



BioStat Prime

Table of Contents

BioStat Prime	6
Index	7
Introduction	8
Licensing Models	10
License Activation Guide	12
Installation	17
How to use BioStat Prime	23
File Menu	33
Dataset- Data Management	41
Aggregate	42
Compare Dataset	44
Expand	50
Find Duplicates	54
Group By	59
Merge	61
ReShape	68
Sampling	70
Sort	75
Subset	79
Transpose	85
Variables-Operations	87
Bin	88
Box Cox	90
Compute	94
Concatenate	101
Convert	102
Date order check	107
Delete Variable	109
Factor levels	110
ID Variable	121
Lag or Lead Variable	124
Missing Values	126
Rank Variable(s)	134

Recode Variables	137
Standardize Variable(s)	140
Transform Variable(s)	143
Data Analysis	145
Cluster	146
Correlations	150
Crosstab.....	152
Distribution Analysis	159
Factor analysis	170
Means	175
ANOVA, 1 and 2 way	176
ANOVA, one-way with random blocks	178
ANOVA, one way with blocks	182
ANOVA, N way	184
ANOVA Repeated Measures, Long	187
ANOVA Repeated Measures, Wide	189
MANOVA	191
t-test, Independent	192
t-test, One Sample	195
t-test, Paired Samples	196
Missing Value	198
Moments	201
Non-Parametric	203
Proportions	219
Summary	226
Survival	232
Tables	237
Variance	239
Distribution Analysis	243
Chi-square test	244
Lognormal	251
Normal	256
Poisson	259
Graphs and Charts	267
Bar Chart	268
Box Plot	269

Contour Plot	270
AB 2D Contour Plot	272
Distribution Plot	273
HeatMap	278
Line Charts	280
Maps	285
Pie Charts	288
Scatter plots	291
Stem And Leaf	293
Strip Chart	294
Violin	295
Six Sigma-Quality Control	296
Six Sigma Overview	297
Cause and Effect	298
Pareto Chart	300
Loss Function Analysis	301
MSA (Measurement System Analysis)	302
Gage R&R-Measurement System Analysis	303
Attribute Agreement Analysis	305
Gage Bias Analysis	308
Process Capability	309
Shewhart Charts	312
Cusum Chart	317
EWMA Chart	318
MQCC Chart	320
Multi-Vari Chart	322
Model-Curve Fitting	323
Regression	324
Cox, Advanced	326
Cox, Basics	330
Cox, Binary Time-dependend covariates	333
Cox, Fine-Gray	338
Cox Regression, Multiple models	342
Cox, Stratified	350
Linear Regression, Advanced	354
Linear Regression, Basics	357

Linear Regression (Legacy)	360
Linear Regression, multiple models	363
Logistic Regression, Advanced	370
Logistic Regression, Basic	373
Logistic Regression, Conditional	377
Logistic Regression, multiple models	380
Multinomial Logit	387
Ordinal Regression	391
Quantile Regression	395
Non-Linear Regression	399
Naive Bayes	404
SEM	406
Save Model to a file	408
Load Model from a file	410
Model Evaluation	412
Compare N Models	413
Confidence Interval	414
FIT	415
Outlier Test	420
Predict	422
Forecasting	425
Automated ARIMA (AR)	426
Exponential Smoothing (ES)	429
Holt Winters, Non-seasonal	433
Holt Winters, Seasonal	436
Plot Time Series, Separate OR Combined	439
Plot Time Series with Correlations	442
Design of Experiment-Quality Control	445
Create DoE Factor Details	449
Import Design Response	451
Export Design Response	453
Create Design	454
Inspect Design	463
Modify Design	467
Analyse Design	469
Features of BioStat Prime that enhance DOE	475

Quality Assurance	476
Settings Menu	479
Advanced Functionalities	488
Marketplace	491

BioStat Prime



BioStat Prime

@BiostatPrime2025

Index

Key Indicators to UserGuide

TIP

- ⚠ This block provide optional information or helpful advice, like an alternative way of doing something.

Note

- ℹ This block provides important information that the reader should be aware of, like known issues or limitations.

Warning

- ⚠ This block provides critical information about potentially harmful consequences, such as damage or data loss.

Block

Blocks

This Block highlights the content that needs more attention

Introduction

BioStat Prime is a cutting-edge biostatistical software designed to simplify and enhance data analysis, visualization, and reporting for scientists and researchers. BioStat Prime's intuitive point-and-click graphical user interface (GUI) ensures that users can navigate and utilize the software effortlessly, maximizing productivity and minimizing the learning curve.

With BioStat Prime, user can seamlessly import, clean, and transform data from various sources, unravel complex biostatistical challenges, and harness the power of advanced statistical methods such as descriptive statistics, hypothesis testing, regression analysis, and multivariate analysis.

BioStat Prime features robust tools for data visualization, enabling user to create compelling charts, graphs, and plots that bring user's data to life. For in-depth exploration, BioStat Prime offers features like summary statistics generation (mean, median, mode, standard deviation, quartiles) and Principal Component Analysis (PCA), complete with tools like Parallel Analysis, Scree Plots, and Biplots to assist in downstream applications such as Principal Component Regression.

The well-organized toolbars and menus provide quick access to all functions, while the built-in help system and interactive tutorials ensure user learns as user goes, making complex analyses easier to master. Whether user is conducting Six Sigma analyses, utilizing statistical process control (SPC) for real-time data quality, or exploring ways to reduce wastage and improve cost efficiency, BioStat Prime empowers user to achieve optimal results.

To simplify reporting, BioStat Prime allows user to export data and analyses directly into PDF, Word, and PowerPoint formats, enabling clear and visually appealing reports for sharing and collaboration. Additionally, the BioStat Prime Marketplace offers a dynamic platform to expand the software's functionality by adding or customizing tools and packages.

What makes BioStat Prime unique?

1. BioStat Prime empowers precision in Life Sciences through Statistical Insight.

2. It is a simple-to-use point and click based platform that makes statistical analysis accessible to users without extensive programming knowledge.
3. BioStat Prime is designed to provide comprehensive statistical analysis like anova, regression, correlation, survival analysis and much more, enabling scientists and researchers to analyze complex datasets with ease.
4. It also generates output and reports summarizing the results of statistical analysis. This facilitates easy interpretation and sharing of findings.
5. BioStat Prime includes graphical tools for data visualization, allowing users to create charts, graphs, and plots to better understand their data.
6. BioStat Prime delivers fast, accurate, and reproducible results, whether user is working with clinical trials, epidemiological studies, or biological experiments.
7. The built-in MARKETPLACE offers a free store where users can add or customize functions and packages, tailoring the software to meet specific analytical needs.
8. Features like Six Sigma and Statistical Process Control (SPC) tools to analyze data in real-time, improve processes, reduce defects, and ensure high-quality results.
9. Built-in integration with the R programming language leverages R's extensive statistical capabilities, allowing advanced analyses without the need for direct coding. Users can also add and execute R code chunks directly in the integrated console for more flexibility.

Licensing Models

BioStat Prime offers four distinct licensing models, allowing users to choose the option that best fits their needs. The licensing model selected serves as a fundamental aspect of the software's usage and access.

Perpetual

A perpetual license grants users indefinite access to the software without an expiration date. This license remains valid permanently; however, it may include specific conditions such as machine activation limits, feature restrictions, or exclusions from major version upgrades.

Subscription

This license type operates based on predefined subscription periods and continuously synchronizes the subscription status with the active state of the license.

Grouped

BioStat Prime's grouped license allows multiple users or devices within a defined group, such as an organization or team, to access BioStat Prime under a single Admin. This model simplifies BioStat Prime's license management by enabling centralized control, ensuring that all authorized users within the group can utilize the software.

Trial

A trial license provides users with temporary access to BioStat Prime for evaluation purposes. This allows potential users to explore the software's features, functionality, and capabilities before committing to a full license.

BioStat Prime's trial licenses are typically valid for a limited period. Once the trial period expires, users must upgrade to a paid license to continue accessing the software.

Floating

A floating license, also referred to as a concurrent or network license, allows a specified number of users or devices to access BioStat Prime simultaneously.

This license operates within a shared license pool, from which authorized users or devices can dynamically obtain access as needed, ensuring efficient resource utilization across multiple users.

Enterprise (Site licensing)

A site license grants an organization or institution the right to install and use BioStat Prime across multiple computers or devices within a designated location or site. This licensing model is ideal for institutions requiring widespread software access.

License Activation Guide

To unlock the full functionality of BioStat Prime, user must activate the license. Follow the steps below to complete the activation process.

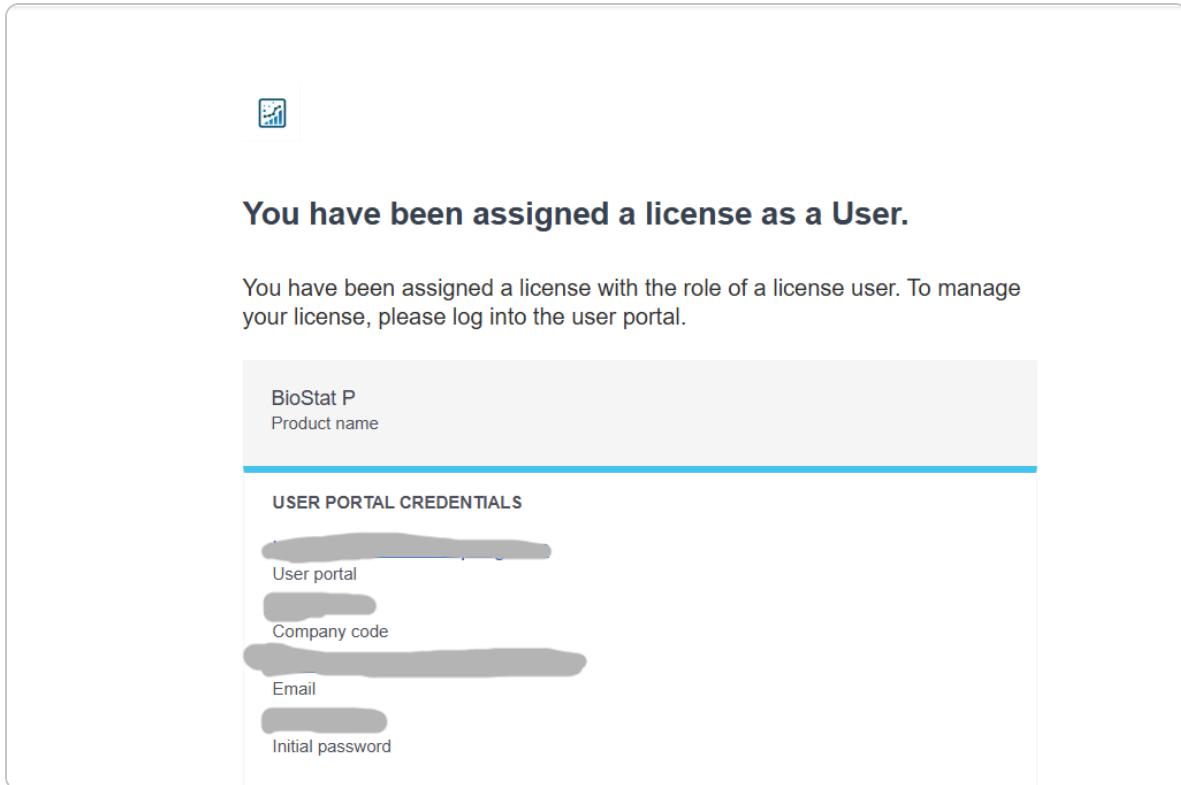
Prerequisites

Before proceeding with the activation of BioStat Prime, ensure the following requirements are met:

1. Stable Internet Connection – A reliable and active internet connection is required for license activation.
2. Valid License Purchase – The license must be successfully purchased and processed.
3. Registered Email Address – Ensure that the email address used for payment is correctly selected and retained, as it will be required for activation and it serves as the primary point of communication for license-related matters.
4. Administrator Privileges – Ensure you have the necessary permissions to install and activate software on your device.

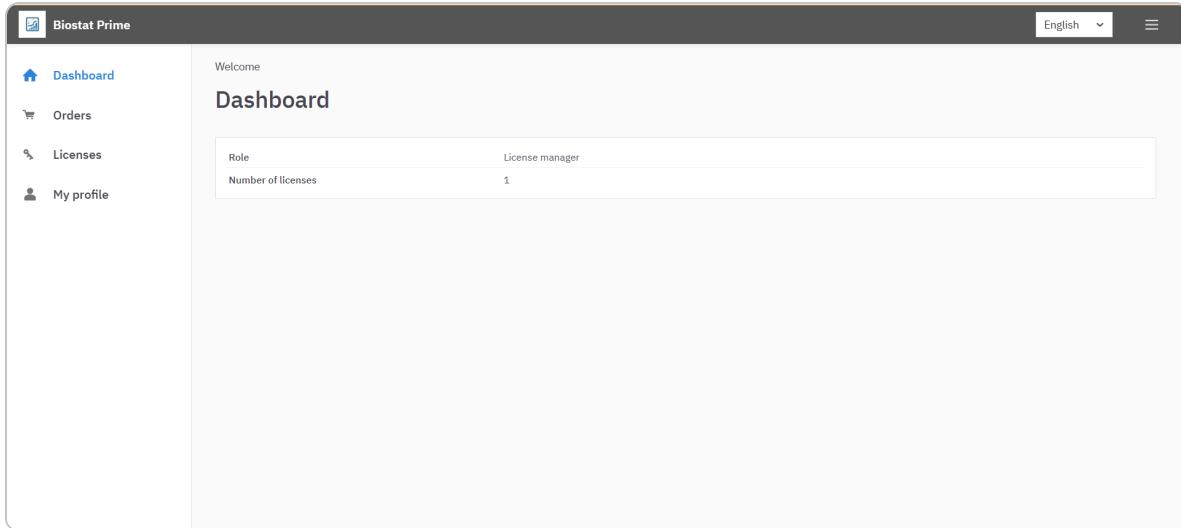
Activation via Account Login

1. After completing the payment successfully, the user will receive an email containing credentials to log in to the user portal. This portal provides access to the software installer link and license activation credentials.



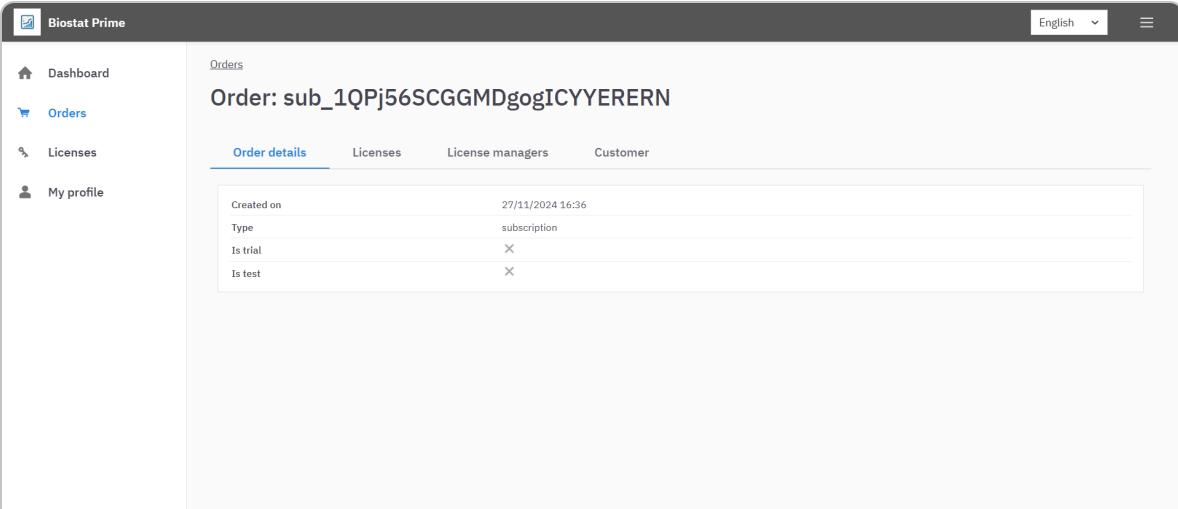
email-after-payment

2. Upon receiving the email, the user should log in to the LicenseSpring Dashboard using the provided credentials to manage their license and access account details.



Dashboard

3. Once logged into the LicenseSpring Dashboard, the user can access the details of the license issued to their registered email ID.

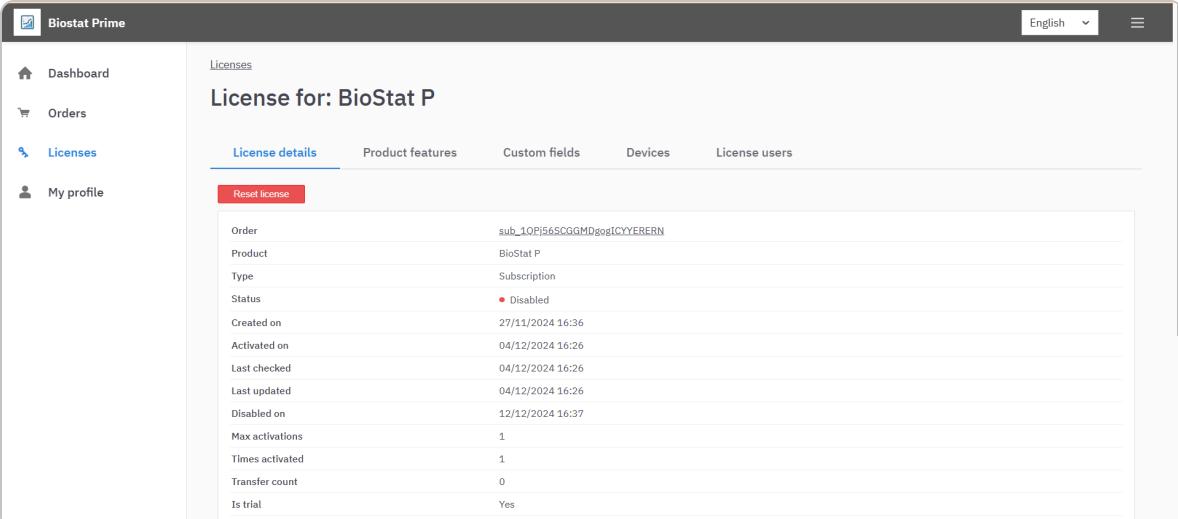


The screenshot shows the 'Orders' section of the Biostat Prime dashboard. The main title is 'Order: sub_1QPj56SCGGMDgogICYYERERN'. Below it, there are tabs for 'Order details', 'Licenses', 'License managers', and 'Customer'. The 'Order details' tab is selected. It displays the following information:

Created on	27/11/2024 16:36
Type	subscription
Is trial	X
Is test	X

order-details

4. Selecting the issued license redirects the user to a detailed page that includes license information and a link to download the software installer.



The screenshot shows the 'Licenses' section of the Biostat Prime dashboard. The main title is 'License for: BioStat P'. Below it, there are tabs for 'License details', 'Product features', 'Custom fields', 'Devices', and 'License users'. The 'License details' tab is selected. It displays the following information:

Order	sub_1QPj56SCGGMDgogICYYERERN
Product	BioStat P
Type	Subscription
Status	● Disabled
Created on	27/11/2024 16:36
Activated on	04/12/2024 16:26
Last checked	04/12/2024 16:26
Last updated	04/12/2024 16:26
Disabled on	12/12/2024 16:37
Max activations	1
Times activated	1
Transfer count	0
Is trial	Yes
Total issue	1

License-details

The screenshot shows the Biostat Prime software interface. On the left, there's a sidebar with navigation links: Dashboard, Orders, Licenses (selected), and My profile. The main content area displays license information:

Last checked	04/12/2024 16:26
Last updated	04/12/2024 16:26
Disabled on	12/12/2024 16:37
Max activations	1
Times activated	1
Transfer count	0
Is trial	Yes
Trial days	15
Max license users	1
Prevent virtual machine	No
Note	-

Below this is a "Product download" section for "Windows 32/64". It shows the file name, version, release date, notes, and a "Download" button.

installer-link

- i** The installation steps, including the process of downloading and installing the software, are provided in the next section ([Installation](#)).

5. Once the software is downloaded and installed, the user can proceed with the activation process. The activation steps outlined below should be followed to activate the license when the software the first time it is run.
6. Within the portal, the user can navigate to the "License Users" tab. Clicking this tab opens a section containing the license activation credentials.

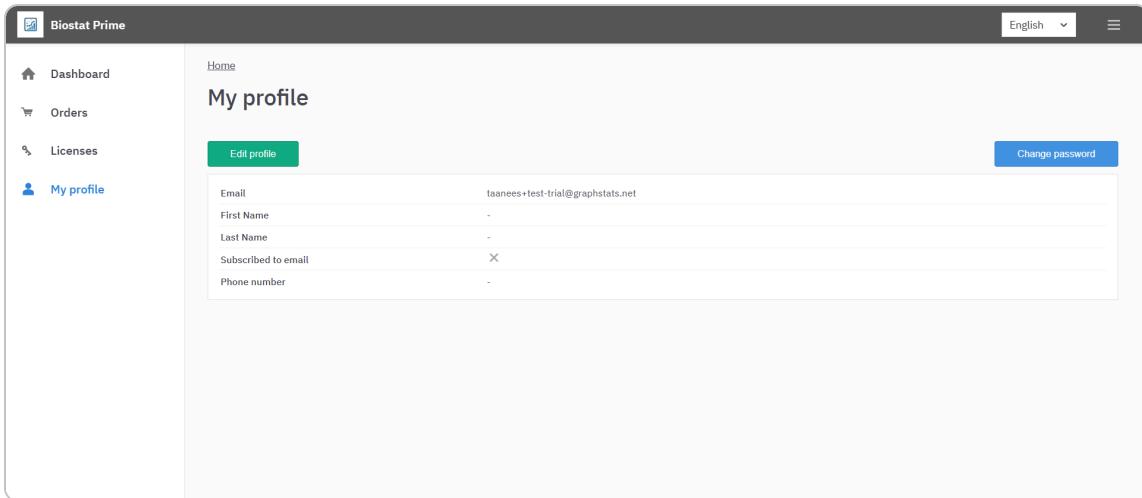
The screenshot shows the "Licenses" section of the Biostat Prime software. The "License users" tab is selected. A table lists one user entry:

Email	First Name	Last Name	Phone number	Initial password	Max activations	Total activations	Actions
taanees+test-trial@graphstats.net	-	-	-	*****	1	1	⋮

Below the table are navigation buttons: Previous, Page 1 of 1, and Next.

license-activation-details

7. The user should copy the provided credentials and use them to complete the license activation process.
8. Additionally, it is strongly recommended to update the temporary password for both the Dashboard and the license at the earliest opportunity to ensure security.



password-change

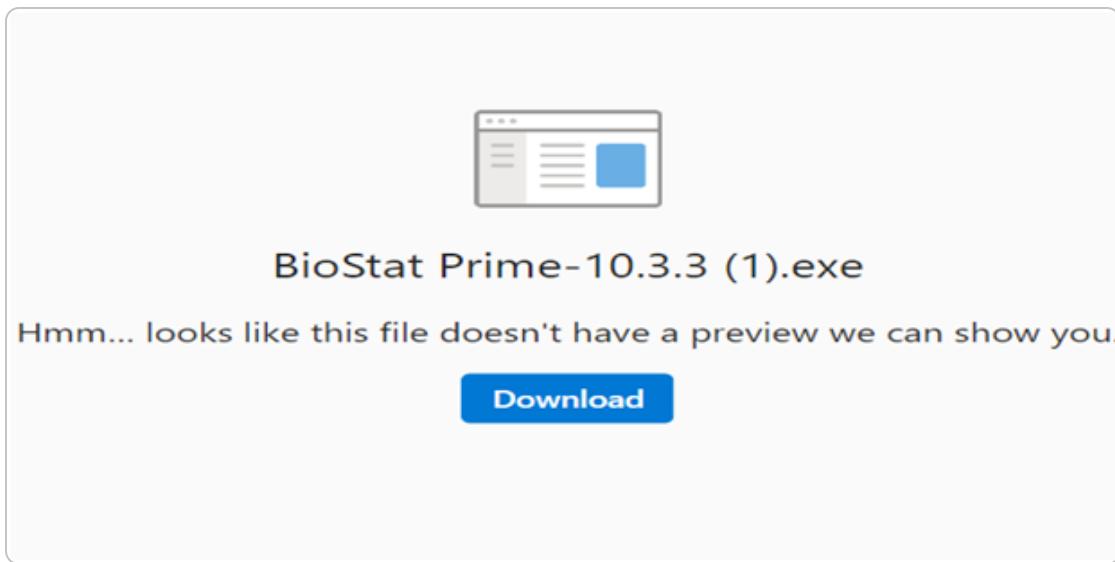
Activating Trial License

Installation

Windows

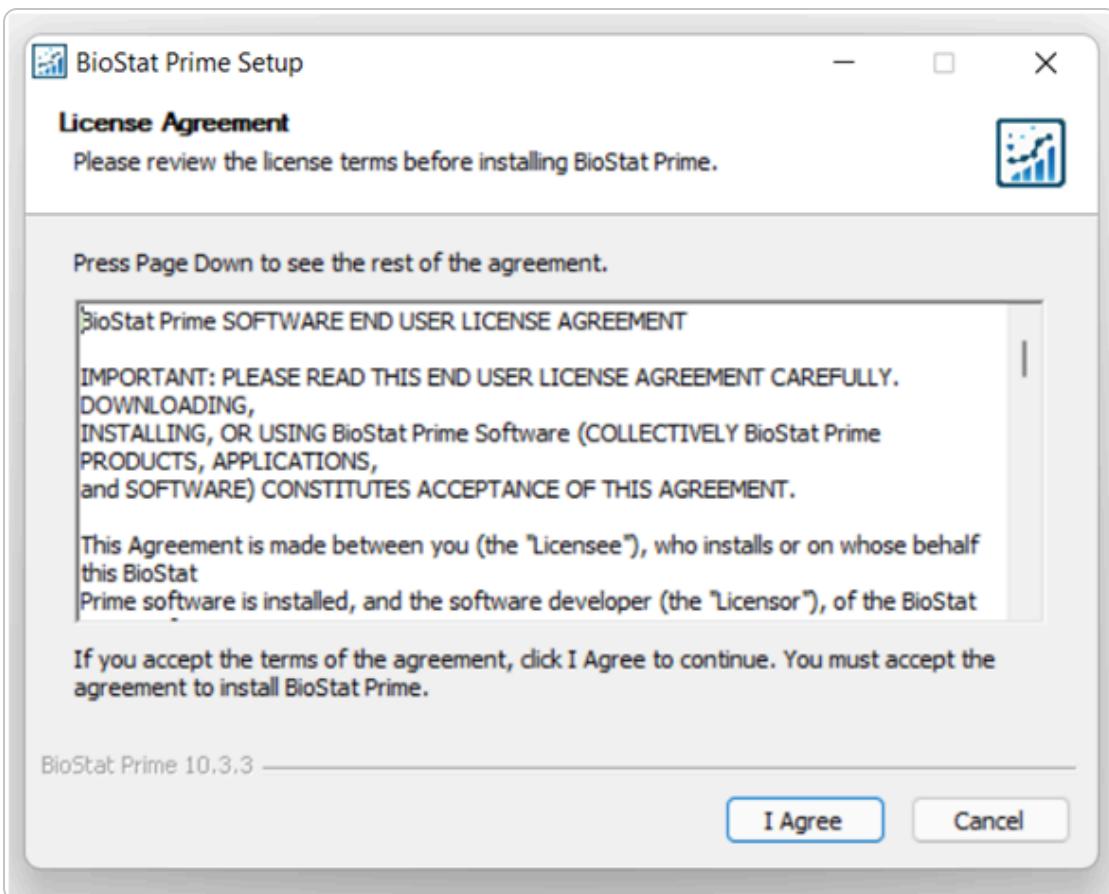
BioStat Prime installation on Windows.

- Use the link on the user dashboard to download the BioStat Prime Windows installer.
- To download BioStat Prime to your PC or laptop, double-click the download button.



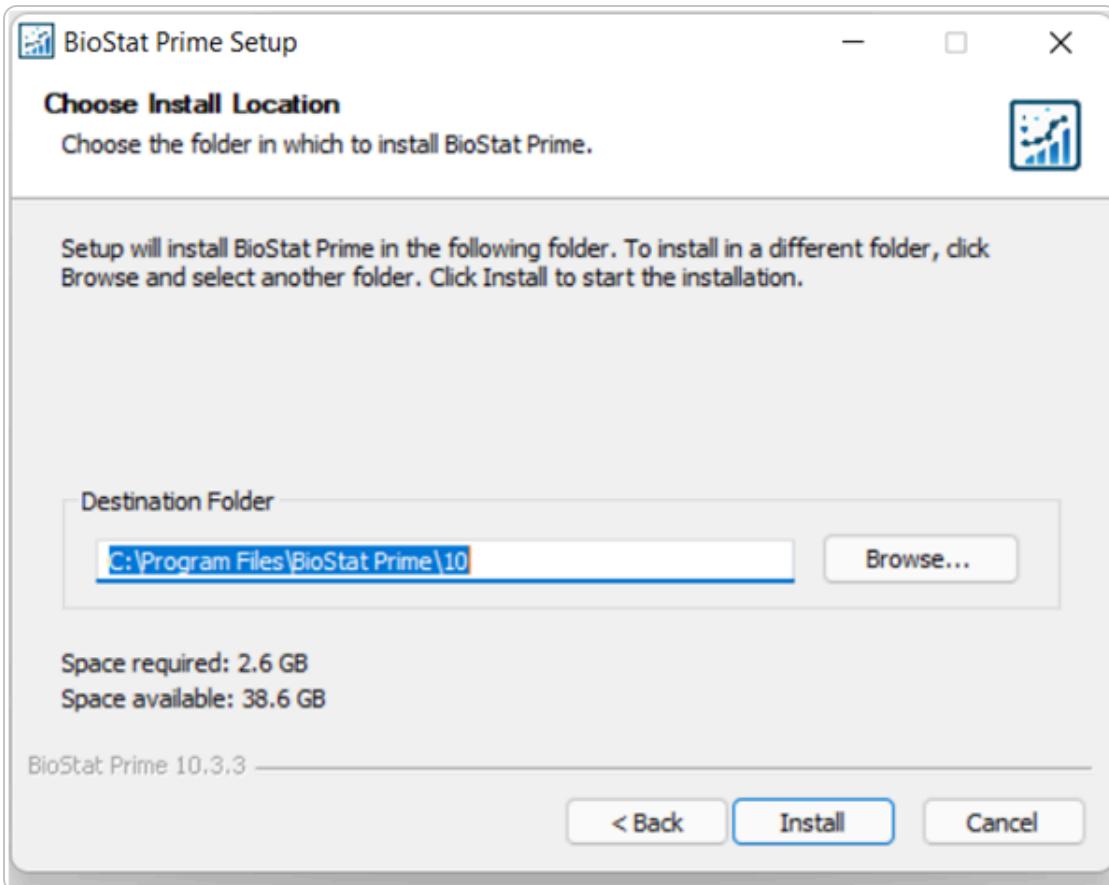
Installer of BioStat Prime

- Review the terms for License Agreement and click on I Agree to proceed with installation.



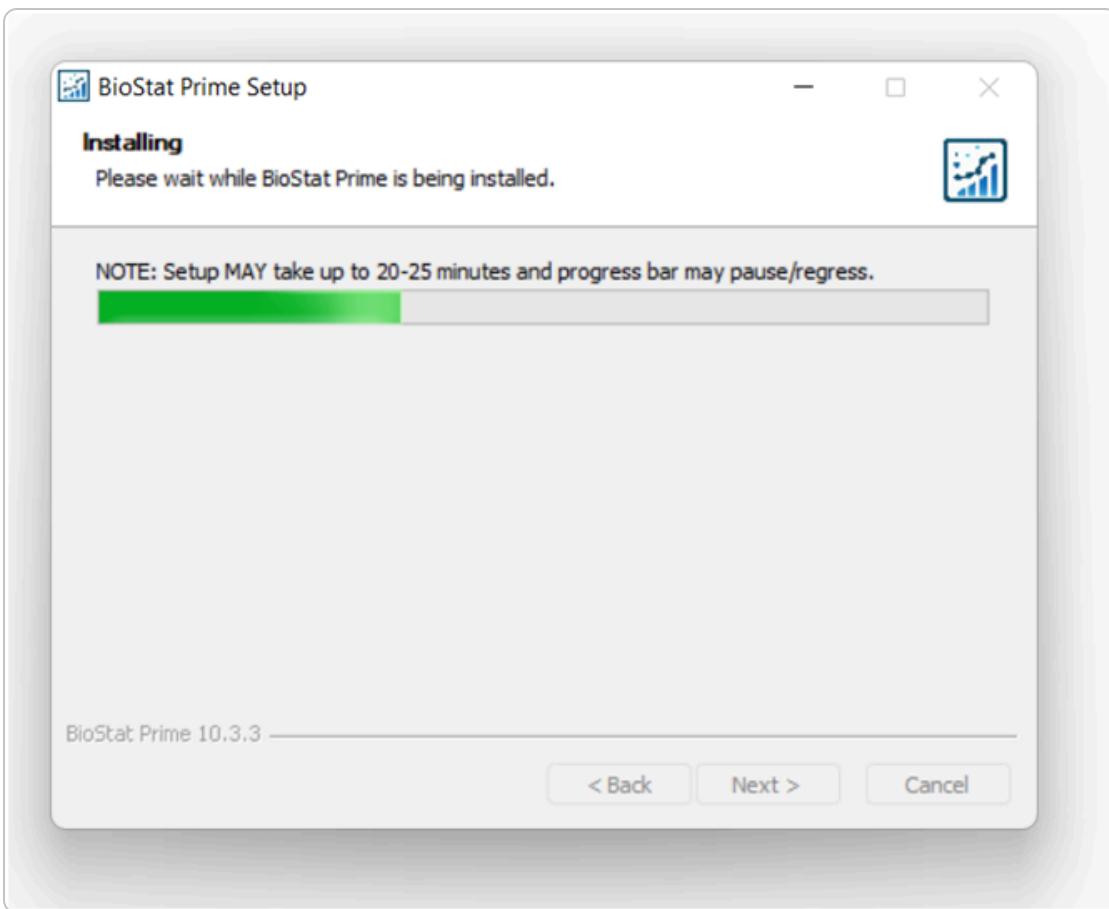
License Agreement

- To initiate the installation of BioStat Prime, select the folder where user wants to install the software and click **Install**.



Location for installation

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



Installing

- BioStat Prime will be installed in the location specified by user.
- Run BioStat Prime.



Logo

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.

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.

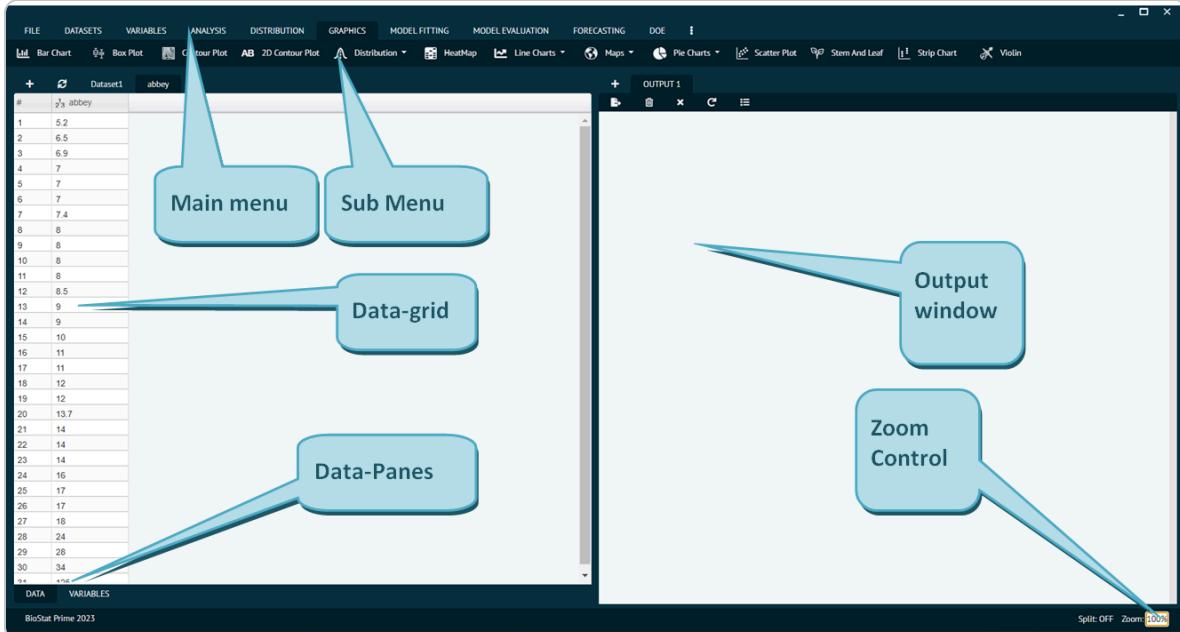
The UI of the software

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 (["Output Window" in "How to use BioStat Prime"](#)), Data-grid (["Working with Data-grid" in "How to use BioStat Prime"](#)) and R console ([Advanced Functionalities](#)), Zoom control.



The UI of the software

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 dropdown.

! 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 a data grid with 11 rows and 6 columns. The columns are labeled '#', 'Name', 'Class', 'Type', 'Measure', and 'Levels'. The data rows represent variables from the 'mtcars' dataset, with 'mpg' as the first row and 'carb' as the last. A context menu is open over the 'Type' column of the 'carb' row, listing options such as 'Add Factor Level', 'Make Factor', 'Make Ordered Factor', 'Make Character' (which is highlighted in red), 'Make Numeric', 'Insert New Date Variable', etc. A blue callout bubble points to the 'Make Character' option with the text 'Changing Variable format to character in variable pane'. The software has a dark theme with a top navigation bar and tabs for 'DATA' and 'VARIABLES'.

#	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	

Working with Data-grid

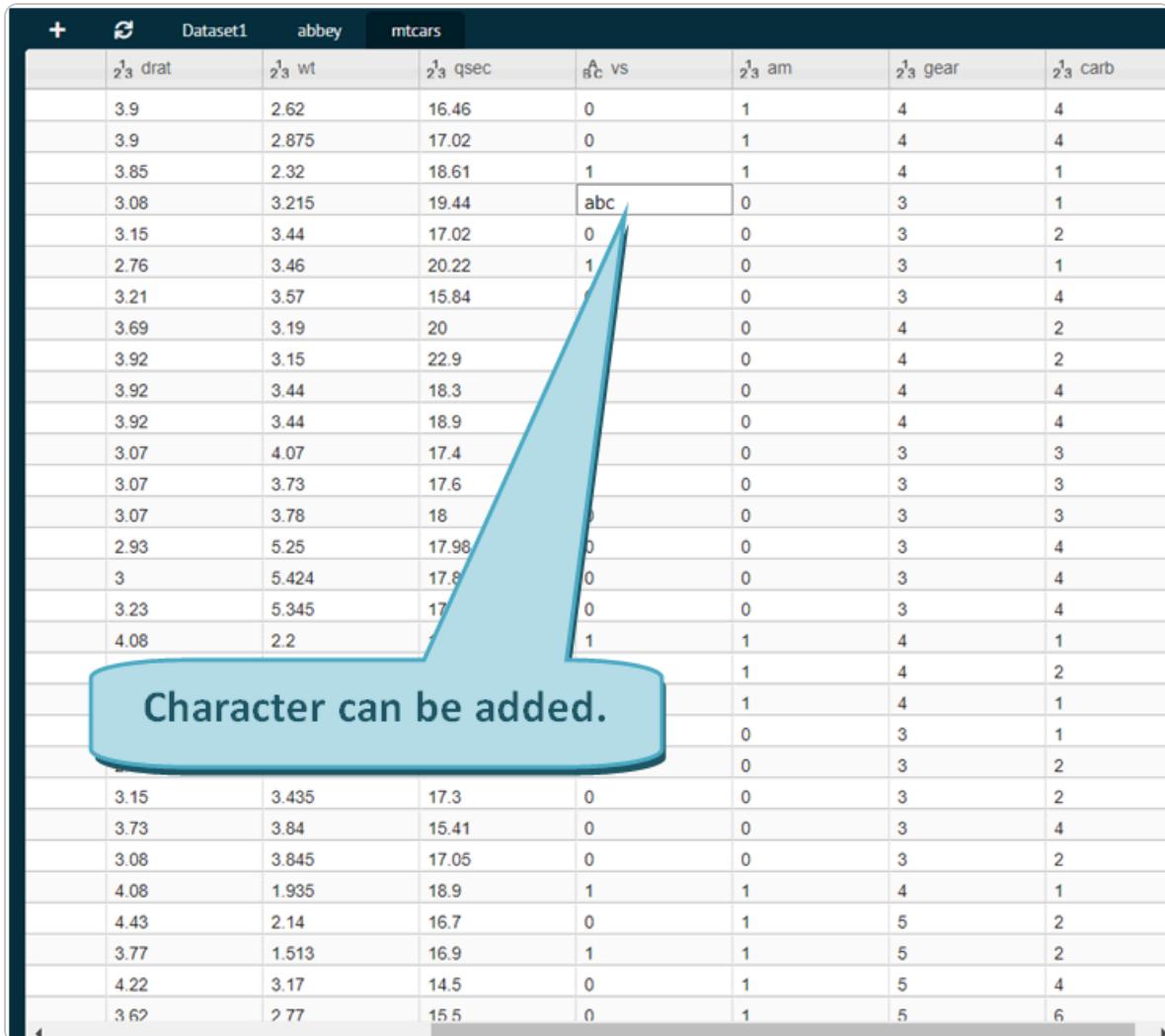
This will open a pop-up window as shown above.

The screenshot shows a software interface for data analysis. At the top, there's 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 right-pointing arrow. A toolbar below the icons includes a plus sign, a refresh symbol, and labels for Dataset1, abbey, and mtcars. The main area displays a data grid for the 'mtcars' dataset. The columns are labeled '#', 'Name', 'Class', 'Type', 'Measure', and 'Levels'. The rows list 11 variables: mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, and carb. The 'Type' column for 'vs' shows 'character' instead of 'double', indicating a format change. A large blue callout bubble points to this row with the text: "Variable format changed to character in variable pane."

#	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	

Working with Data-grid

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



\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
				1	4	1
				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

Working with Data-grid

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

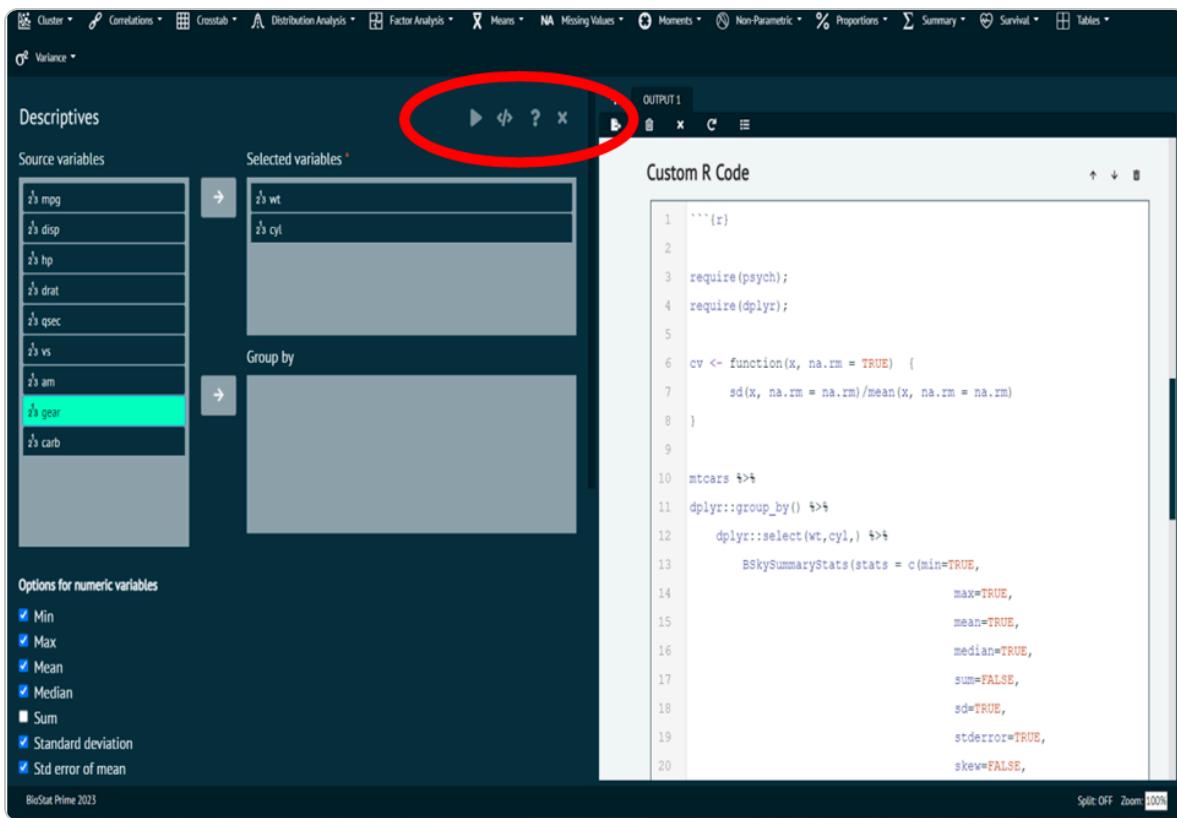
Displays the R syntax for respective dialog analysis.

?

Provides quick summary help.

Cross button

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



Dialog

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 SPSS Output Window titled "OUTPUT 1". The title bar includes standard window controls (minimize, maximize, close) and a toolbar with icons for new, open, close, and others. In the top right corner, there are two red circles highlighting the up and down arrow buttons used for navigating between outputs. The main content area displays the "Descriptives" output for the dataset "data". It includes a "Dataset Overview" table and a detailed "Numerical Statistical Analysis by Variable" table for the variable "disp".

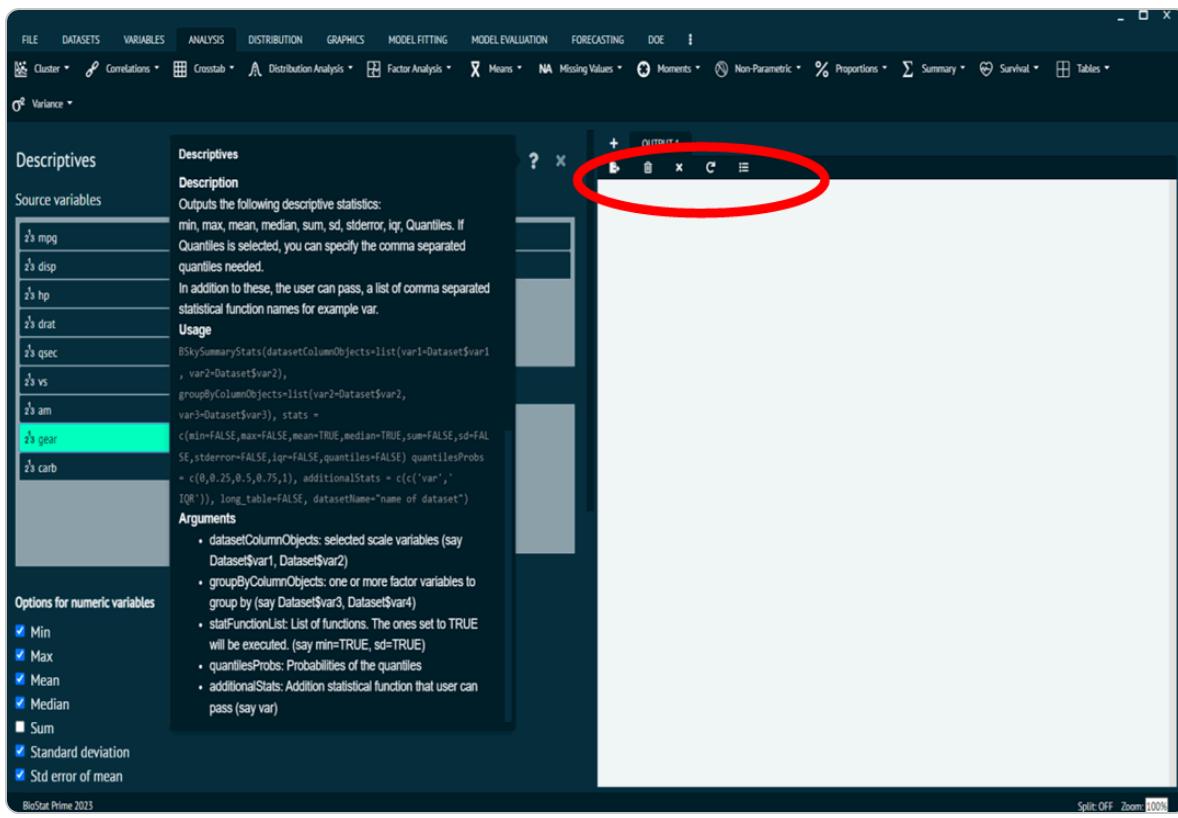
Dataset	Variables	Observations
data	1	32

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

Executing the Dialog

Output Window

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



Output Window

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 Menu

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



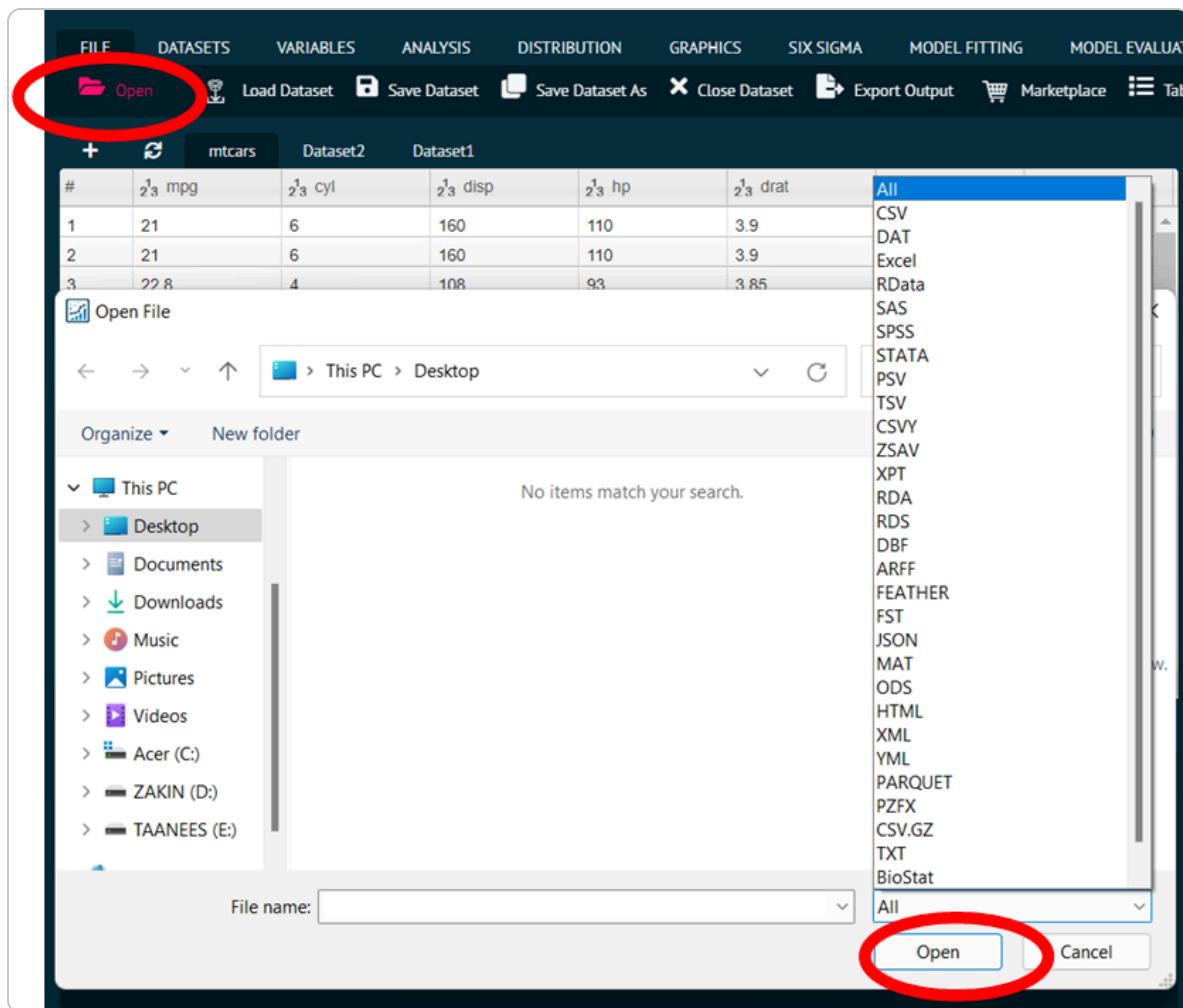
File Menu

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.



Open

Load Dataset

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

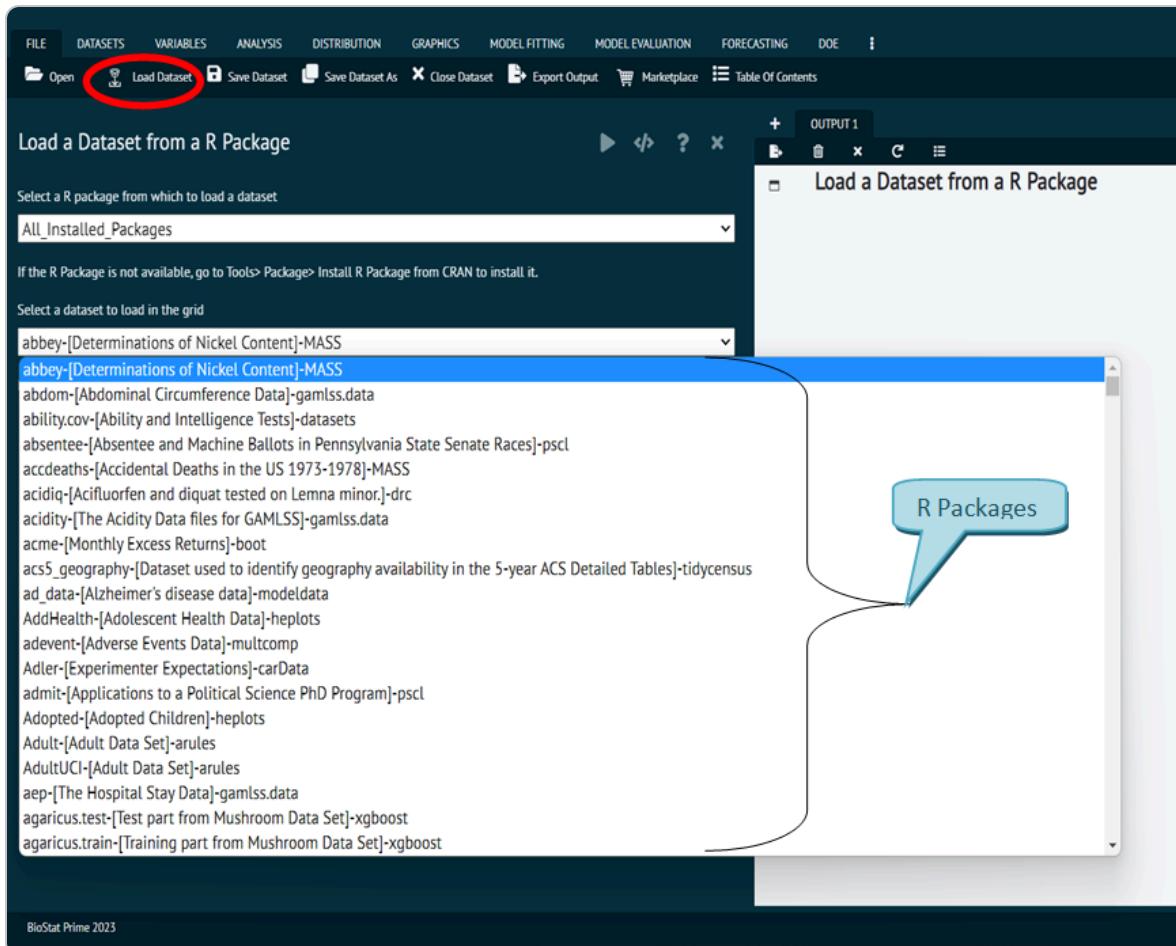
In order to load a dataset, user needs to do the following.

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**.



Load Dataset

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.

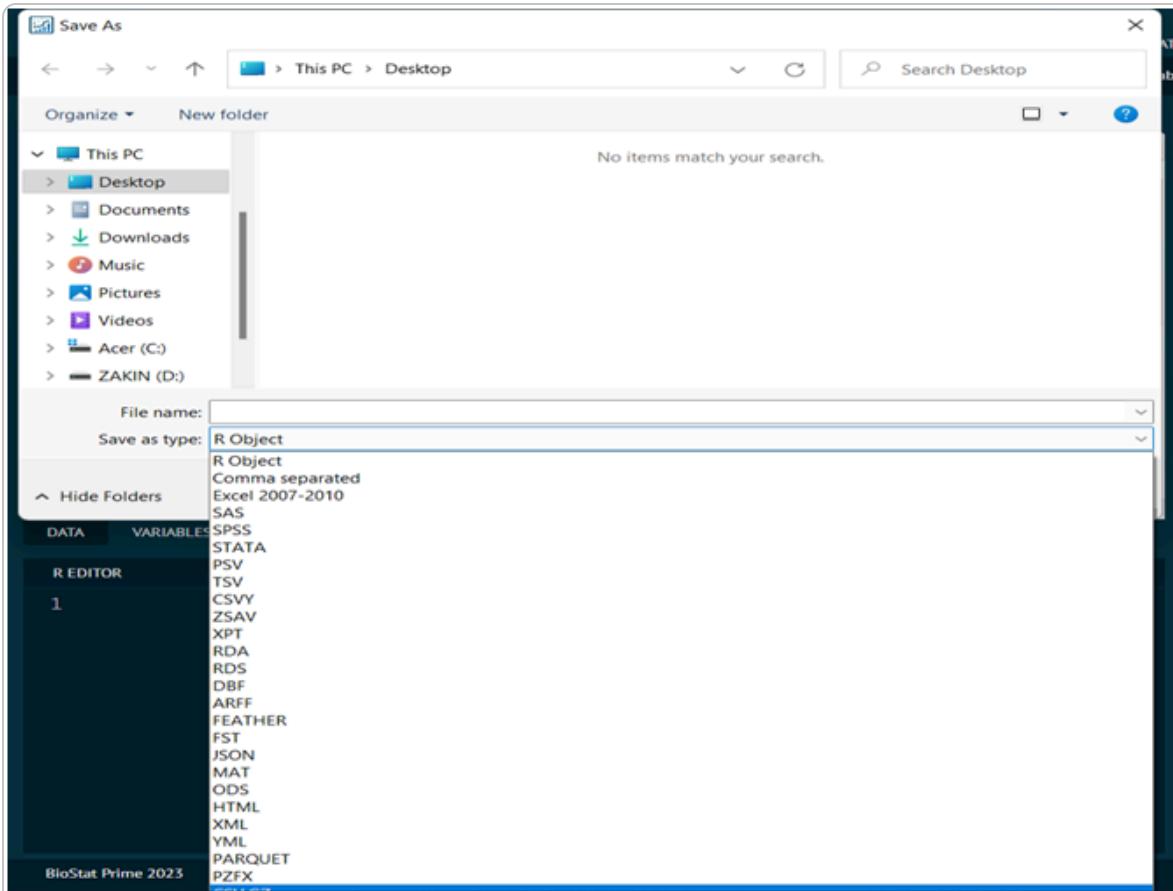
The user can decide as per the requirement which format to choose. Save seeks to update the current content of the previously saved file,

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.



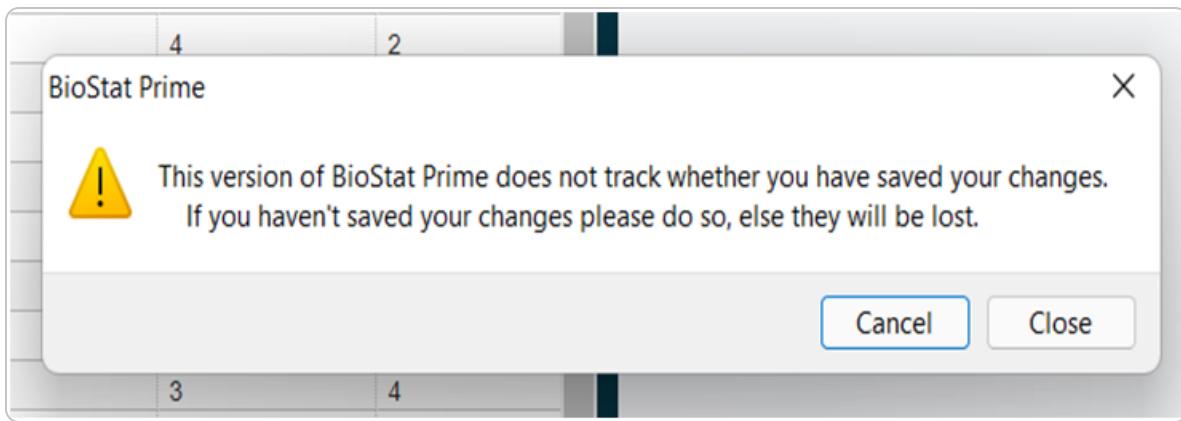
Save Dataset

Close Dataset

Closes the dataset that user had been working on.



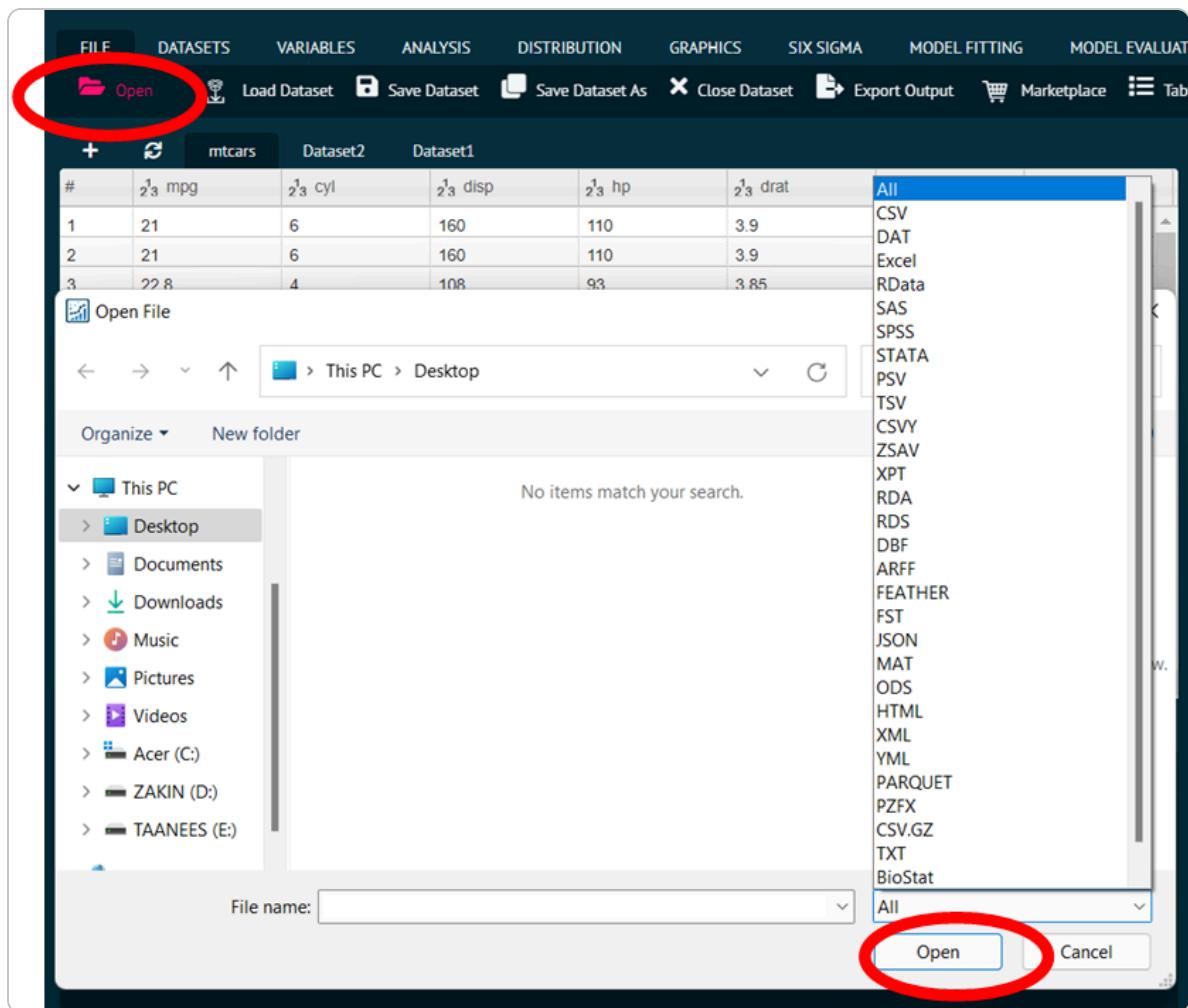
BioStat issues a warning to save the dataset before closing it.



Close Dataset

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.



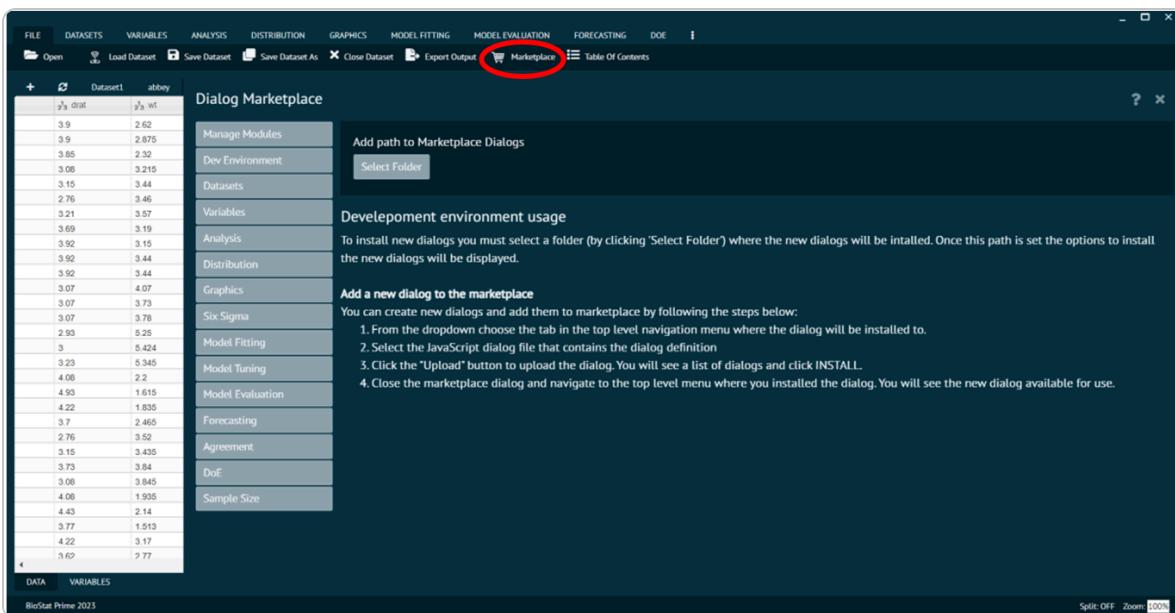
Export output

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.

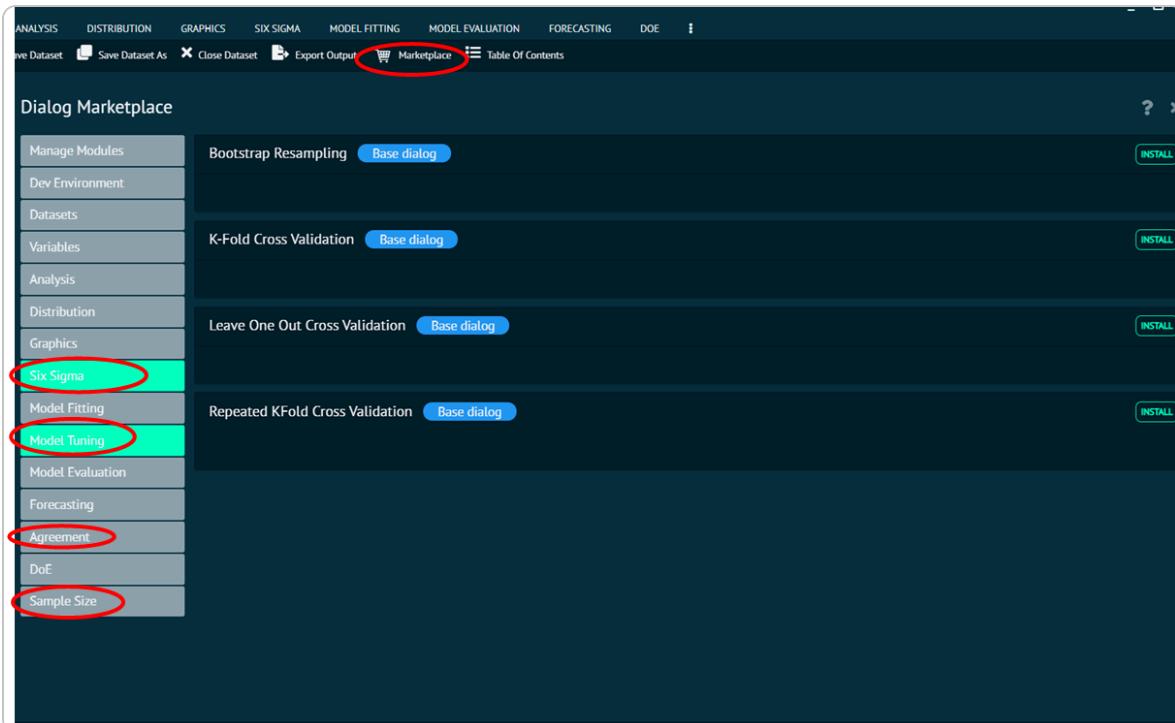
Marketplace is a free store for adding R libraries and functions to BioStat Prime to newer areas of statistics.

It Installs or hides R functions and packages. User can install various new libraries and dialogs like Six Sigma ([Six Sigma-Quality Control](#)), Agreement, Model Fitting ([Model-Curve Fitting](#)), Sample Size and the associated libraries with them.



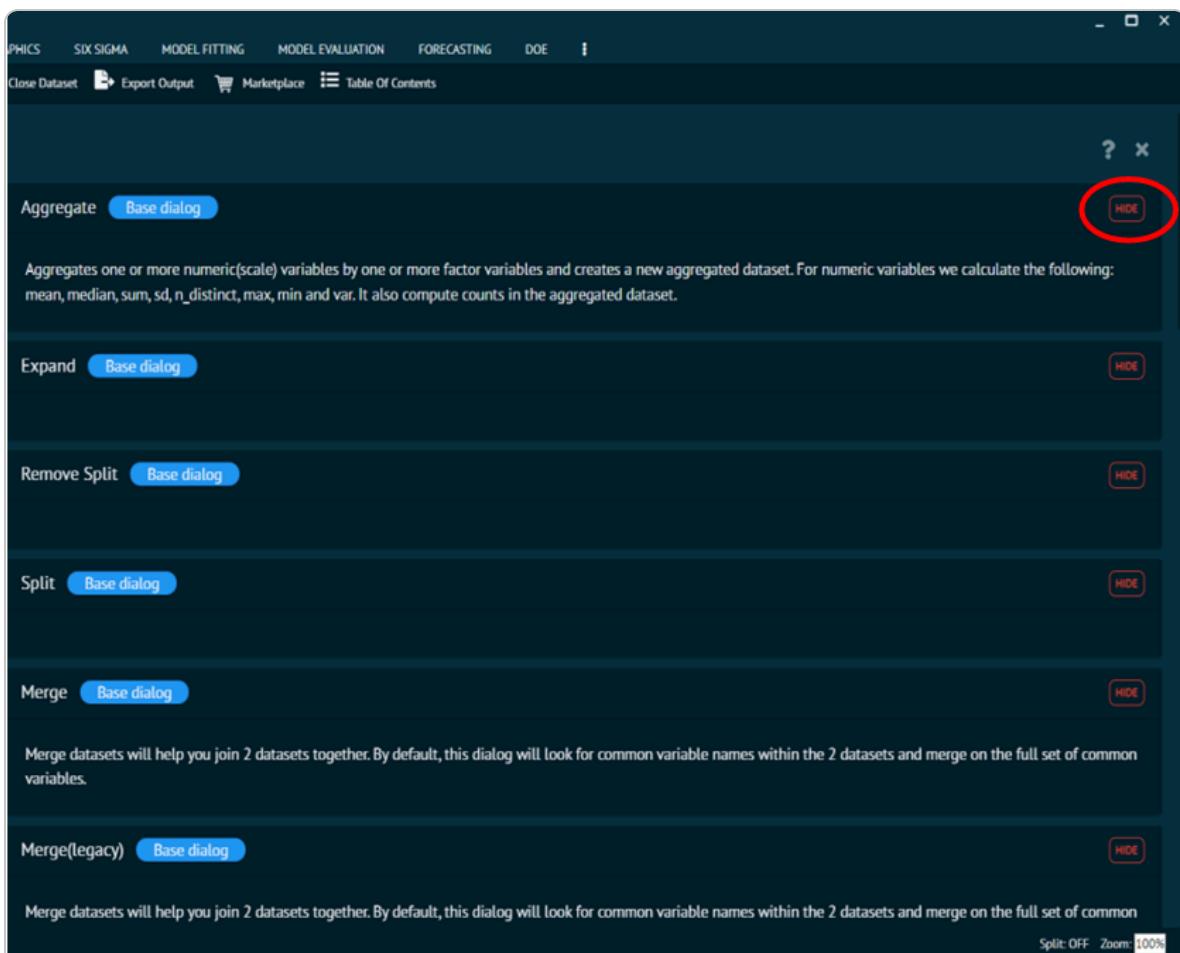
Marketplace

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



Marketplace

- i** 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.

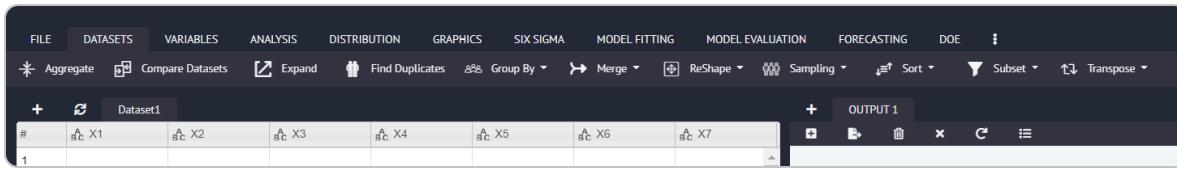


Marketplace

Dataset- Data Management

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](#)).



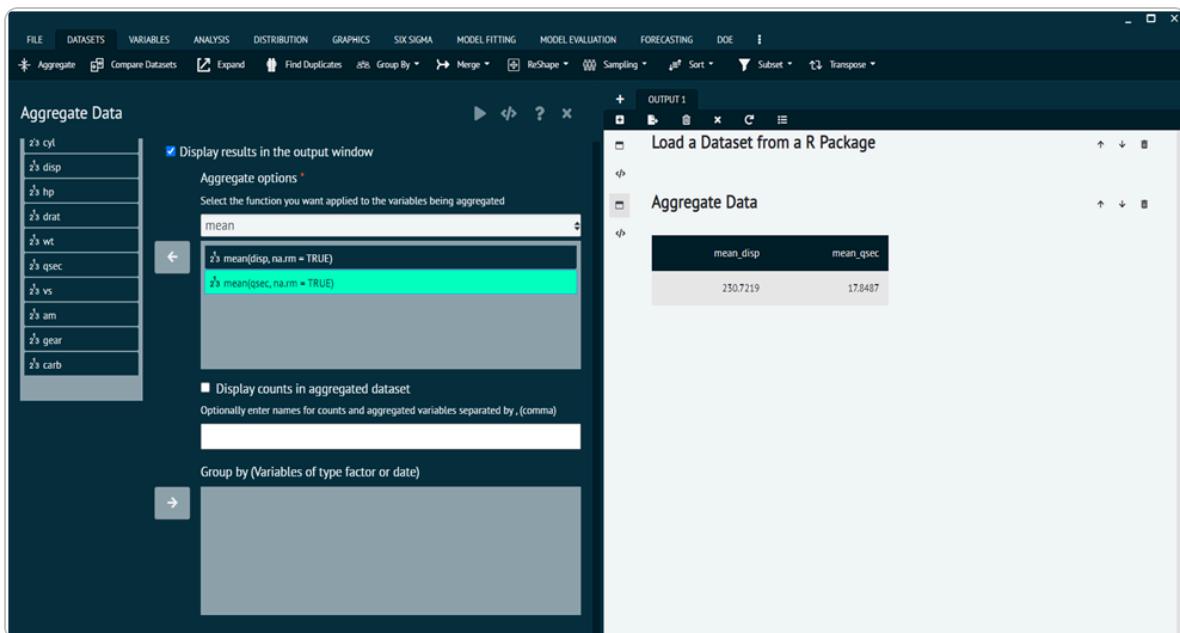
Dataset

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.



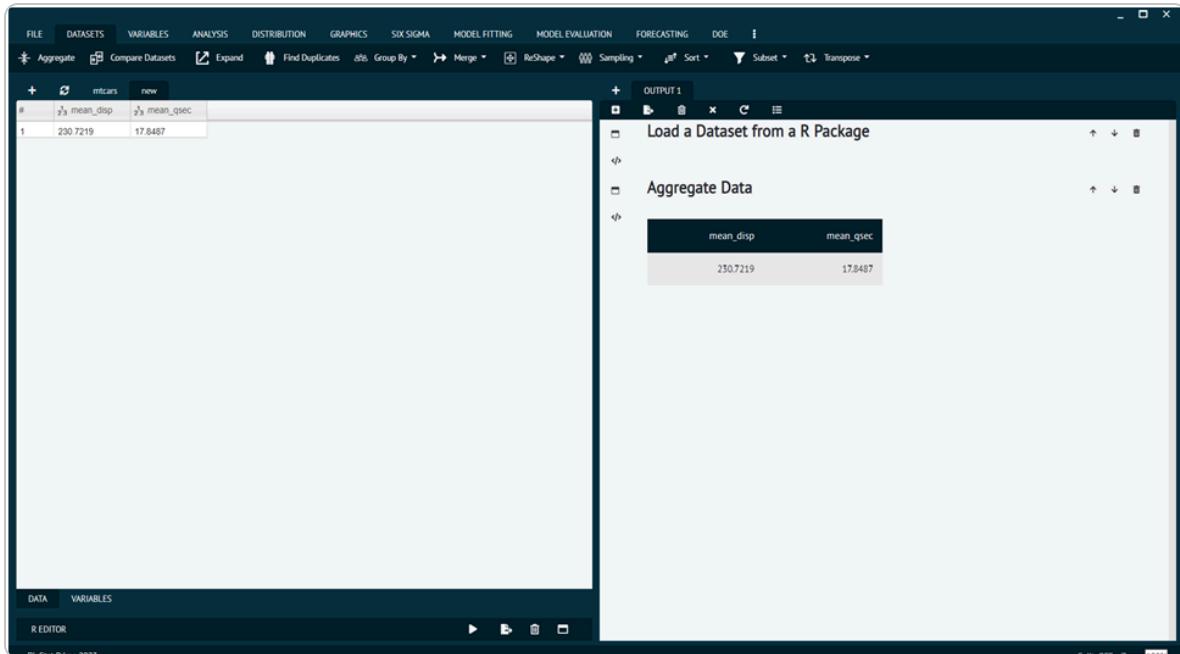
aggregate

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.



aggregate

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 below.

