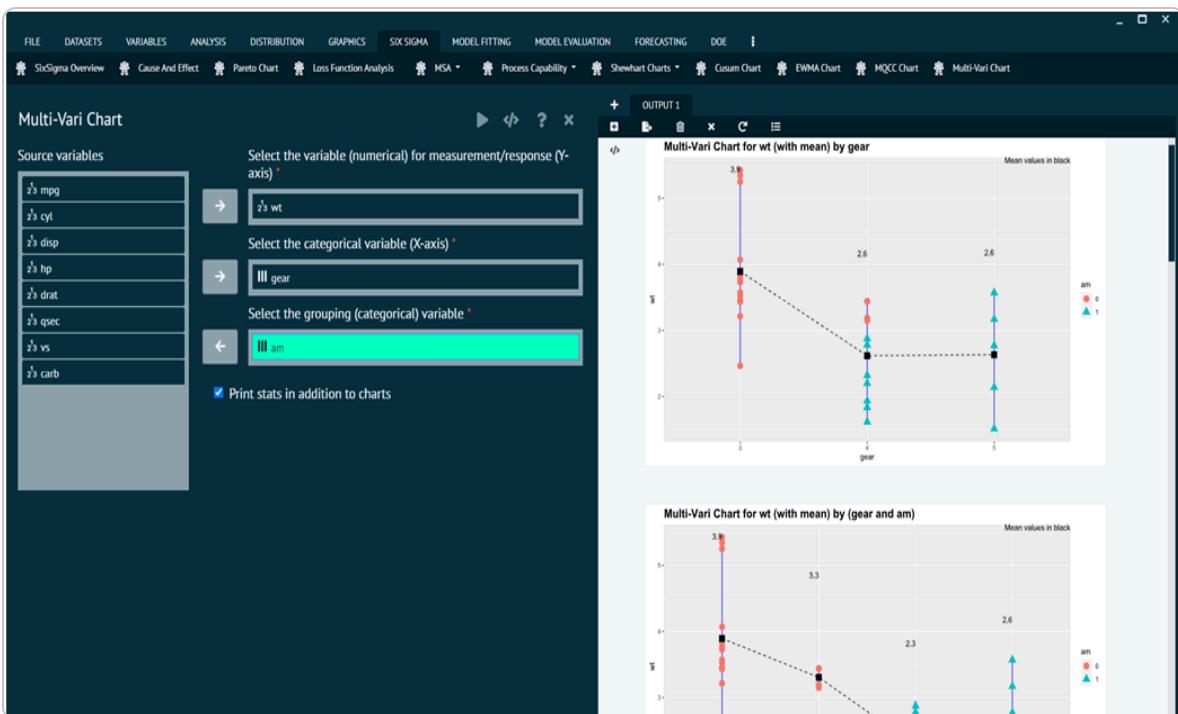
(Optional) Confidence level: leave the default formula as shown (where p will be computed automatically as the number of variables selected). Otherwise specify a

alt text

Multi-Vari Chart

To analyse Multi-Vari chart in BioStat user must follow the steps given below.

Load the dataset -> Click on the Six Sigma tab in main menu -> Select Multi-Vari chart -> This leads to analysis techniques in the dialog -> Selected the various options in the dialog according to the requirement -> Execute and visualise the output in output window.



alt text

Model Fitting

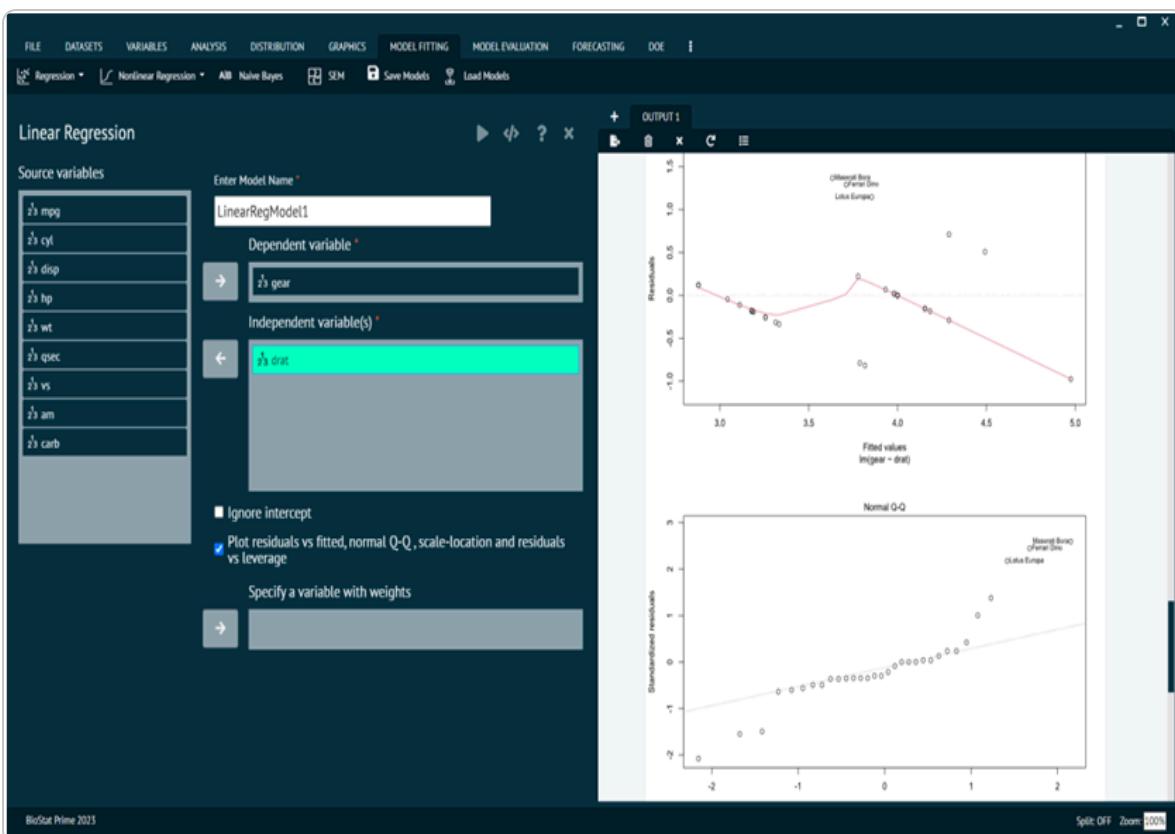
Fitting a model to data means choosing the statistical model that predicts values as close as possible to the ones observed in your population. Fitting a model to data mathematically involves finding the mathematical function or equation that best describes the relationship between the input variables (predictors) and the output variable (response) of a dataset. This process is also called regression analysis. The first step in fitting a model is choosing an appropriate mathematical function to represent the data accurately. BioStat Prime provides an effective way of model fitting via linear and non-linear regression functions present in model fitting tab of main menu.

Regression

Regression analysis is a statistical method used to examine the relationship between one dependent variable and one or more independent variables. The primary goal of regression analysis is to understand how the independent variables contribute to the variation in the dependent variable. It is widely used in various fields, including economics, finance, biology, and social sciences.

To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> Click on the Model Fitting tab in main menu -> Select Linear regression or Non-linear regression based on the requirement -> This leads to analysis techniques in the dialog -> The various options in the dialog can be selected to opt for plot etc -> Finally execute the plot and visualise the output in output window.



alt text

BioStat also provides advanced regression analysis functions that can create models based on interactive terms via formula builder.

The screenshot shows the BioStat Prime 2023 software interface. On the left, the 'Linear Regression' dialog is open, displaying a list of source variables and a dependent variable 'cyl' highlighted in green. A formula builder interface is shown below, with the equation $cyl = \alpha + \beta_1(\text{disp}) + \beta_2(\text{vs}) + \epsilon$. The right side of the screen shows the 'OUTPUT 1' window with the results of the linear regression analysis, including the LM Summary table and the Coefficients table.

	Estimate	Std. Error	t value	Pr(> t)	2.5 %	97.5 %
(Intercept)	4.5312	0.4459	10.1627	4.5700e-11 ***	3.6193	5.4431
disp	0.0095	0.0014	6.9728	1.1500e-07 ***	0.0067	0.0123
vs	-1.2161	0.3345	-3.6358	0.0011 **	-1.9002	-0.5310

alt text

Note that the variables so selected are substituted as quotients in the formula built by the user.

Cox, Advanced

To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> click on the Model Fitting tab in main menu -> Select Regression -> This leads to analysis techniques, choose Cox Basics -> There will appear a dialog, Select the source variables to enter in Time to event or censor and Events (1 = event 1, 0 = censor) options in the dialog -> Populate a formula -> Finally execute.

Using Formula Builder: A Guide

1. To create an expression, click one of the buttons below and drag & drop variables.
2. Toggle the selected button's state by clicking it.
3. Place the cursor where user wants to insert the variable(s) and drag and drop or move it there.

4. Touch a button to see assistance.
5. The All N way button is not able to be toggled.

Cox, Advanced

Click on the ? button on the top right of the dialog for details on sample datasets and the data format supported.

Source variables

13 mpg
13 cyl
13 disp
13 hp
13 drat
13 wt
13 qsec
13 vs
13 am
13 gear
13 carb

Enter model name *

Time to event or censor *

Events (1 = event 1, 0 = censor) *

Model expression builder for independent variables*

Click on a button below and drag and drop variables to create an expression.
Clicking a selected button will toggle its state.
To insert at a position, place the cursor in that position and drag & drop/move variable(s).
Mouse over a button for help.
You cannot toggle the All N way button.

+	-	*	/	
()	%in%		:
All 2 ways		Polynomial terms 2		
df for splines 5		Polynomial degree 5		
B-spline		Natural spline		
Orthogonal polynomial		Raw polynomial		

alt text

Cox, Basics

To analyse it in BioStat Prime user must follow the steps as given.

After loading the dataset, select Regression from the Model Fitting tab in the main menu
 -> This will lead to analytic approaches; select Cox Basics -> A dialog box will then display. In the dialog, select the source variables that need to be established as independent variables -> Lastly, choose which source variables to insert in the Time to event or censor, Events (1 = event 1, 0 = censor) options. -> Finally execute.

The screenshot shows the BioStat Prime interface. The main window is titled "Cox, Basic". On the left, under "Source variables", there is a list of variables: mpg, cyl, disp, drat, wt, qsec, am, gear. To the right of this list are four input fields: "Enter Model Name" (set to "CoxRegModel"), "Time to event or censor" (set to "hp"), "Events (1 = event 1, 0 = censor)" (set to "vs"), and "Independent Variables" (set to "carb"). Below these fields is a "Weights (optional)" section with an empty input field and a "OPTIONS" button. On the right side of the interface, there is an "OUTPUT 1" window titled "Cox, Basic" which displays the "Cox Model Summary for Surv(hp,vs)". The summary includes the following statistics:

	Value
n	32.0000
nevent	14.0000
statistic.log	18.2799
p.value.log	1.9070e-05
statistic.sc	12.1035
p.value.sc	0.0005
statistic.wald	11.2100
p.value.wald	0.0008
statistic.robust	NA
p.value.robust	NA
r.squared	0.4352
r.squared.max	0.9377
concordance	0.8057
std.error.concordance	0.0401

alt text

Cox, Binary Time-depended covariates

To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> Click on the Model Fitting tab in main menu -> Select Regression ->
This leads to analysis techniques, choose Cox, Binary Time-depended covariates ->
There will appear a dialog -> Select the variables in the dialog and populate a formula ->
Finally execute the plot and visualise the output in output window.

Cox, binary time-dependent covariates

Source variables

13 mpg
13 cyl
13 disp
13 hp
13 drat
13 wt
13 qsec
13 vs
13 am
13 gear
13 carb

Enter model name *

Time to event or censor *

Events (1 = event 1, 0 = censor) *

Model expression builder for independent variables

Click on a button below and drag and drop variables to create an expression.
 Clicking a selected button will toggle its state.
 To insert at a position, place the cursor in that position and drag & drop/move variable(s).
 Mouse over a button for help.
 You cannot toggle the All N way button.

+	-	*	/	
()	%in%		:
All 2 ways	Polynomial terms	2		
df for splines	5	Polynomial degree	5	
B-spline	Natural spline			
Orthogonal polynomial	Raw polynomial			

Formula appears here

alt text

Cox, Fine-Gray

To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> Click on the Model Fitting tab in main menu -> Select Regression -> This leads to analysis techniques, choose Cox, Fine-Gray -> There will appear a dialog -> Select the variables in the dialog and populate a formula -> Finally execute the plot and visualise the output in output window.

Cox, Fine-Gray

Source variables

13 mpg
13 cyl
13 disp
13 hp
13 drat
13 wt
13 qsec
13 vs
13 am
13 gear
13 carb

Enter model name*
FineGrayCoxRegModel1

Time to event or censor*
→ []

Events (0 = censor, 1 = event 1, 2 = event 2, ...)*
→ []

Event Code 1

Model expression builder for independent variables*
 Click on a button below and drag and drop variables to create an expression.
 Clicking a selected button will toggle its state.
 To insert at a position, place the cursor in that position and drag & drop/move variable(s).
 Mouse over a button for help.
 You cannot toggle the All N way button.

+	-	*	/	
()	%in%		:
All 2 ways	Polynomial terms	2		
df for splines	5	Polynomial degree	5	
B-spline	Natural spline			

alt text

Cox Regression, Multiple models

Cox Regression, multiple models

Source variables

<code>~mpg</code>	→	
<code>~cyl</code>	→	
<code>~disp</code>	→	
<code>~hp</code>	→	
<code>~drat</code>	→	
<code>~wt</code>	→	
<code>~qsec</code>	→	
<code>~vs</code>	→	
<code>~am</code>	→	
<code>~gear</code>	→	
<code>~carb</code>	→	

Time *

Event (1=event, 0=censor) *

Independent Variables *

Adjustment Variables, Set 1

Adjustment Variables, Set 2

The screenshot shows a software interface for Cox Regression analysis. On the left, a vertical list of source variables from the 'mtcars' dataset is displayed, each preceded by a tilde (~). To the right of each variable is a grey arrow button pointing right, followed by a large grey rectangular input field. The first four variables (~mpg, ~cyl, ~disp, ~hp) have their arrow buttons highlighted in light blue. The top section is titled 'Time *' and contains a field for specifying the event variable (1=event, 0=censor). Below that is a section titled 'Independent Variables *' which is currently empty. Further down are sections for 'Adjustment Variables, Set 1' and 'Adjustment Variables, Set 2', each also containing an empty input field. At the very bottom of the interface is a small 'alt text' label.

alt text

Cox, Stratified

Cox, Stratified

Source variables

z3 mpg
z3 cyl
z3 disp
z3 hp
z3 drat
z3 wt
z3 qsec
z3 vs
z3 am
z3 gear
z3 carb

Enter model name *

Time to event or censor *

Events (1 = event 1, 0 = censor) *

Model expression builder for independent variables*

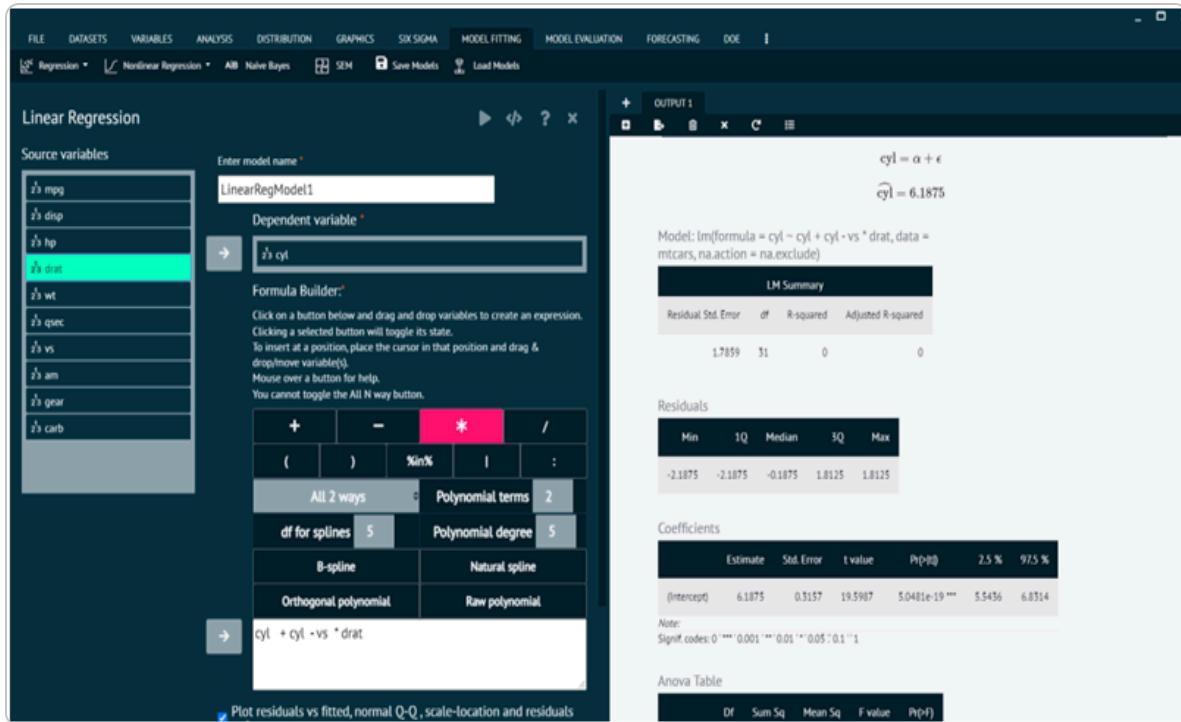
Click on a button below and drag and drop variables to create an expression.
Clicking a selected button will toggle its state.
To insert at a position, place the cursor in that position and drag & drop/move variable(s).
Mouse over a button for help.
You cannot toggle the All N way button.

+	-	*	/	
()	%in%		:
All 2 ways		Polynomial terms 2		
df for splines 5	Polynomial degree 5			
B-spline	Natural spline			
Orthogonal polynomial	Raw polynomial			

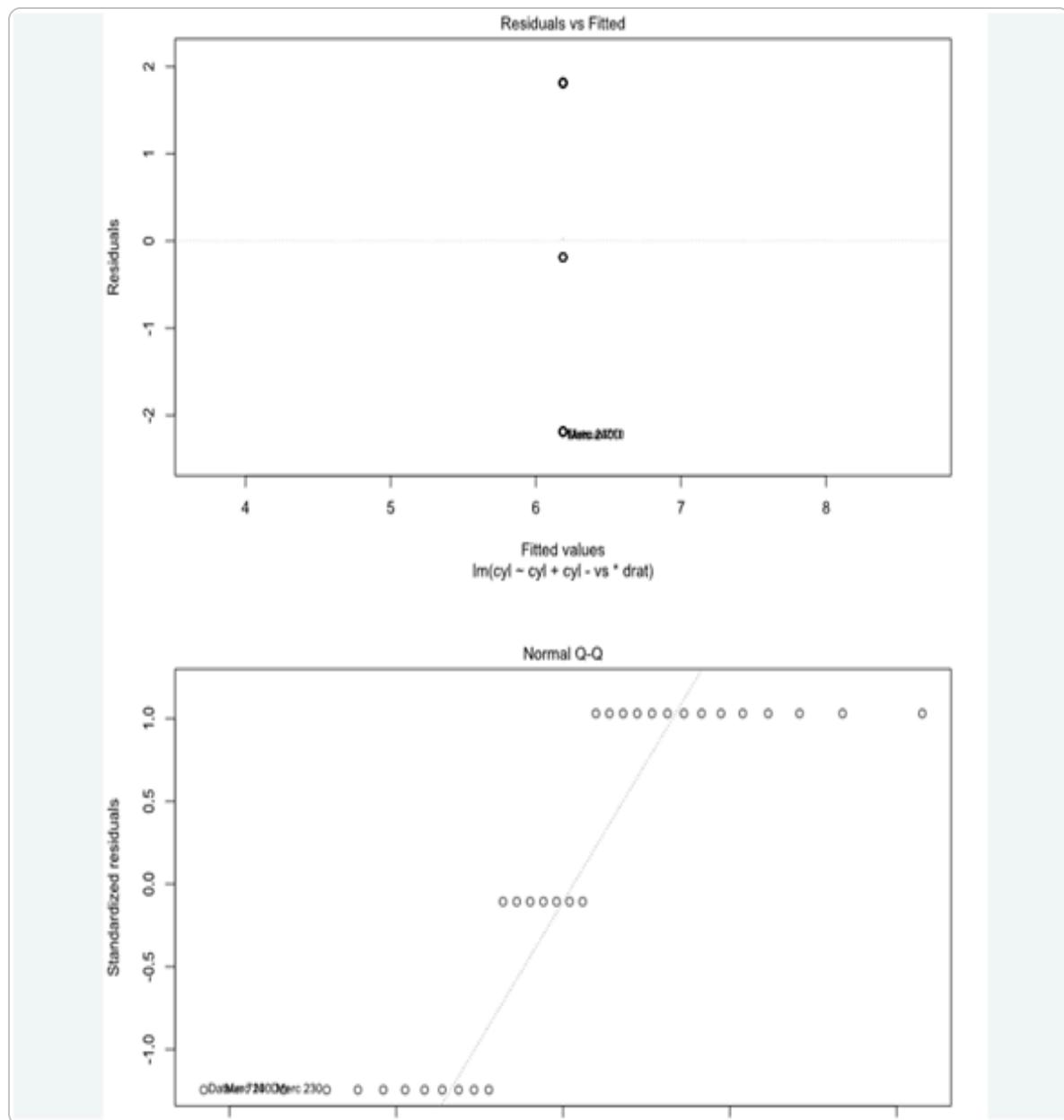
Formula appears here

alt text

Linear Regression, Advanced



alt text



alt text

Linear Regression, Basics

Linear Regression

Source variables

Enter Model Name: LinearRegModel1

Dependent variable: `z\$disp`

Independent variable(s): `z\$vs`, `z\$gear`, `z\$qsec`

Ignore intercept

Plot residuals vs fitted, normal Q-Q, scale-location and residuals vs leverage

Specify a variable with weights

OUTPUT 1

```


$$\widehat{\text{disp}} = 839.9253 - 112.5096(\text{vs}) - 84.6446(\text{gear}) - 13.8863(\text{qsec}) + \epsilon$$


Model: lm(formula = disp ~ vs + gear + qsec, data = mtcars, na.action = na.exclude)

LM Summary
```

Residual Std. Error	df	R-squared	Adjusted R-squared	F-statistic	numdf	dendf	p-value
72.3956	28	0.6918	0.6588	20.9518	3	28	2.5497e-07 ***

Note:
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '

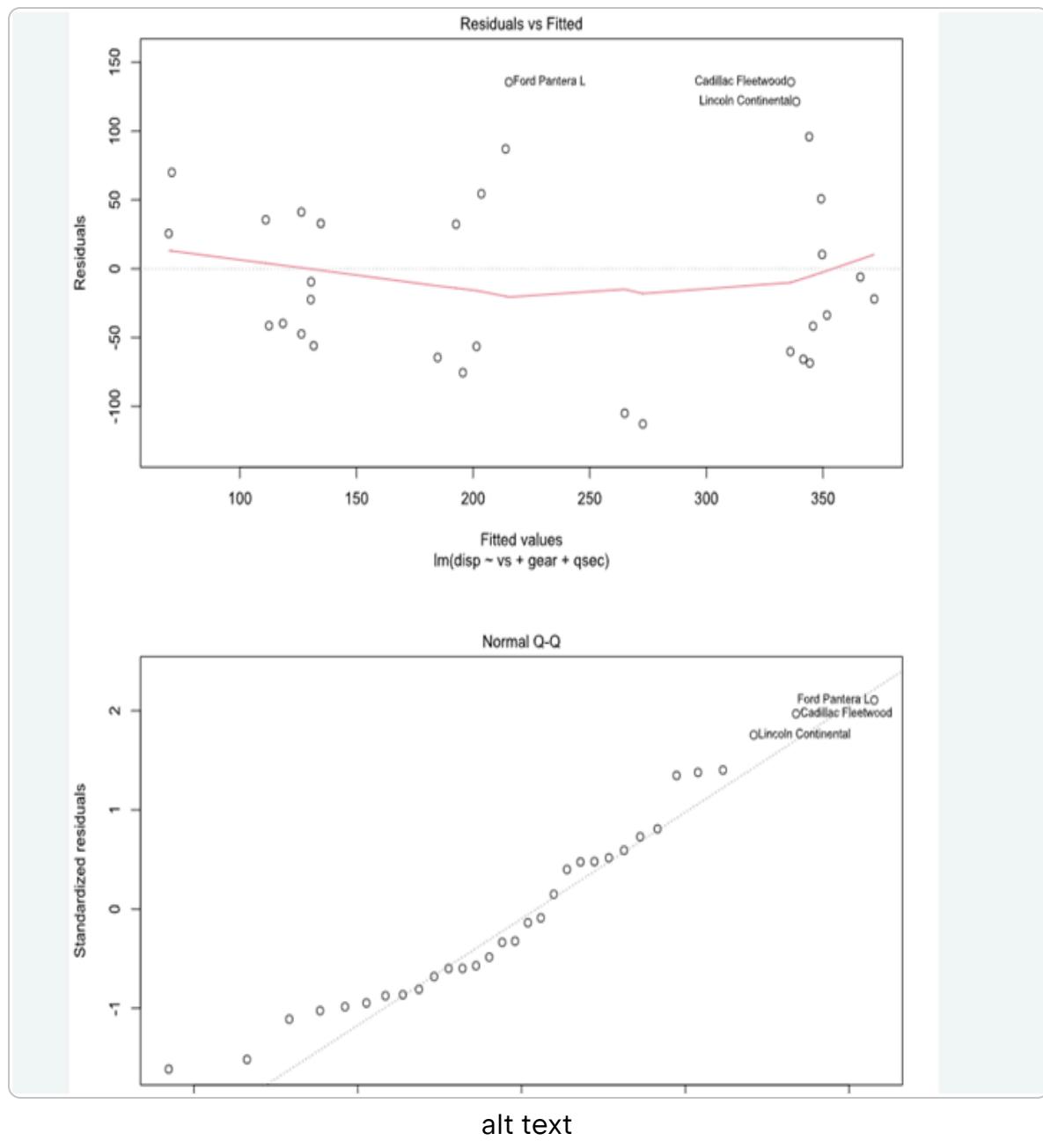
Residuals

Min	1Q	Median	3Q	Max
-112.7780	-56.0880	-15.7775	45.6032	155.6846

Coefficients

	Estimate	Std. Error	t value	Pr(> t)	2.5 %	97.5 %
(Intercept)	839.9253	271.2197	3.0966	0.0044 **	284.3159	1395.5350
vs	-112.5096	46.5902	-2.4149	0.0225 *	-207.9453	-17.0740
gear	-84.6446	21.7447	-3.8926	0.0006 ***	-129.1867	-40.1025

alt text



Linear Regression (Legacy)

FILE DATASETS VARIABLES ANALYSIS DISTRIBUTION GRAPHICS SIX SIGMA MODEL FITTING MODEL EVALUATION FORECASTING DOE

Regression Nonlinear Regression All Naive Bayes SEM Save Models Load Models

Linear Regression

Source variables

- mpg
- disp
- hp
- drat
- wt
- qsec
- am

Enter Model Name: LinearRegModel1

Dependent variable: cyl

Independent variable(s): vs carb gear

Ignore intercept

Plot residuals vs fitted, normal Q-Q, scale-location and residuals vs leverage

Specify a variable with weights

OUTPUT 1

Linear Regression

$y = \alpha + \beta_1(vs) + \beta_2(carb) + \beta_3(gear) + \epsilon$

$$\hat{y} = 10.1888 - 1.7338(vs) + 0.4252(carb) - 1.2037(gear)$$

Model: lm(formula = cyl ~ vs + carb + gear, data = mtcars, na.action = na.exclude)

LM Summary							
Residual Std. Error	df	R-squared	Adjusted R-squared	F-statistic	numdf	dendf	p-value
0.7413	28	0.8444	0.8277	50.6497	3	28	1.9540e-11 ***

Note:
Signif. codes: 0 '***' 0.001 '** 0.01 '*' 0.05 '.' 0.1 ' '

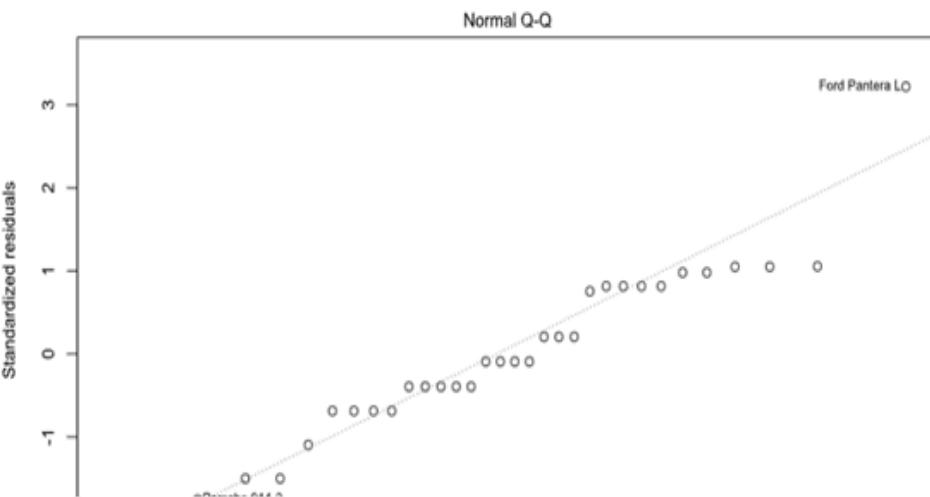
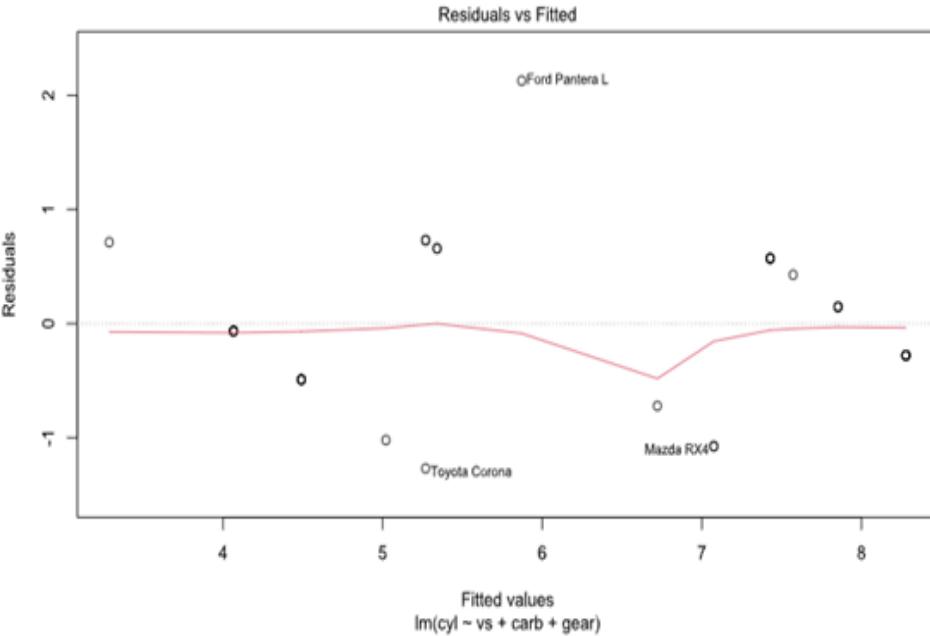
Residuals

Min	IQ	Median	3Q	Max
-1.2691	-0.4906	-0.0654	0.5719	2.1289

Coefficients

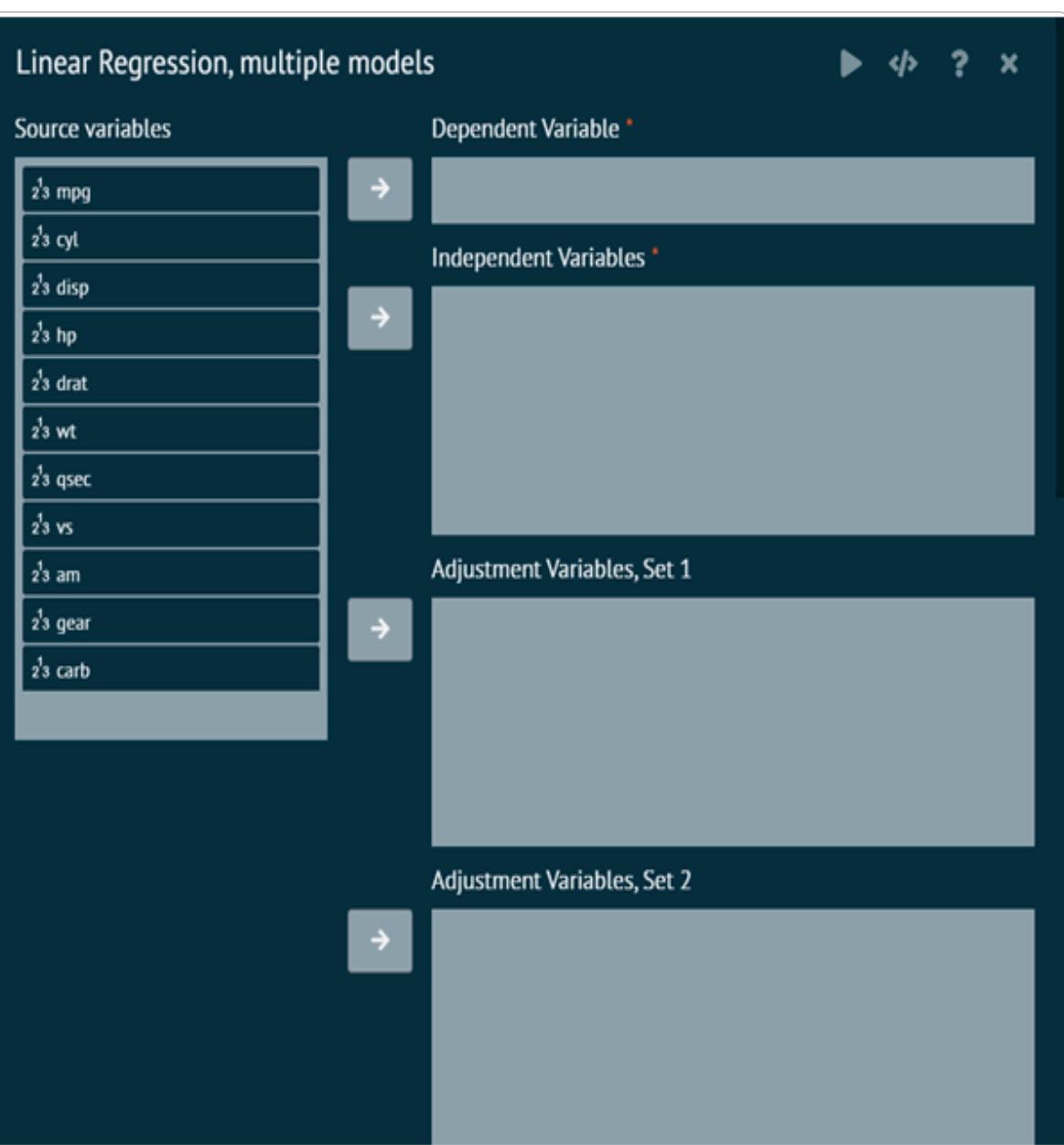
	Estimate	Std. Error	t value	Pr(> t)	2.5 %	97.5 %
(Intercept)	10.1888	0.6805	14.9765	6.7873e-15 ***	8.7952	11.5824
vs	-1.7338	0.3616	-4.7955	4.8513e-05 ***	-2.4744	-0.9932

alt text



alt text

Linear Regression, multiple models



alt text

Logistic Regression, Advanced

Logistic Regression

Source variables

13 mpg
13 cyl
13 disp
13 hp
13 drat
13 wt
13 qsec
13 vs
13 am
13 gear
13 carb

Enter model name *

Dependent variable *

→

Formula Builder:

Click on a button below and drag and drop variables to create an expression.
 Clicking a selected button will toggle its state.
 To insert at a position, place the cursor in that position and drag & drop/move variable(s).
 Mouse over a button for help.
 You cannot toggle the All N way button.

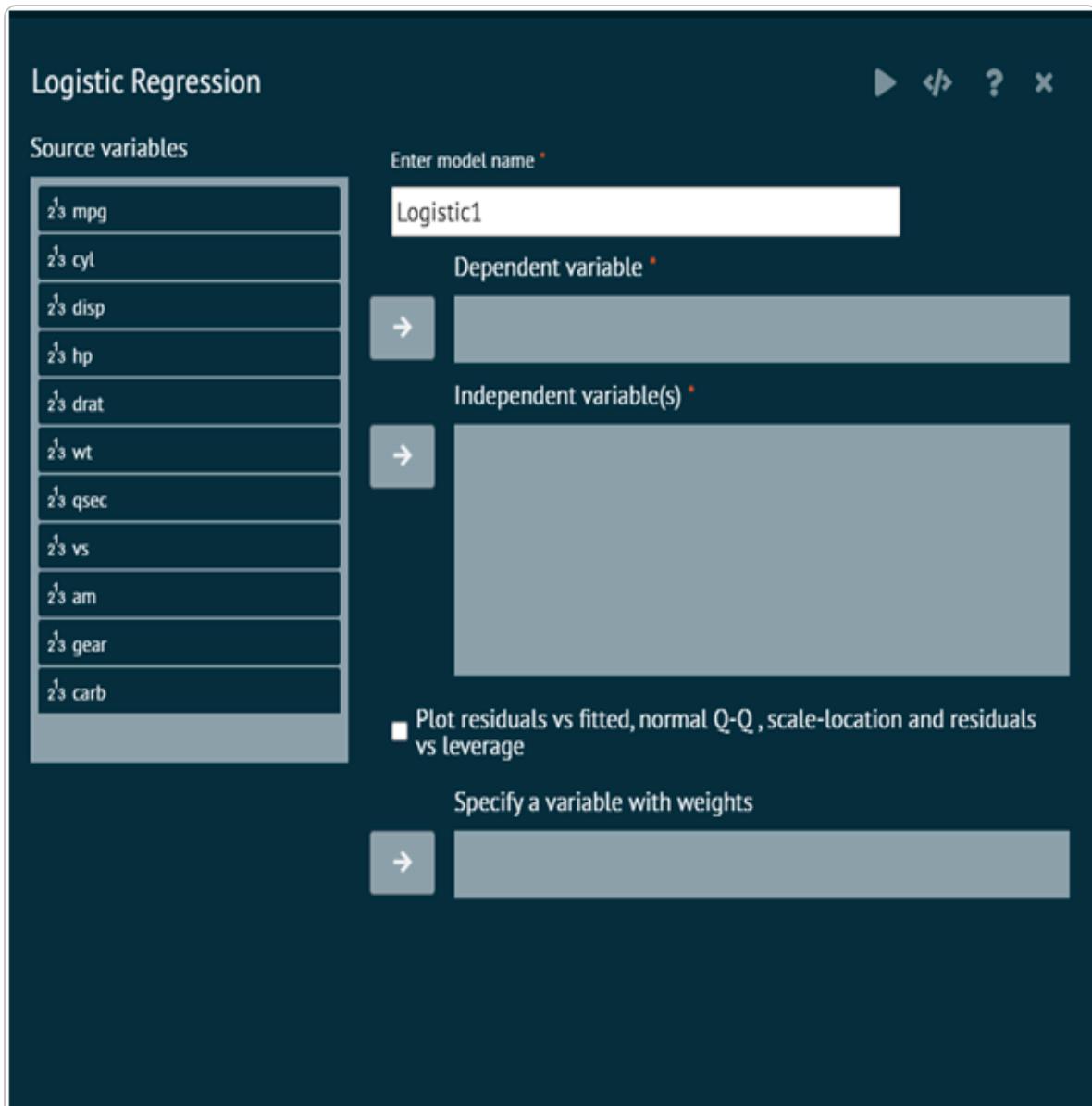
+	-	*	/	
()	%in%		:
All 2 ways	Polynomial terms	2		
df for splines	5	Polynomial degree	5	
B-spline	Natural spline			
Orthogonal polynomial	Raw polynomial			

Formula appears here

Plot residuals vs fitted_normal_Q-Q_scale-location_and_residuals

alt text

Logistic Regression, Basic



alt text

Logistic Regression, Conditional

Logistic, Conditional

Source variables

- `13 mpg`
- `13 cyl`
- `13 disp`
- `13 hp`
- `13 drat`
- `13 wt`
- `13 qsec`
- `13 vs`
- `13 am`
- `13 gear`
- `13 carb`

Enter model name *

Dependent Variable (numeric; 1 = event, 0 = no event) *

→

Formula Builder:

Click on a button below and drag and drop variables to create an expression.
 Clicking a selected button will toggle its state.
 To insert at a position, place the cursor in that position and drag & drop/move variable(s).
 Mouse over a button for help.
 You cannot toggle the All N way button.

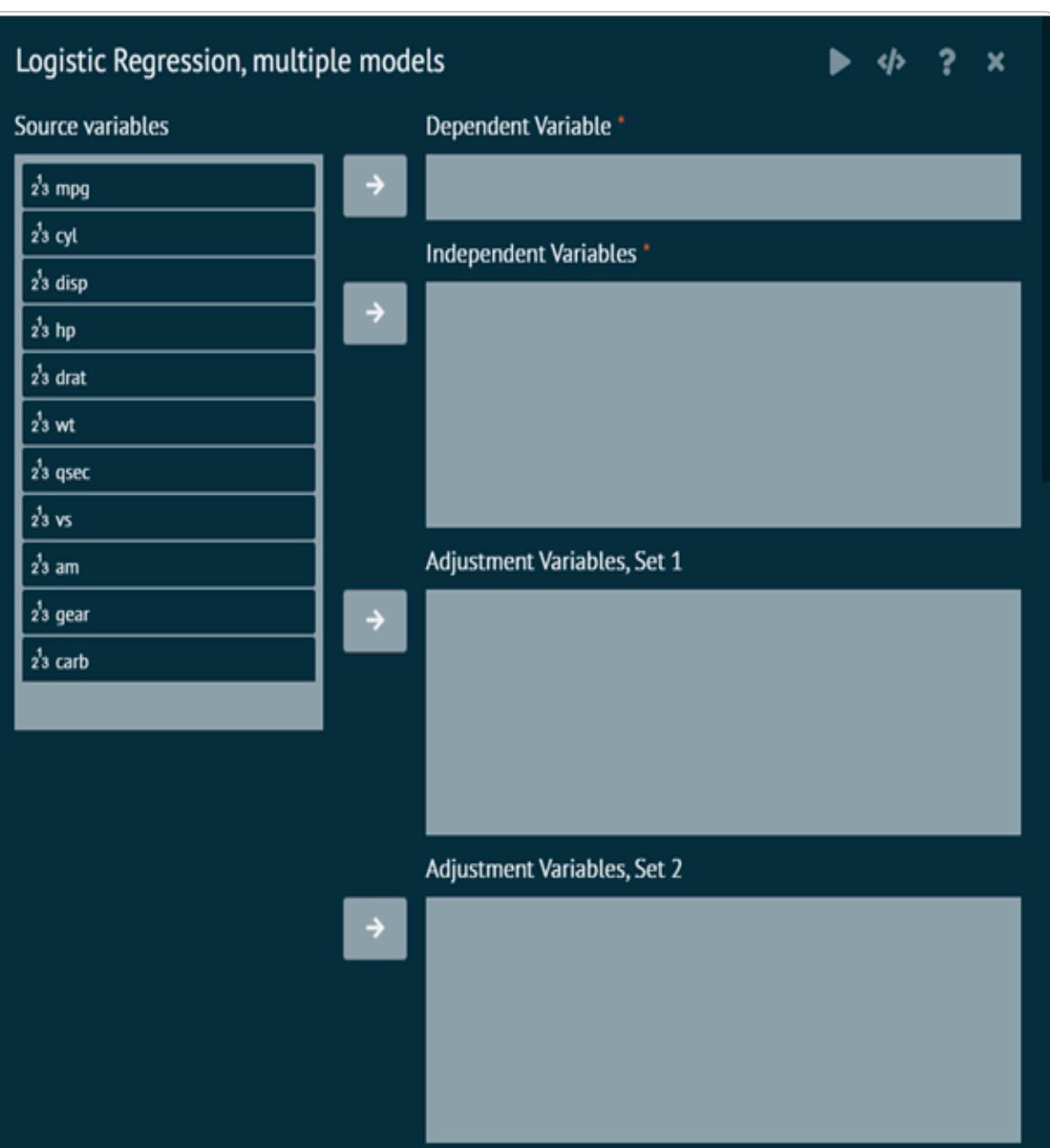
+	-	*	/	
()	%in%		:
All 2 ways		Polynomial terms 2		
df for splines 5		Polynomial degree 5		
B-spline		Natural spline		
Orthogonal polynomial		Raw polynomial		

→ Formula appears here

Strata *

alt text

Logistic Regression, multiple models



alt text

Multinomial Logit

Multinomial Logit

Source variables

- cyl
- disp
- hp
- drat
- wt
- qsec
- vs
- am **selected**
- gear
- carb

Enter model name: MLM

Dependent variable: mpg

Formula Builder:

Click on a button below and drag and drop variables to create an expression. Clicking a selected button will toggle its state. To insert at a position, place the cursor in that position and drag & drop/move variable(s). Mouse over a button for help. You cannot toggle the All N way button.

+	-	*	/	
()	%in%		:
All 2 ways	Polynomial terms	2		
df for splines	5	Polynomial degree	5	
B-spline	Natural spline			
Orthogonal polynomial	Raw polynomial			

Specify a variable with weights

OUTPUT 1

Multinomial Logit

Loading required package: nnet

```
nnet::multinom(formula=mpg ~ qsec - carb * am, data=mtcars, na.action=na.exclude, trace=FALSE)
```

SUMMARYMULTINOM

Residual Deviance	Effective df	AIC
113.8570	48	209.8570

Coefficients

	(Intercept)	qsec
13.3	363.9176	-22.0114
14.3	276.0433	-16.3878
14.7	37.1927	-2.1457
15	466.7576	-28.8613
15.2	18.1466	-1.0197
15.5	151.5841	-8.8310

alt text

Ordinal Regression

Ordinal Regression

Source variables

- `13 mpg`
- `13 cyl`
- `13 disp`
- `13 hp`
- `13 drat`
- `13 wt`
- `13 qsec`
- `13 vs`
- `13 am`
- `13 gear`
- `13 carb`

Enter model name *

Test method

Logit
 Probit

Dependent variable *

→

Formula Builder:

Click on a button below and drag and drop variables to create an expression.
 Clicking a selected button will toggle its state.
 To insert at a position, place the cursor in that position and drag & drop/move variable(s).
 Mouse over a button for help.
 You cannot toggle the All N way button.

+	-	*	/	
()	%in%		:
All 2 ways	Polynomial terms	2		
df for splines	5	Polynomial degree	5	
B-spline	Natural spline			
Orthogonal polynomial	Raw polynomial			

→ Formula appears here

alt text

Quantile Regression

Quantile Regression

▶ ⌂ ? ×

To compare quantile regression model slopes, see "Model Evaluation > Compare > Quant Reg Models"

Source variables

23 mpg
23 cyl
23 disp
23 hp
23 drat
23 wt
23 qsec
23 vs
23 am
23 gear
23 carb

Enter model name *

QuantRegModel1

Dependent Variable *



Formula Builder:

Click on a button below and drag and drop variables to create an expression.
Clicking a selected button will toggle its state.

To insert at a position, place the cursor in that position and drag & drop/move variable(s).

Mouse over a button for help.

You cannot toggle the All N way button.

+	-	*	/	
()	%in%		:
All 2 ways	Polynomial terms	2		
df for splines	5	Polynomial degree	5	
B-spline	Natural spline			
Orthogonal polynomial	Raw polynomial			



Formula appears here

alt text

Non-Linear Regression

Dose Response Curve

Dose Response Curve Model

DRC Model name *

Source variables

- mpg
- cyl
- disp
- hp
- drat
- wt
- qsec
- vs
- am
- gear
- carb

Response variable *

Dose, concentration, enzyme etc *

Choose a dose response equation from the dropdown. The default is LL.4 (a four-parameter Log-logistic model) *

If needed, choose from the dropdown based on the data type being analyzed

Estimation of effective dose response. The default is 50%. You can specify more than one level (e.g. 10, 50, 90) *

A variable if contains the grouping of the data

alt text

Non-Linear Least Square(NLS) Model

Nonlinear Least Squares (NLS) Model

▶ ⏪ ? ×

NLS Model name *

NLS_Model

Source variables

13 mpg
13 cyl
13 disp
13 hp
13 drat
13 wt
13 qsec
13 vs
13 am
13 gear
13 carb

Response (Dependent) Variable



Build or Paste any equation (formula) with Independent (predictor) variable(s) and model parameters e.g. $a * \exp(b * x)$ where a and b are parameters to be estimated and x is the predictor variable. It will create a model equation as $y \sim a * \exp(b * x)$ where y is response variable*

Click on a button below and drag and drop variables to create an expression. Clicking a selected button will toggle its state.

To insert at a position, place the cursor in that position and drag & drop/move variable(s).

Mouse over a button for help.