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 is titled 'Compare Datasets'. In the 'Source Datasets' section, 'mtcars' is selected as the first dataset and 'abey' is selected as the second dataset. A note at the bottom of this section states: 'By default, the comparison is done row-by-row. See ID Options for more options.' Below this are sections for 'Numeric Variable Tolerances' (radio buttons for Unsigned numerical difference and Unsigned percent difference, with a 'Max value of difference' input field), 'Factor Variable Tolerances' (radio buttons for Compare both underlying levels and labels, Compare underlying levels only, Compare underlying labels only, and Treat factor variables as character variables in comparisons), and 'Character Variable Tolerances' (radio button for Treat text as-is). To the right of the dialog, an 'OUTPUT' window titled 'Compare Datasets' displays two tables: 'Summary of data.frames' and 'Summary of overall comparison'. The 'Summary of data.frames' table shows:

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

The 'Summary of overall comparison' table shows:

	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

Compare Dataset

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



Required R Packages: arsenal

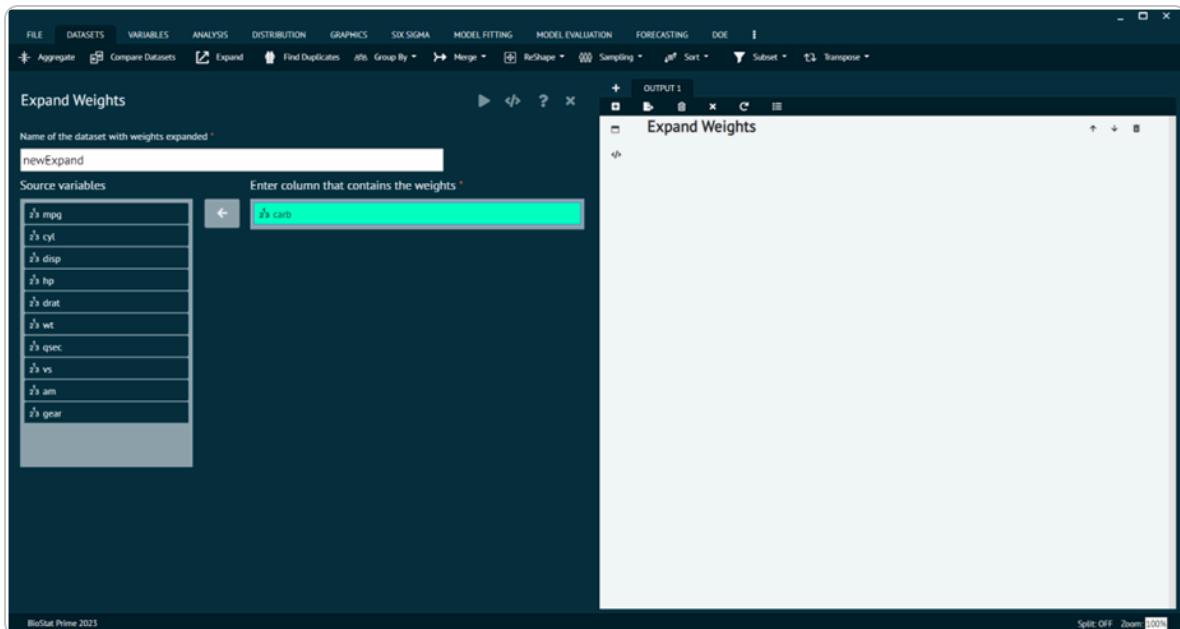
Expand

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 bellow.

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 'mtcars' dataset is displayed as a table with columns: mpg, cyl, disp, hp, drat, wt, qsec, vs, am, and gear. The 'gear' column is circled in red. On the right, an 'OUTPUT 1' window titled 'Expand Weights' is open, showing the command `Expand Weights`.

Before expanding

This screenshot shows the same BioStat Prime 2023 interface. The 'mtcars' dataset table is shown again, but now the entire row for 'gear' is circled in red. The 'Expand Weights' command remains in the output window.

Expand Before expanding

After Expanding weights.

Expand After Expanding

Expand After Expanding

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 interface. On the left, the 'Find Duplicates' dialog is open. It has sections for 'Source variables' (containing mpg, disp, drat, wt, qsec, am, gear, carb) and 'Key Variables (optional)' (containing cyl, hp, vs). There are three checked options under 'Create dataset': 'Create dataset with all rows associated with the duplicates' (Dataset name: allduprows), 'Create dataset with original data and column indicating duplicates' (Dataset name: datadupvar, Duplicate variable name: duplicate), and 'Create dataset with all duplicates removed' (Dataset name: nodupdata). A note at the top of the dialog states: 'NOTE: Specifying no key variables will result in a complete case duplicate search. Specifying key variables will search for duplicates by key variable values only.' On the right, the 'Find Duplicates' output window displays results for the mtcars dataset. It shows 'Key Variables Defining Duplicates for mtcars' (cyl, hp, vs), 'Number of Rows, Duplicates, and Rows Associated with Duplicates for mtcars' (32 total rows, 8 duplicates, 15 rows associated), and 'Frequency of Rows Associated with the Duplicates by Keys for mtcars' (a table with columns cyl, hp, vs, Freq, showing values 4, 6, 6, 2; 6, 110, 0, 2; 6, 125, 1, 2; and 1, 1, 1, 1).

Find Duplicates

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

1. Dataset with all rows associated with the duplicates.

The screenshot shows the QMplus software interface. The top menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, and MODEL EVALUATION. Below the menu is a toolbar with various icons: Aggregate, Compare Datasets, Expand, Find Duplicates, Group By, Merge, ReShape, and others. A red circle highlights the 'allduprows' button in the toolbar. The main workspace displays a dataset named 'mtcars'. The columns are labeled '#', 'cyl', 'hp', 'vs', 'mpg', 'disp', 'drat', and 'wt'. A new column titled 'allduprows' is added to the left of the first column. The 'DATA' tab is selected at the bottom.

#	cyl	hp	vs	mpg	disp	drat	wt	allduprows
1	4	66	1	32.4	78.7	4.08	2.2	
2	4	66	1	27.3	79	4.08	1.935	
3	6	110	0	21	160	3.9	2.62	
4	6	110	0	21	160	3.9	2.875	
5	6	123	1	19.2	167.6	3.92	3.44	
6	6	123	1	17.8	167.6	3.92	3.44	
7	8	150	0	15.5	318	2.76	3.52	
8	8	150	0	15.2	304	3.15	3.435	
9	8	175	0	18.7	360	3.15	3.44	
10	8	175	0	19.2	400	3.08	3.845	
11	8	180	0	16.4	275.8	3.07	4.07	
12	8	180	0	17.3	275.8	3.07	3.73	
13	8	180	0	15.2	275.8	3.07	3.78	
14	8	245	0	14.3	360	3.21	3.57	
15	8	245	0	13.3	350	3.73	3.84	

Dataset with all rows associated with the duplicates

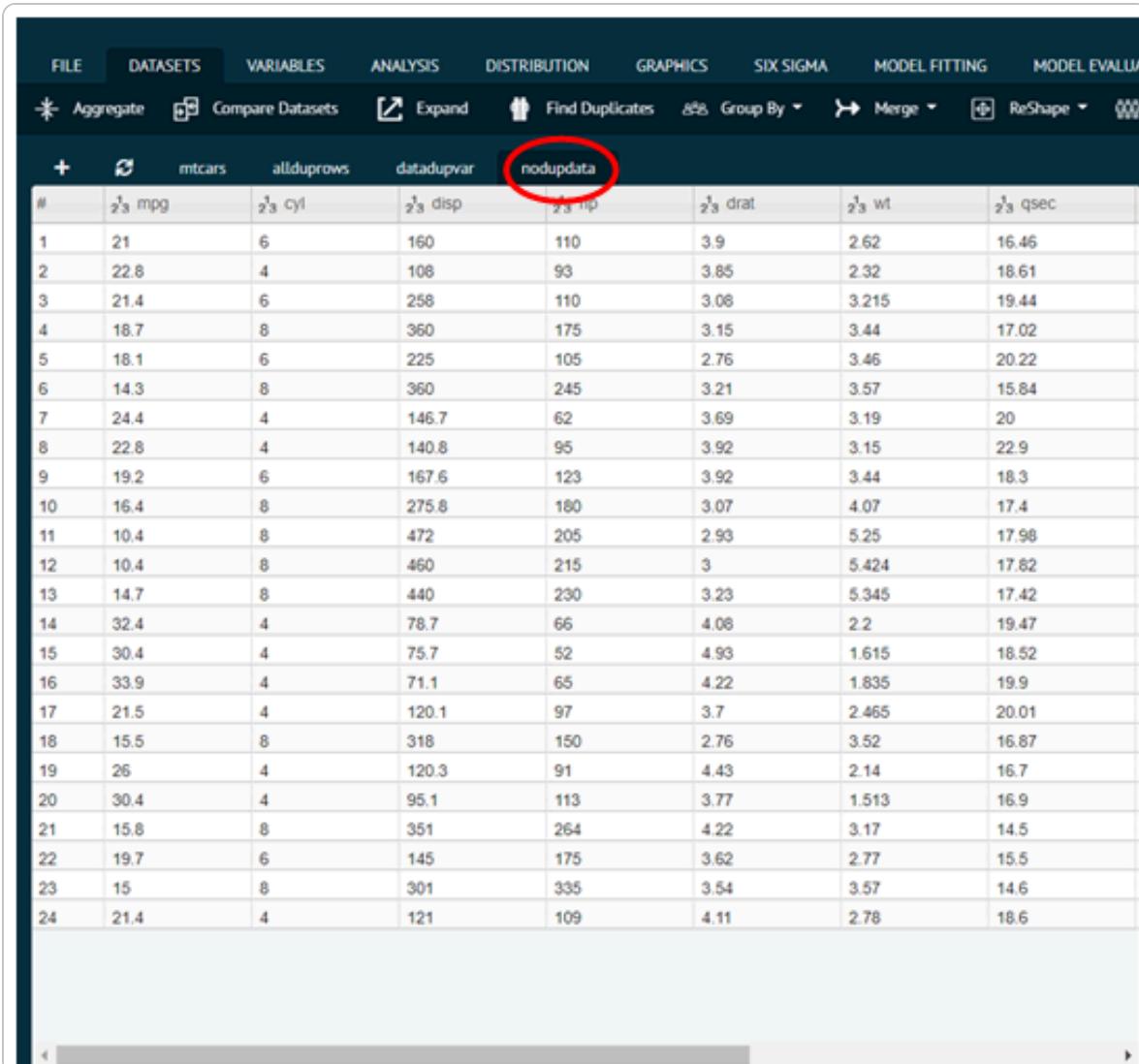
2. Dataset with original data and column indicating duplicates.

The screenshot shows the QMplus software interface with a dataset titled "mtcars". The dataset contains 28 rows of data with columns: #, mpg, cyl, disp, hp, drat, wt, and qsec. A new column, "datadupvar", has been added to the right of the original columns. This column contains binary values: 1 for rows where there are duplicates and 0 for rows where there are no duplicates. The "datadupvar" column is highlighted with a red circle. The software's menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, and MODEL EVALUATION. Below the menu is a toolbar with various icons for aggregate, compare datasets, expand, find duplicates, group by, merge, reshape, and other functions. At the bottom, there are tabs for DATA, VARIABLES, and R EDITOR, along with navigation icons.

#	mpg	cyl	disp	hp	drat	wt	qsec	datadupvar
1	21	6	160	110	3.9	2.62	16.46	0
2	21	6	160	110	3.9	2.875	17.02	0
3	22.8	4	108	93	3.85	2.32	18.61	0
4	21.4	6	258	110	3.08	3.215	19.44	0
5	18.7	8	360	175	3.15	3.44	17.02	0
6	18.1	6	225	105	2.76	3.46	20.22	0
7	14.3	8	360	245	3.21	3.57	15.84	0
8	24.4	4	146.7	62	3.69	3.19	20	0
9	22.8	4	140.8	95	3.92	3.15	22.9	0
10	19.2	6	167.6	123	3.92	3.44	18.3	0
11	17.8	6	167.6	123	3.92	3.44	18.9	0
12	16.4	8	275.8	180	3.07	4.07	17.4	0
13	17.3	8	275.8	180	3.07	3.73	17.6	0
14	15.2	8	275.8	180	3.07	3.78	18	0
15	10.4	8	472	205	2.93	5.25	17.98	0
16	10.4	8	460	215	3	5.424	17.82	0
17	14.7	8	440	230	3.23	5.345	17.42	0
18	32.4	4	78.7	66	4.08	2.2	19.47	0
19	30.4	4	75.7	52	4.93	1.615	18.52	0
20	33.9	4	71.1	65	4.22	1.835	19.9	0
21	21.5	4	120.1	97	3.7	2.465	20.01	0
22	15.5	8	318	150	2.76	3.52	16.87	0
23	15.2	8	304	150	3.15	3.435	17.3	0
24	13.3	8	350	245	3.73	3.84	15.41	0
25	19.2	8	400	175	3.08	3.845	17.05	0
26	27.3	4	79	66	4.08	1.935	18.9	0
27	26	4	120.3	91	4.43	2.14	16.7	0
28	30.4	4	95.1	113	3.77	1.513	16.9	0

Dataset with original data and column indicating duplicates

3. Dataset with all duplicates removed.



The screenshot shows the QMplus software interface. The top menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, and MODEL EVALUATION. Below the menu is a toolbar with various icons: Aggregate, Compare Datasets, Expand, Find Duplicates, Group By, Merge, ReShape, and others. A red circle highlights the 'nodupdata' icon in the toolbar. The main window displays a dataset table with columns: #, mpg, cyl, disp, hp, drat, wt, and qsec. The rows represent individual car observations from the mtcars dataset. The 'nodupdata' option is used to remove any duplicate rows from the dataset.

#	mpg	cyl	disp	hp	drat	wt	qsec
1	21	6	160	110	3.9	2.62	16.46
2	22.8	4	108	93	3.65	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

Dataset with all duplicates removed

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

Key Variables:

Specify optional key variables that define the duplicates.

- i** 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).

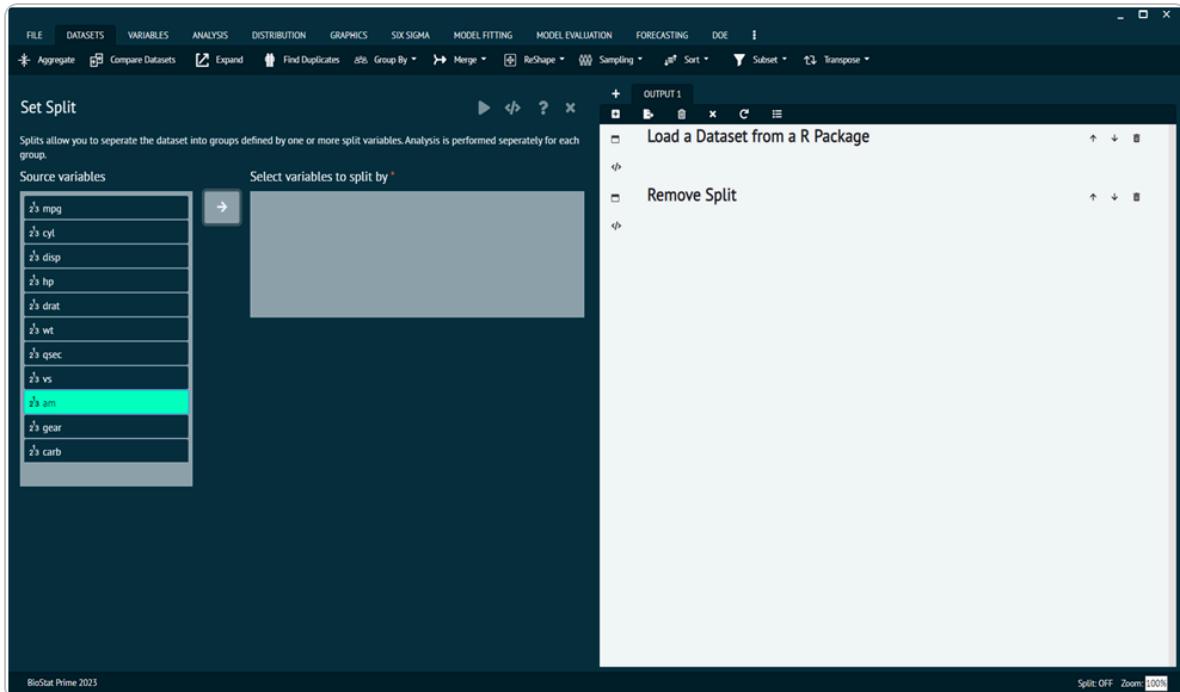
This tab has two options, viz.

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).



Remove Split

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

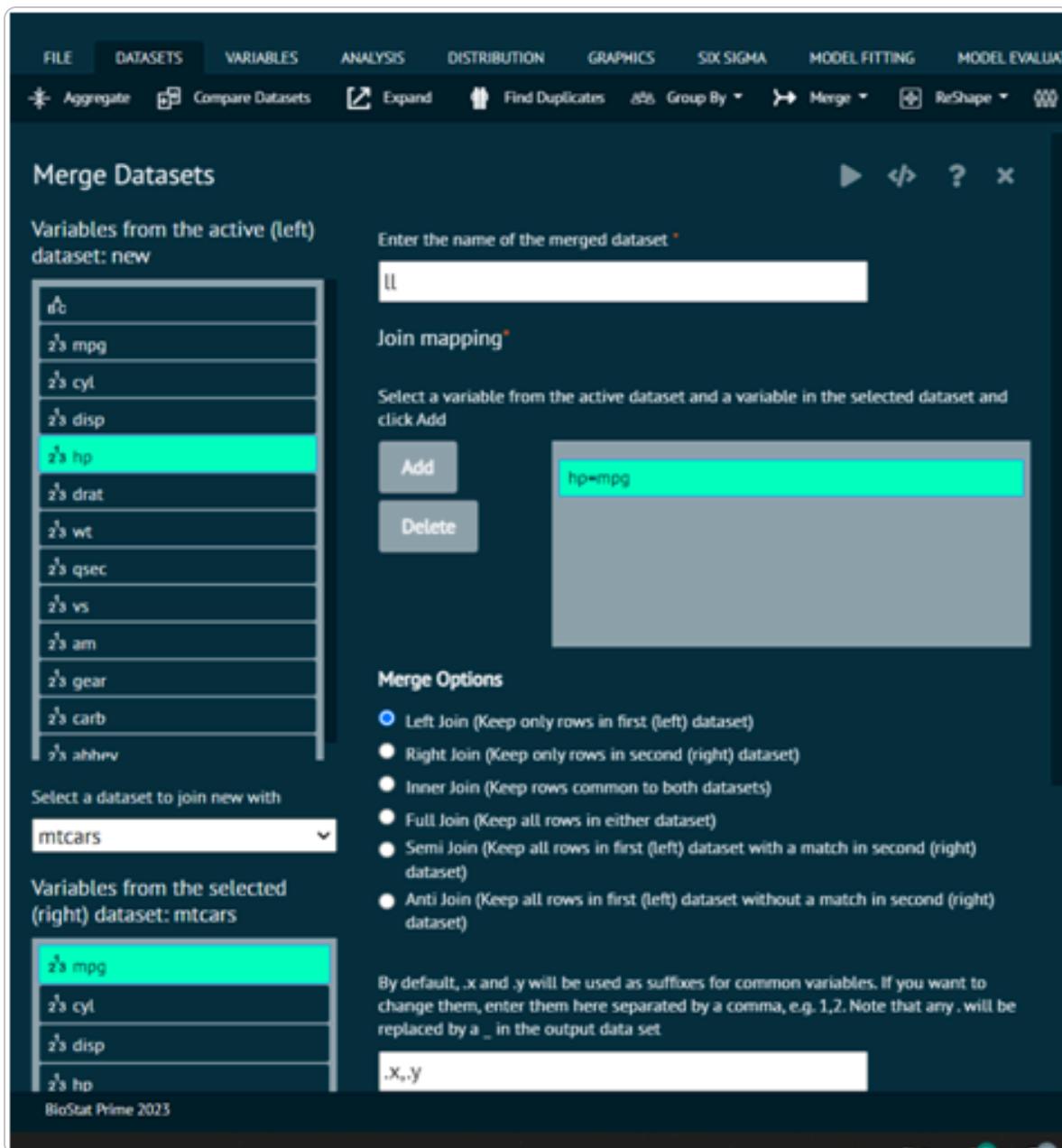
This section id dataset tab has 4 options that are explained as follows.

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 bellow.

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.

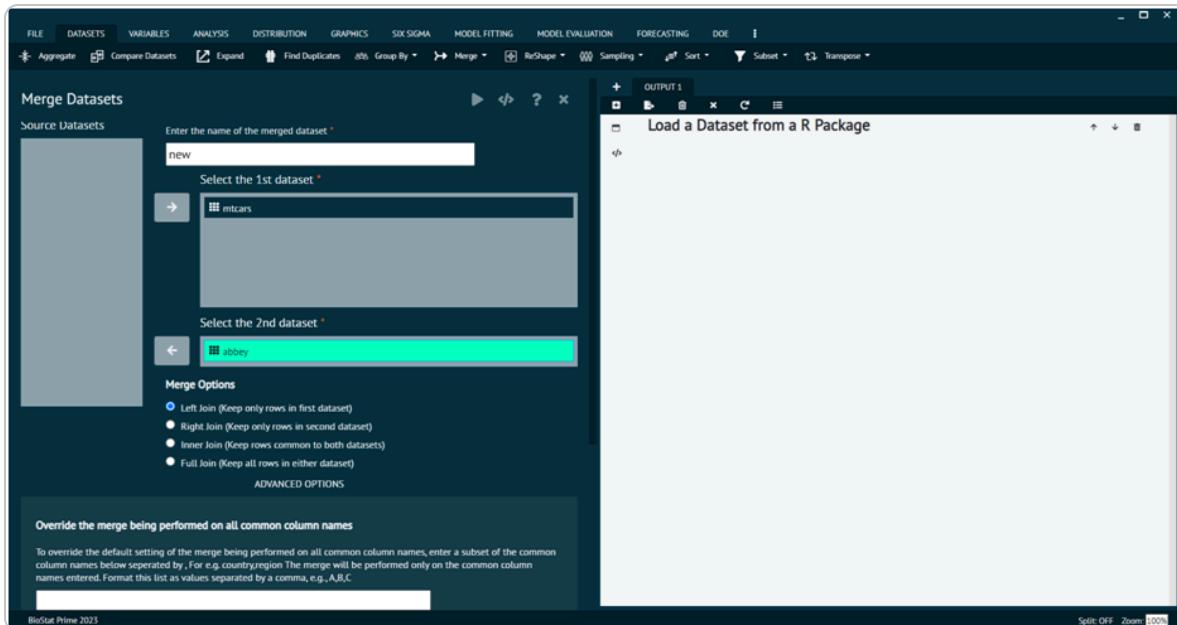


Merge



R Package Required: dplyr

Merge (legacy)



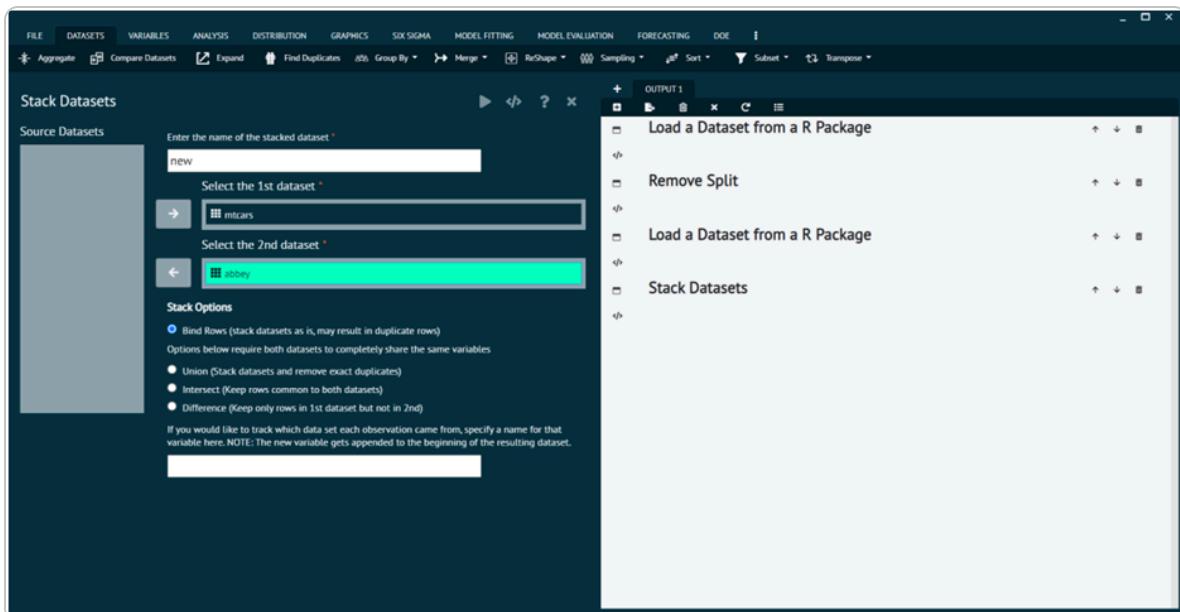
Merge (legacy)

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.



Stack

The screenshot shows the SPSS software interface. The top menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, and MODEL EVALUATION. Below the menu is a toolbar with icons for Aggregate, Compare Datasets, Expand, Find Duplicates, Group By, Merge, ReShape, and others. The main area displays the 'mtcars' dataset. The 'new' tab is active, indicated by a red circle. The 'abbey' column header is also circled in red. The data table contains columns: wt, qsec, vs, am, gear, carb, and abbey. The 'abbey' column has values ranging from 5.2 to 8.

	¹ / ₃ wt	¹ / ₃ qsec	¹ / ₃ vs	¹ / ₃ am	¹ / ₃ gear	¹ / ₃ carb	¹ / ₃ abbey
3.78	18	0	0	3	3		
5.25	17.98	0	0	3	4		
5.424	17.82	0	0	3	4		
5.345	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		
					7.4		
					8		
					8		

Stack

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.

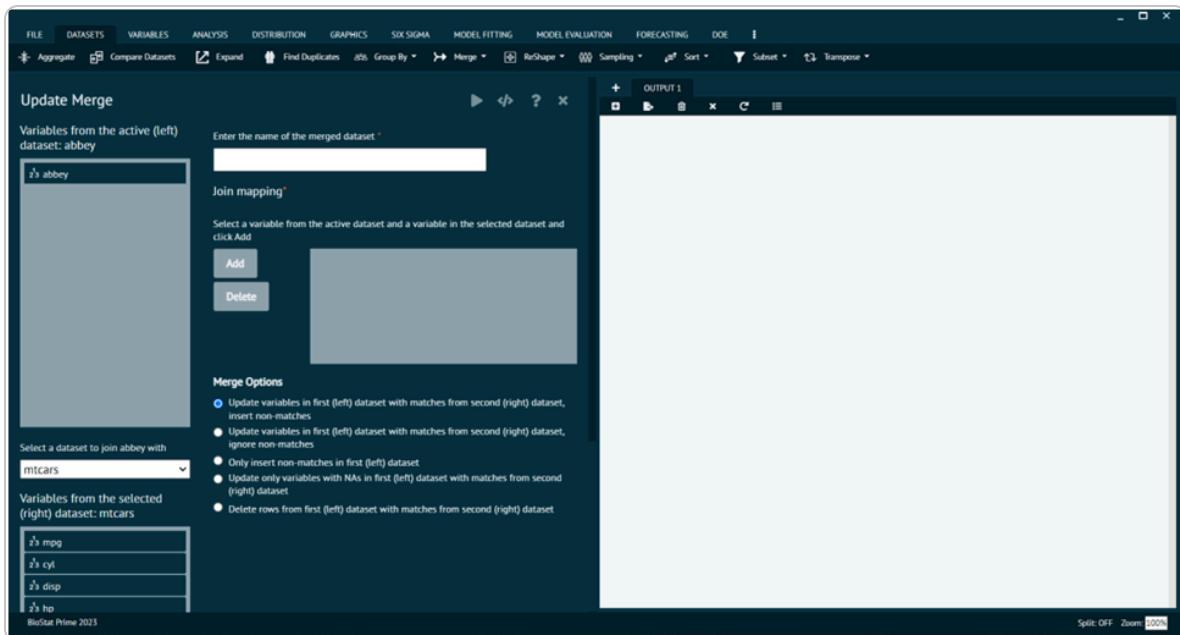
Update only variables with NAs in first (left) dataset with matches from second (right) dataset, ignore non-matches:

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.

i R Packages Required: dplyr



Merge Update

ReShape

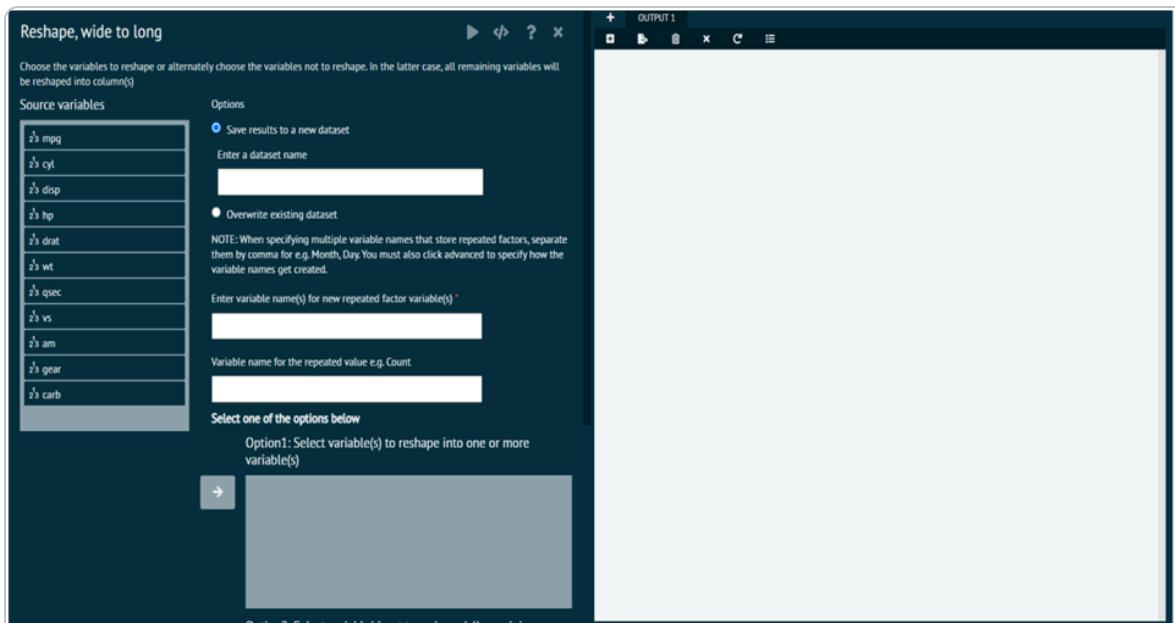
This section of the dataset tab aids the user to Reshape the loaded datasets. This tab has 2 options, viz.

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
,



Reshape wide to long

- i R package Required: tidyverse

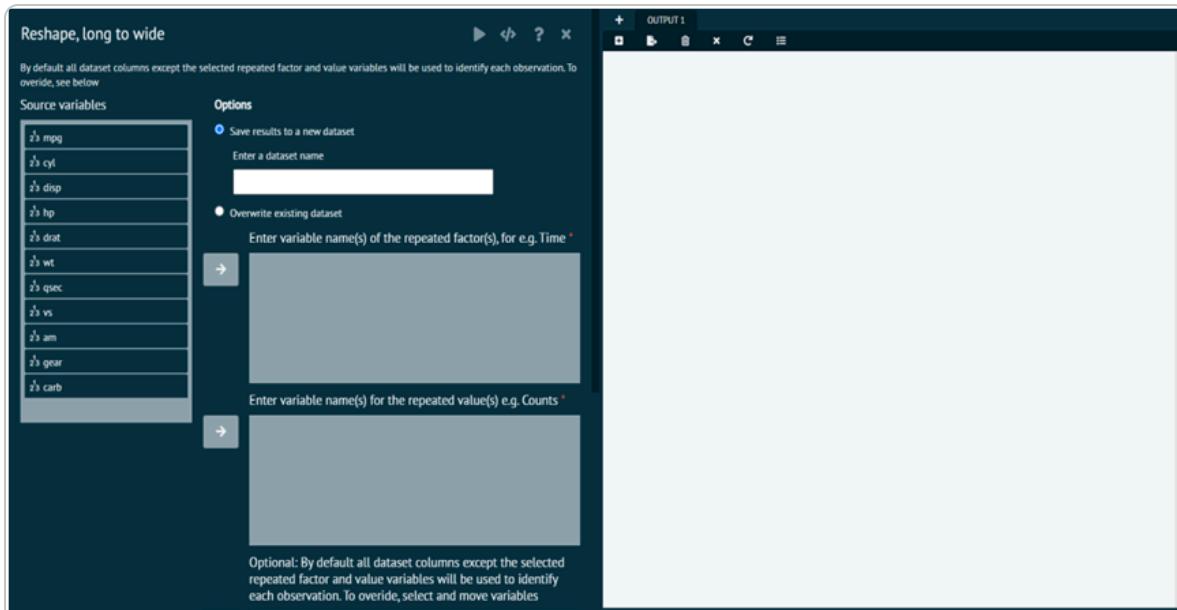
Reshape Long to Wide

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.

- i** 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

- i** By default, all dataset columns except the selected repeated factor and value variables will be used to identify each observation.



Reshape Long to Wide

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

Should sampling be with replacements

Set seed *

Random Split

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

Sample Data

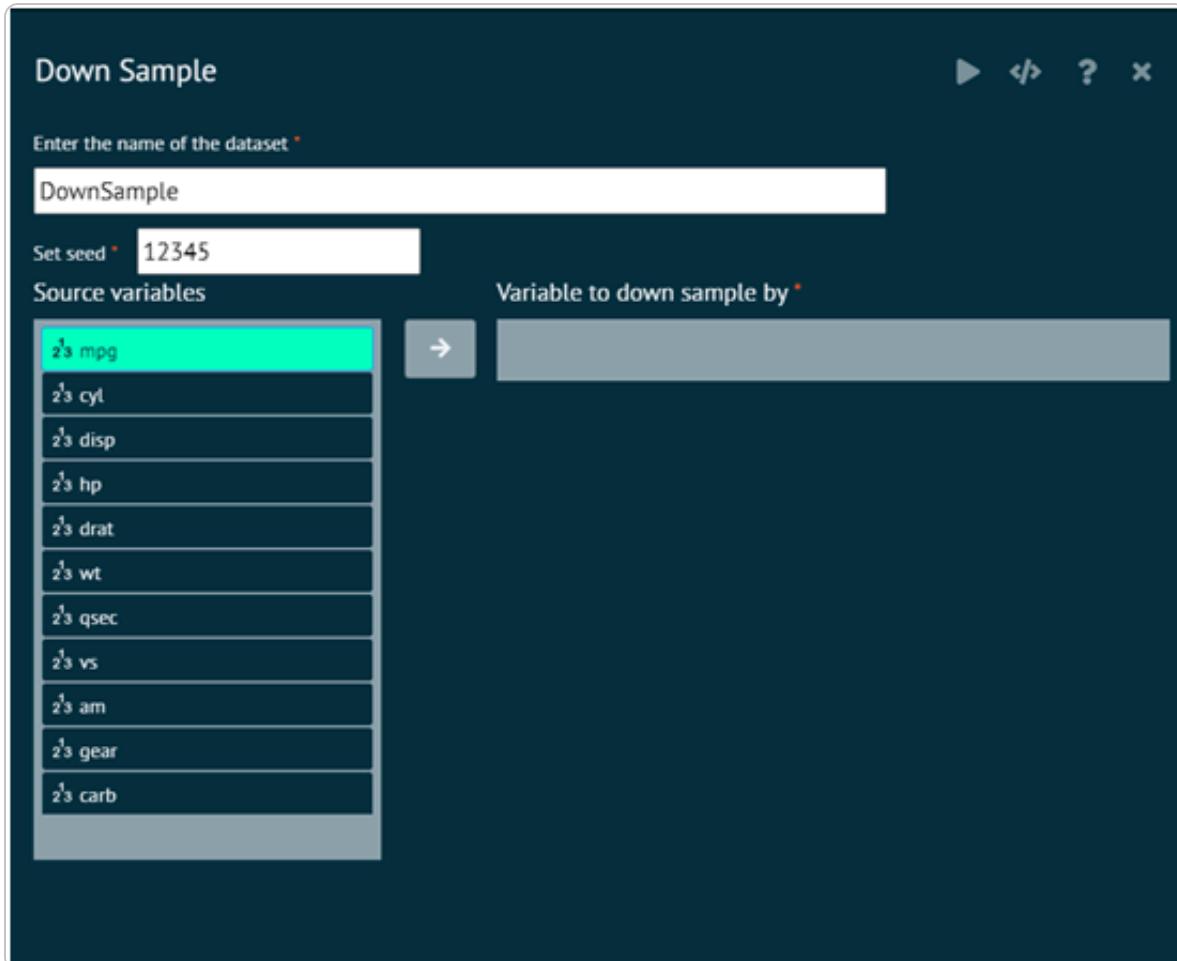
Sample Data takes a random sample of the rows from the existing dataset. Samples a % of rows or a specified number of rows with or without replacement. Saves the result to a new dataset or overwrites the existing dataset.

The screenshot shows the 'Sample Data' dialog box. At the top right are icons for navigation (right arrow, left arrow, question mark, close). The title 'Sample Data' is at the top left. Below it are two sections: 'Dataset options' and 'Sampling options'. In 'Dataset options', the radio button 'Save results to a new dataset' is selected, and there is a text input field for 'Enter a name of a dataset' which is currently empty. In 'Sampling options', the radio button 'Specify the percentage of the dataset to sample' is selected, and there is a text input field for 'Enter the percentage' containing the value '80'. Another radio button 'Specify the number of rows to select' is present, with a text input field for 'Enter the number of rows' containing the value '1'. A checkbox 'Sample with replacement' is checked. Below these options is a text input field for 'Optionally enter a variable name or formula for weights', which is empty. At the bottom is a text input field for 'Optionally set a seed for data reproducibility' containing the value '12345'.

Sample Data

Down Sample

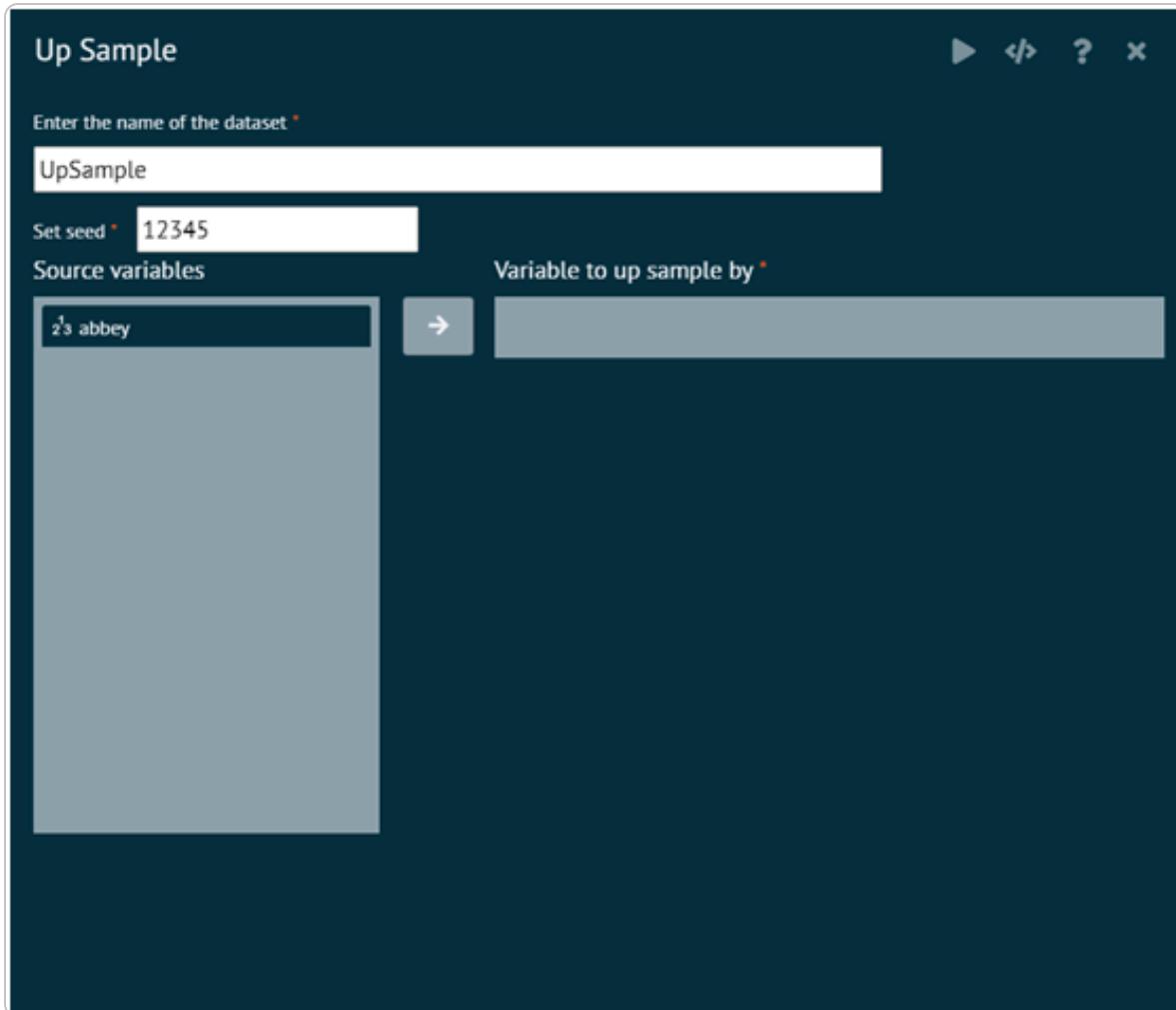
Down-Sampling Imbalanced Data. DownSample will randomly sample a data set so that all classes have the same frequency as the minority class.



Down Sample

Up Sample

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



Up Sample

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 *

traindata

Enter the name of the test dataset *

testdata

Enter the split percentage * 80

Set seed 12345

Source variables

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

Variable to construct stratified samples from *



Stratified Split

Sort

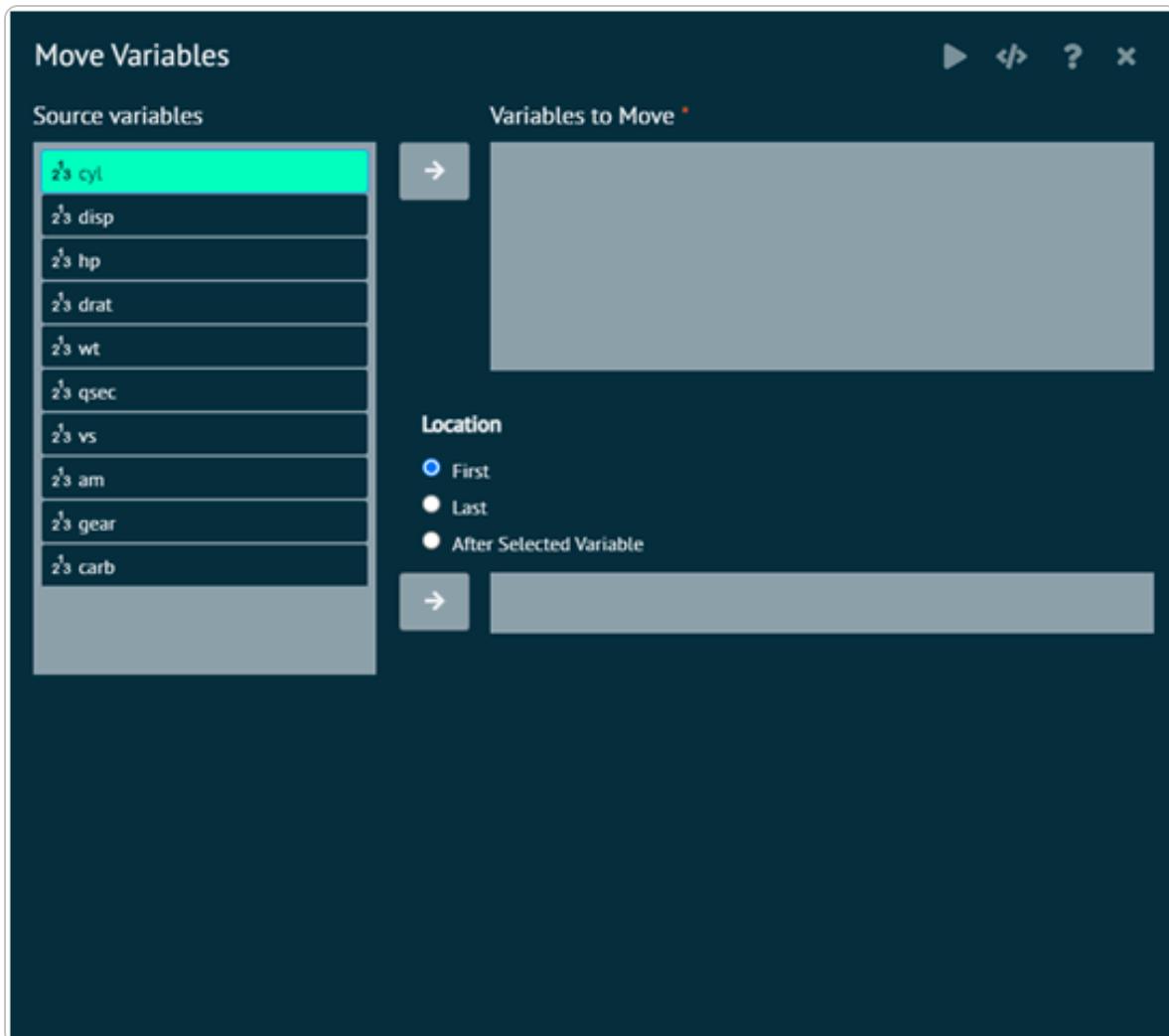
Move Variables

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.

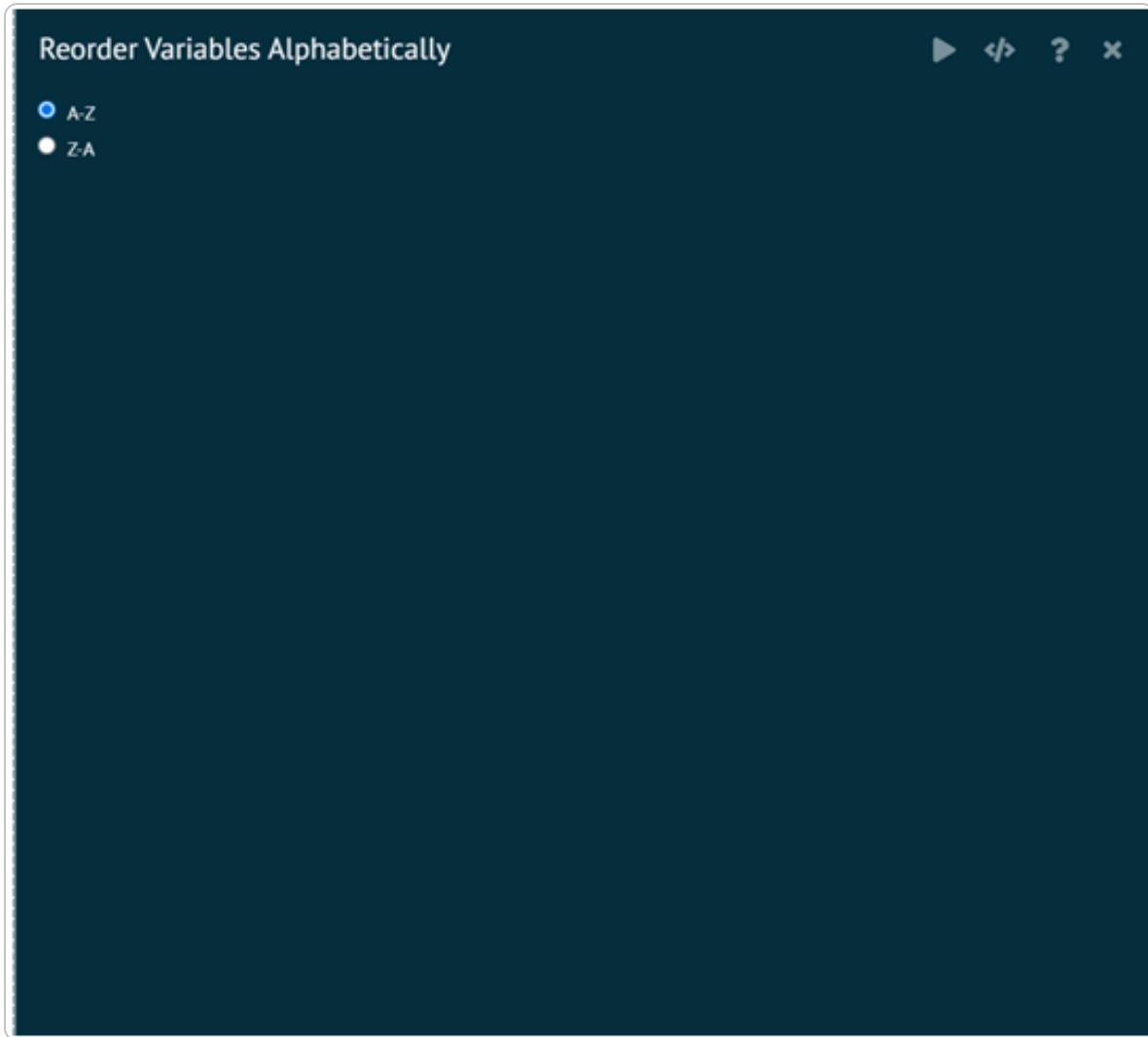
 Required R Packages: dplyr



Move Variables

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.



Reorder Variables

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

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

Sort Dataset

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

- mpg
- cyl
- disp
- hp
- drat
- wt
- qsec
- vs
- am
- gear
- 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:
`(grepl("xxx",var1) ==TRUE) | var1=="abc"`
5: Match a substring by position specify: `substr(var1,2,4) == "abc"`

Subset

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).

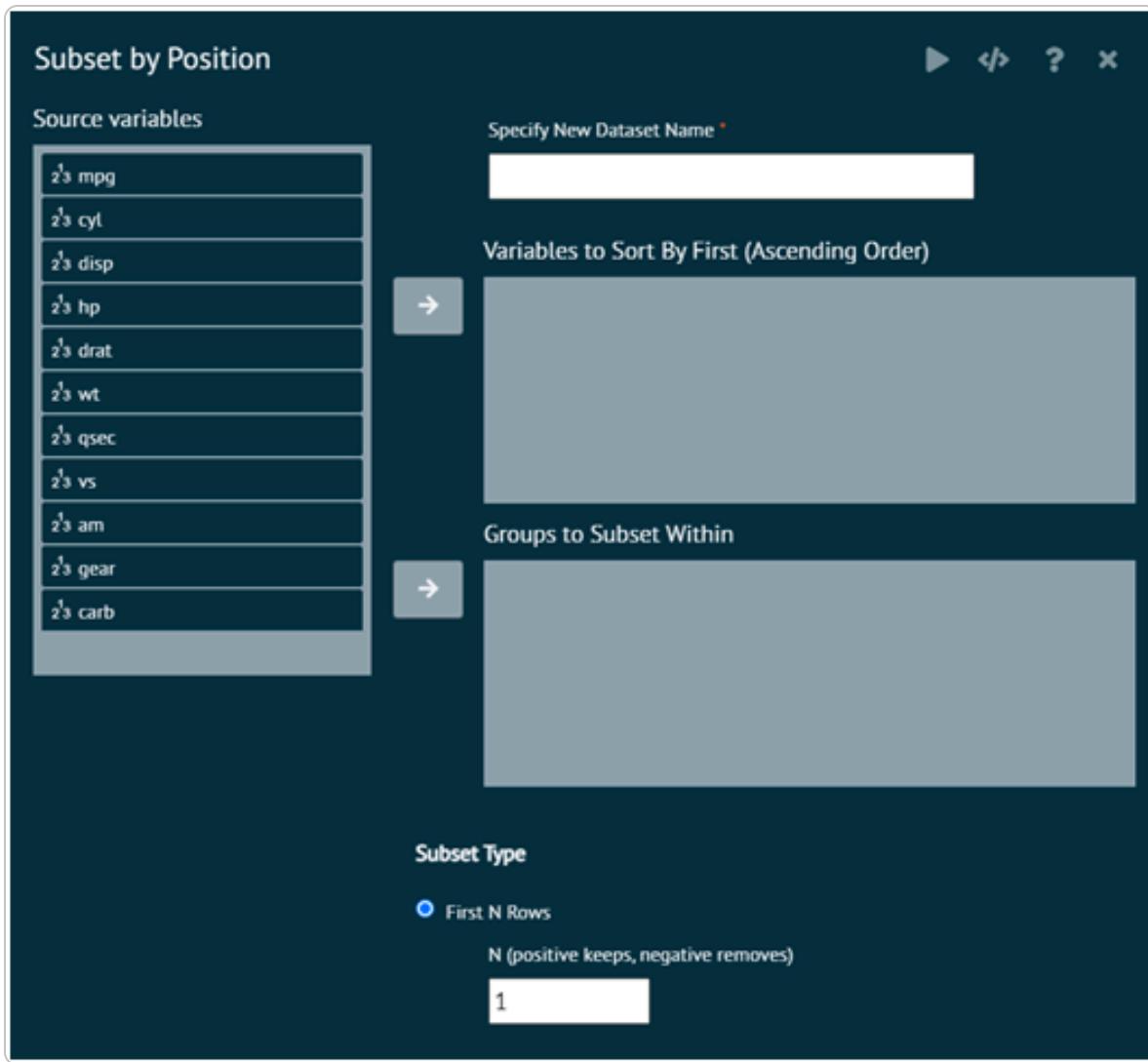
Specify Row Numbers: Keeps the exact numbered rows specified. For example, specifying 1,3,5 would keep the first, third, and fifth rows. Specifying 20:30 would keep

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



Subset Type

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.

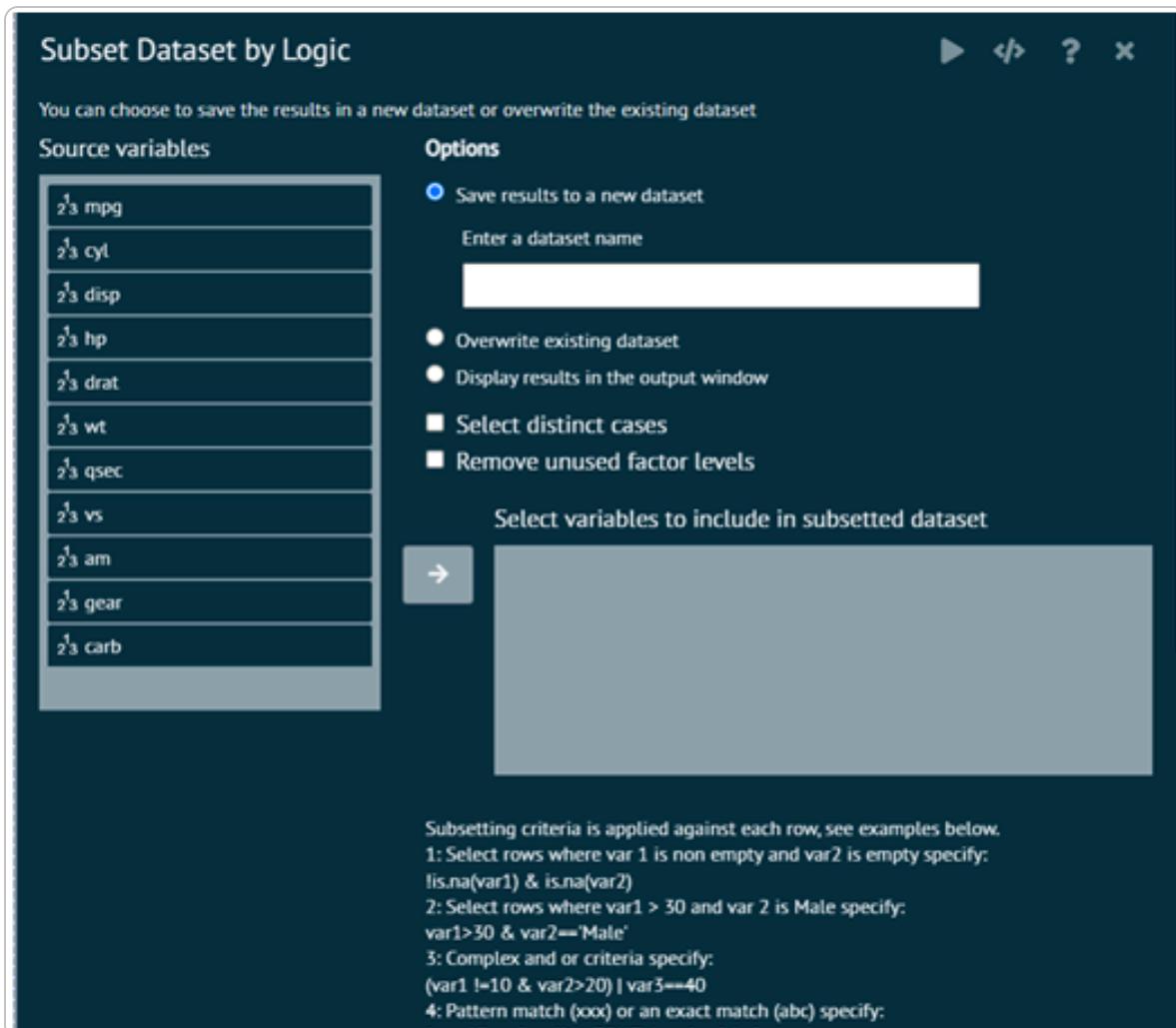
Selected distinct cases: This removes duplicates from the subsetted dataset. All variables have to be the same to be considered a duplicate.

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.

i R Package :dplyr

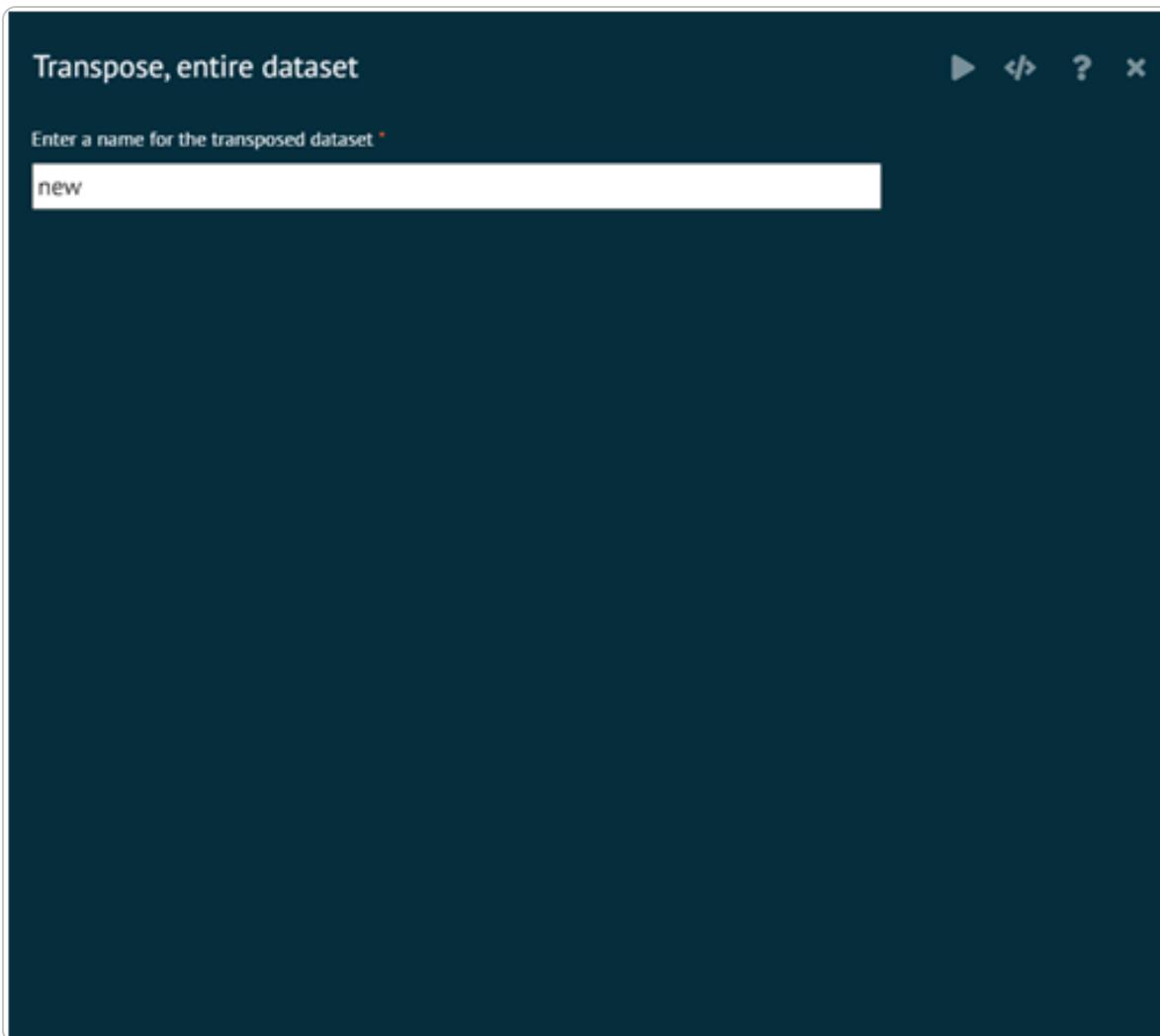


Subset by Logic

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.

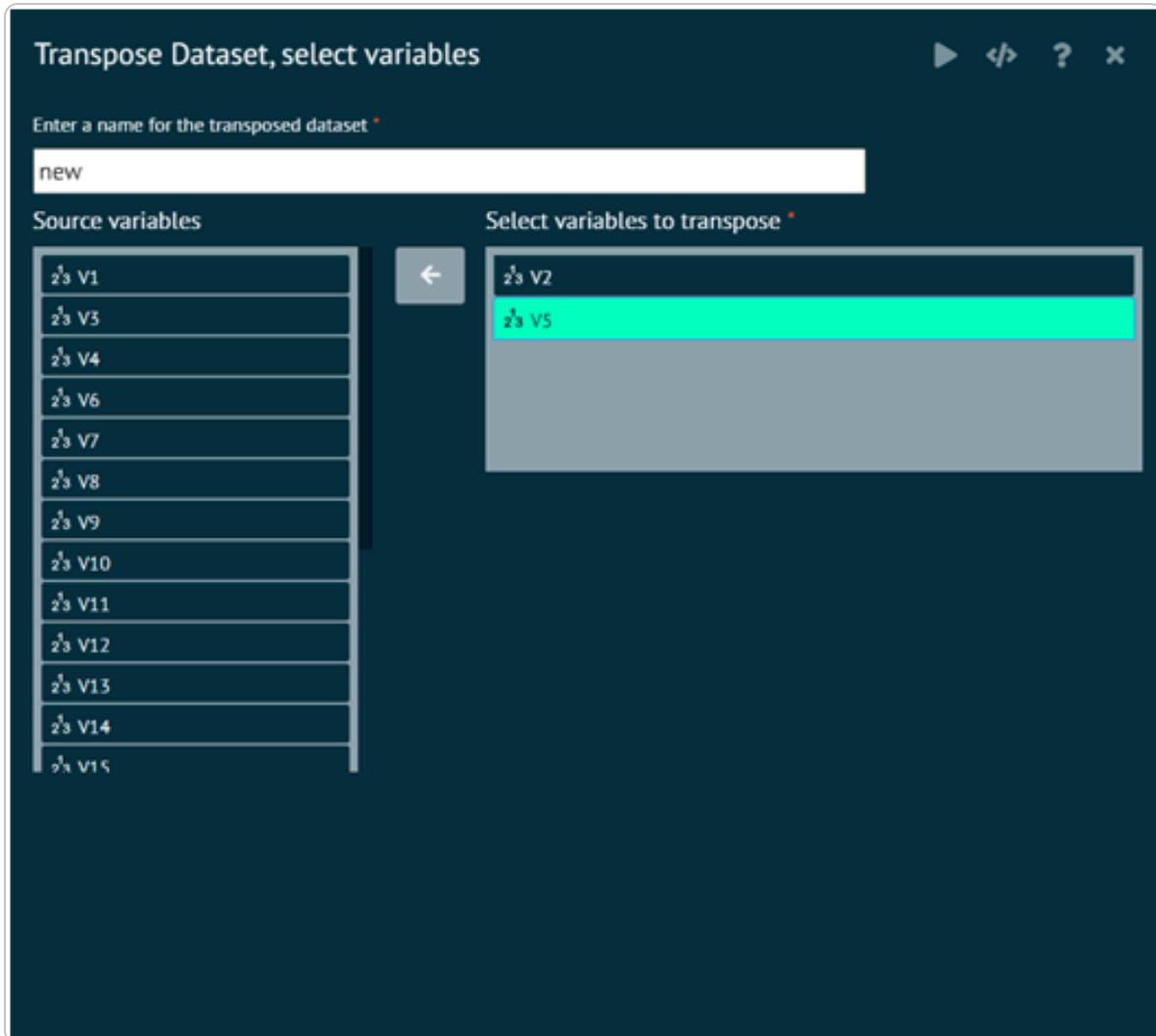


Transpose entire dataset

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.



Transpose dataset, select variables

Variables-Operations

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 ;

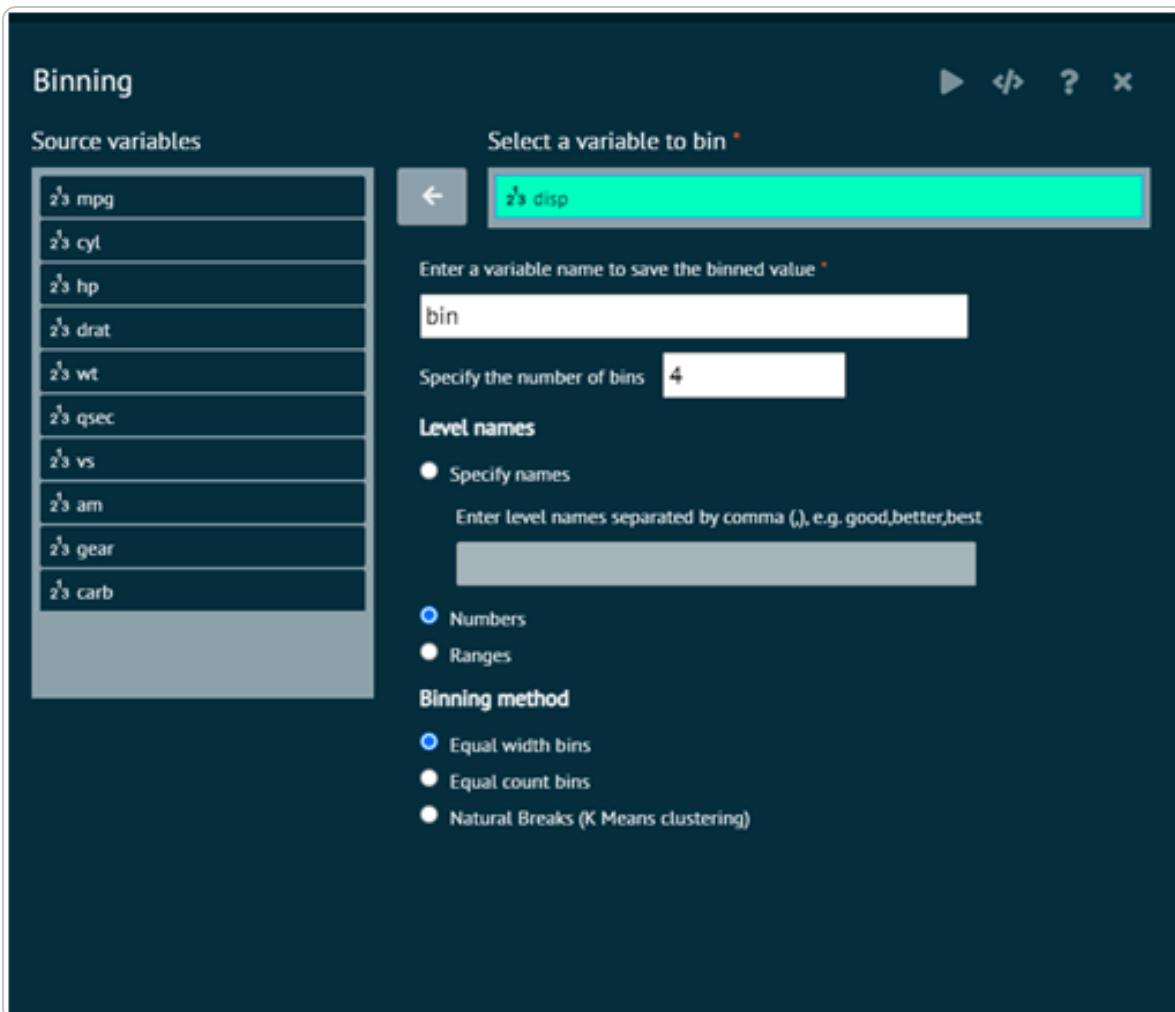
Bin, Box Cox, Compute, Concatenate, Convert, Date Order Check, Delete, Factor Levels, ID Variable, Lag or Lead Variable, Missing values, Rank, Recode, Standardize, Transform.

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.



Bin

Variable binned.

The screenshot shows the QMplus software interface with the 'Transform' tab selected. The top menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, and MODEL EVAL. Below the menu is a toolbar with icons for Bin, Box-Cox, Compute, Concatenate, Convert, Date Order Check, Delete, and a three-line menu. The main workspace displays a table titled 'Dataset1' with the 'mtcars' dataset. The table has columns: #, Name, Class, Type, Measure, and Levels. The 'bin' row is highlighted with a red oval. The 'bin' variable is defined as a factor with integer values ranging from 1 to 4.

#	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	
12	bin	factor	integer	factor	1,2,3,4

Variable binned

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)

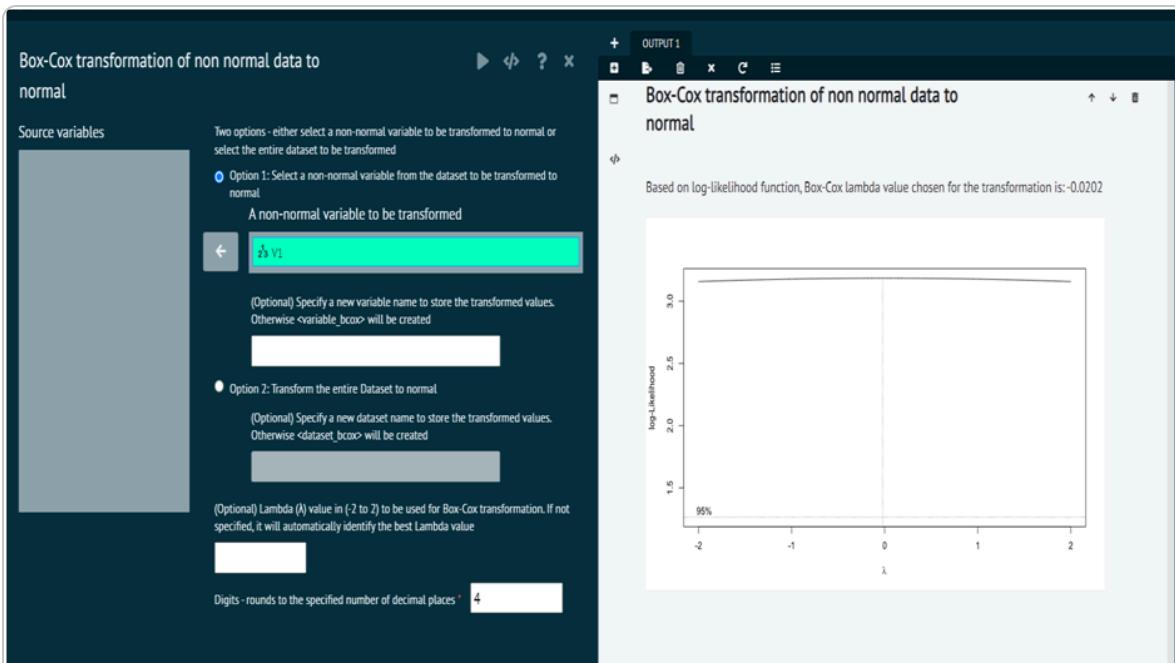
Box Cox

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)



Box Cox

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

- $\lambda = 2$: square transformation (x^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 ($\text{log}(x)$)
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 ($1/x^2$)

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)

Variable	Lambda
V1_bcox	-0.0202

Inspect Lambda

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

Add/Replace/Remove Lambda (λ) for a variable
with prior Box-Cox transformation

Source variables

#	V1	V1_bcox	ne
1	12	2.4296	3.4641
2	16	2.6964	4

Select a variable * **V1_bcox**

Box-Cox Lambda Values

Variable	Lambda
V1	
V1_bcox	-0.0202

Output 1

Lambda (λ) value for variables with prior Box-Cox transformation

Box-Cox Lambda Values

Variable	Lambda
V1	
V1_bcox	-0.0202

Add/Replace/Remove Lambda (λ) for a variable with prior Box-Cox transformation

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

Add/Remove Lambda

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

DATA VARIABLES

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

Source variables

#	V1	V1_bcox	ne
1	12	2.4296	3.4641
2	16	2.6964	4

Select a variable * **V1**

(Optional) Specify a new variable name to store the converted values. Otherwise <variable>.Invbcx will be created
ne

Option 1: Select a variable to be converted back from a prior Box-Cox transformation

Option 2: Type in a numeric value to be converted back from Box-Cox transformation
5

(Optional) Specify a Lambda (λ) value in (-2 to 2) to be used to convert back from Box-Cox transformation. Otherwise, if left blank, the Lambda (λ) value will be used that was used for the original Box-Cox transformation for the variable selected above
2

Digits - rounds to the specified number of decimal places * **4**

Inverse Box-Cox

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)

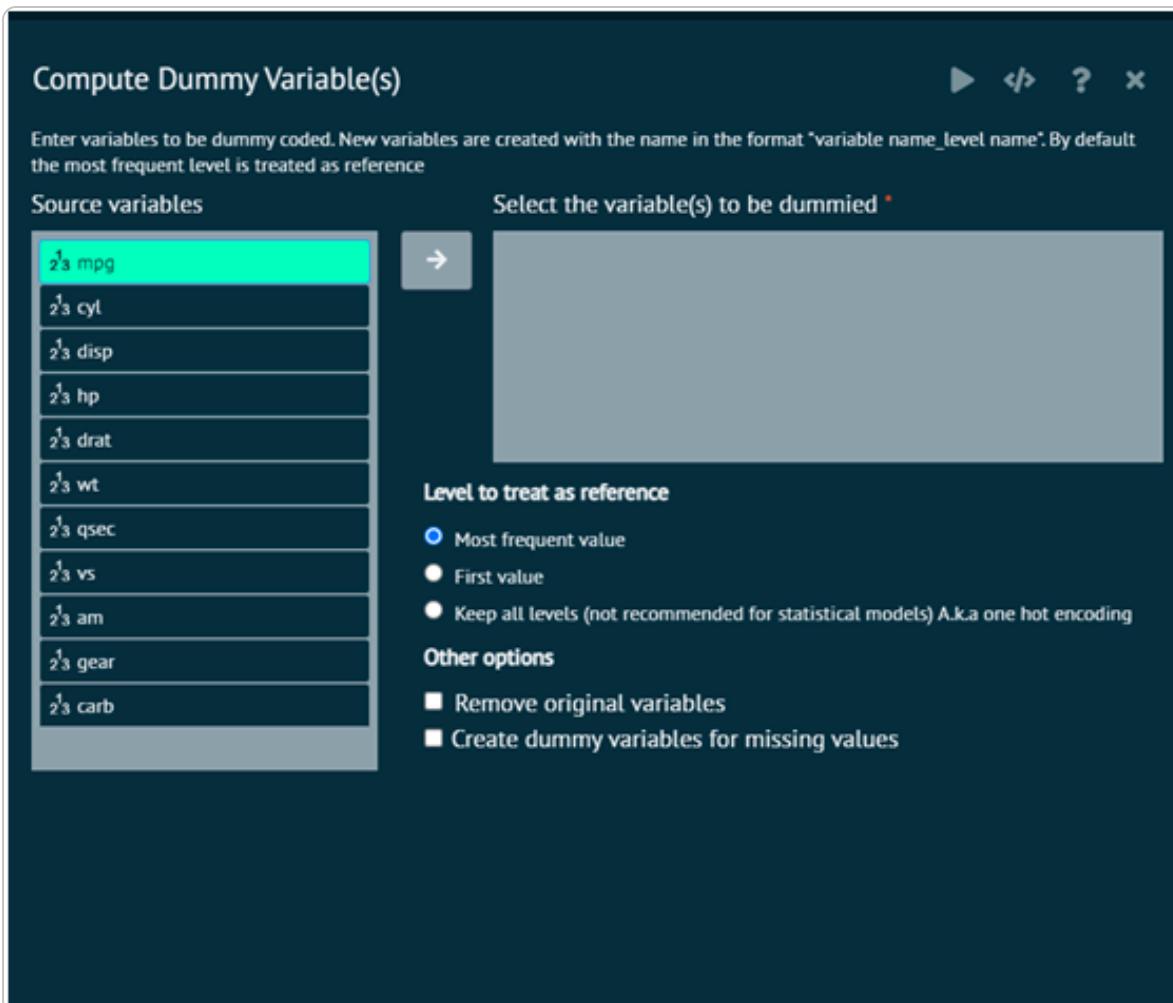
The screenshot shows the Compute interface with the title "Applying a function to all rows of selected variable(s)". It includes a toolbar with icons for back, forward, search, and close. Below the title, a sub-instruction says "Create a new variable or overwrite an existing variable by applying a function to all row values of the selected variable(s)." A "Source variables" sidebar lists columns from the mtcars dataset: mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, carb. To the right, there's a search bar labeled "Enter a new variable/Overwrite an existing variable" and a "Select variable(s)" button. A large gray area is labeled "Select an operation to apply". A dropdown menu shows "mean" highlighted in green, along with median, min, max, and sd.

Applying Function to all Rows

- ⓘ 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".

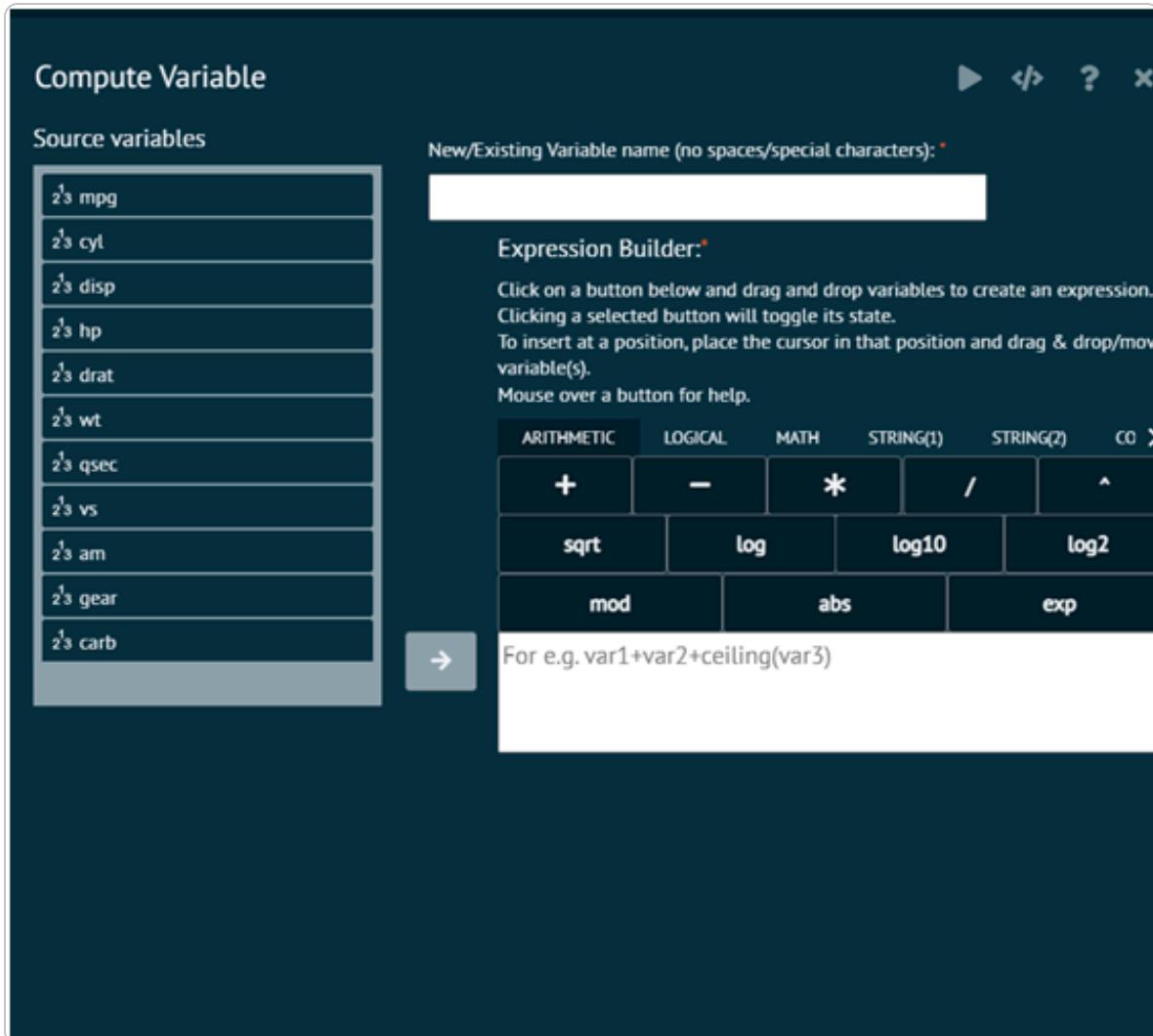


Dummy code

- 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.