ARITHMETIC	LOGICAL	MATH	STRING(1)	STRING(2)	C >
+	-	*	/	^	
sqrt	log	log10	log2		
mod	abs		exp		



Create an expression here:

A variable used as weight (Y) with a power value



alt text

Polynomial Regression Model

FILE DATASETS VARIABLES ANALYSIS DISTRIBUTION GRAPHICS SIX SIGMA MODEL FITTING MODEL EVALUATION FORECASTING DOE

Regression Nonlinear Regression All Naive Bayes SEM Save Models Load Models

Polynomial Regression Model

Polynomial Regression model name *

Poly_Model

Source variables

- $\hat{z} \text{ mpg}$
- $\hat{z} \text{ cyl}$
- $\hat{z} \text{ disp}$
- $\hat{z} \text{ drat}$
- $\hat{z} \text{ wt}$
- $\hat{z} \text{ vs}$
- $\hat{z} \text{ am}$
- $\hat{z} \text{ gear}$
- $\hat{z} \text{ carb}$

Dependent (e.g. response) Variable *

$\hat{z} \text{ hp}$

Independent (e.g. dose, concentration,..) Variable *

$\hat{z} \text{ qsec}$

The degree of the polynomial equation. Must be less than the number of unique data points *

2

Compute additional Polynomial models to compare

Specify one or more degrees to create multiple Polynomial models (e.g. 3, 4, 5)

3,4

OUTPUT 1

Polynomial Regression Model

Model name: Poly_Model

Model formula: $\text{hp} \sim \text{poly}(\text{qsec}, 2)$

Model: $\text{lm}(\text{formula} = \text{hp} \sim \text{poly}(\text{qsec}, 2), \text{data} = \text{mtcars})$

LM Summary

Residual Std. Error	df	R-squared	Adjusted R-squared	F-statistic	numdf	dendf	p-value
45.3306	29	0.5911	0.5629	20.9591	2	29	2.3377e-06 ***

Note:
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '

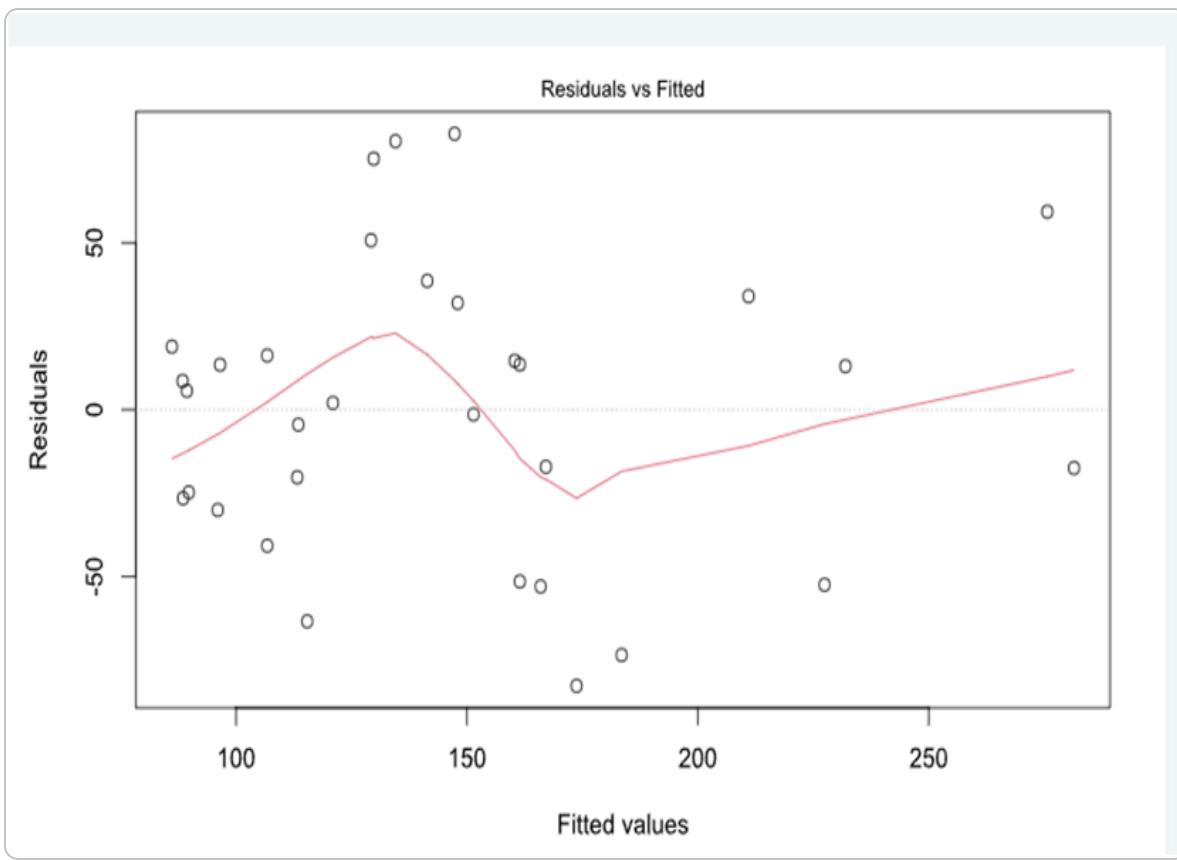
Residuals

Min	Q1	Median	Q3	Max
-92.7287	-27.4012	3.8500	22.1731	82.6803

Coefficients

	Estimate	Std. Error	t value	Pr(> t)	2.5 %	97.5 %
(Intercept)	146.6875	8.0134	18.3053	1.7830e-17 ***	130.2983	165.0767
poly(qsec, 2)1	-270.3585	45.3306	-5.9642	1.7575e-06 ***	-363.0700	-177.6471

alt text



alt text

Naive Bayes

Naive Bayes

Source variables

13 mpg
13 cyl
13 disp
13 hp
13 drat
13 wt
13 qsec
13 vs
13 am
13 gear
13 carb

Enter model name *

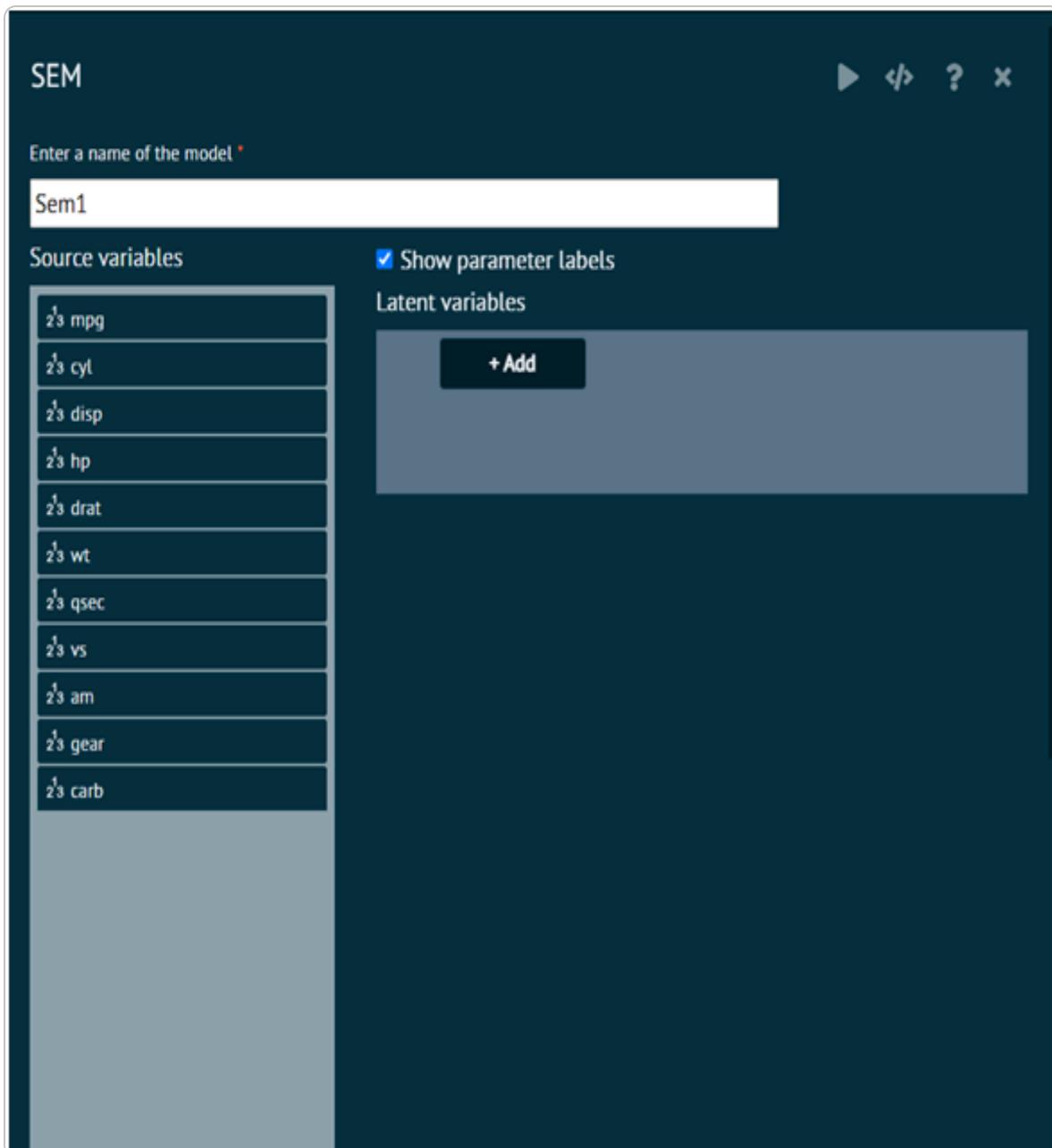
Dependent variable *

Independent variable(s) *

The screenshot shows a software interface for building a Naive Bayes model. On the left, a list of source variables is displayed in a table. On the right, fields for entering a model name, selecting a dependent variable, and specifying independent variables are present. The model name field contains "NaiveBayesModel1". The dependent variable and independent variable fields are currently empty.

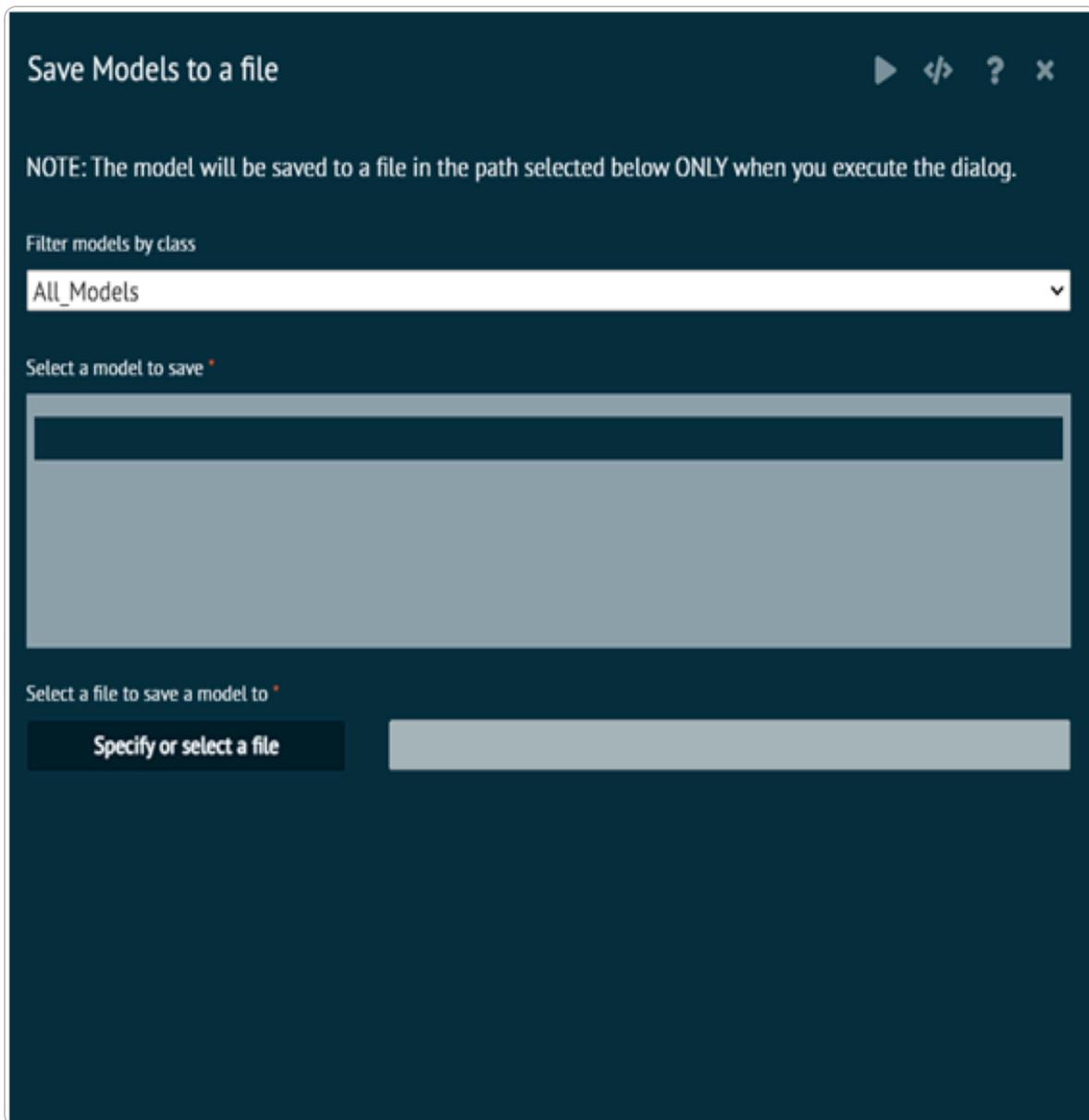
alt text

SEM



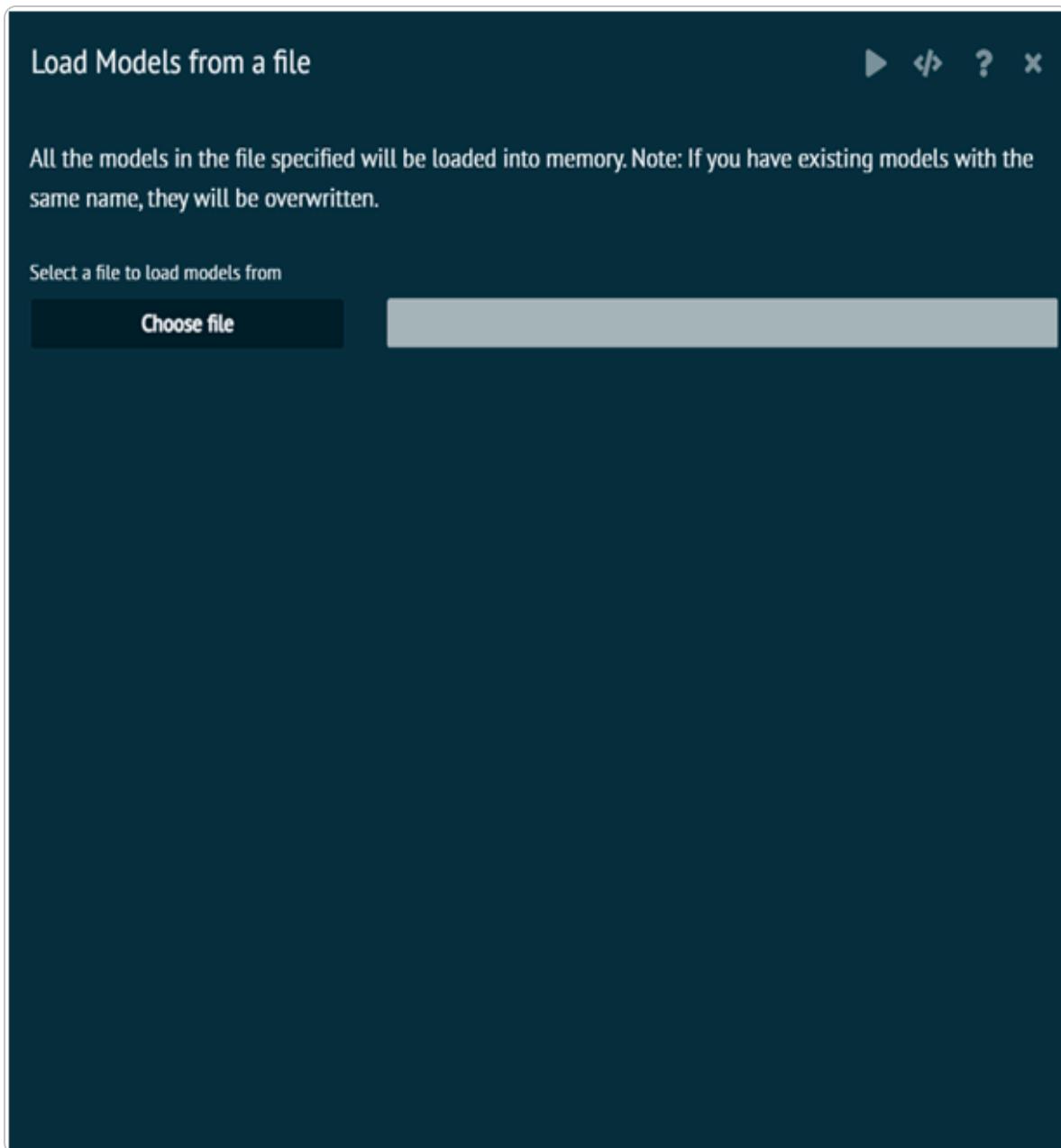
alt text

Save Model to a file



alt text

Load Model from a file



alt text

Model Evaluation

This tab in the main menu aids the user to evaluate the model by comparing, checking the confidence interval, predict the Y values, also perform outlier test and fit the model to check AIC values and BIC values (when comparing).

Compare N Models

The screenshot shows the "Compare N Models" interface in a software application. The top menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION, FORECASTING, and DOE. Below the menu is a toolbar with icons for Compare, Confidence Interval, Fit, Outlier Test, and Predict. The main window has a title bar "Compare N Models" and a status bar "OUTPUT 1". A left sidebar displays a list of models: cox.form, CoxRegModel1, LinearRegModel1, MLH, and Poly_Model. The right panel is titled "Compare N Models" and contains a "Statistical Model Comparison" table. The table has columns for Model 1, Model 2, and Model 3. The data rows show coefficients and standard errors for variables like carb, (Intercept), vs, gear, and various interaction terms.

	Model 1	Model 2	Model 3
carb	-1.1556*** (0.3451)	0.4252*** (0.1148)	
(Intercept)		10.1888*** (0.6803)	
vs		-1.7338*** (0.3616)	
gear		-1.2037*** (0.2111)	
13.3:(Intercept)			363.9176*** (79.1366)
13.3:qsec			-22.0114*** (5.1005)
14.3:(Intercept)			276.0433*** (80.6469)
14.3:qsec			-16.3878*** (4.8382)
14.7:(Intercept)			37.1927

alt text

Confidence Interval

A confidence interval (CI) is a statistical concept used to estimate a range of values within which a population parameter is likely to fall. It provides a measure of the uncertainty or variability associated with estimating a population parameter from a sample of data. Confidence intervals are commonly used in inferential statistics, hypothesis testing, and research to make inferences about a population based on sample data.

To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> Click on the model evaluation tab in main menu -> Select confidence interval -> Select a model -> Execute.

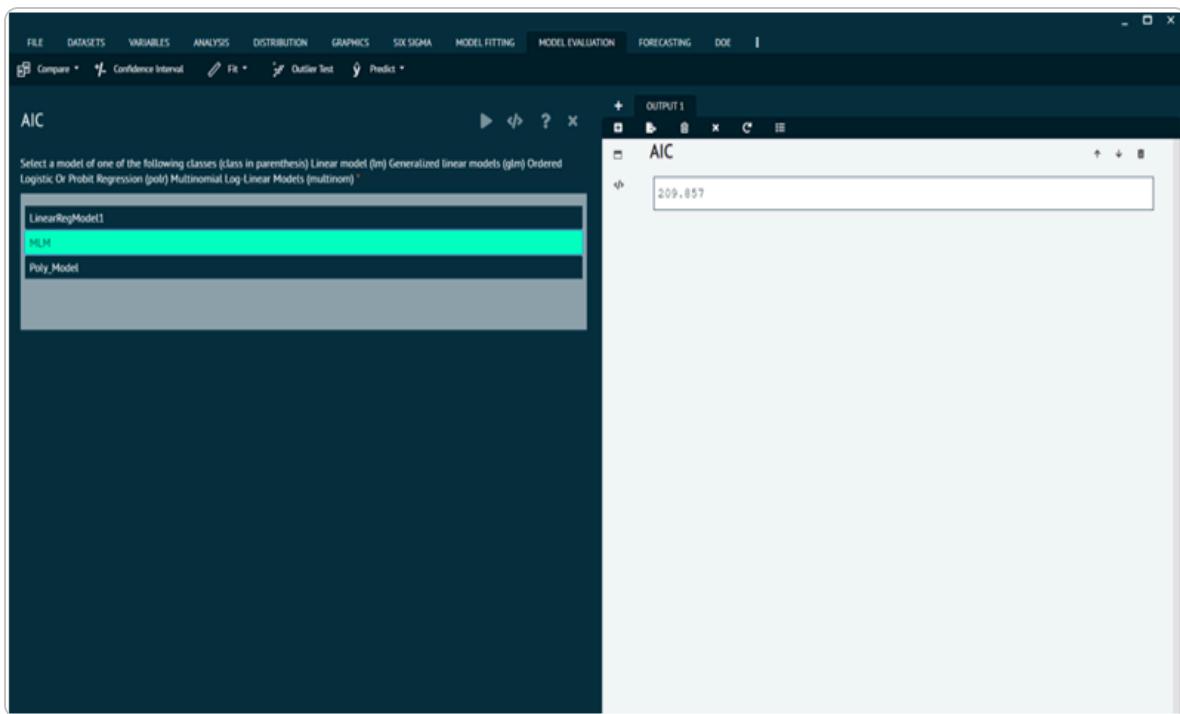
The screenshot shows the BioStat Prime application interface. The main window title is 'Confidence Interval'. In the top menu bar, the 'MODEL EVALUATION' tab is selected. The left panel displays a list of model classes: 'Linear model (lm)', 'Generalized linear model (glm)', 'Nonlinear Least Squares (nls)', 'Ordered Logistic/Probit regression (polr)', and 'Multinomial Log Linear Models (multinom)'. A dropdown menu 'Select a model' is open, showing three options: 'LinearRegModel1' (selected), 'MLM', and 'Poly_Model'. Below this, a 'Confidence interval' section has a slider set to '0.95'. Underneath the slider, two radio buttons are visible: 'Likelihood-ratio statistic' (selected) and 'Wald statistic'. The right panel is titled 'OUTPUT1' and contains a sub-section titled 'Confidence Interval'. This section shows the R code used to load packages and attach them, followed by the command 'Confidence Interval (level = 0.95)'. The resulting table displays the confidence intervals for the coefficients of a linear regression model:

	2.5 %	97.5 %
(Intercept)	8.7952	11.5824
vs	-2.4744	-0.9932
carb	0.1901	0.6604
gear	-1.6361	-0.7713

alt text

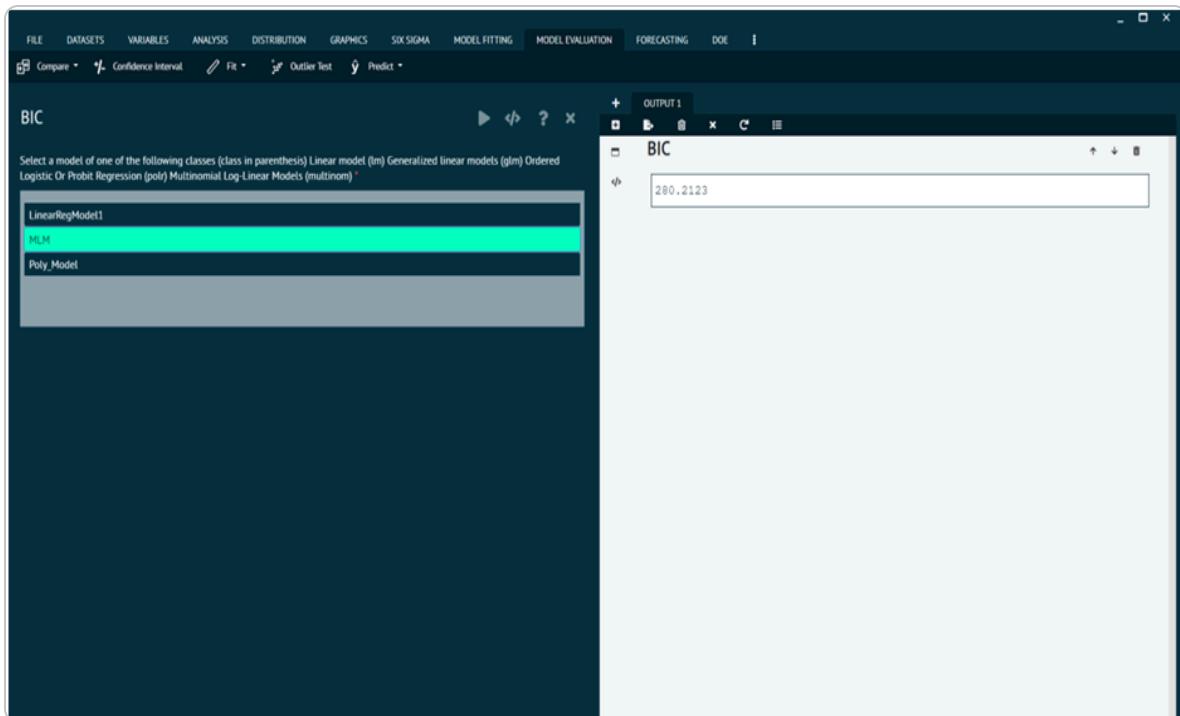
FIT

AIC



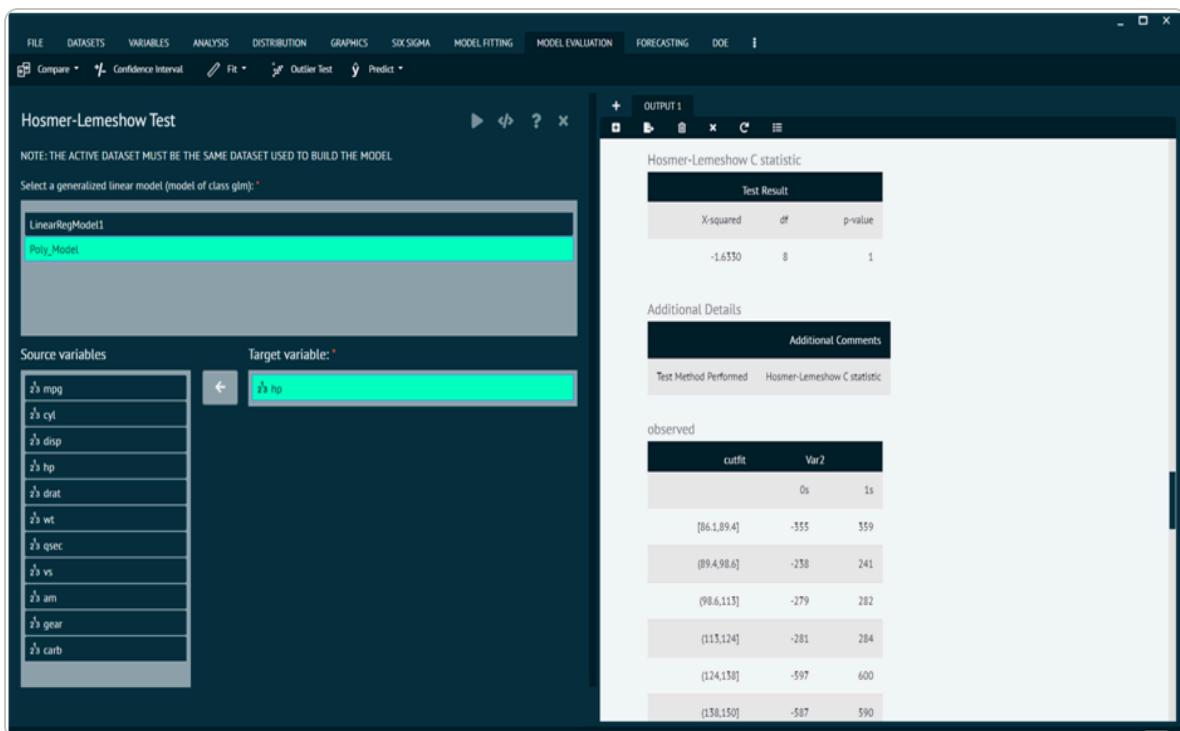
alt text

BIC



alt text

Hosmer-Lemeshow Test



alt text

Pseudo R Squared

The screenshot shows a software application window with a dark theme. The top menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION, FORECASTING, and DOE. Below the menu is a toolbar with icons for Compare, Confidence Interval, Fit, Outlier Test, and Predict.

The main window title is "Pseudo R Squared". A sub-header says "Select a model of one of the following classes (class in parenthesis): Generalized linear models (glm), Multinomial Log Linear Models (multinom), Ordered Logistic Or Probit Regression (polr)". A dropdown menu titled "Select a model" contains the option "MLM", which is highlighted with a green background.

To the right, there is an "OUTPUT 1" panel titled "Pseudo R Squared". It displays the message "fitting null model for pseudo-r2". Below this is a table titled "pseudo-R2 measures" with columns: I_b, I_bNull, G₂, McFadden, r₂ML, and r₂CU. The data row shows values: -56.9285, -101.1995, 88.5420, 0.4375, 0.9371, and 0.9588 respectively.

alt text

Outlier Test

Bonferroni Outlier Test

The screenshot shows a software application window titled "Bonferroni Outlier Test". The menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION, FORECASTING, DOE, and HELP. The ANALYSIS menu is currently selected. The toolbar below the menu bar includes icons for Compare, Confidence Interval, Fit, Outlier Test (which is highlighted), and Predict.

The main panel displays the "Bonferroni Outlier Test" dialog. It asks to "Select a model of one of the following classes (class in parenthesis)- Linear model (lm), Generalized linear model (glm)". A dropdown menu lists "LinearRegModel1" and "Poly_Model", with "LinearRegModel1" highlighted in green. The right panel, titled "OUTPUT 1", shows the results of the test:

rstudent	unadjusted p-value	Bonferroni p
Ford Pantera L	3.983808	0.00046193 0.014782

alt text

Model scoring

Model Scoring

Score A Dataset Using A Model

Filter models by class

All Models

Select a model to score a dataset *

Diagnostic tests

Test Results: As soon as a model is selected, we will run tests to see whether dependent variables specified in the model are available in the dataset to be scored. The results will be displayed here

Save predicted values and supporting statistics.

Predictions and predicted probabilities where applicable are stored in the dataset being scored as new variables with prefix below

Specify column name prefix *

■ Save confidence intervals for individual predicted values **(Valid only for linear models (class lm))

Specify the confidence level

0.95

■ Generate Confusion Matrix

**For dependent variables with 2 levels, the 2nd level is treated as the positive level. See Data > Factor Levels > Reorder Levels Manually to change the order of factor levels and rebuild the model.

When the variable to predict has 2 levels, specify the level of interest. The confusion matrix and related statistics are

alt text

Forecasting

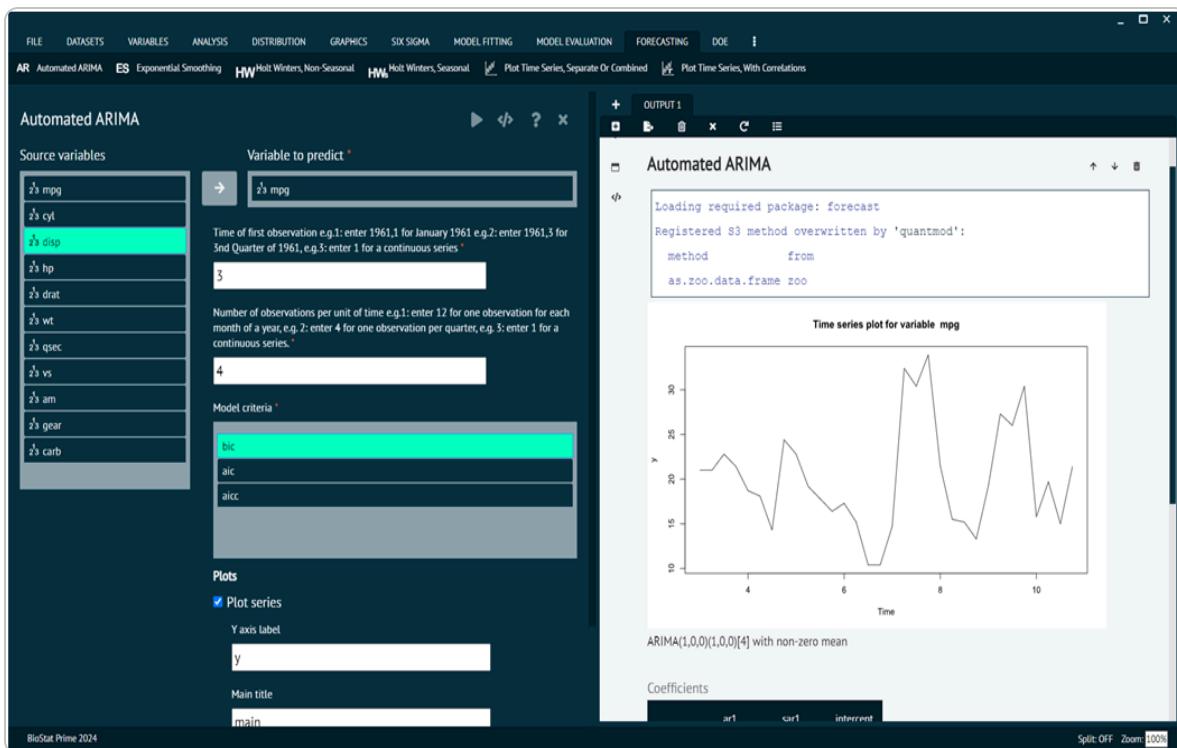
Forecasting in statistics refers to the process of making predictions about future values or trends based on historical data and patterns. It involves using statistical models, techniques, and methods to estimate future outcomes or trends in a time series or set of data points. The primary goal of forecasting is to make informed decisions or plans by leveraging the information available from past observations. Forecasting often deals with time-ordered data, where observations are recorded sequentially over time. Examples include stock prices, sales data, weather measurements, and more. BioStat Prime has leveraged the use of its computing capacity by using R programming language because R provides numerous packages and functions specifically designed for time series forecasting. In BioStat Prime one of the tabs on the main menu is for forecasting. It is in-charge of the analysis of secondary data.

Automated ARIMA (AR)

ARIMA, which stands for Auto-Regressive Integrated Moving Average, is a popular time series forecasting model in statistics. It combines three components: Auto-Regressive (AR), Integrated (I), and Moving Average (MA). ARIMA models are widely used to analyze and forecast time-series data, where observations are collected at regular intervals over time. Automated ARIMA refers to the process of automatically selecting the best parameters (p , d , q) for the ARIMA model.

To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> Click on the Forecasting tab in main menu -> Select Automated ARIMA -> Choose variables to predict -> Write Time of first observation -> Write Number of observations per unit of time, choose model criteria -> Execute.



alt text

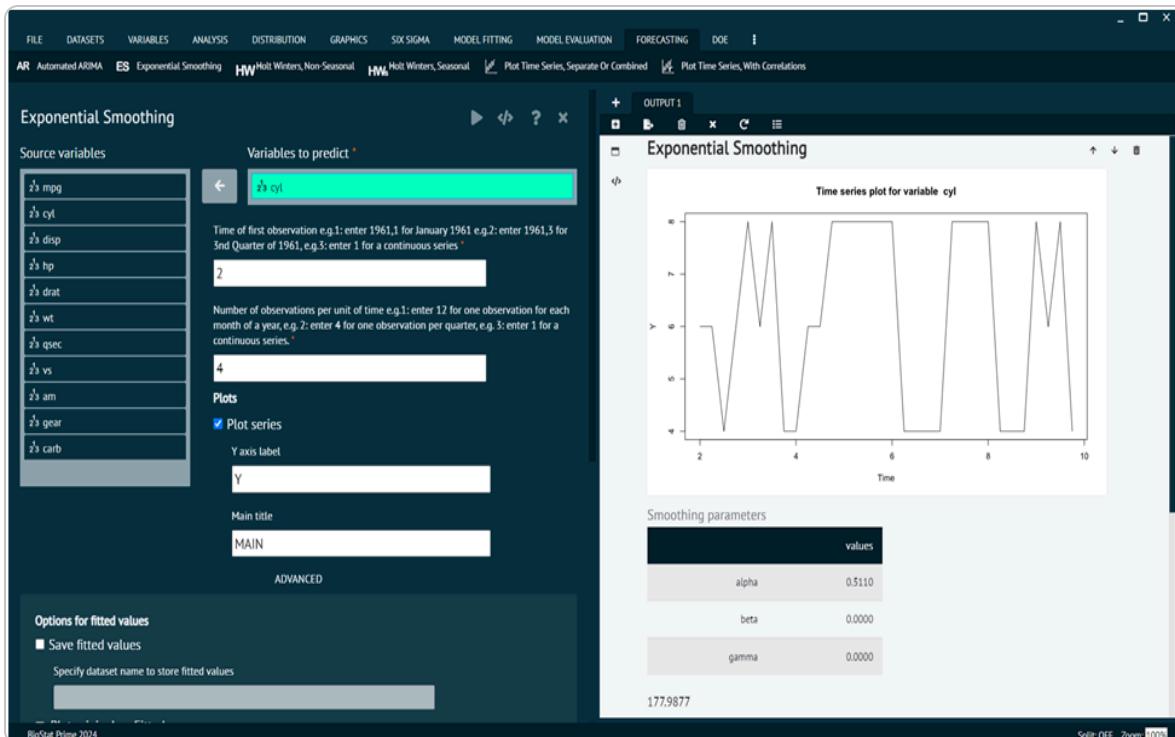
The user can choose additional options like plot options.

Exponential Smoothing (ES)

Exponential smoothing is a time series forecasting method used in statistics. It is particularly useful for forecasting data points that exhibit a consistent pattern or trend over time. Exponential smoothing assigns exponentially decreasing weights to older observations in a time series, with more recent observations receiving higher weights. This approach is effective in capturing short-term fluctuations and trends in the data. The basic idea behind exponential smoothing is to assign weights to past observations, with the weights decreasing exponentially as the observations get older. The most commonly used exponential smoothing method is called Simple Exponential Smoothing (SES).

To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> Click on the Forecasting tab in main menu -> Select Exponential Smoothing (ES) -> Choose variables to predict -> Write Time of first observation -> Write Number of observations per unit of time -> Execute.



alt text

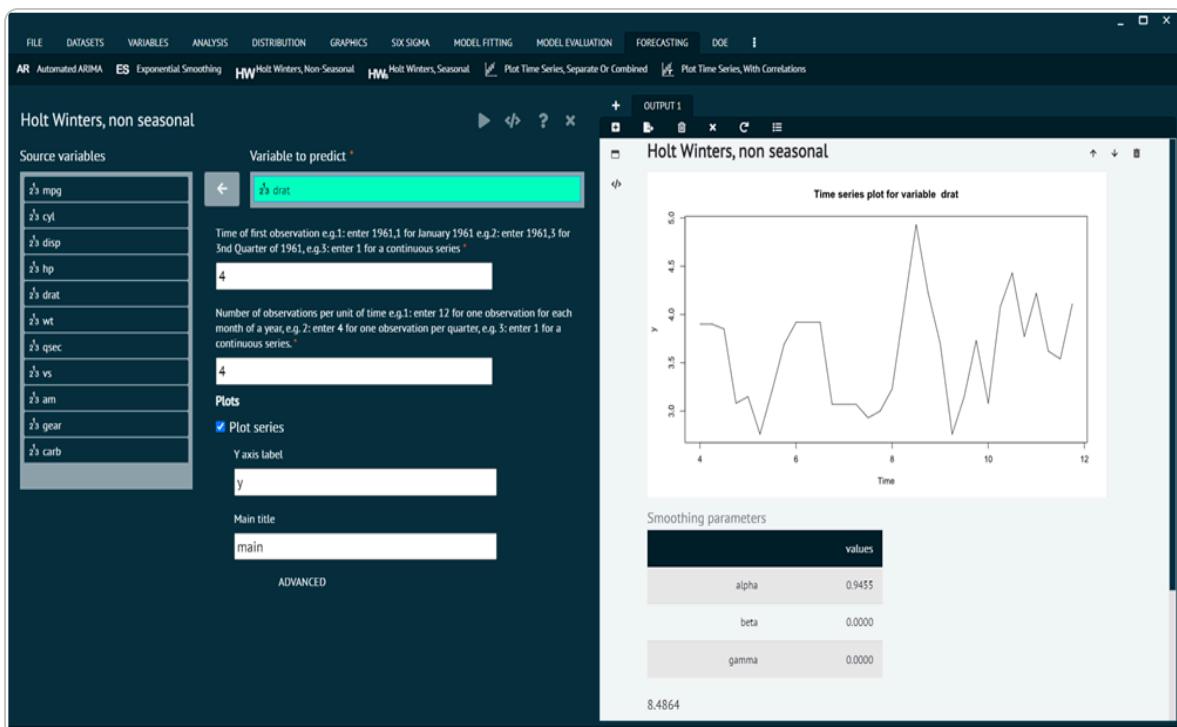
The user can choose additional options like plot options.

Holt Winters, Non-seasonal

Holt-Winters Exponential Smoothing is an extension of simple exponential smoothing that takes into account both level and trend components in a time series, and optionally, seasonality. When seasonality is not present in the data, the method is referred to as Holt-Winters Exponential Smoothing without seasonality, or simply non-seasonal Holt-Winters.

To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> Click on the Forecasting tab in main menu -> Select Holt winters, non-seasonal -> Choose variables to predict -> Write Time of first observation -> Write Number of observations per unit of time -> Execute.



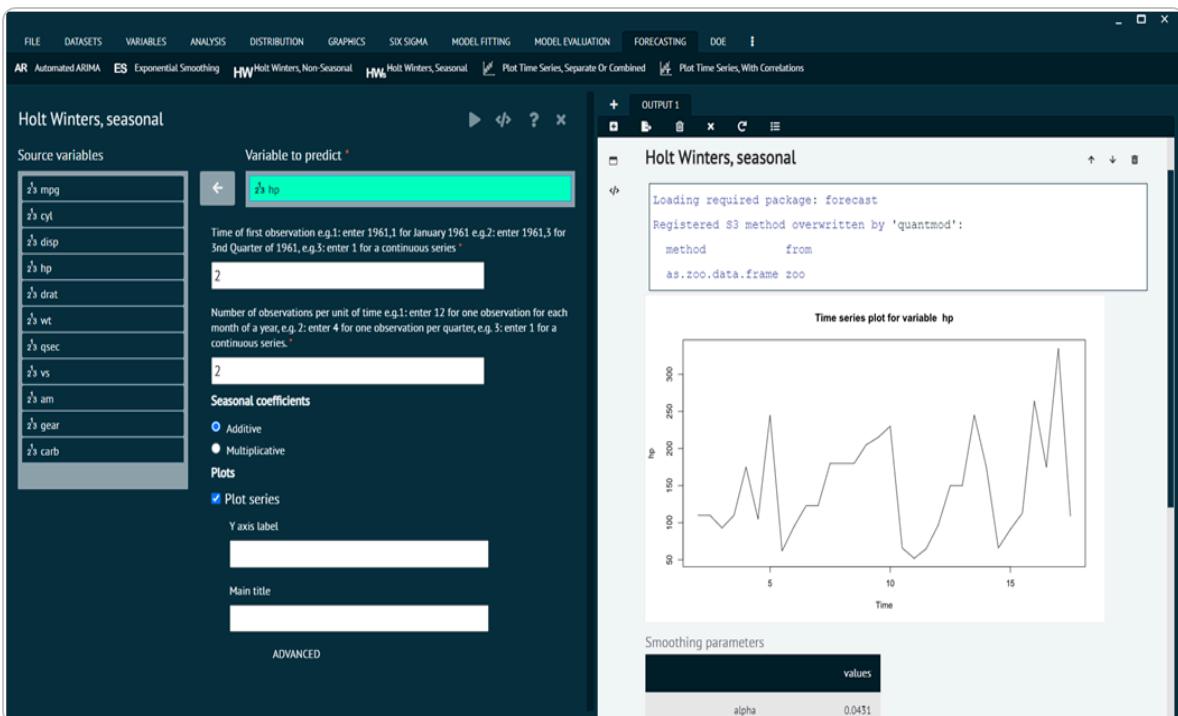
alt text

Holt Winters, Seasonal

The method involves initializing the model parameters, updating them with each new observation, and then using the model to make forecasts. The choice between additive and multiplicative methods depends on the nature of the seasonality in the data. Holt-Winters is a statistical method used for time series forecasting. It's an extension of the exponential smoothing method and is particularly useful for forecasting data with seasonality.

To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> Click on the Forecasting tab in main menu -> Select Holt winters, seasonal -> Choose variables to predict -> Write Time of first observation -> Write Number of observations per unit of time -> Execute.



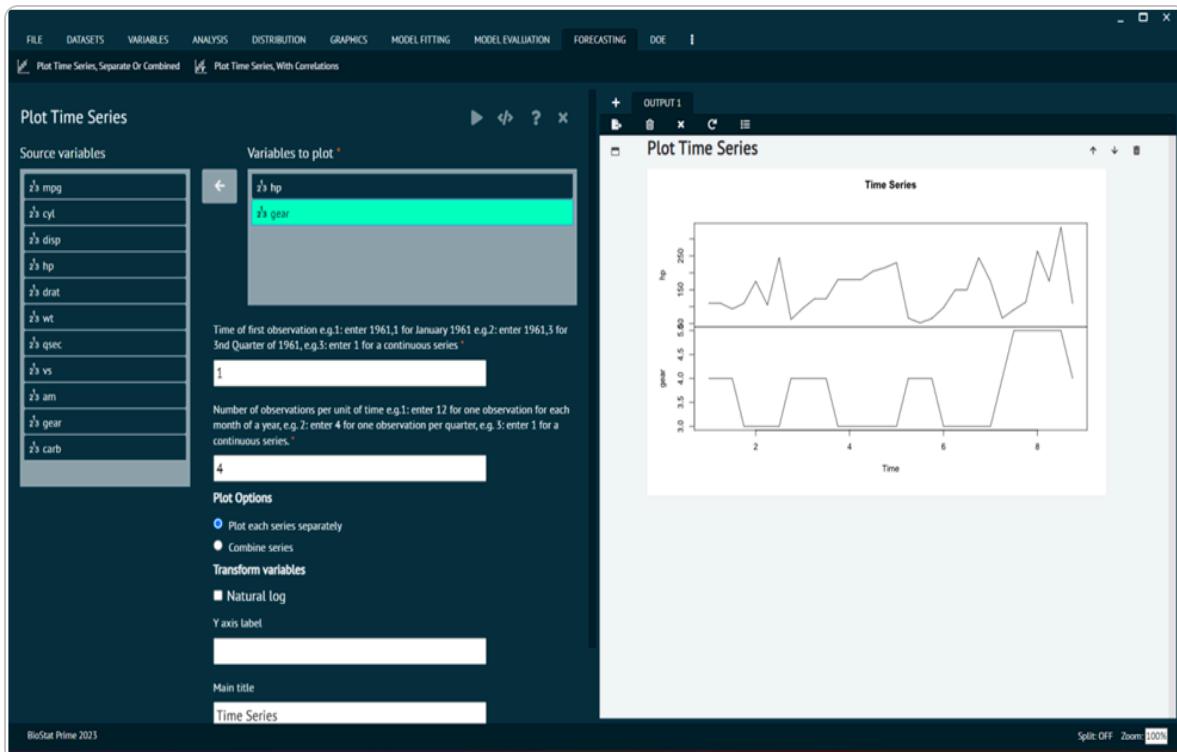
Plot Time Series, Separate OR Combined

Time series analysis is a crucial component of forecasting, especially when dealing with data that is collected sequentially over time. A time series is a set of observations or data points ordered chronologically. These data points could represent measurements, counts, values, or other observations taken at regular intervals.

To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> **Click on the Forecasting tab in main menu** -> **Select Plot Time Series** -> **Choose variables to plot** -> **Write Time of first observation** -> **Write Number of observations per unit of time** -> **Execute**.

The user can choose additional options like transform variables, and plot options to decide whether to plot each series separately or combine the series.



alt text

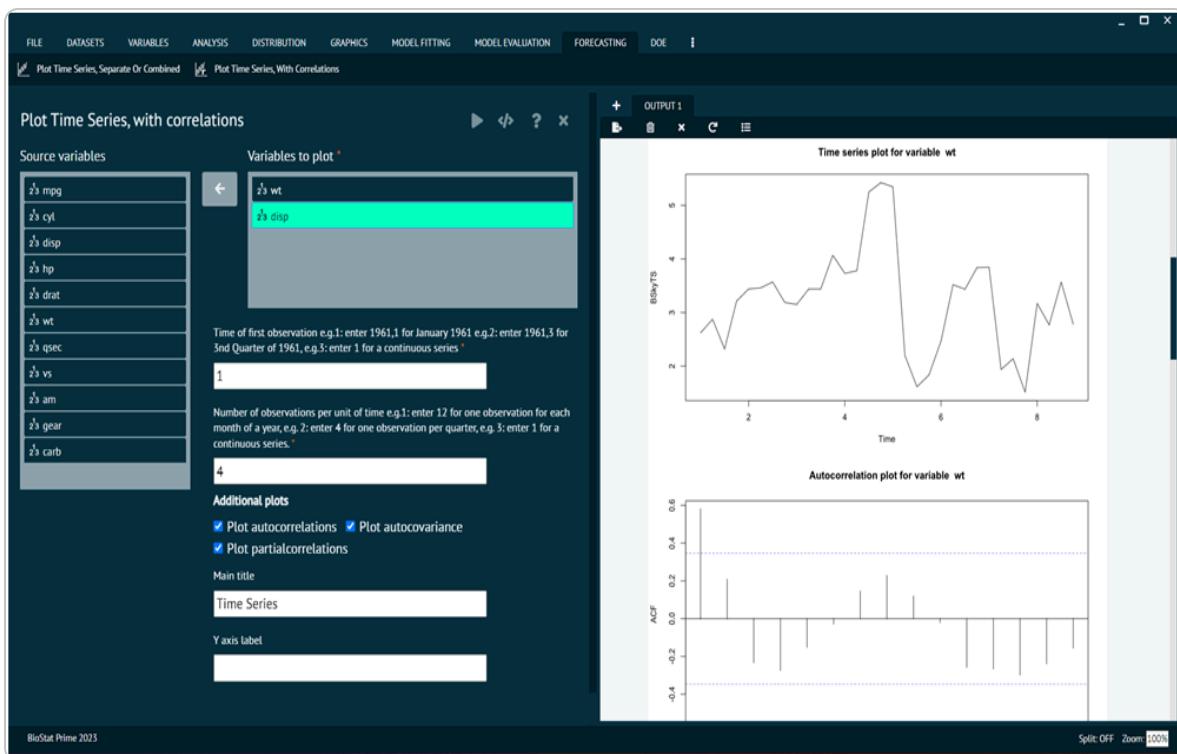
Plot Time Series with Correlations

When dealing with time series forecasting, the concept of correlation can be relevant, particularly in cases where there are multiple time series, and user want to understand the relationships between them. Correlation measures the strength and direction of a linear relationship between two variables. Correlation analysis can guide the selection of variables for inclusion in forecasting models. Variables with strong correlations might have predictive power and contribute to the accuracy of the model.

To analyse it in BioStat Prime user must follow the steps as given.

Load the dataset -> Click on the Forecasting tab in main menu -> Select Plot Time Series With Correlations -> Choose variables to plot -> Write Time of first observation -> Write Number of observations per unit of time -> Execute.

The user can choose additional plot options like autocorrelation, partial correlation, autocovariance. Apart from this user can decide the Y axis label and main title for the plot. In correlation user can opt for additional plots options to get more plots according to the needs and a clear comparison.



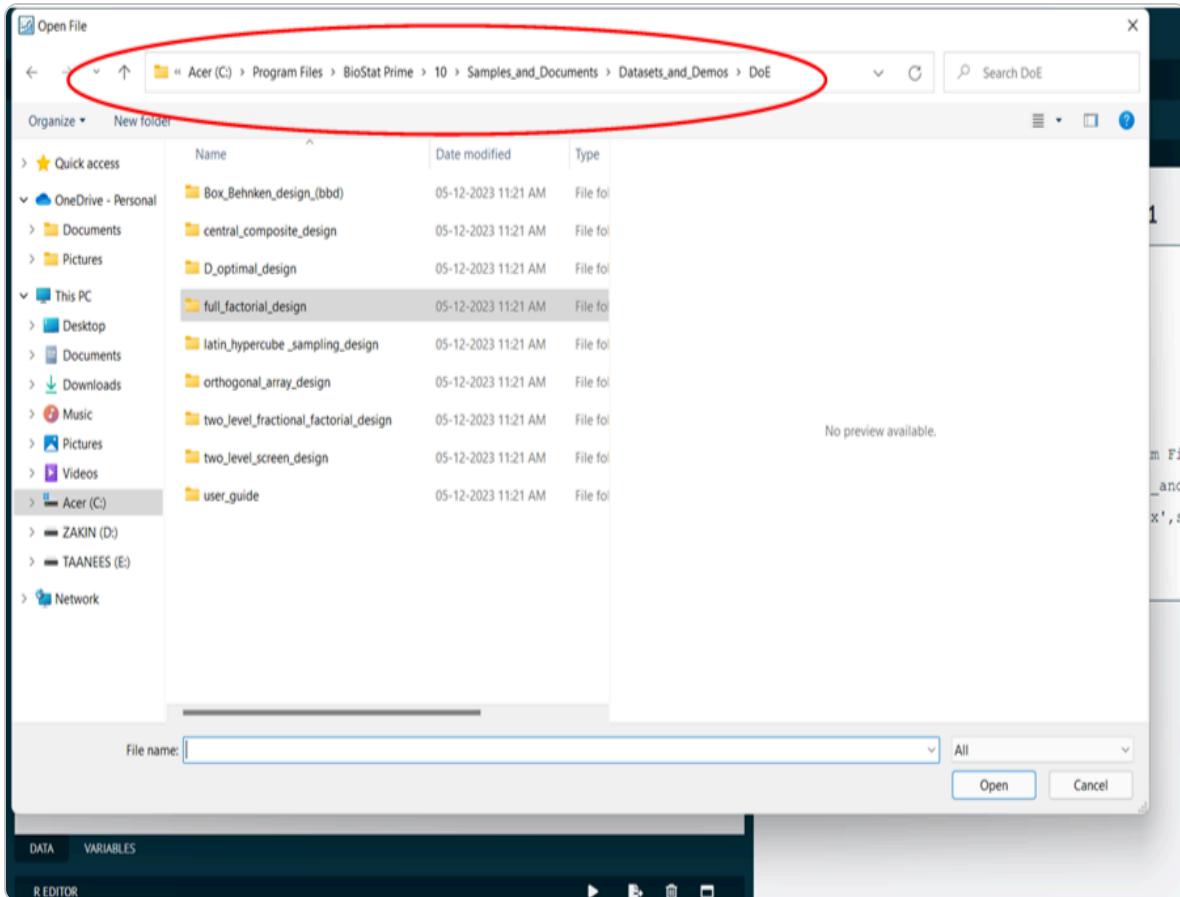
alt text

Design of Experiment

Using Design of Experiments (DOE) techniques, user can determine the individual and interactive effects of various factors that can influence the output results of your measurements. The primary goal of DOE is to optimize processes, improve product or system performance, and understand the relationship between input variables and the output response. User can also use DOE to gain knowledge and estimate the best operating conditions of a system, process or product.

To analyse it in BioStat user must follow the steps as given.