Compute Variable

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

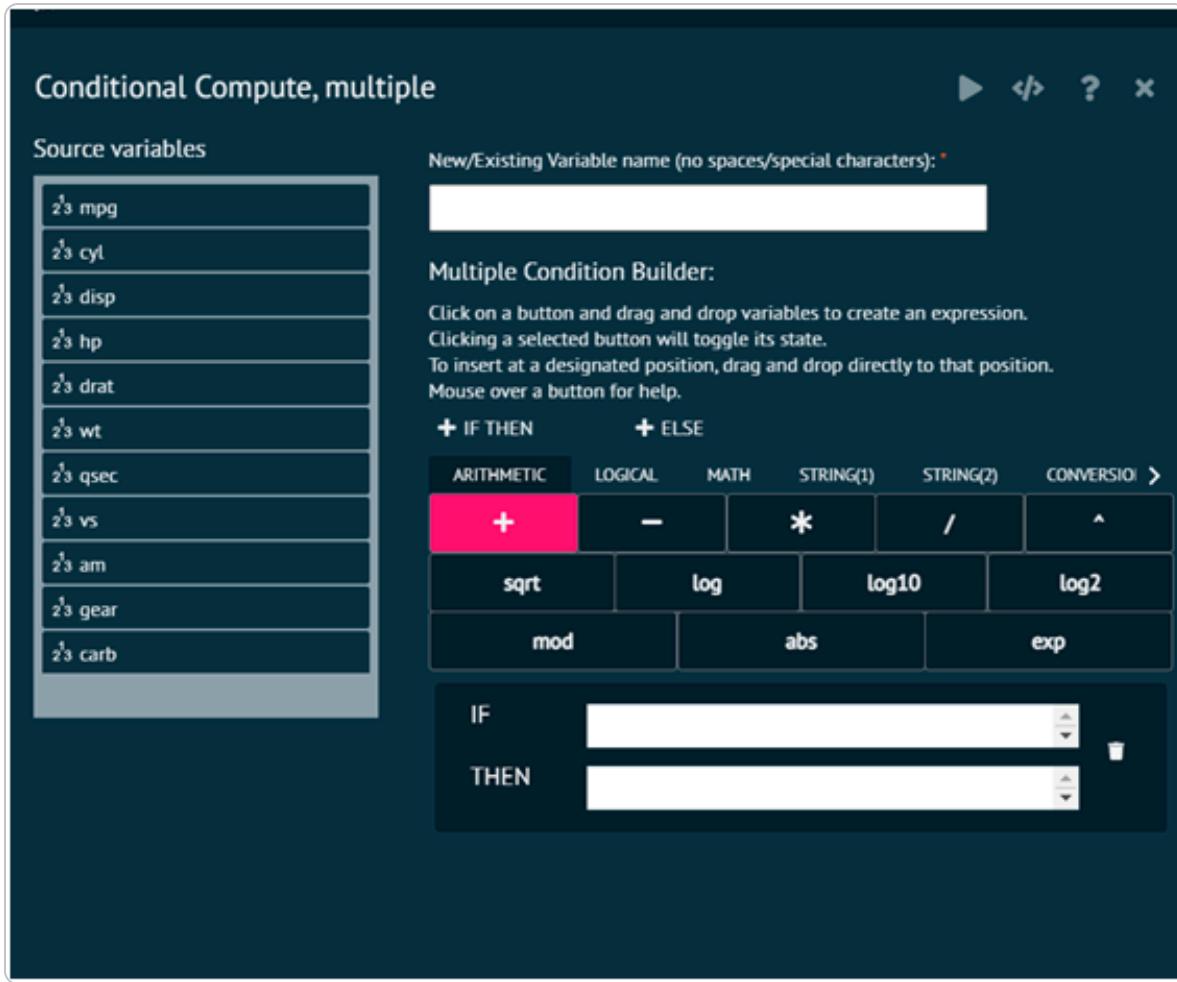
Value when condition is TRUE*

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 >
+	-	*	/	^	

Conditional Compute

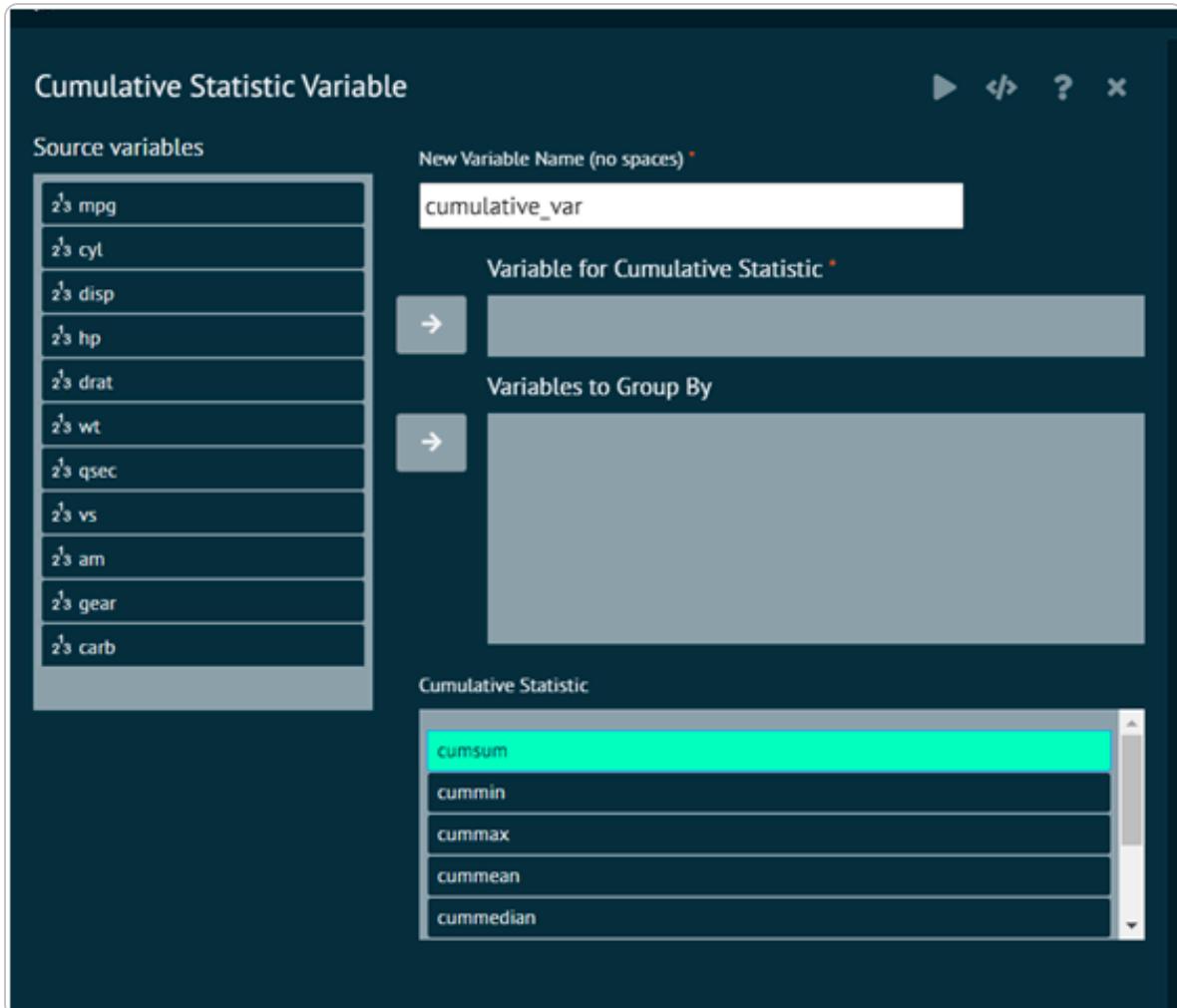
Conditional Compute Multiple



Conditional Compute Multiple

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.



Cumulative Statistics Variable

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

i 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)

Concatenate Variables

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 new/existing variable for concatenated results *

Select the variables to concatenate *

←

Enter an optional separator for the concatenated variables

Concatenate

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.

Convert Date Variables To Character

Select a suffix or prefix for converted variables

Suffix

Prefix

Enter a prefix or 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

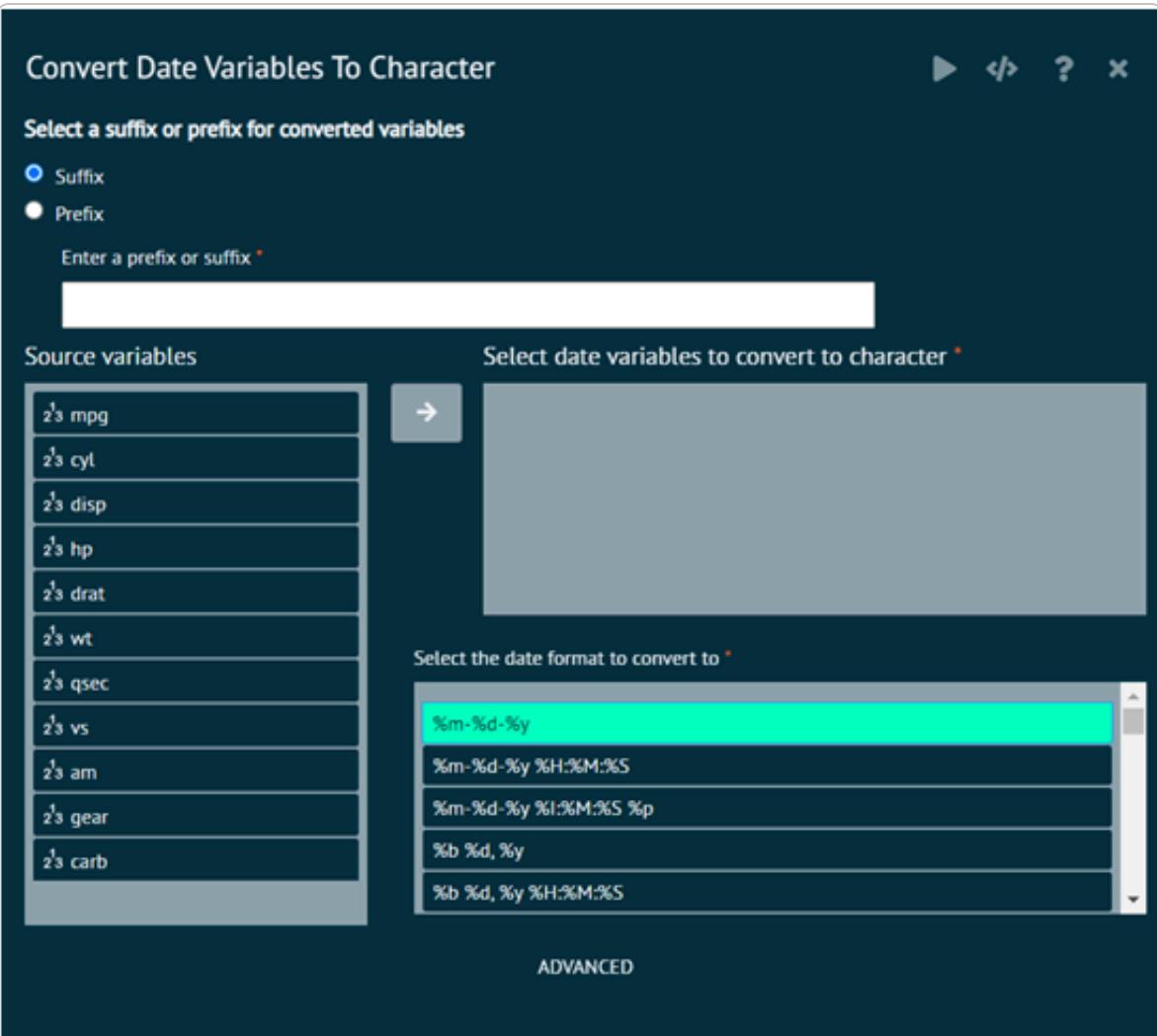
Select date variables to convert to character *

→

Select the date format to convert to *

- %m-%d-%y
- %m-%d-%y %H:%M:%S
- %m-%d-%y %I:%M:%S %p
- %b %d, %y
- %b %d, %y %H:%M:%S

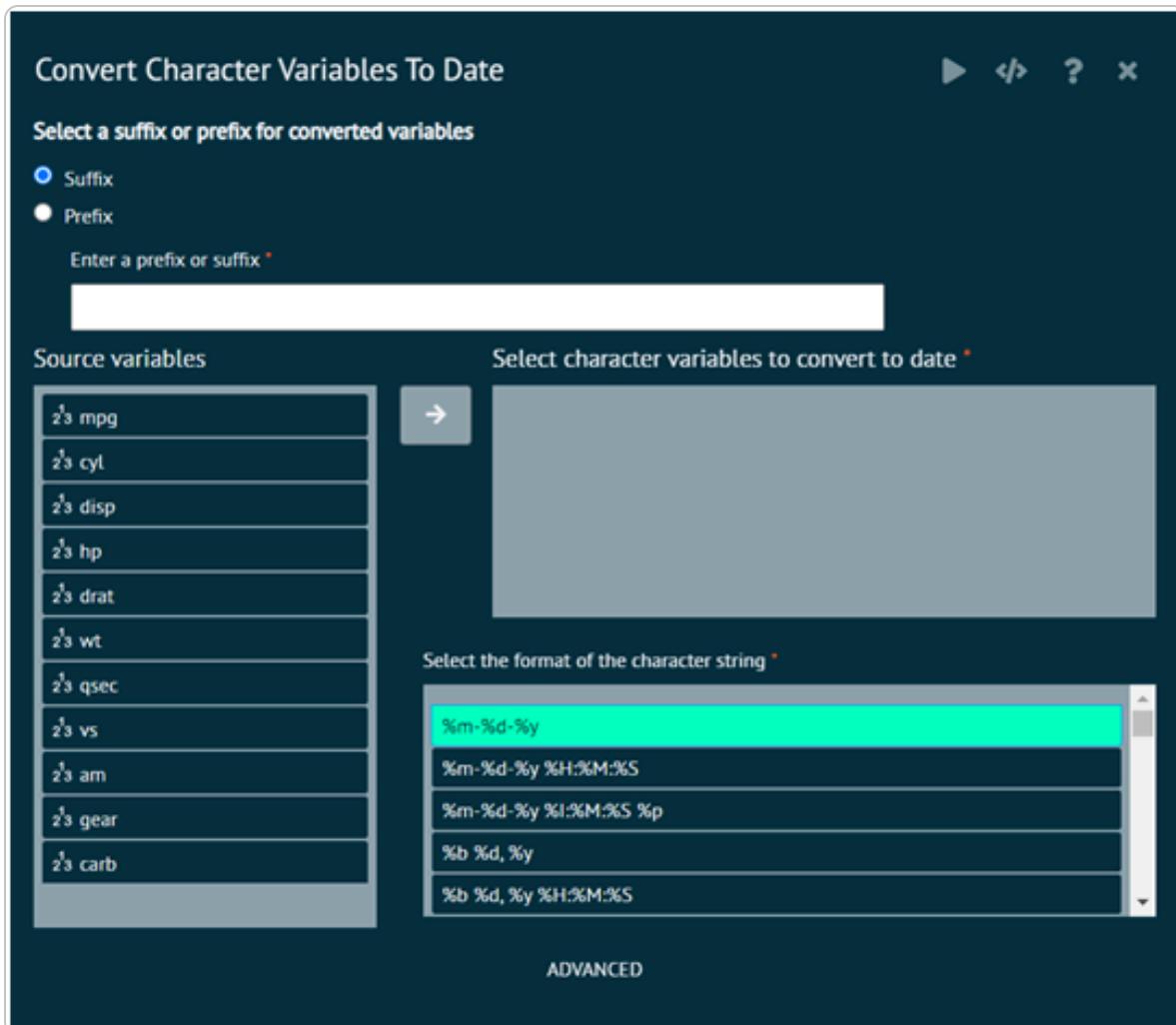
ADVANCED



Date to Character

Character to Date

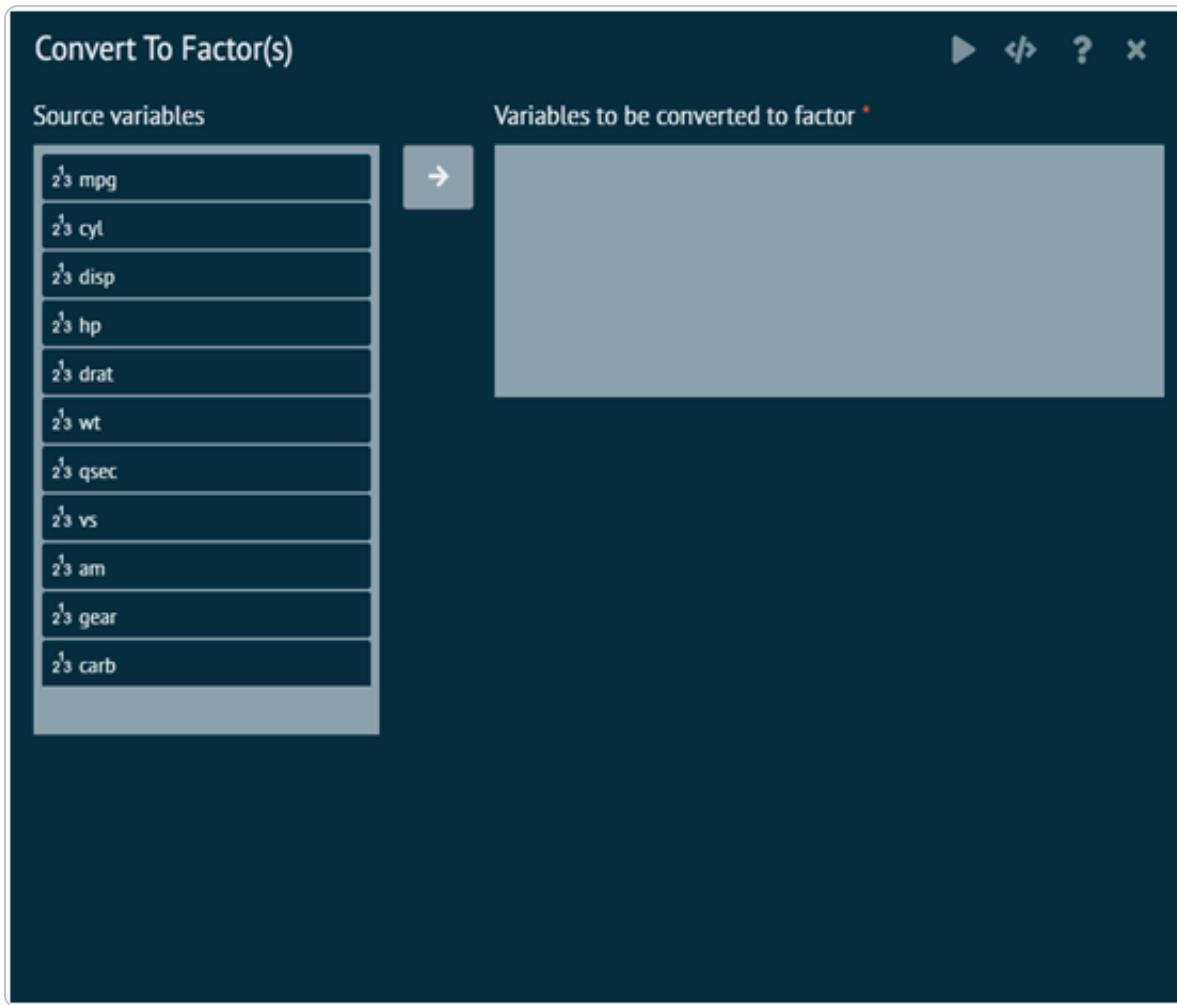
Converts a character to a date (POSIXct class). User needs to specify the format of the date stored in a character string.



Character to Date

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.



Convert to Factor

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

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

Variables to be converted to ordered factor/ordinal *



Convert to ordered factor/ordinal

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.

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

Date Order Check

Source variables

1's mpg
1's cyl
1's disp
1's hp
1's drat
1's wt
1's qsec
1's vs
1's am
1's gear
1's carb

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

Comparison

Note: This is the comparison between all dates specified above that will be checked for order errors, e.g. values should be date1 < date2 < date3.

<

<=

Row Identification Variables (optional)

Create dataset with date error variable

Date order check

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)

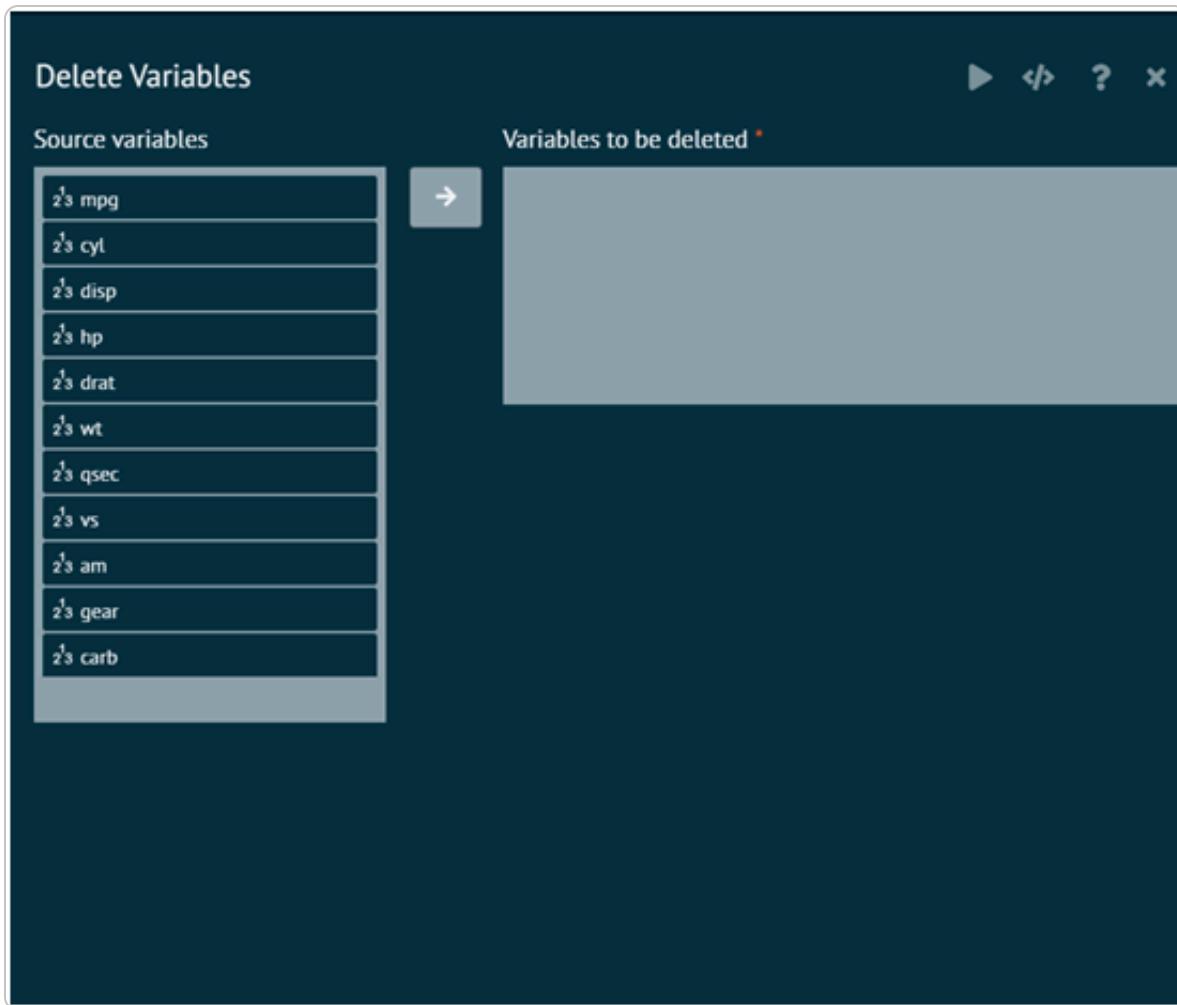
Specify one or more variables that may be useful to identify the rows. For example, subject identification number. These will be included in the list of errors. If no variables are specified, the row number of the dataset will be the only identifier.

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



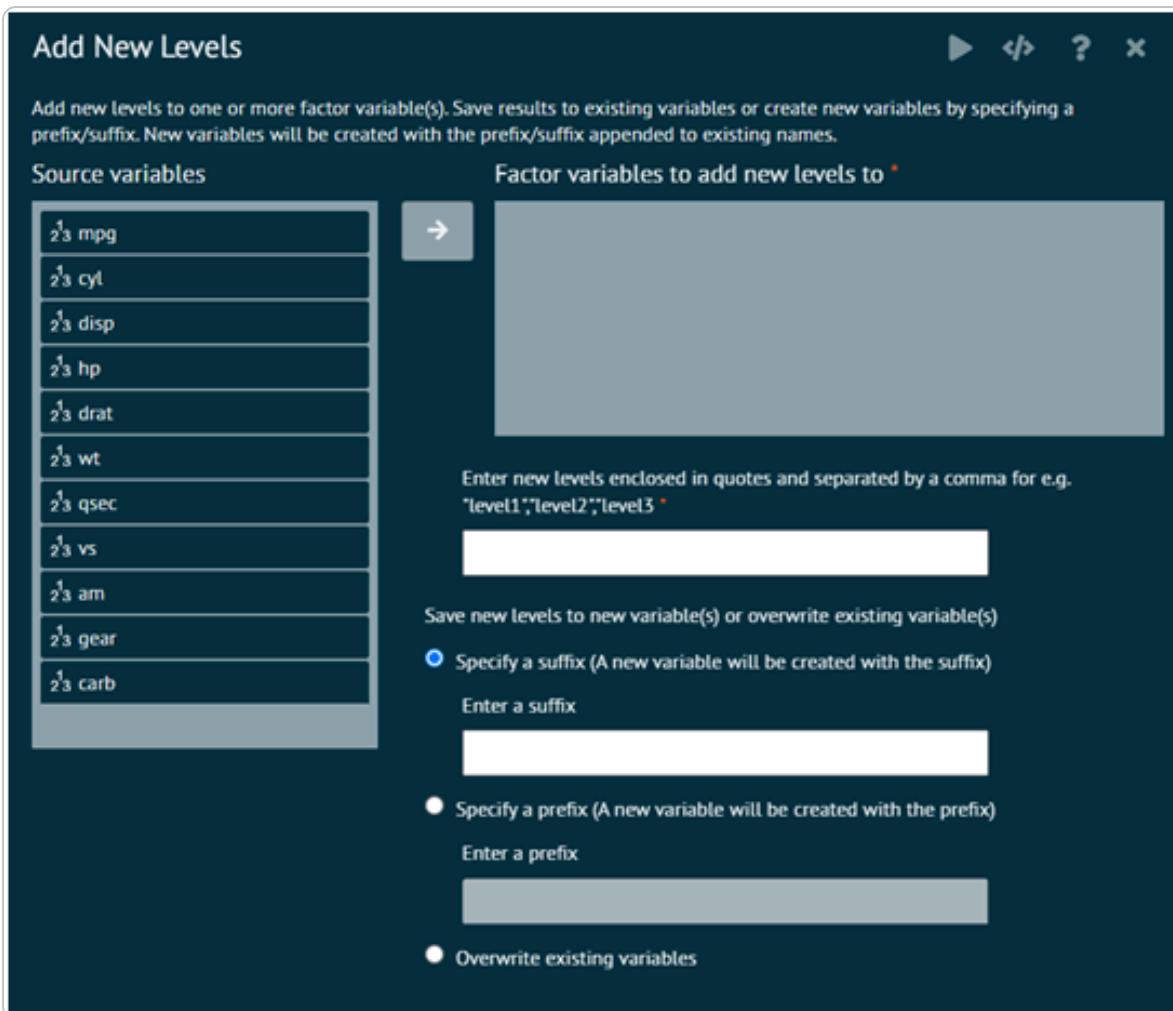
Delete Variable

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.



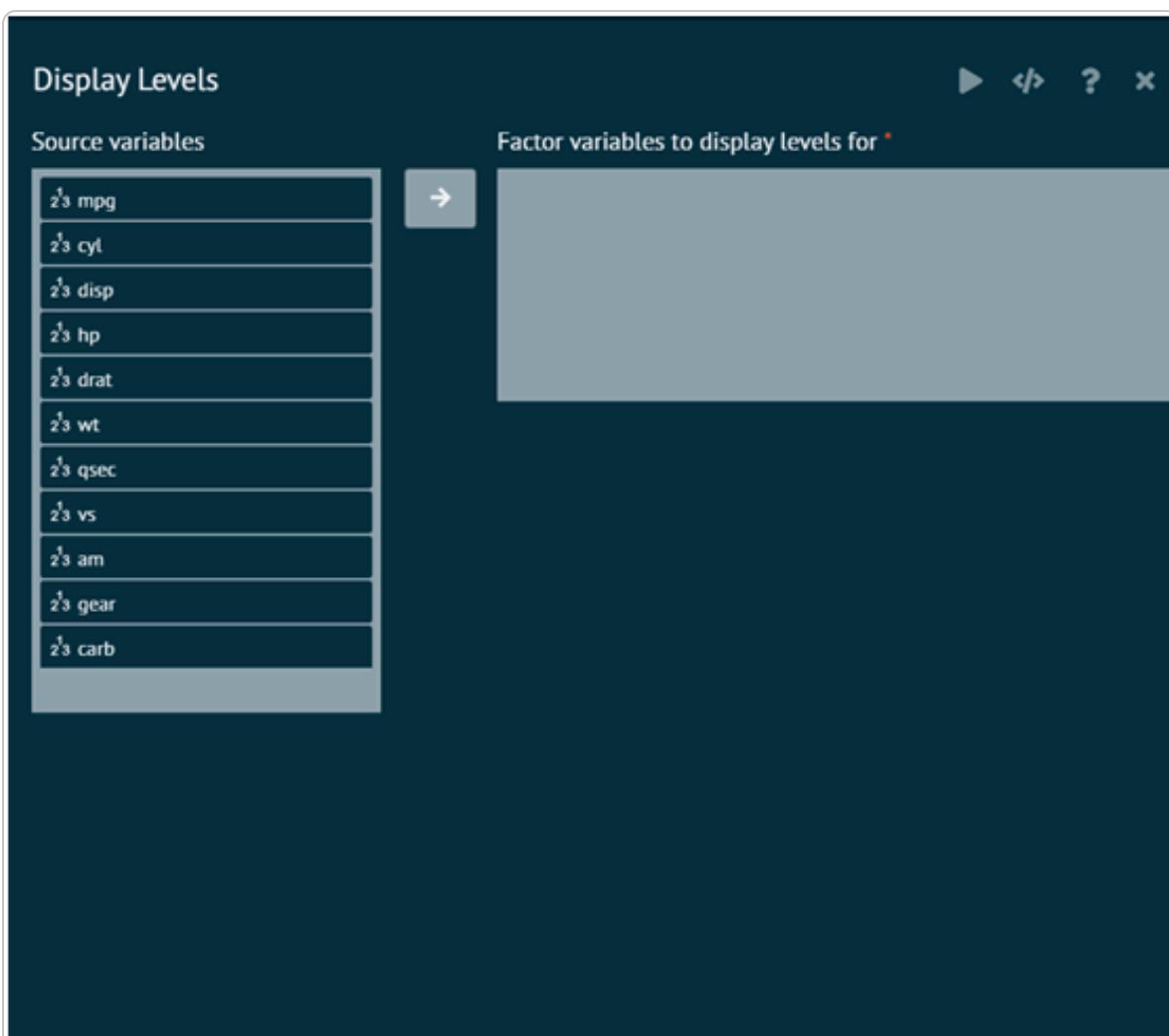
DON'T ENCLOSE LEVELS IN DOUBLE QUOTES OR SINGLE QUOTES.

i THERE CANNOT BE SPACES IN THE LEVEL NAMES.

i 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.



Display levels

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)

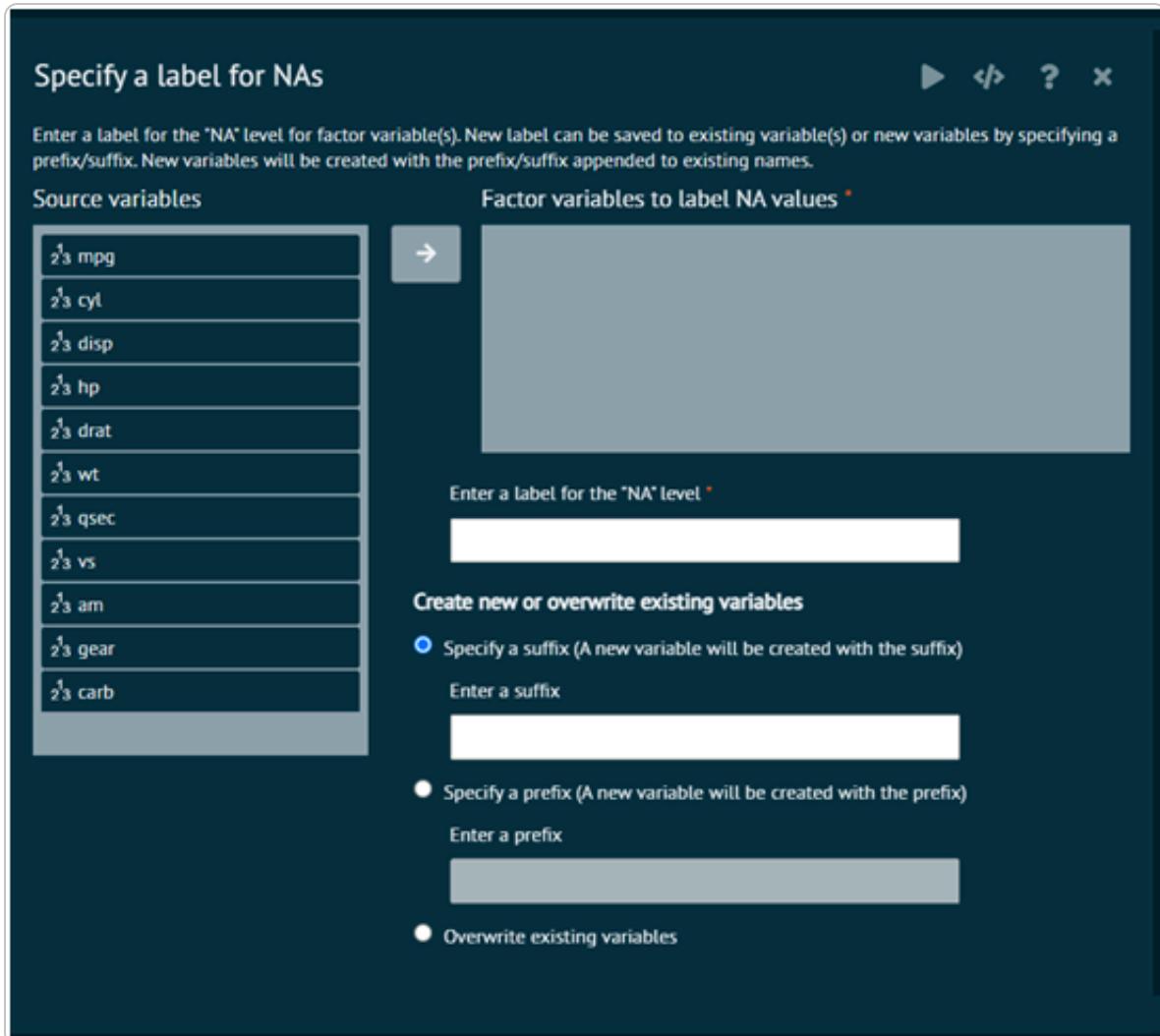
Enter a prefix

Drop used levels

The dialog box is titled "Drop Unused Levels". It contains a section for "Source variables" listing 13 variables: mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, carb. To the right is a section for "Factor variables to drop levels for *". Below this is a "Method to use" section with two radio buttons: "Drop all unused levels" (selected) and "Specify levels to drop". A note says "Enter levels to drop separated by comma, for e.g. level1,level2,level3 NOTE: Don't use spaces as separators between the levels". At the bottom, there are two options for "Save new levels to new variable(s) or overwrite existing variable(s)": "Specify a suffix (A new variable will be created with the suffix)" (selected) and "Specify a prefix (A new variable will be created with the prefix)". Each option has an input field for "Enter a suffix" and "Enter a prefix".

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.



Specify a label for NAs

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

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 lump sparse levels for *

Name for the lumped level *

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

Enter the number of categories

Enter the proportion

Variable weights

Lump the least or most common factor levels

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 *
23 mpg	
23 cyl	
23 disp	
23 hp	
23 drat	
23 wt	
23 qsec	
23 vs	
23 am	
23 gear	
23 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)

Specify levels to keep or replace by other

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

Reorder Factor levels by Count

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

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

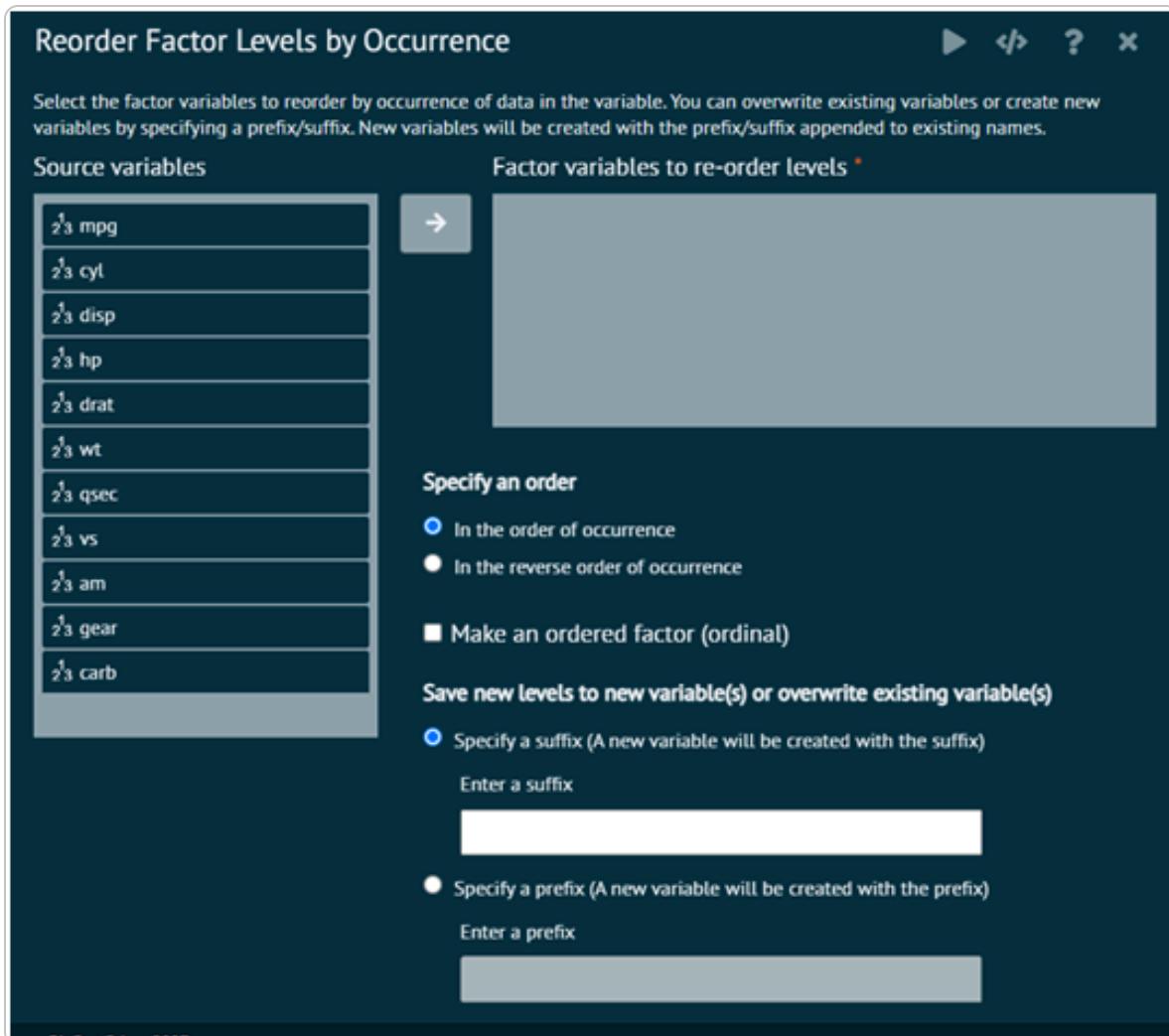
Specify a suffix (A new variable will be created with the suffix)

Enter a suffix

Reorder Factor levels by another variable

Reorder by occurrence

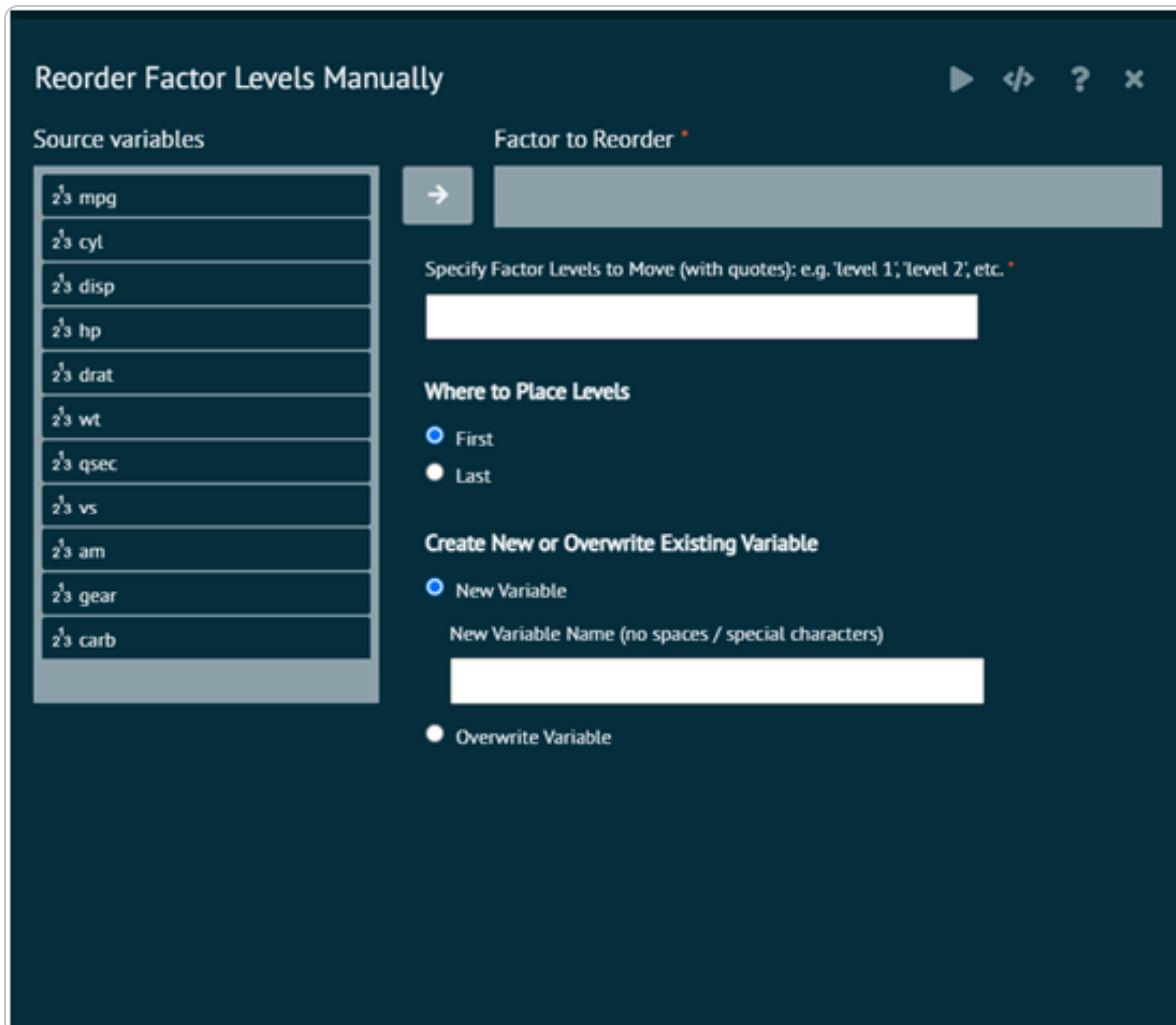
Reorder factors levels by first appearance (occurrence). See reorder by count for ordering by count/frequency.



Reorder by occurrence

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.



Reorder Manually

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".

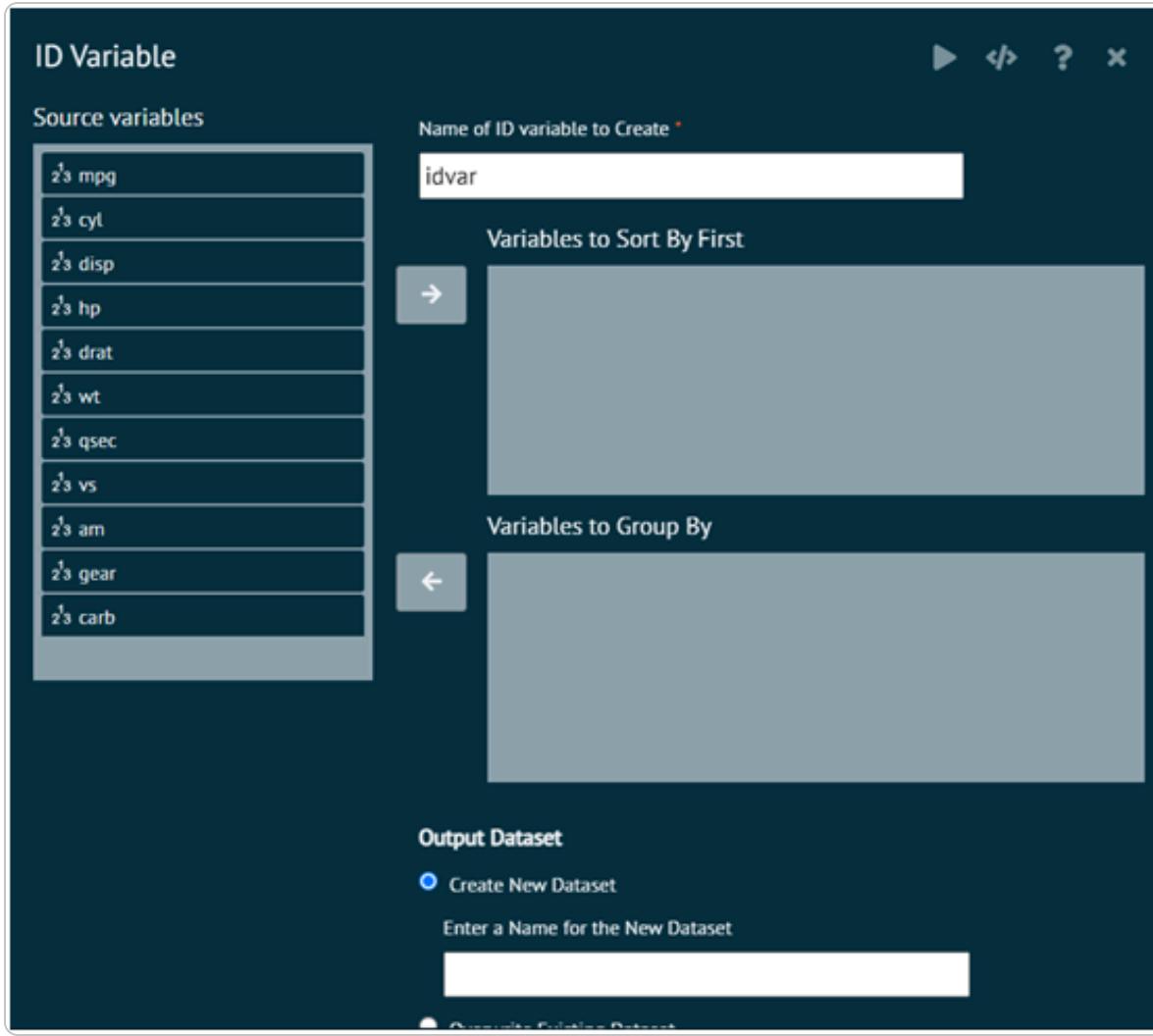
- ⓘ Required R Packages: dplyr,forcats

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.



ID Variable

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

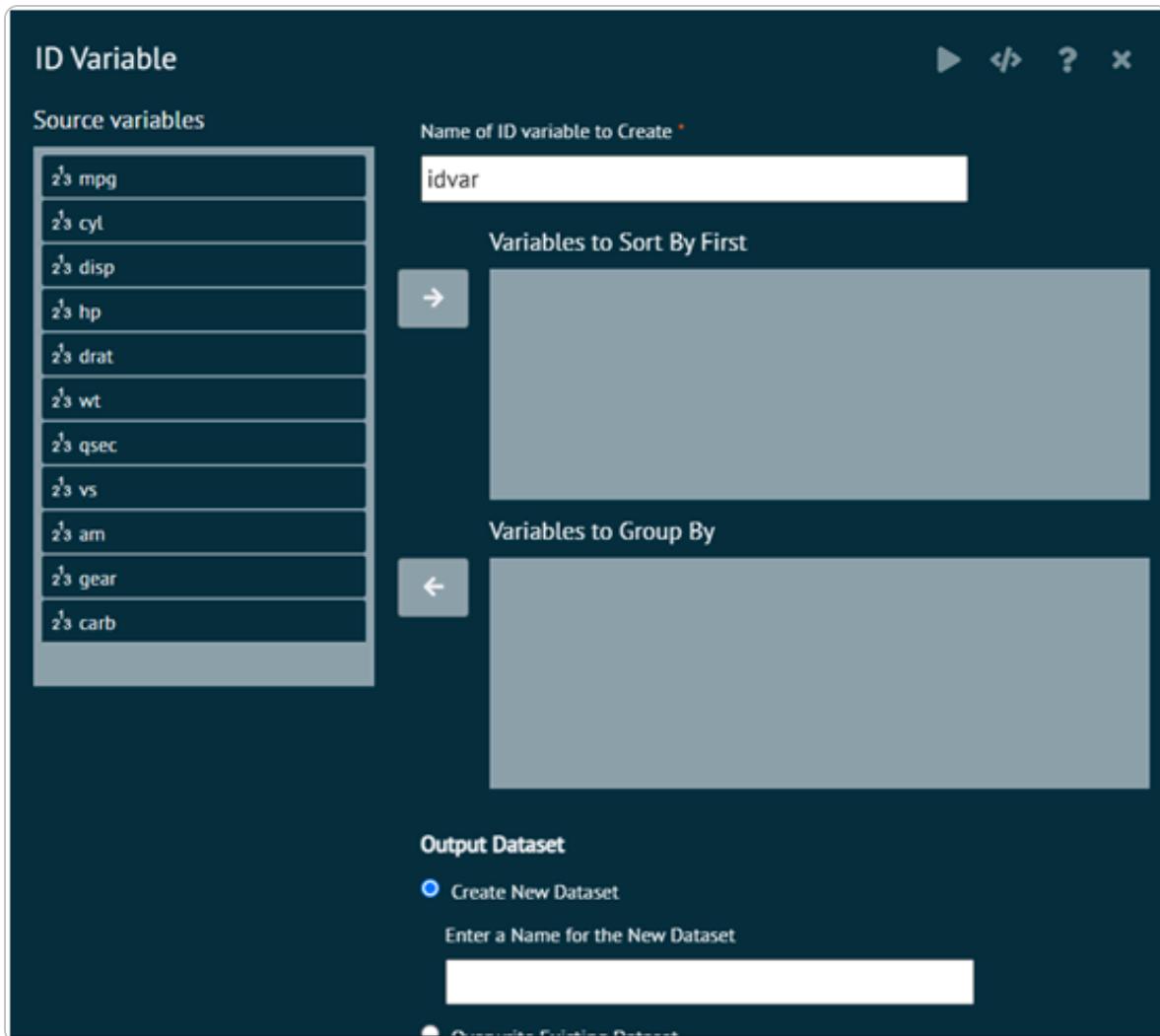
Variables to Group By (optional)

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

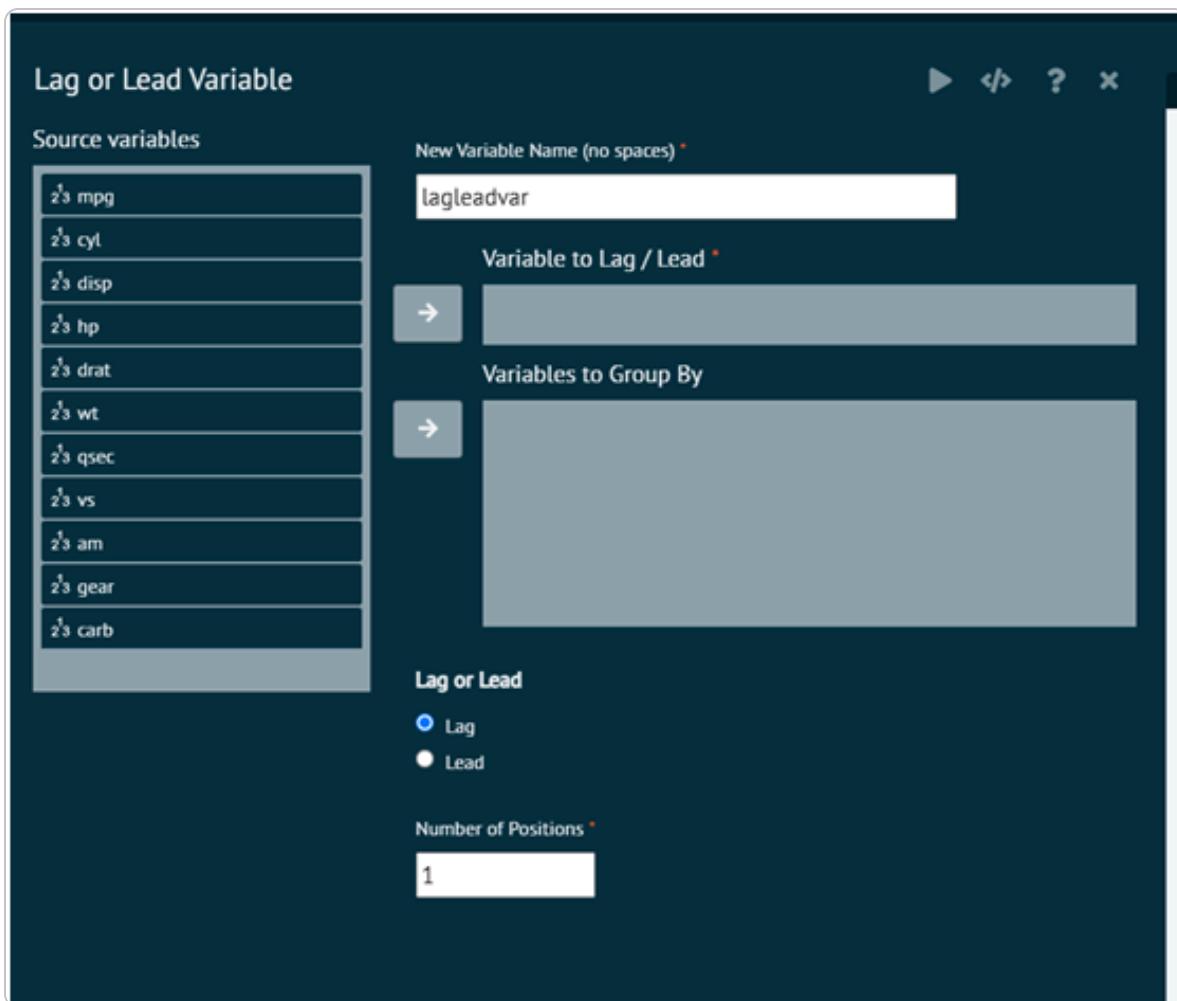
- i** Required R Packages: tidyverse



ID Variable

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.



Lag or Lead Variable

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

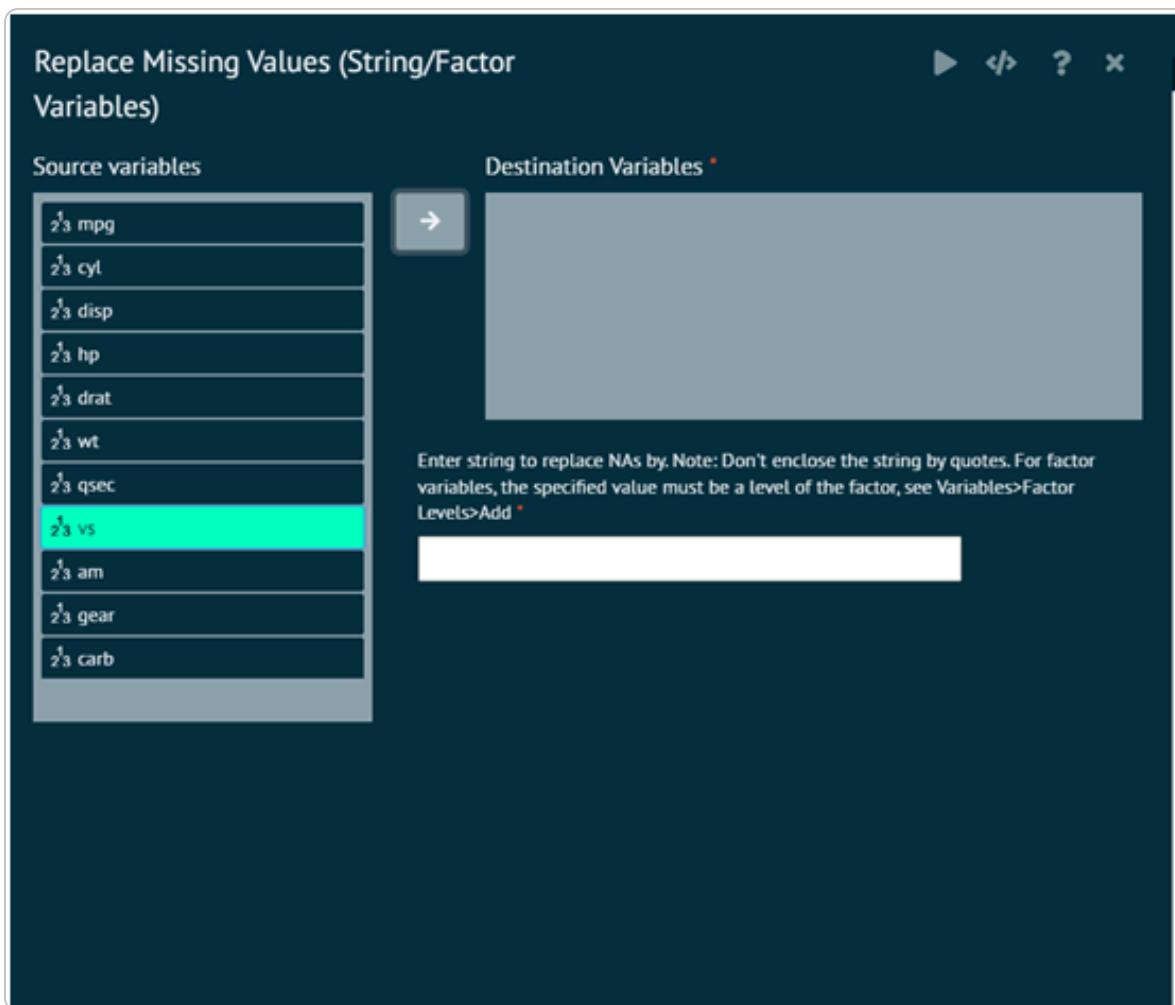
Specify the number of positions to lag or lead by. For example, a lagged value of 1 would extract the previous value and a lagged value of 2 would extract the value 2 positions previous.

- ⓘ 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



Character/Factor

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

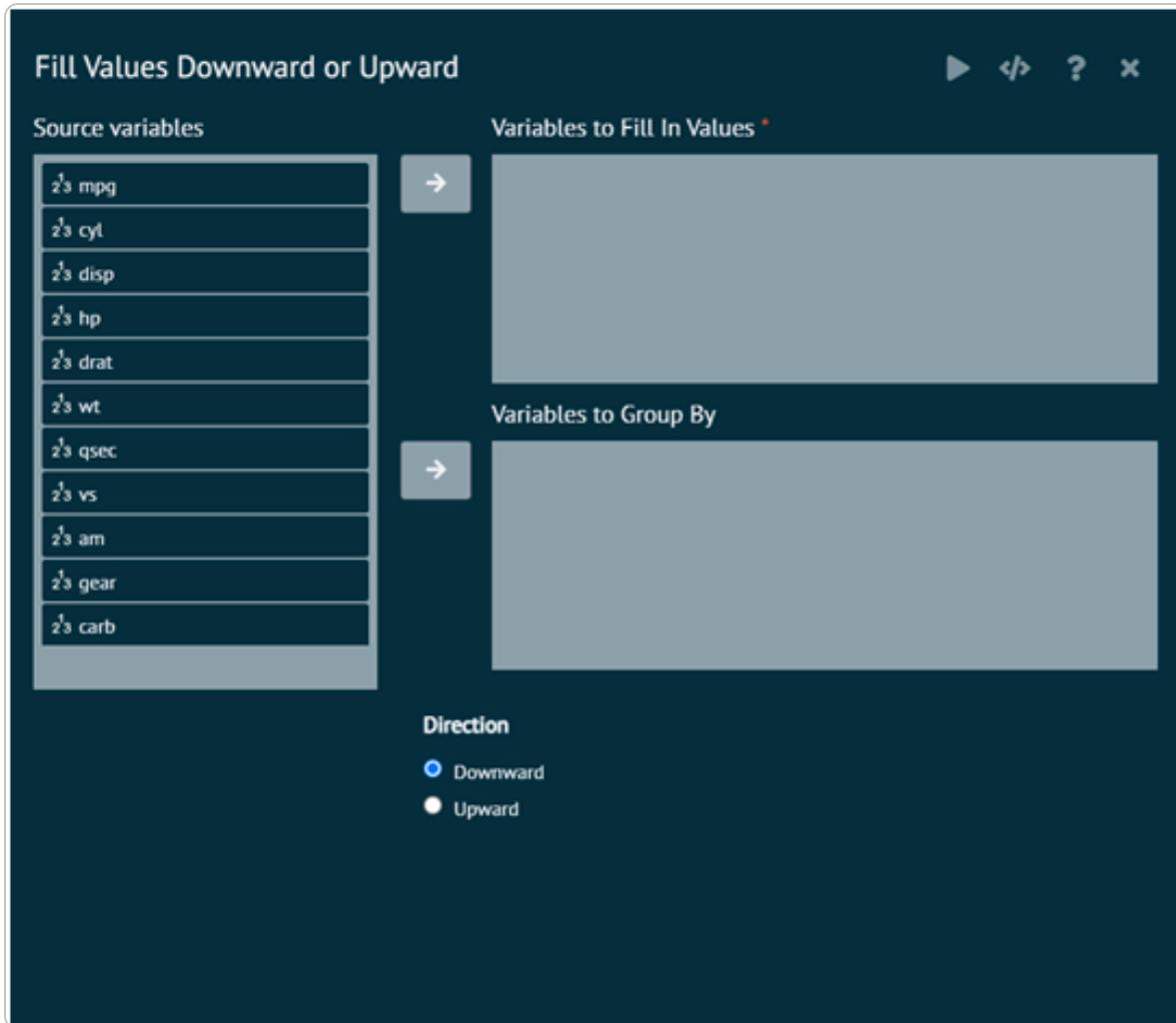
Specify variables that group rows together. Missing values will be filled in within groups defined by these variables. For example, grouping by a subject identifier would fill in values within subjects.

Direction

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



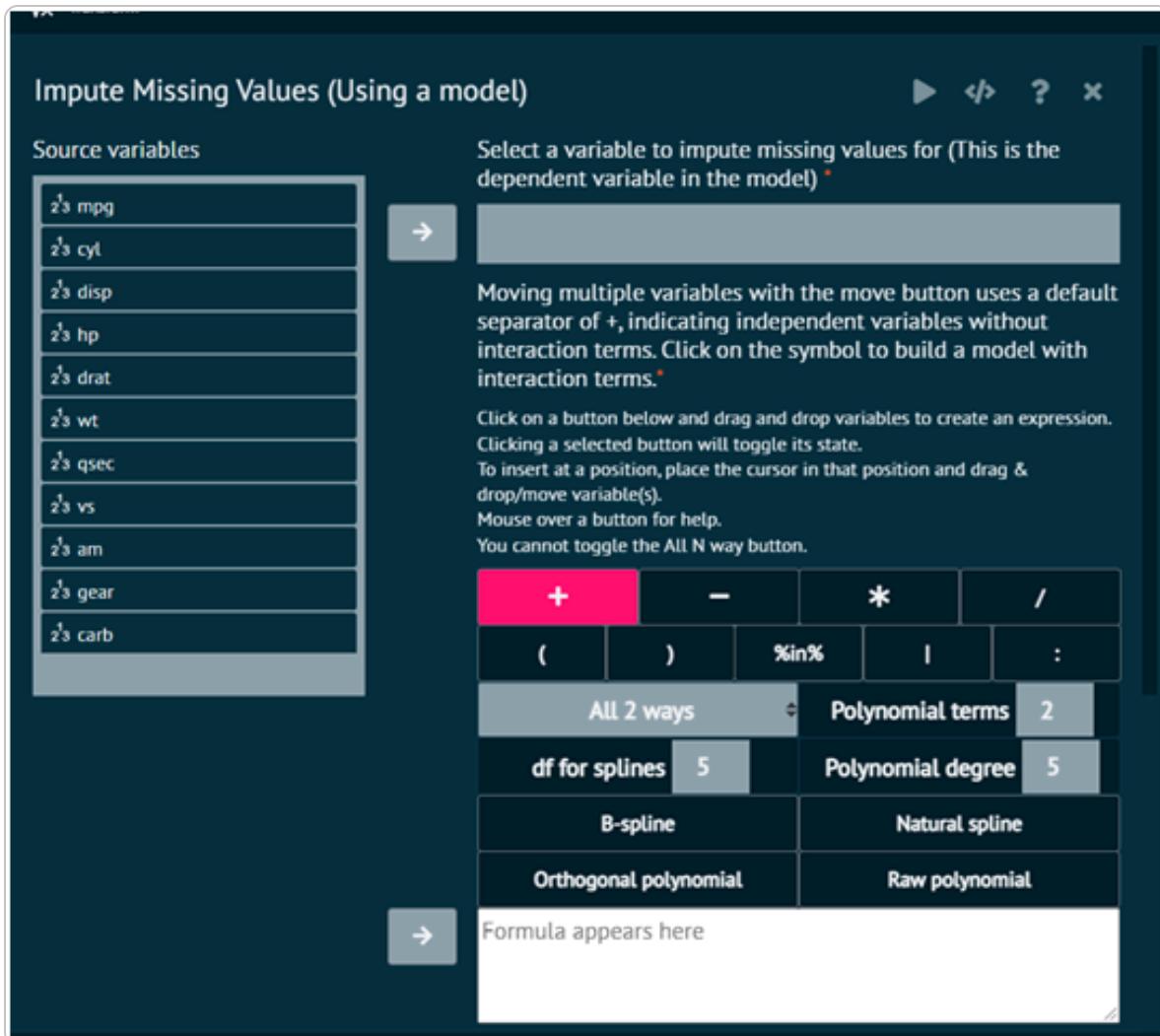
R Packages Required: tidyverse



Fill Values Downward or Upward

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.



Model Imputation

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



- 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

The screenshot shows the 'Replace Missing values (Numeric variables)' dialog box. At the top, it says 'Replace Misssing values (Numeric variables)' and 'Missing values (NAs) in the variables selected are replaced by applying a function i.e. median, mean, min, max or the value specified.' Below this, there are two main sections: 'Source variables' on the left and 'Select Variables to Replace Missing Values for *' on the right. The 'Source variables' list contains the following items: mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, carb. An arrow button points from the source list to the selection list. The 'Select Variables to Replace Missing Values for *' section is currently empty. Below these, there is a section titled 'Select a function or specify a value to replace NAs'. A radio button is selected for 'Use a function to compute missing values'. Underneath, it says 'Select the function' and lists five options: mean (highlighted in green), median, min, max, and getmode. Another radio button is selected for 'Specify a numeric value to replace missing values'. Below this, there is a text input field with the value '0'.

Numeric

⚠ Arguments

var

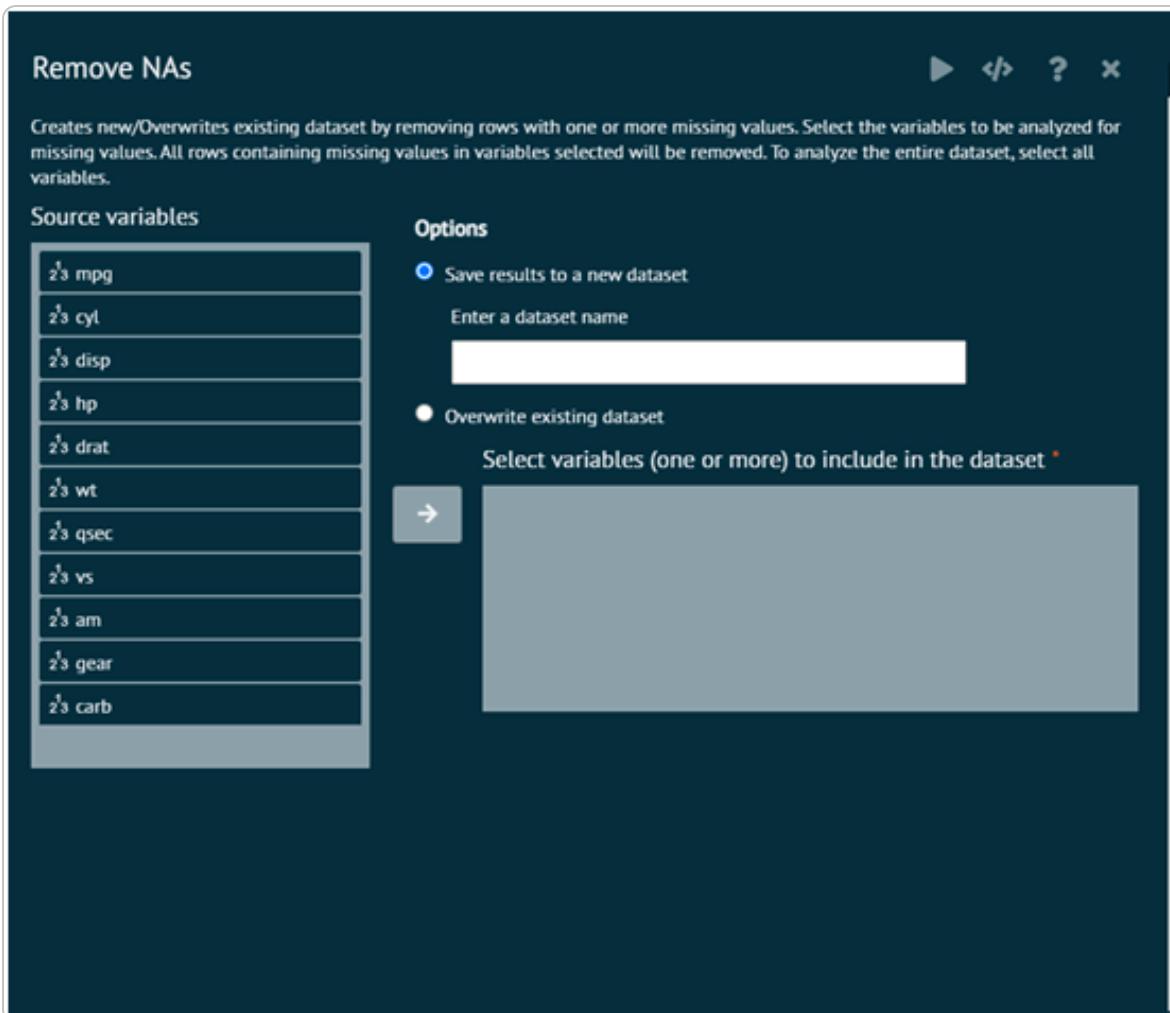
Character string representing the numeric variable with missing values (na), for e.g.
var = c('sales')

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



Remove NAs

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.

The screenshot shows a software interface titled "Impute Missing Values (Using a formula)". On the left, a list of "Source variables" is shown, including mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, and carb. A large arrow button points from these variables to the right side of the screen. The right side contains a text input field with placeholder text: "Select a variable to impute missing values for (This is the dependent variable in the model) *". Below this, there is descriptive text about moving variables and building models with interaction terms. Further down, instructions for creating expressions using operators (+, -, *, /, (,), %in%, |, :) and buttons for "All 2 ways", "df for splines", "Polynomial terms", "Polynomial degree", "B-spline", "Natural spline", "Orthogonal polynomial", and "Raw polynomial" are provided. At the bottom, a text area says "Formula appears here".

Impute Missing Values using a formula

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.

The screenshot shows the 'Rank Variable(s)' dialog box. At the top, it says 'Enter a suffix or prefix for the new ranked variables'. There are two radio buttons: 'Suffix' (selected) and 'Prefix'. Below this is a text input field labeled 'Enter a suffix/prefix' with a placeholder 'ranks'. On the left, under 'Source variables', there is a list of variables from the 'mtcars' dataset: mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, carb. To the right of this list are two greyed-out sections: 'Select the variable(s) to rank' and 'Optionaly select variable(s) to rank values within'. At the bottom, it says 'Specify a ranking function' and 'Select a ranking function, click on help for additional information'. The footer of the dialog box reads 'BioStat Prime 2023'.

Rank Variable(s)

⚠ 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()

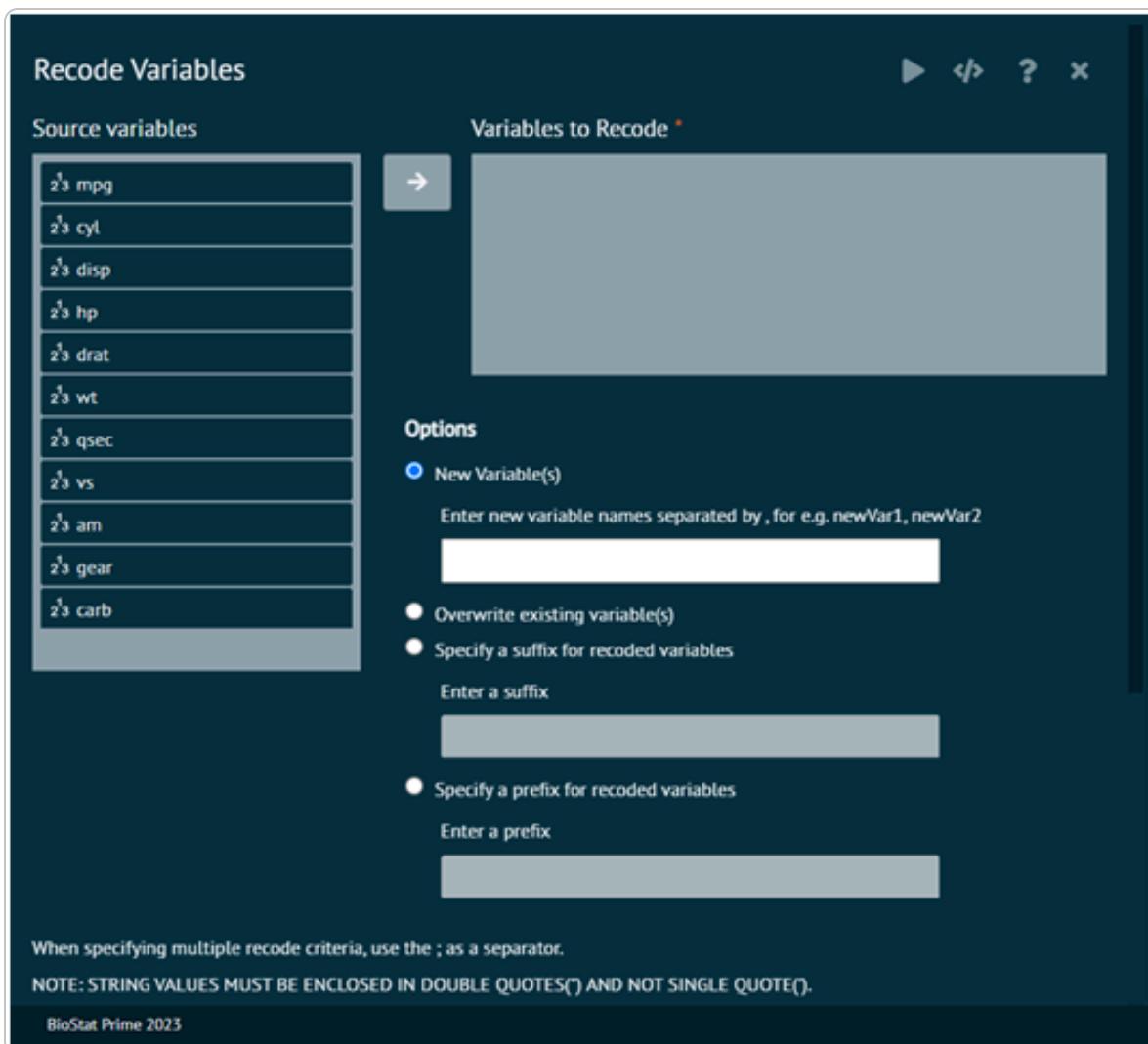
a cumulative distribution function. Proportion of all values less than or equal to the current rank.

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.



Recode Variables

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.

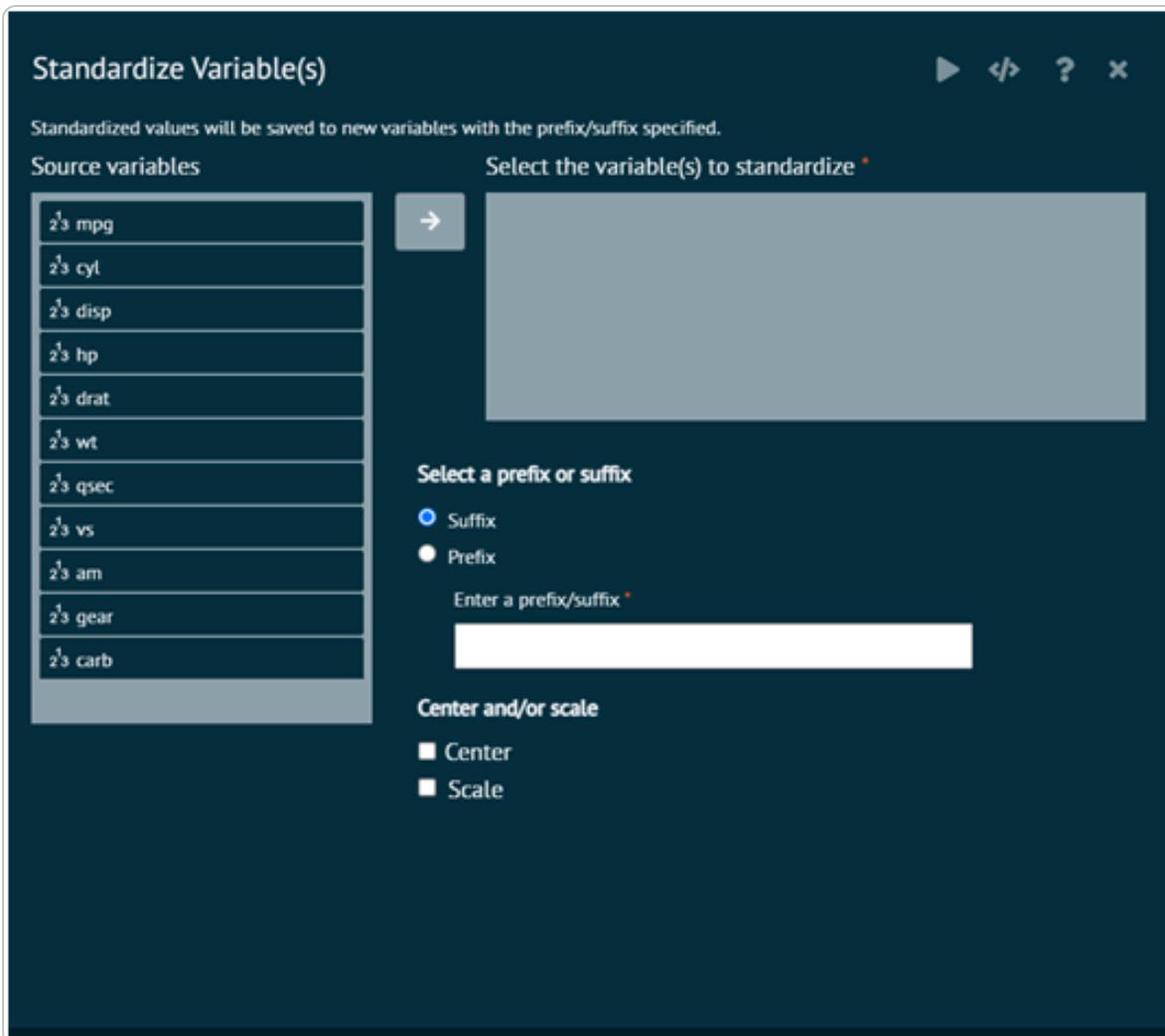
dataSetNameOrIndex

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 provide to center and/or scale.