1. To create any Design under DOE -> create design menu, first user needs a dataset with factor details to create the design from.
2. To get started, choose one of the sample datasets (Excel file) provided in the sample dataset directory in your BioStat Prime install directory or user can create a factor detail table/dataset on the fly with DOE -> Create DoE Factor Details menu.



alt text

3. Once a dataset is opened with file open menu or created on the fly in step two above, go under DOE -> create design menu to create an appropriate design.

The screenshot shows the BioStat Prime software interface. On the left, there is a dataset grid titled "factor_grid_full_factorial_Design_Sheet1" with four columns: X1, X2, X3, and X4. The data rows are:

#	X1	X2	X3	X4
1	Algorithm	DeckArrange	Program	MemPages
2	LRUV	GROUP	Small	24P
3	FIFO	FREQY	Medium	20P
4	RAND	ALPHA	Large	16P

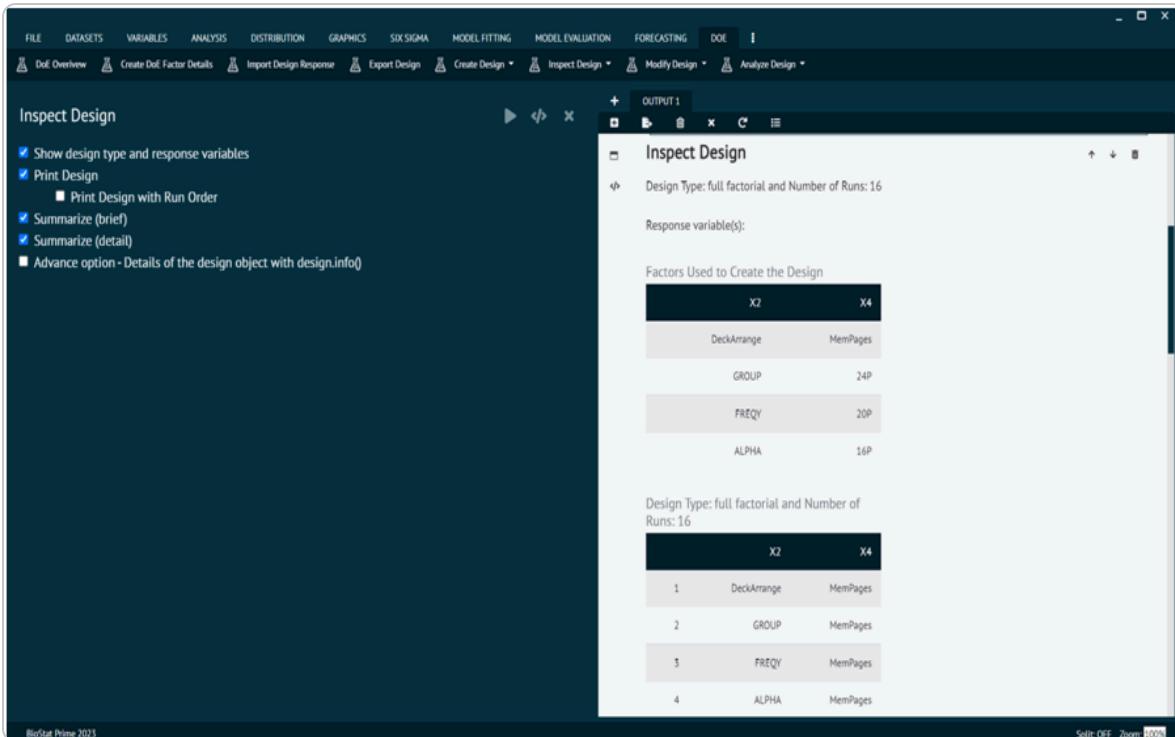
Below the grid are tabs for "DATA" and "VARIABLES". At the bottom is an "R EDITOR" tab. On the right side of the interface, there is an "OUTPUT" panel titled "Open Dataset: factor_grid_full_factorial_Design_Sheet1". It displays the following R code and output:

```
New names:
* `` -> `...1` 
* `` -> `...2` 
* `` -> `...3` 
* `` -> `...4` 

Successfully opened using:
[1] "readxl::read_excel(path='C:/Program Files/BioStat
Prime/10/Samples_and_Documents/Datasets_and_Demos/DoE/full_factorial_desig
n/factor_grid_full_factorial_Design.xlsx',sheet='Sheet1',
col_names=FALSE)"
```

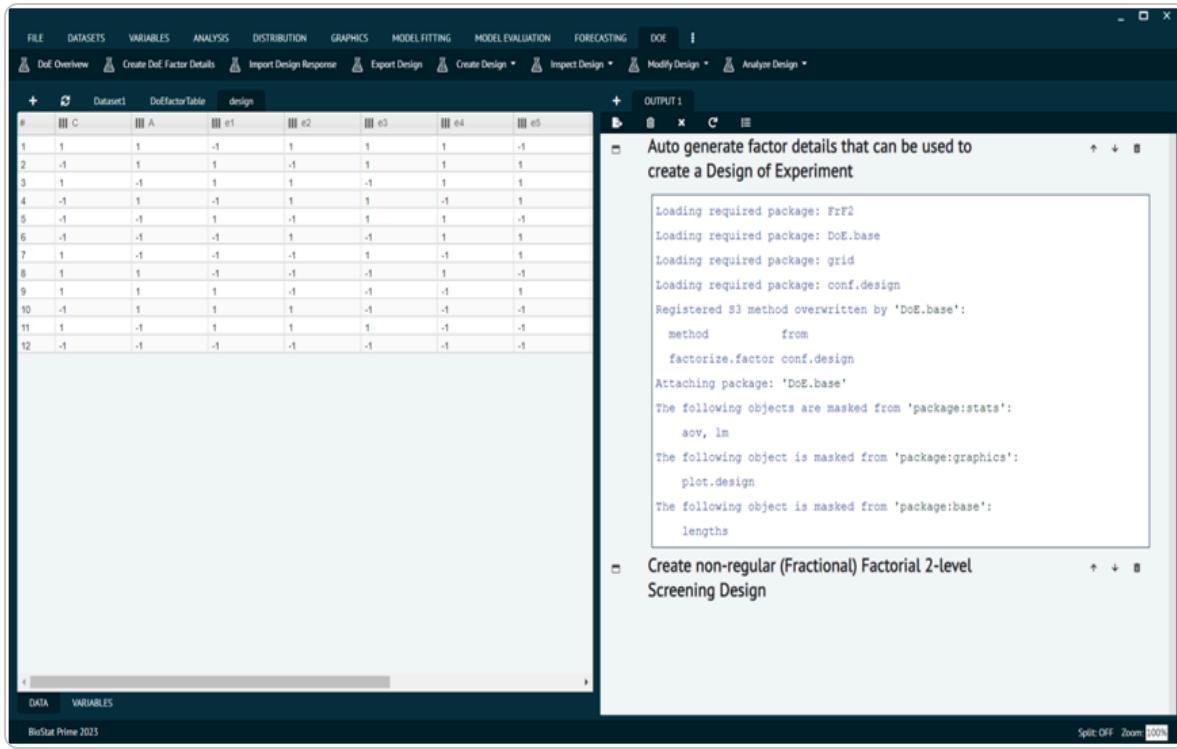
alt text

4. After a design is successfully created, it will show up on the dataset UI grid. Users can use DOE -> Inspect Design menu to inspect the design just created.



alt text

5. Export the design to a file system directory with DOE -> Export Design menu. It will automatically create three files (.csv, rda, .html) with the same names as the design dataset on the UI grid in the file directory path specified.
6. The csv file exported out in step 5 is meant to be used to set up the experiments in the real world as specified in the design to collect/record results for later analysis.
7. The results are recorded and added as separate column(s) called responses in the DoE vocabulary into the csv file. Do not change the csv file extension to any other file format. See the sample dataset directory for DoE, examples of csv files tagged as with respect to get the values for the result/response columns to copy from to create response columns in your own csv file that was exported as part of design export in step 5 above.
8. Import the csv design file with response column(s) added back to BioStat Prime app with DOE -> Import Design Response menu.

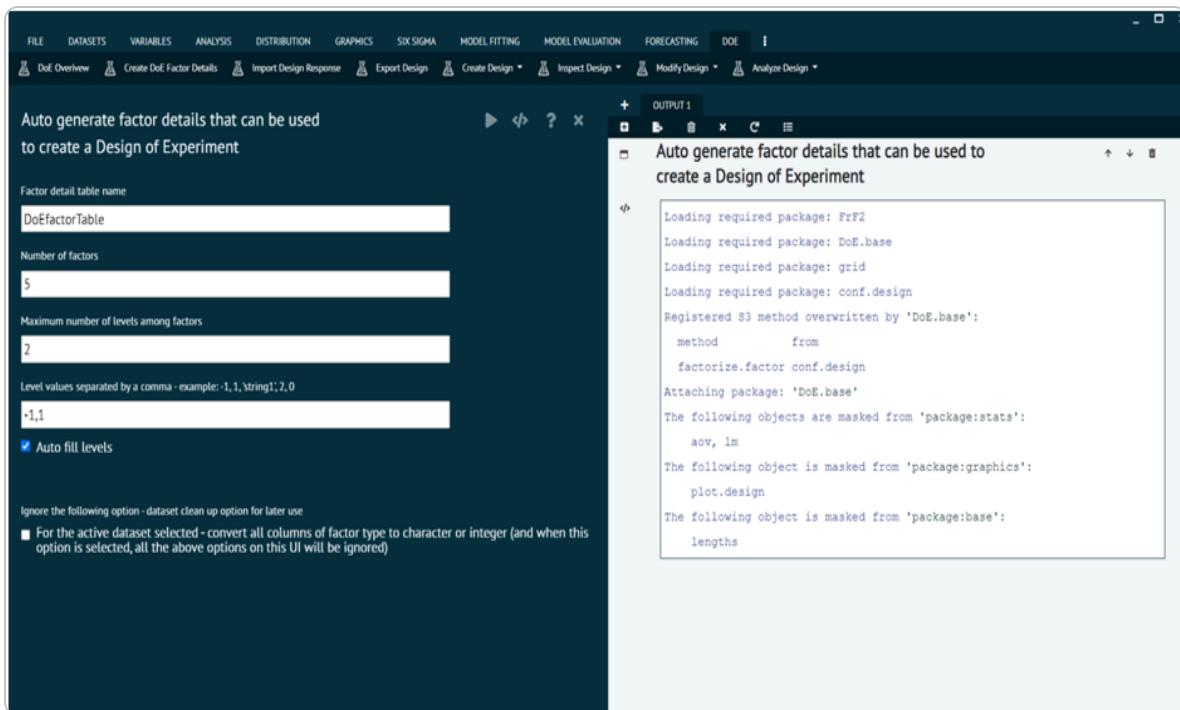


alt text

9. To import the design csv file in step 8, the original design (that was exported) needs to be available in the dataset UI grid. If it is not available, use the file open menu to load the design .rda file that was created as part of the design export in step 5.
10. After the csv file is successfully imported against the right/active design dataset and created a new design with the response column(s), user can use DOE -> Inspect Design menu to inspect the design with response column(s).
11. Now the design with response column(s) is ready for analysis with DOE -> Analyze Design menu with various analysis methods e.g. Linear model, Response Surface model, etc.
12. The datasets to use to test the DoE dialog will be indicated in BioStat Prime DoE dialog help (?). User may find all sample DoE datasets in the installation directory of BioStat Prime.

The various sub menus available in DoE menu are explained in up-coming section.

Create DoE Factor Details



alt text

The screenshot shows a software application window with a dark header bar. The header contains several icons and text labels: 'DoE Overview', 'Create DoE Factor Details', 'Import Design Response', 'Export Design', 'Create Design', and 'Inspect D...'. Below the header is a navigation bar with tabs: '+', 'Dataset1', 'mtcars', and 'DoEfactorTable'. The 'DoEfactorTable' tab is currently selected and highlighted with a dark background. The main area of the window displays a table with the following data:

#	2 ³ A	2 ³ B	2 ³ C	2 ³ D	2 ³ E
1	-1	-1	-1	-1	-1
2	1	1	1	1	1

alt text

Import Design Response

Import Design - must be a csv file with added response column(s) to the previously exported design

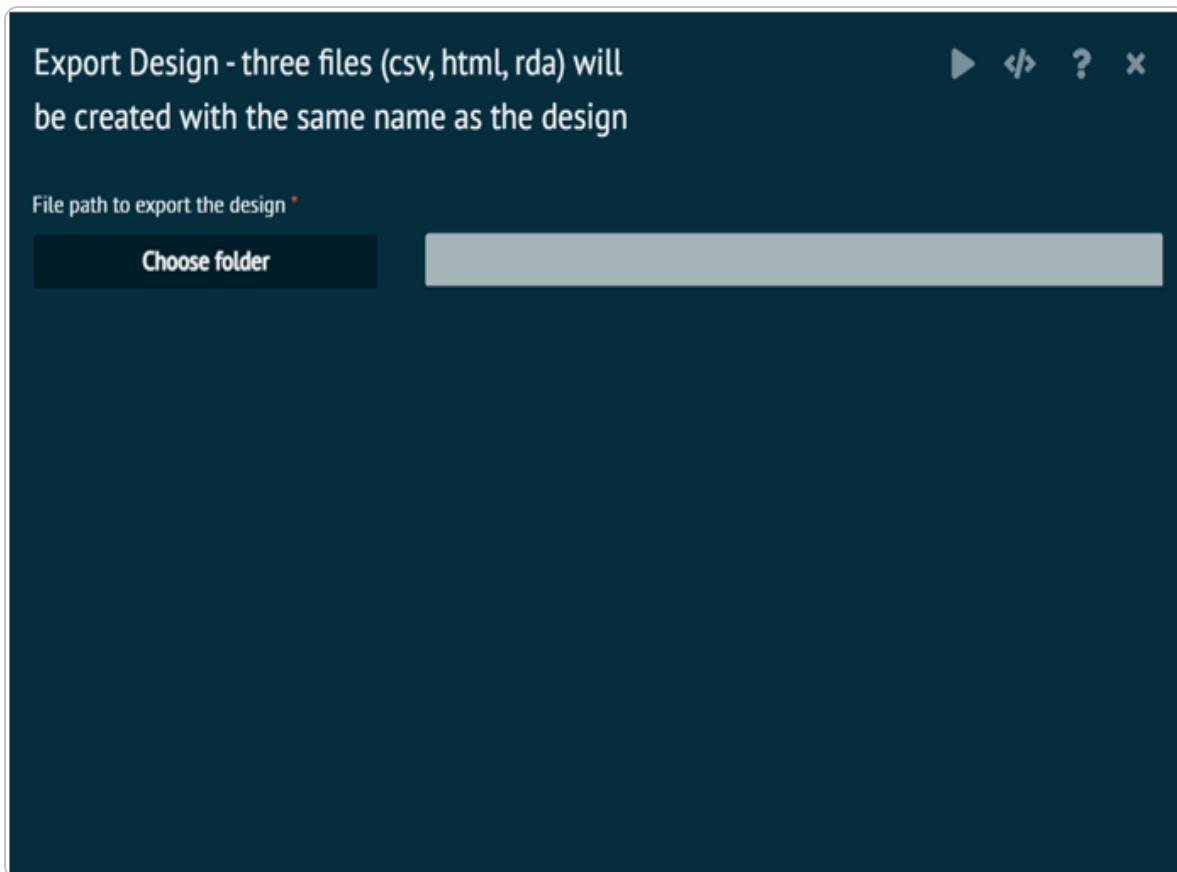
Name to create the design with response(s) *

File path for the design file (must be a csv file) that contains response column(s) *

Choose file

alt text

Export Design Response



alt text

Create Design

Create 2-level Screening Design

The screenshot shows the Minitab software interface with the 'DOE' tab selected. A dialog box titled 'Create non-regular (Fractional) Factorial 2-level Screening Design' is open. On the left, there is a data grid labeled 'Dataset1' with columns C, D, e1, e2, e3, e4, and e5, containing 12 rows of data. On the right, the 'Source variables' section lists variables z'A, z'B, z'E, z'C, and z'D. The 'Select variables' section has z'C and z'D highlighted with a green background. Below these, settings include 'Design name' (new), 'Number of runs (multiple of 4, >=8)' (12), 'Number of center points (if used, have minimum 2)' (0), 'Number of positions for center point distribution (have >1)' (2), 'Replications' (1), and 'Repeat only'. A note says 'You may not need to change the randomization settings' and 'Seed for randomization' is set to 1234.

alt text

Create Regular (Fractional) Factorial 2-Level Design

Create Regular (Fractional) Factorial 2-level Design

► ◄ ? ×

Source variables

2³ mpg
2³ cyl
2³ disp
2³ hp
2³ drat
2³ wt
2³ qsec
2³ vs
2³ am
2³ gear
2³ carb

Design name *

Select variables *



Size and randomization

Number of runs (if specified, it must be a power of 2 otherwise make it 0)

Number of blocks * 1

blocks may be aliased with 2fis

Number of center points (if used, have minimum 2) * 0

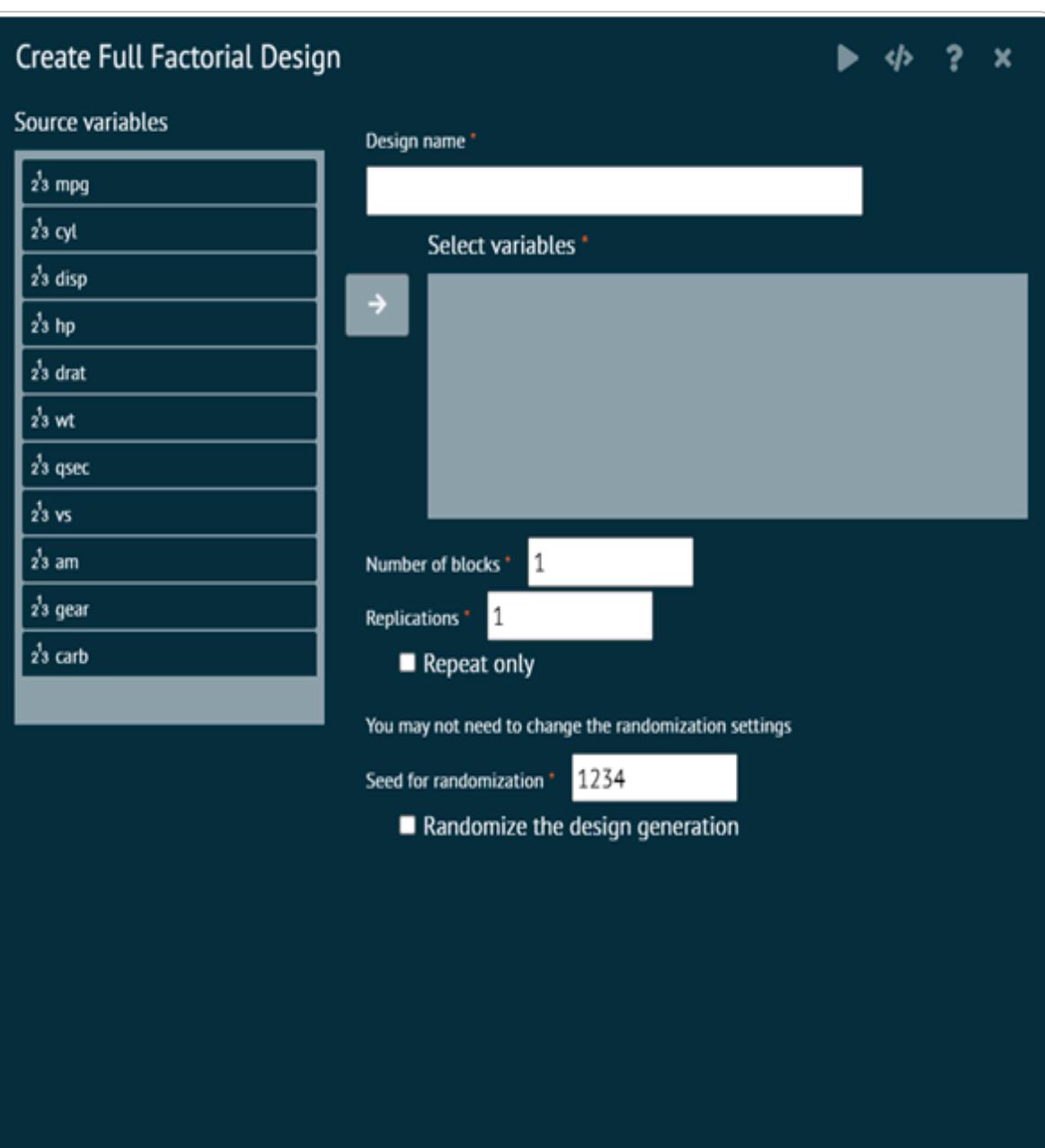
Number of positions for center point distribution (have >1) *

Replications * 1

Repeat only

alt text

Create Full Factorial Design



alt text

Create Orthogonal Array Design

Create Orthogonal Array Design

Source variables

13 mpg
13 cyl
13 disp
13 hp
13 drat
13 wt
13 qsec
13 vs
13 am
13 gear
13 carb

Design name *

Select variables *

→

Minimum number of runs (can be left blank)

Minimum number of residual degrees of freedom * 0

Replications * 1

Repeat only

You may not need to change the randomization settings

Seed for randomization * 1234

Randomize the design generation

(Optional) Orthogonal design array(example: from oacat\$name as L12.2.2.6.1)

Column optimization (default is order, other choices min3, min34, min3.rela, min34.rela, minRPFT, minRelProjAberr)

alt text

Create D-Optional Design

Create D-Optimal Design

▶ ⌂ ? ×

Source variables

2³ mpg
2³ cyl
2³ disp
2³ hp
2³ drat
2³ wt
2³ qsec
2³ vs
2³ am
2³ gear
2³ carb

D-optimal Design name *

Create D-optimal design from an existing candidate design (full factorial, FrF2, Orthogonal, or Latin) - make sure this dialog is opened on the existing design on the data grid to choose the candidate design implicitly

Select variables (Ignored if candidate design is checked above) to create a D-optimal design not from an existing candidate design



Number of runs * 8

Formula - leave it default to include all factors in the model or type in a linear model formula e.g. ~quad()

~.

Number of optimization Repeats * 5

Number of blocks * 1

Name of the block *

alt text

Create Central Composite (Quantitative) Design

Create Central Composite (Quantitative) Design
from an existing FrF2 Design

Source Datasets

Design name *

Select an existing FrF2 (Quantitative) design *

→

Number of center points, or two numbers separated by comma (for cube and the star portion)

4

Name of the block

Block.ccd

Number of star points(alpha)

orthogonal

You may not need to change the randomization settings

Seed for randomization * 1234

Randomize the design generation

alt text

Create Box-Behnken (Quantitative) Design

Create Box-Behnken (Quantitative) Design

Source variables

13 mpg
13 cyl
13 disp
13 hp
13 drat
13 wt
13 qsec
13 vs
13 am
13 gear
13 carb

Design name *

Select variables *

→

integer number of center points for each block * 4

Name of the block

Block

You may not need to change the randomization settings

Seed for randomization * 1234

Randomize the design generation

alt text

Create Latin Hypercube(Quantitative) Design

Create Latin Hypercube Design (for Quantitative Factors)

▶ ⌂ ? ×

Source variables

23 mpg
23 cyl
23 disp
23 hp
23 drat
23 wt
23 qsec
23 vs
23 am
23 gear
23 carb

Design name *

Select variables *



Size and randomization

Number of runs *

20

Number of decimal places *

2

You may not need to change the randomization settings

Seed for randomization *

1234

Randomize the design generation

latin hypercube sampling designs (check lhs or DiceDesign packages for other types) *

optimum

alt text

Create Taguchi Parameter Design

Create Taguchi Style Inner-Outer Parameter Design

▶ ⌂ ? ×

Source Datasets

mtcars
abbey

Design name *

Inner Design (must have been randomized already) *



Outer Design *



Direction - Generate Design in Long or Wide format

Long format

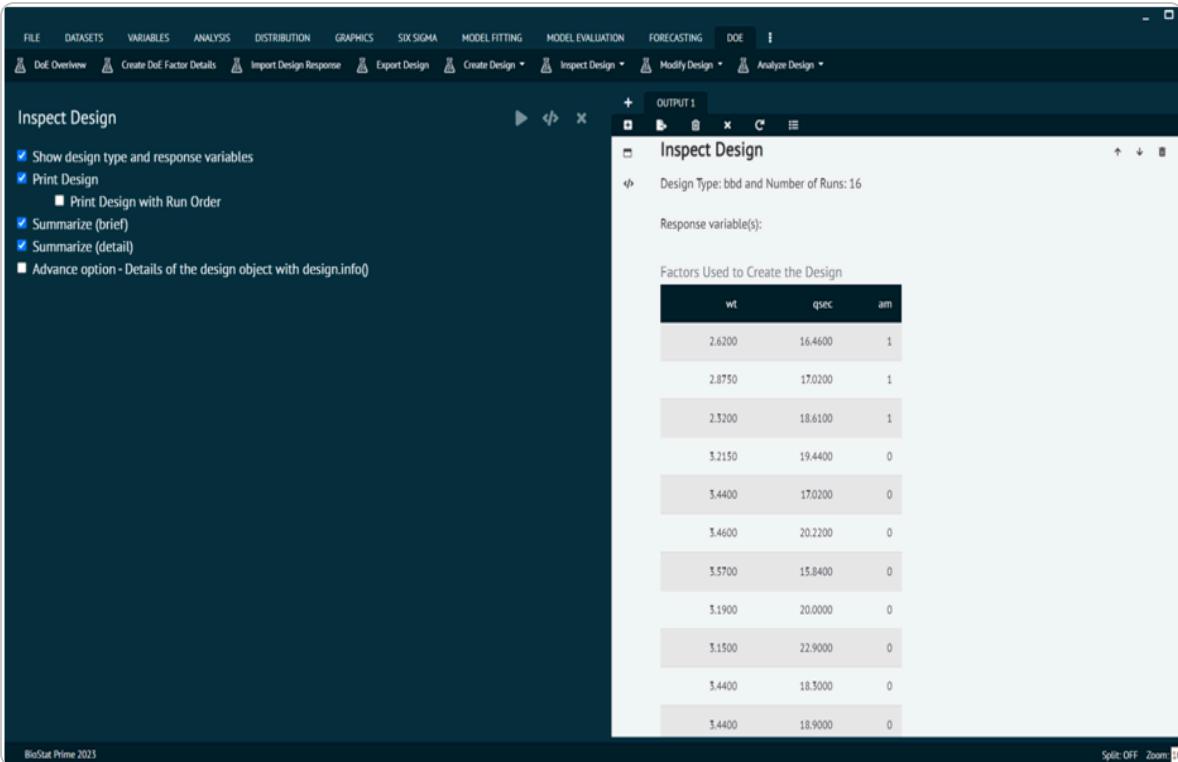
Wide format

Leave it blank or specify one or more response names (separated by comma without any quote or space in between names)

alt text

Inspect Design

Inspect design



The screenshot shows the BioStat Prime 2023 software interface. The main menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION, FORECASTING, DOE, and other options like Import Design Response, Export Design, Create Design, Inspect Design, Modify Design, and Analyze Design. A sub-menu for 'Inspect Design' is open, listing several options with checkboxes: Show design type and response variables (checked), Print Design (checked), Print Design with Run Order (unchecked), Summarize (brief) (checked), Summarize (detail) (checked), and Advance option - Details of the design object with design.info (unchecked). To the right, a window titled 'Inspect Design' displays the message 'Design Type: bbd and Number of Runs: 16'. It also shows a section for 'Response variable(s):' and a table titled 'Factors Used to Create the Design' with columns 'wt', 'qsec', and 'am'. The table contains 11 rows of data:

wt	qsec	am
2.6200	16.4600	1
2.8750	17.0200	1
2.3200	18.6100	1
3.2150	19.4400	0
3.4400	17.0200	0
3.4600	20.2200	0
3.5700	15.8400	0
3.1900	20.0000	0
3.1500	22.9000	0
3.4400	18.3000	0
3.4400	18.9000	0

alt text

+

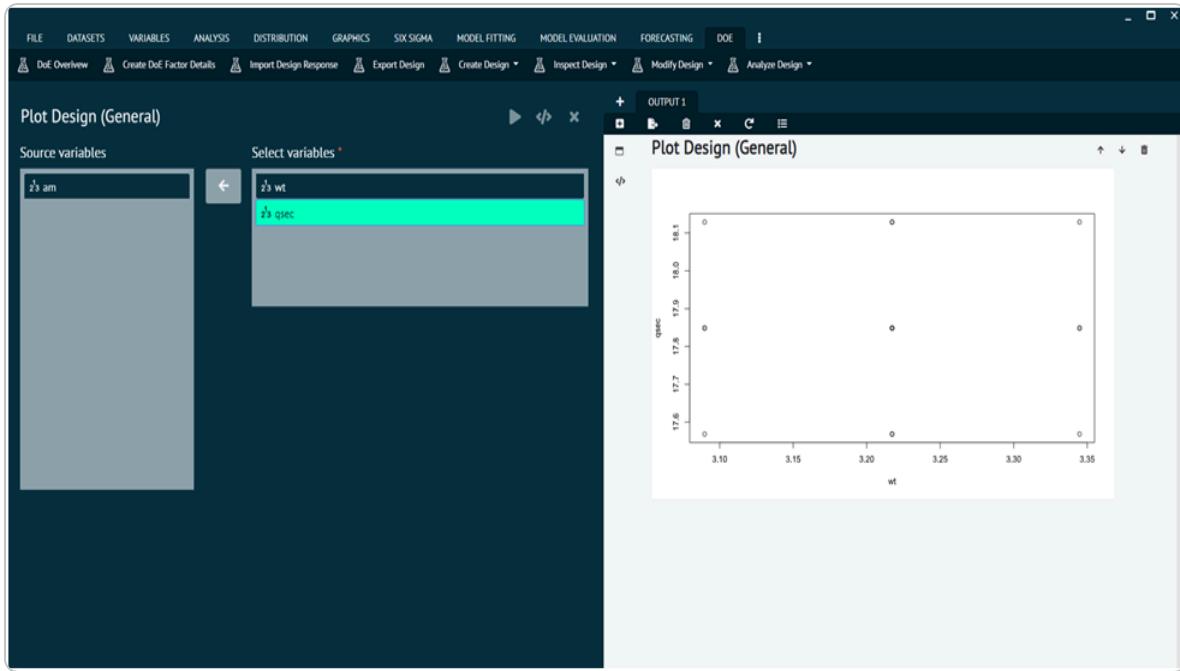
OUTPUT 1

Design Type: bbd and Number of Runs: 16

	wt	qsec	am
1	3.0898	17.5687	NaN
2	3.3448	17.5687	NaN
3	3.0898	18.1287	NaN
4	3.3448	18.1287	NaN
5	3.0898	17.8487	NaN
6	3.3448	17.8487	NaN
7	3.0898	17.8487	NaN
8	3.3448	17.8487	NaN
9	3.2173	17.5687	NaN
10	3.2173	18.1287	NaN
11	3.2173	17.5687	NaN
12	3.2173	18.1287	NaN
13	3.2173	17.8487	NaN

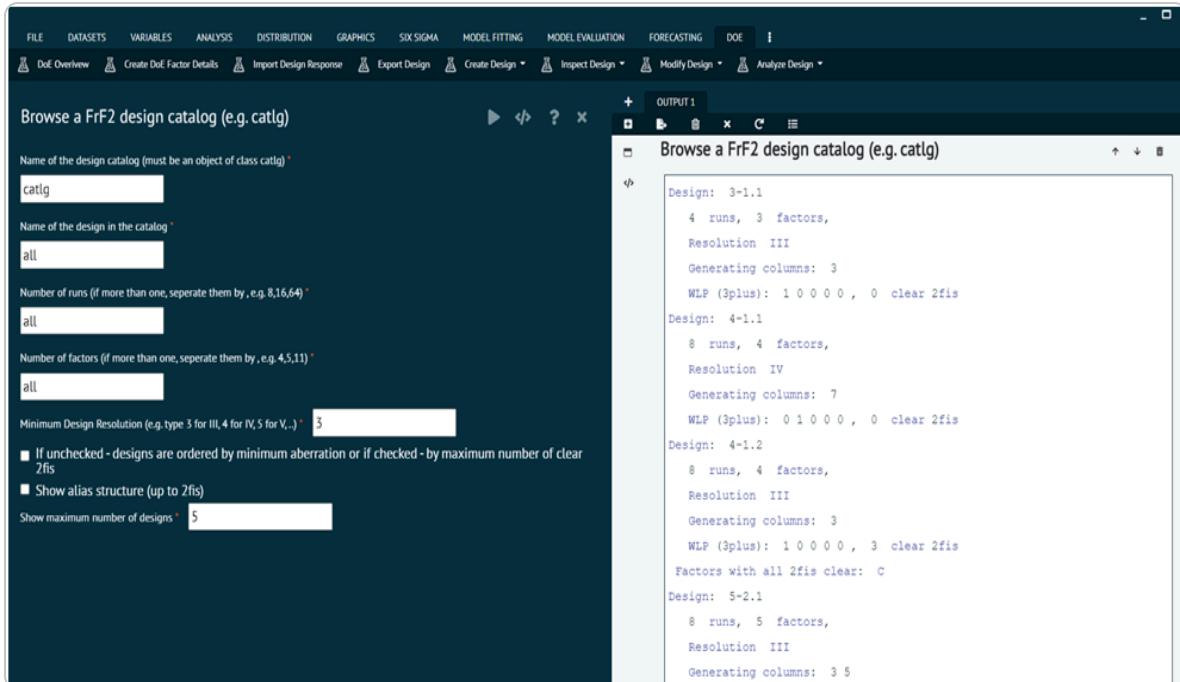
alt text

Plot Design



alt text

Browse FrF2 Design Catalog



alt text

Browse Orthogonal Design Catalog

The screenshot shows a software application window titled "Browse the Orthogonal Array (oacat) design catalog". The interface has a dark theme with a top navigation bar containing various menu items like FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION, FORECASTING, and DOE. Below the navigation bar are several tool icons. The main area is divided into two sections: a left panel for input parameters and a right panel for displaying results.

Left Panel (Input Parameters):

- Name of the Orthogonal Array design in the catalog (e.g. L18.3.6.6.1):
- Number of runs or a 2-element vector e.g. 4,16 with a minimum and maximum for the number of runs:
- Number of levels (separate them by , e.g. 3,2,5):
- Number of factors (separate them by , e.g. 4,2,1):
- Show all array quality metrics with the resulting arrays;
- Show maximum number of designs:

Right Panel (Output Results):

OUTPUT 1

Browse the Orthogonal Array (oacat) design catalog

71 resolution IV or more arrays found,
the first 5 are listed

	name	nruns	lineage
1	L27.3.4	27	
2	L32.2.9	32	
3	L32.2.16	32	
4	L32.2.4.4.2	32	
5	L40.2.6.5.1	40	

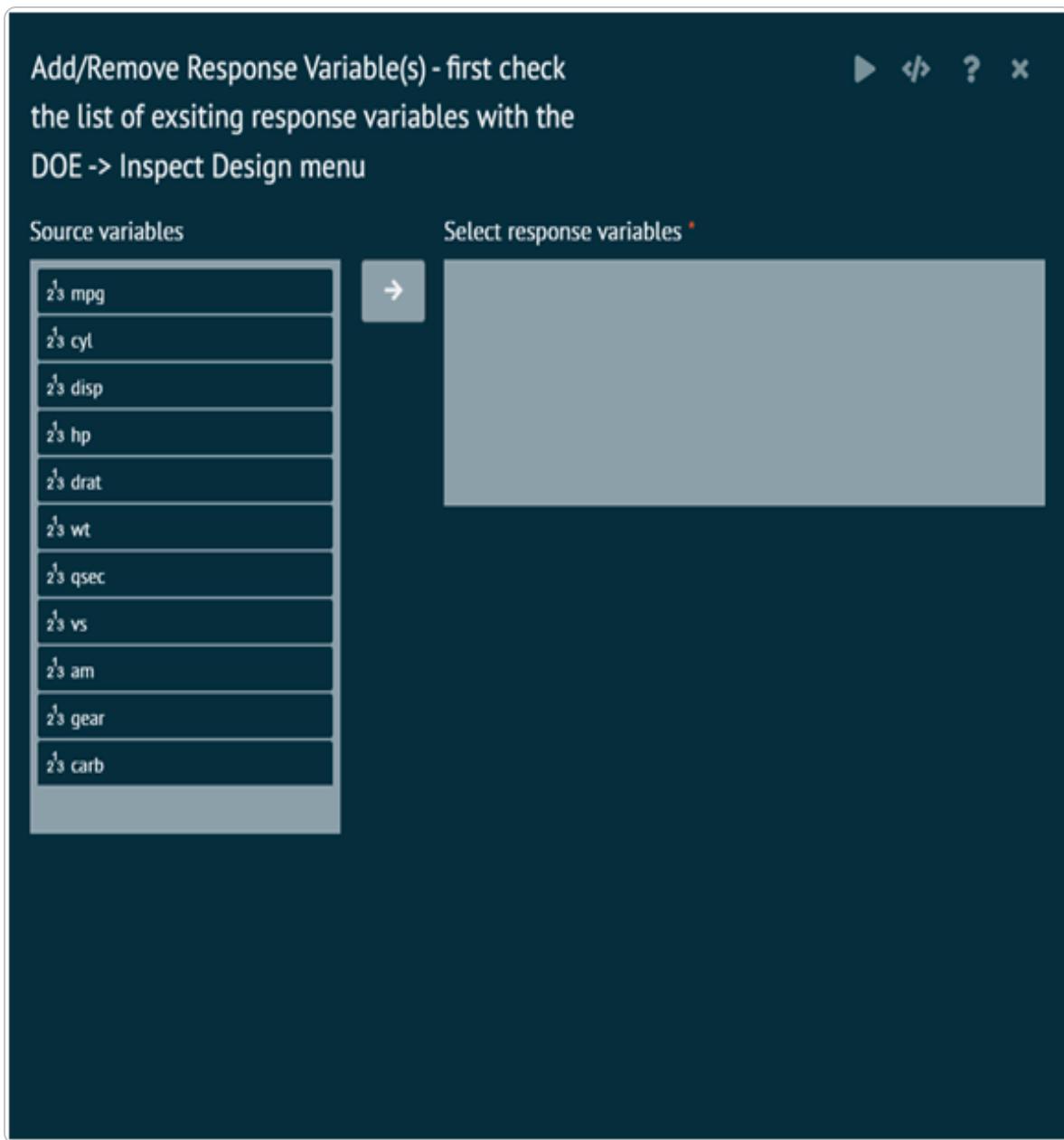
1837 orthogonal arrays found,
the first 5 are listed

	name	nruns	lineage
1	L4.2.3	4	
2	L6.2.1.3.1	6	
3	L8.2.7	8	2~4;4~1;:(4~1!2~3;)
4	L8.2.4.4.1	8	
5	L9.3.4	9	

alt text

Modify Design

Add/Remove Response



alt text

Add Centerpoint 2-Level Design (Quantitative)

Add centerpoint to a 2-level Design
(quantitative) with no prior centerpoint) - the
selected design is the active design on the
dataset UI grid



Modified Design name with centerpoints *

Number of center points (if used, must be minimum 2) *

0

Number of positions for center point distribution (must be >1) *

0

alt text

Analyse Design

Design Analysis-Linear Model

The screenshot shows the 'Design of Experiments analysis with Linear Model' interface. In the 'Source variables' list on the left, items like 'z\\$mpg', 'z\\$cyt', 'z\\$disp', etc., are listed. The 'Enter Model Name' field contains 'LinearRegModel1'. The 'Response (dependent) variable' is set to 'z\\$hp'. Under 'Independent variable(s)', 'z\\$am' and 'z\\$vs' are selected. A note at the bottom says 'Degree (leave it blank or type 2 for the linear model with main effects and 2-factor interactions)' followed by three checkboxes: 'Ignore intercept (if checked, then do not check the specific options below for the 2-level Factor Design)', 'All effects plot (generated only when Degree is not specified)', and 'Plot residuals vs fitted, normal Q-Q, scale-location and residuals vs leverage'. The 'Specify a variable with weights' section is empty. On the right, the 'OUTPUT' panel displays the R code used:

```
hp = alpha + beta1(am) + beta2(vs) + epsilon
```

 and the resulting output:

```
hp = 195.4481 - 17.1778(am) - 95.5021(vs)
```

 with the note 'Model: lm.default(formula = hp ~ am + vs, data = mtcars, na.action = na.exclude)'. Below this is the 'LM Summary' table:

Residual Std. Error	df	R-squared	Adjusted R-squared	F-statistic	numdf	dendf	p-value
48.1799	29	0.5581	0.5062	16.8890	2	29	1.3695e-05 ***

Below the summary is the 'Residuals' table:

Min	1Q	Median	3Q	Max
-87.2704	-20.4481	-4.1082	23.0540	156.7196

Finally, the 'Coefficients' table:

Estimate	Std. Error	t value	Pr(> t)	2.5 %	97.5 %

alt text

Design Analysis-Response Surface Model

Design of Experiments analysis with Response Surface Model (Quantitative)

Source variables

23 mpg
23 cyl
23 disp
23 hp
23 drat
23 wt
23 qsec
23 vs
23 am
23 gear
23 carb

Enter Response Surface model name *

ResponseSurfaceModel1

Response (dependent) variable *



Formula Builder:

Click on a button below and drag and drop variables to create an expression.
Clicking a selected button will toggle its state.

To insert at a position, place the cursor in that position and drag & drop/move variable(s).

Mouse over a button for help.

You cannot toggle the All N way button.

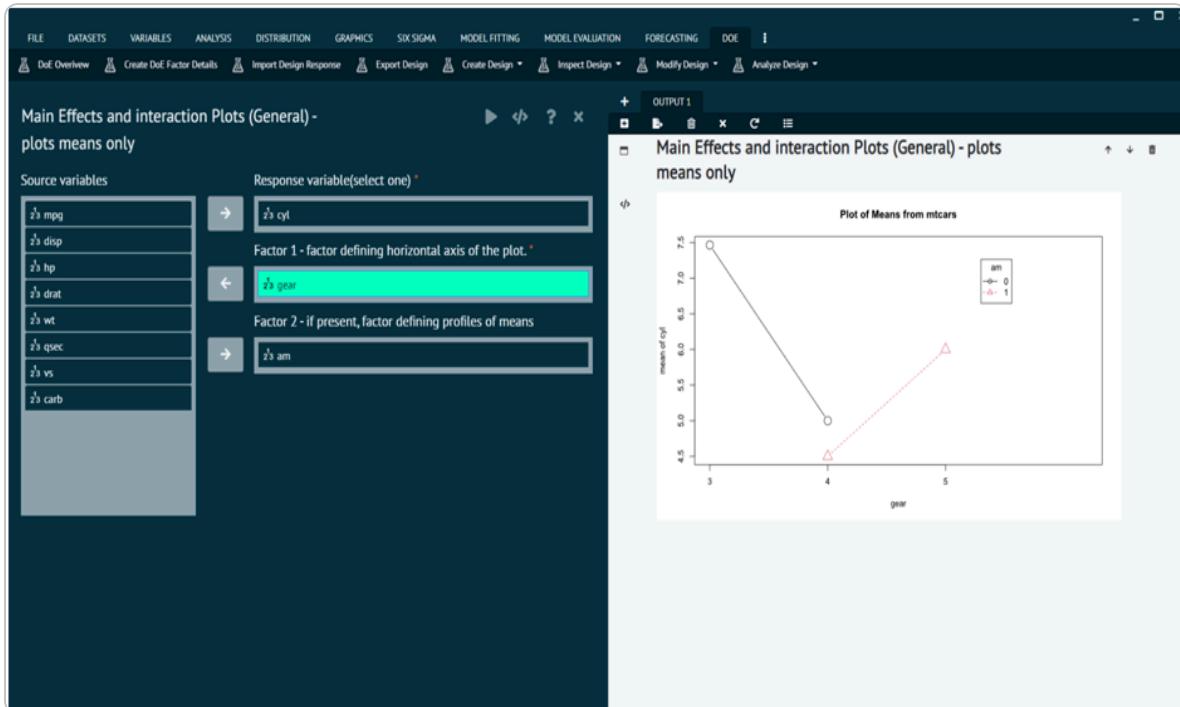
+	-	*	/	
()	%in%		:
All 2 ways	Polynomial terms	2		
df for splines	5	Polynomial degree	5	
B-spline	Natural spline			
Orthogonal polynomial	Raw polynomial			

Formula appears here

You can also enter Sdfvar1 var2 ... 1 for second-order effects and interactions

alt text

Main Effect And Interaction Plots (General)



alt text

Half Normal Plot For 2 levels

Daniel Plot - Effects Normal (or Half) Plot for 2 Level Factor

Source variables

- `2^3 mpg`
- `2^3 cyl`
- `2^3 disp`
- `2^3 hp`
- `2^3 drat`
- `2^3 wt`
- `2^3 qsec`
- `2^3 vs`
- `2^3 am`
- `2^3 gear`
- `2^3 carb`

Response variable(select one) *



Half normal plot



Label effects with codes instead of names

Enter significance level for labelling



Label significant effects only

alt text

Full Factor Analysis (in detail)

Full factorial analysis is a statistical method used in experimental design to study the effects of multiple factors on a response variable. It involves examining all possible combinations of factor levels in a systematic way. Two factors, each with two levels (A

and B), a full factorial design would involve testing all combinations (AA, AB, BA, BB). In experimentation, factorial experiments are highly prevalent. Most experiments are conducted using only two-level components, especially in industrial settings. It is crucial to have software that non-statisticians may use safely since subject-matter experts frequently design and carry out industrial experiments on their own without the assistance of a statistical specialist. Simultaneously, statisticians are frequently engaged in more significant experimental initiatives. A statistician greatly values assistance from robust software.

BioStat Prime aids this Statistical analysis technique by merging the powers of R language in this statistical method. The design of experiment section provides an extensive help to perform Full Factorial Analysis. BioStat Prime also provides some sample datasets to explore the functioning.

To analyse it in BioStat user must follow the steps as given.

Load the dataset (as specified above in DoE section) -> Click on the DoE tab in main menu -> Select Create Design -> Choose Create Full Factorial Design -> This leads to analysis techniques in the dialog -> In the dialog window select the options according to the requirements -> Execute.

The output will be represented in output window. The output window shows the message that Full Factorial Design has been created.

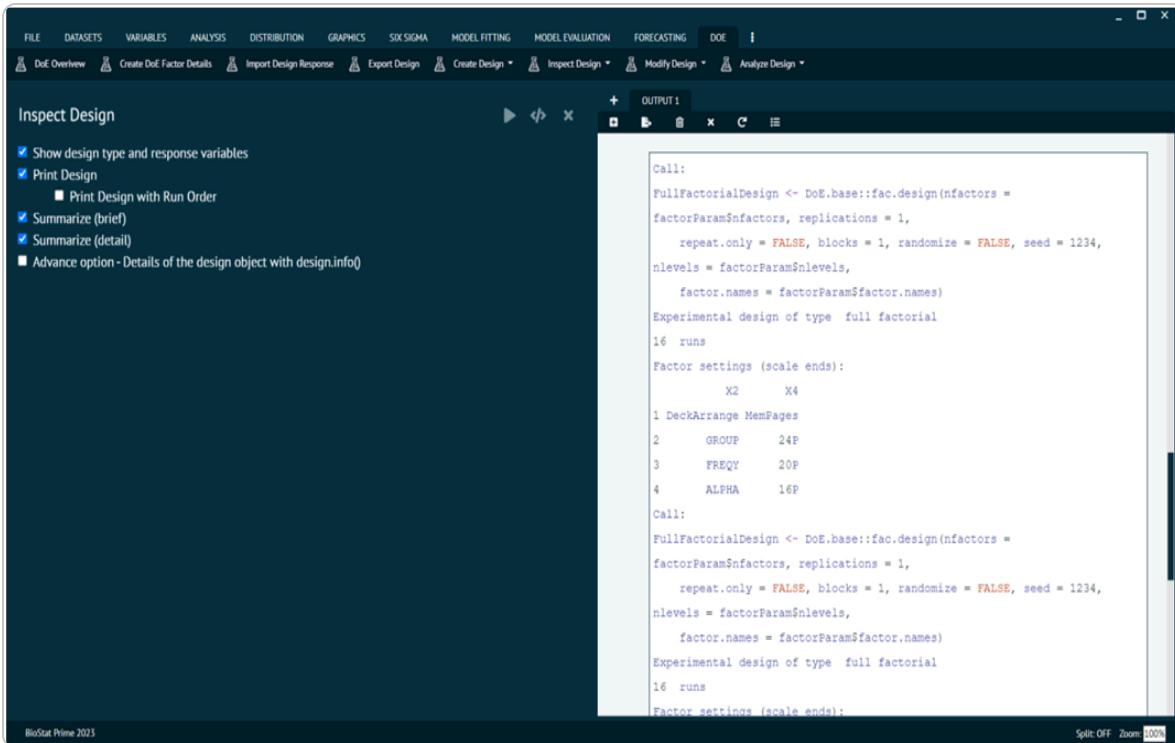
The screenshot shows the BioStat Prime 2023 software interface. On the left, a table titled "factor_grid_full_factorial_Design_Sheet1" displays 16 rows of data with columns X2 and X4. The data includes factors like DeckArrange, GROUP, FREQY, and ALPHA with their corresponding levels (e.g., MemPages, 24P, 20P, 16P). Below the table are tabs for "DATA" and "VARIABLES". On the right, an "OUTPUT 1" window shows the command used to open the dataset: `readxl::read_excel(path='C:/Program Files/BioStat Prime/10/Samples_and_Documents/Datasets_and_Demos/DoE/full_factorial_design/factor_grid_full_factorial_Design.xlsx', sheet='Sheet1', col_names=FALSE)`. It also shows the message "Successfully opened using:" followed by the R code. Below this, a progress bar indicates "creating full factorial with 16 runs ...". The bottom status bar shows "BioStat Prime 2023" and "Split OFF Zoom 100%".

alt text

The user can now inspect the design as shown below.

The screenshot shows the BioStat Prime 2023 software interface with the "Inspect Design" feature selected. On the left, a sidebar lists inspection options: "Show design type and response variables" (checked), "Print Design" (checked), "Print Design with Run Order" (unchecked), "Summarize (brief)" (checked), "Summarize (detail)" (checked), and "Advance option - Details of the design object with design.info" (unchecked). The main area displays the "Inspect Design" output. It shows the "Design Type: full factorial and Number of Runs: 16" and the "Response variable(s):". Below this, it lists the "Factors Used to Create the Design" with columns X2 and X4. The factors listed are DeckArrange, GROUP, FREQY, and ALPHA, each with their respective levels (MemPages, 24P, 20P, 16P). At the bottom, it shows the "Design Type: full factorial and Number of Runs: 16" again, followed by a table of factor levels for runs 1 through 4. The bottom status bar shows "BioStat Prime 2023" and "Split OFF Zoom 100%".

alt text



alt text

Features of BioStat Prime that enhance DOE

Randomization:

To remove bias and other source of extraneous variation which are not controllable, BioStat Prime randomly assigns material, people order in the experimental trials to be conducted.

Replication:

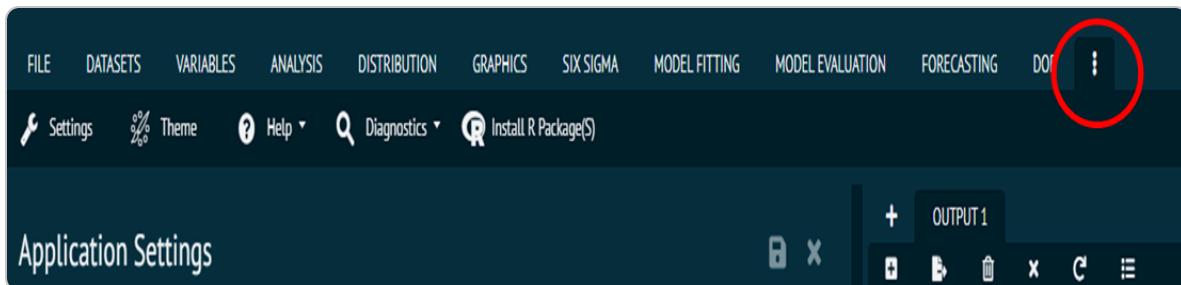
To increase the precision of estimate of experimental error, BioStat Prime provides repetition of basic experiment without changing factor settings.

Blocking:

To increase the efficiency of experimental design by decreasing experimental error, BioStat Prime breaks the experiment into homogenous segments (block) in order to control block variability.