Standardize Variable(s)

⚠ 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

- 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 transform *

Select an operation to apply *

- log10
- log
- log2
- abs
- ceiling

Create new or overwrite existing variables

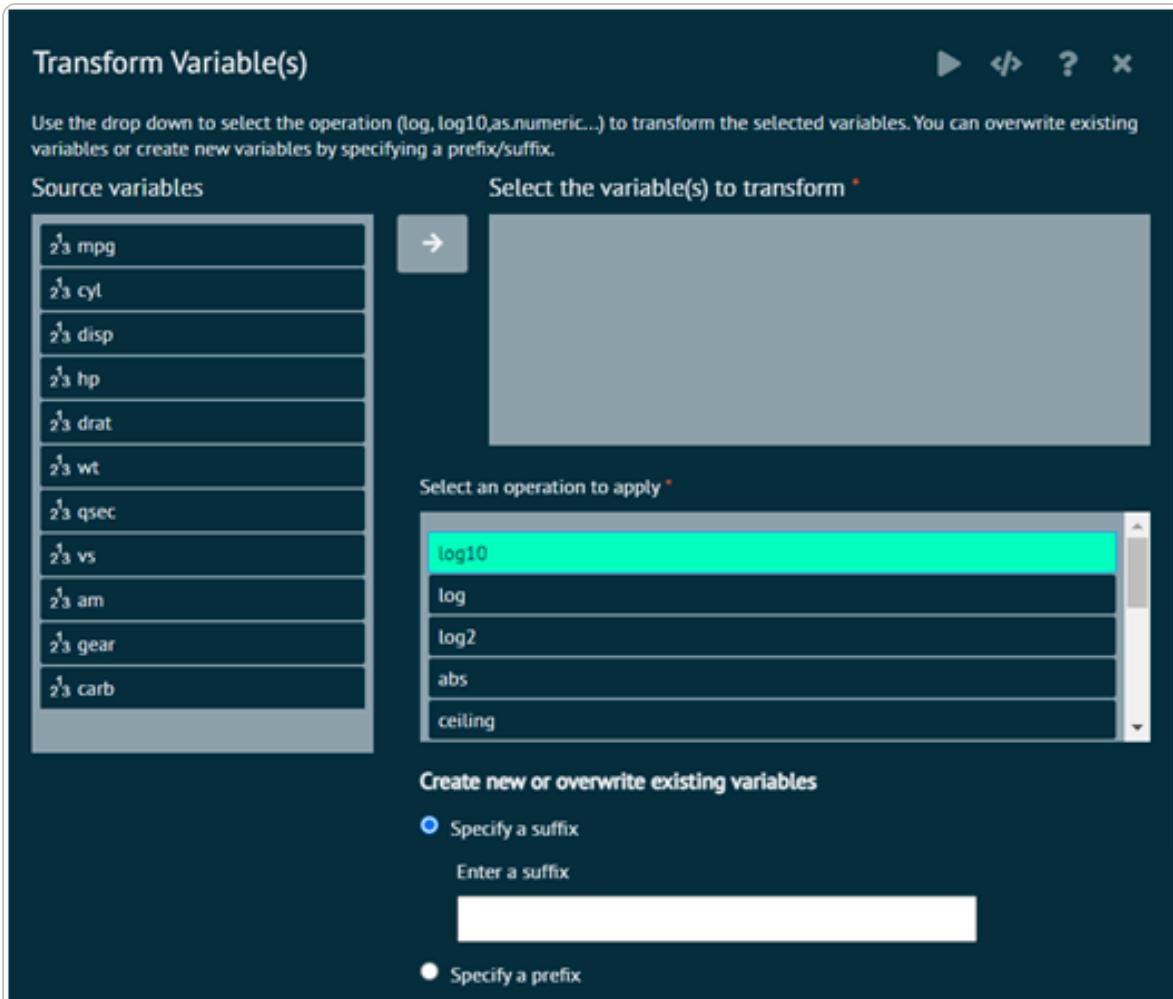
Specify a suffix

Enter a suffix

Specify a prefix

Enter a prefix

Transform Variable(s)



Arguments

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

Data Analysis

Functions under 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.

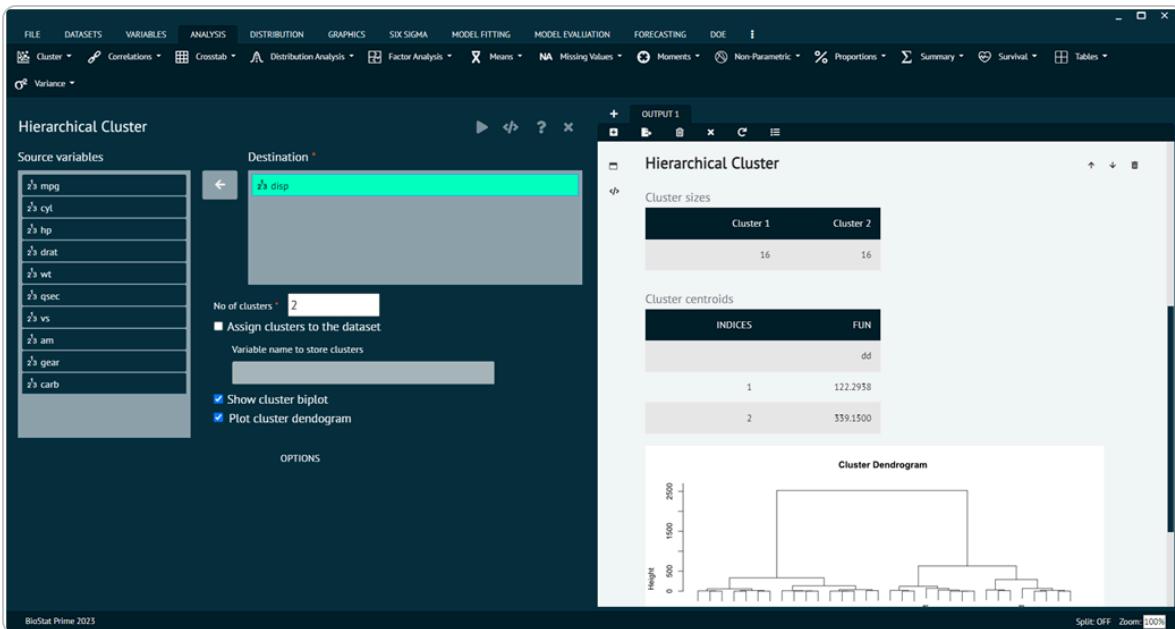
To analyse it in BioStat Prime user must follow the steps as given.

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.



Hierarchical Clustering

⚠ 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.

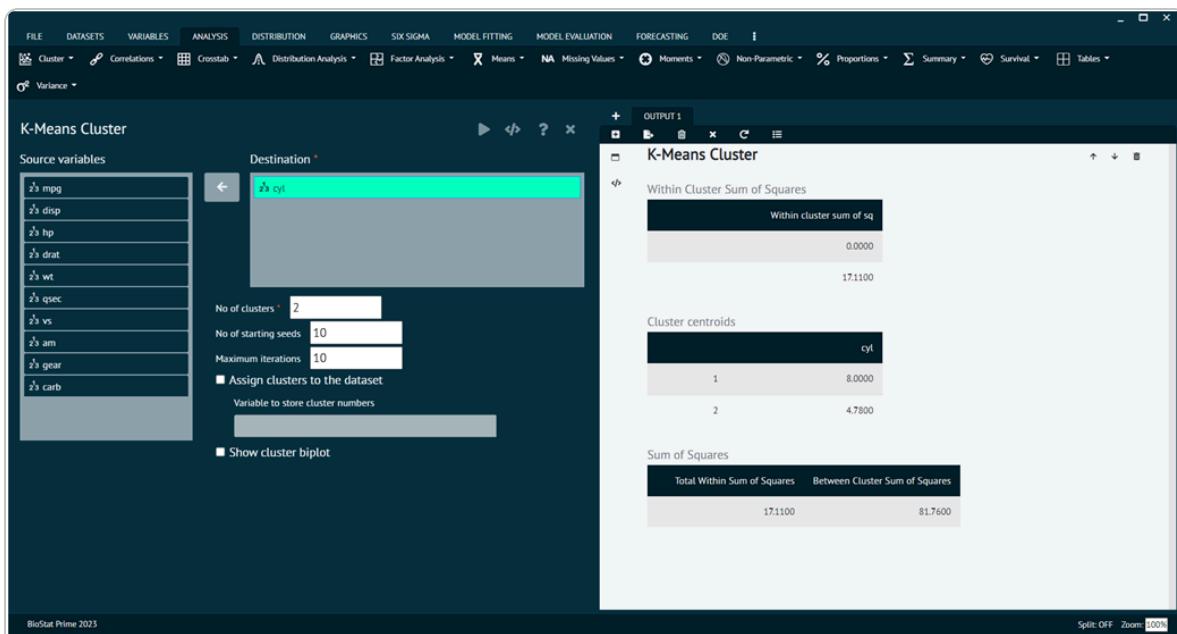
- ⚠** Partition clustering divides the data into non-overlapping clusters in a single step.

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.



K-Means Clustering

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

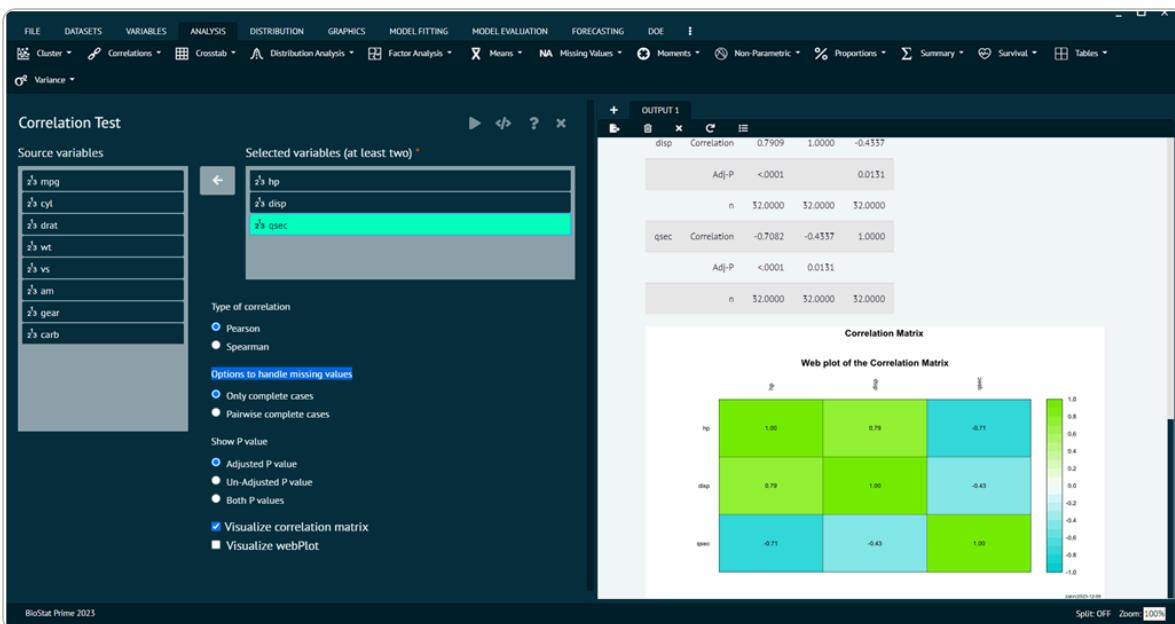
The Pearson correlation test and the Spearman correlation test are statistical methods used to assess the strength and direction of the relationship between two variables. However, they differ in terms of the types of relationships they can detect and the assumptions they make about the data.

⚠ 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.



Pearson, SpearmanCorrelation

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

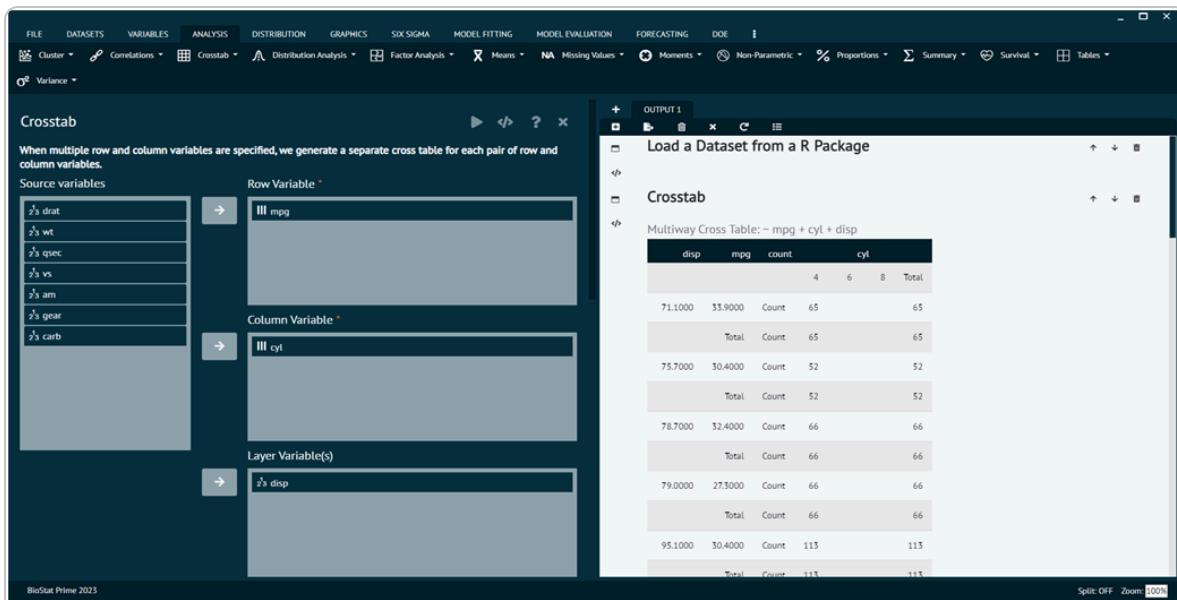
1. Expected counts
2. Row and column percentages
3. Unstandardized, standardized and adjusted residuals
4. Chisq with odds ratio, McNemar and Fisher statistics

⚠ 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.



Crosstab

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.

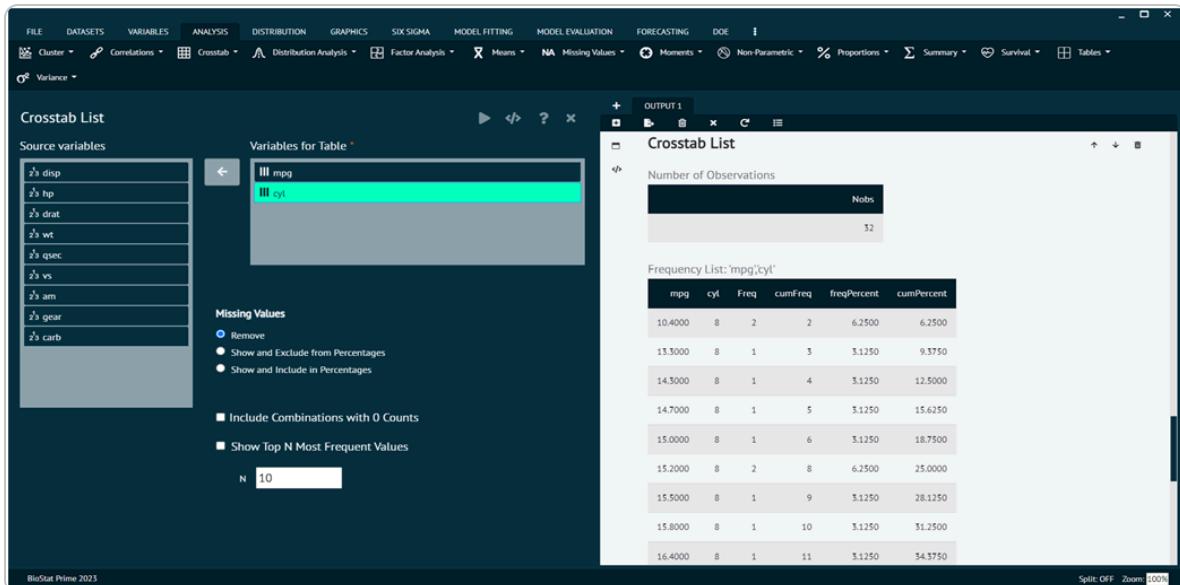
- Care should be taken about how many variables to cross-classify and how many possibilities can result, as some tables may take longer to produce.

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.



Crosstab List

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.

- 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.

! 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.

The screenshot shows the BioStat Prime software interface. The main window is titled "Odds Ratios / Relative Risks, M by 2 Table". In the "Source variables" section, several variables are listed: disp, hp, drat, wt, qsec, vs, am, gear, and carb. Two variables are selected: "mpg" as the "Outcome Variable (Binary Factor)" and "cyl" as the "Exposure/Predictor Variable (Factor)". The "Statistic" section has "Odds Ratios" selected. The "Odds Ratio Estimation Type" section has "Wald" selected. The "Relative Risk Estimation Type" section has "Small Sample Adjusted" selected. The "Category Reversal" section has "No Reversal" selected. The "Output" window displays two tables: "Frequencies of Predictor = cyl vs Outcome = mpg" and "Odds Ratios with Confidence Intervals (0.95 level) and Wald Chi-Square".

Odds Ratio Relative Risks, M by 2 Table

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 interface. On the left, the 'Odds Ratios / Relative Risks, M by 2 Table' dialog is open. It displays a list of source variables (disp, hp, drat, wt, qsec, vs, am, gear, carb) and allows selecting an outcome variable (mpg) and an exposure/predictor variable (cyl). The 'Statistic' section is set to 'Odds Ratios'. The 'Output' window on the right shows two tables: 'Frequencies of Predictor = cyl vs Outcome = mpg' and 'Odds Ratios with Confidence Intervals (0.95 level) and Wald Chi-Square'. The first table provides frequency counts for combinations of cyl and mpg. The second table lists odds ratios, confidence intervals, and p-values for each cylinder level (4, 6, 8).

Anderson-Darling Normality Test

Kolmogorov-Smirnov Normality Test

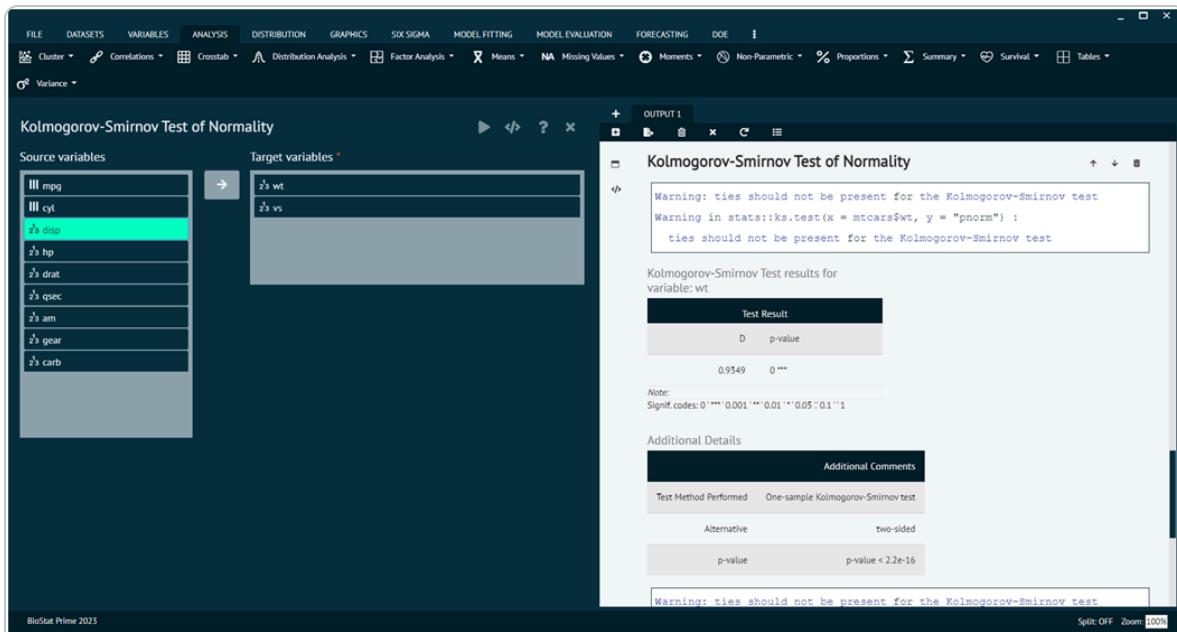
The Kolmogorov-Smirnov (K-S) test for normality is a non-parametric test used to determine whether a sample comes from a normal distribution. It is based on the **cumulative distribution function (CDF)** of the normal distribution and involves comparing the observed cumulative distribution of the data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



Kolmogorov Smirnov Normality Test

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.

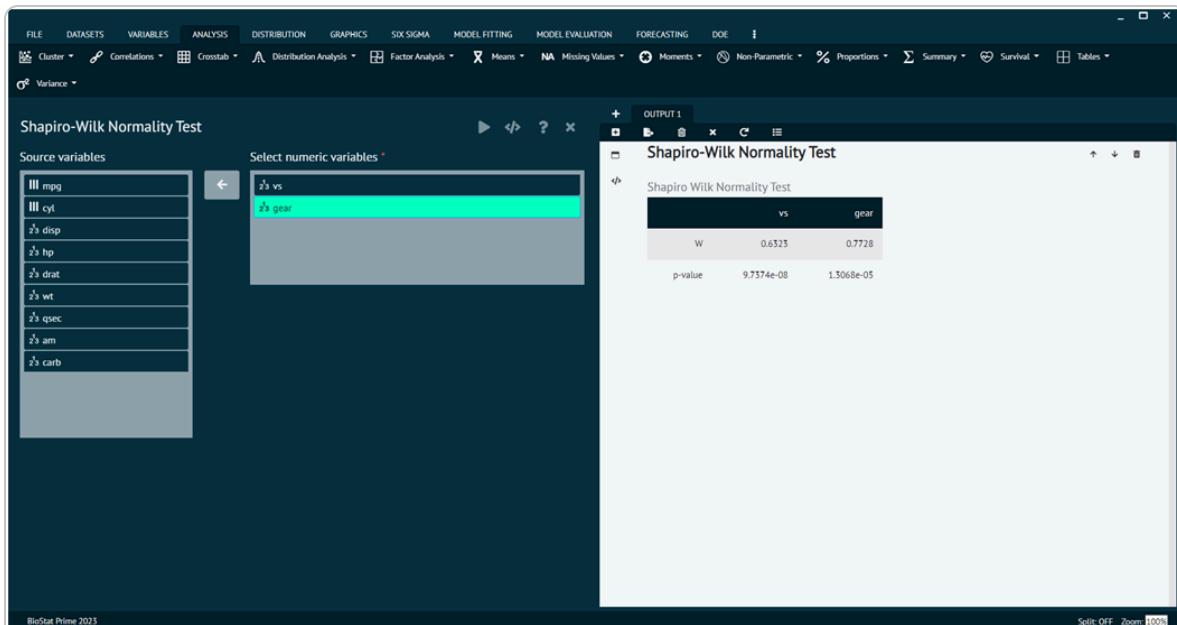
i The test is particularly useful when dealing with smaller sample sizes.

⚠ 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.



Shapiro-Wilk Normality Test

Distribution Fit

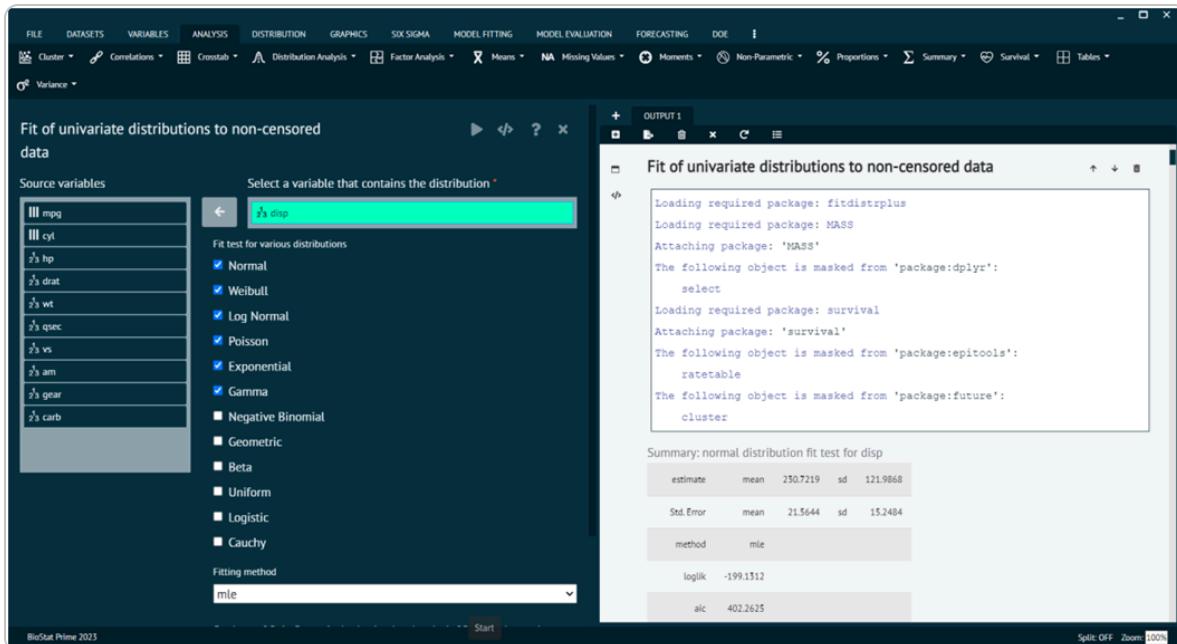
Distribution fitting is a statistical technique used to model and describe the distribution of a dataset by finding the probability distribution that best fits the observed data.

⚠ 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.



Distribution Fit

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 mgdist 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 mgdist for more details). This method is not suitable for discrete distributions.

mse

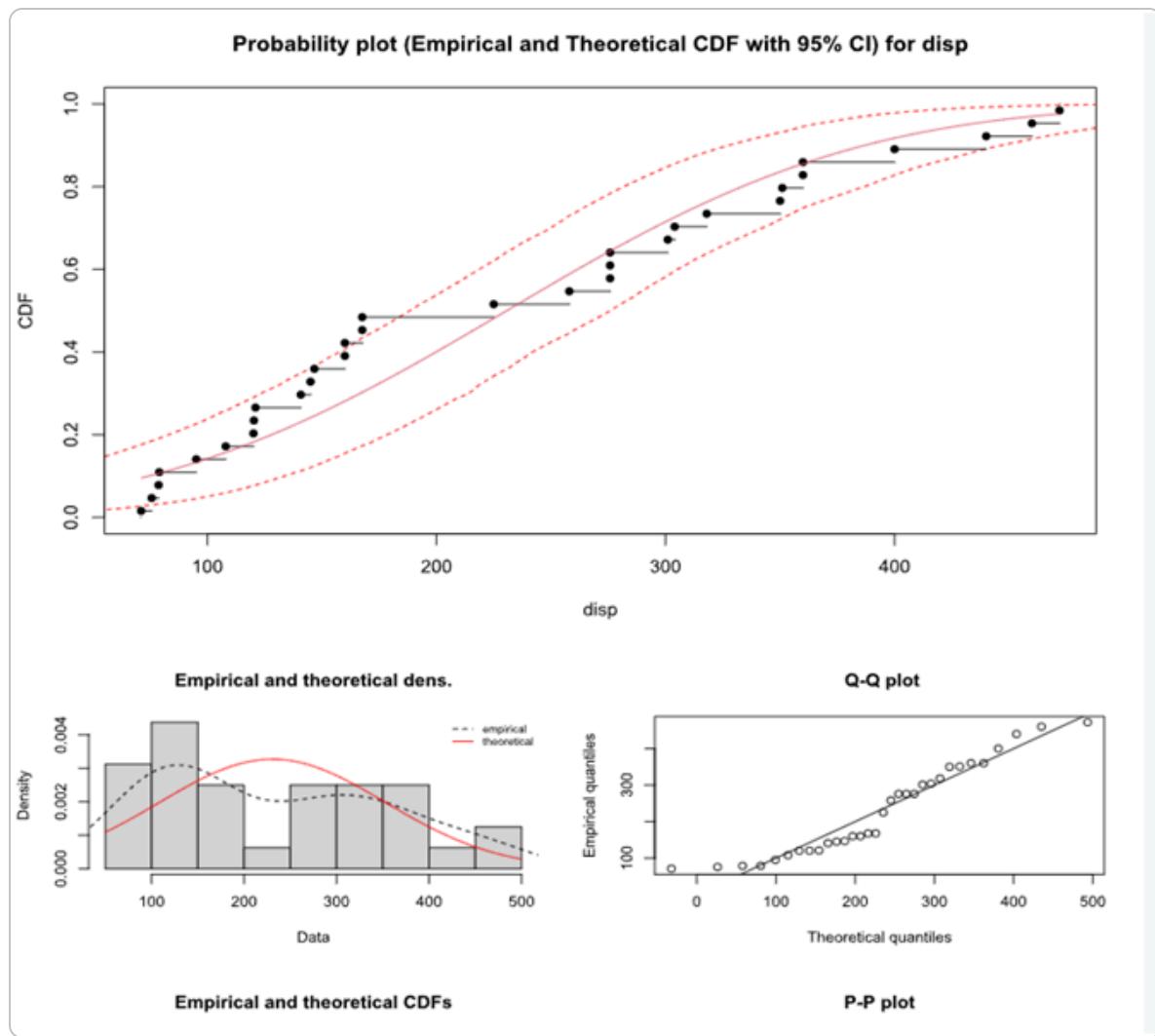
When method = "mse" Maximum goodness-of-fit estimation consists in maximizing the average log spacing. A numerical optimization is carried out in msedist via optim.

⚠ 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 Plot

Distribution Fit with Gamlss

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

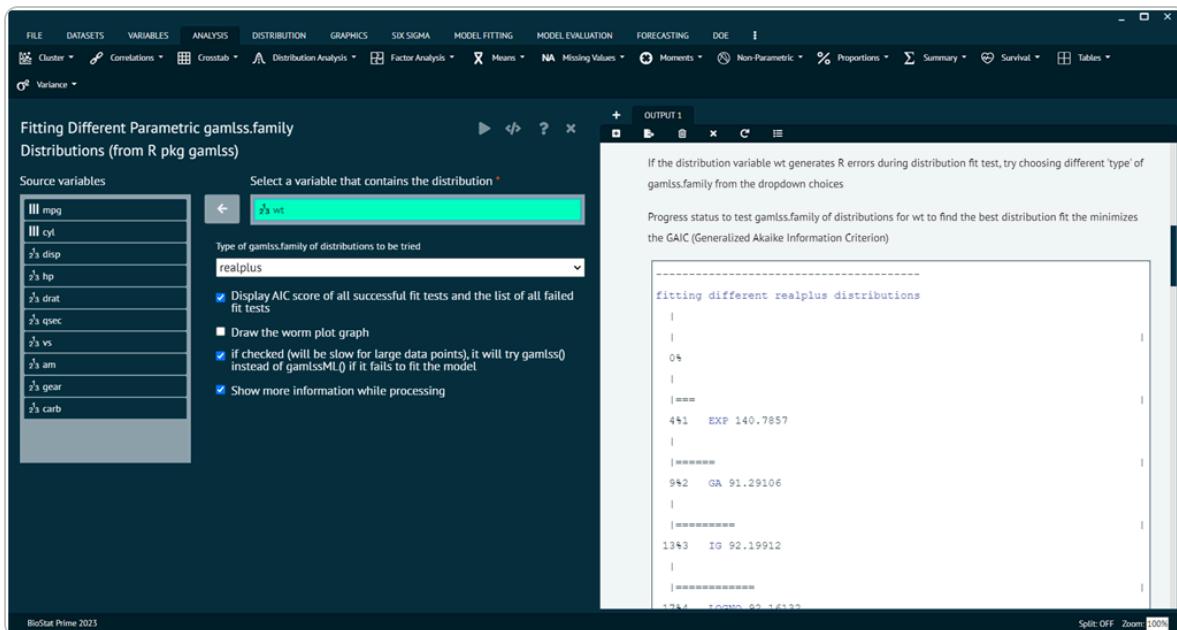
GAMLSS is a flexible framework for modeling distributions and is capable of handling a wide range of distributional shapes.

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.



Distribution Fit with Gamlss

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", "GEMO", "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 gamlss.family distributions for binomial type data :"BI", "BB", "DB", "ZIBI", "ZIBB", "ZABI", "ZABB"

Distribution Analysis Cullen and Frey Graph

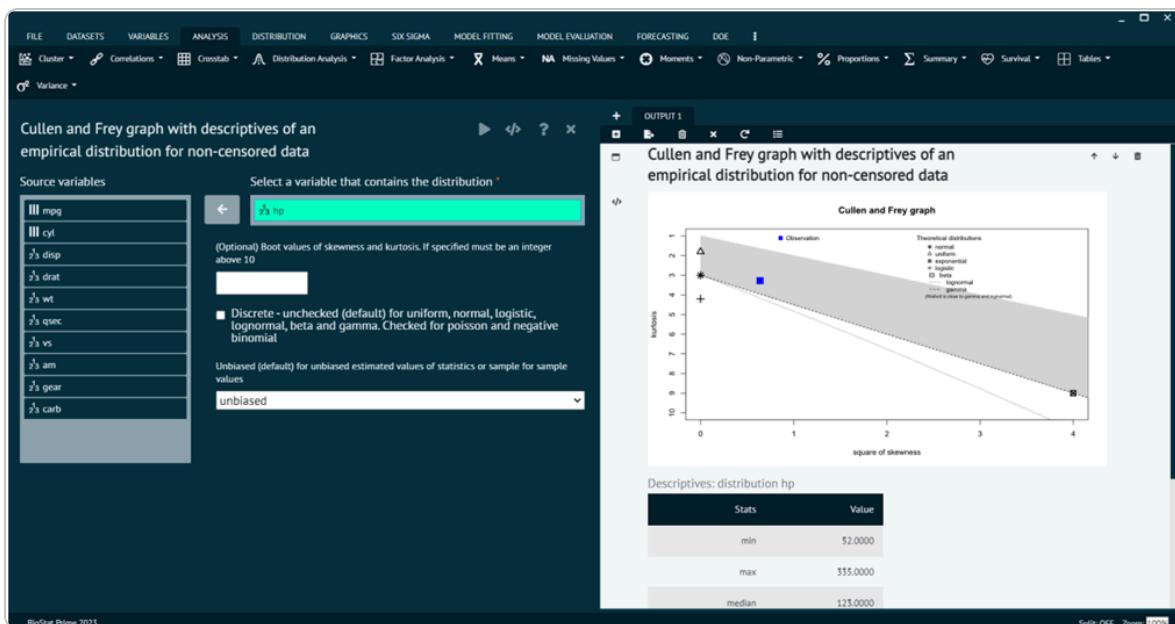
The Cullen and Frey graph, also known as the Cullen and Frey graph for skewness and kurtosis, is a graphical method for assessing the skewness and kurtosis of a dataset. It's a visual tool that helps you quickly inspect the departure from normality in terms of skewness and kurtosis.

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.



Distribution Analysis Cullen and Frey Graph

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.

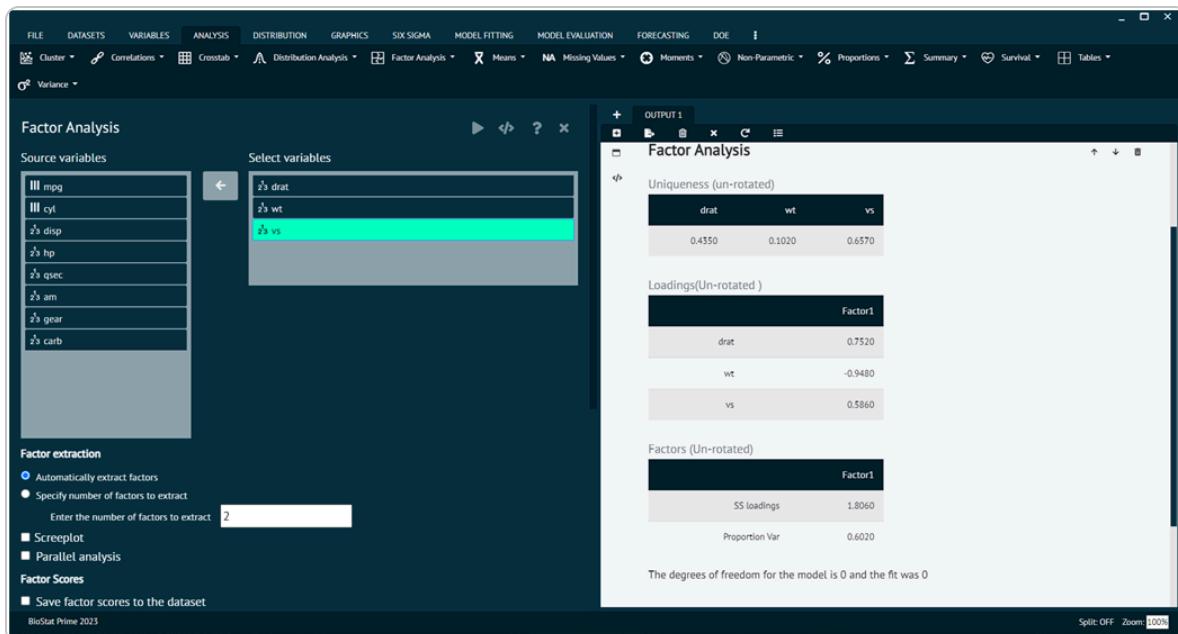
Factor Analysis

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.



Factor Analysis

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

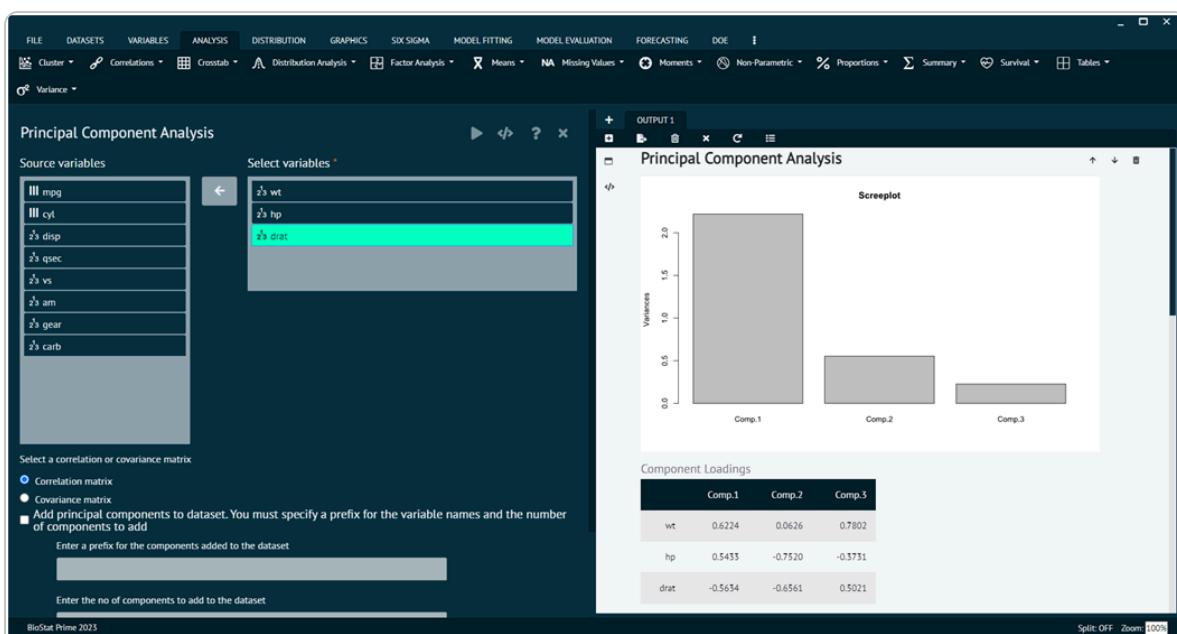
Performs a principal components analysis on the given numeric data matrix and returns the results as an object of class princomp.

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.



Principal Component Analysis

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.

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.

The screenshot shows the BioStat Prime 2023 software interface. The main window is titled 'ANOVA, 1 and 2 way'. On the left, under 'Source variables', several car model variables are listed: disp, hp, wt, qsec, vs, am, gear, carb. A 'Target variable (numeric/scale)' dropdown is set to 'drat'. Below it, a section for 'Specify a maximum of 2 factor variables' lists 'mpg' and 'cyl'. At the bottom of this panel, there's an 'OPTIONS' section with a checked checkbox for 'Ignore interaction terms in model' and a dropdown for 'Select type I/II/III Sums of squares' currently set to 'III'. To the right, an 'OUTPUT 1' window displays a table titled 'Summaries for drat by factor variable mpg'. The table includes columns for mpg, n, mean, median, min, max, sd, and variance. Data rows are shown for various values of mpg, such as 10.4000, 13.3000, 14.3000, etc. The software interface includes a top menu bar with various statistical analysis options like FILE, DATASETS, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION, FORECASTING, DOE, and a bottom status bar indicating 'BioStat Prime 2023'.

ANOVA 1 and 2 way

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

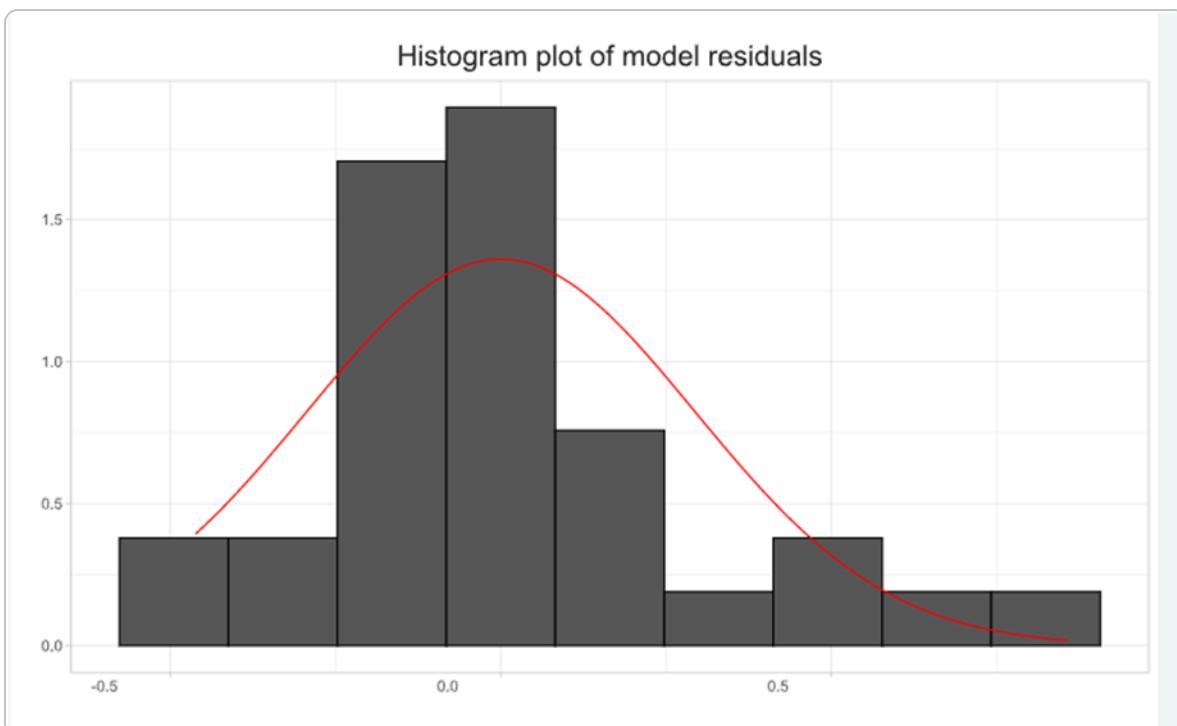
Load the dataset -> Click on the analysis tab in main menu -> Select means -> The means tab leads to the ANOVA, one-way with random blocks analysis technique in the dialog -> In the dialog select the variable and options according to the requirement -> Execute the dialog.

The screenshot shows the BioStat Prime 2023 interface. The main window is titled "ANOVA, one-way with random blocks". In the "Source variables" list, "mpg" is selected. The "Response Variable" is set to "wt". The "Fixed Effect" is "drat". The "Blocking Variable(s)" is "gear". Below the dialog are two options: "Histogram of residuals" and "Post-hoc analysis". The output window is titled "ANOVA, one-way with random blocks" and displays a table of "Summaries for fixed effect" with 17 rows of data.

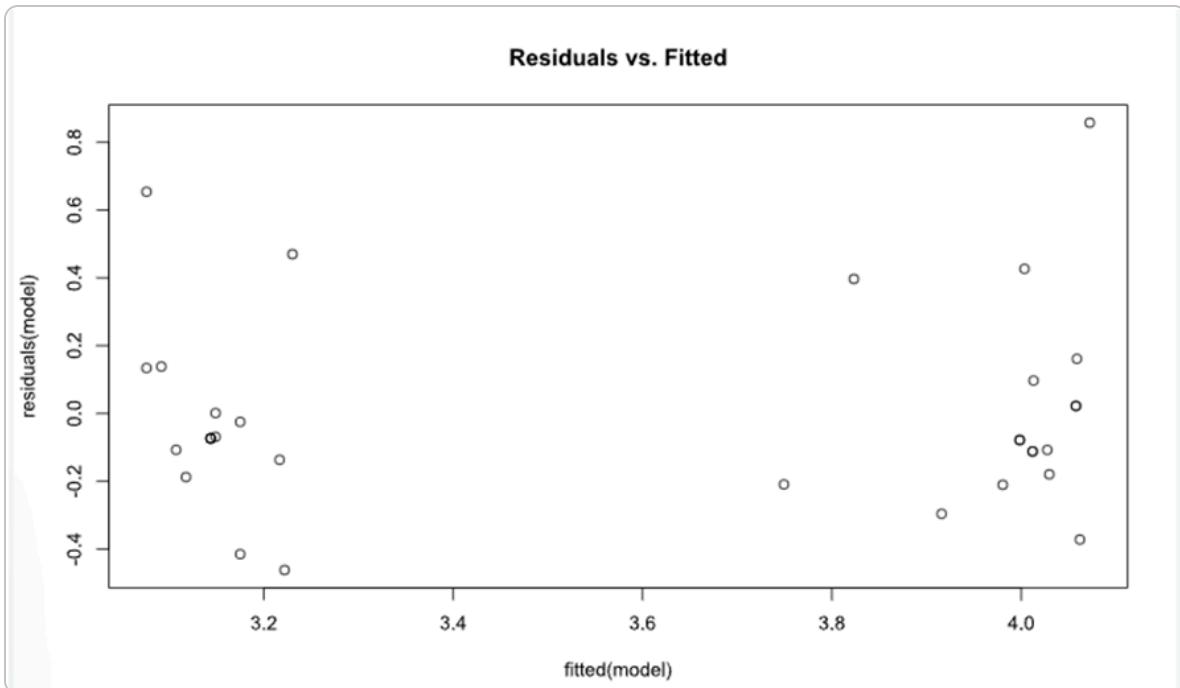
	drat	n	mean	sd	min	Q1	median	Q3	max
2.7600	2	3.4900	0.0420	3.4600	3.4750	3.4900	3.5050	3.5200	
2.9700	1	5.2500	NA	5.2500	5.2500	5.2500	5.2500	5.2500	
3.0000	1	5.4240	NA	5.4240	5.4240	5.4240	5.4240	5.4240	
3.0700	3	3.8600	0.1840	3.7300	3.7550	3.7800	3.9250	4.0700	
3.0800	2	3.5300	0.4450	3.2150	3.3720	3.5300	3.6880	3.8450	
3.1500	2	3.4580	0.0040	3.4350	3.4360	3.4380	3.4390	3.4400	
3.2100	1	3.5700	NA	3.5700	3.5700	3.5700	3.5700	3.5700	
3.2300	1	5.3450	NA	5.3450	5.3450	5.3450	5.3450	5.3450	
3.5400	1	3.5700	NA	3.5700	3.5700	3.5700	3.5700	3.5700	
3.6200	1	2.7700	NA	2.7700	2.7700	2.7700	2.7700	2.7700	
3.6900	1	3.1900	NA	3.1900	3.1900	3.1900	3.1900	3.1900	
3.7000	1	2.4650	NA	2.4650	2.4650	2.4650	2.4650	2.4650	

ANOVA, one-way with random blocks

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



ANOVA, one-way with random blocks, plot1



ANOVA, one-way with random blocks, plot2

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 of getOption("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.

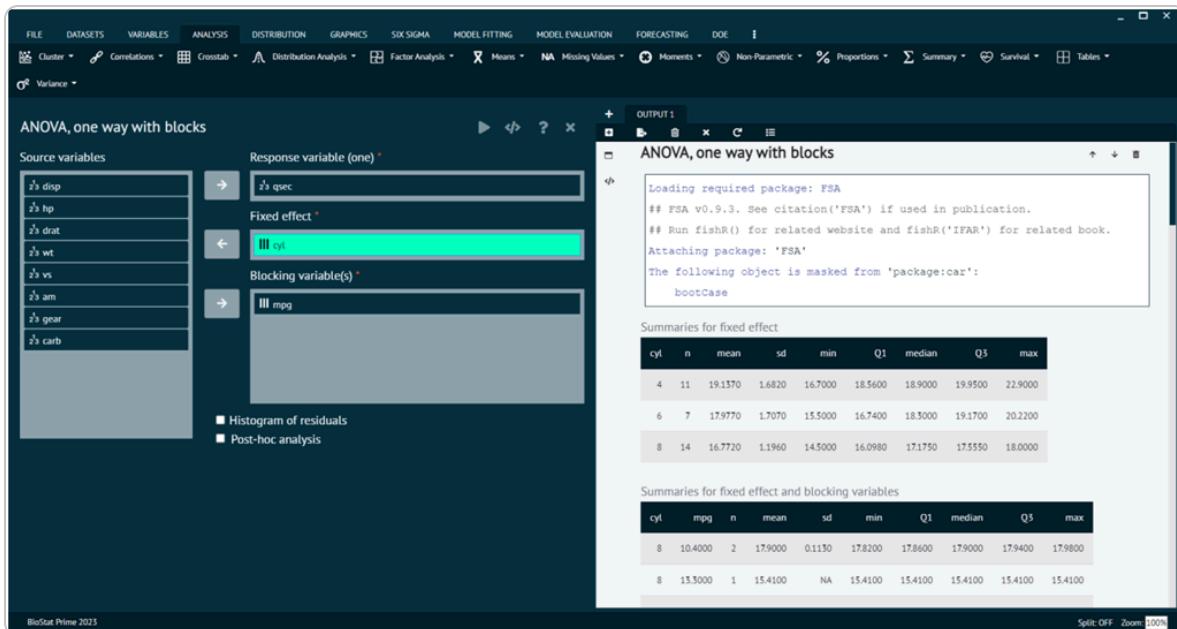


Blocks are used to create more homogeneous groups within which the experimental units are similar.

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.



ANOVA, one way with blocks

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

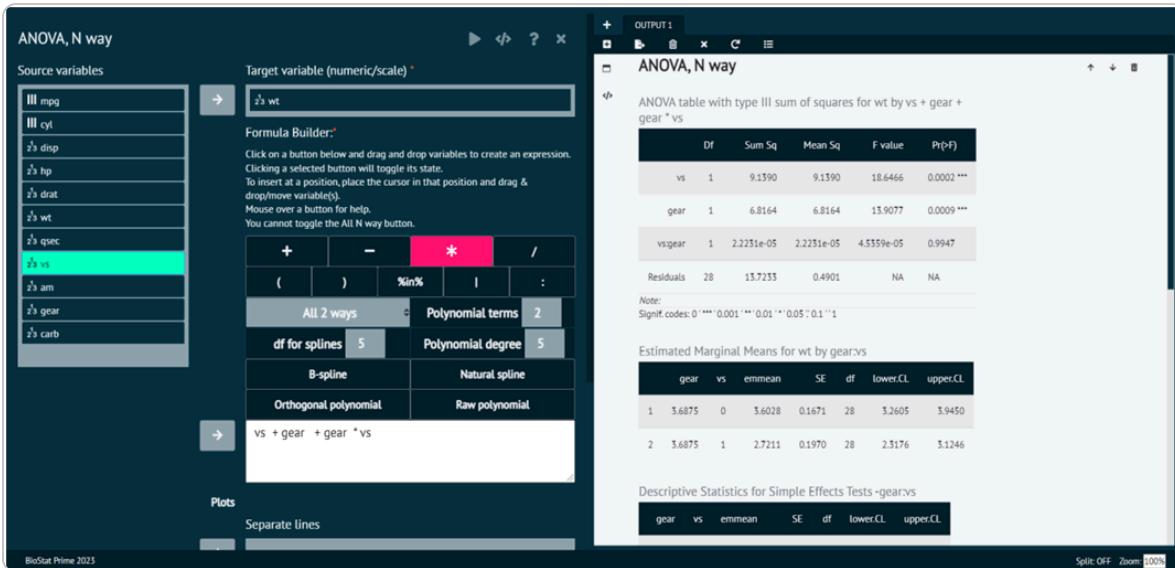
In statistics, an N-way ANOVA (Analysis of Variance) with blocks refers to a statistical analysis that involves multiple independent variables or factors (more than two). The term "blocks" in this context refers to a **way of handling potential sources of variability** that are not the primary focus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



ANOVA, N way

⚠ 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.

⚠

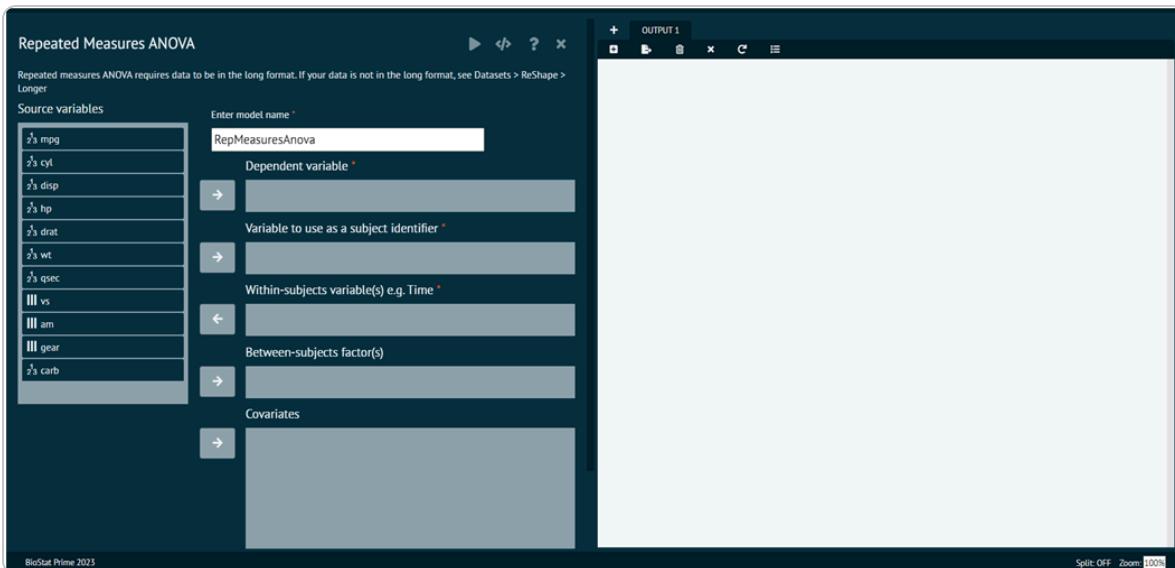
1. Click on a button in formula builder and drag and drop variables to create an expression.
2. Clicking a selected button will toggle its state.

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.



ANOVA Repeated Measures, Long

⚠ 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.

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

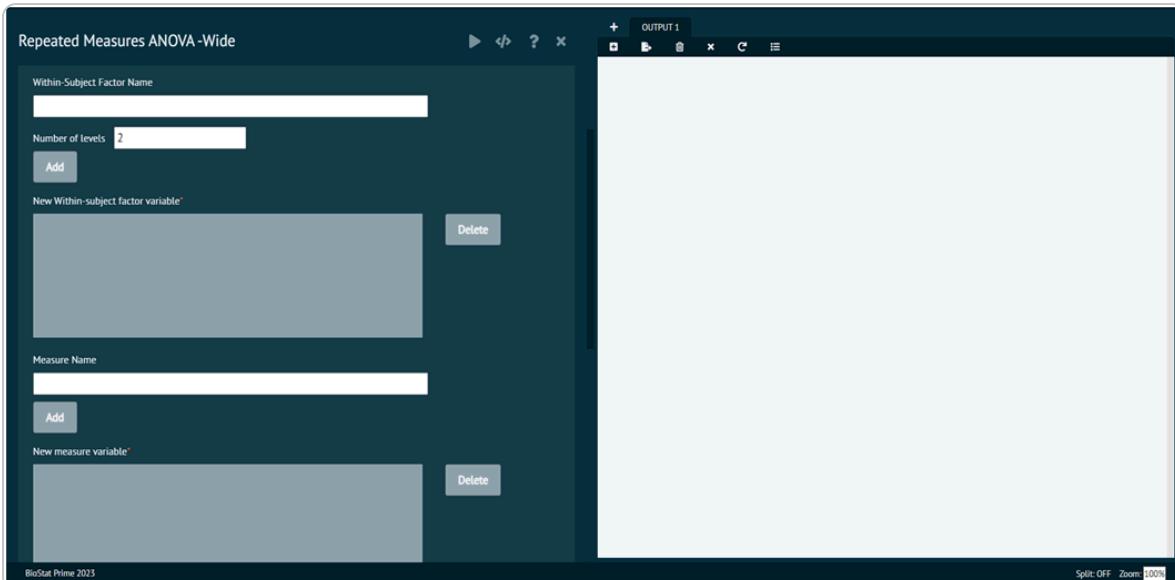
anovatable

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



ANOVA Repeated Measures, Wide

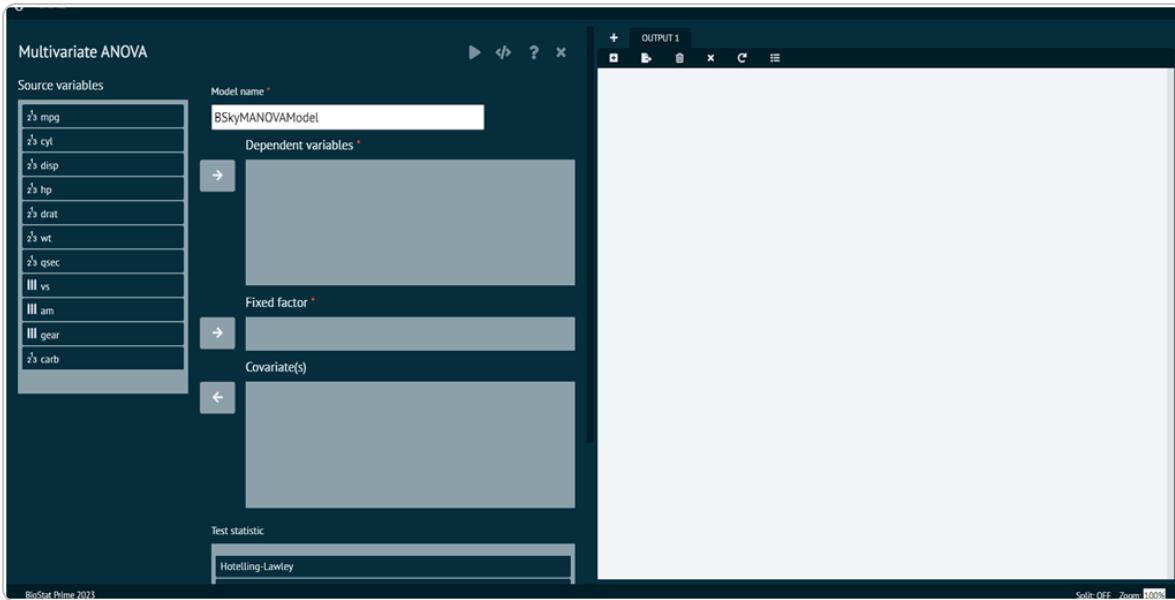
⚠ NOTE:

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.



MANOVA

Multivariate ANOVA

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

i NOTE: BioStat Prime currently support a single independent factor

i 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

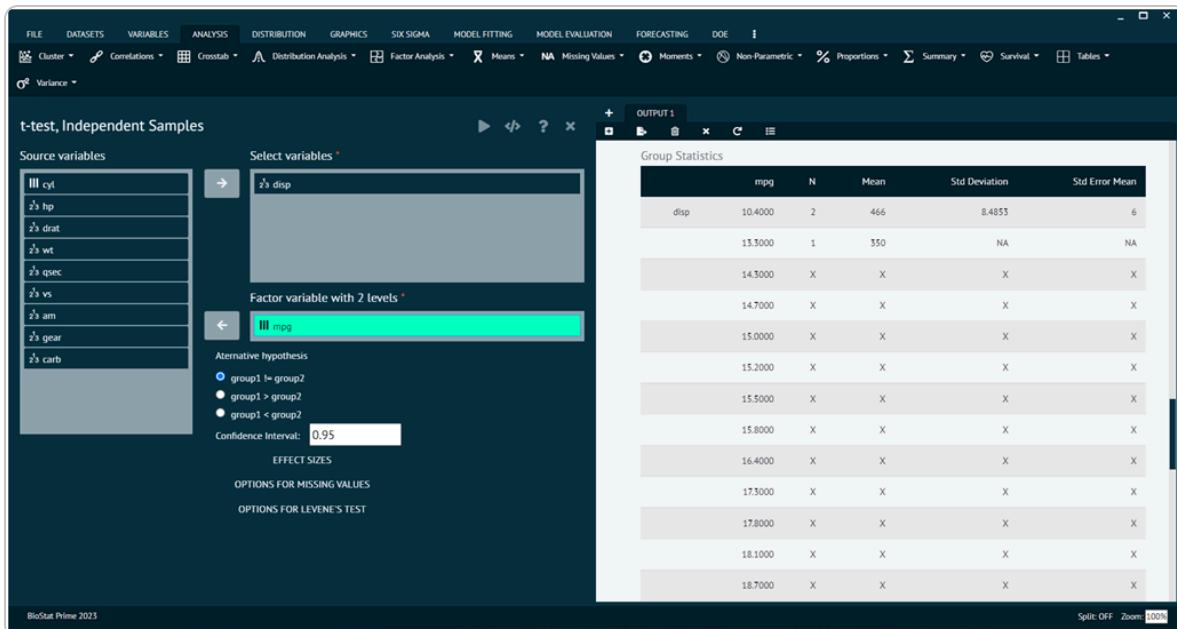
An independent samples t-test is a statistical test used to compare the means of two independent groups to determine if there is a significant difference between them. It's commonly employed when user has two separate groups of observations, and user wants to assess whether the means of these groups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



t-test, Independent

⚠ 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 2023 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, the 'Select variables' list shows 'qsec' highlighted. Below this, the 'Alternative hypothesis' section has three radio button options: 'Population mean = mu' (selected), 'Population mean > mu', and 'Population mean < mu'. The 'Test value (mu)' field contains '0' and the 'Confidence interval:' field contains '0.95'. To the right, the 'OUTPUT' tab is active, showing the results of the t-test. The output includes a note about an error with the 'disp' variable, followed by the 't-test, One Sample' results. The 'One Sample Statistics' table shows:

	N	Mean	Std Deviation	Std Error Mean
qsec	32	17.8487	1.7869	0.3159

The 'One Sample t-test' table shows:

	Test Value = 0					
	t	df	Sig.(2-tail)	mean difference	lower	upper
qsec	56.5031	31	7.7905e-33 ***	17.8487	17.2045	18.4950

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

t-test, One Sample

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.

The screenshot shows the Biostat Prime 2023 software interface. The main window displays a 't-test, Paired Samples' dialog. In the 'Source variables' list, 'mpg' and 'cyl' are selected. The 'First numeric variable' is set to 'vs' and the 'Second numeric variable' is set to 'wt'. The 'Alternative hypothesis' is set to 'Difference < mu'. The 'Null hypothesis (mu)' is set to 0, and the 'Confidence level' is 0.95. The 'EFFECT SIZES' section is collapsed.

The right side of the screen shows the 'OUTPUT 1' tab with the following content:

t-test, Paired Samples

Summary Statistics

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew
vs	1	32	0.4375	0.5040	0.0000	0.4231	0.0000	0.0000	1.0000	1.0000	0.2403
wt	2	32	5.2172	0.9785	5.3250	5.1527	0.7672	5.1530	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-15 ***	-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

t-test, Paired Samples

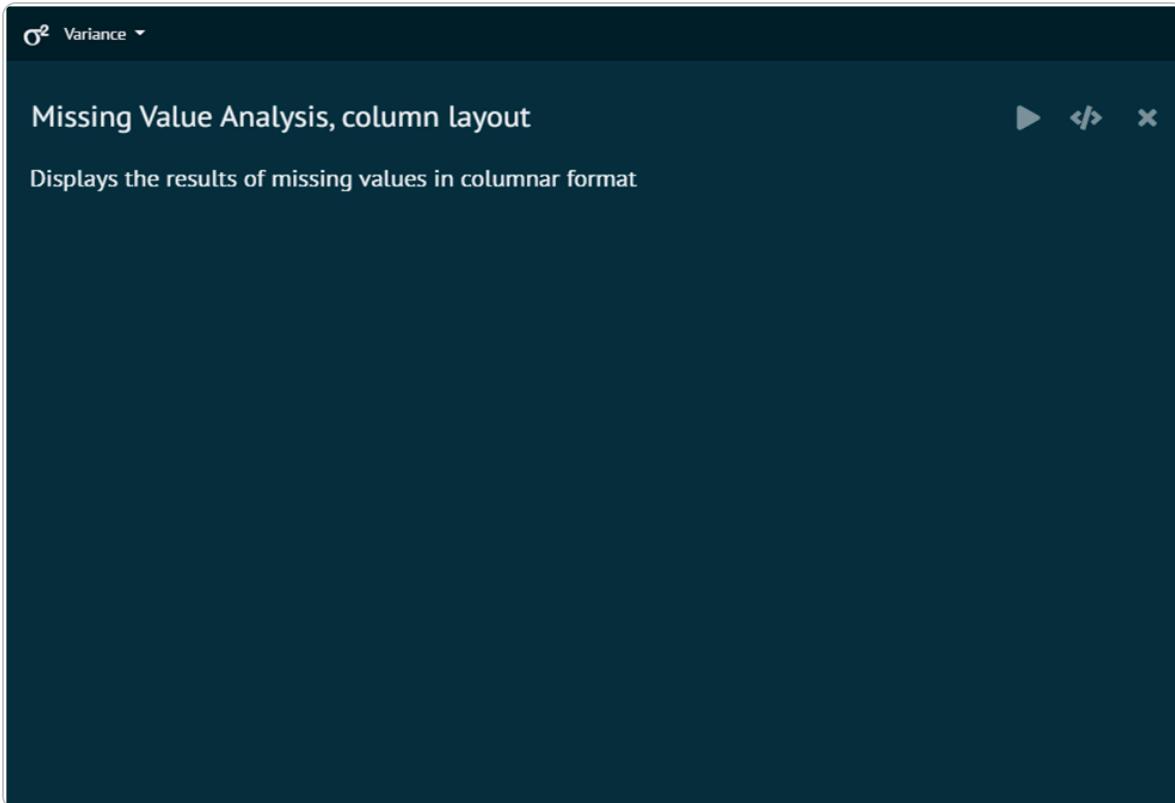
Missing Value

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.



Column layout

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.

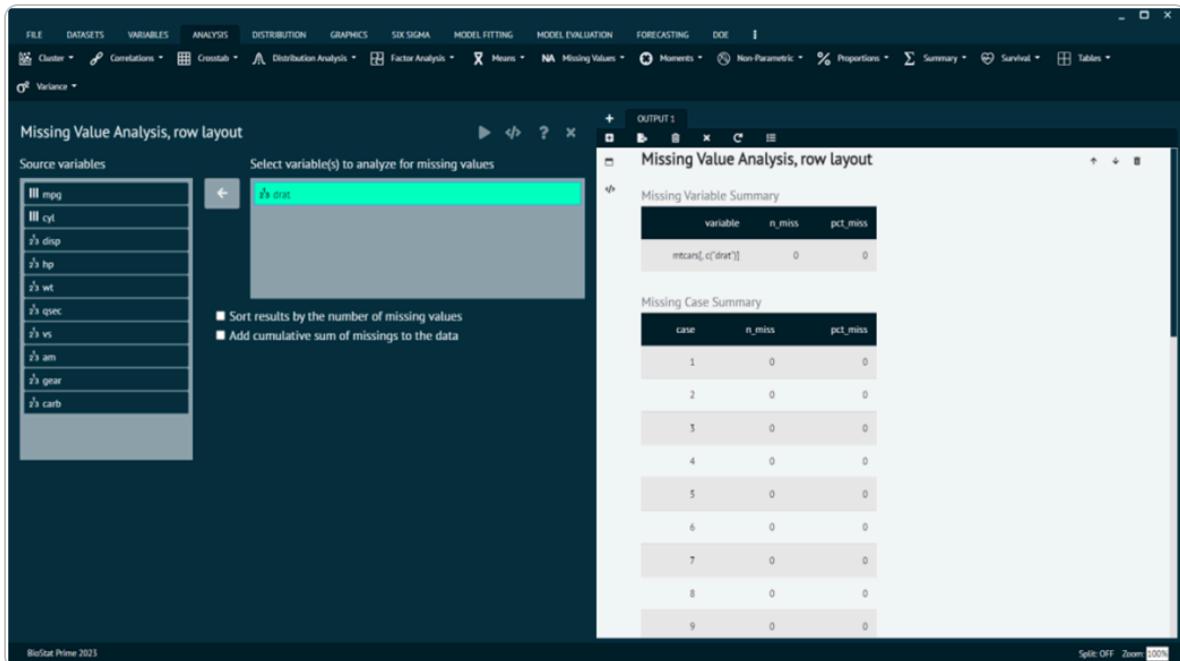
Analyzes missing values and displays results in rows, displays summary information of missing values at the variable level and lists the number of missing values on each row for variables being analyzed.

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.



Row layout

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

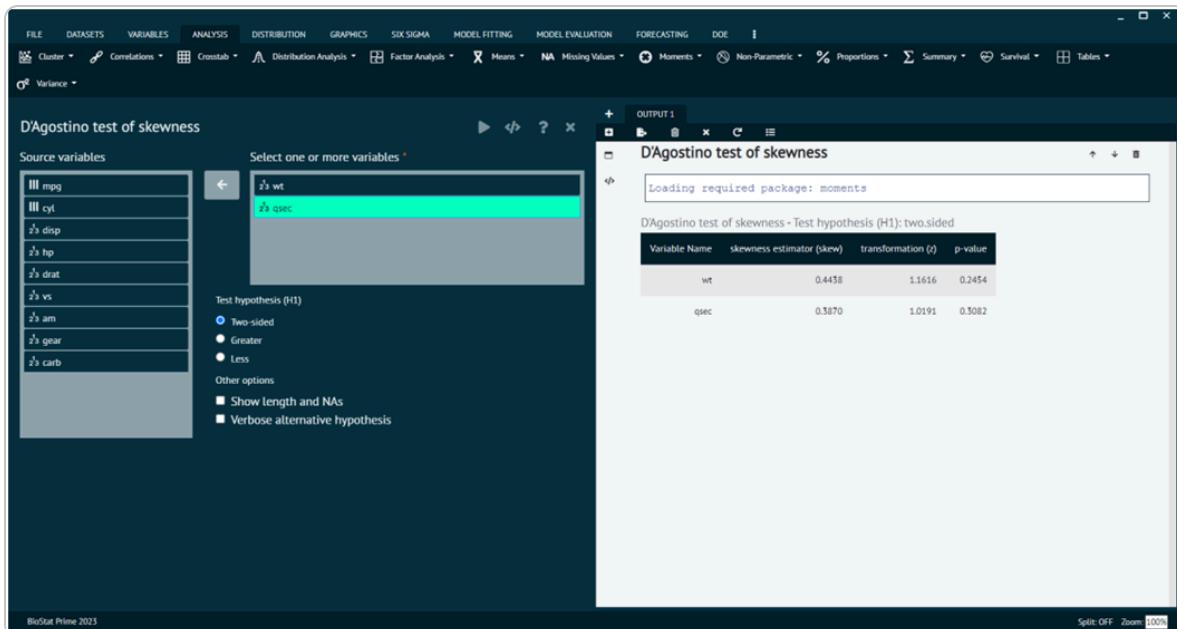
The D'Agostino skewness test is a statistical test used to assess whether the skewness of a sample differs from what would be expected in a normal distribution. Skewness measures the asymmetry of a distribution, indicating whether the data is skewed to the left or right. The D'Agostino skewness test 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.



D'Agostino skewness test

! 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 2021 software interface. The main window displays the 'Chi-squared Test' dialog. On the left, under 'Source variables', a list of variables is shown: cyl, mpg, am, vs, carb, gear, wt, drat, hp, disp. 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 right panel shows the 'OUTPUT 1' tab with the following results:

Chi-squared test for given probabilities		
Test Result		
X-squared	df	p-value
3.9375	24	1.0000

Additional Details:
Test Method Performed: Chi-squared test for given probabilities

Frequencies for variable cyl:

	Observed	Expected	Residuals
4	11	10.6667	0.1021
6	7	10.6667	-1.1227
8	14	10.6667	1.0206

Chi-squared test for given probabilities

Test Result		
X-squared	df	p-value

Chi-Square test

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.

The screenshot shows the BioStat Prime 2023 software interface. The main window displays the 'Friedman Test' dialog. On the left, under 'Source variables', there is a list of variables: mpg, cyl, disp, drat, spec, am, gear, carb. On the right, under 'Response Variables, 2 or more', three variables are listed: hp, wt, vs. The 'vs' variable is highlighted with a green background. The output window on the right is titled 'Friedman Test'. It contains sections for 'Medians' (showing values for hp, wt, vs), 'Friedman rank sum test' (Test Result table with Friedman chi-squared: 64, df: 2, p-value: 1.2664e-14 ***), and 'Additional Details' (Additional Comments section). The status bar at the bottom indicates 'BioStat Prime 2023' and 'Split OFF Zoom 100%'. The overall title of the dialog is 'Friedman Test'.

Friedman Test

⚠ 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.

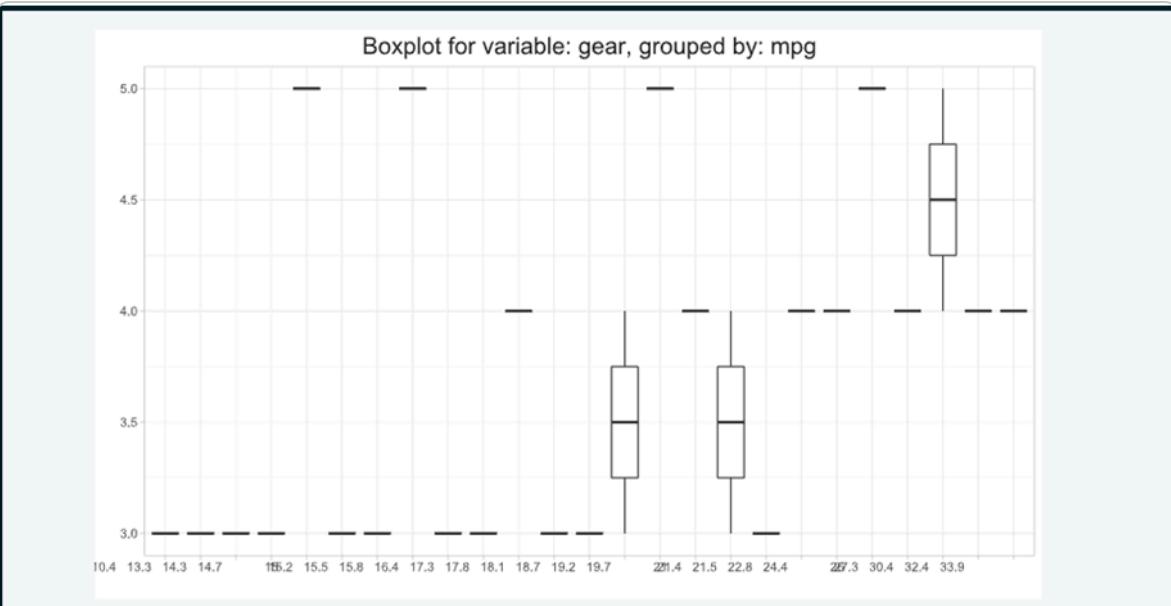
Load the dataset -> Click on the analysis tab in main menu -> Select Non-Parametric -> The non-parametric tab leads to Kruskal-Wallis Rank Sum Test -> In the dialog select the variable and options according to the requirement -> Execute the dialog.

The screenshot shows the BioStat Prime 2023 software interface. On the left, the 'Kruskal-Wallis Rank Sum Test' dialog is open. It has several sections: 'Source variables' (containing 'cyt', 'disp', 'hp', 'drat', 'wt', 'qsec', 'vs', 'am', 'carb'), 'Response variable' (set to 'gear'), 'Factor variable' (set to 'mpg'), 'Estimation method' (radio buttons for 'Asymptotic' (selected), 'Monte Carlo', and 'Exact'), 'Enter the number of simulations' (set to 10000), 'Multiple comparison adjustment' (list including 'holm', 'hochberg', 'hommel', 'bonferroni', and 'fdr'), and 'Options for handling ties' (list including 'mid-ranks' (selected) and 'averane-krusk'). On the right, the 'OUTPUT 1' window displays the results of the test. It shows a message 'Loading required package: coin'. Below it is a table titled 'Summary Statistics' with the following data:

TargetVariable	mpg	count	min	Quantile_1st_25	mean	median	Quantile_3rd_75	max
gear	10.4000	2	3	5.0000	3.0000	3.0000	5.0000	3
gear	13.9000	1	3	1.0000	3.0000	3.0000	3.0000	3
gear	14.3000	1	3	1.0000	3.0000	3.0000	3.0000	3
gear	14.7000	1	3	1.0000	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

Kruskal-Wallis Rank Sum Test

Box plot for variable.



Kruskal-Wallis Rank Sum Test,plot

⚠ 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")`.

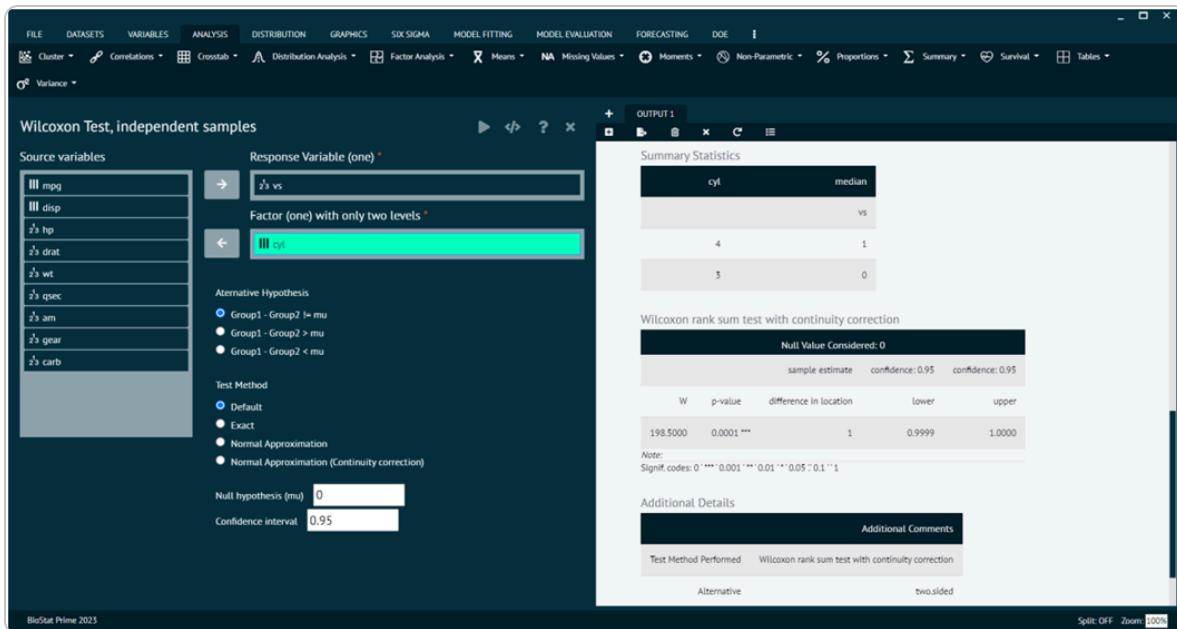
Wilcoxon Test, independent samples

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

! 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")`.

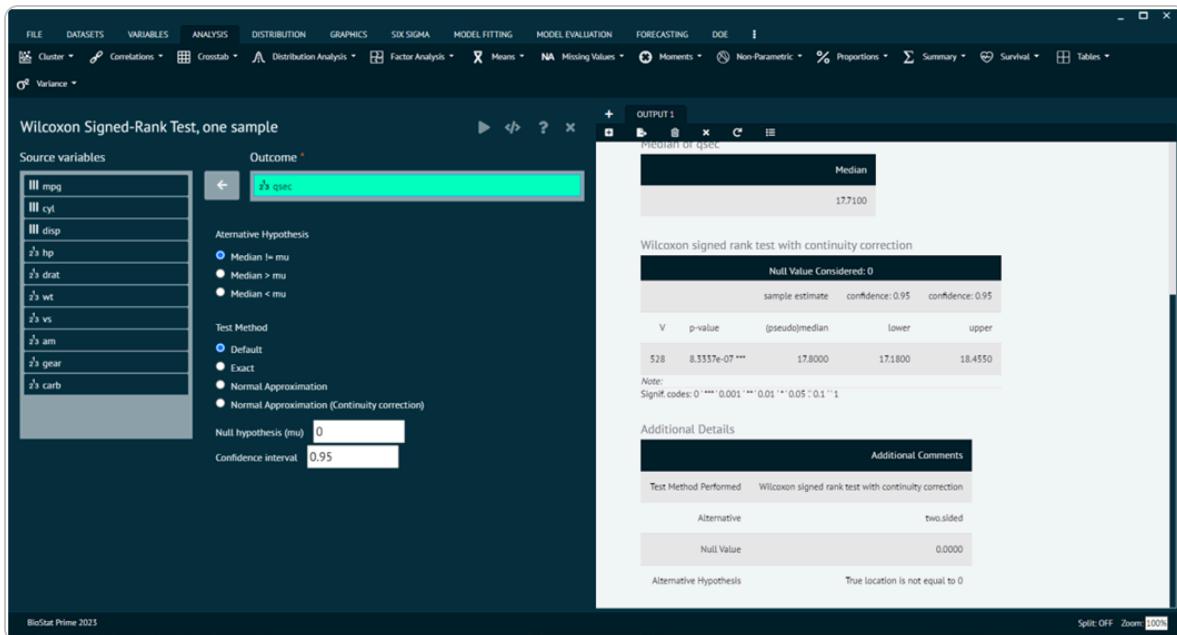
Wilcoxon Signed-Rank Test, one sample

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.



Wilcoxon Signed-Rank Test, one sample

⚠ 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")`.

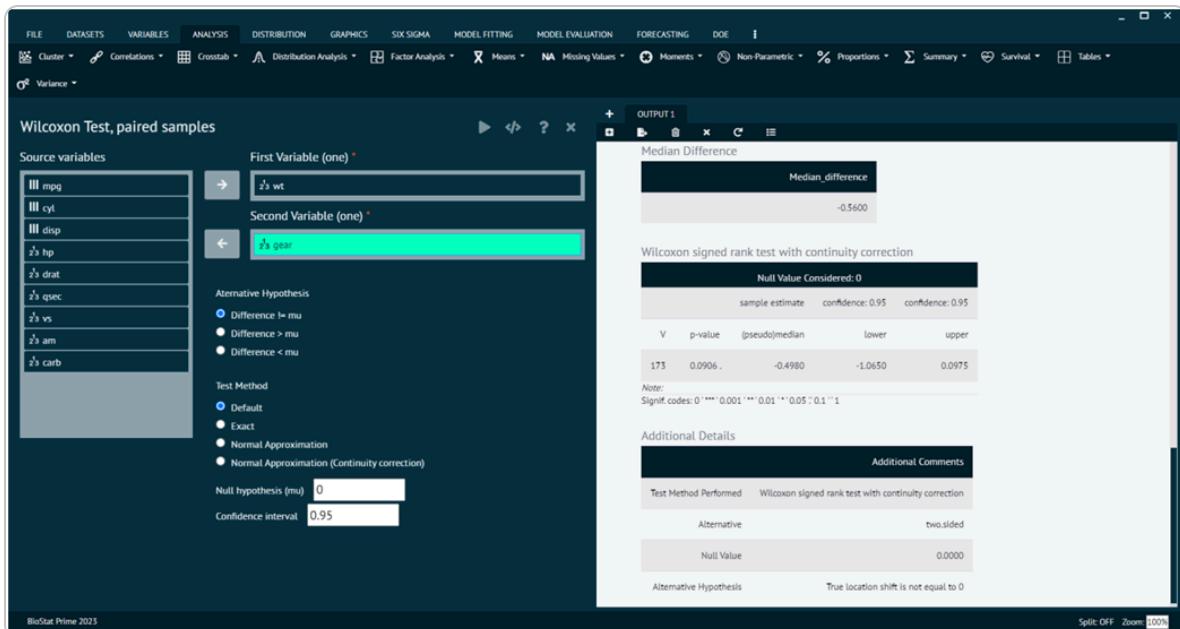
Wilcoxon Test, Paired samples

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.



Wilcoxon Test, Paired samples

! 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.

The screenshot shows the BioStat Prime 2021 software interface. On the left, the 'Two Sample Proportion Test' dialog is open. It lists various car attributes as source variables: disp, hp, drat, wt, qsec, vs, am, gear, carb. Under 'GroupBy, specify a factor variable', 'mpg' is selected. Under 'Response variable, factor variable with 2 levels only', 'cyl' is selected. The 'Alternative hypothesis' section has 'Population mean != mu' selected. The 'Confidence level' is set to 0.95. A checkbox for 'With continuity correction' is checked. To the right of the dialog, an 'OUTPUT 1' window titled 'Percentage Table' displays a table of data:

	mpg	cyl	Total	Count
10.4000	0	0	100	2
13.3000	0	0	100	1
14.3000	0	0	100	1
14.7000	0	0	100	1
15.0000	0	0	100	1
15.2000	0	0	100	2
15.5000	0	0	100	1
15.8000	0	0	100	1
16.4000	0	0	100	1
17.3000	0	0	100	1
17.8000	0	100	0	1
18.1000	0	100	0	1

Two Sample Proportion Test

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.

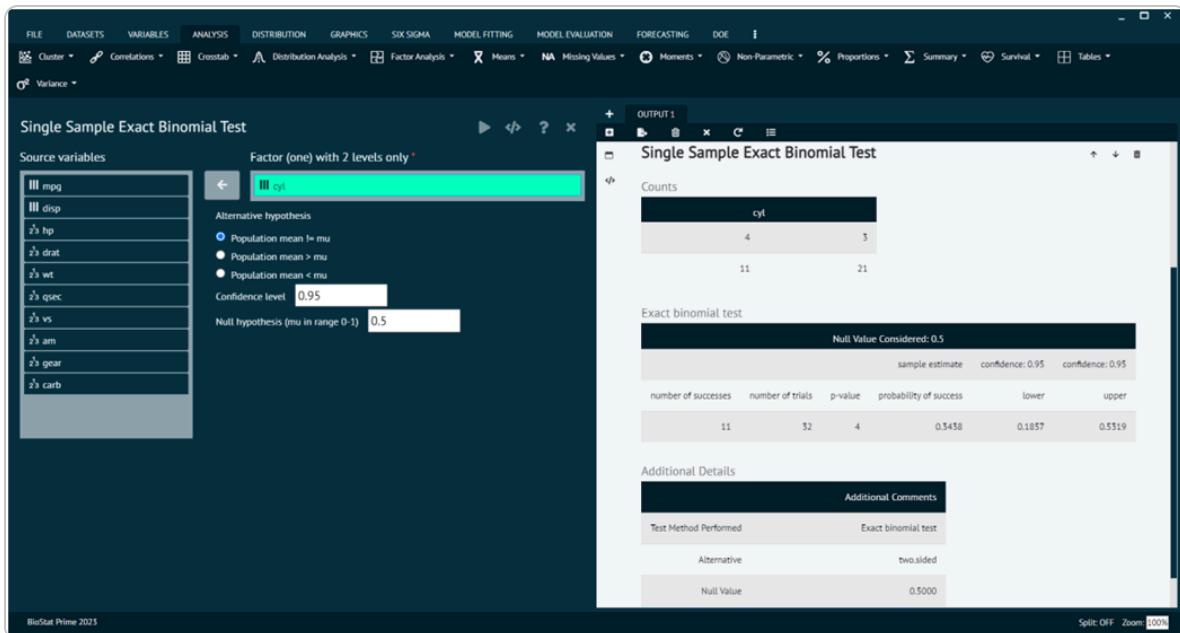


Performs an exact test of a simple null hypothesis about the probability of success in a Bernoulli experiment.

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.



Single Sample Exact Binomial Test

! 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.

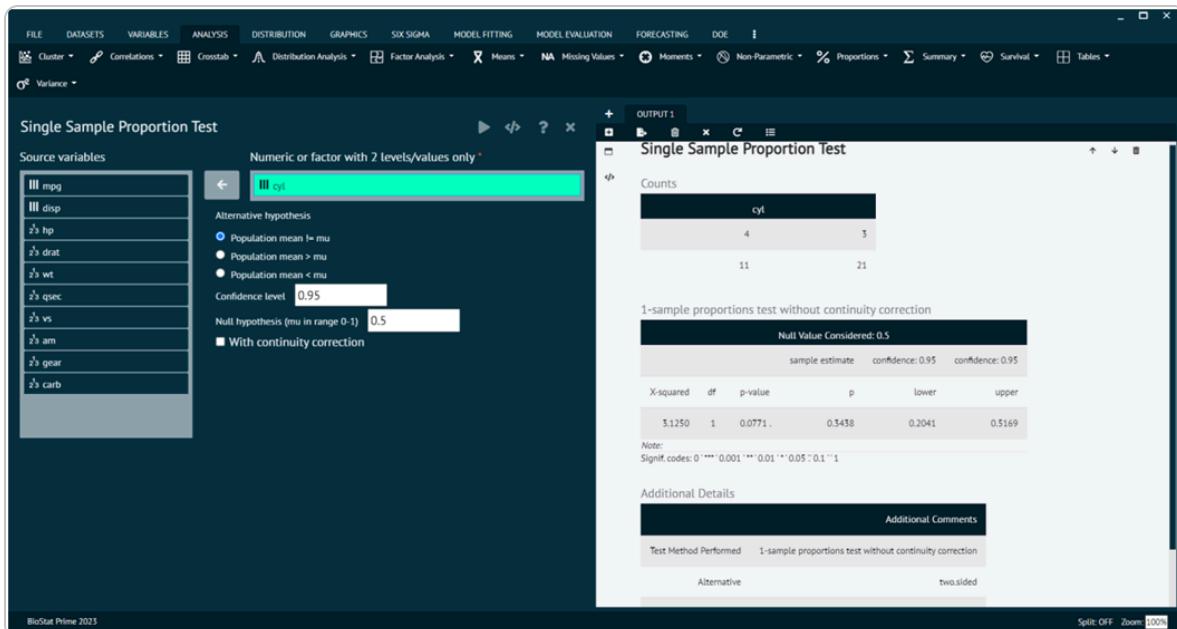


prop.test can be used for testing the null that the proportions (probabilities of success) in several groups are the same, or that they equal certain given values.

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.



Single Sample Proportion Test

! 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.

The screenshot shows the BioStat Prime software interface. The main menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, MODEL FITTING, MODEL EVALUATION, FORECASTING, DOE, and various statistical analysis options like Cluster, Correlations, Crosstab, Distribution Analysis, Factor Analysis, Means, Missing Values, Moments, Non-Parametric, Proportions, Summary, Survival, and Tables. The ANALYSIS tab is selected. Below the menu is a toolbar with icons for Cluster, Correlations, Crosstab, Distribution Analysis, Factor Analysis, Means, Missing Values, Moments, Non-Parametric, Proportions, Summary, Survival, and Tables. The main workspace shows the 'Descriptives' dialog. On the left, under 'Source variables', there is a list of variables: mpg, disp, hp, drat, qsec, vs, am, gear, carb. The variable 'gear' is highlighted with a green background. An arrow points from this list to the 'Selected variables' section on the right, which contains 'wt' and 'cyl'. Another arrow points from the 'Selected variables' section to a 'Group by' section, which is currently empty. At the bottom of the dialog, there is a section titled 'Options for numeric variables' with checkboxes for Min, Max, Mean, Median, Sum, Standard deviation, and Std error of mean, all of which are checked. To the right of the dialog is an 'OUTPUT 1' window titled 'Numerical Statistical Analysis by Variable'. It displays a table with three columns: stats, wt, and cyl. The table rows show statistical values: min (1.5130, 2.5812), mean (3.2172, 6.1875), median (3.3250, 6.0000), max (5.4240, 8.0000), sd (0.9785, 1.7859), std.error (0.1730, 0.3157), cv (0.3041, 0.2886), var (0.9574, 5.1895), n (32.0000, 32.0000), and NAs (0.0000, 0.0000). The bottom right corner of the output window shows 'Split OFF' and 'Zoom 100%'. The overall title of the image is 'Descriptive Statistics'.

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.

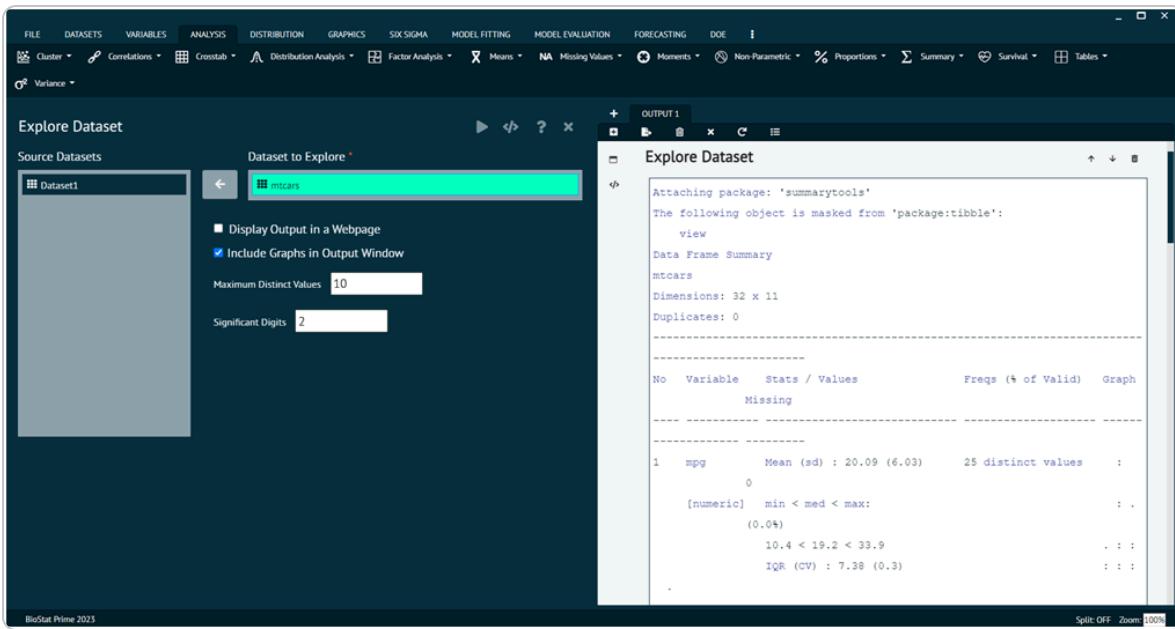
Explore Dataset

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.



Explore Dataset

⚠ 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.

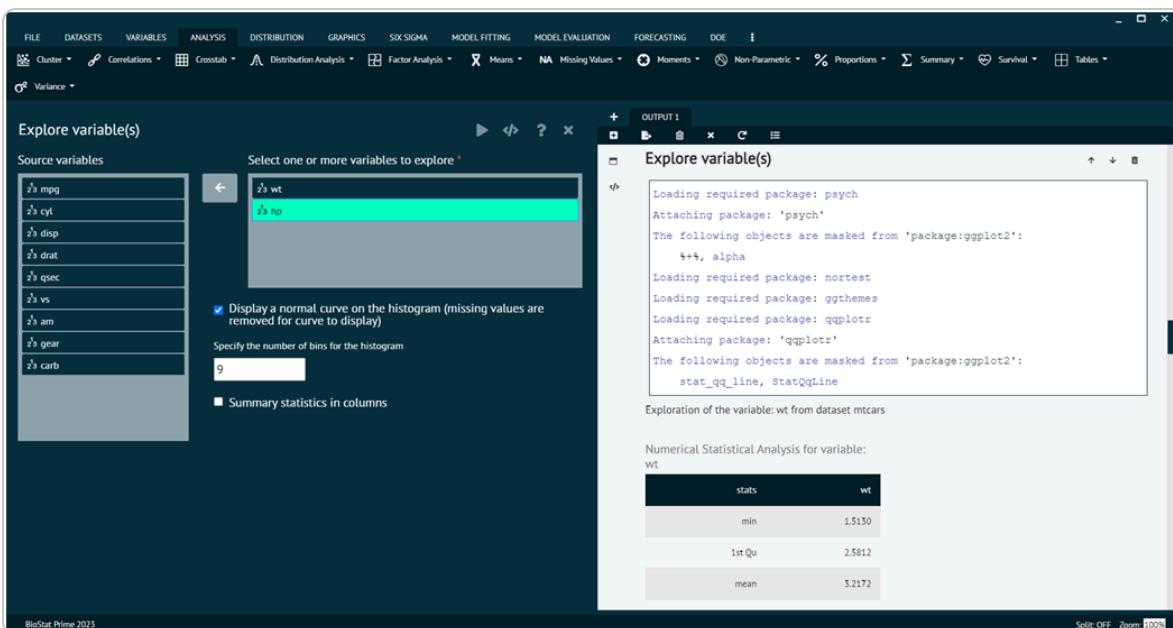
Explore Variables

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.



Explore Variables

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 menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION, FORECASTING, DOE, and various statistical analysis options like Cluster, Correlations, Crosstab, Distribution Analysis, Factor Analysis, Means, Missing Values, Moments, Non-Parametric, Proportions, Summary, Survival, and Tables. The main window title is "Frequency Table". On the left, under "Source variables", the "wt" variable is highlighted with a green selection bar. An arrow button points from the source list to a "Select variables" list on the right, which contains "hp". The output panel on the right displays a table titled "Frequency Table for hp" with the following data:

hp	Frequency	Percent	CumPercent	Valid Percent	Valid CumPercent
110	3	9.3750	45.7500	9.3750	45.7500
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	55.1250	6.2500	55.1250
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

Frequency

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.

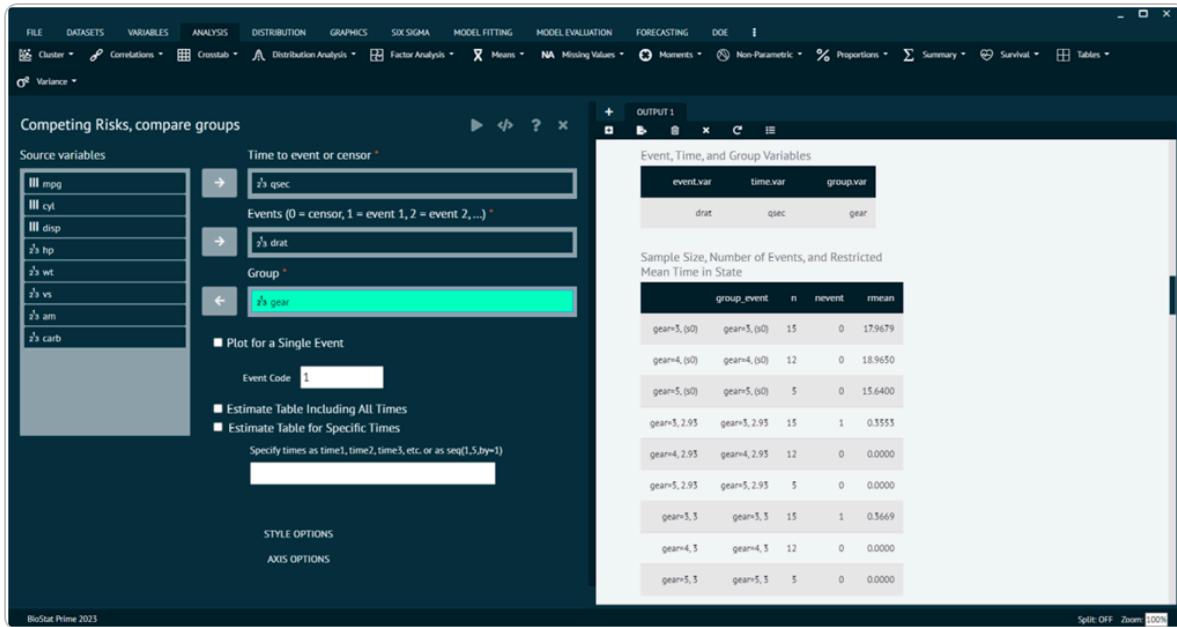
Competing Risks, Compare groups

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.

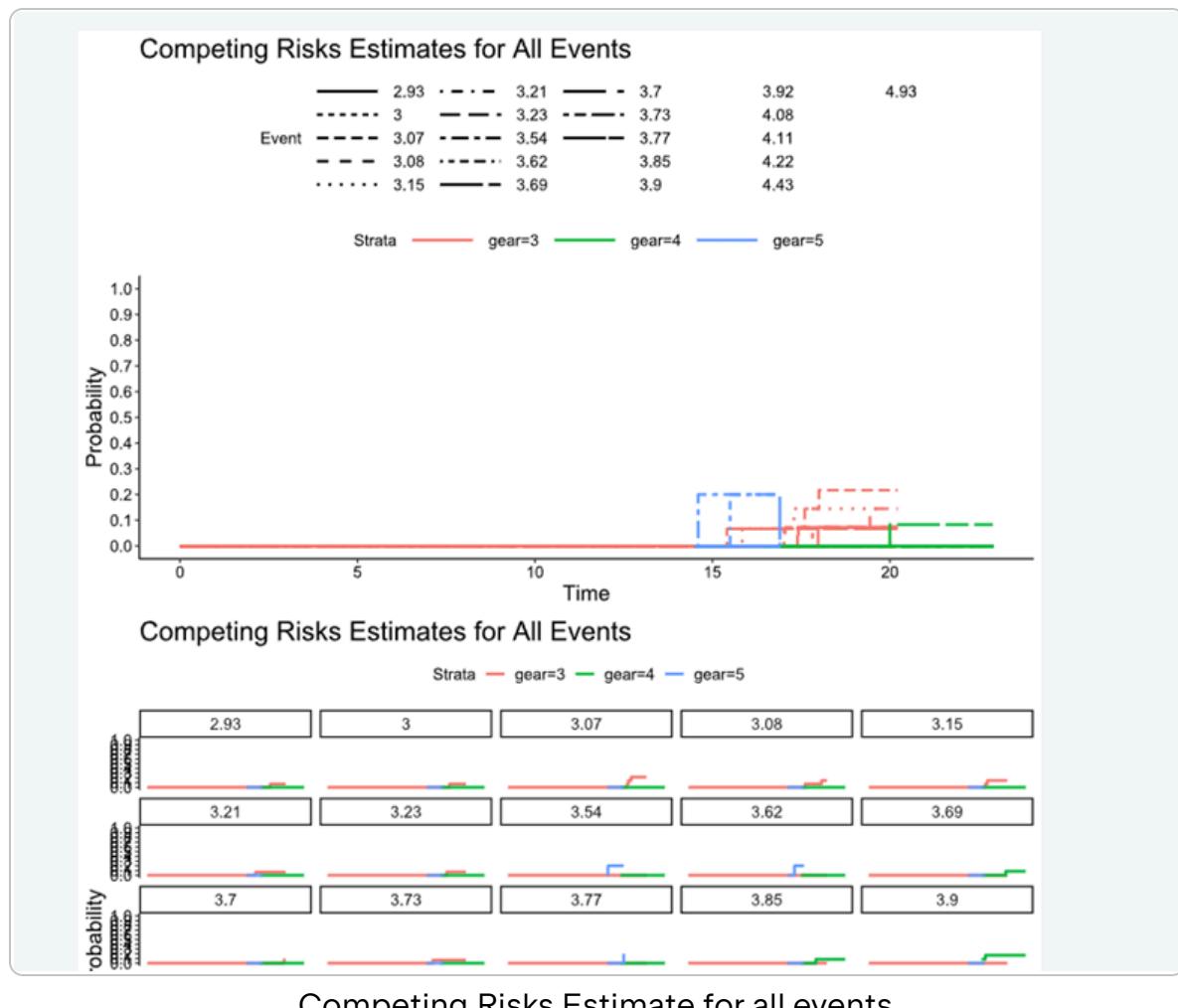
Steps

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.



Competing Risks, Compare groups

Competing Risks Estimate for all events in the output window.



Competing Risks, One group

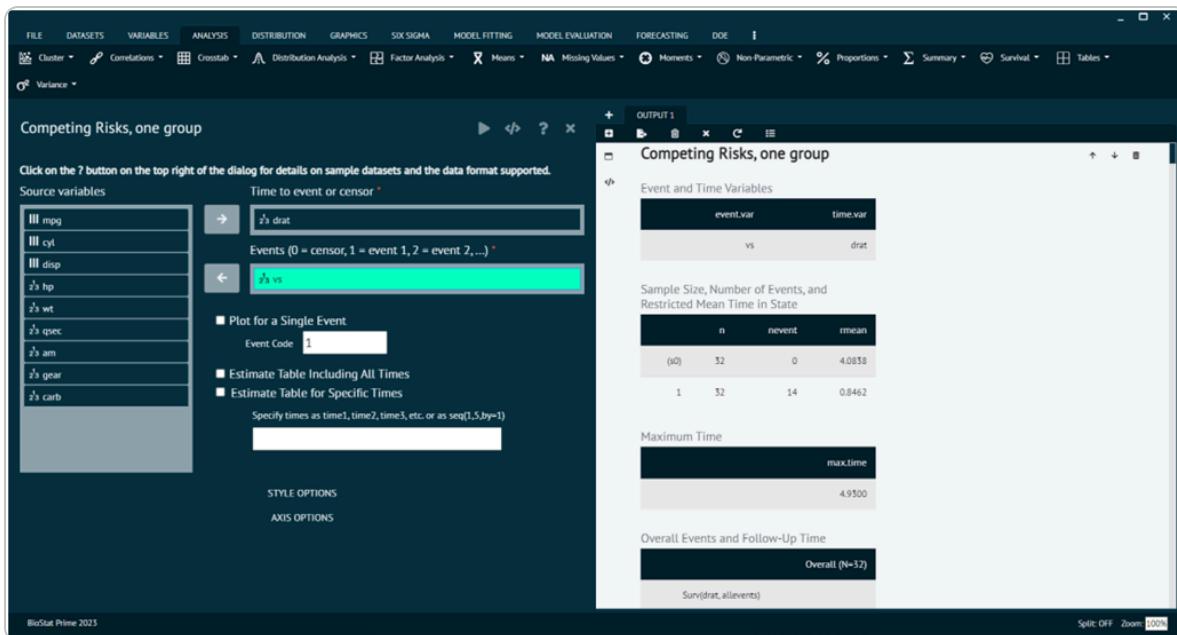
When dealing with competing risks in a single group, one is interested in understanding the probabilities and risks associated with different types of events that may occur, but one does not have a distinct comparison group. The analysis will focus on estimating and comparing the cumulative incidence functions for the competing events within the same group.

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

Steps

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.



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.

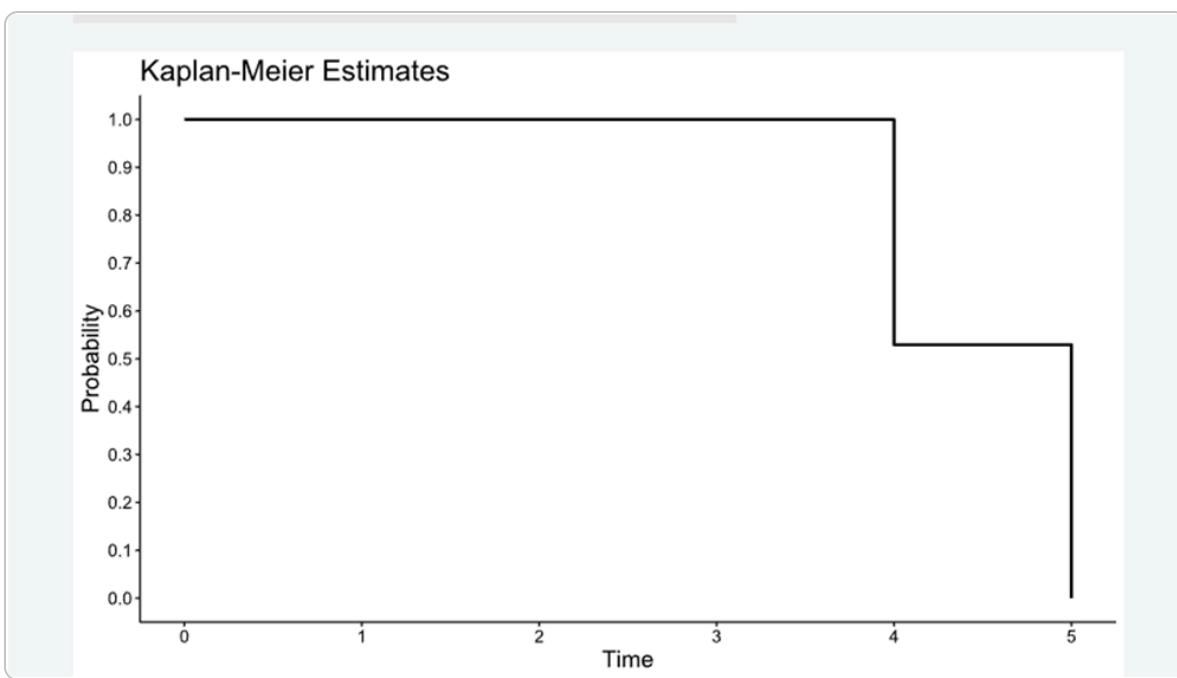
Steps

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.

The screenshot shows the Biostat Prime 2023 software interface. The main window displays the 'Kaplan-Meier Estimation, One Group' dialog. In the 'Source variables' list, 'mpg' is selected. The 'Time to event or censor' field contains 'gear'. The 'Event (1 = event, 0 = censor)' field contains 'am'. Under 'Plot Type', 'Survival' is selected. In the 'OUTPUT' tab, the 'Survival Summary' section shows 'Overall (N=32)' with 'N' at 32.0000, 'Events' at 13.0000, 'Median Survival' at 5.0000, and 'Median Follow-Up' at 4.0000. Below it, 'Restricted Mean and Median Survival Times' are listed with values: rmean 4.5294, rmean.std.error 0.1211, median 5, conflow 4, and confhigh NA. A Kaplan-Meier Estimates plot is shown with the y-axis labeled 'Probability' from 0.0 to 1.0 and the x-axis labeled 'Time' from 0 to 5. The survival probability starts at 1.0 and drops to approximately 0.55 at time 4.

Kaplan-Meier Estimation, One group

Kaplan-Meier Estimates in the output.



Kaplan-Meier Estimates in the output

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 BiStat Prime 2023 software interface. On the left, the 'Table, Advanced' panel is open, showing a list of source variables: cyl, hp, wt, am, carb. Below this, there are three sections for statistical tests: 'Groups to Compare' (with 'drat' selected), 'Variables for ANOVA Test' (with 'vs' selected), and 'Variables for Kruskal-Wallis Test' (with 'gear' selected). A note at the top of this panel 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'. On the right, an 'OUTPUT 1' window displays 'Variable Summaries' for the selected variables. The table shows mean, median, and quartiles across different groups defined by the 'Groups to Compare' variable 'drat'. The columns represent groups with N=1, N=5, N=2, N=1, N=1, and N=1 respectively. The 'Variable Summaries' table is as follows:

	2.76 (N=1)	2.95 (N=1)	5 (N=1)	3.07 (N=5)	3.08 (N=2)	3.15 (N=2)	5.21 (N=1)	5.25 (N=1)	3.54 (N=1)	
vs	- Mean (SD)	0.500 (0.707)	0.000 (NA)	0.000 (0.000)	0.500 (0.707)	0.000 (0.000)	0.000 (NA)	0.000 (NA)	0.000 (NA)	
gear	- Mean (SD)	3.000 (0.000)	3.000 (NA)	3.000 (0.000)	3.000 (0.000)	3.000 (0.000)	3.000 (NA)	3.000 (NA)	3.000 (NA)	
qsec	- Mean (SD)	18.545 (2.369)	17.980 (NA)	17.820 (NA)	17.667 (0.306)	18.245 (1.690)	17.160 (0.198)	15.840 (NA)	17.420 (NA)	14.600 (NA)
	- Median (Q1, Q3)	3.000 (2.750, 3.000)	3.000 (0.000, 0.000)	3.000 (0.000, 0.000)	3.000 (0.750, 0.750)	3.000 (0.000, 0.000)				

Table Advanced

Table, Basic

The examples for this category are Frequency Distribution Table, Summary Statistics Table, Contingency Table (Cross-tabulation) etc.

The screenshot shows the BiStat Prime 2023 software interface. The main window is titled "Table, Basic". In the "Source variables" list, "mpg", "cyl", "disp", "wt", "qsec", and "gear" are selected. The "Variables to Summarize" list contains "drat", "hp", and "vs". The "Groups to Compare (optional)" list contains "am". The "Strata (optional)" list contains "carb". The "Table Title" is set to "Variable Summaries". Under "Digits After Decimal", "Continuous Values" is set to 3, "Percentages" to 1, and "P-Values" to 3. The "OUTPUT 1" window displays the "Variable Summaries" for "carb" across groups 0 (N=19) and 1 (N=15). The table includes columns for Mean (SD), Median (Q1, Q3), and Range (Min, Max).

carb	0 (N=19)	1 (N=15)
1 drat	- Mean (SD) 5.180 (0.478)	4.058 (0.153)
	- Median (Q1, Q3) 3.080 (2.920, 3.590)	4.080 (4.022, 4.115)
2 hp	- 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)
3 vs	- Mean (SD) 1.000 (0.000)	1.000 (0.000)
	- Median (Q1, Q3) 1.000 (1.000, 1.000)	1.000 (1.000, 1.000)
4 carb	- 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)

Table, Basic

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.

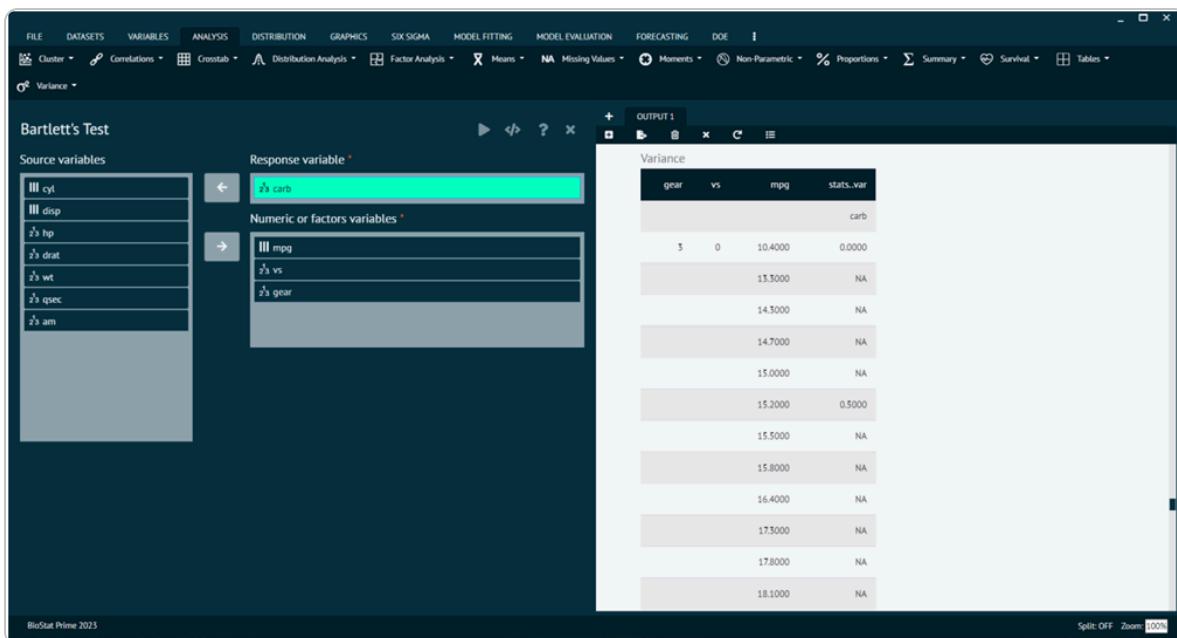
Bartlett's Test

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.

Steps

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.



Bartlett's Test

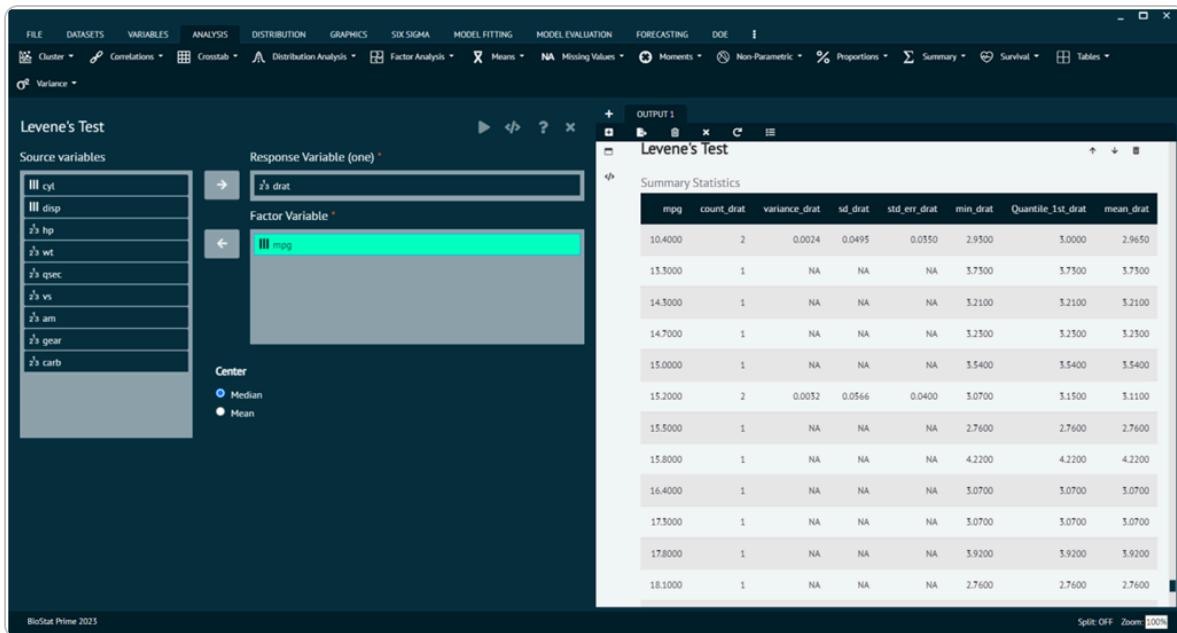
Levene's Test

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.

Steps

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.



Levene's Test

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.

The screenshot shows the Noesis BiStat Pro software interface. The main window is titled "Variance Test, F-test". In the "Source variables" list, items like mpg, disp, hp, wt, qsec, vs, am, gear, carb are listed. The "Response variable" is set to "drat" and the "Factor variable, with only two levels" is set to "cyl". Under "Alternative hypothesis", the radio button for "Difference != 1" is selected. The "Confidence level" is set to 0.95. The output window titled "Variance Test, F-test" displays the following results:

cyl	var
4	0.1336
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

Test Method Performed		Additional Comments	
Alternative	two.sided	F test to compare two variances	

Variance Test F-test

Distribution Analysis

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.

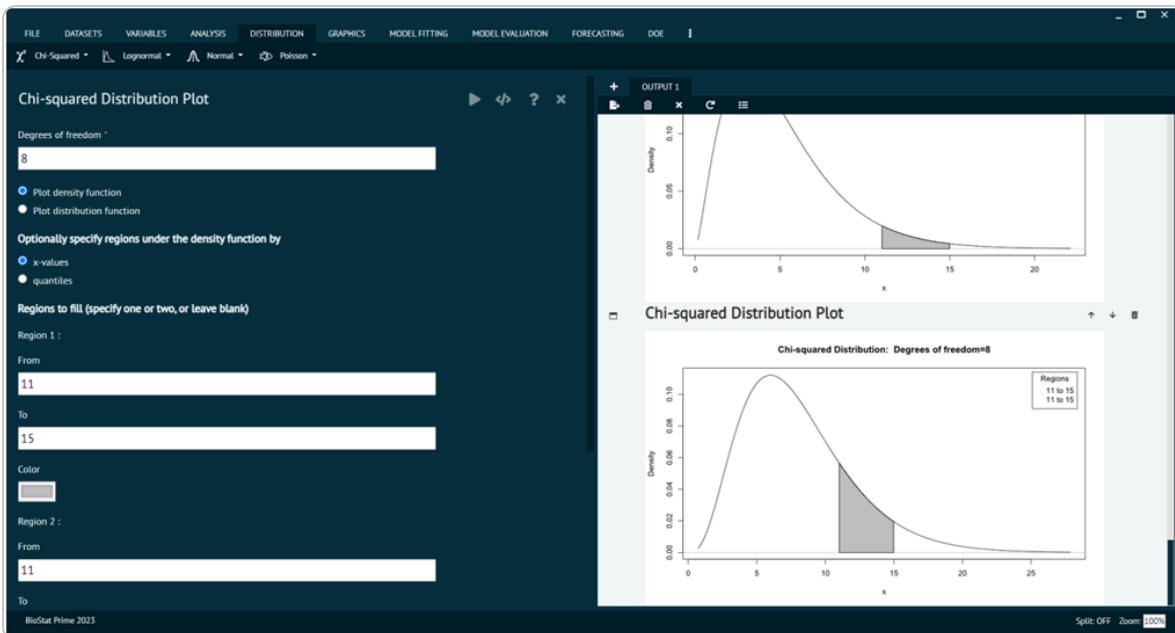
Chi-square Distribution plot

Density, distribution function, quantile function and random generation for the chi-squared (chi²) distribution with df degrees of freedom and optional non-centrality parameter ncp.

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

Steps

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 Distribution plot

Usage

i `dchisq(x, df, ncp = 0, log = FALSE)`

i `pchisq(q, df, ncp = 0, lower.tail = TRUE, log.p = FALSE)`

i `qchisq(p, df, ncp = 0, lower.tail = TRUE, log.p = FALSE)`

i `rchisq(n, df, ncp = 0)`

Value

`dchisq` gives the density, `pchisq` gives the distribution function, `qchisq` gives the quantile function, and `rchisq` generates random deviates. Invalid arguments will result in return value `NaN`, with a warning.

The length of the result is determined by `n` for `rchisq`, and is the maximum of the lengths of the numerical arguments for the other functions. The numerical arguments other than

n are recycled to the length of the result. Only the first elements of the logical arguments are used.

⚠ Note Supplying ncp = 0 uses the algorithm for the non-central distribution, which is not the same algorithm used if ncp is omitted. This is to give consistent behaviour in extreme cases with values of ncp very near zero.

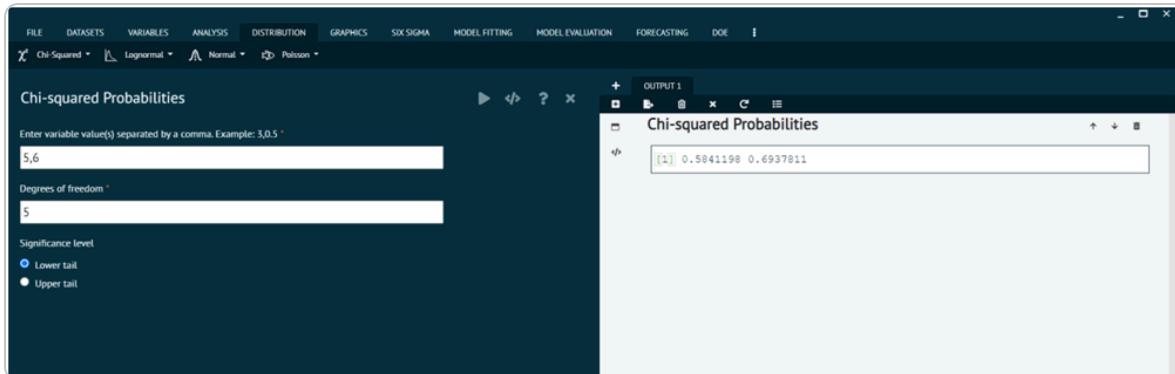
⚠ The code for non-zero ncp is principally intended to be used for moderate values of ncp: it will not be highly accurate, especially in the tails, for large values.

Chi-square Probabilities

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

Steps

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.



Chi-square Probabilities

Usage

i `dchisq(x, df, ncp = 0, log = FALSE)`

i `pchisq(q, df, ncp = 0, lower.tail = TRUE, log.p = FALSE)`

i `qchisq(p, df, ncp = 0, lower.tail = TRUE, log.p = FALSE)`

i `rchisq(n, df, ncp = 0)`

Value

`dchisq` gives the density, `pchisq` gives the distribution function, `qchisq` gives the quantile function, and `rchisq` generates random deviates. Invalid arguments will result in return value `NaN`, with a warning.

The length of the result is determined by `n` for `rchisq`, and is the maximum of the lengths of the numerical arguments for the other functions. The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

⚠ Note Supplying `ncp = 0` uses the algorithm for the non-central distribution, which is not the same algorithm used if `ncp` is omitted. This is to give consistent behaviour in extreme cases with values of `ncp` very near zero.

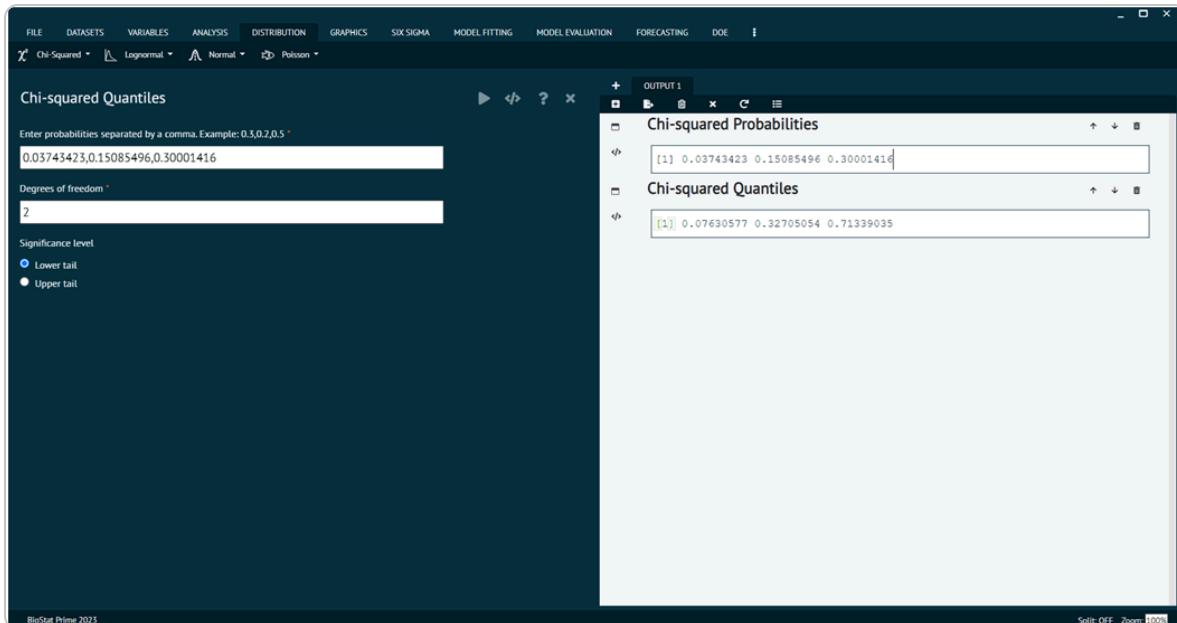
⚠ The code for non-zero `ncp` is principally intended to be used for moderate values of `ncp`: it will not be highly accurate, especially in the tails, for large values.

Chi-square Quantiles

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

Steps

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.



Chi-square Quantiles

Sample from Chi-square Distribution

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

Steps

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. On the left, there is a data grid titled "Chi-Squared Samples" containing 28 rows of numerical data. The columns are labeled "#", "Chi-Squared", and "P-value". The data includes values such as 3.3425, 3.6706, 1.7874, etc. Below the grid are tabs for "DATA" and "VARIABLES". On the right, a dialog box titled "Sample from Chi-squared Distribution" is open. It has fields for "Enter name for dataset" (set to "ChiSquaredSamples"), "Degrees of freedom" (set to "3"), "Number of samples (rows)" (set to "100"), "Number of observations (columns)" (set to "1"), and "Seed" (set to "12345"). Under "Add to dataset", there is a checked checkbox for "Sample means" and two unchecked checkboxes for "Sample sums" and "Sample standard deviations".

Sample from Chi-square Distribution

Sample from Chi-squared Distribution													
<p>We don't calculate sample mean, sum or standard deviation when there is a single row or column</p>													
Samples from Chi-squared Distribution													
<table border="1"><thead><tr><th>obs1</th></tr></thead><tbody><tr><td>sample1 3.3425</td></tr><tr><td>sample2 3.6706</td></tr><tr><td>sample3 1.7874</td></tr><tr><td>sample4 2.0465</td></tr><tr><td>sample5 9.3165</td></tr><tr><td>sample6 2.1787</td></tr><tr><td>sample7 1.4721</td></tr><tr><td>sample8 6.7020</td></tr><tr><td>sample9 1.2030</td></tr><tr><td>sample10 2.8099</td></tr><tr><td>sample11 3.1757</td></tr></tbody></table>		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
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%													

Sample from Chi-square Distribution,output

Lognormal

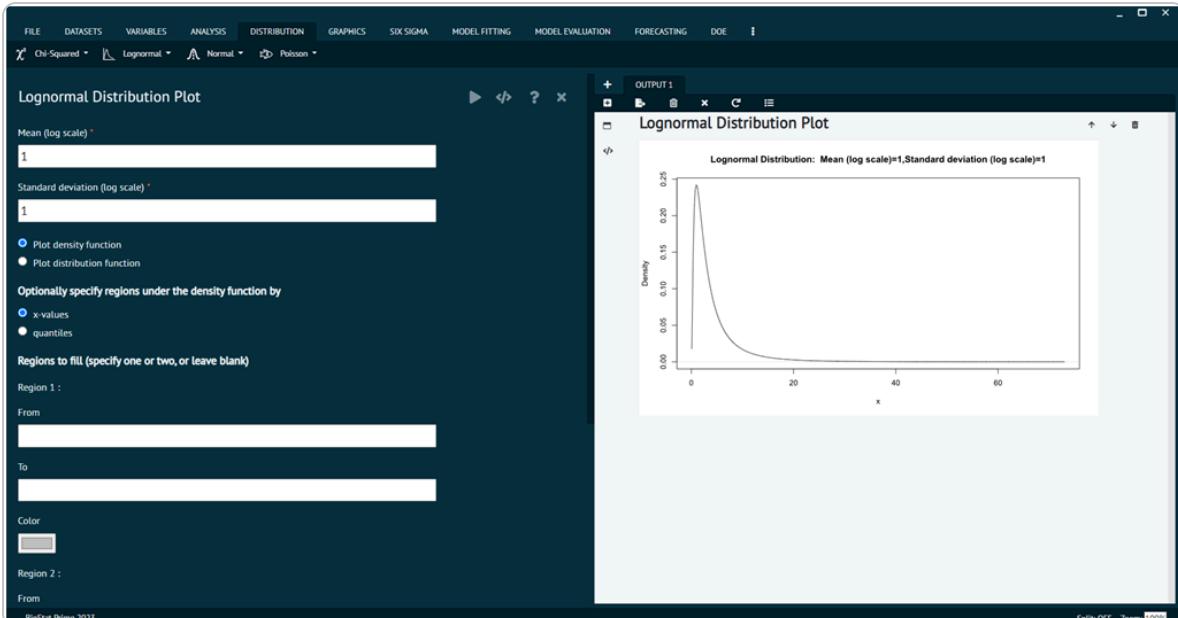
Lognormal Distribution Plot

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.

Steps

Load the dataset -> Click on the Distribution tab in main menu -> Select Lognormal -> Select Lognormal Distribution-> 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.



Lognormal Distribution Plot

Density, distribution function, quantile function and random generation for the log normal distribution whose logarithm has mean equal to `meanlog` and standard deviation equal to `sdlog`.

Usage

i `dlnorm(x, meanlog = 0, sdlog = 1, log = FALSE)`

i `plnorm(q, meanlog = 0, sdlog = 1, lower.tail = TRUE, log.p = FALSE)`

i `qlnorm(p, meanlog = 0, sdlog = 1, lower.tail = TRUE, log.p = FALSE)`

i `rlnorm(n, meanlog = 0, sdlog = 1)`

Value

`dlnorm` gives the density, `plnorm` gives the distribution function, `qlnorm` gives the quantile function, and `rlnorm` generates random deviates.

The length of the result is determined by `n` for `rlnorm`, and is the maximum of the lengths of the numerical arguments for the other functions.

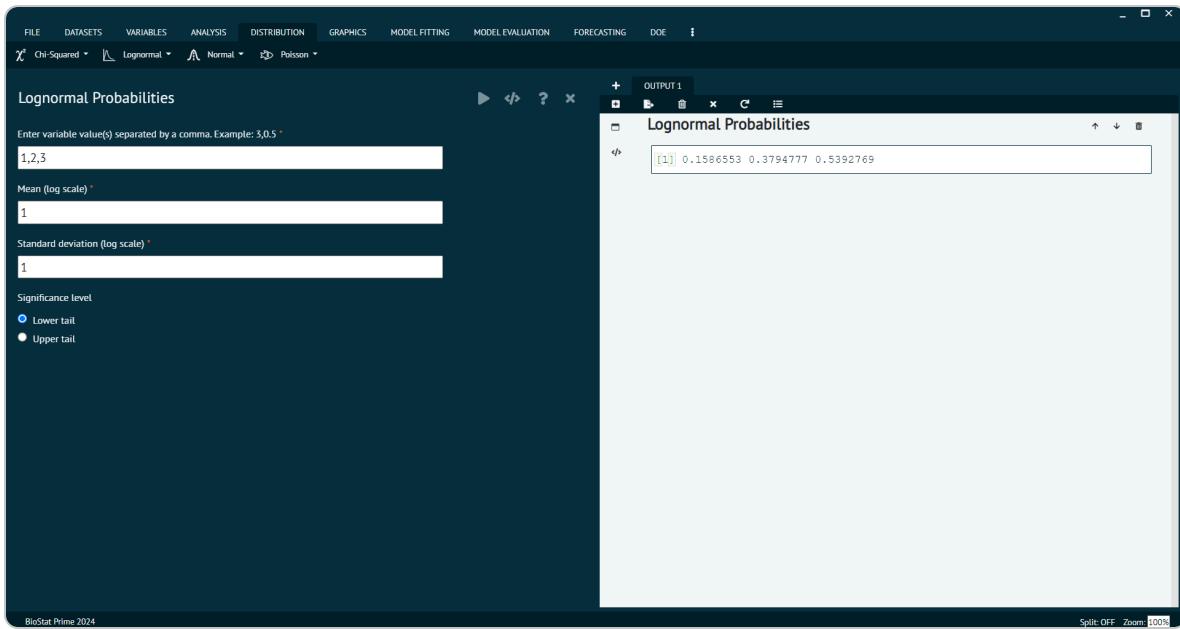
The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

Lognormal Probabilities

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

Steps

Load the dataset -> Click on the Distribution tab in main menu -> Select Lognormal -> Select Lognormal Probabilities -> 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.



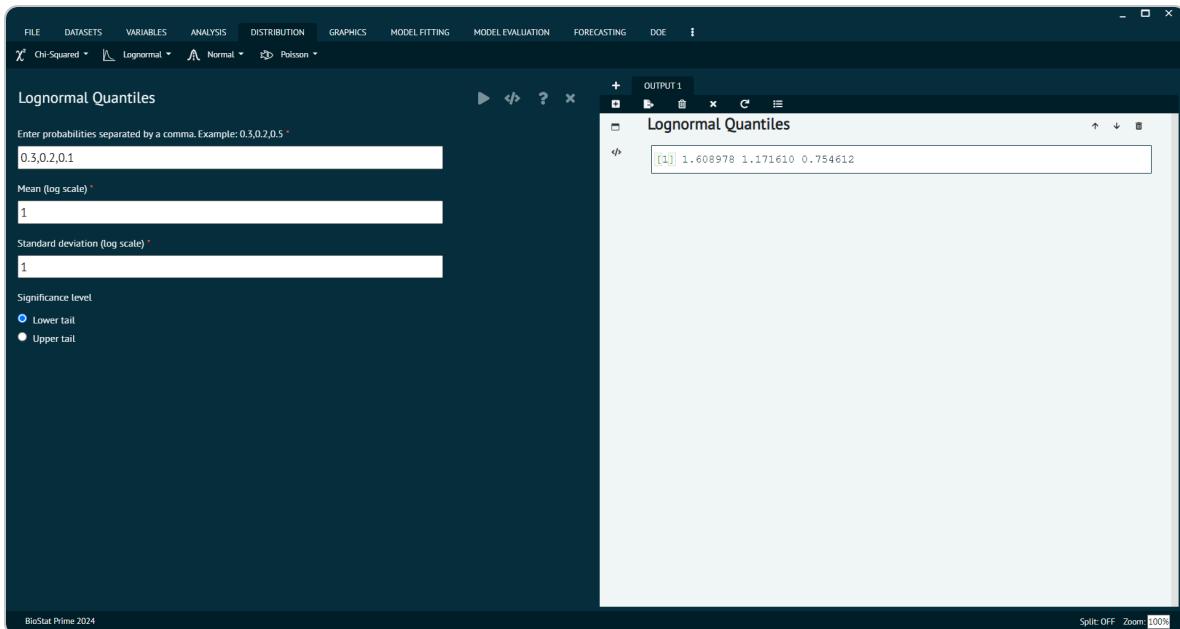
Lognormal Probabilities

Lognormal Quantiles

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

Steps

Load the dataset -> Click on the Distribution tab in main menu -> Select Lognormal -> Select Lognormal Quantiles -> 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.



Lognormal Quantiles

Sample from Lognormal Distribution

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

Steps

Load the dataset -> Click on the Distribution tab in main menu -> Select Lognormal -> Select Sample from Lognormal Distribution -> 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.

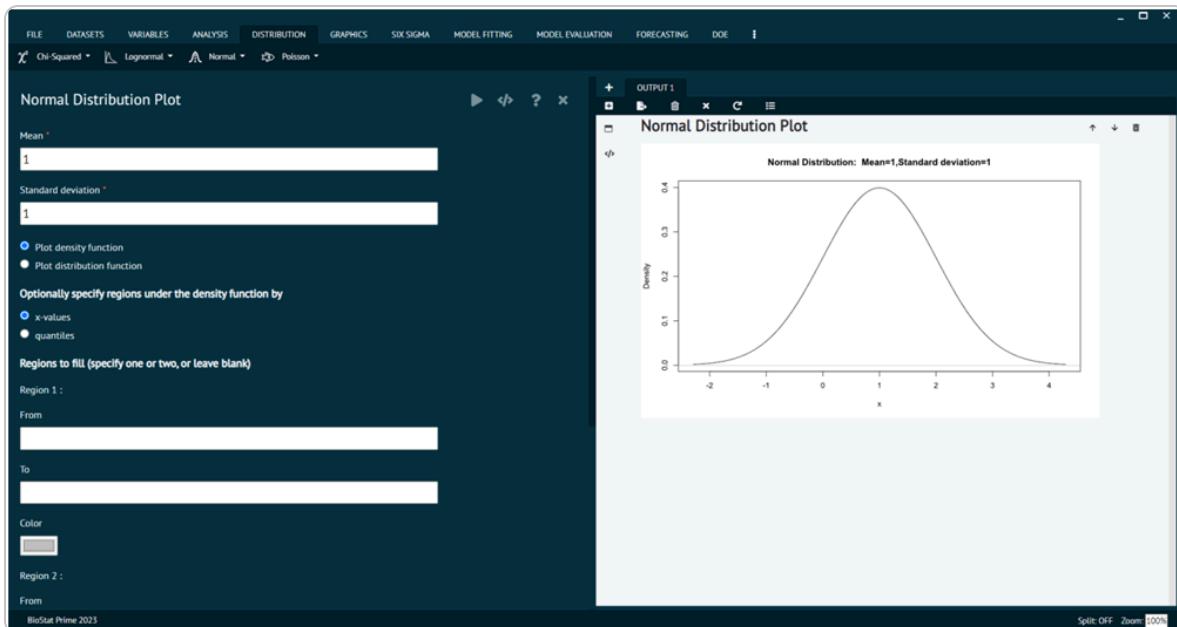
The screenshot shows the BioStat Prime 2024 software interface. On the left, the main workspace displays a dataset named 'Dataset1' with 29 rows of data. The columns are labeled '#', 'obs1', and 'obs2'. The data includes numerical values such as 4.8819, 5.5260, 2.4368, etc. Below this is an 'R EDITOR' section. On the right, an 'OUTPUT' window titled 'Sample from Lognormal Distribution' is open. 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 Lognormal Distribution' lists 11 rows, each labeled 'sample1' through 'sample11', corresponding to the values in the 'obs1' column of the dataset. The bottom right corner of the output window shows 'Split OFF' and 'Zoom 100%'. The top menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, MODEL FITTING, MODEL EVALUATION, FORECASTING, DOE, and HELP.

Sample from Lognormal Distribution

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.

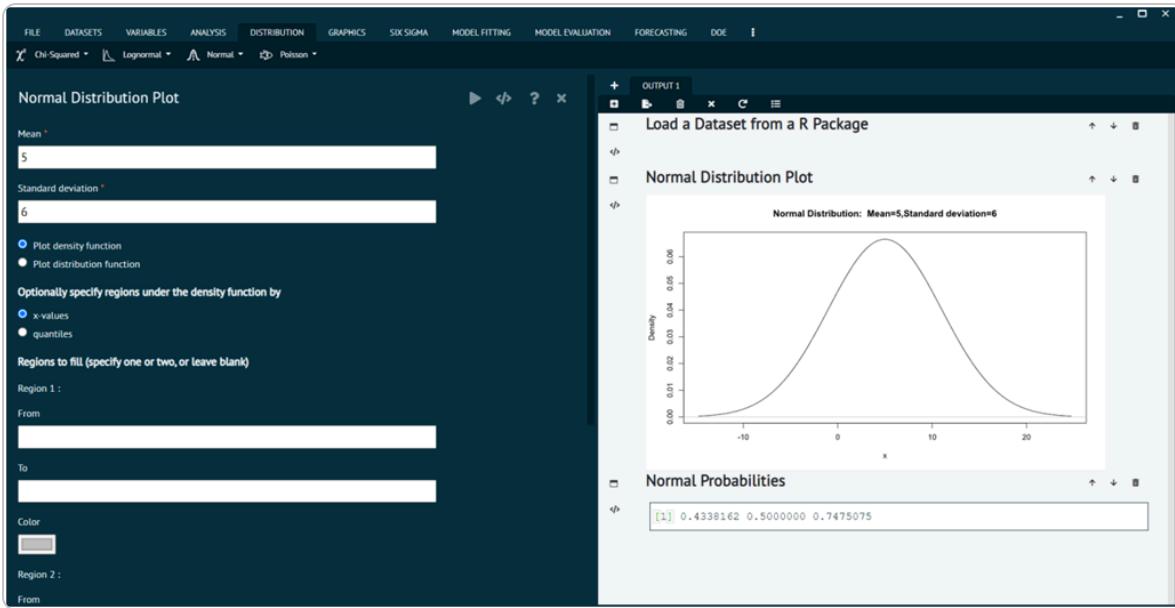
Many natural phenomena, such as heights, weight, and IQ scores, tend to follow a normal distribution. The normal distribution is characterized by two parameters: the mean (μ) and the standard deviation (σ). The mean determines the center of the distribution, while the standard deviation determines the spread or dispersion of the data points around the mean.



Normal

Normal Probabilities

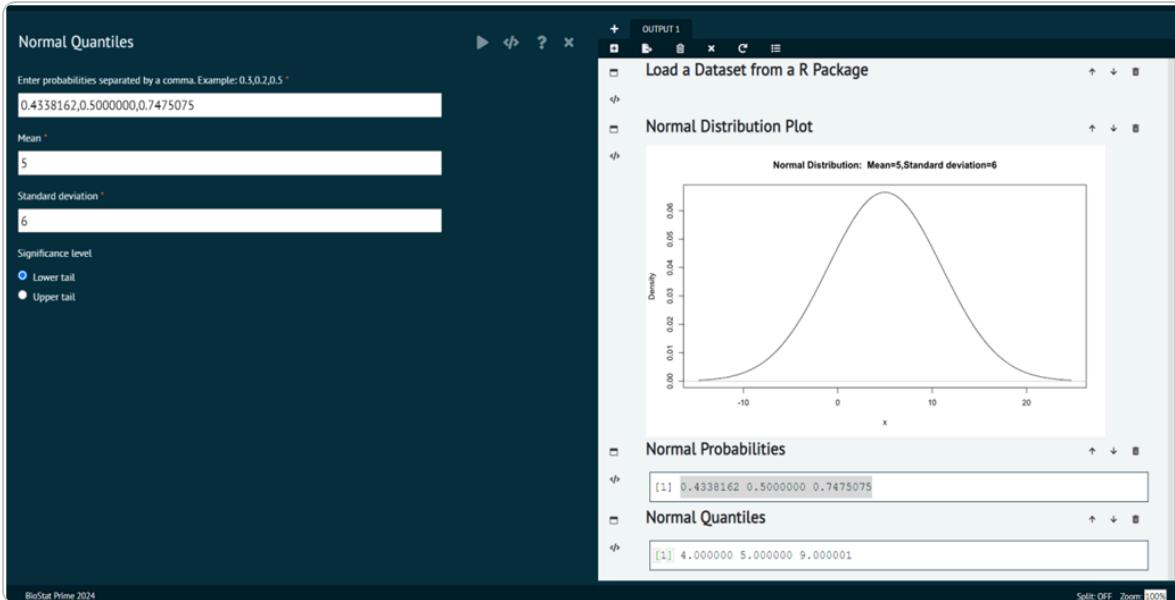
Normal probabilities refer to the probabilities associated with the normal distribution. These probabilities describe the likelihood of observing certain values or ranges of values within a normal distribution.



Normal Probabilities

Normal Quantiles

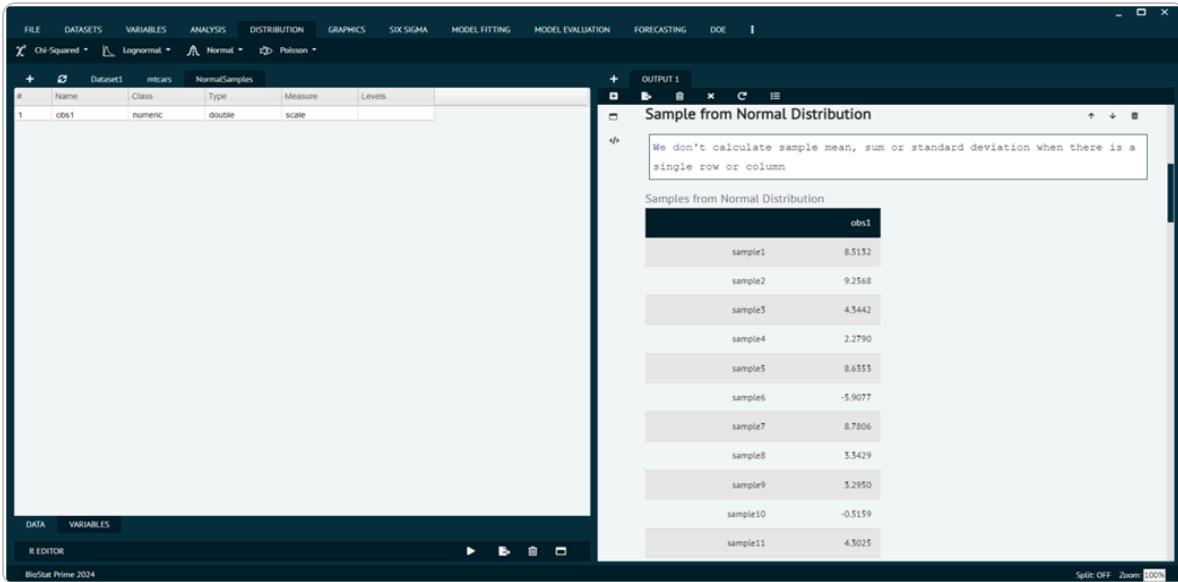
Normal Quantiles refer to the values that divide a normal distribution into intervals with equal probabilities. These Quantiles are often used in statistical analysis for constructing confidence intervals, hypothesis testing, and understanding the distribution of data.



Normal Quantiles

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.



Sample from Normal Distribution

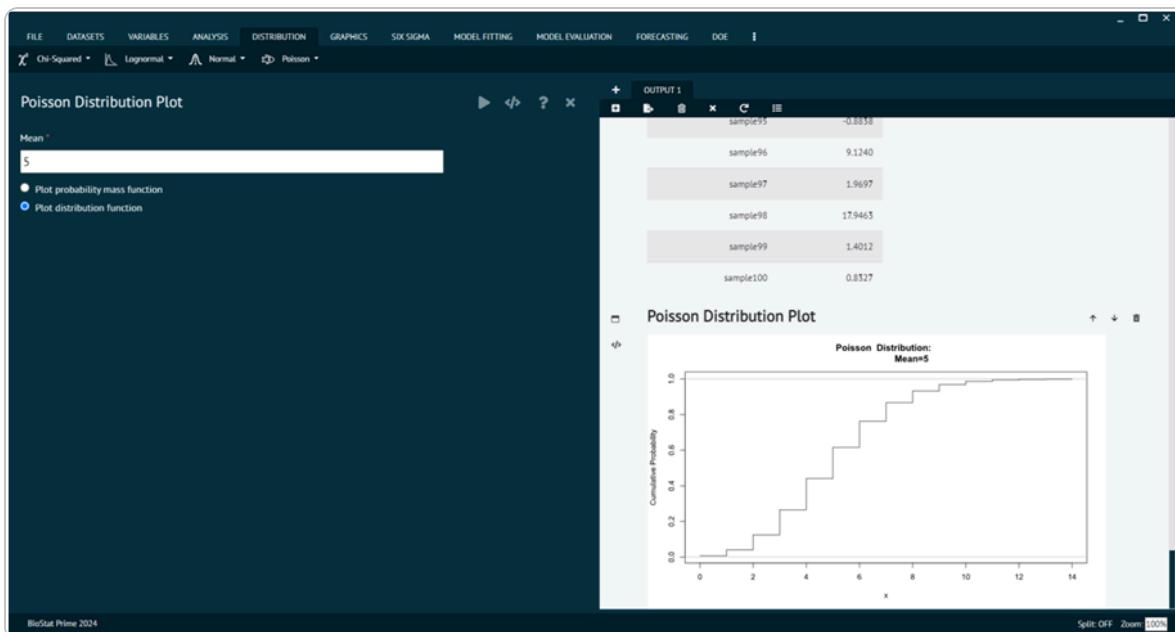
Poisson

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.

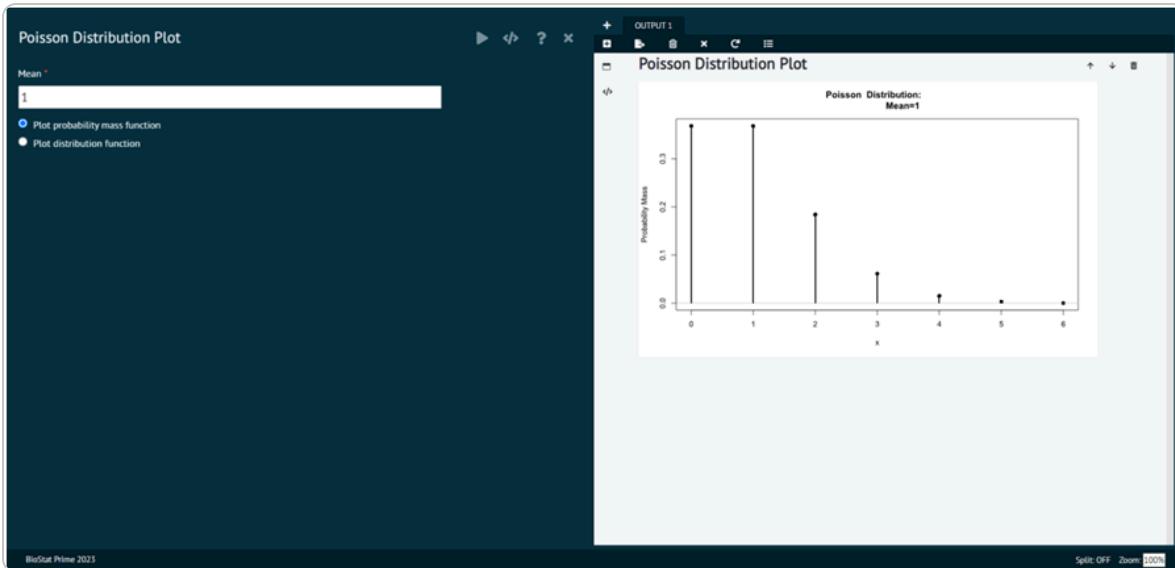
Poisson Distribution Plot

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



Poisson Distribution Plot

Plot probability mass function



Plot probability mass function

Usage

i `dpois(x, lambda, log = FALSE)`

i `ppois(q, lambda, lower.tail = TRUE, log.p = FALSE)`

i `qpois(p, lambda, lower.tail = TRUE, log.p = FALSE)`

i `rpois(n, lambda)`

Details

The Poisson distribution has density $p(x) = \lambda^x \exp(-\lambda)/x!$ for $x = 0, 1, 2, \dots$. The mean and variance are $E(X) = \text{Var}(X) = \lambda$.

! Note that $\lambda = 0$ is really a limit case (setting $0^0 = 1$) resulting in a point mass at 0, see also the example.

If an element of x is not integer, the result of $dpois$ is zero, with a warning. $p(x)$ is computed using Loader's algorithm, see the reference in $dbinom$.

The quantile is right continuous: $qpois(p, \lambda)$ is the smallest integer x such that $P(X \leq x) \geq p$.

Setting `lower.tail = FALSE` allows to get much more precise results when the default, `lower.tail = TRUE` would return 1, see the example below.

Value

`dpois` gives the (log) density, `ppois` gives the (log) distribution function, `qpois` gives the quantile function, and `rpois` generates random deviates.

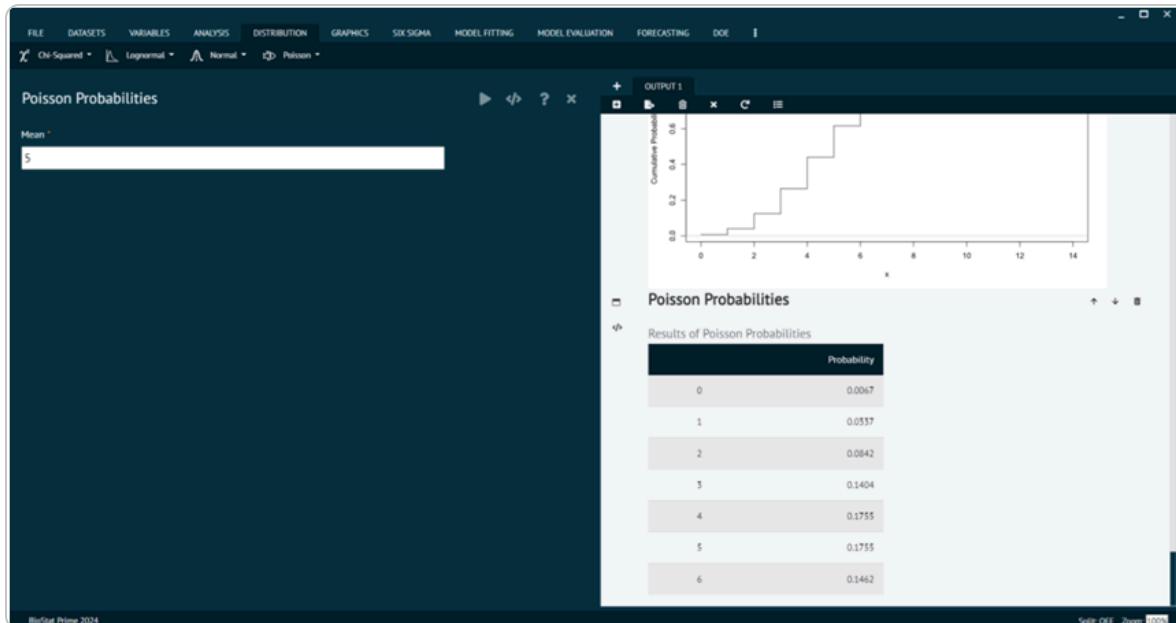
Invalid λ will result in return value `NaN`, with a warning.

The length of the result is determined by n for `rpois`, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than n are recycled to the length of the result. Only the first elements of the logical arguments are used.

Poisson Probabilities

Poisson's probabilities refer to the probabilities associated with the Poisson distribution.



Poisson Probabilities

Details

The Poisson distribution has density $p(x) = \lambda^x \exp(-\lambda)/x!$ for $x = 0, 1, 2, \dots$. The mean and variance are $E(X) = \text{Var}(X) = \lambda$.

⚠ Note that $\lambda = 0$ is really a limit case (setting $0^0 = 1$) resulting in a point mass at 0, see also the example.

If an element of x is not integer, the result of `dpois` is zero, with a warning. $p(x)$ is computed using Loader's algorithm, see the reference in `dbinom`.

The quantile is right continuous: `qpois(p, lambda)` is the smallest integer x such that $P(X \leq x) \geq p$.

Setting `lower.tail = FALSE` allows to get much more precise results when the default, `lower.tail = TRUE` would return 1, see the example below.

Value

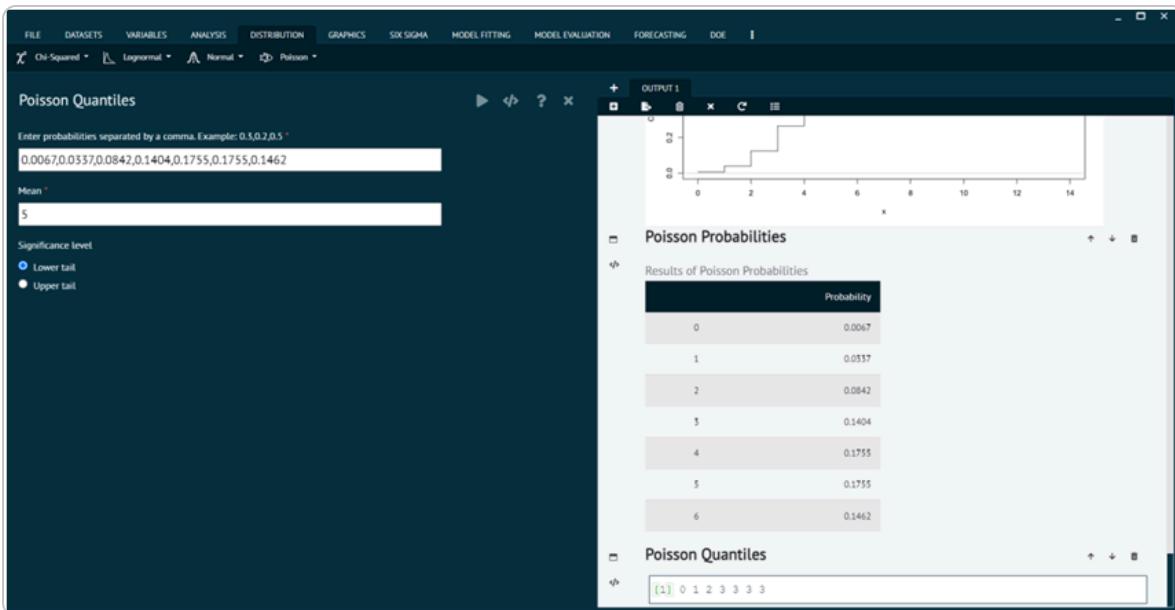
`dpois` gives the (log) density, `ppois` gives the (log) distribution function, `qpois` gives the quantile function, and `rpois` generates random deviates.

Invalid `lambda` will result in return value `NaN`, with a warning.

The length of the result is determined by `n` for `rpois`, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

Poisson Quantiles



Poisson Quantiles

Details

The Poisson distribution has density $p(x) = \lambda^x \exp(-\lambda)/x!$ for $x = 0, 1, 2, \dots$. The mean and variance are $E(X) = \text{Var}(X) = \lambda$.

⚠ Note that $\lambda = 0$ is really a limit case (setting $0^0 = 1$) resulting in a point mass at 0, see also the example.

If an element of x is not integer, the result of `dpois` is zero, with a warning. $p(x)$ is computed using Loader's algorithm, see the reference in `dbinom`.

The quantile is right continuous: `qpois(p, lambda)` is the smallest integer x such that $P(X \leq x) \geq p$.

Setting `lower.tail = FALSE` allows to get much more precise results when the default, `lower.tail = TRUE` would return 1, see the example below.

Value

`dpois` gives the (log) density, `ppois` gives the (log) distribution function, `qpois` gives the quantile function, and `rpois` generates random deviates.