Triple dots

This is the last section of main menu of the software that comprises 5 sub menus. It is represented by 3 dots. Functionality of each menu is discussed below.

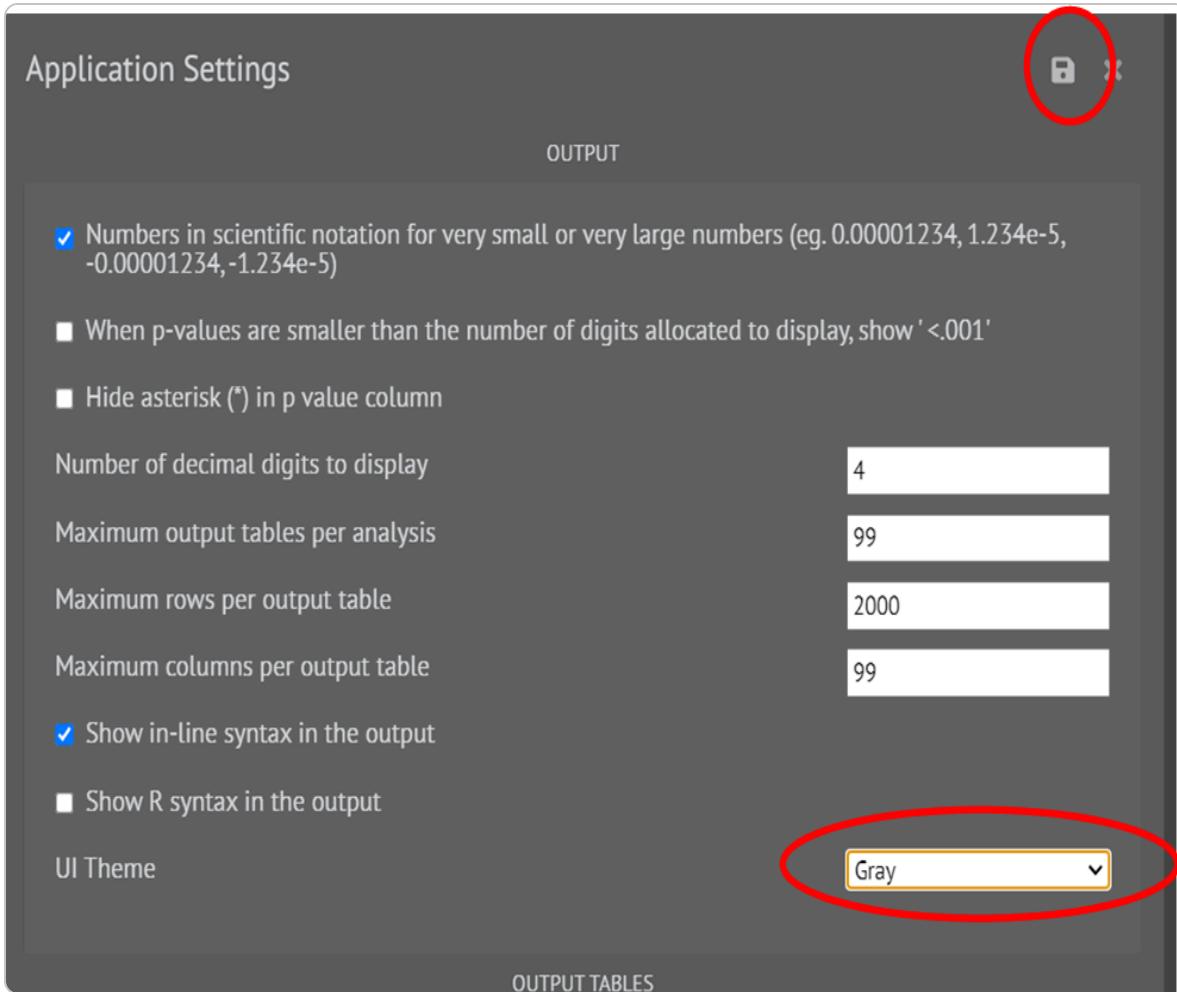


alt text

Settings

This section of the software provides the user with ability to modify the application settings according to user's personal requirements. This section has five subsections namely.

OUTPUT: Used to modify no. of rows and columns in output, no. of tables in output and other settings related to UI of the software.



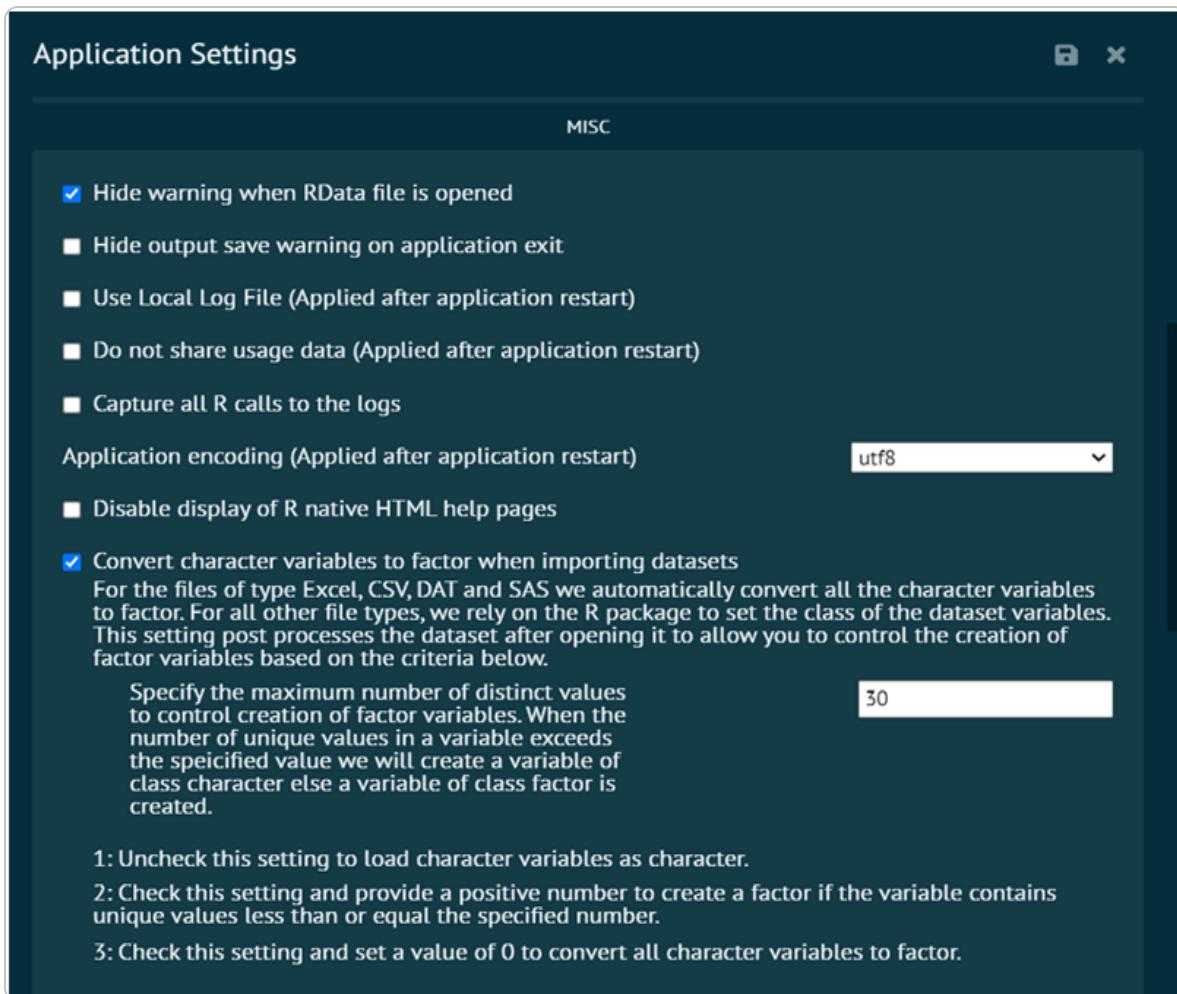
The icon at top right corner is used to save the changes.

OUTPUT TABLES: Used to modify settings related to theme, font, LaTeX of the tables that appear in the output.



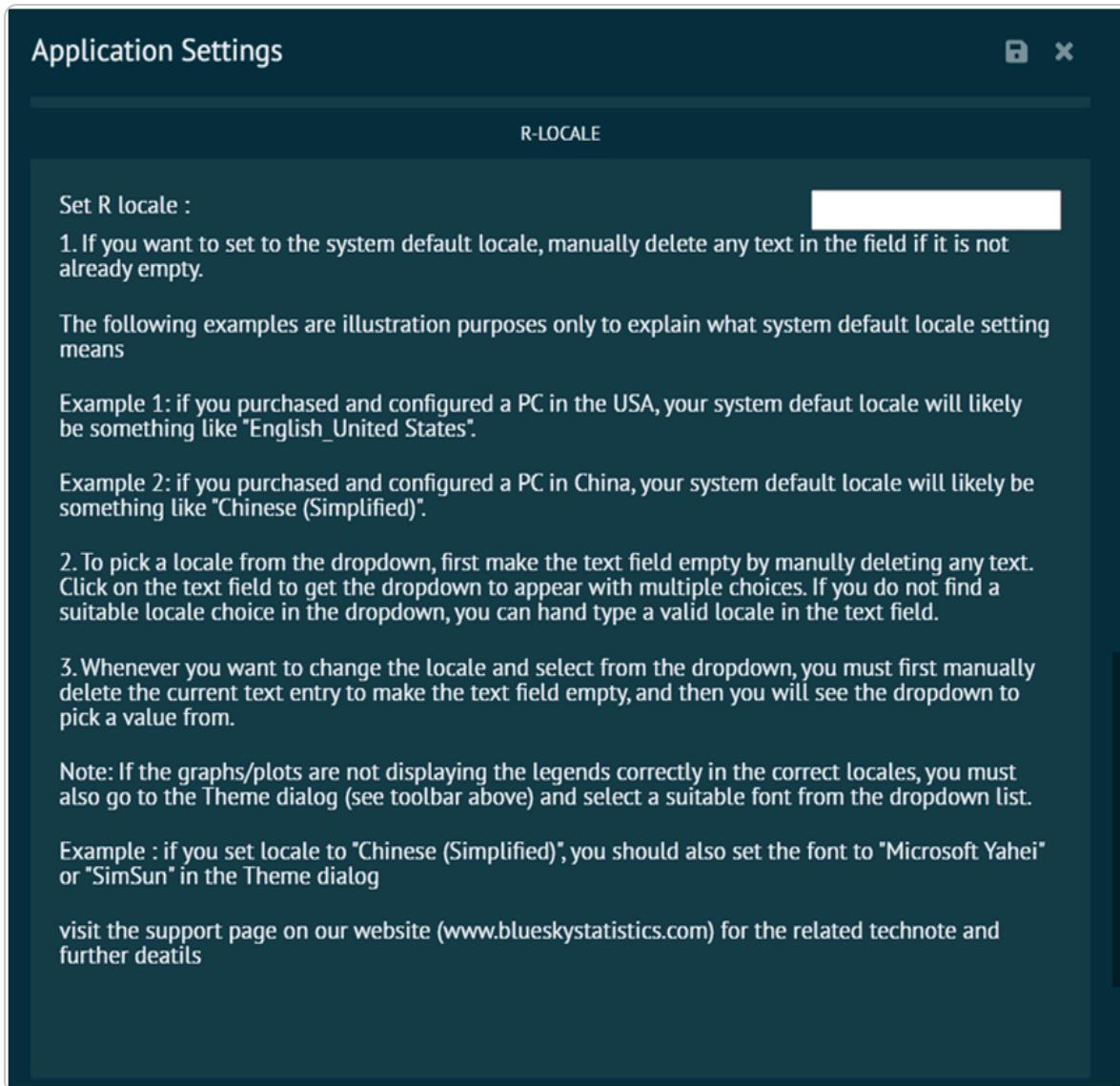
alt text

MISC: This section of the settings enables the user to modify miscellaneous settings of the application.



alt text

R-LOCALE: Used for language setting of R engine.



alt text

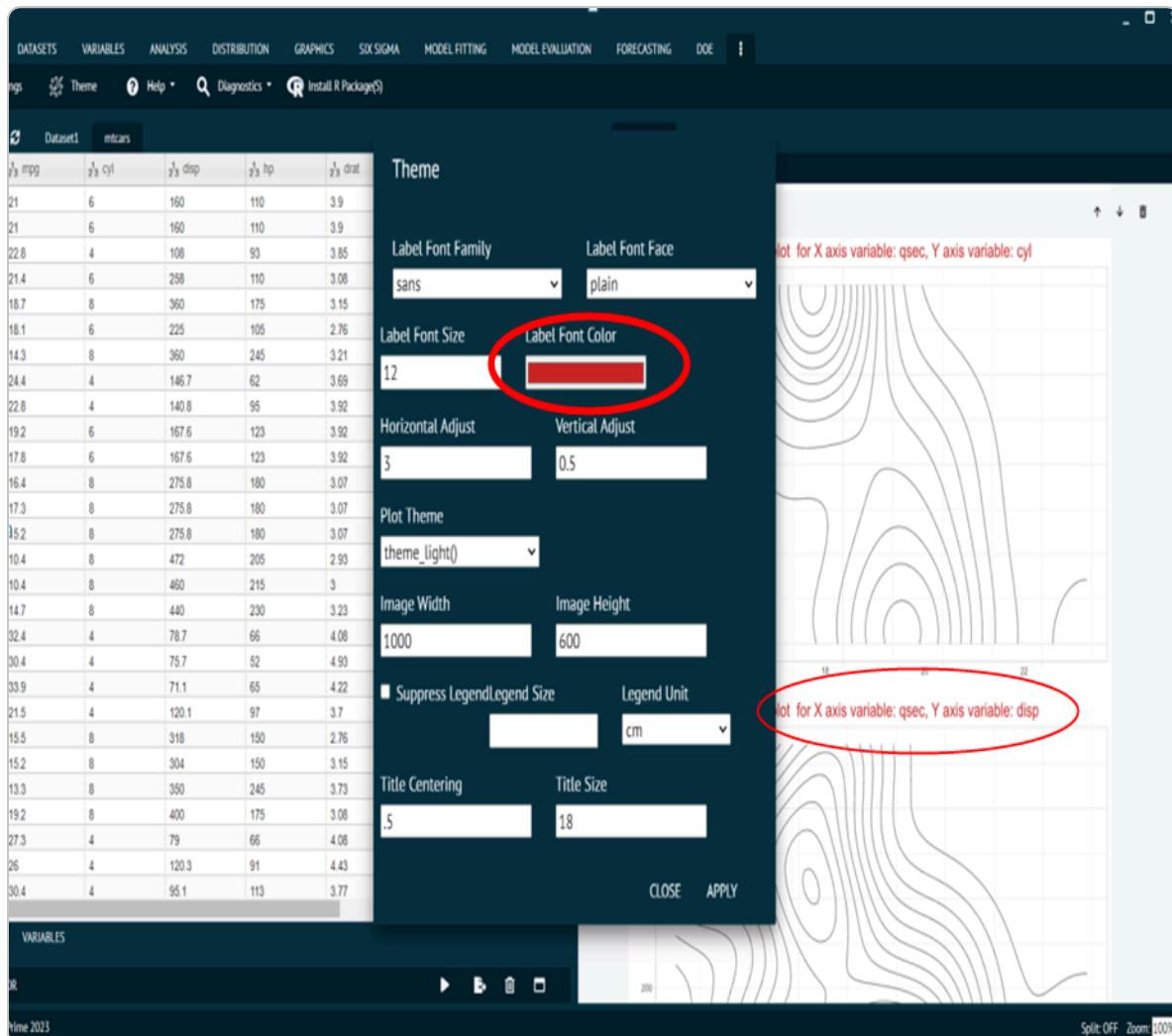
DATABASE: Enables the user to choose the location for database.



alt text

Themes

This section of the software provides the user with ability to customize the look of the output labels.



Help

This section of the software provides the user with the documentation regarding

1. Functions of R language used in the software. User just needs to enter the name of the function.
2. Packages of R language used in the software. User just needs to enter the name of the

package.

3. Help about the R version used in the software. User just needs to press the execute button.

The screenshot shows a software interface with a top navigation bar containing FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, and MODEL EVALUAT. Below the navigation bar is a toolbar with icons for Settings, Theme, Help, Diagnostics, and Install R Package(S). A modal window titled 'About' is open, displaying three options: R Function Help, R Package Help, and R Version. In the background, there is a data grid titled 'Dataset1' showing the 'mtc' dataset. The data grid has columns for #, mpg, cyl, drat, wt, and qsec. The first few rows of data are visible.

#	mpg	cyl	drat	wt	qsec
1	21	6	3.9	2.62	16.46
2	21	6	3.9	2.875	17.02
3	22.8	4	3.85	2.32	18.61
4	21.4	6	3.08	3.215	19.44
5	18.7	8	3.15	3.44	17.02

alt text

Diagnoses

This section of the software provides the user the information regarding.

1. Details about selected package. The user needs to select a package and execute the dialog. The details about the package will appear in the output window as shown in the picture.

R Package Details

```
> Package: abind
Version: 1.4-5
Date: 2016-06-19
Title: Combine Multidimensional Arrays
Author: Tony Plate <tplate@acm.org> and Richard Heiberger
Maintainer: Tony Plate <tplate@acm.org>
Description: Combine multidimensional arrays into a single array. This
is a generalization of 'cbind' and 'rbind'. Works with
vectors, matrices, and higher-dimensional arrays. Also
provides functions 'adrop', 'asub', and 'afill' for
manipulating, extracting and replacing data in arrays.
Depends: R (>= 1.5.0)
Imports: methods, utils
License: LGPL (>= 2)
NeedsCompilation: no
Packaged: 2016-07-19 15:24:25 UTC; tap
Repository: CRAN
Date/Publication: 2016-07-21 19:18:05
Built: R 4.1.1; ; 2021-09-10 15:52:31 UTC; windows
-- File: C:/Program Files/BioStat Prime/10/resources/package/R-
4.1.3/library/abind/Meta/package.rds
```

alt text

2. List of installed packages. The user needs to select a library path. Packages installed to the selected library path will be displayed and then execute the dialog. The details about the package will appear in the output window as shown in the picture.

List Installed R Packages				
	Package	LibPath	Version	Priority
	abind	abind C:/Program Files/BioStat Prime/10/resources/package/R-4.1.3/library	1.4-5	NA
	acepack	acepack C:/Program Files/BioStat Prime/10/resources/package/R-4.1.3/library	1.4.1	NA
	acs	acs C:/Program Files/BioStat Prime/10/resources/package/R-4.1.3/library	2.1.4	NA
	admisc	admisc C:/Program Files/BioStat Prime/10/resources/package/R-4.1.3/library	0.2900	NA
	afex	afex C:/Program Files/BioStat Prime/10/resources/package/R-4.1.3/library	1.1-1	NA
	AlgDesign	AlgDesign C:/Program Files/BioStat Prime/10/resources/package/R-4.1.3/library	1.2.1	NA

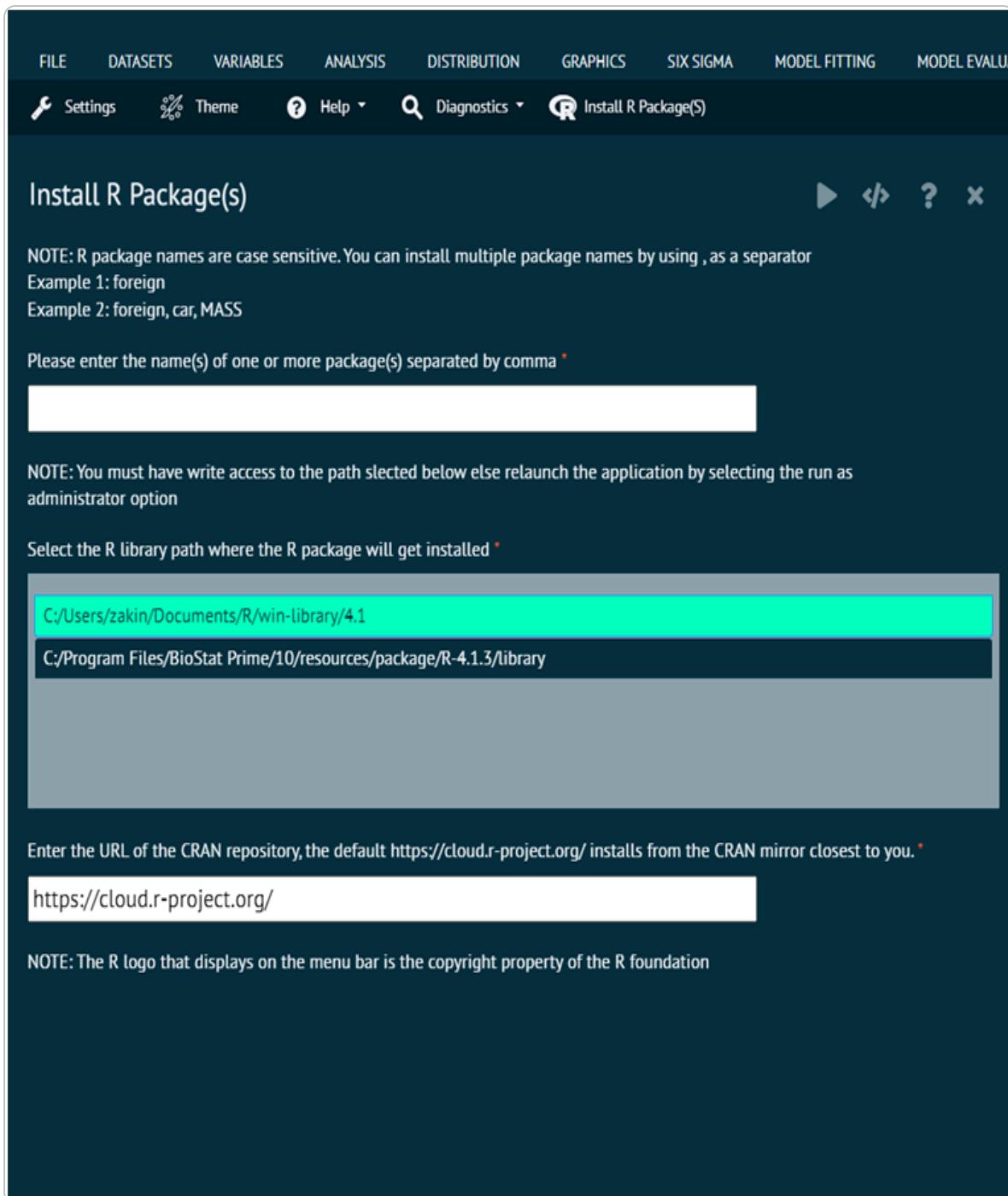
alt text

Install R package(s)

This section of the software aids the user to install R packages. Here are a few guidelines that user should follow while installation.

1. Case matters in R package names. User can use (,) as a separator to install several package names.
2. User must have write access to the path selected where the R package will get installed.

3. The user needs to Enter the URL of the CRAN repository.



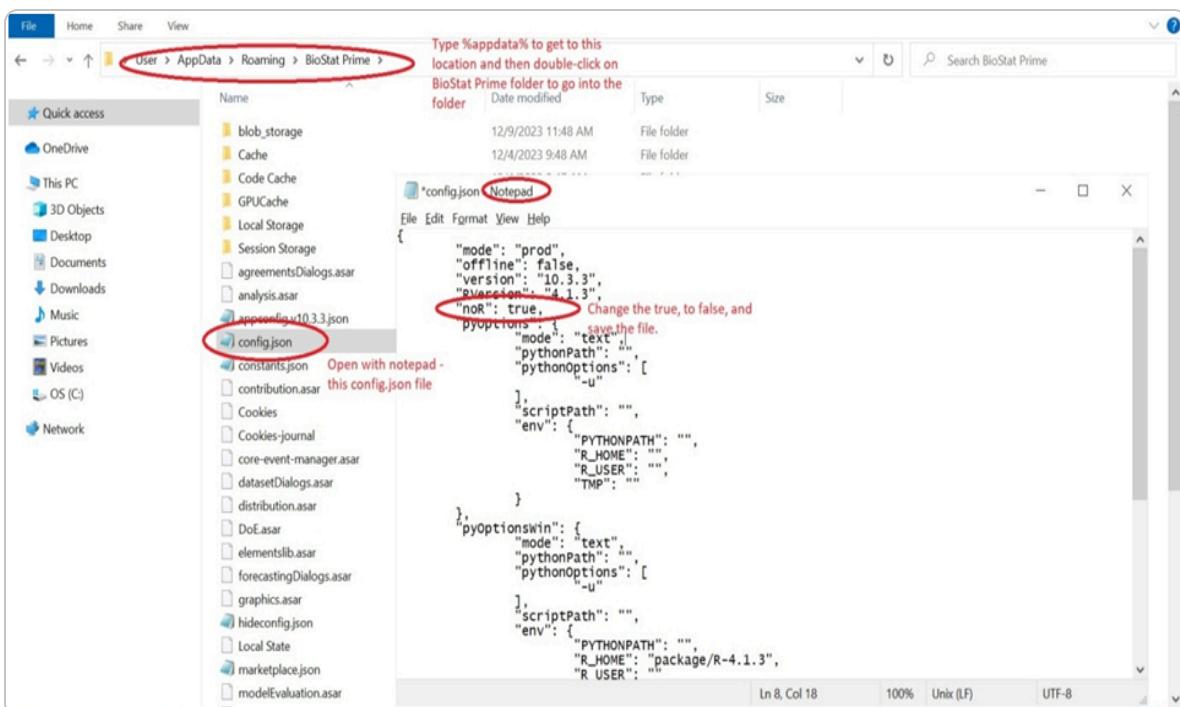
alt text

Advanced Users

In order to extend the functionalities of BioStat Prime, user can go a step further by enabling R console inside the software whenever needed. The R console provides user an opportunity for users, who knows the R programming language, to write, edit and execute the R code in console.