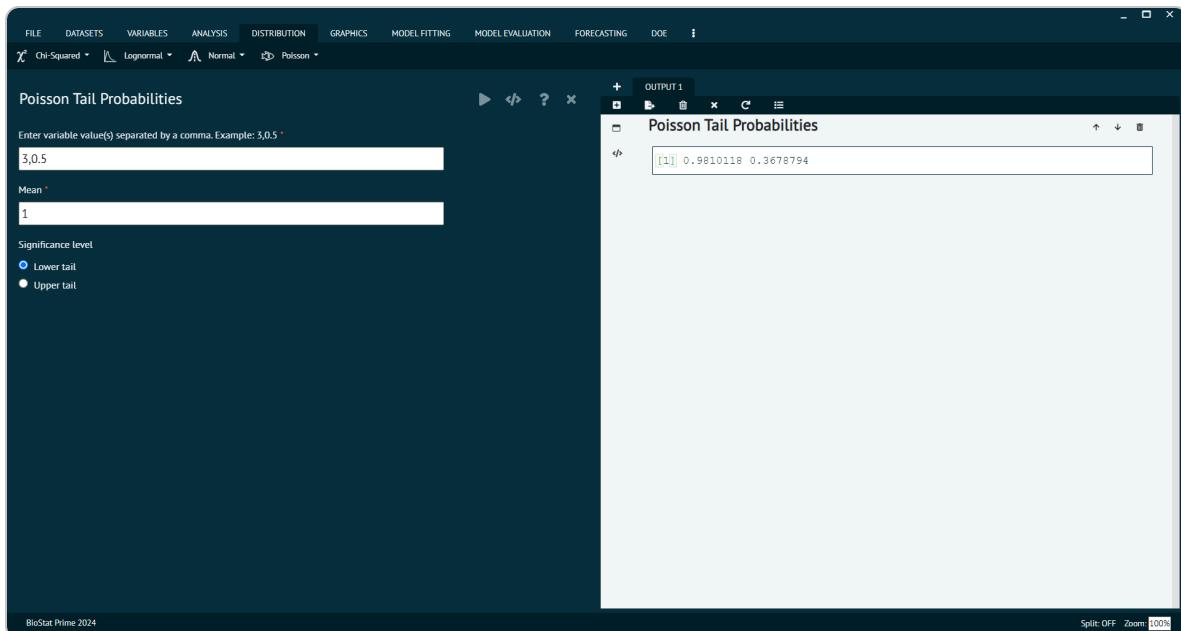
Invalid `lambda` will result in return value `NaN`, with a warning.

The length of the result is determined by `n` for `rpois`, and is the maximum of the lengths

of the numerical arguments for the other functions.

The numerical arguments other than n are recycled to the length of the result. Only the first elements of the logical arguments are used.

Poisson Tail Probabilities



Poisson Tail Probabilities

Usage

i `dpois(x, lambda, log = FALSE)`

i `ppois(q, lambda, lower.tail = TRUE, log.p = FALSE)`

i `qpois(p, lambda, lower.tail = TRUE, log.p = FALSE)`

i `rpois(n, lambda)`

Details

The Poisson distribution has density $p(x) = \lambda^x \exp(-\lambda)/x!$ for $x = 0, 1, 2, \dots$. The mean and variance are $E(X) = \text{Var}(X) = \lambda$.

⚠ Note that $\lambda = 0$ is really a limit case (setting $0^0 = 1$) resulting in a point mass at 0, see also the example.

If an element of x is not integer, the result of `dpois` is zero, with a warning. $p(x)$ is computed using Loader's algorithm, see the reference in `dbinom`.

The quantile is right continuous: `qpois(p, lambda)` is the smallest integer x such that $P(X \leq x) \geq p$.

Setting `lower.tail = FALSE` allows to get much more precise results when the default, `lower.tail = TRUE` would return 1, see the example below.

Value

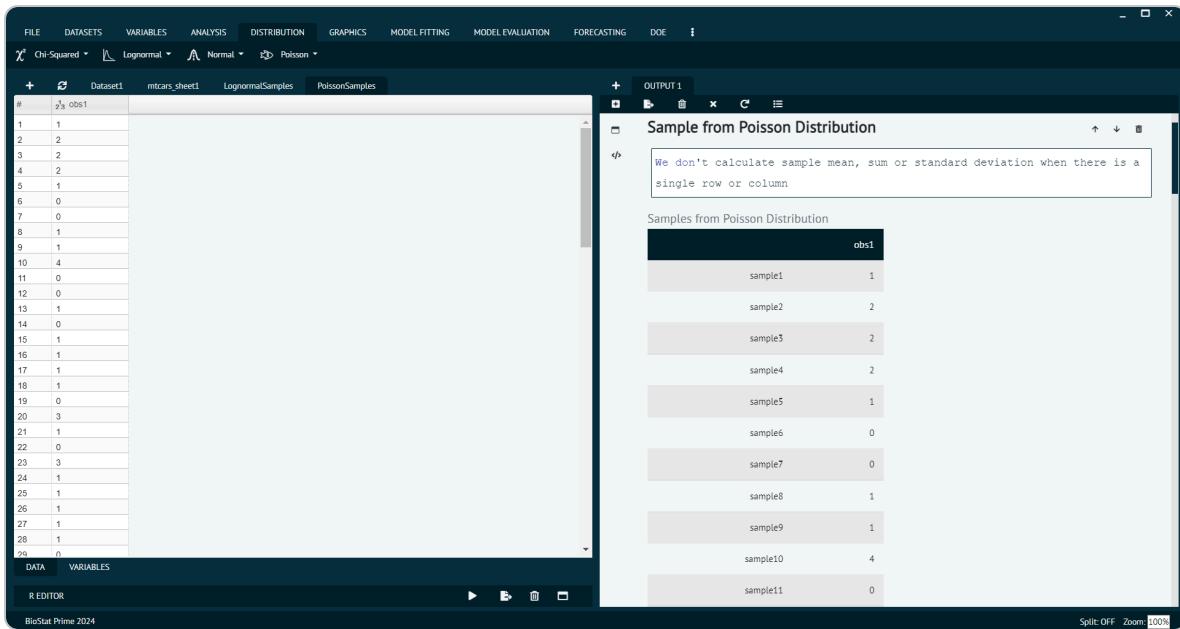
`dpois` gives the (log) density, `ppois` gives the (log) distribution function, `qpois` gives the quantile function, and `rpois` generates random deviates.

Invalid `lambda` will result in return value `NaN`, with a warning.

The length of the result is determined by `n` for `rpois`, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

Sample from Poisson Distribution

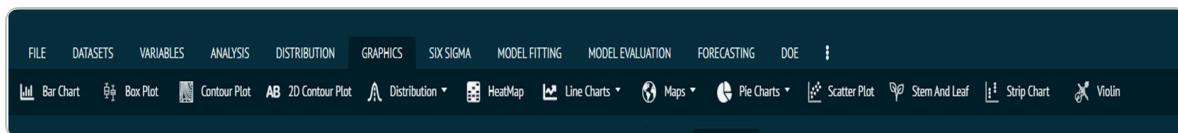


Sample from Poisson Distribution

Graphs and Charts

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.



Graphs and Charts

Bar Chart

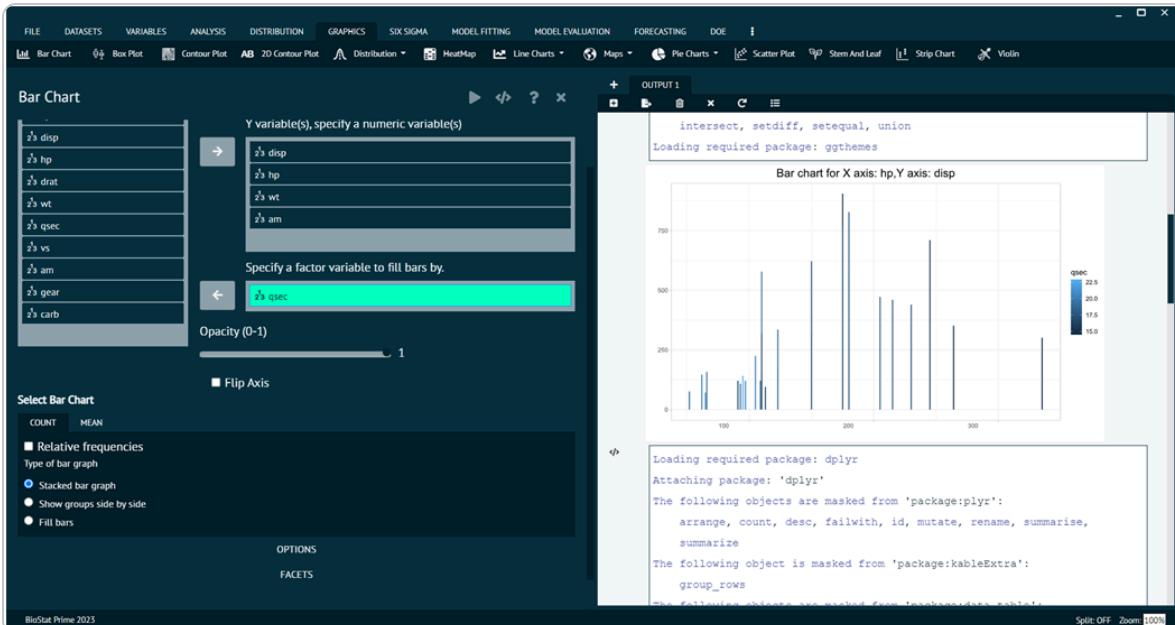
For representing any dataset in terms of Bar Chart.

Steps

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.

- i** 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.



Bar Chart

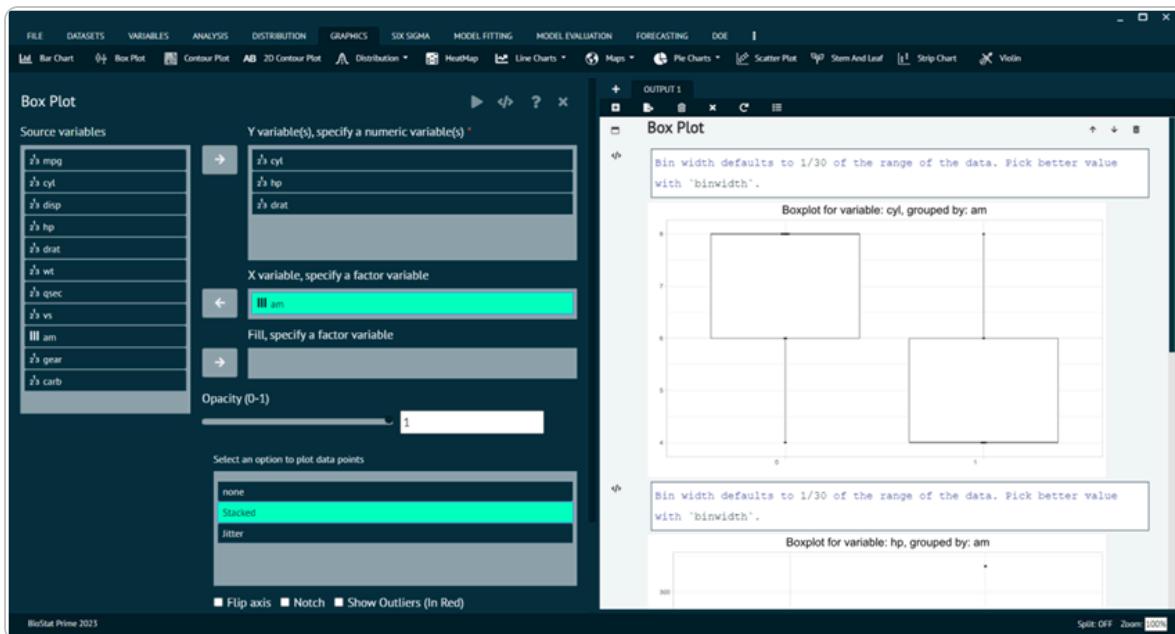
Box Plot

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.



Box Plot

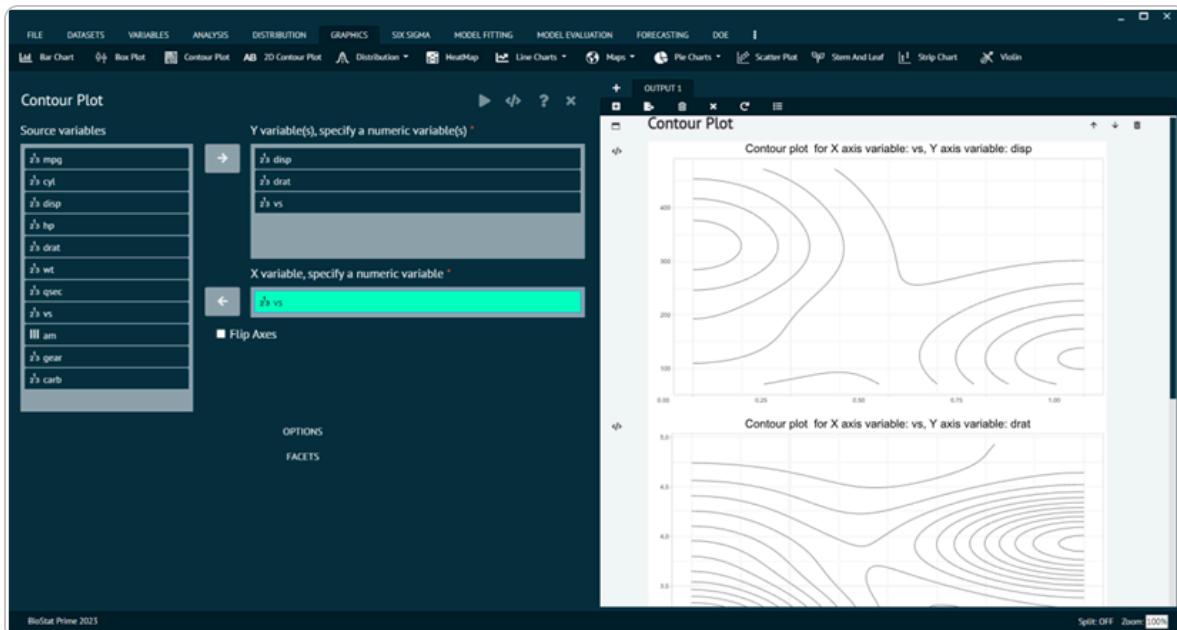
Contour Plot

For representing any dataset in terms of Contour Plot.

Steps

Load the dataset that needs to be visualized -> Go to Graphics -> Contour Plot ->

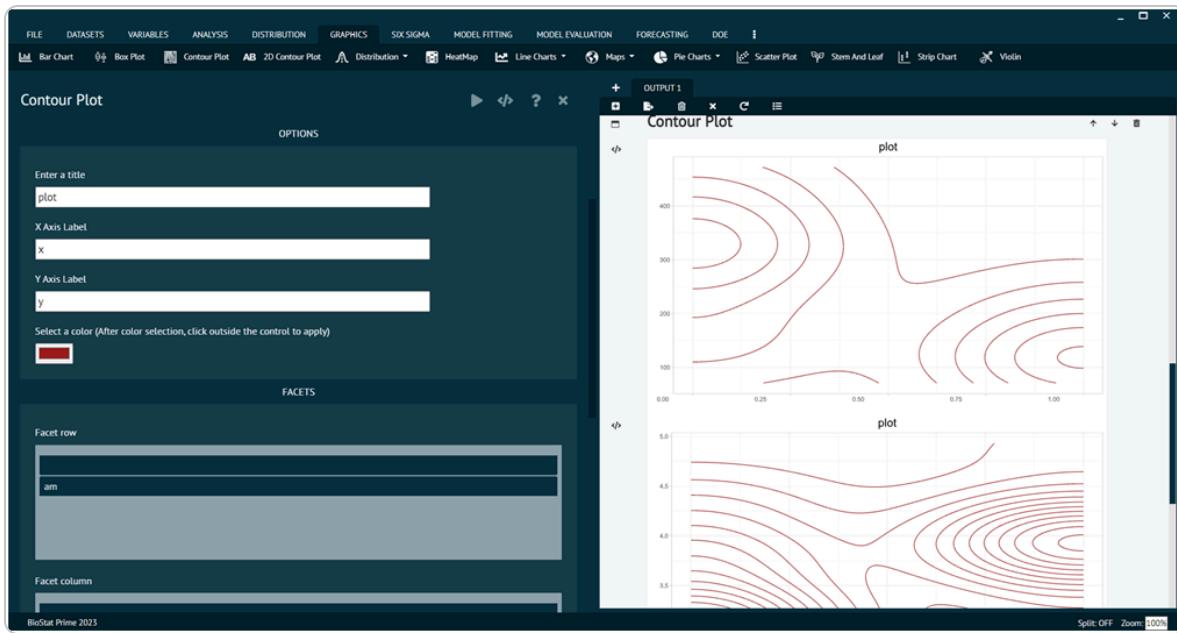
Put in the values for variables -> Choose additional options (like opacity, data points, flip axis, etc.) as per the user's requirement -> Execute the dialog.



Contour Plot

- 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.



Contour Plot

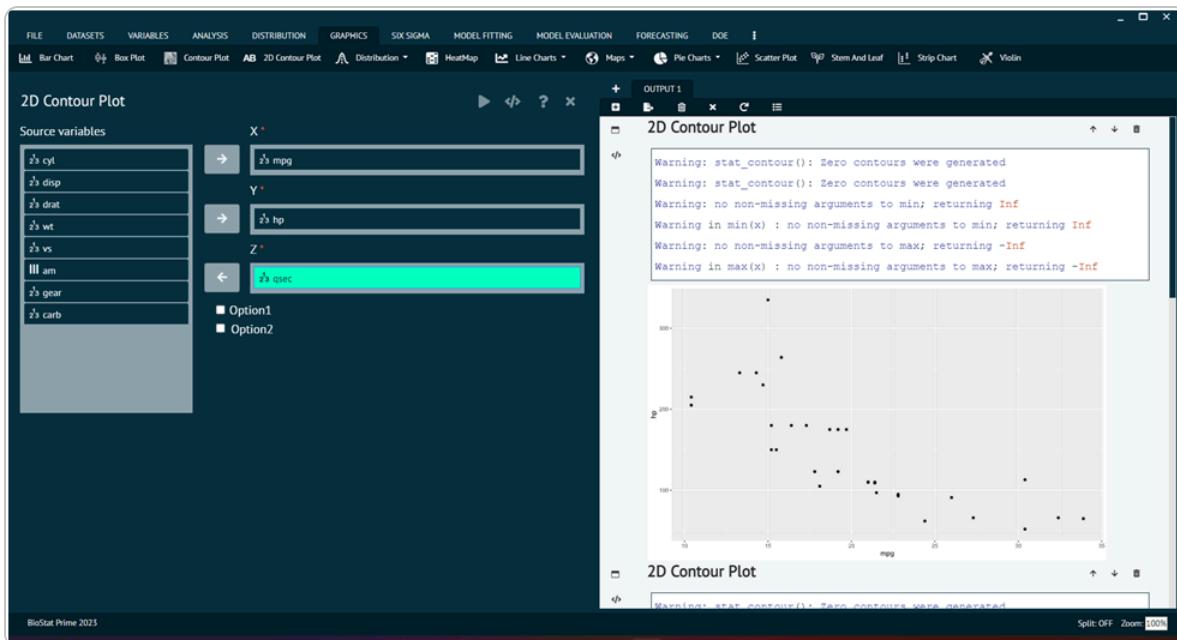
AB 2D Contour Plot

For representing any dataset in terms of AB 2D Contour Plot.

Steps

Load the dataset that needs to be visualized -> Go to Graphics -> AB 2D Contour Plot -> Put in the values for variables -> Execute the dialog.

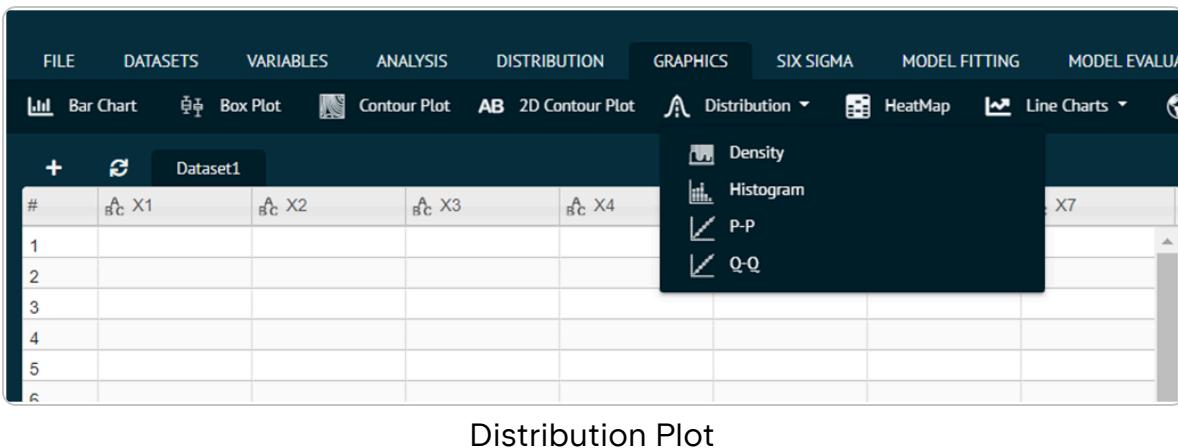
The output of the 2D Contour Plot of a sample dataset can be seen in the picture below.



AB 2D Contour Plot

Distribution Plot

The distribution tab of graphics menu contains 4 options of data visualization i.e., Density, Histogram, P-P plot, Q-Q plot.



The function of each option is discussed below.

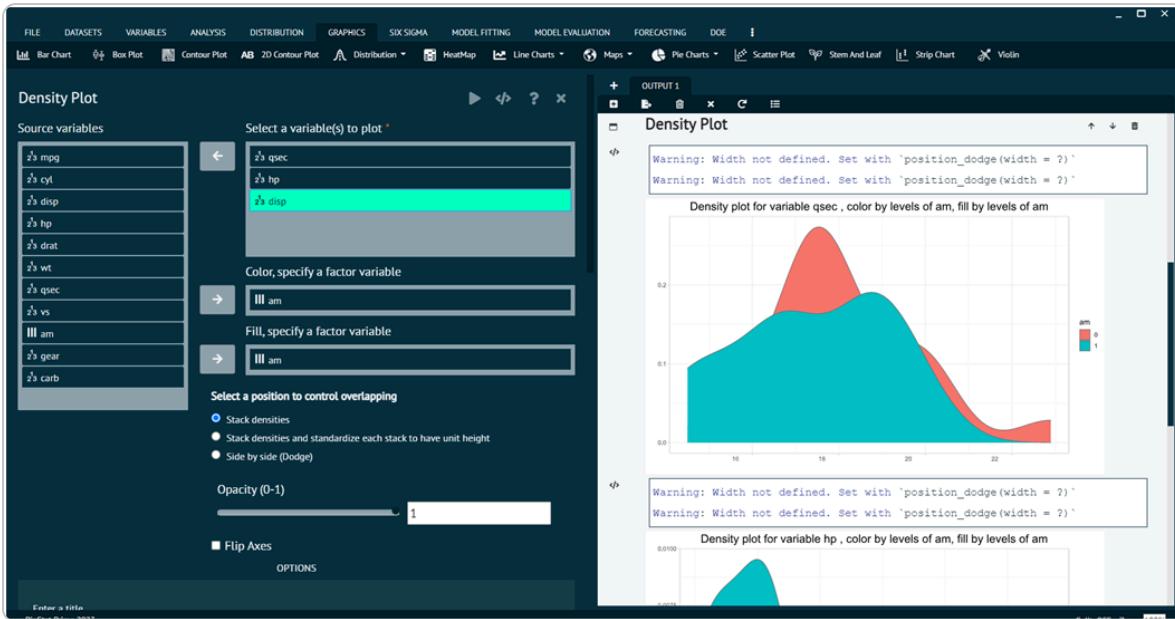
Density

For representing any dataset in terms of Density plot.

Steps

Load the dataset that needs to be visualized -> Go to Graphics -> Distribution -> Density -> Put in the values for variables -> Execute the dialog.

The output of the Density Plot of a sample dataset can be seen in the picture below.



Density

i The Options tab and Facets tab at the bottom can be utilized to add more features to the output.

i User can also select the position to control overlapping, flip axes and opacity of the output.

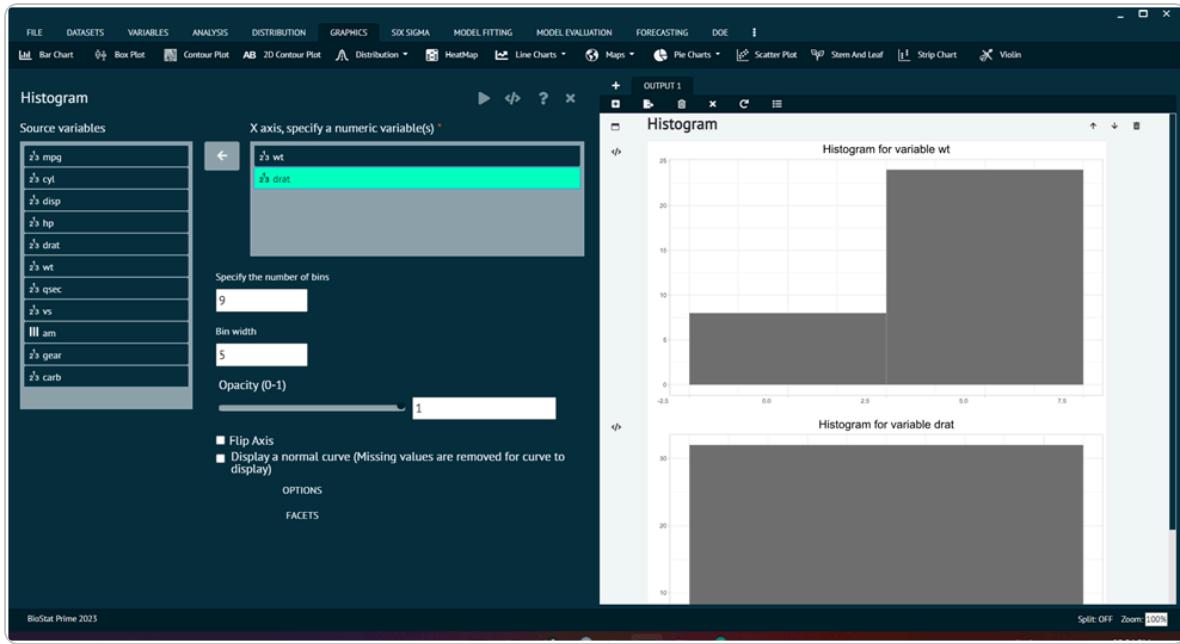
Histogram

For representing any dataset in terms of Histogram.

Steps

Load the dataset that needs to be visualized -> Go to Graphics -> Distribution -> Histogram -> Put in the values for variables -> Execute the dialog.

The output of the Histogram of a sample dataset can be seen in the picture below.



Histogram

- i** The Options tab and Facets tab at the bottom can be utilized to add more features to the output.
- i** User can also control opacity, flip axes and display normal curve of the output.

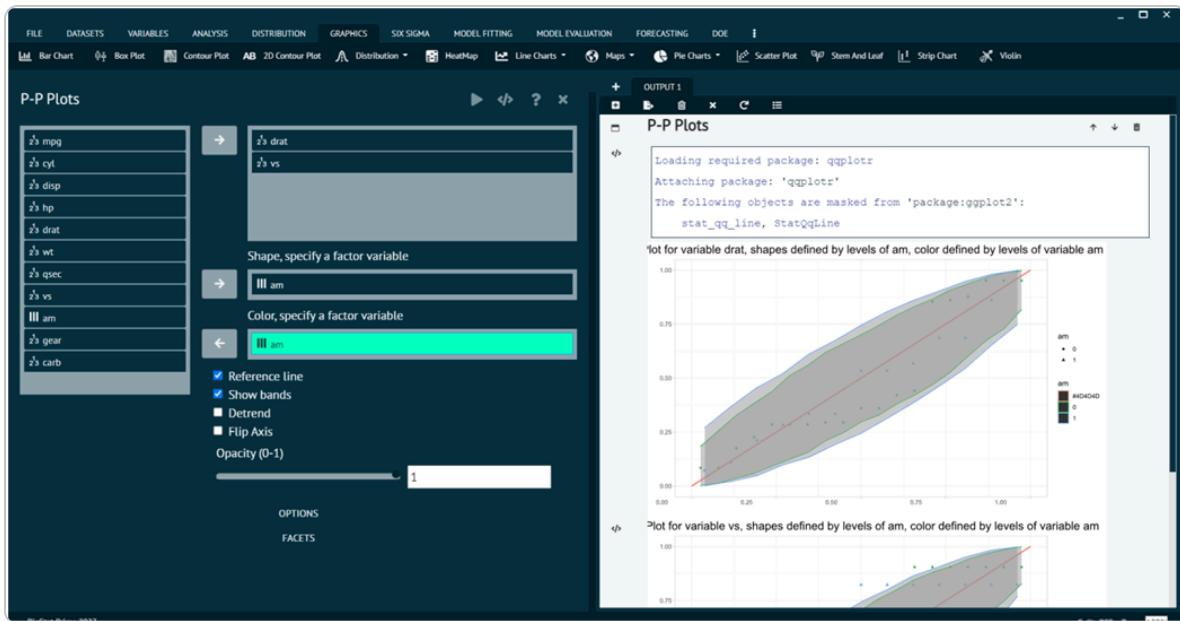
PP Plot

For representing any dataset in terms of PP Plot.

Steps

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.



PP Plot

i The Options tab and Facets tab at the bottom can be utilized to add more features to the output.

i User can also control opacity, flip axes and display reference line or bands or detrend in the output.

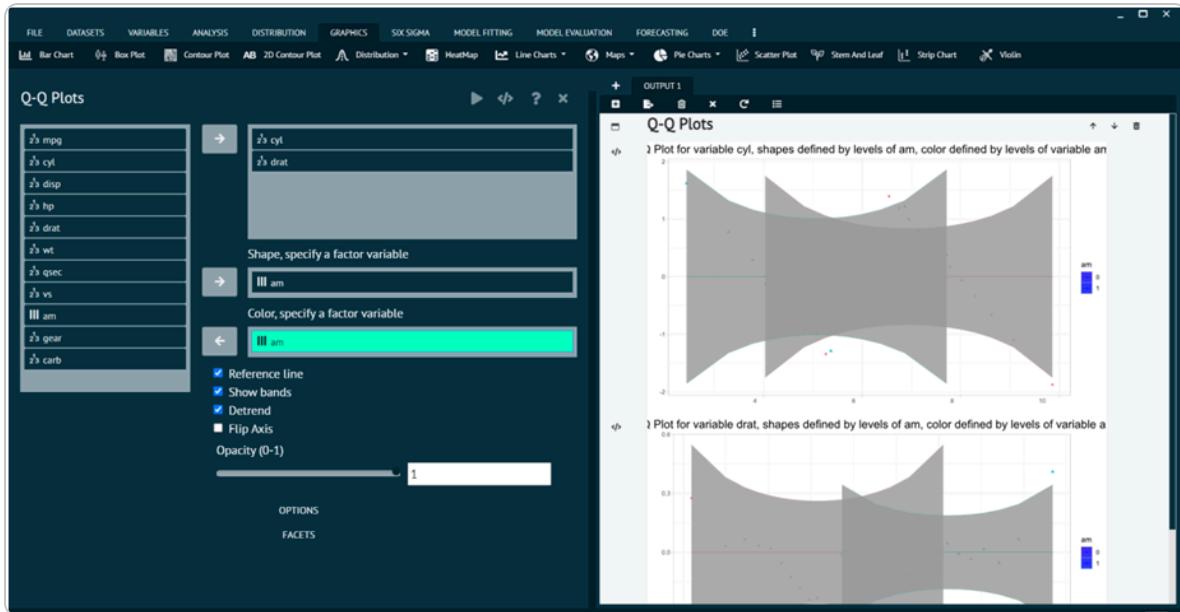
QQ Plot

For representing any dataset in terms of QQPlot.

Steps

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.



QQ Plot

- i** The Options tab and Facets tab at the bottom can be utilized to add more features to the output.

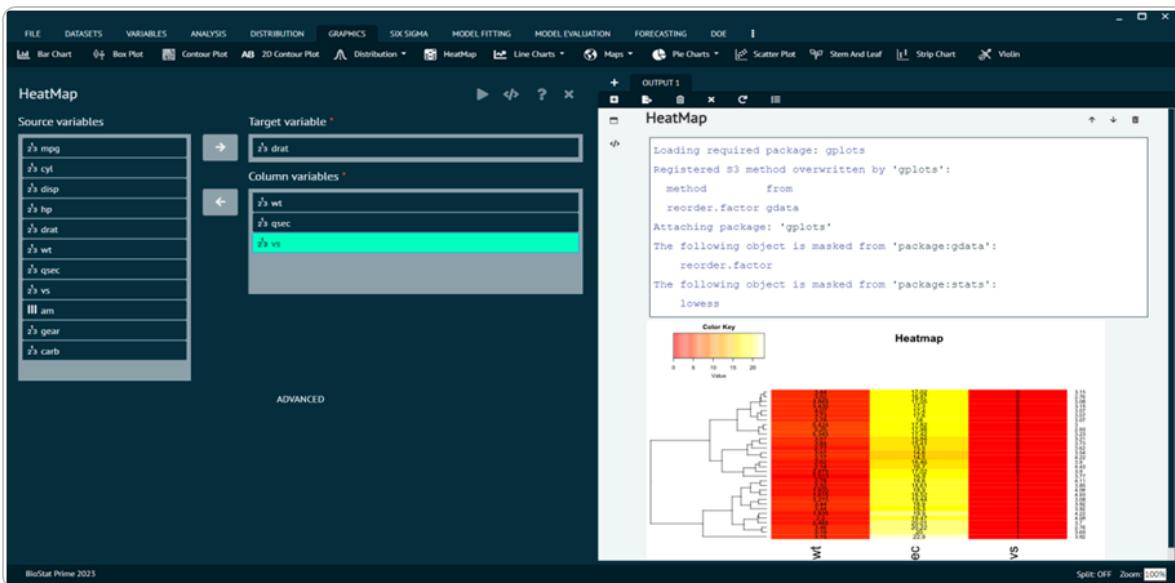
- i** User can also control opacity, flip axes and display reference line or bands or detrend in the output.

HeatMap

For representing any dataset in terms of HeatMap

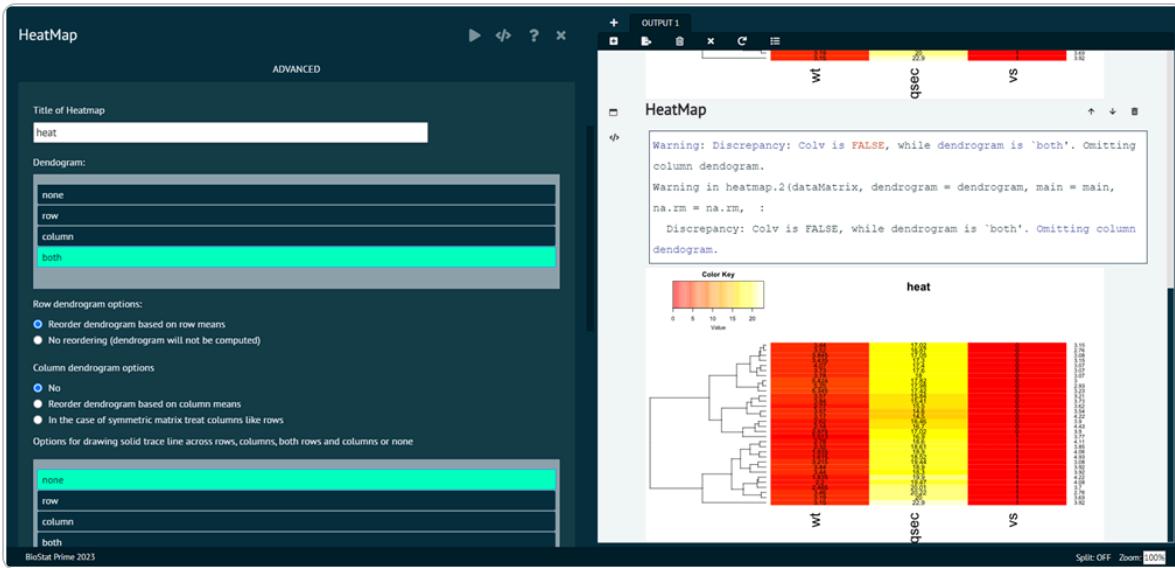
Steps

Load the dataset that needs to be visualized -> Go to Graphics -> HeatMap -> Put in the values for variables -> Execute the dialog.



HeatMap

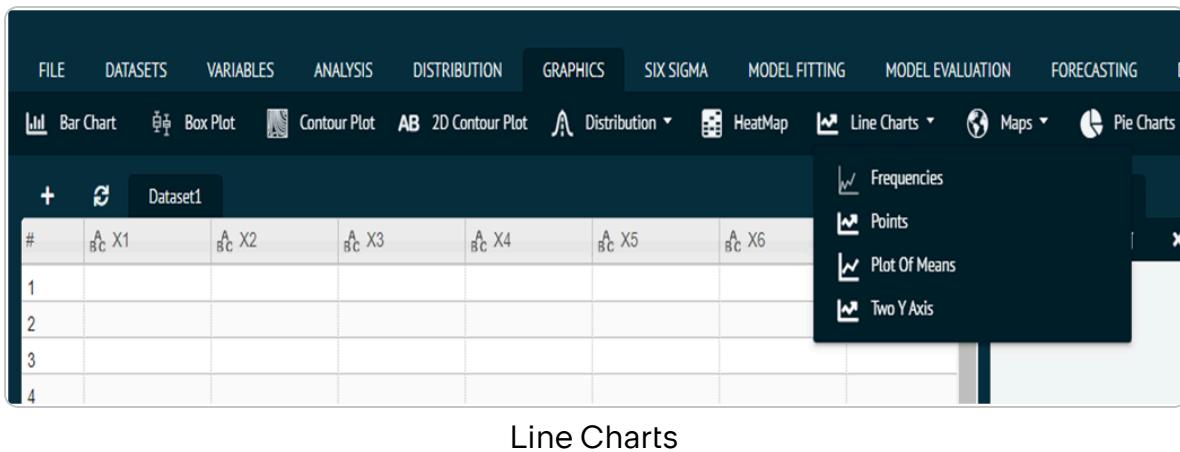
The advanced tab at the bottom leads to some advanced features of the as shown in the picture below.



HeatMap

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.



Line Charts

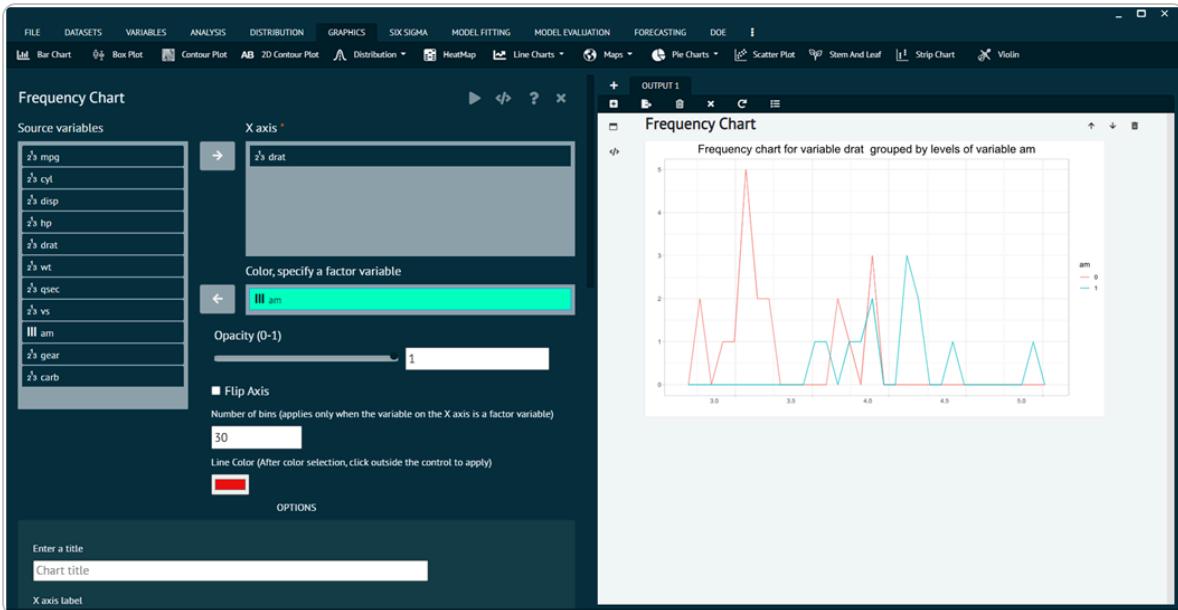
Frequency Chart

For representing any dataset in terms of Frequency Chart.

Steps

Load the dataset that needs to be visualized -> Go to Graphics -> Line Chart
 Frequencies -> Put in the values for variables -> Execute the dialog.

The output of the Frequency Chart of a sample dataset can be seen in the picture below.



Frequency Chart

- i** The Options tab and Facets tab at the bottom can be utilized to add more features to the output.
- i** User can also control opacity, flip axes, no. of bins and line colour of the output.

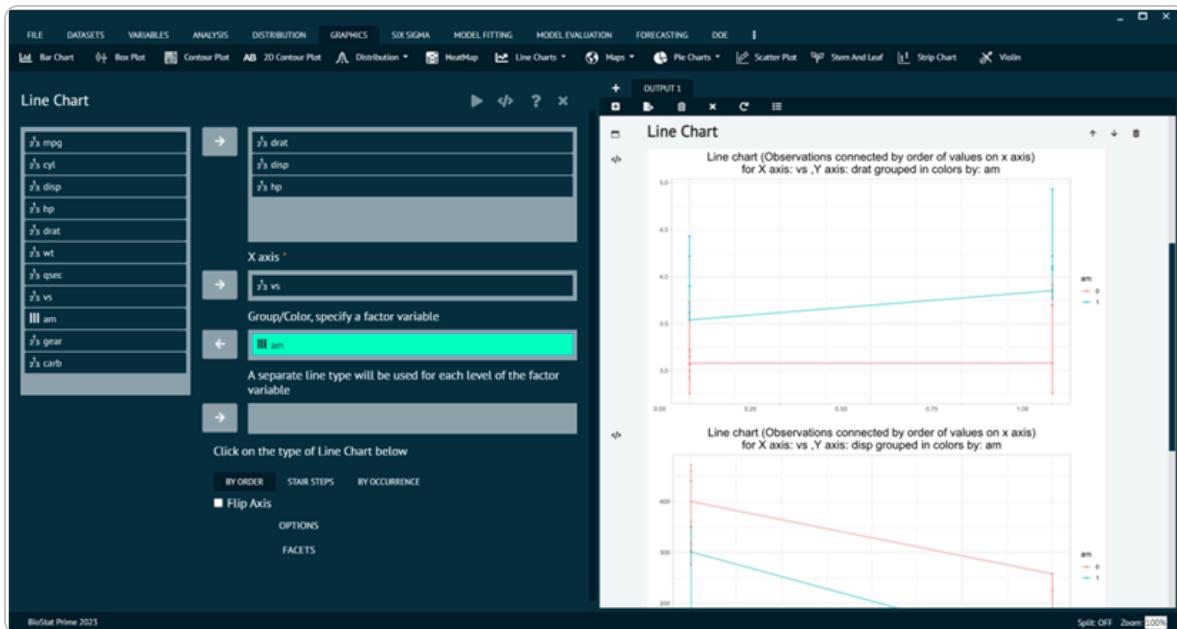
Line Chart

For representing any dataset in terms of Line Chart.

Steps

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.



Line Chart

- i** The Options tab and Facets tab at the bottom can be utilized to add more features to the output.
- i** User can also flip axes, type of line chart in the output.

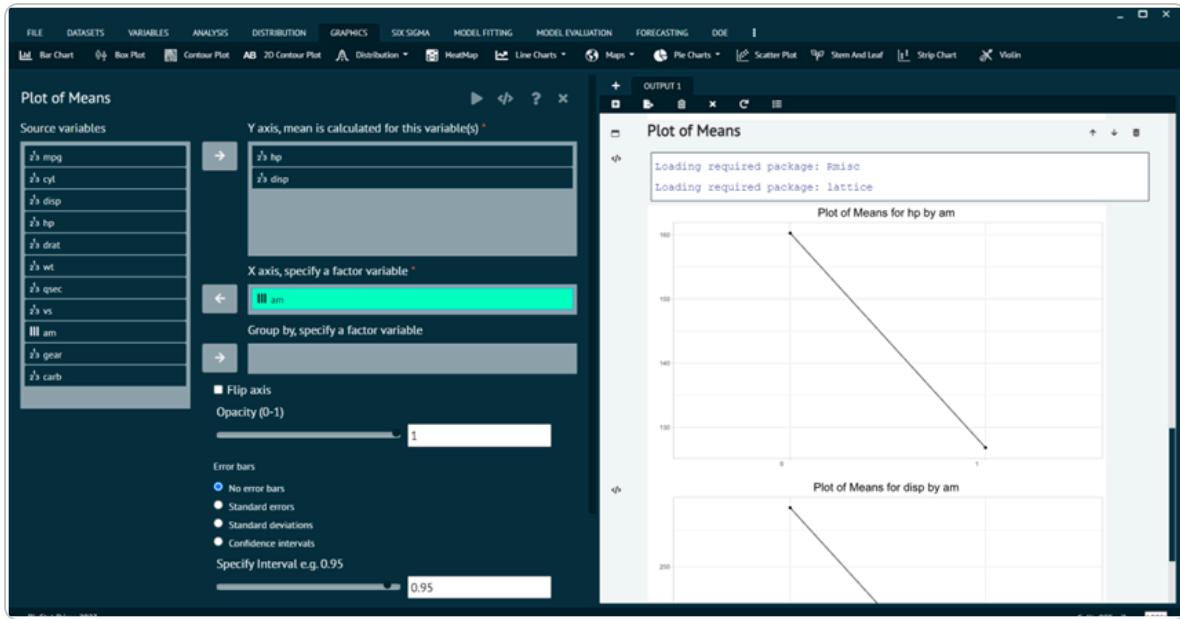
Plot of Means

For representing any dataset in terms of Plot of Means.

Steps

Load the dataset that needs to be visualized -> Go to Graphics -> Line Chart -> Plot of Means -> 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.



Plot of Means

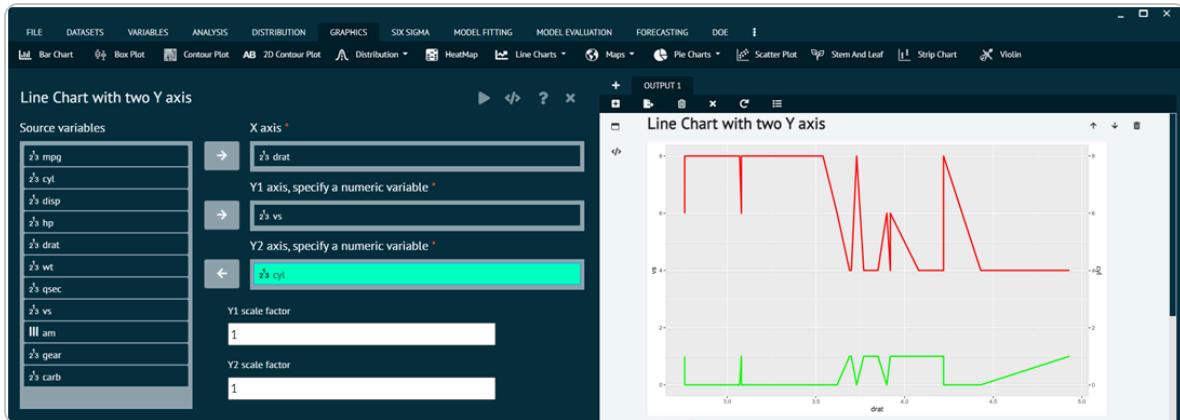
- i** User can also flip axes, Control opacity, error bars, specify intervals for the output.

Two Y axis

For representing any dataset in terms of Line Chart with Two Y axis.

Steps

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.

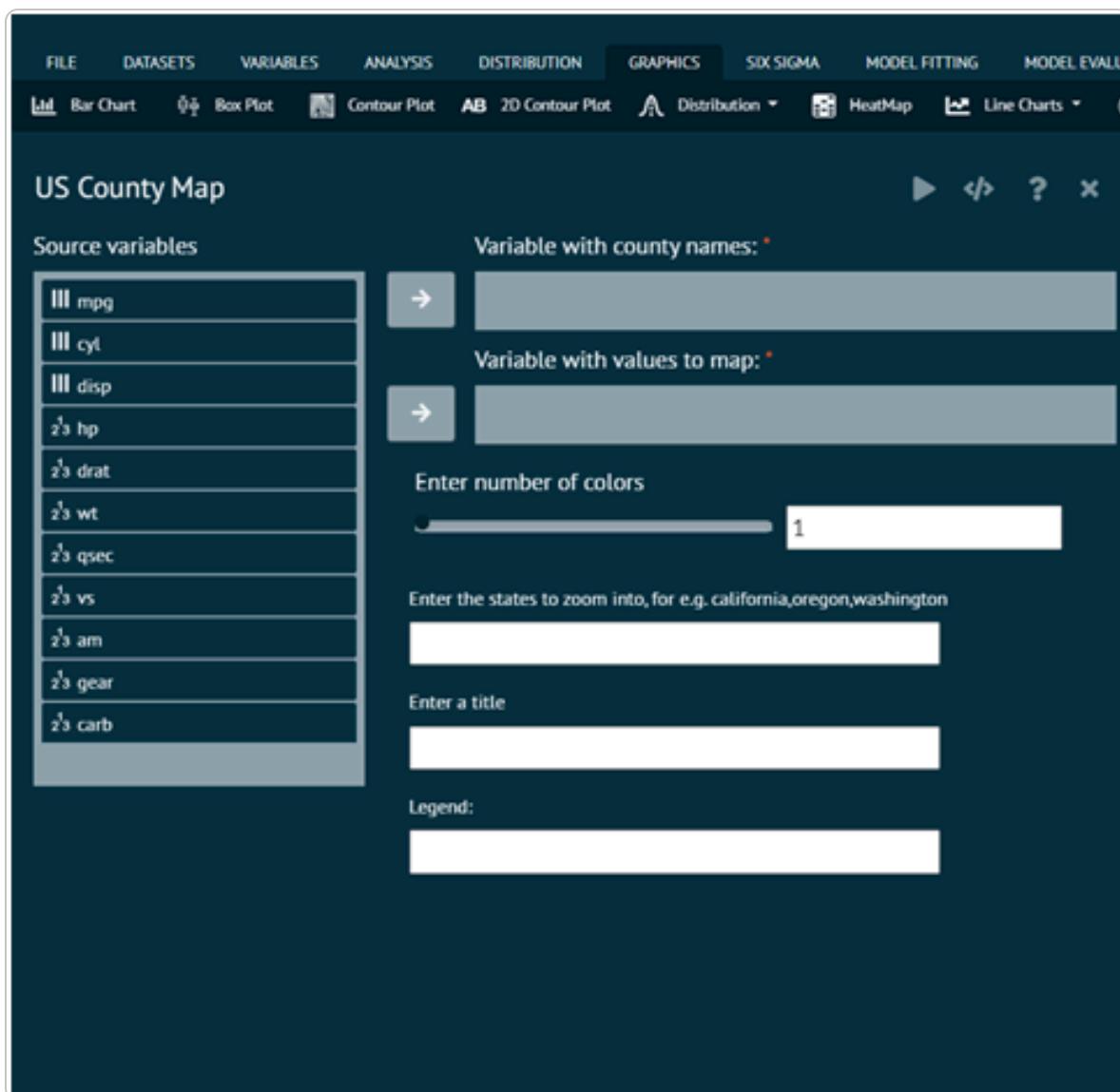


Two Y axis

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**.



US Country map

The screenshot shows a software interface for creating a US State Map. The top navigation bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS (selected), SIX SIGMA, MODEL FITTING, and MODEL EVALUATION. Below the menu bar are various chart types: Bar Chart, Box Plot, Contour Plot, 2D Contour Plot, Distribution (with a dropdown arrow), HeatMap, Line Charts, and a magnifying glass icon.

The main area is titled "US State Map". On the left, a list of "Source variables" is shown, including mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, and carb. The "carb" variable is currently selected, highlighted with a dark blue background.

On the right, there are several configuration fields:

- "Variable with US State names:" (with an arrow button to map from the source list)
- "Variable with values to map:" (with an arrow button to map from the source list)
- "Enter number of colors": A slider set to 1, with a value input field showing "1".
- "Enter the states to zoom into, for e.g. california,oregon,washington": An empty input field.
- "Enter a title": An empty input field.
- "Legend": An empty input field.

US State map

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:

The screenshot shows a software interface for creating a world map. At the top, there's a navigation bar with tabs: FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS (which is selected), SIX SIGMA, MODEL FITTING, and MODEL EVALUATION. Below the navigation bar are several chart type icons: Bar Chart, Box Plot, Contour Plot, AB 2D Contour Plot, Distribution, HeatMap, and Line Charts. The main area is titled 'World Map' and contains the following sections:

- Source variables:** A list of variables from a dataset, including mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, and carb.
- Variable with country names:** An input field with an arrow icon to map variables to countries.
- Variable with values to map:** Another input field with an arrow icon.
- Enter number of colors:** A slider set to 1.
- Enter the countries to zoom into:** A text input field.
- Enter a title:** A text input field.
- Legend:** A text input field.

World Map

Pie Charts

The Pie Charts tab of graphics menu contains 2 options of data visualization i.e., Coxcomb plot, PIE Chart.

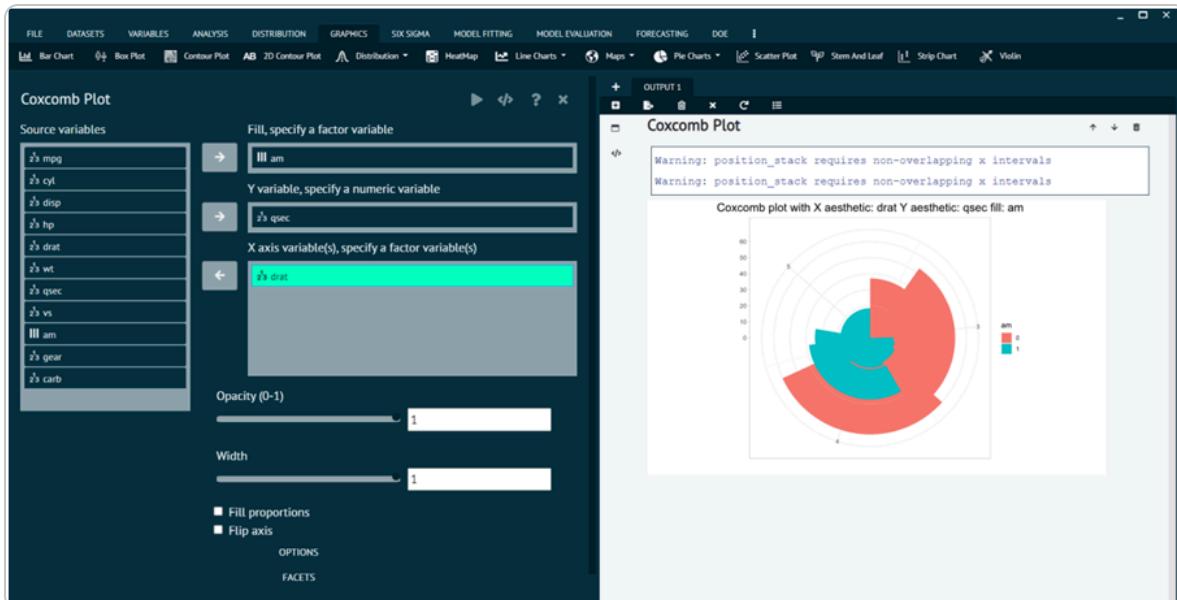
Coxcomb Plot

For representing any dataset in terms of Coxcomb Plot.

Steps

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.



Coxcomb Plot

- i** The Options tab and Facets tab at the bottom can be utilized to add more features to the output.

- i** User can also flip axis, fill proportions, control the opacity, width of the pie chart in the output.

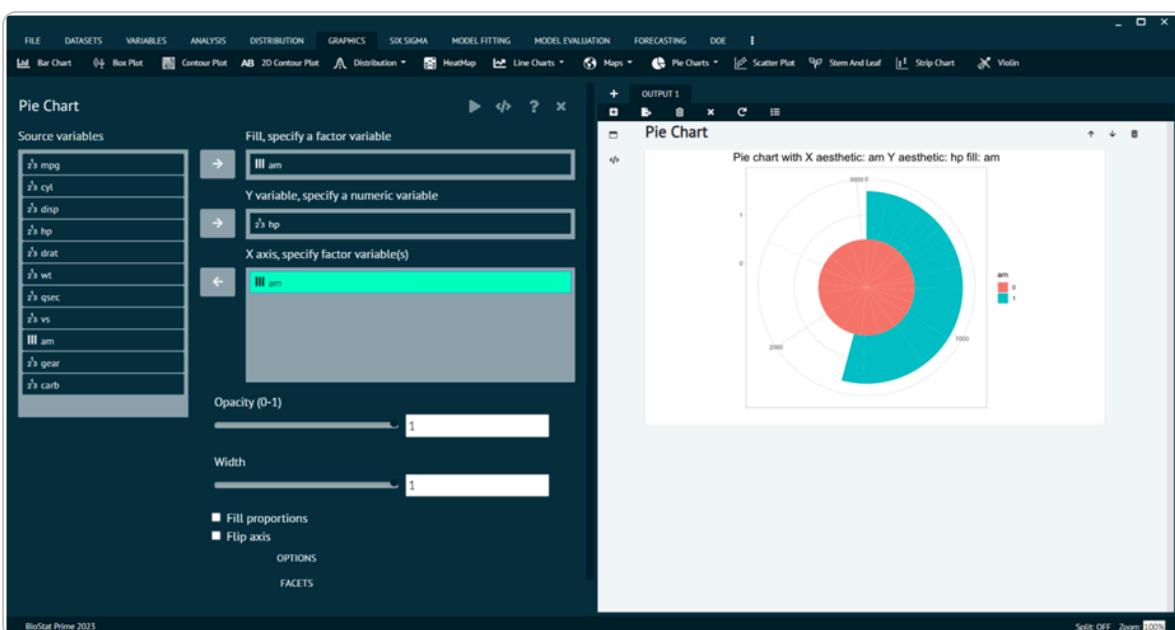
Pie Chart

For representing any dataset in terms of Pie Chart.

Steps

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 below.



Pie Chart

- i** The Options tab and Facets tab at the bottom can be utilized to add more features to the output.

- i** User can also flip axis, fill proportions, control the opacity, width of the pie

chart in the output.

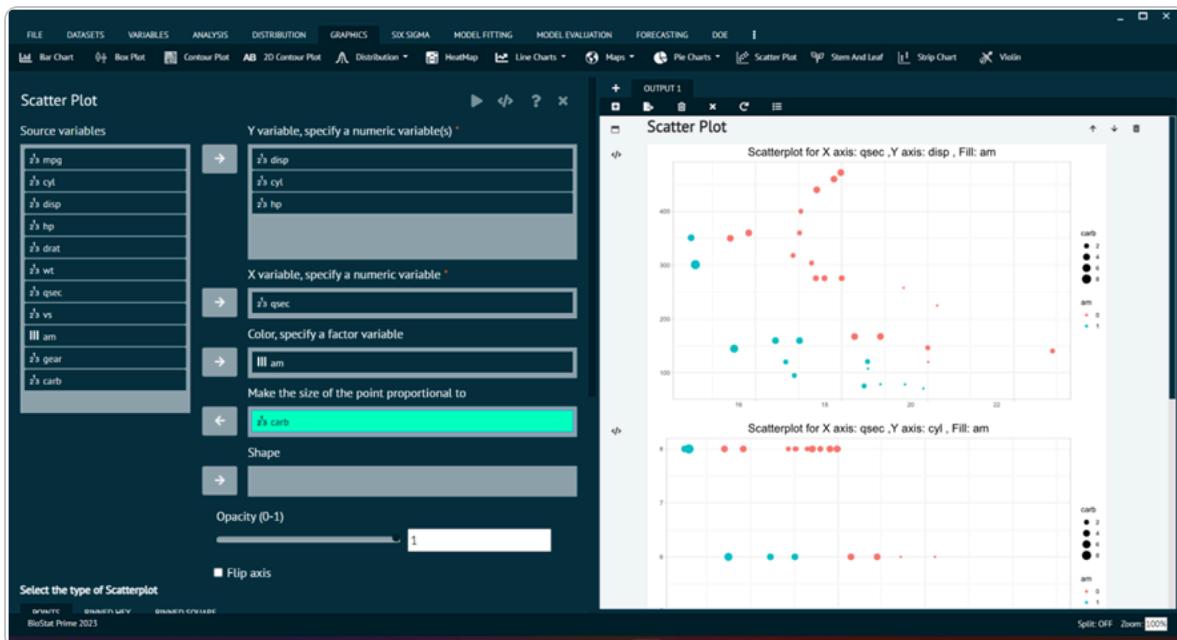
Scatter plots

For representing any dataset in terms of Scatter Plot.

Steps

Load the dataset that needs to be visualized -> Go to Graphics -> Scatter Plots -> Put in the values for variables -> Execute the dialog.

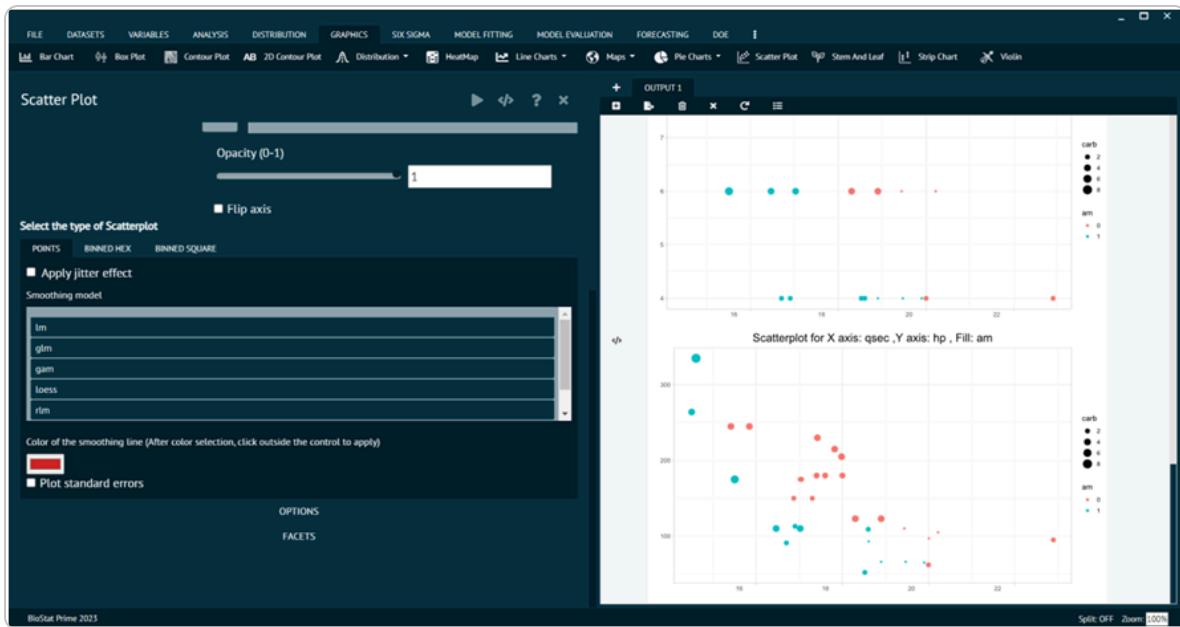
The output of the Scatter Plot of a sample dataset can be seen in the picture below.



Scatter plots

- User can also flip axis, control the opacity of the plot in the output.

The Select type of Scatter plot tab at the bottom can be utilized to add more features to the output as shown below.



Scatter plots

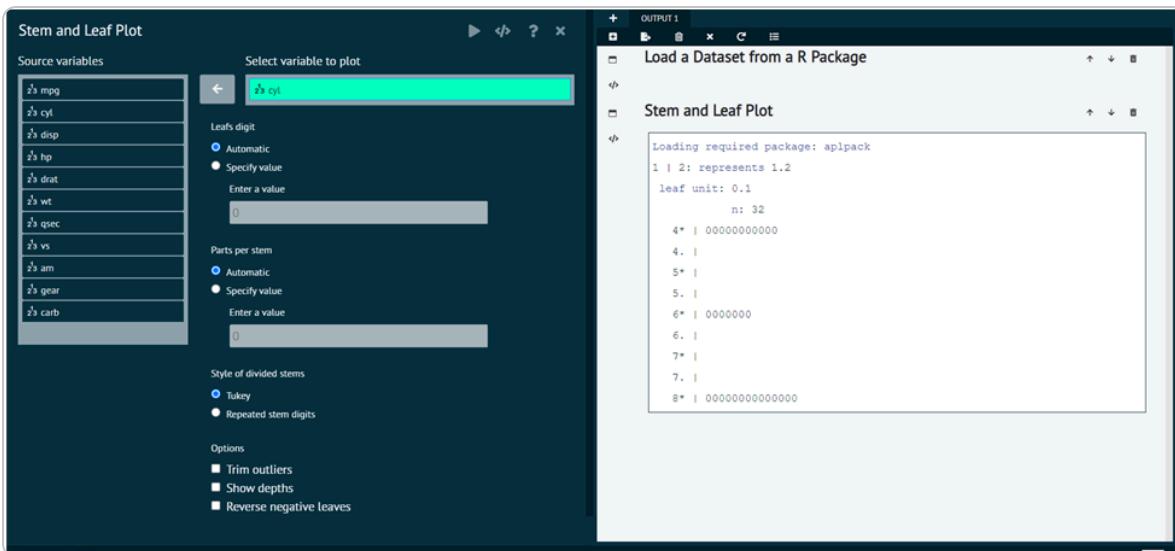
Stem And Leaf

For representing any dataset in terms of Stem and Leaf.

Steps

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.



Stem And Leaf

- i User can also Trim the outlines, show depths, Reverse negative leaves.

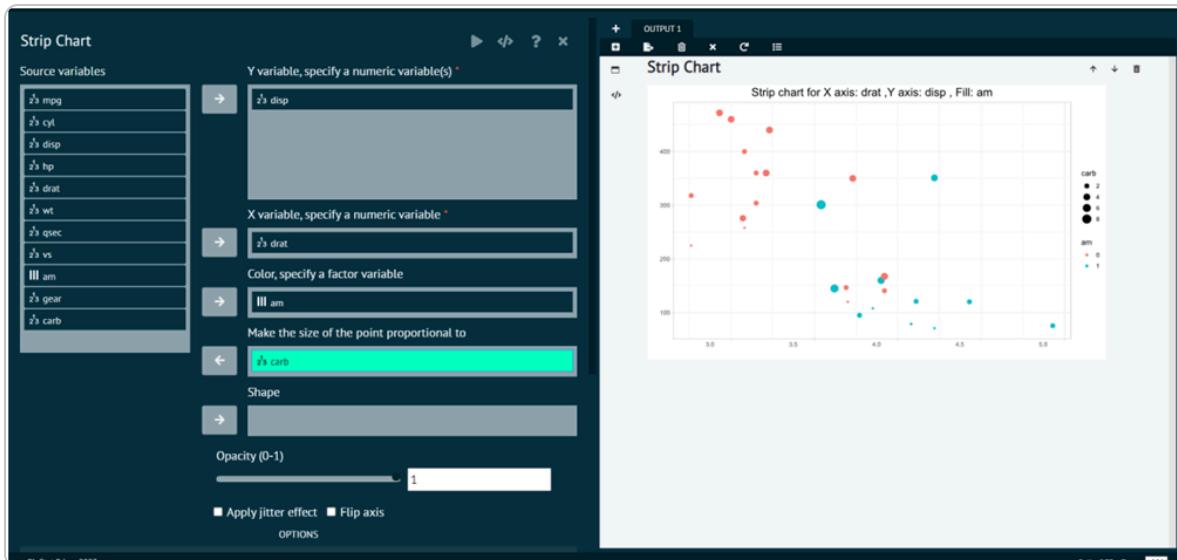
Strip Chart

For representing any dataset in terms of Scatter Plot.

Steps

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.



Strip Chart

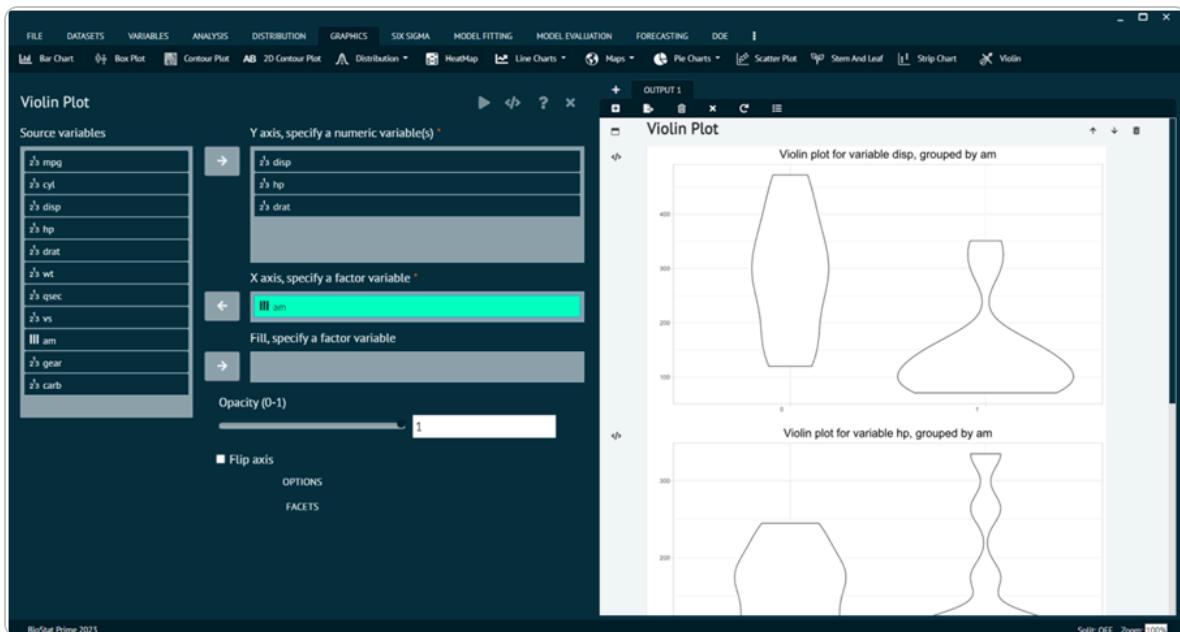
- User can also flip axis, apply jitter effect, control the opacity of the plot in the output.

Violin

For representing any dataset in terms of Violin.

Steps

Load the dataset that needs to be visualized -> Go to Graphics -> Violin -> Put in the values for variables -> Execute the dialog.



Violin

Six Sigma-Quality Control

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 (which is circled in red), MODEL FITTING, and MODEL EVALUATION. Below the navigation bar, there is a toolbar with icons for SixSigma Overview, Pareto Chart, and others. The main content area displays the "Six Sigma reference tutorial by Thomas Pyzdek (Pyzdek Institute)". It includes a list of resources under "Pyzdek Institute's Six Sigma tutorial" and "R Package (QCC) - quality control charting tutorial". There is also a section for "Additional Six Sigma tutorial found on the internet (based on R SixSigma Package)" with a list of five parts. At the bottom left of the content area, it says "BioStat Prime 2023".

Six Sigma reference tutorial by Thomas Pyzdek
(Pyzdek Institute)

Pyzdek Institute's Six Sigma tutorial

- What is Six Sigma and How Does It Work?

R Package (QCC) - quality control charting tutorial

- R quality control charting tutorial

Additional Six Sigma tutorial found on the internet (based on R SixSigma Package)

- Part-1
- Part-2
- Part-3
- Part-4
- Part-5

BioStat Prime 2023

Six Sigma Overview

Cause and Effect

In Six Sigma, Cause and Effect analysis is often used to identify potential causes of a problem or defect within a process. A common tool for this analysis is the Cause and Effect Diagram, also known as the Fishbone Diagram or Ishikawa Diagram.

Purpose of Cause and Effect in Six Sigma

The goal of using a Cause and Effect analysis is to systematically explore all possible causes of a problem, particularly in the Analyze phase of the DMAIC process (Define, Measure, Analyze, Improve, Control). This helps to focus on the root causes of defects, rather than symptoms.

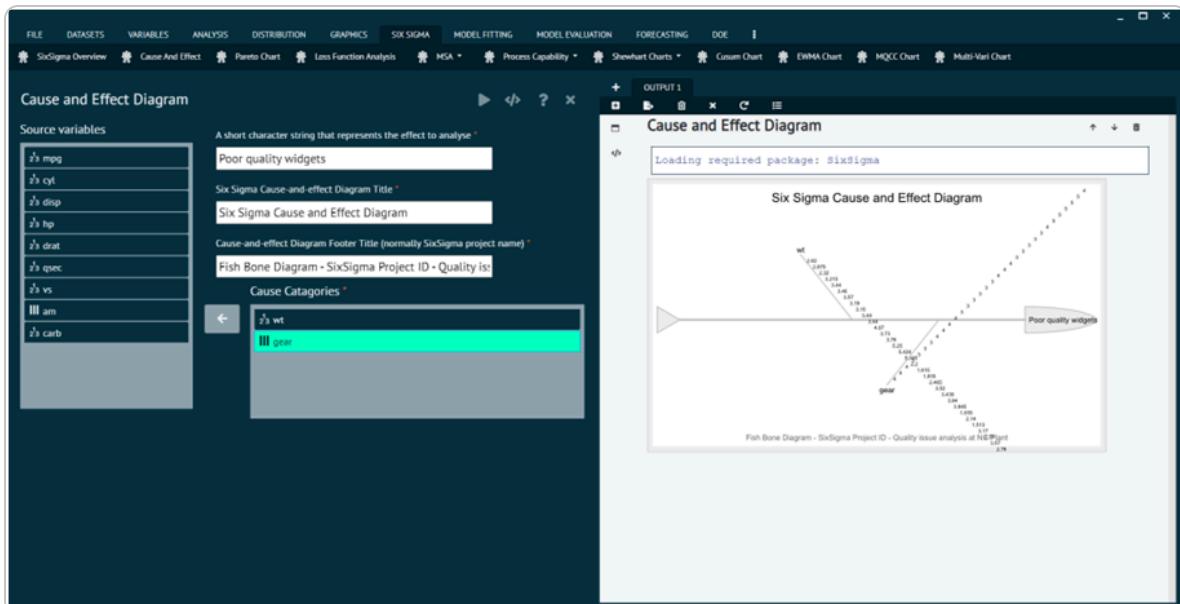
How Cause and Effect Diagrams Work

A typical Fishbone Diagram is structured with the main problem or effect at the head of the fish, and the potential causes branching off the spine into various categories. Each branch represents a factor that may contribute to the issue. The analysis aims to trace all root causes contributing to the observed issue.

To analyse Cause and Effect in BioStat user must follow the steps given below.

Step

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.



How Cause and Effect Diagrams Work

Pareto Chart

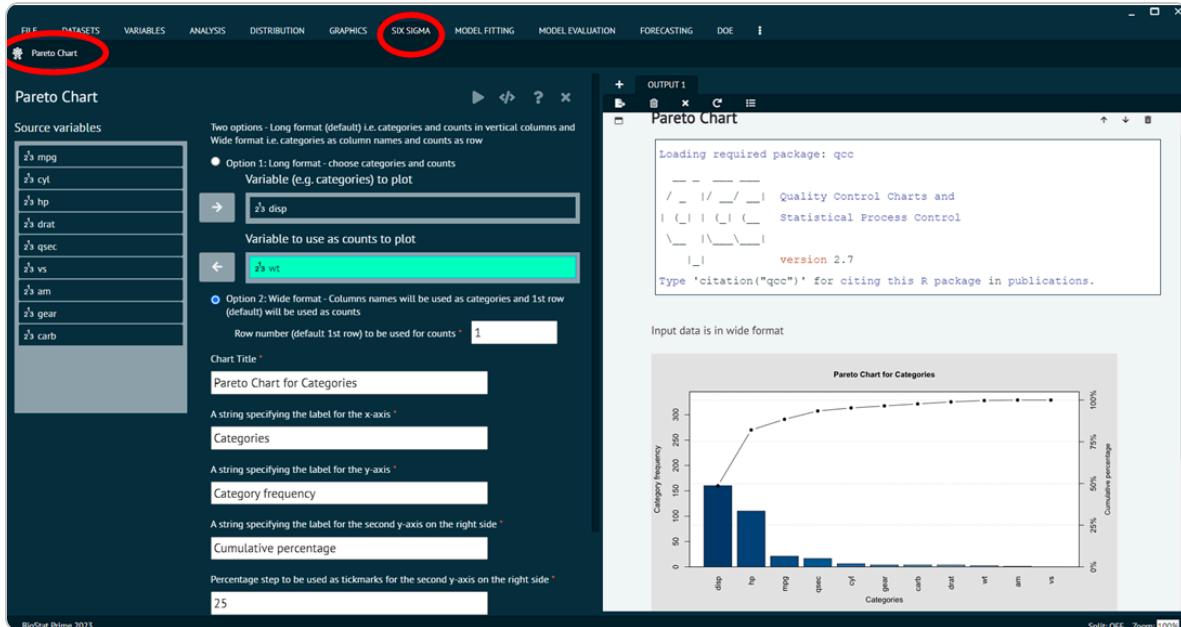
A Pareto chart is a specific type of chart used in statistics that combines both bar and line charts. Pareto chart is designed to highlight the most important factors among a set of variables. The chart is based on the Pareto principle, which states that, **for many phenomena, roughly 80% of the effects 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.

Steps

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.



Pareto Chart

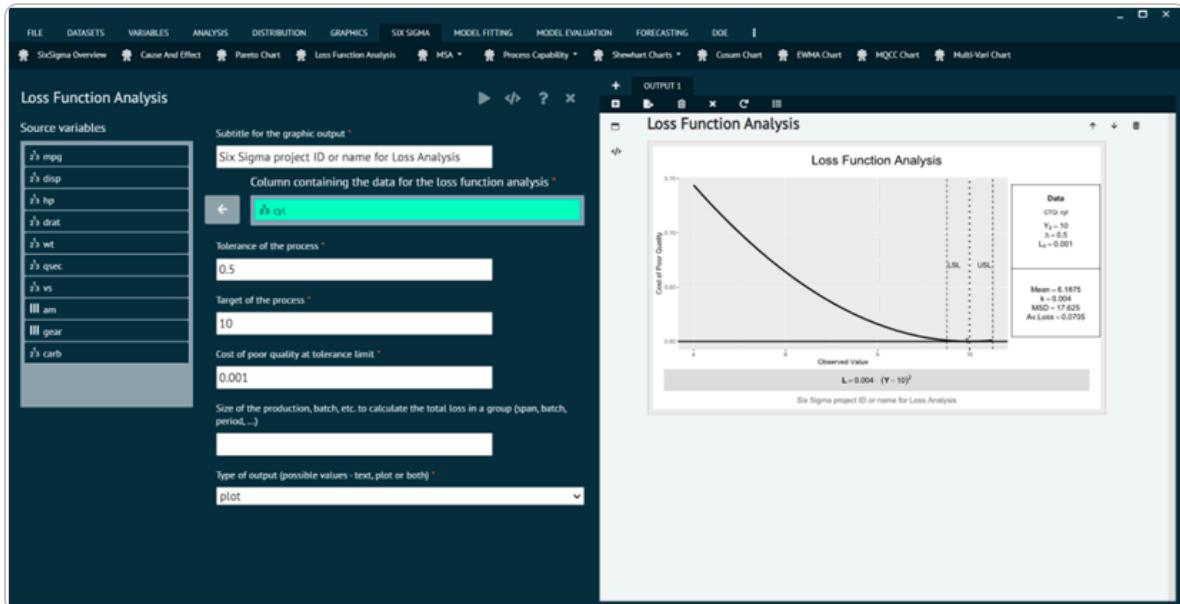
Loss Function Analysis

Loss Function Analysis is a key concept in statistics, machine learning, and optimization, used to quantify the cost of errors or the difference between predicted and actual values. The purpose of a loss function is to guide a model during the training phase by minimizing the "loss" (i.e., error) so that the model can make accurate predictions.

To analyse Loss Function Analysis in BioStat user must follow the steps given below.

Steps

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.



Loss Function Analysis
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

Gage Repeatability and Reproducibility (Gage R&R) is a critical tool in Measurement System Analysis (MSA) used to evaluate the accuracy and precision of a measurement system. It quantifies the amount of variability in measurements due to the measurement system itself, helping to determine whether the system is reliable enough for use in quality control or process improvement.

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.

Steps

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

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

© SixSigma-GageR&R.com. All rights reserved. SixSigma-GageR&R.com is a trademark of SixSigma-GageR&R.com Inc.

Gage R&R-Measurement System Analysis

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.