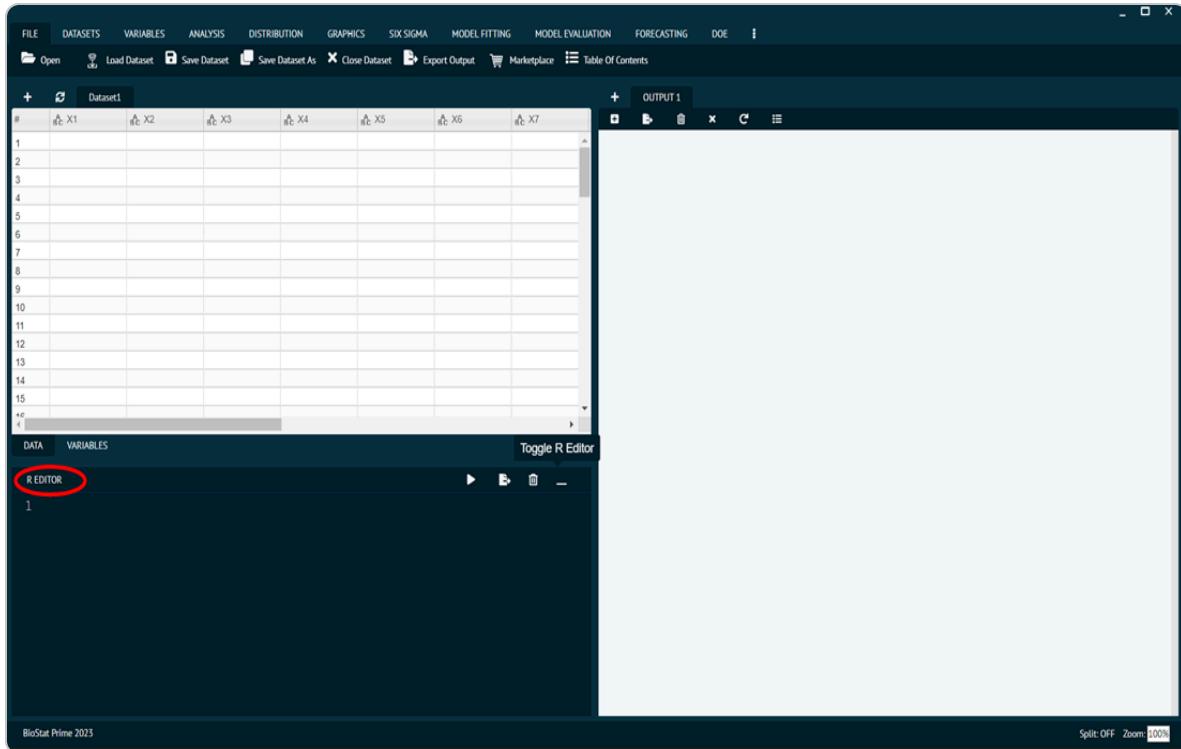
How to Enable/ Disable R console

Go to BioStat Prime folder in your directory -> Open config.json file(or config file of json type) with notepad -> Change the value true to false as shown in the picture below.



alt text

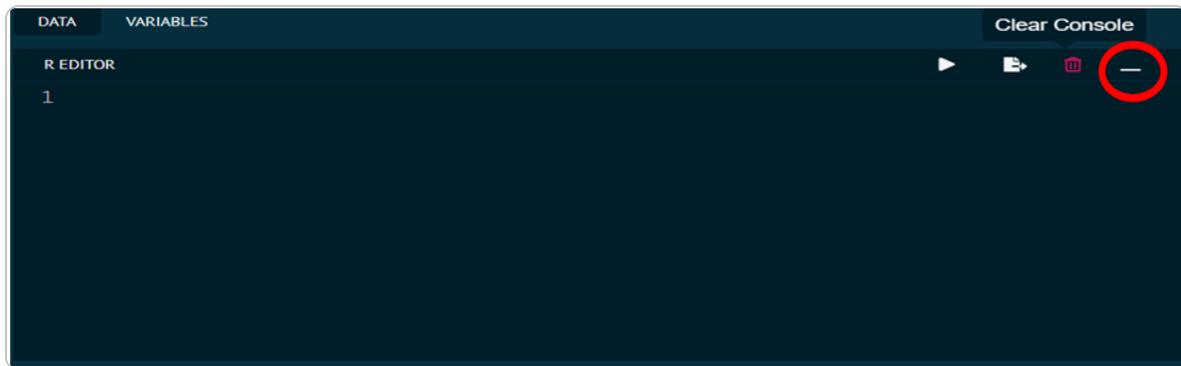
Exit the BioStat Prime app if it is running before making the change to the configuration flag. Once user saves the configuration -> restart the app -> see the R Editor panel. Do the same steps to reverse the configuration to hide the R Editor.



alt text

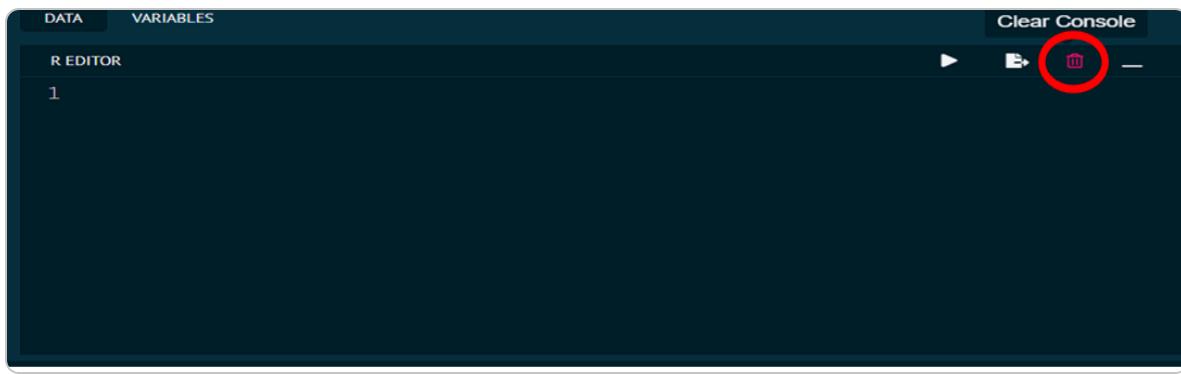
The options at top right corner of the R console are (from right to left).

Toggle R Editor: Used to minimize or maximize the R console.



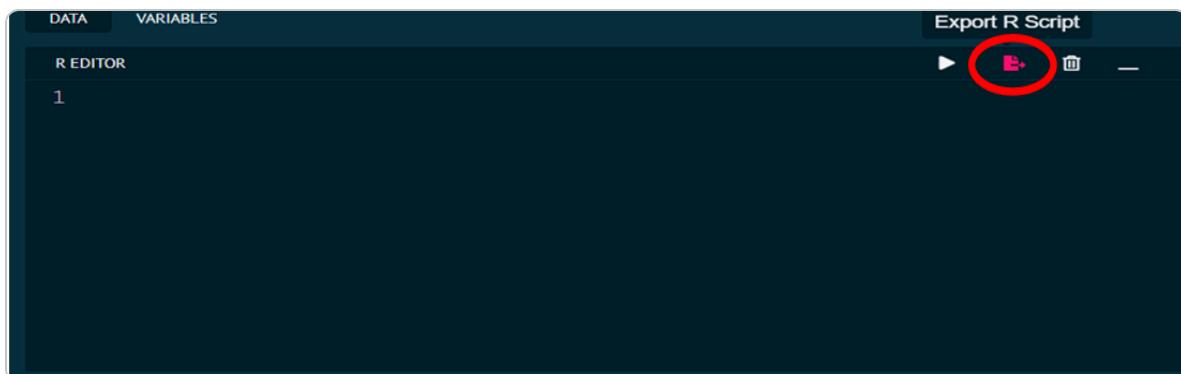
alt text

Clear Console: Clears the entire code in R console.



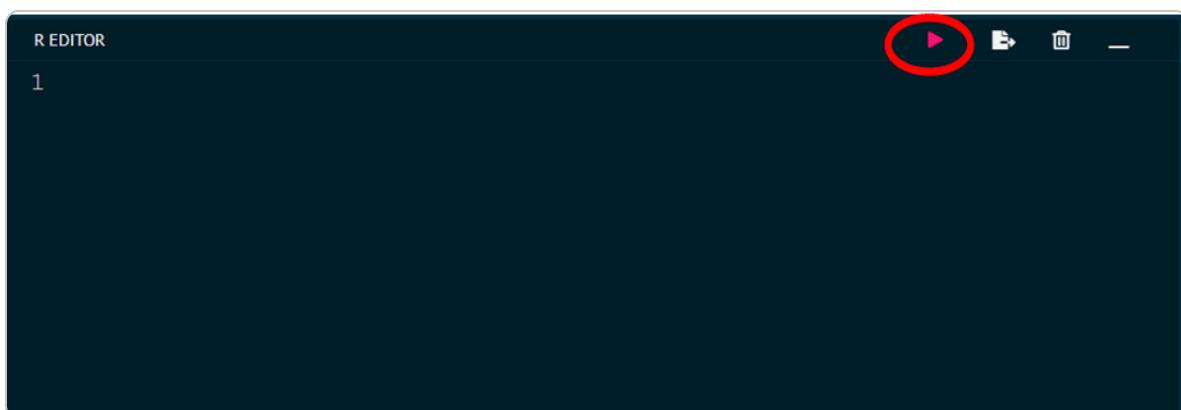
alt text

Export R Script: Used to save R script exporting it to the PC/Laptop.



alt text

Execute Button: Executes the R script.



alt text

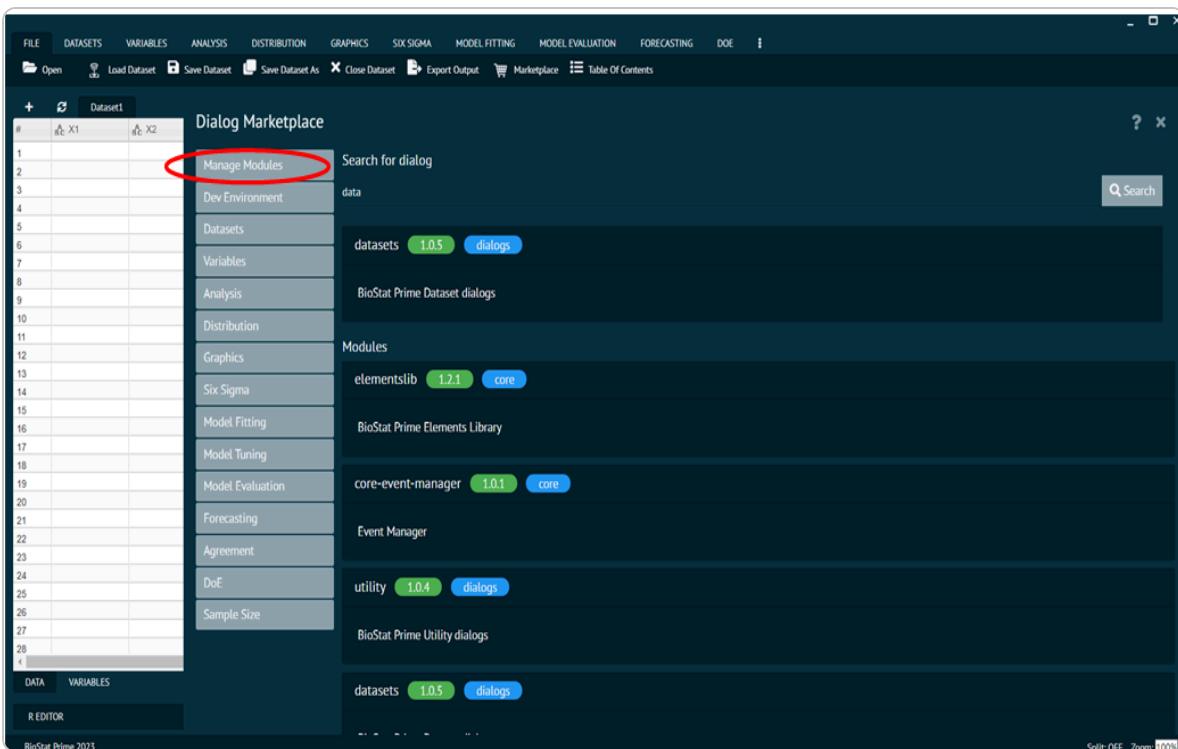
Marketplace

How to use Marketplace

One of BioStat Prime's signature features is the Marketplace, which lets users customize the program to suit their needs and increase its usefulness. The Marketplace is a free shop where R functions and libraries can be added to BioStat Prime to cover more recent statistical topics. R functions and packages are either installed or hidden.

Manage Modules

The marketplace's top most option is Manage Modules. It is in charge of looking for dialogs in the marketplace that is accessible.

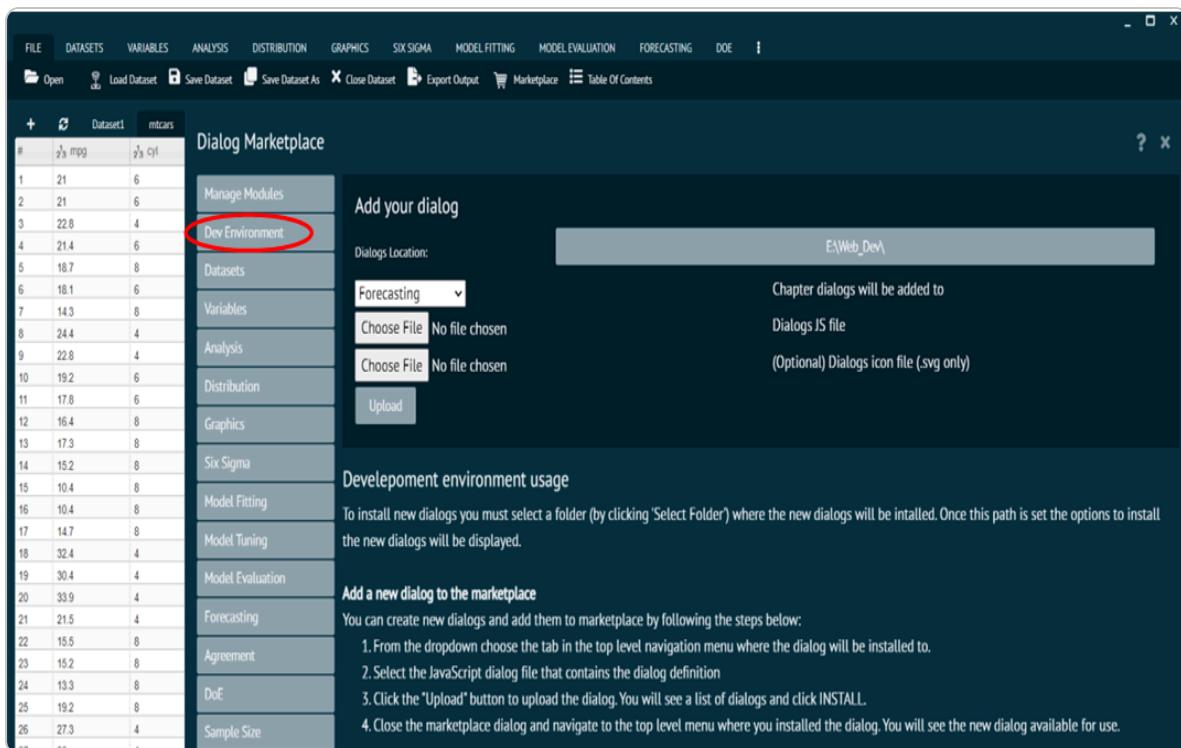


alt text

Dev Environments

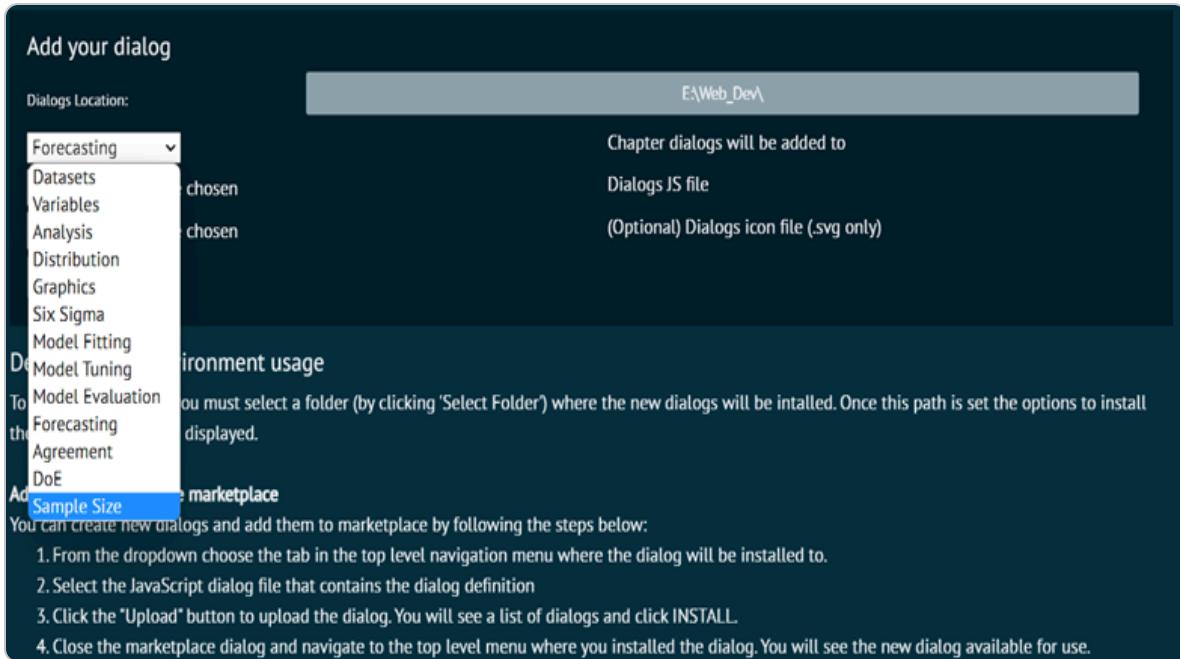
To install the new dialogs, users need to go to the Dev Environment tab in Marketplace section and must pick the location (by clicking "Select Folder") where they want to be installed -> Once the place is specified, the options to install the new dialogs will show

up -> Select the tab in the top level navigation menu where the dialog will be installed from the dropdown menu -> Choose the JavaScript dialog file containing the definition of the dialog -> To upload the dialog, click the "Upload" button -> After selecting INSTALL, a list of dialogs will appear -> After closing the marketplace dialog, user needs to select the dialog installed from the top-level menu -> The new dialog will appear and will be usable.



alt text

Selecting the tab where the dialog will be installed.



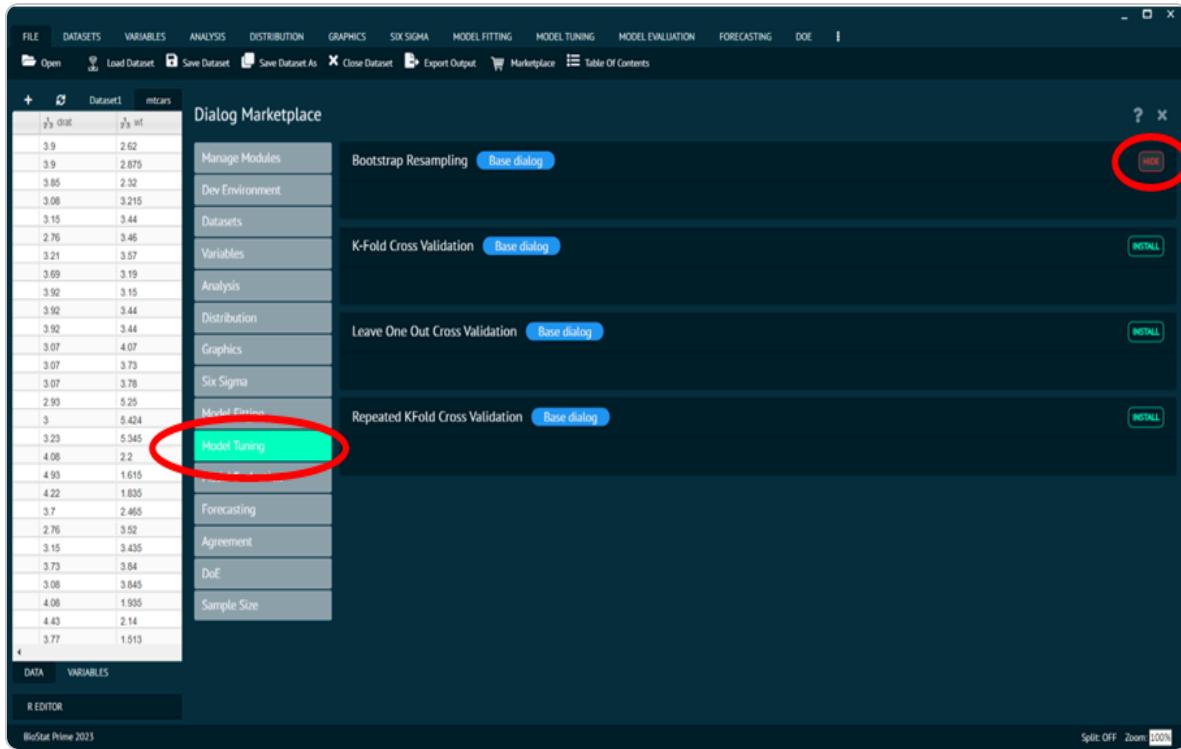
alt text

Installing R Libraries from Marketplace

To enable a new Menu and sub menu in BioStat Prime, user needs to install the R libraries from marketplace. The steps to take the same are as follows.

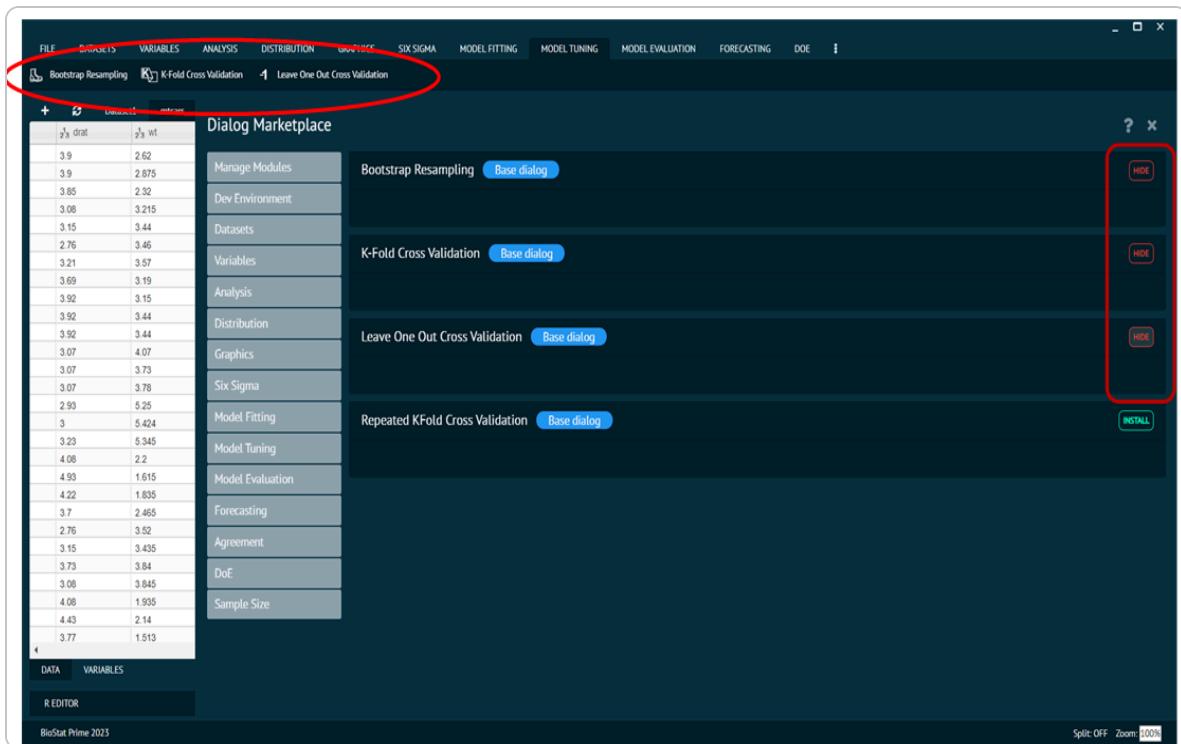
Go to file Menu -> Marketplace -> Choose the package to be installed (say Model Tuning) -> Click install next to respective functions that user wants in the sub menus.

BioStat Prime will add the library in the main menu and its functions in sub menu.



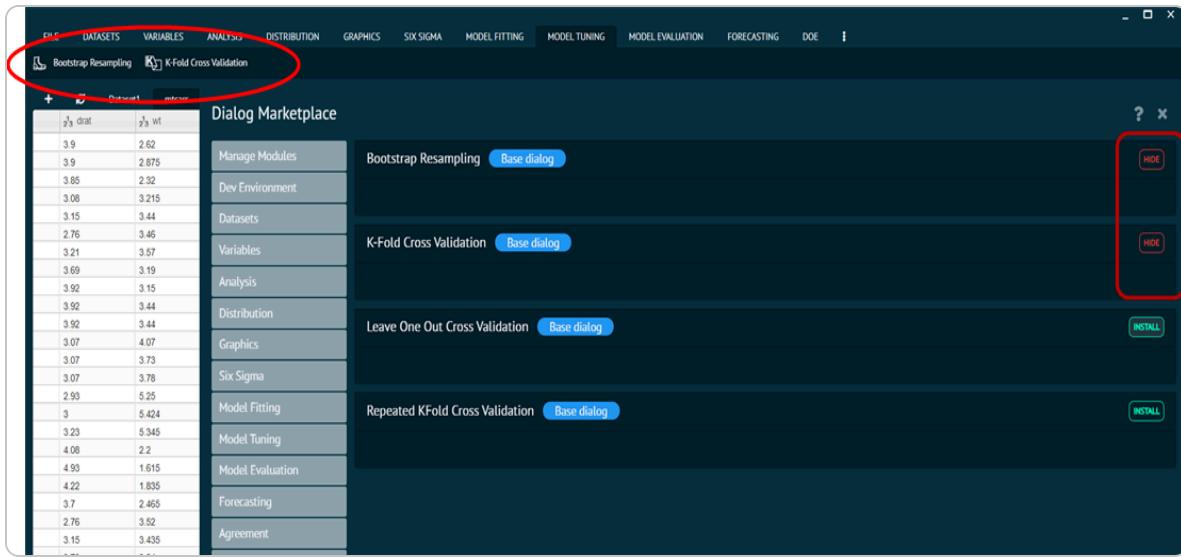
alt text

As the user proceeds to install the packages, the functions appear in the main menu and sub menu.



alt text

User can hide the sub function whenever needed, by clicking the hide button next to the respective function in marketplace.



alt text