Steps

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 Minitab software interface for performing an Attribute Agreement Analysis. The main dialog box is titled "Attribute Agreement Analysis". It has several input fields and dropdown menus:

- Source variables:** A list of variables from a dataset, including cyl, hp, drat, wt, qsec, vs, gear, and carb.
- Select the variable for sample/part:** A dropdown menu showing "19 mpg".
- Select the variable for appraiser/operator:** A dropdown menu showing "19 disp".
- Select the variable for attribute/response:** A dropdown menu showing "19 am".
- (Optional) Select the variable for reference/standard response:** A dropdown menu showing "19 am".
- Confidence interval (alpha) between 0 to 1:** An input field containing "0.95".
- 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)"

To the right, an "OUTPUT 1" window displays the results of the analysis:

Within Appraiser Agreement

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
167.4000	1	25	4	0.1012	20.3517

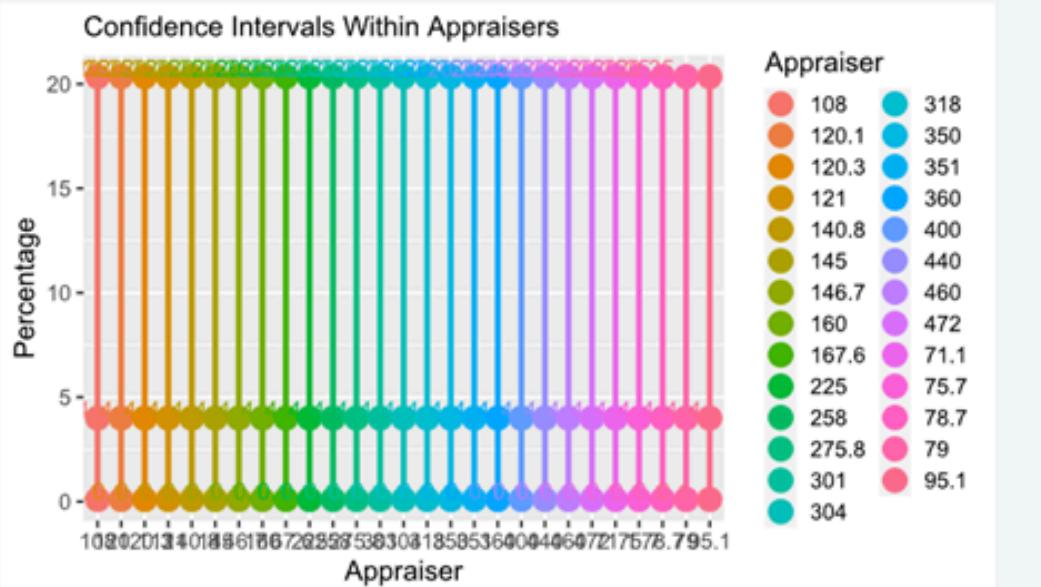
Attribute Agreement Analysis

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 ' '



Attribute Agreement Analysis

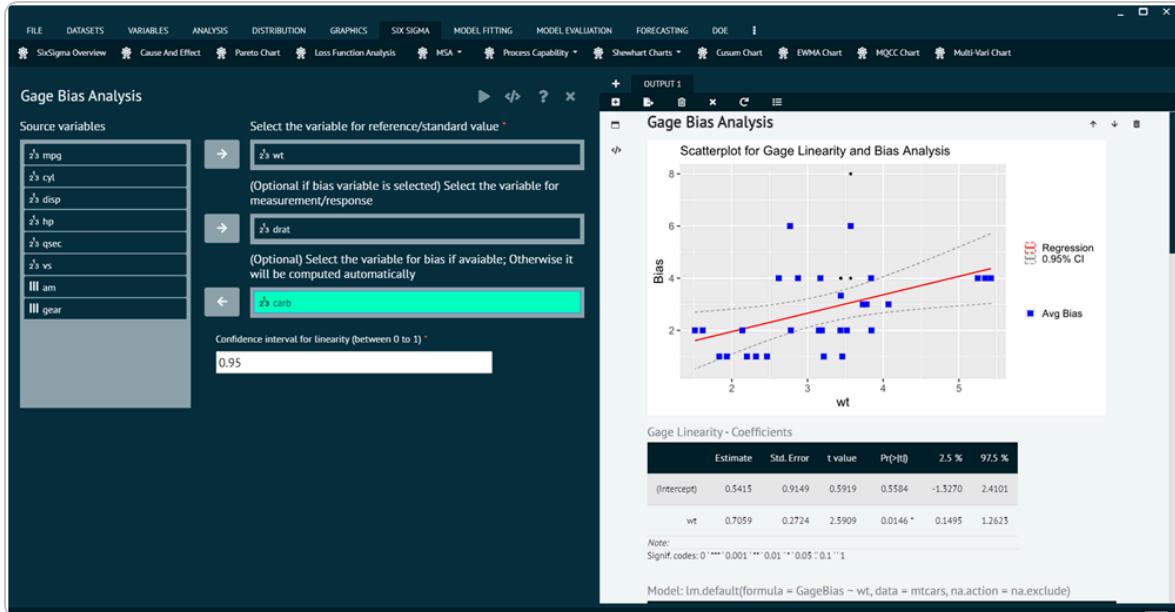
Gage Bias Analysis

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.

Steps

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.



Gage Bias Analysis

Process Capability

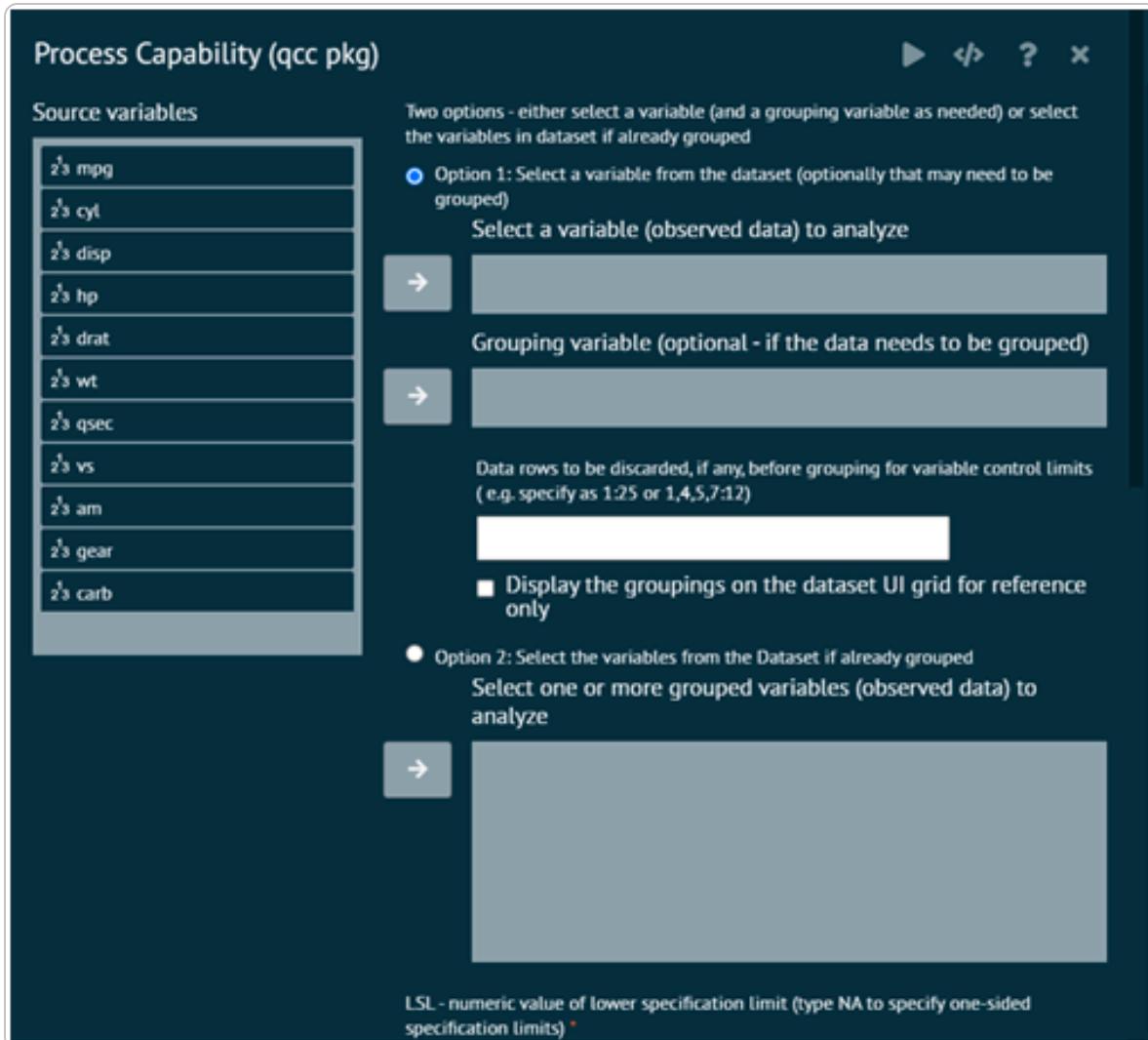
Process Capability is a statistical measure that assesses how well a process can produce products or deliver services within specified limits. It is a key concept in statistical quality control and is used to determine whether a process is capable of meeting predefined specifications. The main objective 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.

Steps

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.



Process Capability (Qcc Pkg)

Process Capability (SixSigma Pkg)

To analyse in Process Capability (SixSigma Pkg) BioStat user must follow the steps given below.

Steps

- _ 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.

Process Capability Analysis

Source variables

- `~hp`
- `~mpg`
- `~cyl`
- `~disp`
- `~drat`
- `~wt`
- `~vs`
- `~am`
- `~gear`
- `~carb`

Variable with the data of the process performance *

`~hp`

LSL - numeric value of lower specification limit *

USL - numeric value of upper specification limit *

Compute a Confidence Interval

Alpha - Type I Error (α) for the Confidence Interval *

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 *

Subtitle for the graphic output (e.g. the name of the Six Sigma project) *

Six Sigma project ID or name for Process Capa

OUTPUT1

Process Capability Analysis

Cp value without Confidence Intervals

Cp

0

Cpk value without Confidence Intervals

Cpk

-0.6888

Cp Z value

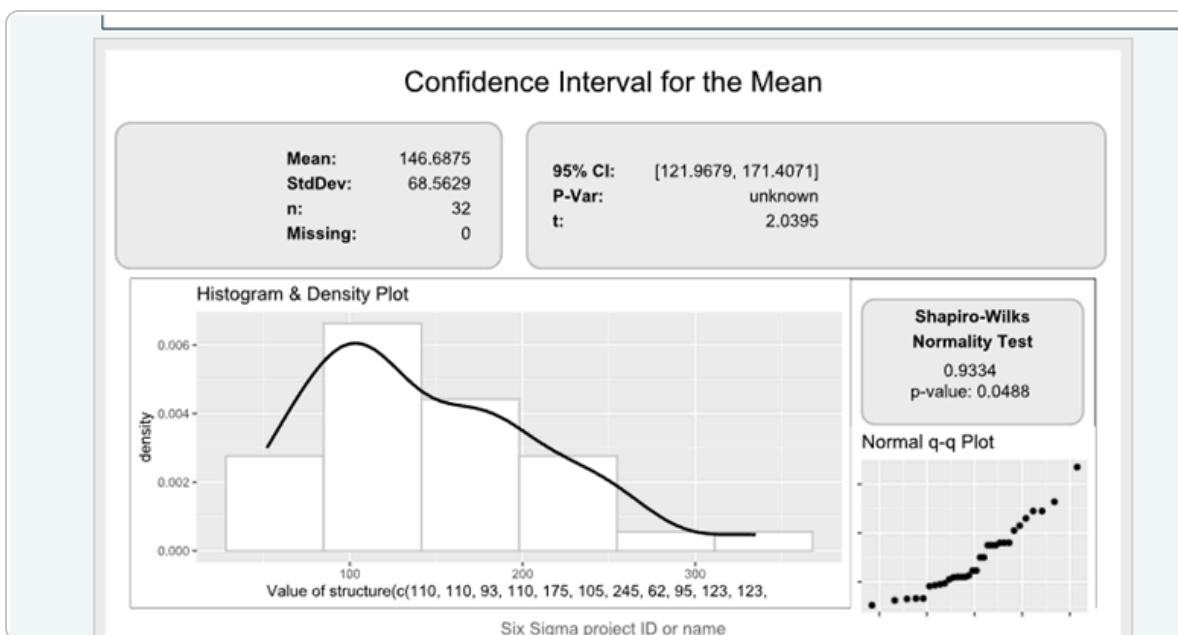
Cp Z

-2.0665

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

Process Capability (SixSigma Pkg)



Process Capability SixSigma Pkg

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.

- ❶ Shewhart Charts are fundamental in quality management and Six Sigma methodologies, providing a visual and statistical approach to process control.

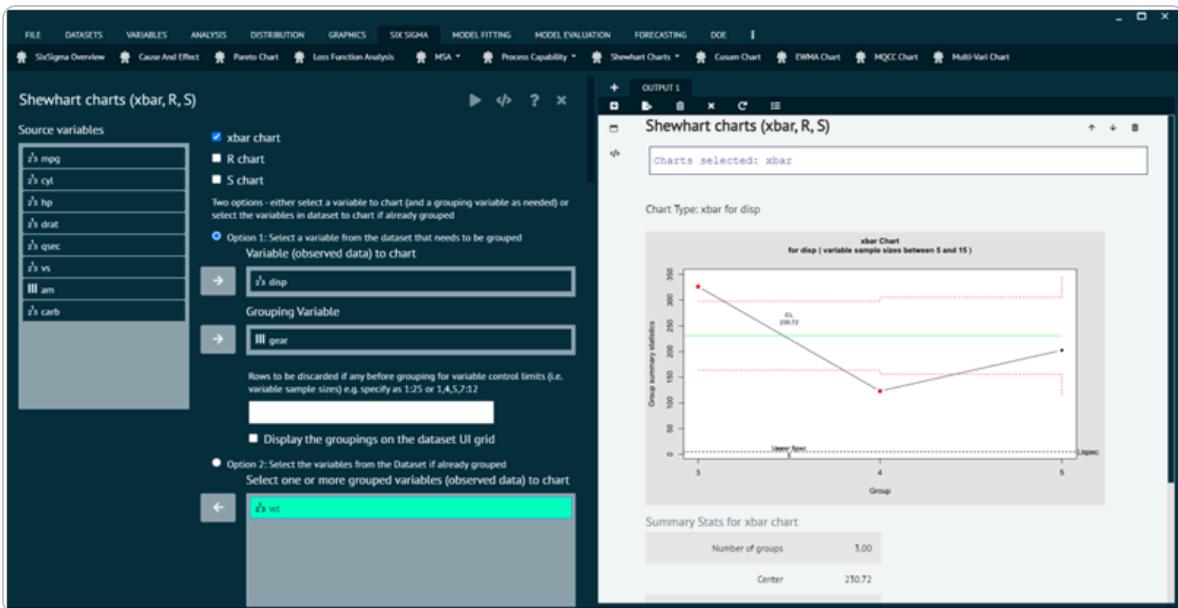
There are several types of Shewhart Charts, each designed to monitor different aspects of a process.

Shewhart Chart (Xbar,R,S)

To analyse in Shewhart Chart (Xbar,R,S) BioStat user must follow the steps given below.

Steps

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.



Shewhart Chart (Xbar,R,S)

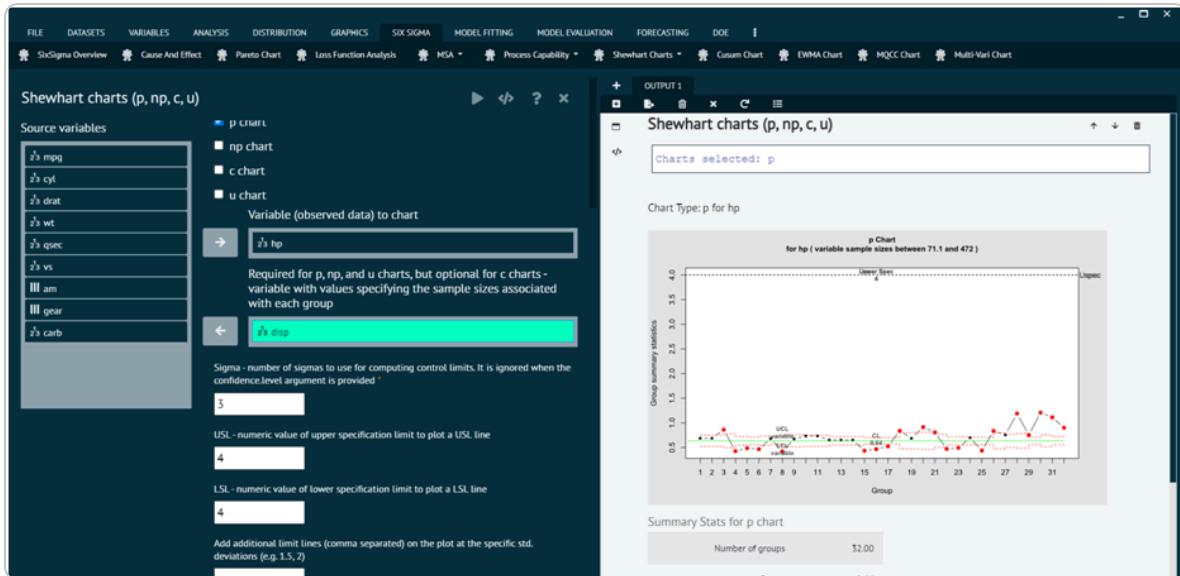
Shewhart Chart (P,NP,C,U)

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.

Steps

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.



Shewhart Chart (P,NP,C,U)

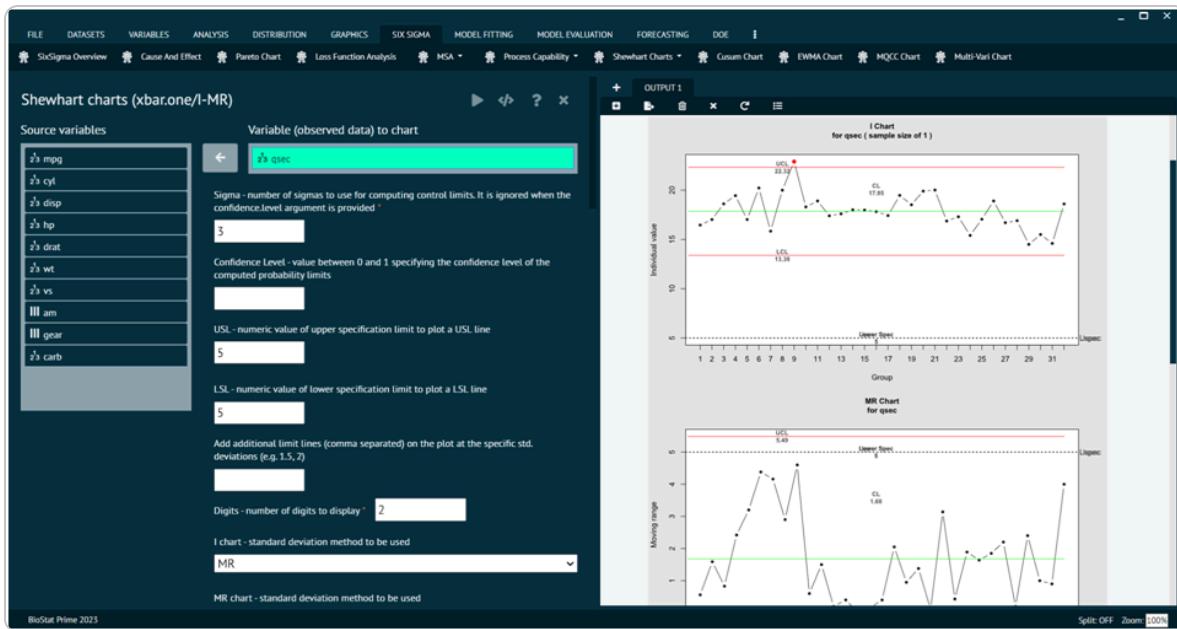
Shewhart Chart (Xbar.One/I-MR)

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.

Steps

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.



Shewhart Chart (Xbar.One/I-MR)

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.

Steps

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

Shewhart Chart (I-MR Between/Within)

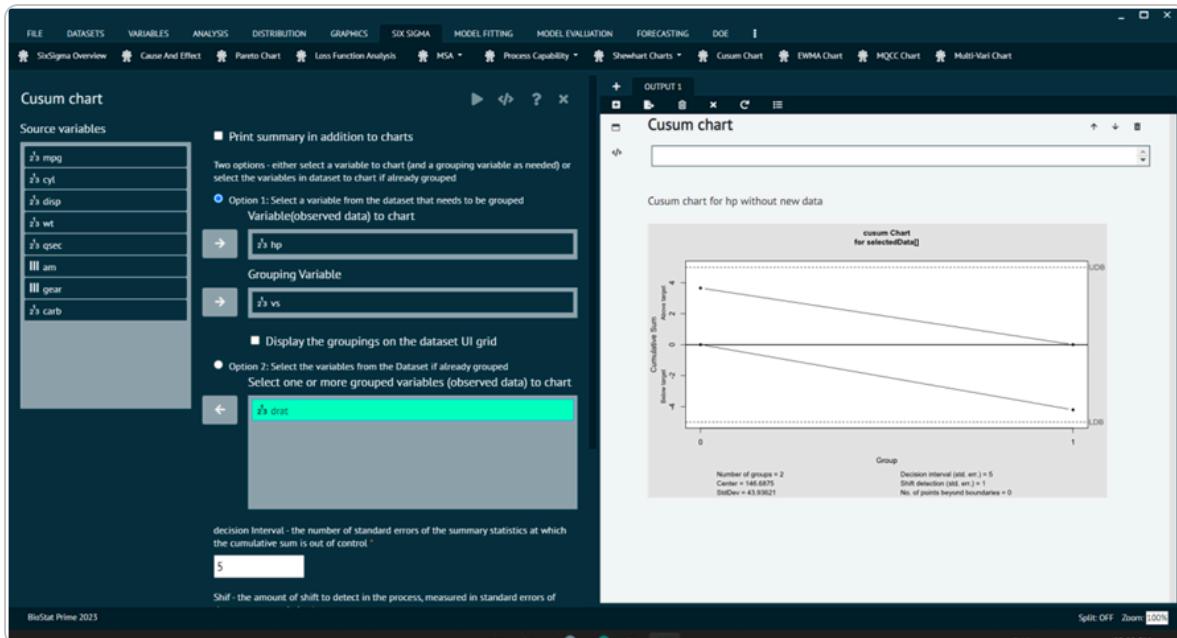
Cusum Chart

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.

Steps

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.



Cusum Chart

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.

Steps

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

\bar{x} mpg
\bar{x} cyl
\bar{x} disp
\bar{x} hp
\bar{x} drat
\bar{x} wt
\bar{x} qsec
\bar{x} vs
\bar{x} am
\bar{x} gear
\bar{x} 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 *

3

The smoothing parameter (between 0 and 1) *

0.2

EWMA Chart

MQCC Chart

The MQCC Chart (Moving Average Quality Control Chart) is not a widely recognized term in traditional statistics or quality control literature. However, I believe you might be referring to a Moving Average Quality Control Chart (MAQCC), or simply a Moving Average Chart (MA Chart). These charts are used in statistical process control (SPC) to monitor the performance of a process over time.

To analyse MQCC chart in BioStat user must follow the steps given below.

Steps

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

\bar{x}_3 mpg
\bar{x}_3 cyl
\bar{x}_3 disp
\bar{x}_3 hp
\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

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 separated 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

MQCC Chart

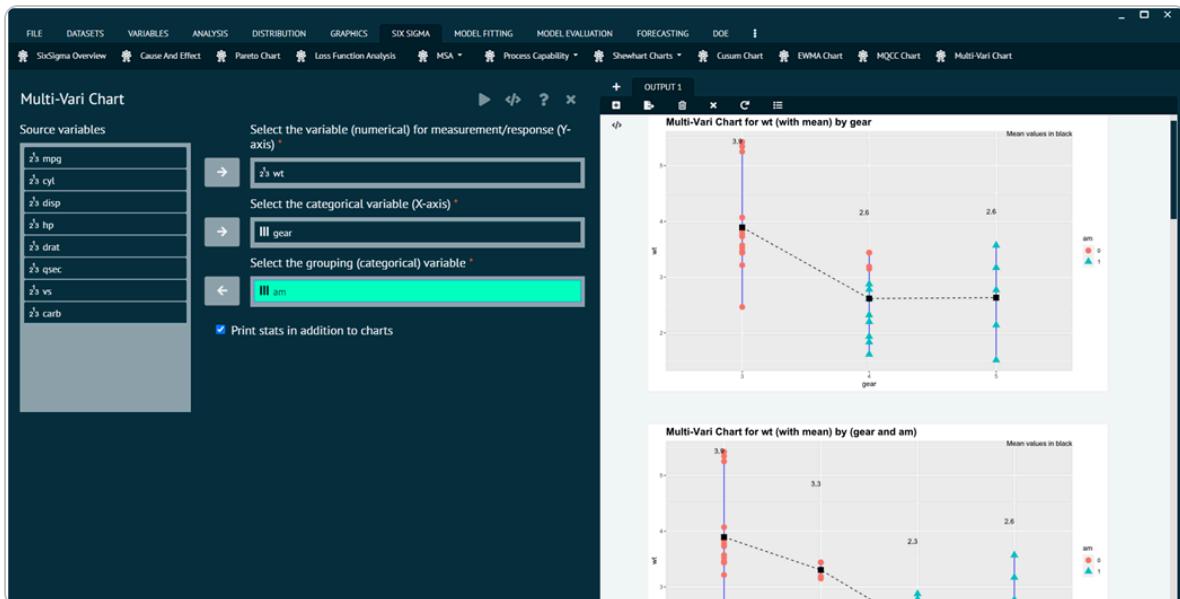
Multi-Vari Chart

A Multi-Vari Chart is a graphical tool used in statistics, especially in quality control and Six Sigma, to study the variability in a process by displaying multiple sources of variation in one chart.

To analyse Multi-Vari chart in BioStat user must follow the steps given below.

Steps

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.



Multi-Vari Chart

Model-Curve 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.

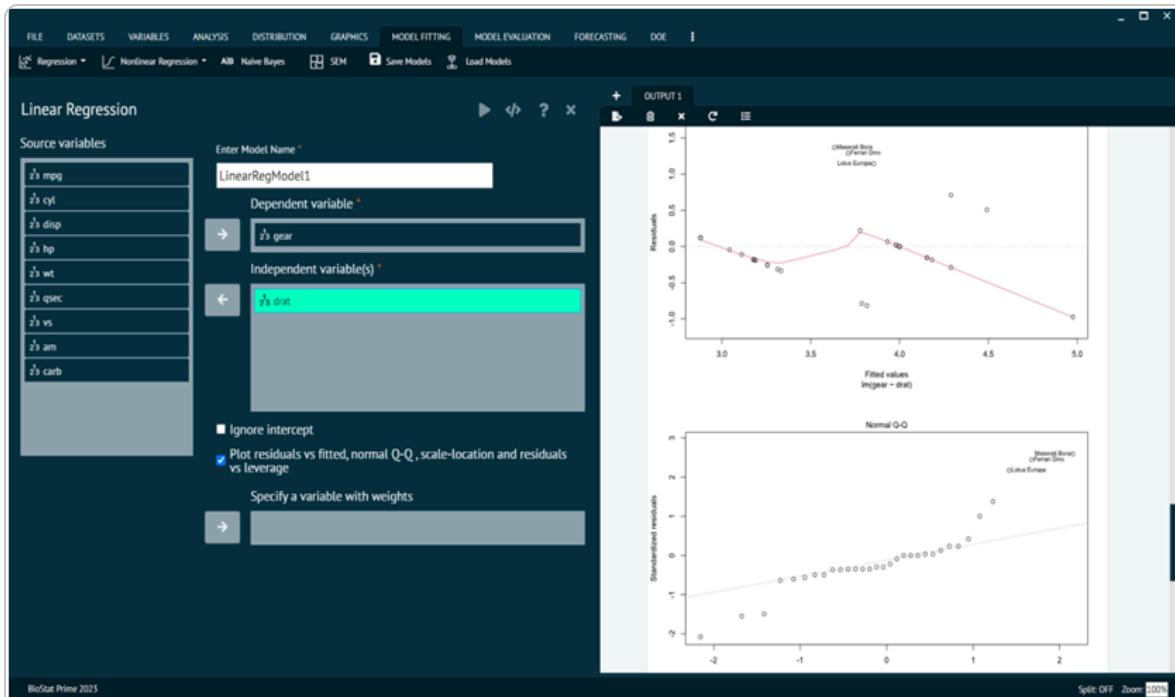
Regression

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.

Steps

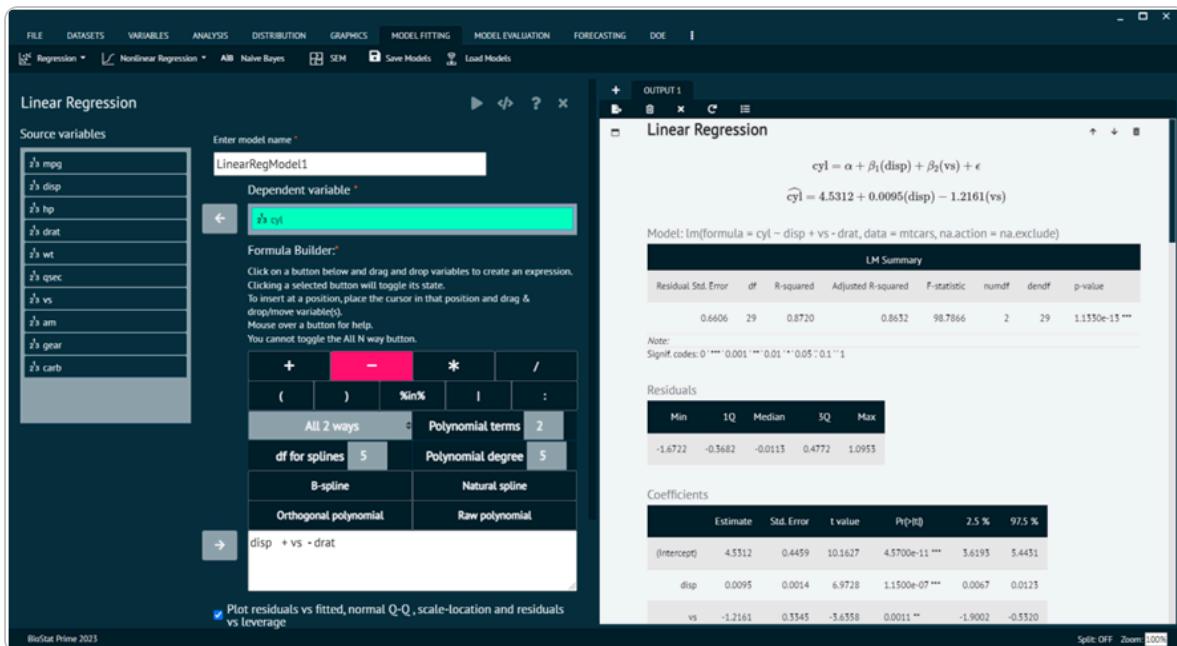
Load the dataset -> Click on the Model Fitting tab in main menu -> Select Linear regression -> This leads to analysis techniques dropdown -> The various options in the dialog can be selected to opt for plot etc -> Finally execute the plot and visualise the output in output window.



Regression

BioStat also provides advanced regression analysis functions that can create models

based on interactive terms via formula builder.



Regression formula builder

- i** Note that the variables so selected are substituted as quotients in the formula built by the user.

Cox, Advanced

Cox proportional hazards model, often called the Cox model, this model is widely used in the analysis of survival data to investigate the effect of explanatory variables on the time a specified event takes to occur.

Cox Proportional Hazards Model

Fits a Cox proportional hazards model for time-to-event data with censored observations. Model fitting statistics, parameter estimates, and hazard ratios are provided. Options available include the **tied time method**, **model diagnostics** such as **proportional hazards** and **covariate functional form assessments**, and a **forest plot of hazard ratios with confidence intervals**. The model is fit using the `coxph` function in the survival package.

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

Steps

Load the dataset -> click on the Model Fitting tab in main menu -> Select Regression -> This leads to analysis techniques, choose Cox Advanced -> 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 with formula builder -> 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

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

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			

Cox Proportional Hazards Model

Arguments

Time

Time to event for those experiencing the event or time to last follow-up for those not experiencing the event

Event

Numerical event indicator; 1=event, 0=censor

Independent Variables

Independent variables to include in the model. Factors, strings, and logical variables will be dummy coded.

Weights

Numeric variable for observation weights. Useful in situations where each record should not be counted as one observation.

Required packages

survival, broom, survminer, car, BlueSky

i Click the Get R Help button to get detailed R help about the coxph function.

Options

Tied Time Method

Method of breaking tied observed times. Efron is usually the better choice when there aren't many tied times. The exact method can be beneficial if there are many tied times, as in discrete time situations, but can take a little longer for the model to be fit.

Forest Plot

Plot of hazard ratios and confidence intervals for each predictor in the model.

Model Diagnostics

If selected, proportional hazards tests and plots will be provided, in addition to assessments of functional form for each covariate in the model. The null model Martingale residual axis minimum value option might need to be changed so that all residuals appear in the plot. To get functional form assessments, you must specify only numeric predictors and have no missing data. See Variables > Missing Values > Remove NAs.

Analysis of Deviance (Type II)

Global test of each predictor in the model. Multi-degree of freedom tests will be provided for effects with more than 2 levels. Wald and Likelihood ratio tests can be obtained, with likelihood ratios tests having better small sample properties.

Cox, Basics

The Cox proportional hazards model is used to analyze and interpret the effect of several variables on the time it takes for a particular event to occur. This event could be anything that marks the end of a period, such as death, failure, recovery, or relapse.

The Cox proportional hazards model is extensively used in medical research, particularly in studies of treatment efficacy, clinical trials, and epidemiological research. It is also used in various fields where time-to-event data is relevant.

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

Steps

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 software interface. The top menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION, FORECASTING, DOE, and HELP. The MODEL FITTING tab is selected. Below the menu is a toolbar with icons for Regression (highlighted), Nonlinear Regression, All Naive Bayes, SEM, Save Models, and Load Models. The main workspace is titled 'Cox, Basic'. It contains a 'Source variables' list with items like mpg, cyl, disp, drat, wt, qsec, am, and gear. A 'Time to event or censor' field is set to 'hp', and an 'Events (1 = event 1, 0 = censor)' field is set to 'vs'. An 'Independent Variables' list has 'carb' selected. A 'Weights (optional)' field is empty. At the bottom is an 'OPTIONS' button. To the right is an 'OUTPUT 1' window titled 'Cox, Basic' showing the 'Cox Model Summary for Surv(hp,vs)' results:

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	0.4552
r.squared.max	0.9377
concordance	0.8037
std.error.concordance	0.0401

Cox, Basics

- i** Click on the ? button on the top right of the dialog for details on sample datasets and the data format supported.

Arguments

Time

Time to event for those experiencing the event or time to last follow-up for those not experiencing the event

Event

Numerical event indicator; 1=event, 0=censor

Independent Variables

Independent variables to include in the model. Factors, strings, and logical variables will be dummy coded.

Weights

Numeric variable for observation weights. Useful in situations where each record should not be counted as one observation.

Required packages

survival, broom, survminer, car, BlueSky

- i** Click the Get R Help button to get detailed R help about the coxph function.

Options

Tied Time Method

Method of breaking tied observed times. Efron is usually the better choice when there aren't many tied times. The exact method can be beneficial if there are many tied times, as in discrete time situations, but can take a little longer for the model to be fit.

Forest Plot

Plot of hazard ratios and confidence intervals for each predictor in the model.

Model Diagnostics

If selected, proportional hazards tests and plots will be provided, in addition to assessments of functional form for each covariate in the model. The null model Martingale residual axis minimum value option might need to be changed so that all residuals appear in the plot. To get functional form assessments, you must specify only numeric predictors and have no missing data. See Variables > Missing Values > Remove NAs.

Analysis of Deviance (Type II)

Global test of each predictor in the model. Multi-degree of freedom tests will be provided for effects with more than 2 levels. Wald and Likelihood ratio tests can be obtained, with likelihood ratios tests having better small sample properties.

Cox, Binary Time-depended covariates

Fits a Cox proportional hazards model for time-to-event data with censored observations that includes one or more binary time-dependent "exposure" covariates. This type of covariate is one where the occurrence of a "yes" can happen after the start of follow-up and once that "yes" occurs it stays a "yes" for the follow-up duration.

An example would be rejection of a graft post-transplant (i.e. before the rejection occurs, the patient has not been "exposed"; after rejection occurs, the patient has been "exposed").

Treating such a covariate as being known at the start of follow-up is a form of looking into the future and leads to biased estimates (also known as "immortal time bias").

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

Steps

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

- 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 *

CoxRegModel1

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%	I	:
All 2 ways	Polynomial terms	2		
df for splines	5	Polynomial degree	5	
B-spline	Natural spline			
Orthogonal polynomial	Raw polynomial			

Formula appears here

Cox, Binary Time-depended covariates

⚠ 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.

Arguments

Enter model name

Name where the model results will be stored.

Time to event or censor

Time to outcome event for those experiencing the event or time to last follow-up for those not experiencing the outcome event.

Events (1=event, 0=censor)

Numerical event indicator; 1=event, 0=censor.

Formula Builder

Construct terms to include in the model. Factors, strings, and logical variables will be dummy coded. The provided buttons allow user to specify main effects, full factorial effects (main effects and all interactions with the involved variables), polynomials, specific interactions, and delete terms from the list.

Exposure time variables for time-dependent covariates

Numeric variables storing the time when the subject was first "exposed". This must be on the same time scale as the Time variable. Missing values should be used for subjects who were never exposed. Each variable specified here will create a separate time-dependent covariate. A specified time assumes that a subject is not exposed prior to this time and exposed after this time (i.e. when the predictor change from "no" to "yes"). Specifying only positive values means that subjects are not exposed for some time after follow-up starts. Specifying some positive and negative times indicates that some subjects were exposed after and some before

follow-up time starts, respectively. If user knows that a subject was exposed prior to the follow-up start time, but user doesn't know exactly when, then user can use any negative time and the model will correctly treat that subject as being exposed for their entire follow-up time. If subjects are exposed after follow-up, then they are correctly treated as not being exposed for their entire follow-up time.

Prefix for time-dependent covariates

Desired prefix to be used for every time-dependent covariate specified in the Exposure time variables field. The name of each time-dependent covariate will start with this prefix.

Subject identifier

The variable storing the subject identifier. This is required for purposes of creating the underlying counting process data set.

Weights

Numeric variable for observation weights. Useful in situations where each record should not be counted as one observation.

Options

Model fitting statistics, parameter estimates, and hazard ratios are provided. Options available include the tied time method, forest plots, model diagnostics, and the ability to view the underlying counting process data set that gets created.

Tied Time Method

Method of breaking tied observed times. Efron is usually the better choice when there aren't many tied times. The exact method can be beneficial if there are many tied times, as in discrete time situations, but can take a little longer for the model to be fit.

Forest Plot

Will create a forest plot of hazard ratios and confidence intervals

Show (start, stop) Data Set

Will show the underlying counting process data set used in the computations. This breaks each subject's follow-up time into parts, depending on when the time-dependent covariate should change values.

Model Diagnostics

If selected, proportional hazards tests and plots will be provided, in addition to assessments of functional form for each covariate in the model. The null model Martingale residual axis minimum value option might need to be changed so that all residuals appear in the plot. To get functional form assessments, you must specify only numeric predictors and have no missing data. See Variables > Missing Values > Remove NAs.

Analysis of Deviance (Type II)

Global test of each predictor in the model. Multi-degree of freedom tests will be provided for effects with more than 2 levels. Wald and Likelihood ratio tests can be obtained, with likelihood ratio tests having better small sample properties.



The model is fit using the `coxph` function in the `survival` package.

Cox, Fine-Gray

Fits a Fine-Gray Cox proportional hazards model for time-to-event data with censored observations when competing risks are present. Fine-Gray models model the effect of covariates on the cumulative incidence function.

An alternative is to model the effect of covariates on the cause-specific hazard, for which standard Cox regression models can be used.

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

Steps

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

mpg
cyl
disp
hp
drat
wt
qsec
vs
am
gear
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			

Cox Fine-Gray

Attributes

Enter model name

Specify the name of the model where the results will be stored.

Time to event or censor

Time to the first event for those experiencing an event or time to last follow-up for those not experiencing any event

Events (0 = censor, 1 = event 1, 2 = event 2, ...)

Numerical event indicator; 0=censor, 1=event 1, 2=event 2, etc.

Event Code

Select the event of interest you want to model. Can either be selected or typed in.

Formula Builder

Construct terms to include in the model. Factors, strings, and logical variables will be dummy coded. The provided buttons allow user to specify main effects, full factorial effects (main effects and all interactions with the involved variables), polynomials, specific interactions, and delete terms from the list. Interactions with stratification variables is allowed.

Stratification Variables

Specify one or more stratification variables. These can be numeric, factor, ordered factor, or character variables. The strata divide the subjects into separate groups whereby each group has a distinct baseline hazard function. If multiple stratification variables are given, a separate baseline hazard function is used for every combination of stratification variable levels.

Weights

Numeric variable for observation weights. Useful in situations where each record should not be counted as one observation.

Options

Tied Time Method

Method of breaking tied observed times. Efron is usually the better choice when there aren't many tied times.

Model Diagnostics

If selected, an assessment of proportional hazards and functional form of covariates will be assessed, including relevant plots. If there are stratification variables or non-numeric predictors, functional form plots will not be produced. The Null Model Martingale Residual Axis Minimum Value might need to be changed in order to see all Martingale residuals.

Analysis of Deviance (Type II)

If selected, whole variable tests (including multi-degree of freedom tests for multi-category covariates) will be provided. Wald tests are used.

- i** Required R packages: survival, broom, survminer, car, dplyr
- i** The model is fit using the finegray and coxph functions in the survival package.
- i** Click the R Help button to get detailed R help about the coxph function. Go to Help-> R Function Help to get more information about the finegray function, which creates the needed dataset.

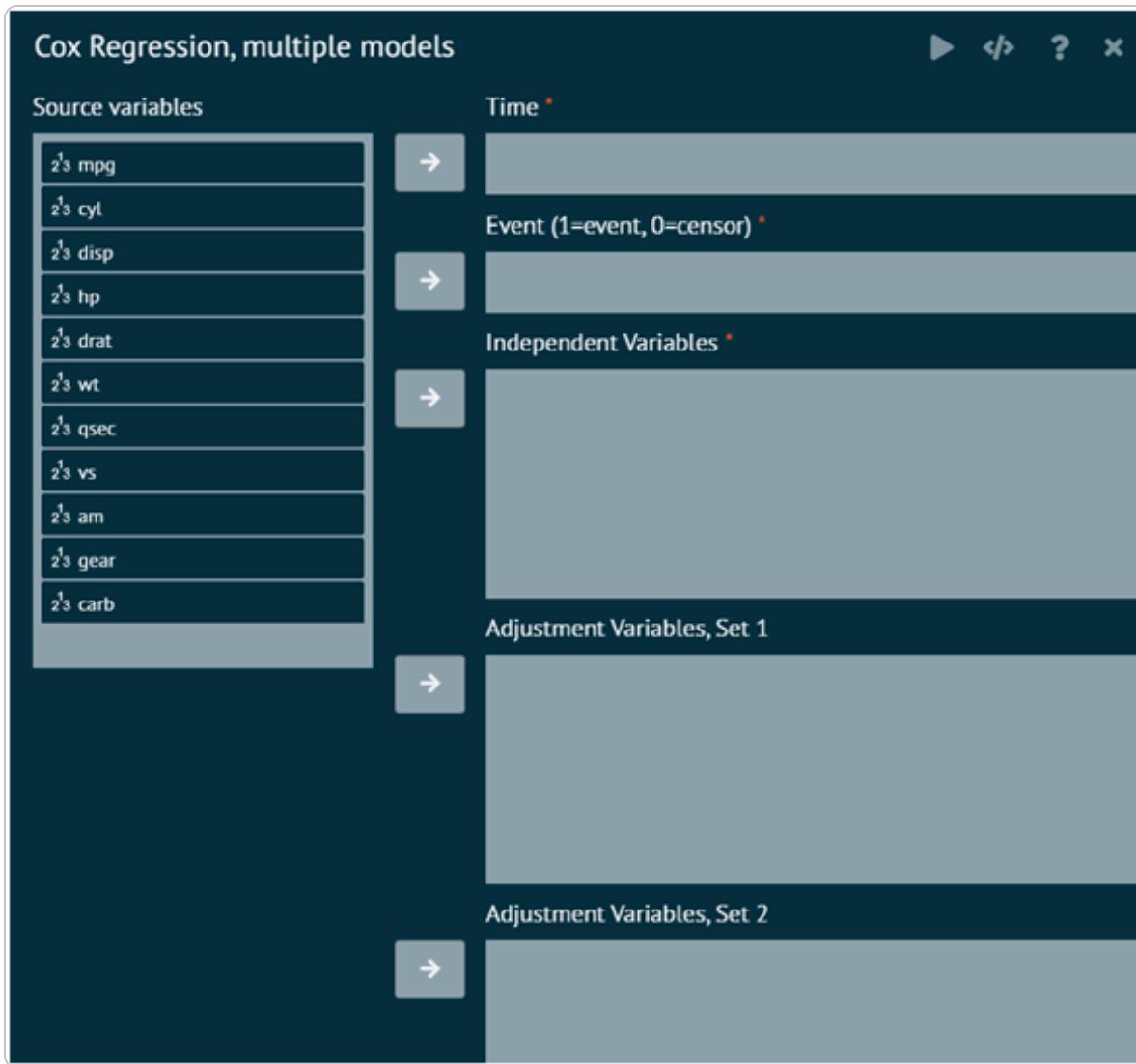
Cox Regression, Multiple models

This creates a table containing results from Cox regression models for provided time and event variables. Separate Cox regression models will be fit for each independent variable, optionally adjusted for a set of additional variables. If a strata variable is specified, separate models will be fit for each of the stratification variable values. As an example, if no adjustor or stratification variables are specified, then the table will include all univariate models for the list of independent variables. Various statistics from each model can be output.

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

Steps

Load the dataset -> Click on the Model Fitting tab in main menu -> Select Regression -> This leads to analysis techniques, choose Cox Regression, Multiple models -> There will appear a dialog -> It is required to select the independent variables in the dialog, set time and set event -> Finally execute the plot and visualise the output in output window.



Cox Regression Multiple models

Attributes

Time

Time variable for each Cox regression model. The variable class must be a numeric type.

Event (1=event, 0=censor)

Event variable for each Cox regression model. A value of 1 indicates the event occurred and 0 indicates the event did not occur. The variable class must be a numeric type.

Independent Variables

Independent variables to include in the models. The variable classes can be a numeric type, character, factor, or ordered factor.

Adjustment Variables (Sets 1-5)

Optional variables to be included in a model with the independent variables. The variable classes can be a numeric type, character, factor, or ordered factor.

Specifying more than one set of adjustor variables will provide separate models with each set of adjustor variables.

Strata

Optional stratification variable. Separate models will be fit for the subset defined by each of the stratification variable values. The variable class can be character, numeric, factor, or ordered factor.

Weights

Optional case-weights to be used in the models. Specifying a weights variable will fit weighted regression models.

Digits After Decimal Continuous Values

The number of decimal places to show for all continuous values in the table
(default=4) P-Values

The number of decimal places to show for all p-values in the table (default=4)
Hazard Ratios

The number of decimal places to show for all hazard ratios in the table (default=4)

Options

Parameter Estimates and Hazard Ratios

Parameter Estimates

Show parameter estimates (coefficients) from each model.

Standard Errors

Show standard errors of the parameter estimates.

Confidence Interval Level

Level for the parameter estimate and hazard ratio confidence intervals (default=0.95).

Parameter Wald Confidence Intervals

Show Wald-based confidence intervals for the parameter estimates.

Hazard Ratios

Show hazard ratios for each parameter estimate ($\exp(\text{coefficient})$).

Hazard Ratios Wald Confidence Intervals

Show Wald-based confidence intervals for the hazard ratios.

Adjustment Variables

Show model output for the adjustment variables.

Adjustment Names

Show a column delineating model types (unadjusted and different adjustment variable sets). Mostly useful when you don't want to show model output for the adjustor variables.

Sample Size

Sample Size

Show the sample size used from each model.

Number Missing, if any

Show the number of observations not used in each model (missing values), only if there are some not used.

Number Missing, always

Show the number of observations not used in each model (missing values), regardless of whether there are some observations not used.

Number of Events

Show the number of events from each model.

Fit Statistics

Concordance

Show the model concordance statistic.

Concordance Standard Error

Show the standard error of the model concordance statistic.

R-Squared

Show a pseudo R-squared value from each model (Nagelkerke's R-squared)

R-Squared Maximum

Show the maximum possible value for the pseudo R-squared value from each model (Nagelkerke's R-squared)

Akaike Information Criterion (AIC)

Show the model Akaike Information Criterion

Bayesian Information Criterion (BIC)

Show the model Bayesian Information Criterion

Log-Likelihood

Show the model log-likelihood value

P-Values

Parameter Estimates (Wald Test)

Show the p-values from the individual parameter Wald tests

Likelihood Ratio Tests (not adjustors)

Show the p-values for each independent variable based on a likelihood ratio test. This compares a model with the independent variable to a model without the independent variable, including any adjustor variables in both models.

Model Score Test

Show the p-value from the overall model score test.

Model Likelihood Ratio Test

Show the p-value from the overall model likelihood ratio test.

Model Wald Test

Show the p-value from the overall model Wald test.

Test Statistics

Parameter z-statistics (Wald Test)

Show the z-statistics from the individual parameter Wald tests.

Model Score Test

Show the overall model score statistic.

Model Likelihood Ratio Test

Show the overall model likelihood ratio test statistic.

Model Wald Test

Show the overall model Wald test statistic.

- ❶ Required R Packages: `arsenal`, `survival`, `dplyr`

Cox, Stratified

Fits a stratified Cox proportional hazards model for time-to-event data with censored observations. This is a Cox model that allows a separate baseline hazard function for each strata level. Model fitting statistics, parameter estimates, and hazard ratios are provided. Options available include the tied time method and model diagnostics. The model is fit using the coxph function in the survival package.

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

Steps

Load the dataset --> Click on the Model Fitting tab in main menu --> Select Regression --> This leads to analysis techniques, choose Cox, Stratified --> There will appear a dialog --> Select the model name, time to event or sensor, events, populate a formula and select Stratification variables in the dialog --> Finally execute the plot and visualise the output in output window.

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

Cox, Stratified

⚠️ 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.

Attributes

Time to event or censor

Time to event for those experiencing the event or time to last follow-up for those not experiencing the event

Events (1=event, 0=censor)

Numerical event indicator; 1=event, 0=censor

Formula Builder

Construct terms to include in the model. Factors, strings, and logical variables will be dummy coded. The provided buttons allow you to specify main effects, full factorial effects (main effects and all interactions with the involved variables), polynomials, specific interactions, and delete terms from the list. Interactions with stratification variables is allowed.

Stratification Variables

Specify one or more stratification variables. These can be numeric, factor, ordered factor, or character variables. The strata divide the subjects into separate groups whereby each group has a distinct baseline hazard function. If multiple stratification variables are given, a separate baseline hazard function is used for every combination of stratification variable levels.

Weights

Numeric variable for observation weights. Useful in situations where each record should not be counted as one observation.

i Required packages: survival, broom, survminer

i Click the R Help button to get detailed R help about the coxph function.

Options

Tied Time Method

Method of breaking tied observed times. Efron is usually the better choice when there aren't many tied times. The exact method can be beneficial if there are many tied times, as in discrete time situations, but can take a little longer for the model to be fit.

Model Diagnostics

If selected, proportional hazards tests and plots will be provided, in addition to a Martingale residual plot.

Linear Regression, Advanced

Builds a linear regression model by creating a formula using the formula builder.

Internally calls function `lm` in stats package. Returns an object called

`BSkyLinearRegression` which is an object of class `lm`.

Displays a summary of the model, coefficient table, Anova table and sum of squares table and plots the following residuals vs. fitted, normal Q-Q, theoretical quantiles, residuals vs. leverage.

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

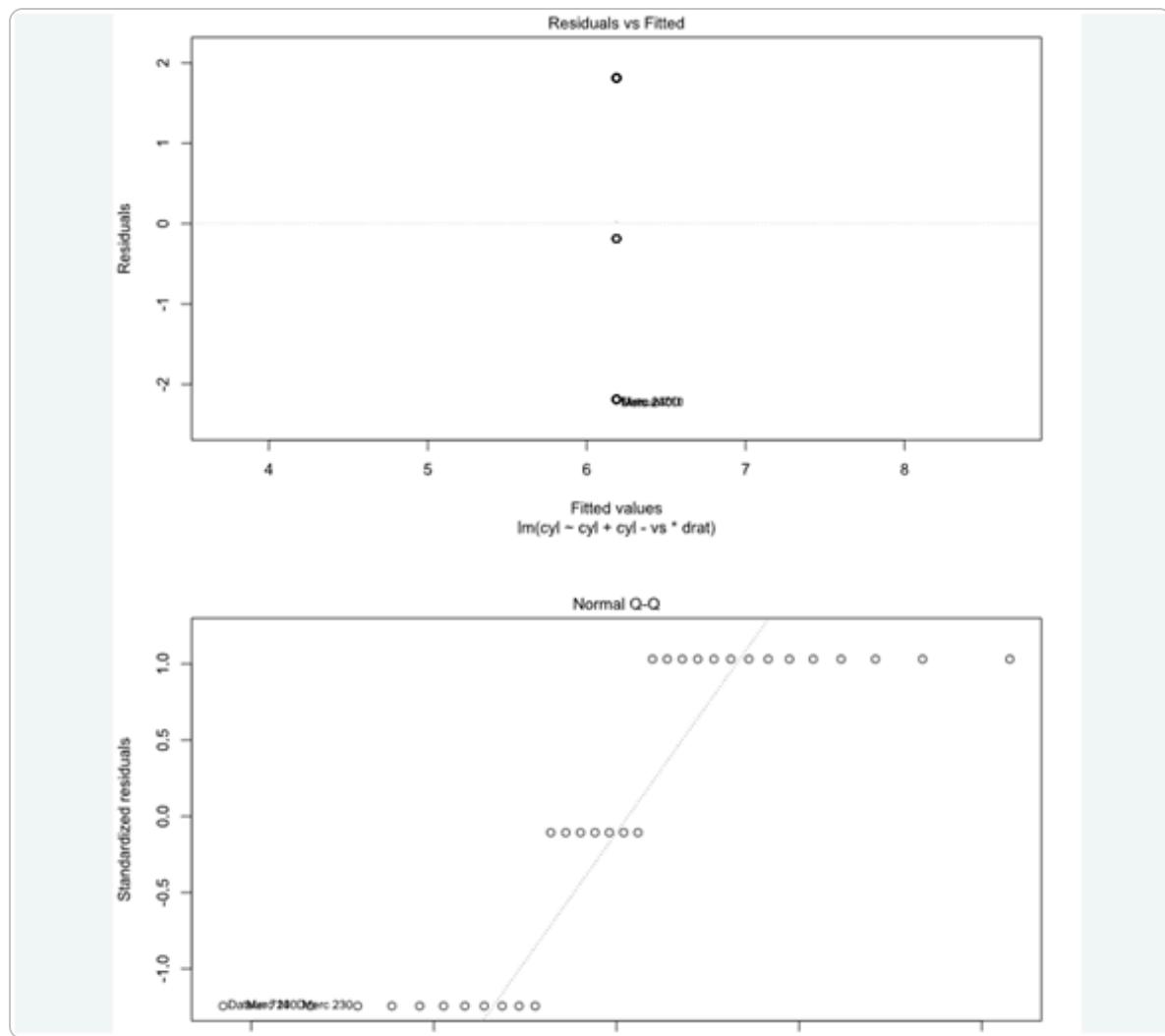
Steps

Load the dataset -> Click on the Model Fitting tab in main menu -> Select Regression -> This leads to analysis techniques, choose Linear, Advanced -> There will appear a dialog -> Select the model name, dependent variables and populate a formula in the dialog -> Check the radio button to display a plot in the output -> Finally execute the plot and visualise the output in output window.

The screenshot shows the BioStat Prime interface. On the left, the 'Linear Regression' dialog is open. It has sections for 'Source variables' (listing columns from the 'mtcars' dataset like mpg, disp, hp, drat, wt, qsec, vs, am, gear, carb), 'Enter model name' (set to 'LinearRegModel1'), 'Dependent variable' (set to 'cyl'), and a 'Formula Builder'. The formula builder shows 'cyl ~ cyl + cyl - vs * drat'. The right side of the screen shows the 'OUTPUT 1' window with the following content:

- Summary:
Model: `lm(formula = cyl ~ cyl + cyl - vs * drat, data = mtcars, na.action = na.exclude)`
 $cyl = \alpha + \epsilon$
 $\hat{cyl} = 6.1875$
- LM Summary:
Residual Std. Error df R-squared Adjusted R-squared
1.7859 31 0 0
- Residuals:
Min 1Q Median 3Q Max
-2.1875 -2.1875 -0.1875 1.8125 1.8125
- Coefficients:
Estimate Std. Error t value Pr(>|t|) 2.5 % 97.5 %
(Intercept) 6.1875 0.3157 19.5987 5.048e-19 *** 5.5436 6.8314
Note:
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ''
- Anova Table:
Df Sum Sq Mean Sq F value Pr(>F)

Linear Regression, Advanced



Linear Regression, Advanced plot

⚠️ 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.

Arguments

depVar

Name of the dependent variable. If we have a dataset cars, with a variable mpg that we want to predict mpg (dependent variable is mpg) enter mpg

indepVars

Names of the dependent variable. If we have a dataset cars, with dependent variable horsepower, enginesize, enter horsepower+enginesize. Categorical variables are automatically dummy coded.

dataset

Name of the dataframe. When you open data frames or datasets e.g. csv, Excel files, SAS files in BioStat Prime, they are named Dataset1, Dataset2, Dataset3 so enter Dataset1

Linear Regression, Basics

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

Steps

Load the dataset → Click on the Model Fitting tab in main menu → Select Regression → This leads to analysis techniques, choose Linear, Basics → There will appear a dialog → Select the model name, dependent variables and independent variable in the dialog → Check the radio buttons to display a plot in the output → Finally execute the plot and visualise the output in output window.

The screenshot shows the BioStat Prime interface. On the left, the 'Linear Regression' dialog is open. It displays a list of source variables: z\\$mpg, z\\$cyl, z\\$hp, z\\$drat, z\\$wt, z\\$am, and z\\$carb. The 'Enter Model Name' field contains 'LinearRegModel1'. The 'Dependent variable' is set to 'z\\$disp'. The 'Independent variable(s)' are 'z\\$vs', 'z\\$gear', and 'z\\$qsec'. Below these fields are two checked options: 'Ignore intercept' and 'Plot residuals vs fitted, normal Q-Q, scale-location and residuals vs leverage'. At the bottom, there is a note: 'Specify a variable with weights' followed by a '→' button. On the right, the 'OUTPUT 1' window shows the R code used to fit the model:

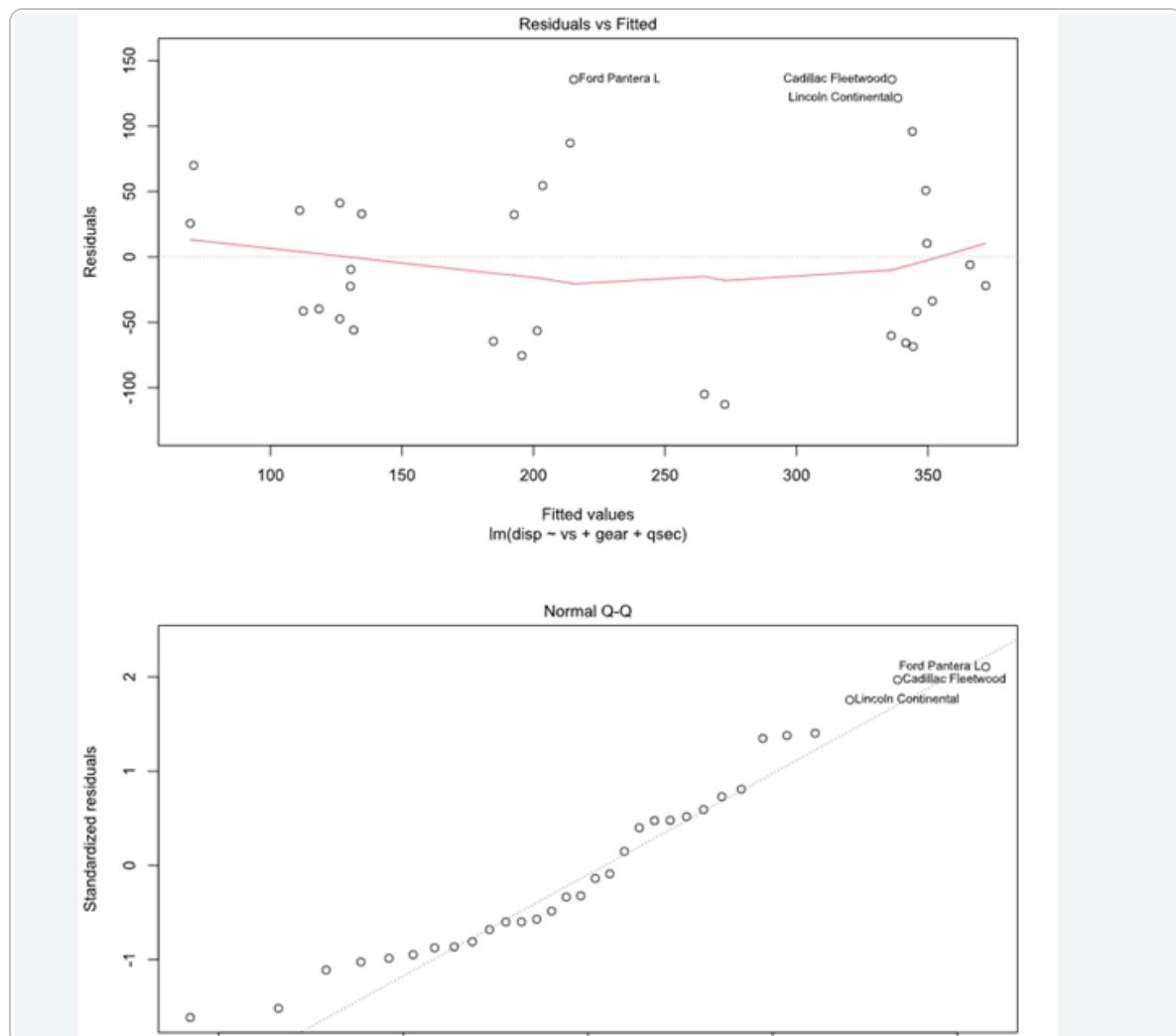
```
disp = α + β1(vs) + β2(gear) + β3(qsec) + ε
```

 and the resulting output:

```
disp = 839.9253 - 112.5096(vs) - 84.6446(gear) - 13.8863(qsec)
```

. The 'LM Summary' table includes columns: Residual Std. Error, df, R-squared, Adjusted R-squared, F-statistic, numdf, dendf, and p-value. The values are: 72.3956, 28, 0.6918, 0.6588, 20.9518, 3, 28, 2.5497e-07 ***. The 'Residuals' table shows statistics for Min, 1Q, Median, 3Q, and Max. The 'Coefficients' table lists the Estimate, Std. Error, t value, Pr(>|t|), 2.5 %, and 97.5 % for each variable. The values are: (Intercept) 839.9253, 271.2597, 3.0966, 0.0044 **, 284.5159, 1395.5550; vs -112.5096, 46.5902, -2.4149, 0.0225 *, -207.9453, -170740; gear -84.6446, 21.7447, -3.8926, 0.0006 ***, -129.1867, -40.1025.

Linear Regression, Basics



Linear Regression, Basics plot

⚠️ 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.

Arguments

depVar

Name of the dependent variable. If we have a dataset cars, with a variable mpg that we want to predict mpg (dependent variable is mpg) enter mpg

indepVars

Names of the dependent variable. If we have a dataset cars, with dependent variable horsepower, enginesize, enter horsepower+enginesize. Categorical variables are automatically dummy coded.

dataset

Name of the dataframe. When you open data frames or datasets e.g. csv, Excel files, SAS files in BioStat Prime, they are named Dataset1, Dataset2, Dataset3 so enter Dataset1

Linear Regression (Legacy)

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

Steps

Load the dataset -> Click on the Model Fitting tab in main menu -> Select Regression -> This leads to analysis techniques, choose Linear, Basics(Legacy) -> There will appear a dialog -> Select the model name, dependent variables and independent variable in the dialog -> Check the radio buttons to display a plot in the output -> Finally execute the plot and visualise the output in output window.

The screenshot shows the BioStat Prime interface. The main window is titled 'Linear Regression' and contains a configuration dialog. In the dialog, the 'Source variables' list includes 'z\\$ mpg', 'z\\$ disp', 'z\\$ hp', 'z\\$ drat', 'z\\$ wt', 'z\\$ qsec', and 'z\\$ am'. The 'Enter Model Name' field is set to 'LinearRegModel1'. The 'Dependent variable' is 'z\\$ cyl'. The 'Independent variable(s)' list includes 'z\\$ vs', 'z\\$ carb', and 'z\\$ gear', with 'z\\$ gear' highlighted in green. Below the dialog, two checkboxes are visible: 'Ignore intercept' (unchecked) and 'Plot residuals vs fitted, normal Q-Q , scale-location and residuals vs leverage' (checked). The 'Specify a variable with weights' section is empty. To the right of the dialog, the 'OUTPUT 1' window displays the results of the regression analysis. It shows the regression equation $y = \alpha + \beta_1(\text{vs}) + \beta_2(\text{carb}) + \beta_3(\text{gear}) + \epsilon$ and the estimated model $\hat{y} = 10.1888 - 1.7338(\text{vs}) + 0.4252(\text{carb}) - 1.2037(\text{gear})$. The 'LM Summary' table provides statistical details:

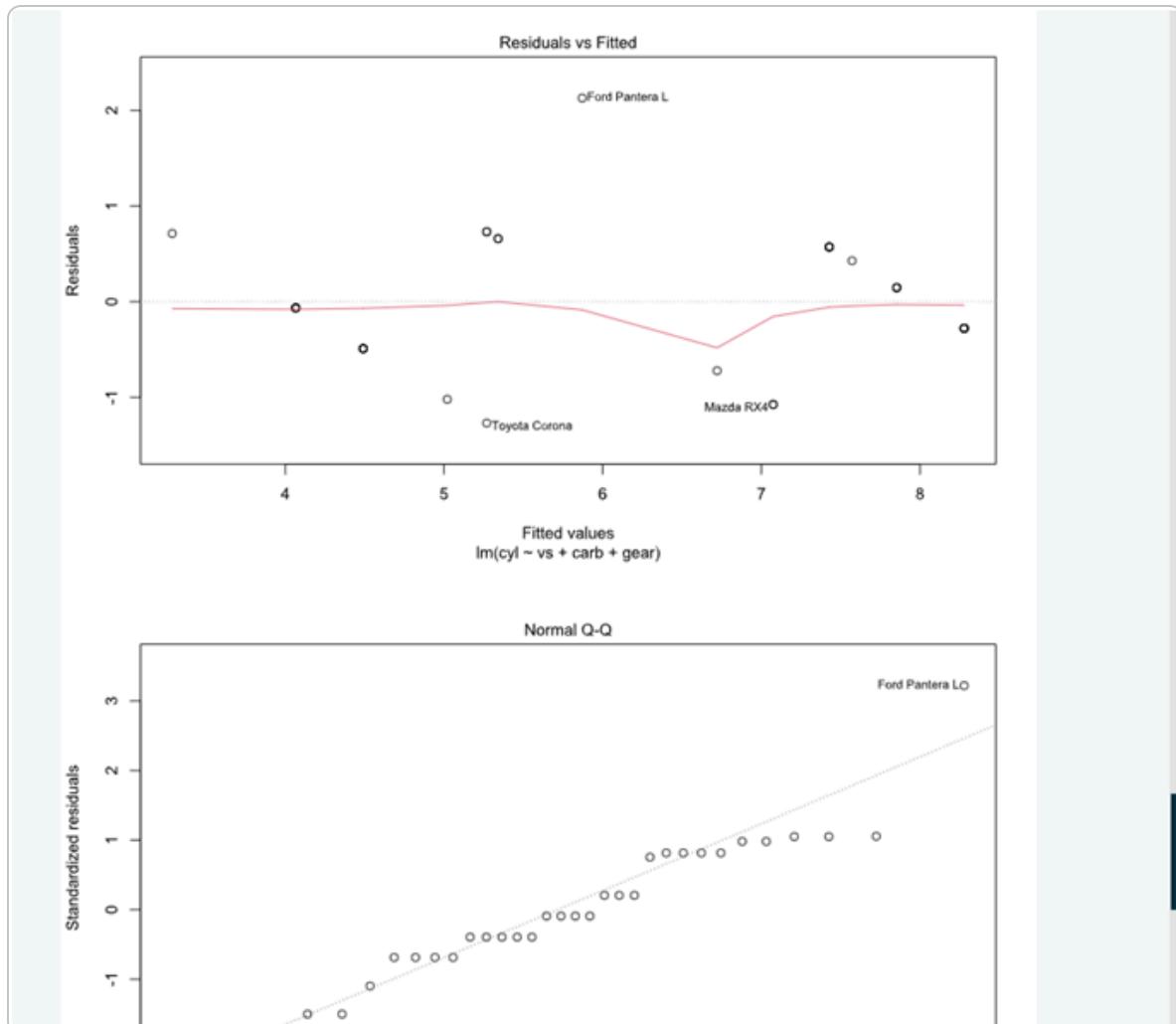
Residual Std. Error	df	R-squared	Adjusted R-squared	F-statistic	numdf	dendf	p-value
0.7413	28	0.8444	0.8277	50.6497	3	28	1.9540e-11 ***

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

The 'Residuals' table shows quantiles: Min, 1Q, Median, 3Q, Max. The 'Coefficients' table shows the following data:

	Estimate	Std. Error	t value	Pr(> t)	2.5 %	97.5 %
(Intercept)	10.1888	0.6803	14.9765	6.7873e-15 ***	8.7952	11.5824
vs	-1.7338	0.5616	-4.7955	4.8513e-05 ***	-2.4744	-0.9932

Linear Regression (Legacy)



Linear Regression Legacy plot

Arguments

depVar

Name of the dependent variable. If we have a dataset cars, with a variable mpg that we want to predict mpg (dependent variable is mpg) enter mpg

indepVars

Names of the dependent variable. If we have a dataset cars, with dependent variable horsepower, enginesize, enter horsepower+enginesize. Categorical variables are

automatically dummy coded.

dataset

Name of the dataframe. When you open data frames or datasets e.g. csv, Excel files, SAS files in BioStat Prime, they are named Dataset1, Dataset2, Dataset3 so enter Dataset1

Linear Regression, multiple models

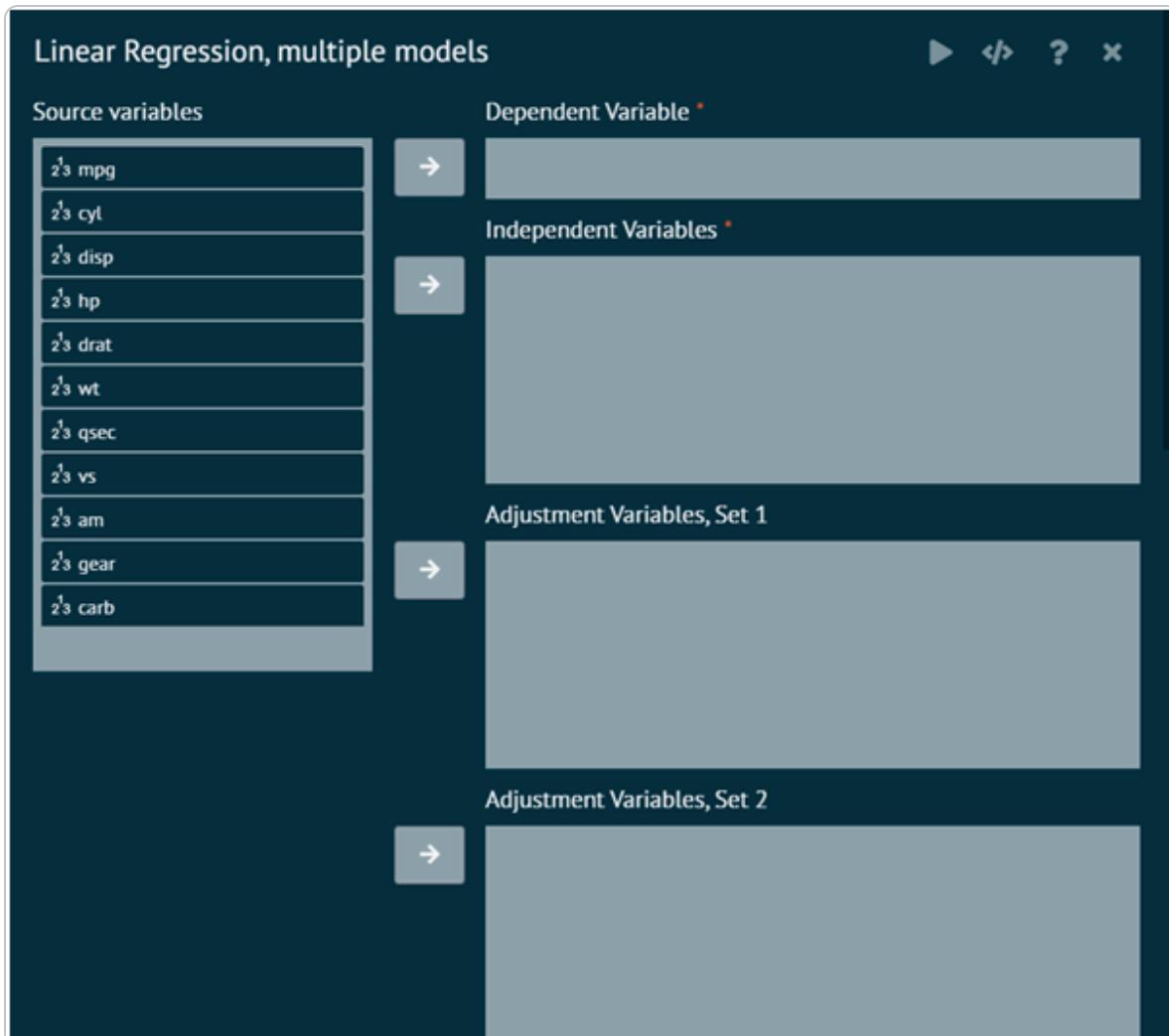
This creates a table containing results from linear regression models for a given dependent variable. Separate linear regression models will be fit for each independent variable, optionally adjusted for a set of additional variables. If a strata variable is specified, separate models will be fit for each of the stratification variable values.

As an example, if no adjustor or stratification variables are specified, then the table will include all univariate models for the list of independent variables. Various statistics from each model can be output.

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

Steps

Load the dataset -> Click on the Model Fitting tab in main menu -> Select Regression -> This leads to analysis techniques, choose Linear Regression, multiple models -> There will appear a dialog -> Select dependent variables, independent variable and Adjustment Variables, Sets in the dialog -> Finally execute the plot and visualise the output in output window.



Linear Regression multiple models

Attributes

Dependent Variable

Dependent variable for each linear regression model. The variable class must be a numeric type.

Independent Variables

Independent variables to include in the models. The variable classes can be a numeric type, character, factor, or ordered factor.

Adjustment Variables (Sets 1-5)

Optional variables to be included in a model with the independent variables. The variable classes can be a numeric type, character, factor, or ordered factor.

Specifying more than one set of adjustor variables will provide separate models with each set of adjustor variables.

Strata

Optional stratification variable. Separate models will be fit for the subset defined by each of the stratification variable values. The variable class can be character, numeric, factor, or ordered factor.

Weights

Optional case-weights to be used in the models. Specifying a weights variable will fit weighted regression models.

Digits After Decimal

Continuous Values

The number of decimal places to show for all continuous values in the table (default=4)

P-Values

The number of decimal places to show for all p-values in the table (default=4)

Options

Parameter Estimates

Parameter Estimates

Show parameter estimates (coefficients) from each model.

Standard Errors

Show standard errors of the parameter estimates.

Confidence Interval Level

Level for the parameter estimate confidence intervals (default=0.95).

Intercepts

Show the y-intercepts from each model.

Adjustment Variables

Show model output for the adjustment variables.

Standardized Estimates

Show standardized parameter estimates from each model. These are the parameter estimates corresponding to a standardized version of all continuous variables ((value - mean) / standard deviation).

Confidence Intervals

Show confidence intervals for the parameter estimates.

Adjustment Names

Show a column delineating model types (unadjusted and different adjustment variable sets). Mostly useful when you don't want to show model output for the adjustor variables.

Sample Size

Sample Size

Show the sample size used from each model.

Number Missing, if any

Show the number of observations not used in each model (missing values), only if there are some not used.

Number Missing, always

Show the number of observations not used in each model (missing values), regardless of whether there are some observations not used.

Fit Statistics

R-Squared

Show the model R-Squared value

Adjusted R-Squared

Show the model adjusted R-Squared value

Akaike Information Criterion (AIC)

Show the model Akaike Information Criterion

Bayesian Information Criterion (BIC)

Show the model Bayesian Information Criterion

Log-Likelihood

Show the model log-likelihood value

P-Values

Parameter t-statistics

Show the p-values from the individual parameter t-tests

Model F-test

Show the p-value from the overall model F-test

Likelihood Ratio Tests (not adjustors)

Show the p-values for each independent variable based on a likelihood ratio test. This compares a model with the independent variable to a model without the independent variable, including any adjustor variables in both models.

Test Statistics

Parameter t-statistics

Show the t-statistics from the individual parameter t-tests

Model F-Test

Show the F statistic from the overall model F-test



Required R Packages: `arsenal`, `dplyr`

Logistic Regression, Advanced

Logistic regression is a data analysis technique that uses mathematics to find the relationships between two data factors. It then uses this relationship to predict the value of one of those factors based on the other. The prediction usually has a finite number of outcomes, like yes or no.

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

Steps

Load the dataset -> Click on the Model Fitting tab in main menu -> Select Regression -> This leads to analysis techniques, choose Logistics, Advanced -> There will appear a dialog -> Select the model name, dependent variables and independent variable in the dialog -> Check the radio buttons to display a plot in the output -> Finally execute the plot and visualise the output in output window.

The screenshot shows a software interface for Logistic Regression. On the left, a list of source variables is displayed, including mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, and carb. In the center, the model name is set to "Logistic1". Below it, the dependent variable is selected. To the right, a "Formula Builder" section is shown with a grid of operators (+, -, *, /), parentheses, and other symbols. Below the grid, dropdown menus show "All 2 ways" and "Polynomial terms 2", and buttons for "df for splines 5" and "Polynomial degree 5". At the bottom of the builder, there are options for "B-spline" and "Natural spline" and "Orthogonal polynomial" and "Raw polynomial". A note at the bottom states: "Plot residuals vs fitted normal O-O scale-location and residuals".

Logistic Regression Advanced



NOTE

When specifying a variable containing weights, be aware that since we use the option `na.exclude` to build the model, all NA values are automatically removed from the dependent and independent variables.

This can cause a mismatch as NA values are NOT automatically removed from the weighting variable. In this situation you will see the error variable lengths differ (found for (weights))

- i** To address this error go to Variables>Missing Values>Remove NAs and select the dependent, independent variables and the weighting variable to remove missing values from and rebuild the model.

Arguments

depVar

Name of the dependent variable. If we have a dataset cars, with a variable class that we want to predict (dependent variable is class) enter class

indepVars

Names of the independent variable, separated by +. If we have a dataset cars, with independent variable horsepower, enginesize, specify horsepower+enginesize). Categorical variables are automatically dummy coded.

data

Name of the dataframe. When you open data frames or datasets e.g. csv, Excel files, SAS files in BioStat Prime, they are named Dataset1, Dataset2, Dataset3 So enter data=Dataset1.

Logistic Regression, Basic

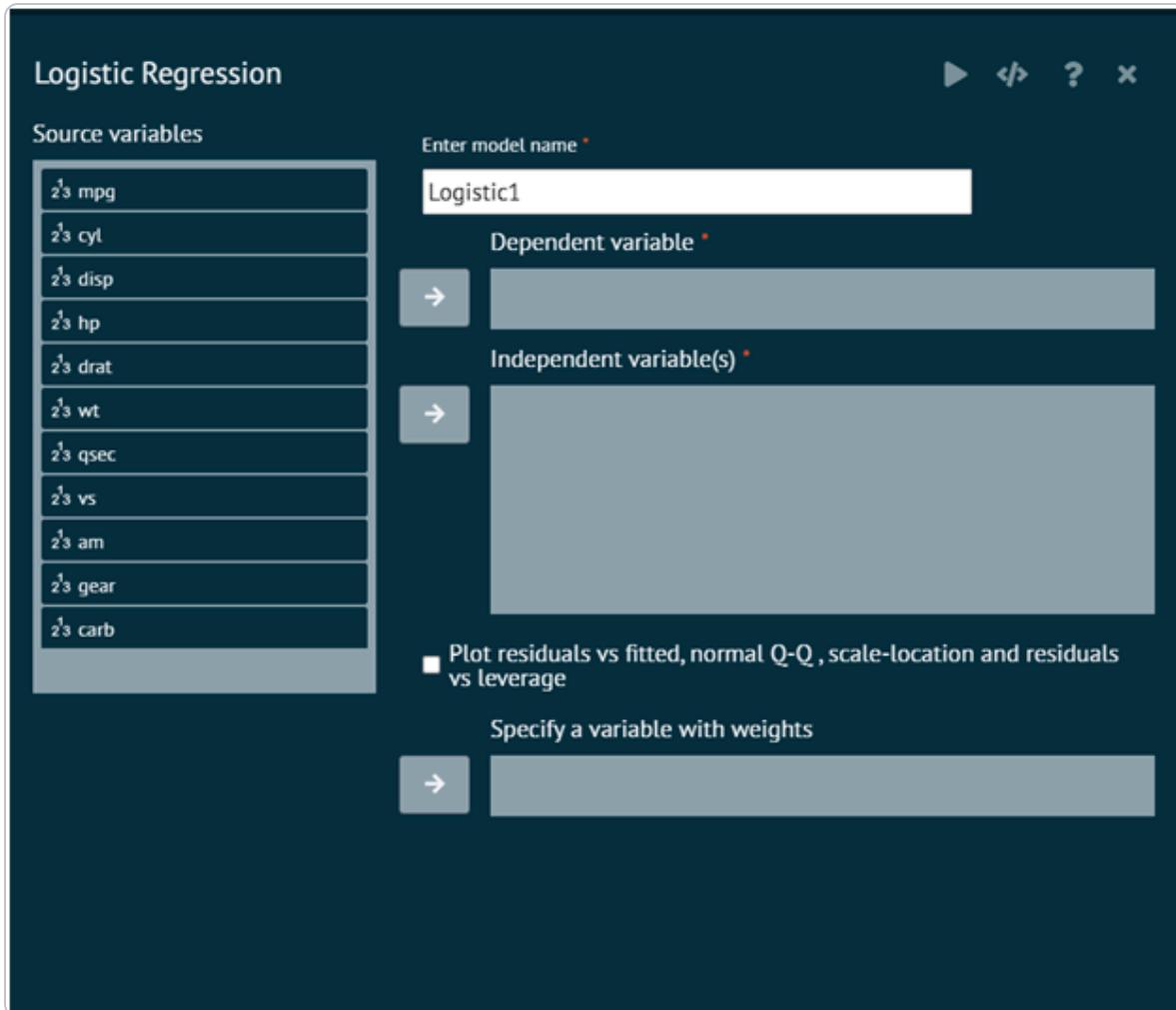
This function builds a binary logistic regression model using a formula builder. BioStat Prime uses `glm` function passing the parameter `family =binomial(link='logit')`. BioStat Prime displays a summary of the model, analysis of variance tables and McFadden R2. User can score the model by selecting the model created on the top right hand corner of the main application screen and select the Score button. User can choose to display a confusion matrix and a ROC curve

i The default model name is Logistic1 which user can change.

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

Steps

Load the dataset -> Click on the Model Fitting tab in main menu -> Select Regression -> This leads to analysis techniques, choose Logistics, Basic -> There will appear a dialog -> Select the model name, dependent variables and populate the formula builder in the dialog -> Check the radio buttons to display a plot in the output -> Finally execute the plot and visualise the output in output window.



Logistic Regression Basic

⚠️ 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.

Arguments

depVar

Name of the dependent variable. If we have a dataset cars, with a variable class that we want to predict (dependent variable is class) enter class

indepVars

Names of the independent variable, separated by +. If we have a dataset cars, with independent variable horsepower, enginesize, specify horsepower+enginesize). Categorical variables are automatically dummy coded.

data

Name of the dataframe. When you open data frames or datasets e.g. csv, Excel files, SAS files in BioStat Prime, they are named Dataset1, Dataset2, Dataset3 So enter data=Dataset1.



NOTE

When specifying a variable containing weights, be aware that since we use the option na.exclude to build the model, all NA values are automatically removed from the dependent and independent variables.

This can cause a mismatch as NA values are NOT automatically removed from the weighting variable. In this situation you will see the error variable lengths differ (found for (weights))



- To address this error go to Variables>Missing Values>Remove NAs and select the dependent, independent variables and the weighting variable to remove

missing values from and rebuild the model.

Logistic Regression, Conditional

This function fits a conditional logistic regression model, which is similar to standard logistic regression, but incorporates a stratification variable. Common applications of this model are in matched case-control, matched cohort, and nested case-control studies.

The output includes the number of strata used in the analysis, outcome frequency patterns within strata, various model summary statistics, parameter estimates, and odds ratios.

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

Steps

Load the dataset -> Click on the Model Fitting tab in main menu -> Select Regression -> This leads to analysis techniques, choose Logistics, Conditional -> There will appear a dialog -> Select the model name, dependent variables, Strata, estimation method and populate the formula builder in the dialog -> Finally execute the plot and visualise the output in output window.

Logistic, Conditional

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: CondLogisticModel1

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

Logistic Regression Conditional

⚠ 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.

Attributes

Dependent Variable

This is the variable indicating the event for each subject, with 1 signifying an event, and 0 signifying a non-event. This variable must be numeric.

Formula Builder

Specify the desired model terms.

Strata

Specify the stratification variable. Can be numeric, character, or a nominal/ordinal factor.

Estimation Method

Specify the model estimation method. The "Exact" method is the default. As long as there are not too many ways to select events within strata, this should be the option chosen. The estimation may take some time or lead to errors if many combinations can result within some strata, say 100 events out of 500 subjects. The Efron and Breslow approximations are adequate in these cases, with Efron being preferred. See the references in the R help for details.



Required R Packages: survival, broom, arsenal, dplyr

Logistic Regression, multiple models

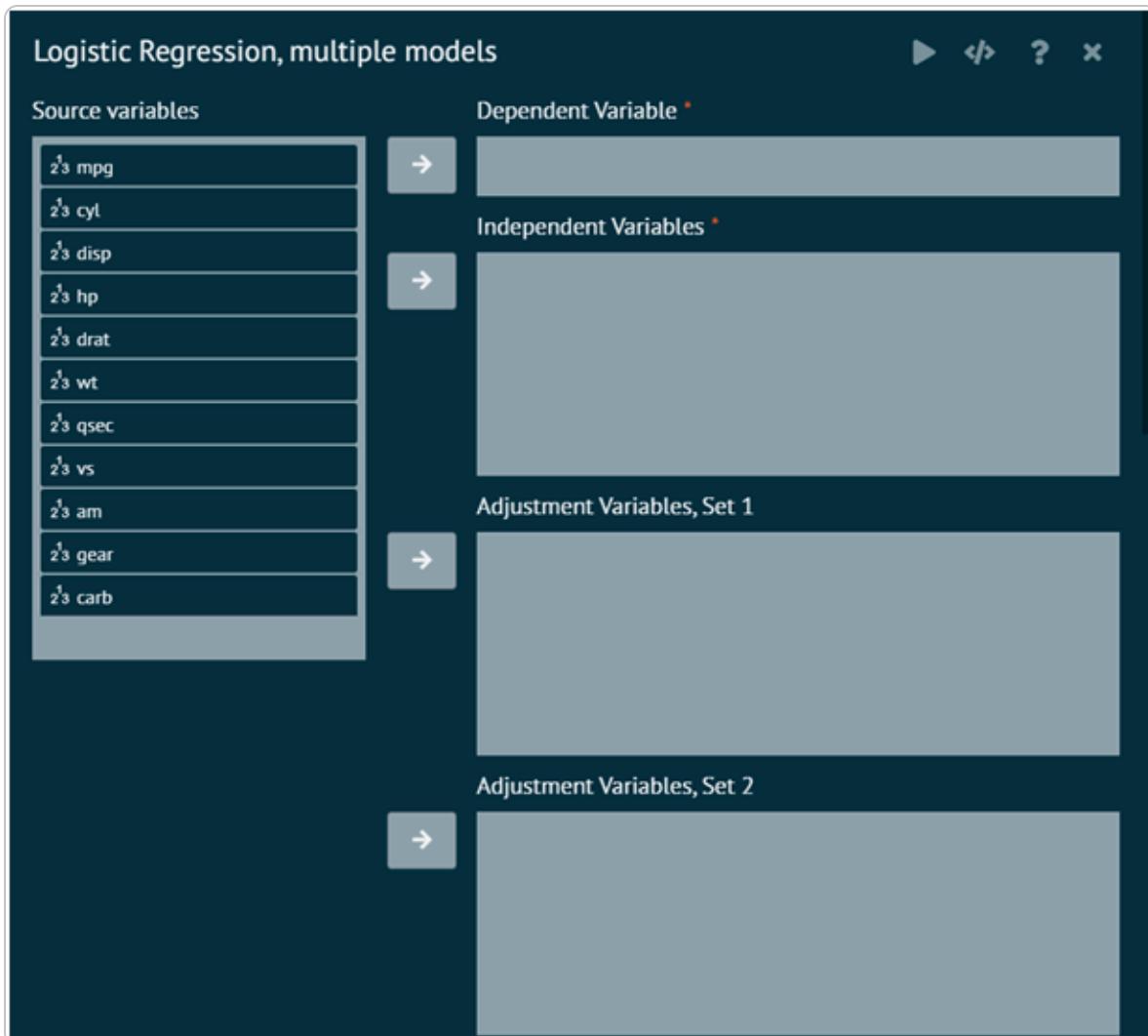
This creates a table containing results from logistic regression models for a given dependent variable. Separate logistic regression models will be fit for each independent variable, optionally adjusted for a set of additional variables. If a strata variable is specified, separate models will be fit for each of the stratification variable values.

As an example, if no adjustor or stratification variables are specified, then the table will include all univariate models for the list of independent variables. Various statistics from each model can be output.

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

Steps

Load the dataset -> Click on the Model Fitting tab in main menu -> Select Regression -> This leads to analysis techniques, choose Logistic Regression, multiple models -> There will appear a dialog -> Select the model name, dependent variables, and Adjustment Variables, Sets in the dialog -> Finally execute the plot and visualise the output in output window.



Logistic Regression multiple models

Attributes

Dependent Variable

Dependent variable for each logistic regression model. The variable class must be a numeric type or factor.

Independent Variables

Independent variables to include in the models. The variable classes can be a numeric type, character, factor, or ordered factor.

Adjustment Variables (Sets 1-5)

Optional variables to be included in a model with the independent variables. The variable classes can be a numeric type, character, factor, or ordered factor.

Specifying more than one set of adjustor variables will provide separate models with each set of adjustor variables.

Strata

Optional stratification variable. Separate models will be fit for the subset defined by each of the stratification variable values. The variable class can be character, numeric, factor, or ordered factor.

Weights

Optional case-weights to be used in the models. Specifying a weights variable will fit weighted regression models.

Digits After Decimal

Continuous Values

The number of decimal places to show for all continuous values in the table (default=4)

P-Values

The number of decimal places to show for all p-values in the table (default=4)

Odds Ratios

The number of decimal places to show for all odds ratios in the table (default=4)

Options

Parameter Estimates and Odds Ratios

Parameter Estimates

Show parameter estimates (coefficients) from each model.

Standard Errors

Show standard errors of the parameter estimates.

Confidence Interval Level

Level for the parameter estimate and odds ratio confidence intervals (default=0.95).

Parameter Wald Confidence Intervals

Show Wald-based confidence intervals for the parameter estimates.

Parameter Profile Likelihood Confidence Intervals

Show profile likelihood-based confidence intervals for the parameter estimates.

Odds Ratios

Show odds ratios for each parameter estimate ($\exp(\text{coefficient})$).

Odds Ratios Wald Confidence Intervals

Show Wald-based confidence intervals for the odds ratios.

Odds Ratios Profile Likelihood Confidence Intervals

Show profile likelihood-based confidence intervals for the odds ratios.

Intercepts

Show the intercepts from each model.

Adjustment Variables

Show model output for the adjustment variables.

Adjustment Names

Show a column delineating model types (unadjusted and different adjustment variable sets). Mostly useful when you don't want to show model output for the adjustor variables.

Sample Size

Sample Size

Show the sample size used from each model.

Number Missing, if any

Show the number of observations not used in each model (missing values), only if there are some not used.

Number Missing, always

Show the number of observations not used in each model (missing values), regardless of whether there are some observations not used.

Fit Statistics

Concordance (AUC)

Show the model concordance statistic. This is equivalent to the area under the curve (AUC) from a Receiver Operating Characteristic (ROC) curve.

Akaike Information Criterion (AIC)

Show the model Akaike Information Criterion

Bayesian Information Criterion (BIC)

Show the model Bayesian Information Criterion

Log-Likelihood

Show the model log-likelihood value

Null Deviance

Show the model null deviance value

Deviance

Show the model deviance value

Null Model Degrees of Freedom

Show the degrees of freedom from a model with no predictors in it.

Residual Degrees of Freedom

Show the residual degrees of freedom (total degrees of freedom minus the number of parameters fit).

P-Values

Parameter Estimates (Wald Test)

Show the p-values from the individual parameter Wald tests

Likelihood Ratio Tests (not adjustors)

Show the p-values for each independent variable based on a likelihood ratio test. This compares a model with the independent variable to a model without the independent variable, including any adjustor variables in both models.

Test Statistics

Parameter z-statistics (Wald Test)

Show the z-statistics from the individual parameter Wald tests

 Required R Packages: `arsenal`, `dplyr`

Multinomial Logit

This function fits multinomial log-linear models via neural networks.

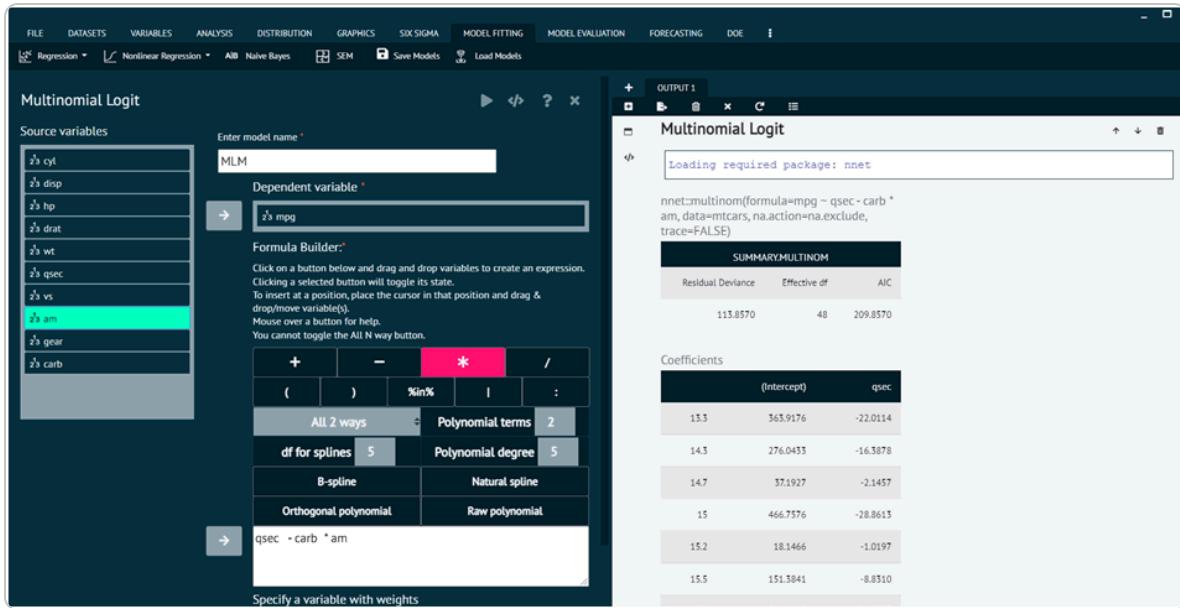
To analyse it in BioStat Prime user must follow the steps as given.

Steps

Load the dataset -> Click on the Model Fitting tab in main menu -> Select Regression -> This leads to analysis techniques, choose Multinomial Logit -> There will appear a dialog -> Select the model name, dependent variables, and populate the formula builder in the dialog -> Finally execute the plot and visualise the output in output window.

⚠️ 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.



Multinomial Logit

Arguments

formula

a formula expression as for regression models, of the form `response ~ predictors`. The response should be a factor or a matrix with K columns, which will be interpreted as counts for each of K classes. A log-linear model is fitted, with coefficients zero for the first class. An offset can be included: it should be a numeric matrix with K columns if the response is either a matrix with K columns or a factor with $K \geq 2$ classes, or a numeric vector for a response factor with 2 levels. See the documentation of `formula()` for other details.

data

an optional data frame in which to interpret the variables occurring in `formula`.

weights

optional case weights in fitting.

subset

expression saying which subset of the rows of the data should be used in the fit. All observations are included by default.

na.action

a function to filter missing data.

contrasts

a list of contrasts to be used for some or all of the factors appearing as variables in the model formula.

Hess

logical for whether the Hessian (the observed/expected information matrix) should be returned.

summ

integer; if non-zero summarize by deleting duplicate rows and adjust weights.
Methods 1 and 2 differ in speed (2 uses C); method 3 also combines rows with the same X and different Y, which changes the baseline for the deviance.

censored

If Y is a matrix with K columns, interpret the entries as one for possible classes, zero for impossible classes, rather than as counts.

model

logical. If true, the model frame is saved as component model of the returned object.

... additional arguments for nnet

Ordinal Regression

This function fits a logistic or probit regression model to an ordered factor response. The default logistic case is proportional odds logistic regression, after which the function is named.

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

Steps

Load the dataset -> Click on the Model Fitting tab in main menu -> Select Regression -> This leads to analysis techniques, choose Ordinal Regression -> There will appear a dialog -> Select the model name, dependent variables, and populate the formula builder in the dialog -> Finally execute the plot and visualise the output in output window.

Ordinal Regression

This model is what Agresti (2002) calls a cumulative link model. The basic interpretation is as a coarsened version of a latent variable Y_i which has a logistic or normal or extreme-value or Cauchy distribution with scale parameter one and a linear model for the mean. The ordered factor which is observed is which bin Y_i falls into with breakpoints $\zeta_0 = -\infty < \zeta_1 < \dots < \zeta_K = \infty$

This leads to the model $\text{logit } P(Y \leq k | x) = \zeta_k - \eta$ with logit replaced by probit for a normal latent variable, and η being the linear predictor, a linear function of the explanatory variables (with no intercept). Note that it is quite common for other software to use the opposite sign for η (and hence the coefficients β).

In the logistic case, the left-hand side of the last display is the log odds of category k or less, and since these are log odds which differ only by a constant for different k , the odds

are proportional. Hence the term proportional odds logistic regression.

The log-log and complementary log-log links are the increasing functions $F^{-1}(p) = -\log(-\log(p))$ and $F_{-1}(p) = \log(-\log(1-p))$; some call the first the ‘negative log-log’ link. These correspond to a latent variable with the extreme-value distribution for the maximum and minimum respectively.

A proportional hazards model for grouped survival times can be obtained by using the complementary log-log link with grouping ordered by increasing times. predict, summary, vcov, anova, model.frame and an extractAIC method for use with stepAIC (and step). There are also profile and confint methods.

Arguments

formula

a formula expression as for regression models, of the form response ~ predictors. The response should be a factor (preferably an ordered factor), which will be interpreted as an ordinal response, with levels ordered as in the factor. The model must have an intercept: attempts to remove one will lead to a warning and be ignored. An offset may be used. See the documentation of formula for other details.

data

an optional data frame in which to interpret the variables occurring in formula.

weights

optional case weights in fitting. Default to 1.

start

initial values for the parameters. This is in the format c(coefficients, zeta): see the Values section.

... additional arguments to be passed to optim, most often a control argument.

subset

expression saying which subset of the rows of the data should be used in the fit. All observations are included by default.

na.action

a function to filter missing data.

contrasts

a list of contrasts to be used for some or all of the factors appearing as variables in the model formula.

Hess

logical for whether the Hessian (the observed information matrix) should be returned. Use this if you intend to call summary or vcov on the fit.

model

logical for whether the model matrix should be returned.

method

logistic or probit or (complementary) log-log or cauchit (corresponding to a Cauchy latent variable).

Quantile Regression

This function fits a quantile regression model, which models a desired quantile (i.e. percentile) of the outcome variable. A typical quantile to model is 0.5, i.e. the median. A model summary and parameter estimates with 95% confidence intervals are provided.

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

Steps

Load the dataset -> Click on the Model Fitting tab in main menu -> Select Regression -> This leads to analysis techniques, choose Quantile Regression -> There will appear a dialog -> Select the model name, dependent variables, estimation method, standard error method and populate the formula builder in the dialog -> Finally execute the plot and visualise the output in output window.

Quantile Regression

To compare quantile regression model slopes, see "Model Evaluation > Compare > Quant Reg Models"

Source variables

2's mpg
2's cyl
2's disp
2's hp
2's drat
2's wt
2's qsec
2's vs
2's am
2's gear
2's 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

Quantile Regression

Attributes

Enter Model Name

the desired name of the model

Dependent Variable

Specify the dependent variable for the model. The desired quantile of this variable will be modeled. This must be numeric.

Formula Builder

Specify the model terms using formula notation. Numeric, factor, ordered factor, and character variables are allowed. Character variables will be coerced to factors.

Quantile (0-1)

Specify the desired quantile to model for the dependent variable. 0.5 (the median) is the default and is a typical quantity.

Estimation Method

Specify the estimation method for the model parameters. The Barrodale and Roberts method is the default and is efficient for models with several thousand observations. The Frisch-Newton and the Frisch-Newton, preprocessing approach might be advantageous for large and very large problems, respectively, especially in cases with a small number of estimated parameters. For large sample sizes with a large number of parameters, the Frisch-Newton, sparse method may be needed. See the references in the R Help for details.

Standard Error Method

Specify the method used to estimate standard errors and confidence intervals. The Rank method provides confidence intervals only, can be slow to run for larger sample sizes ($n > 1000$), and is based on inverting a rank test. The IID method assumes the errors are independent and identically distributed (iid). The NID method presumes local linearity in the quantile and computes a sandwich estimate using a local estimate of sparsity. The Kernel method uses a kernel estimate of the sandwich. The Bootstrap method uses a re-sampling bootstrap approach to estimate the standard errors. See the references in the R Help for details.

Bootstrap Samples

Desired number of bootstrap samples for the bootstrap standard error approach.
The default is 2000 samples.

i Required R Packages: quantreg, broom

Non-Linear Regression

Nonlinear regression is a statistical technique that uses a nonlinear function to model the relationship between a dependent variable and one or more independent variables.

Dose Response Curve

Analysis of dose-response data through a suite of flexible and versatile model fitting and after-fitting analysis functions to estimate parameters with the various dose response equation function

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

Steps

Load the dataset -> Click on the Model Fitting tab in main menu -> Select Non-Linear Regression -> This leads to analysis techniques, choose Dose Response Curve -> There will appear a dialog -> Select the NLS Model name, Response(dependent) variables, Dose, concentration, enzyme etc, in the dialog -> Fill in the other options. -> Finally execute the plot and visualise the output in output window.

Dose Response Curve Model

DRC Model name *

Source variables

2's mpg
2's cyl
2's disp
2's hp
2's drat
2's wt
2's qsec
2's vs
2's am
2's gear
2's 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

→

Dose Response Curve

Non-Linear Least Square(NLS) Model

This function performs nonlinear regression. Build or type in any equation (formula) to build the nonlinear regression model. Determine the nonlinear (weighted) least-squares estimates of the parameters of the nonlinear model. Analyze model fit with graphs.

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

Steps

Load the dataset → Click on the Model Fitting tab in main menu → Select Non-Linear Regression → This leads to analysis techniques, choose Non-Linear Least Square(NLS) Model → There will appear a dialog → Select the DRC model name,

Response variables in the dialog -> Fill in the other options. -> Finally execute the plot and visualise the output in output window.

NLS Model name *

NLS_Model

Source variables

- 2's mpg
- 2's cyl
- 2's disp
- 2's hp
- 2's drat
- 2's wt
- 2's qsec
- 2's vs
- 2's am
- 2's gear
- 2's 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 >
+	-	*	/	^a	
sqrt	log	log10	log2		
mod	abs		exp		

Create an expression here:

A variable used as weight (Y) with a power value

Non-Linear Least Square(NLS) Model

Polynomial Regression Model

This dialog is used to compute and fit an orthogonal polynomial model with specified degree

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

Steps

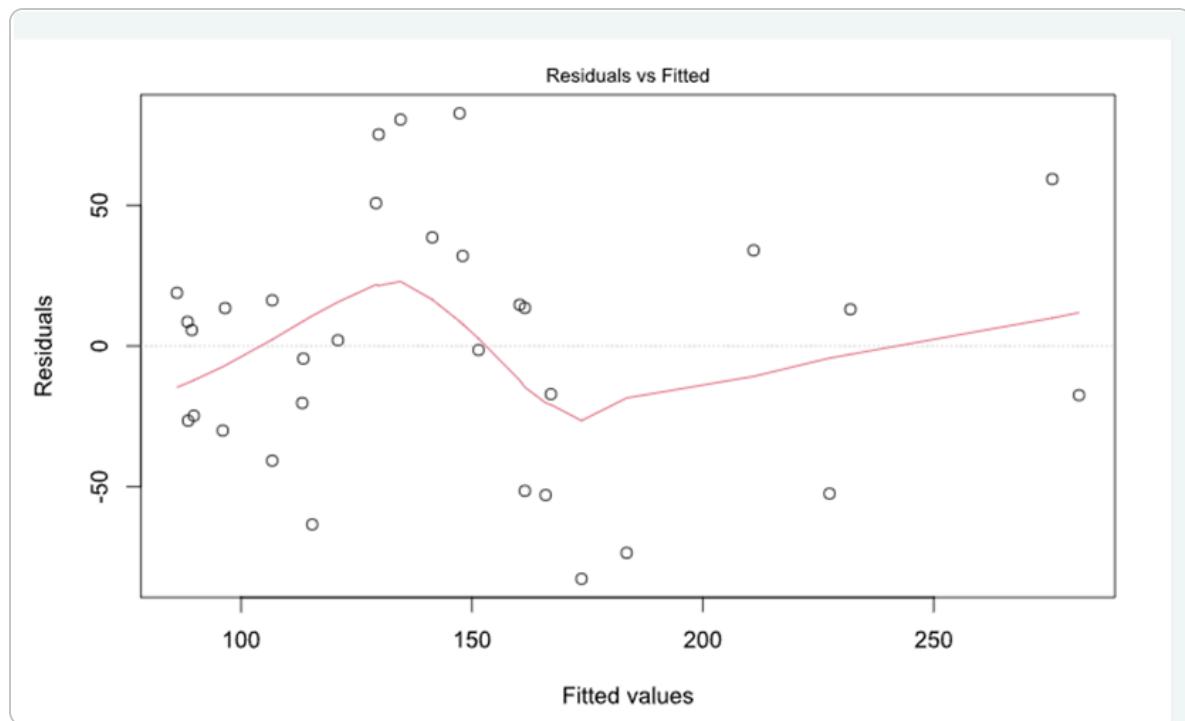
Load the dataset -> Click on the Model Fitting tab in main menu -> Select Non-Linear Regression -> This leads to analysis techniques, choose Polynomial Regression Model -> There will appear a dialog -> Select the Polynomial Regression model name, Dependent (e.g. response) Variable, Independent (e.g. dose, concentration,..) Variable, The degree of the polynomial equation in the dialog -> Fill in the other options. -> Finally execute the plot and visualise the output in output window.

The screenshot shows a software interface with a top navigation bar containing FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION, FORECASTING, and DOE. The MODEL FITTING tab is selected. A dropdown menu shows 'Regression' (selected), 'Nonlinear Regression', 'All', 'Naive Bayes', 'SEM', 'Save Models', and 'Load Models'. Below the navigation bar is a sub-menu for 'Polynomial Regression Model' with fields for 'Model name' (set to 'Poly_Model'), 'Dependent (e.g. response) Variable' (set to 'hp'), 'Independent (e.g. dose, concentration,..) Variable' (set to 'qsec'), and a dropdown for 'The degree of the polynomial equation' (set to '2'). There is also a checkbox for 'Compute additional Polynomial models to compare' with values '3,4' listed. To the right, an 'OUTPUT 1' window displays the results of the model fit. It includes a summary table:

	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 ***

Below the table are sections for 'Residuals' and 'Coefficients'.

Polynomial Regression Model



Polynomial Regression Model plot

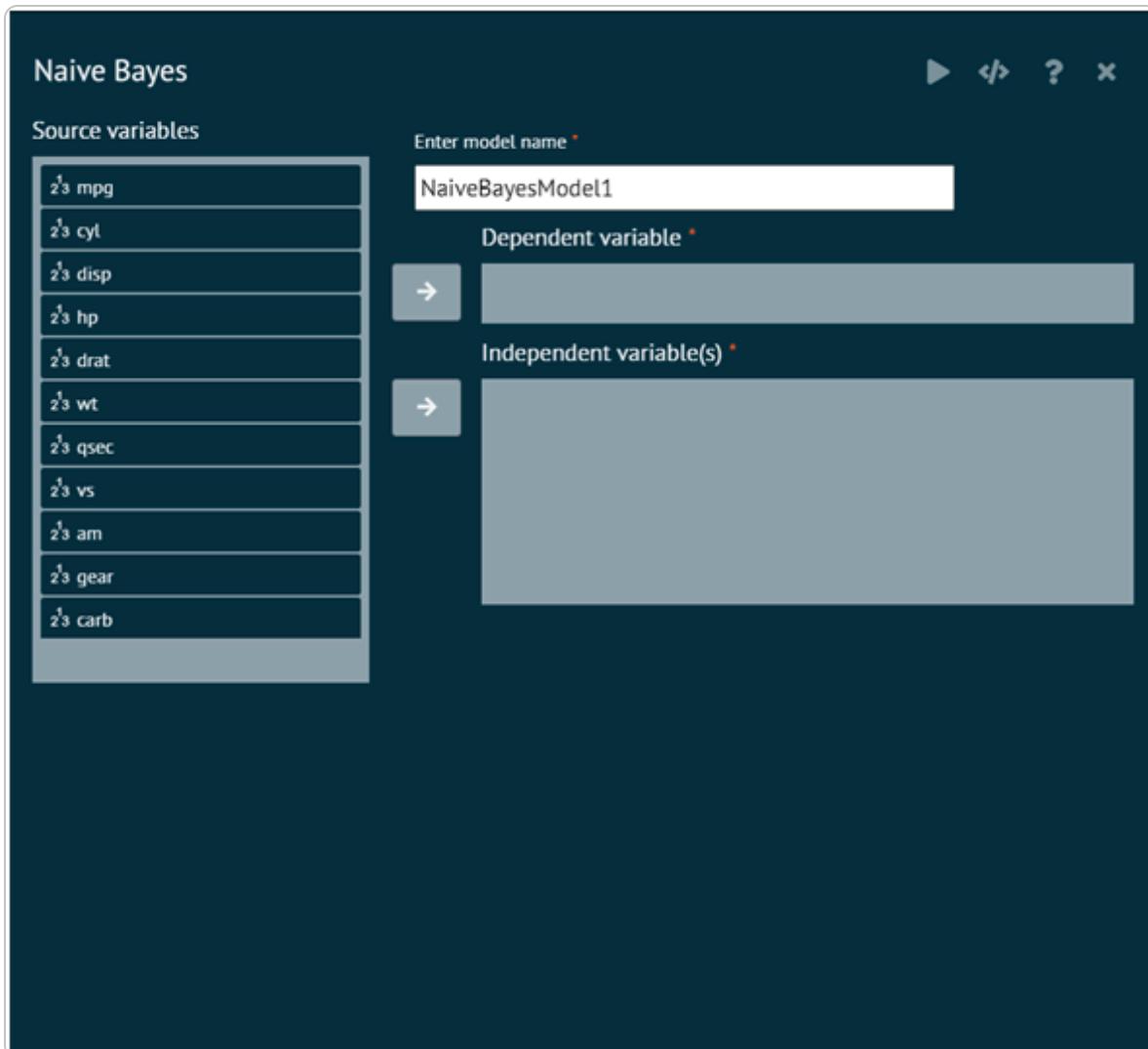
Naive Bayes

This function computes the conditional a-posterior probabilities of a categorical class variable given independent predictor variables using the Bayes rule.

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

Steps

Load the dataset -> Click on the Model Fitting tab in main menu -> Select Naive Bayes -> There will appear a dialog -> Select the Model name, dependent variables, and independent variable in the dialog -> Fill in the other options. -> Finally execute the plot and visualise the output in output window.



Naive Bayes

- For detailed help click on the R icon on the top right hand side of the Help dialog overlay

SEM

This function Fits a Structural Equation Model (SEM).

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

Steps

Load the dataset -> Click on the Model Fitting tab in main menu -> Select SEM -> There will appear a dialog -> Select the Model name, Latent variables in the dialog -> Fill in the other options. -> Finally execute the plot and visualise the output in output window.

SEM

▶ ⌂ ? ×

Enter a name of the model *

Sem1

Source variables

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

Show parameter labels

Latent variables

+ Add

SEM

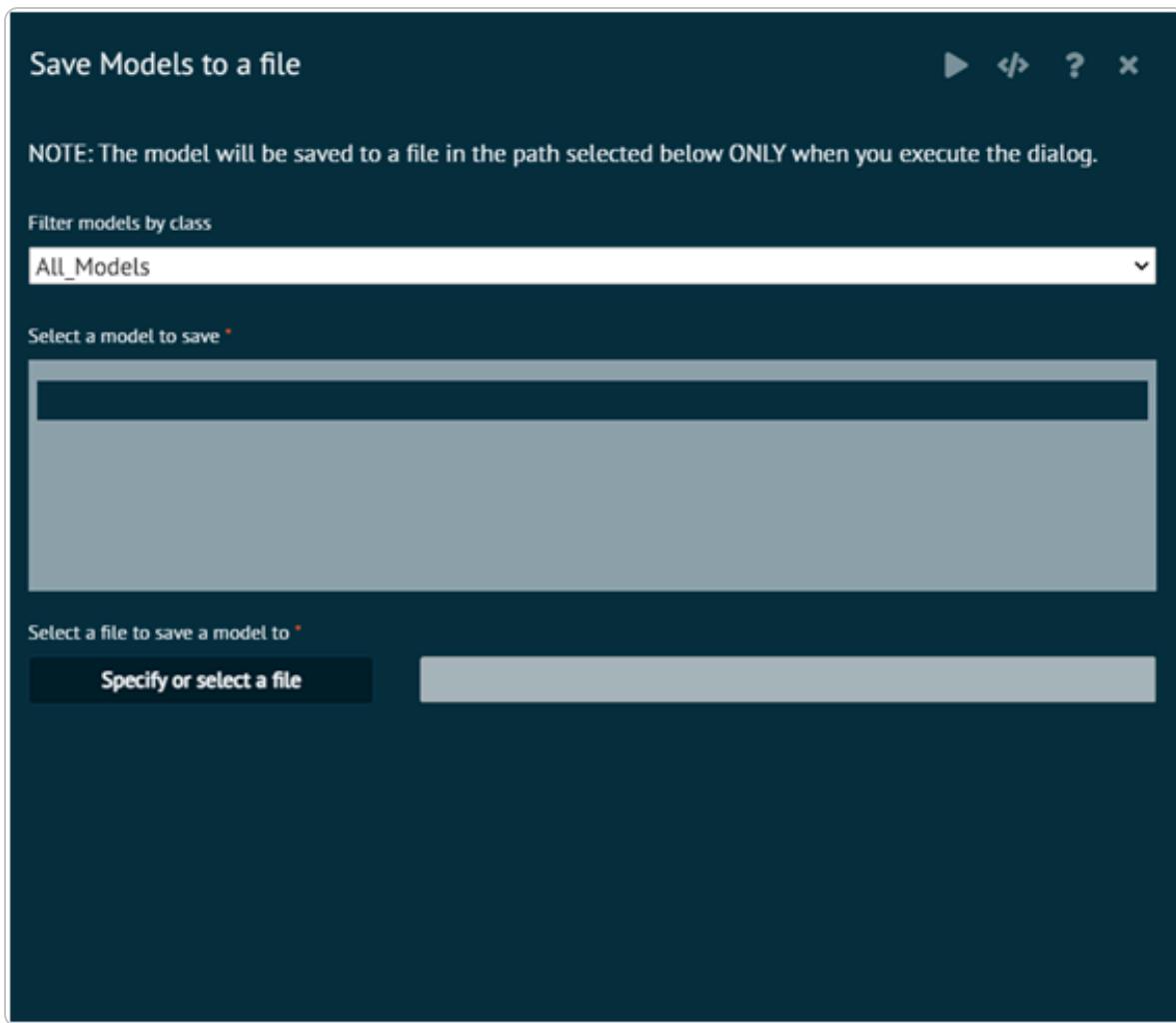
Save Model to a file

Models selected will be saved to a file in the R data format (.RData extension)

User can specify a new file by entering a name or selecting an existing file.

If user selects an existing file, it will be overwritten.

User has to click the execute button (the horizontal triangle button) to run the dialog and save the objects to a file.



Save Model to a file

Arguments

model1, model2...

fitted model objects

file :file name with path, the model objects selected will be saved to this RData file

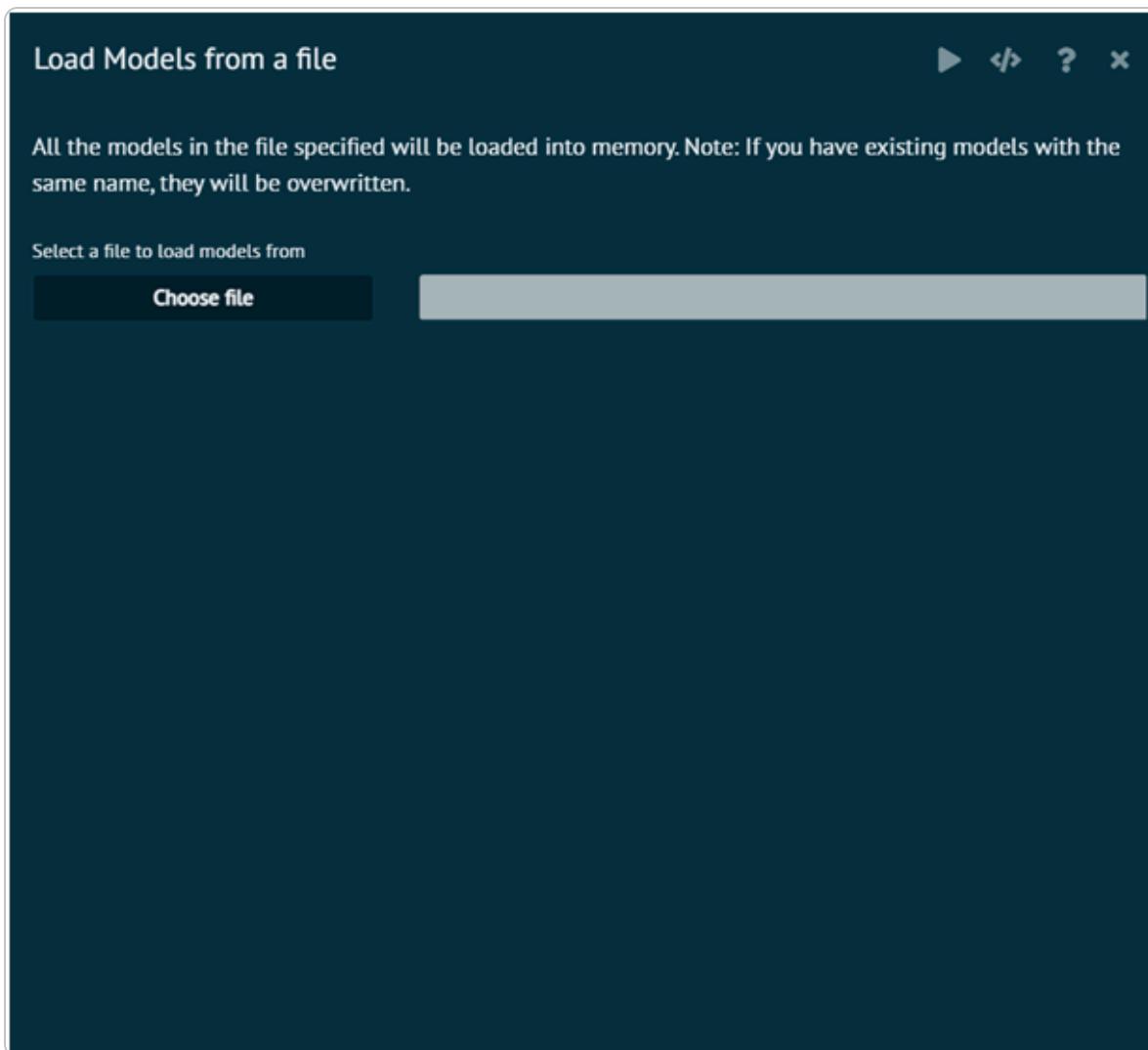
- For detailed help click on the R icon on the top right hand side of the help dialog overlay

Load Model from a file

R Model objects saved to the file selected will be loaded. BioStat Prime will load ALL R objects saved to the selected file. If objects with the same name exist in the R global environment, they will be overwritten.

BioStat Prime will display a message confirming the models built with the BioStat application that were loaded.

User has to click the execute button (the horizontal triangle button) to load R objects saved to the selected file.



Load Model from a file

Arguments

file

file name with path, the model objects selected will be saved to this RData file

- ❶ For detailed help click on the R icon on the top right hand side of the help dialog overlay

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

This function compares 2 nested modes using a F or a Chi-sq test depending on estimation. F tests are used for least squares estimation, chi-sq test are used for maximum likelihood estimation. Both models should be created on the same dataset as differences in missing values can cause problems

The screenshot shows the Rcmdr graphical user interface. The main window displays the 'Confidence Interval' dialog, which lists several model classes: Linear model (lm), Generalized linear model (glm), Nonlinear Least Squares (nls), Ordered Logistic/Probit regression (pol), and Multinomial Log Linear Models (multinom). A dropdown menu 'Select a model' contains three entries: 'Linear_ReqModel1', 'MLM', and 'Poly_Model'. The 'Linear_ReqModel1' entry is highlighted with a green background. Below this, a 'Confidence interval' slider is set to 0.95. Two radio button options are present: 'Likelihood-ratio statistic' (selected) and 'Wald statistic'. To the right, the 'Output' window shows R code and its execution results. The R code includes loading packages like RcmdrMisc, sandwich, and knitr, and attaching the RcmdrMisc package. It also lists masked objects from the BlueSky package. The output table displays confidence intervals for variables vs, carb, and gear.

	2.5 %	97.5 %
(Intercept)	8.7952	11.5824
vs	-2.4744	-0.9932
carb	0.1901	0.6604
gear	-1.6361	-0.7713

Compare N Models



The comparison between two or more models will only be valid if they are fitted to the same dataset. This may be a problem if there are missing values and R's default of `na.action = na.omit` is used.

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.

Steps

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 software interface. The top menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION (which is highlighted), FORECASTING, DOE, and HELP. 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'. A left panel displays a list of models: CoxRegModel1, LinearRegModel1, MLM, and Poly_Model. The right panel shows a 'Statistical Model Comparison' table for three models (Model 1, Model 2, Model 3) across various variables: carb, (Intercept), vs, gear, 13.3: (Intercept), 13.3: qsec, 14.3: (Intercept), 14.3: qsec, and 14.7: (Intercept). The table includes coefficients and standard errors in parentheses, with some values preceded by asterisks indicating statistical significance.

	Model 1	Model 2	Model 3
carb	-1.1556*** (0.3451)	0.4252*** (0.1148)	
(Intercept)		10.1888*** (0.6803)	
vs		-1.7358*** (0.3616)	
gear		-1.2037*** (0.2111)	
13.3: (Intercept)			365.9176*** (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

Confidence Interval

FIT

When comparing models fitted by maximum likelihood to the same data, the smaller the AIC or BIC, the better the fit. The theory of AIC requires that the log-likelihood has been maximized: whereas AIC can be computed for models not fitted by maximum likelihood, their AIC values should not be compared.

Examples of models not ‘fitted to the same data’ are where the response is transformed (accelerated-life models are fitted to log-times) and where contingency tables have been used to summarize data.

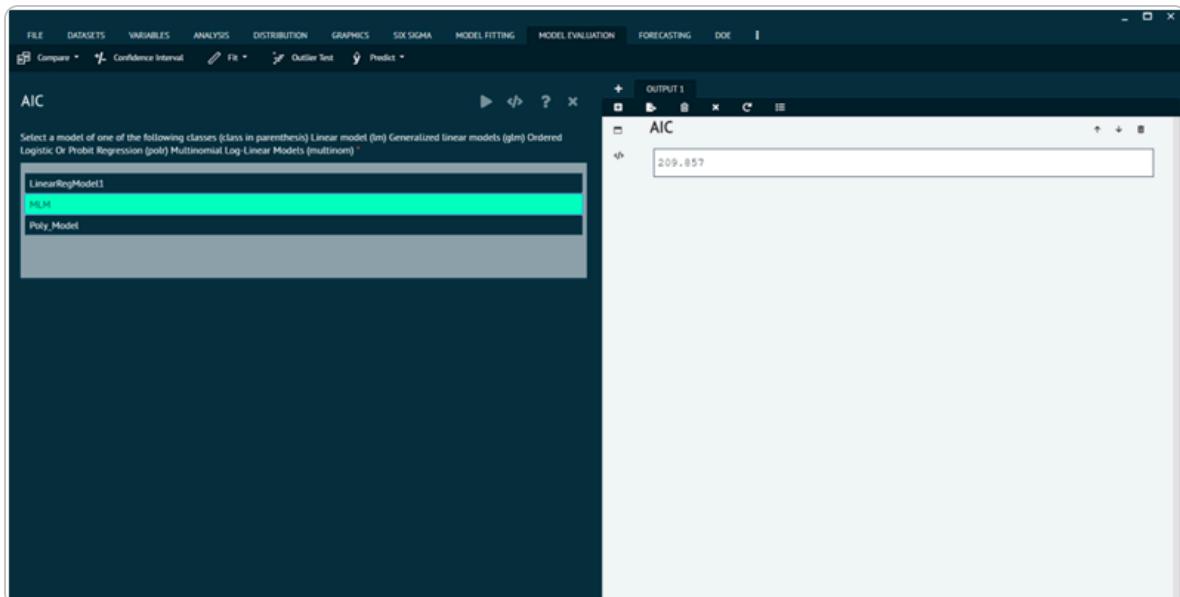
These are generic functions (with S4 generics defined in package stats4): however methods should be defined for the log-likelihood function `logLik` rather than these functions: the action of their default methods is to call `logLik` on all the supplied objects and assemble the results. Note that in several common cases `logLik` does not return the value at the MLE: see its help page.

The log-likelihood and hence the AIC/BIC is only defined up to an additive constant. Different constants have conventionally been used for different purposes and so `extractAIC` and `AIC` may give different values (and do for models of class "Im": see the help for `extractAIC`). Particular care is needed when comparing fits of different classes (with, for example, a comparison of a Poisson and gamma GLM being meaningless since one has a discrete response, the other continuous).

`BIC` is defined as `AIC(object, ..., k = log(nobs(object)))`. This needs the number of observations to be known: the default method looks first for a "nobs" attribute on the return value from the `logLik` method, then tries the `nobs` generic, and if neither succeed returns `BIC` as NA.

AIC

Generic function calculating Akaike's 'An Information Criterion' for one or several fitted model objects for which a log-likelihood value can be obtained, according to the formula $-2\log\text{-likelihood} + knpar$, where `npar` represents the number of parameters in the fitted model, and `k = 2` for the usual AIC, or `k = log(n)` (`n` being the number of observations) for the so-called BIC or SBC (Schwarz's Bayesian criterion).



AIC

Arguments

object

a fitted model object for which there exists a `logLik` method to extract the corresponding log-likelihood, or an object inheriting from class `logLik`.

... optionally more fitted model objects.

k

numeric, the penalty per parameter to be used; the default `k = 2` is the classical AIC.

BIC

Arguments

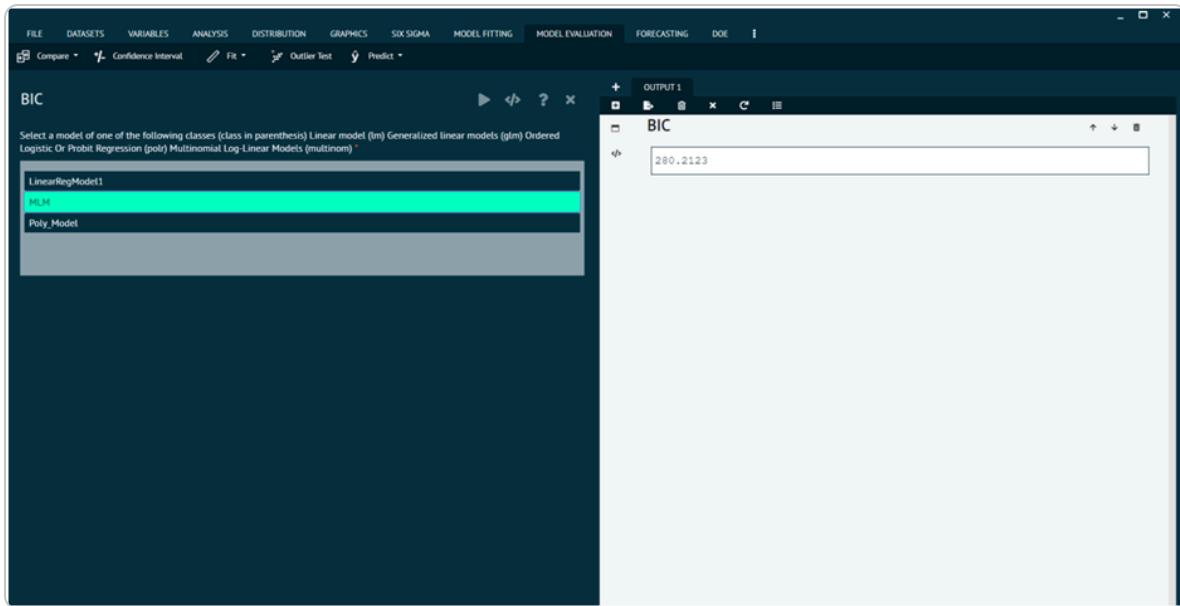
object

a fitted model object for which there exists a `logLik` method to extract the corresponding log-likelihood, or an object inheriting from class `logLik`.

... optionally more fitted model objects.

k

numeric, the penalty per parameter to be used; the default `k = 2` is the classical AIC.

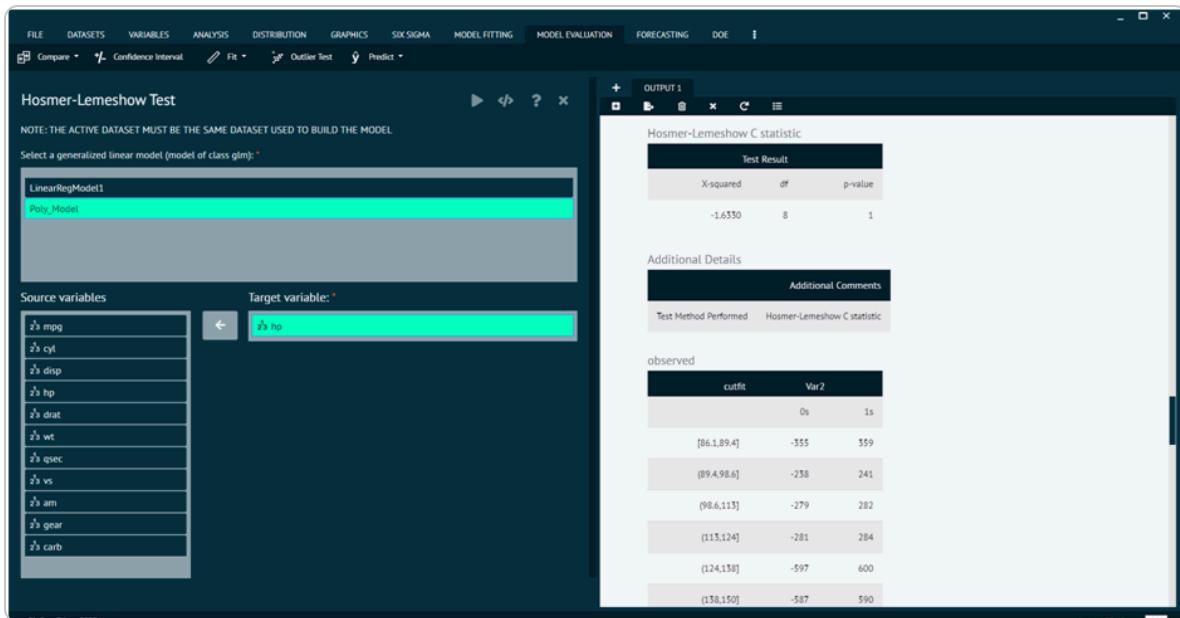


BIC

Hosmer-Lemeshow Test

The function computes Hosmer-Lemeshow goodness of fit tests for C and H statistic as well as the le Cessie-van Houwelingen-Copas-Hosmer unweighted sum of squares test for global goodness of fit.

Hosmer-Lemeshow goodness of fit tests are computed; see Lemeshow and Hosmer (1982). If `X` is specified, the le Cessie-van Houwelingen-Copas-Hosmer unweighted sum of squares test for global goodness of fit is additionally determined; see Hosmer et al. (1997). A more general version of this test is implemented in function `residuals.lrm` in package `rms`.



Hosmer-Lemeshow Test

Arguments

fit

numeric vector with fitted probabilities.

obs

numeric vector with observed values.

ngr

number of groups for C and H statistic.

X

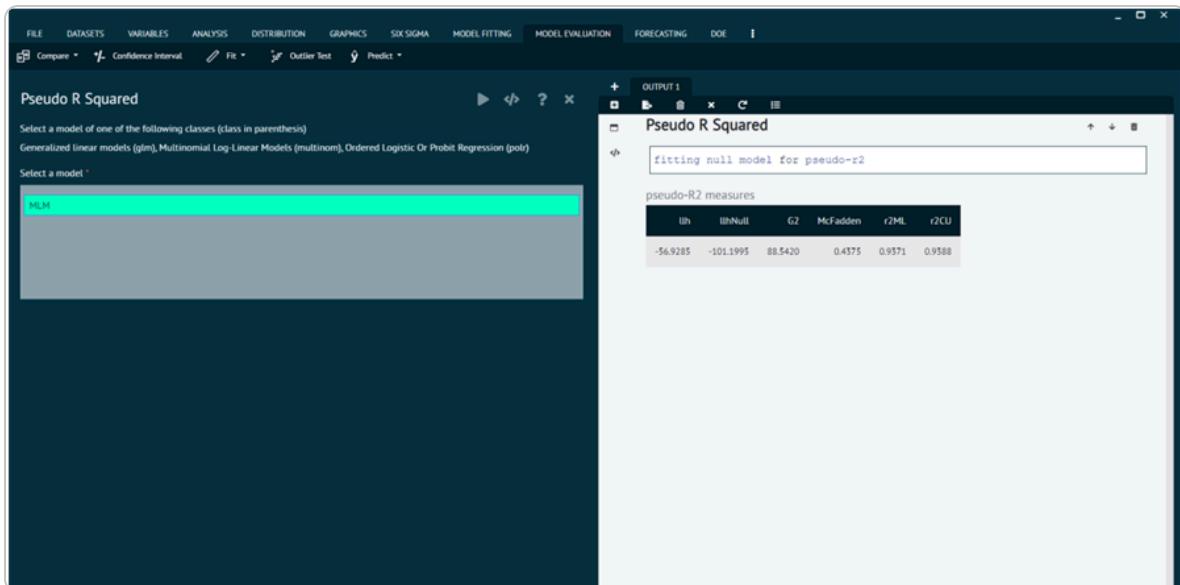
covariate(s) for le Cessie-van Houwelingen-Copas-Hosmer global goodness of fit test.

verbose

logical, print intermediate results.

Pseudo R Squared

Numerous pseudo r-squared measures have been proposed for generalized linear models, involving a comparison of the log-likelihood for the fitted model against the log-likelihood of a null/restricted model with no predictors, normalized to run from zero to one as the fitted model provides a better fit to the data (providing a rough analogue to the computation of r-squared in a linear regression).



Pseudo R Squared

Arguments

object

a fitted model object, for now of class `glm`, `polr`, or `multinom`

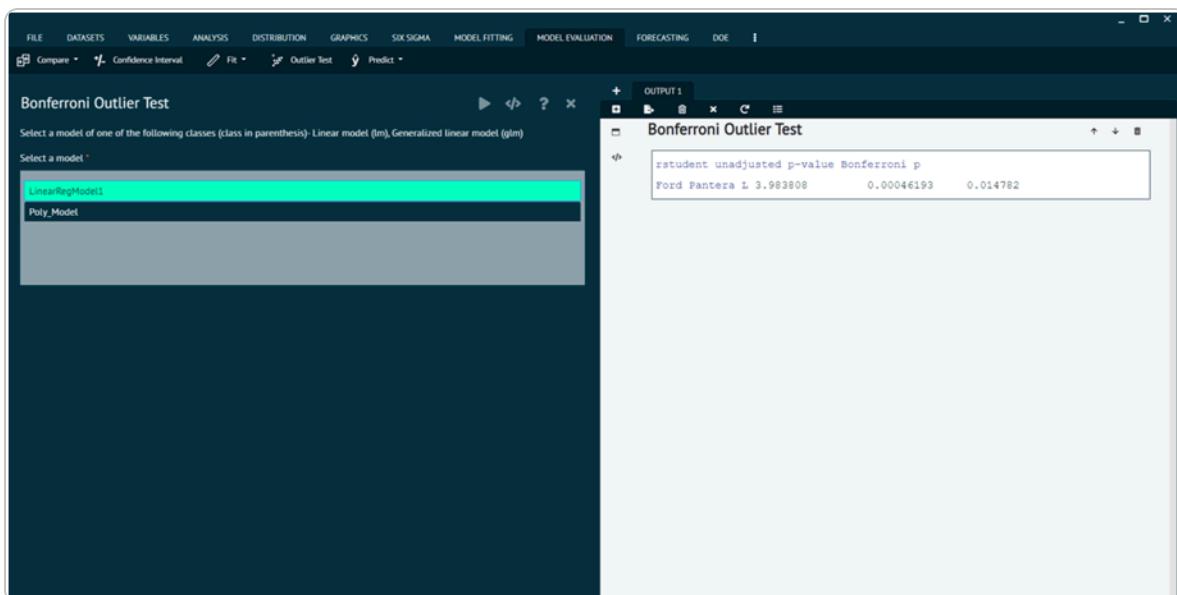
... additional arguments to be passed to or from functions

Outlier Test

Bonferroni Outlier Test

For a linear model, p-values reported use the t distribution with degrees of freedom one less than the residual df for the model. For a generalized linear model, p-values are based on the standard-normal distribution. The Bonferroni adjustment multiplies the usual two-sided p-value by the number of observations.

This function reports the Bonferroni p-values for testing each observation in turn to be a mean-shift outlier, based Studentized residuals in linear (t-tests), generalized linear models (normal tests), and linear mixed models.



Bonferroni Outlier Test

Arguments

model

an lm, glm, or lmerMod model object; the "lmerMod" method calls the "lm" method and can take the same arguments.

cutoff

observations with Bonferroni p-values exceeding cutoff are not reported, unless no observations are nominated, in which case the one with the largest Studentized residual is reported.

n.max

maximum number of observations to report (default, 10).

order

report Studenized residuals in descending order of magnitude? (default, TRUE).

labels

an optional vector of observation names.

...

arguments passed down to methods functions.

x

outlierTest object.

digits

number of digits for reported p-values.

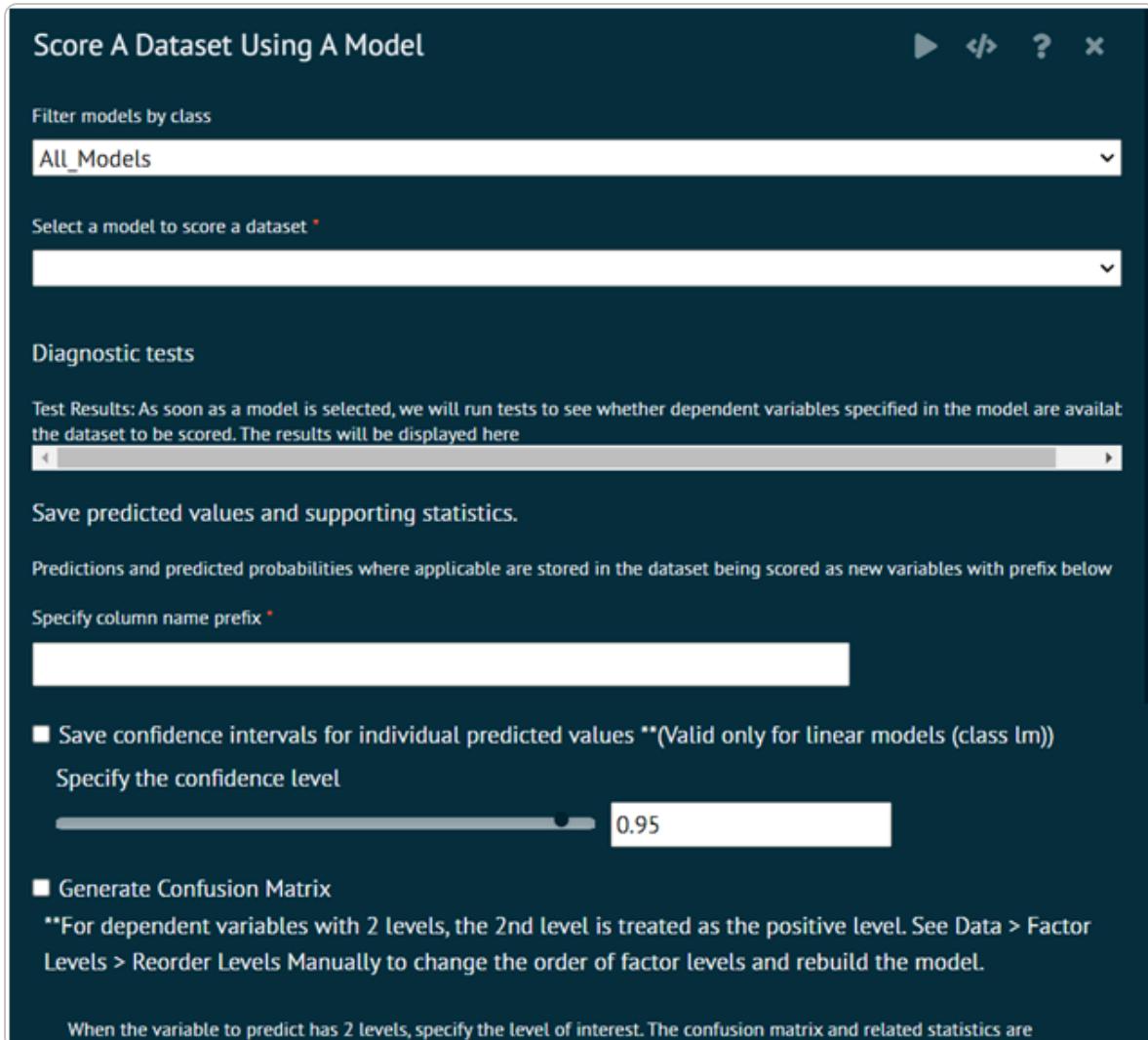
Predict

Stores predictions with the specified confidence interval in the current dataset using the specified prefix.

Model Scoring

Model scoring does the following

1. Scores the current dataset using the selected prebuilt model. Stores predictions with the specified confidence interval in the current dataset using the specified prefix.
2. Optionally creates a confusion matrix and a ROC curve
3. In the case where you are scoring a training dataset that contains the dependent variable/variable to predict and the dependent variable has 2 levels, you have the option to select the reference level/level of interest.
4. The confusion matrix and related statistics are created using the specified level of interest.



Predict

Arguments

modelname

a model object for which prediction is desired.

prefix

prefix string that will be used to create new variables containing the predictions.

datasetname

is the current dataset to score and save predictions to.

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)

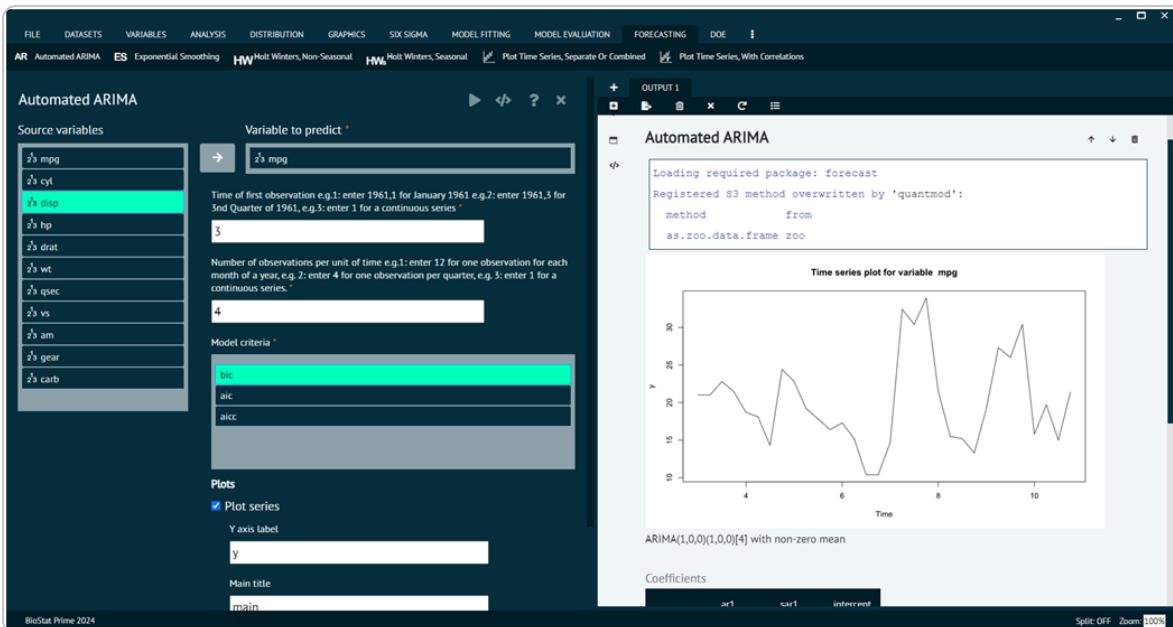
ARIMA, which stands for Auto-Regressive Integrated Moving Average, is a popular time series forecasting model in statistics. It combines three components: Auto-Regressive (AR), Integrated (I), and Moving Average (MA). ARIMA models are widely used to analyze and forecast time-series data, where observations 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.

This Function returns best ARIMA model according to either AIC, AICc or BIC value. The function conducts a search over possible model within the order constraints provided. Internally calls auto.arima in the forecast package

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

Steps

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.



Automated ARIMA (AR)

Arguments

vars

selected variables to build an automatic arima model for.

start

Time of first observation should be entered in the format year,month or year,quarter
e.g.(if your data is organized in months the 1992,1 for Jan 1992 or if your data is
organized in quarters then 1992,1 refers to the first quarter of 1992.

frequency

Number of observations in unit time. Example: for monthly there are 12 observation
in a year. For quarterly there are 4 observation in a year.

ic

Information criterion to be used in model selection. It must be one of "aic", "aicc" or
"bic"

plotSeries

if TRUE a time series plot will also be generated.

plotResiduals

if TRUE residuals will also be plotted.

predict

if TRUE predicted values will also be generated.

savePredictedVals

predicted values will be saved.

PlotPredictedValues

predicted values will also be plotted.

correlogram

if TRUE a correlogram will be generated.

main

main title of the plot

ylab

title for the y axis

dataset

the name of the dataset from which the vars have been picked.

⚠ The user can choose additional options like plot options.

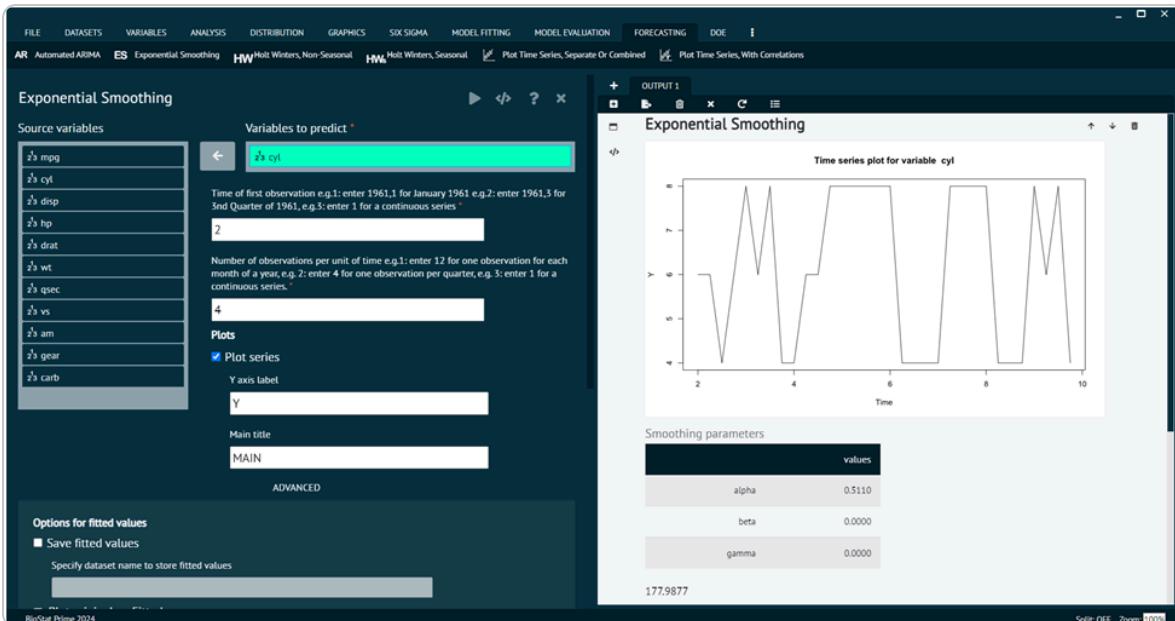
Exponential Smoothing (ES)

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.

Steps

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.



Exponential Smoothing ES

Arguments

vars

select a variable to build a model for

start

Time of first observation should be entered in the format year,month or year,quarter
e.g.(if your data is organized in months the 1992,1 for Jan 1992 or if your data is
organized in quarters then 1992,1 refers to the first quarter of 1992.

frequency

Number of observations in unit time. Example: for monthly there are 12 observation
in a year. For quarterly there are 4 observation in a year.

exponential

Determines whether exponential smoothing will be done, value set to TRUE

seasonal

a character string "None" for exponential smoothing.

plotSeries

if TRUE a time series plot will also be generated.

saveFitted

if TRUE fit values are saved.

plotOriginalandForecast

Plot original and forecasted series

predict

if TRUE predicted values will also be generated.

savePredictedVals

predicted values will be saved.

plotPredictedValues

predicted values will also be plotted.

correlogram

if TRUE a correlogram will be generated.

main

main title of the plot

ylab

title for the y axis

dataset

the name of the dataset from which the variables have been selected

⚠ The user can choose additional options like plot options.

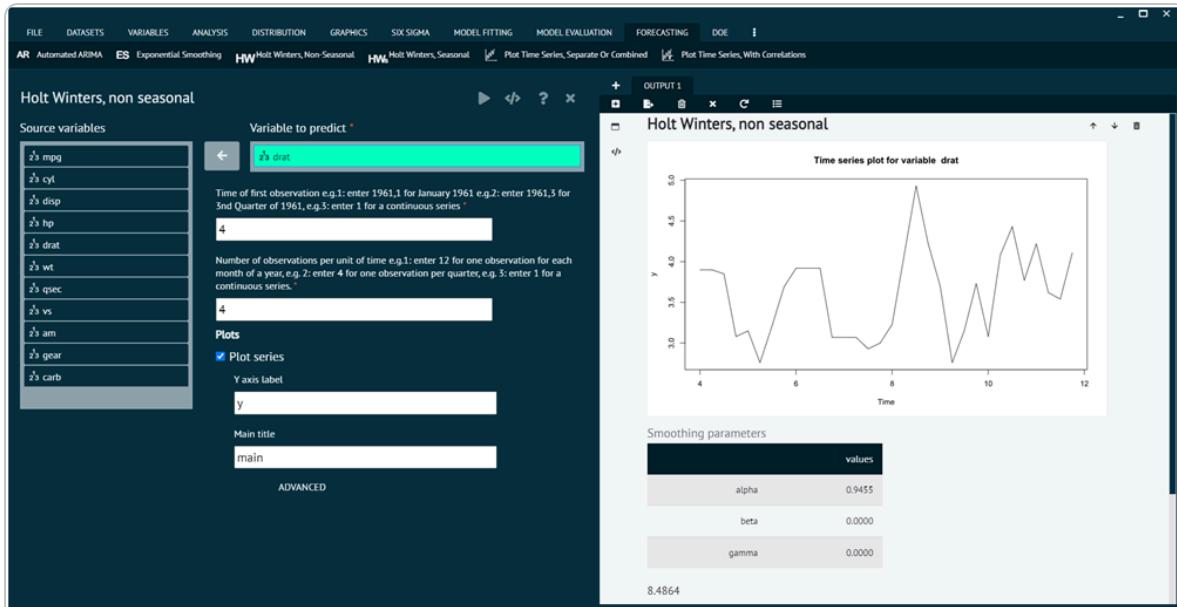
Holt Winters, Non-seasonal

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.

Steps

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.



Holt Winters, Non-seasonal

Arguments

vars

select a variable to build a model for

start

Time of first observation should be entered in the format year,month or year,quarter
e.g.(if your data is organized in months the 1992,1 for Jan 1992 or if your data is
organized in quarters then 1992,1 refers to the first quarter of 1992.

frequency

Number of observations in unit time. Example: for monthly there are 12 observation
in a year. For quarterly there are 4 observation in a year.

exponential

Determines whether exponential smoothing will be done, value set to FALSE

seasonal

a character string "Non Seasonal" for a non seasonal model.

plotSeries

if TRUE a time series plot will also be generated.

saveFitted

if TRUE fit values are saved.

plotOriginalandForecast

Plot original and forecasted series

predict

if TRUE predicted values will also be generated.

savePredictedVals

predicted values will be saved.

plotPredictedValues

predicted values will also be plotted.

correlogram

if TRUE a correlogram will be generated.

main

main title of the plot

ylab

title for the y axis

dataset

the name of the dataset from which the variables have been selected

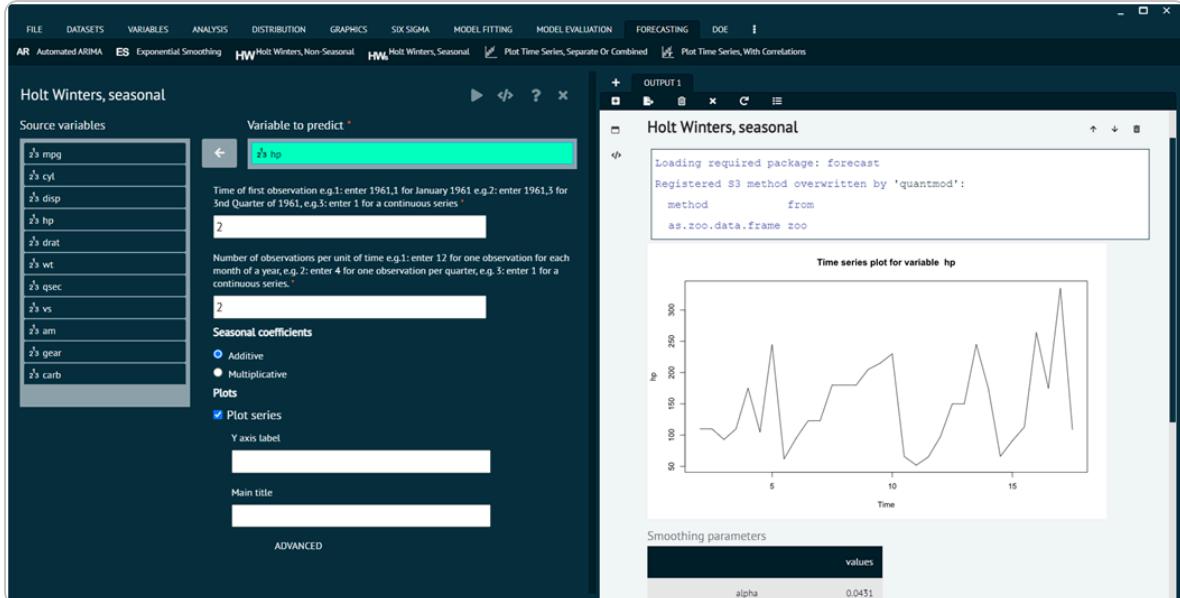
Holt Winters, Seasonal

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.

Steps

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.



Arguments

vars

select a variable to build a model for

start

Time of first observation should be entered in the format year,month or year,quarter
e.g.(if your data is organized in months the 1992,1 for Jan 1992 or if your data is
organized in quarters then 1992,1 refers to the first quarter of 1992.

frequency

Number of observations in unit time. Example: for monthly there are 12 observation
in a year. For quarterly there are 4 observation in a year.

exponential

Determines whether exponential smoothing will be done, value set to FALSE

seasonal

Character string to select an "additive" (the default) or "multiplicative" seasonal
model. The first few characters are sufficient e.g. "add" or "mult".

plotSeries

if TRUE a time series plot will also be generated.

saveFitted

if TRUE fit values are saved.

plotOriginalandForecast

Plot original and forecasted series

predict

if TRUE predicted values will also be generated.

savePredictedVals

predicted values will be saved.

plotPredictedValues

predicted values will also be plotted.

correlogram

if TRUE a correlogram will be generated.

main

main title of the plot

ylab

title for the y axis

dataset

the name of the dataset from which the variables have been selected

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.

This function creates time series plot in combined or separately. When combined multiple variables are plotted together, when separate each variable is plotted independently.

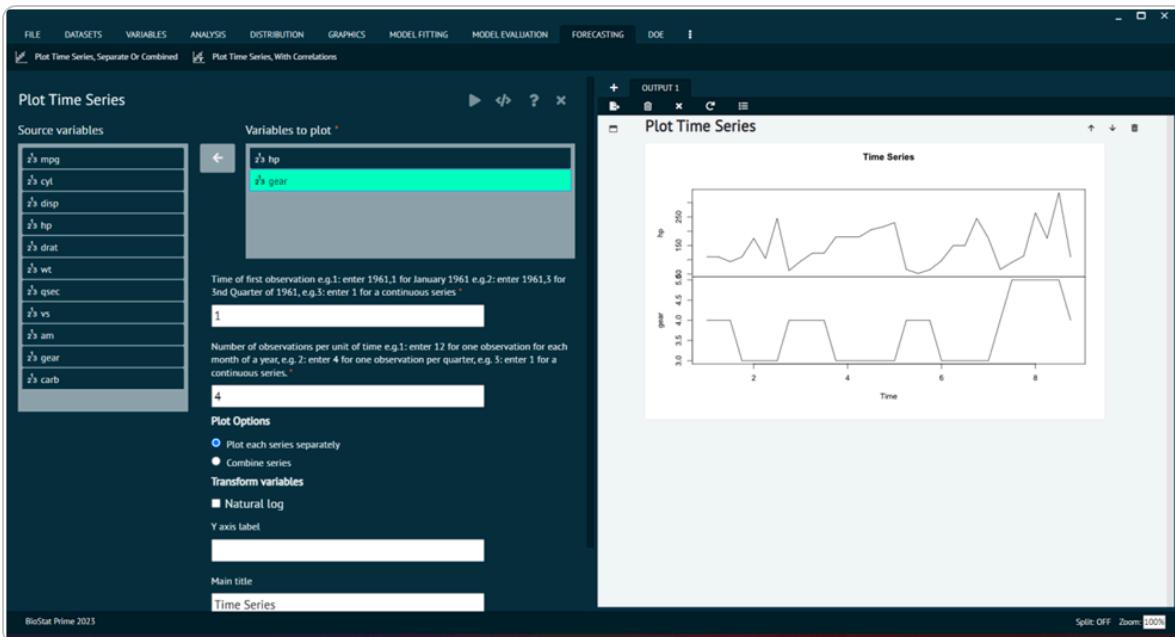
To analyse it in BioStat Prime user must follow the steps as given.

Steps

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.



Plot Time Series, Separate OR Combined

Arguments

vars

selected variables to plot

start

Time of first observation should be entered in the format year,month or year,quarter e.g.(if your data is organized in months the 1992,1 for Jan 1992 or if your data is organized in quarters then 1992,1 refers to the first quarter of 1992.

frequency

Number of observations in unit time. Example: for monthly there are 12 observation in a year. For quarterly there are 4 observation in a year.

plot.type

"multiple" for separate and "single" for combined plot.

naturalLogYaxis

if TRUE an Y axis is shown as natural log value.

main

main title of the plot

ylab

title for the y axis

dataset

the name of the dataset from which the vars has been picked.

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.

Creates time series plot with autocorrelations, autocovariance and partial correlations.

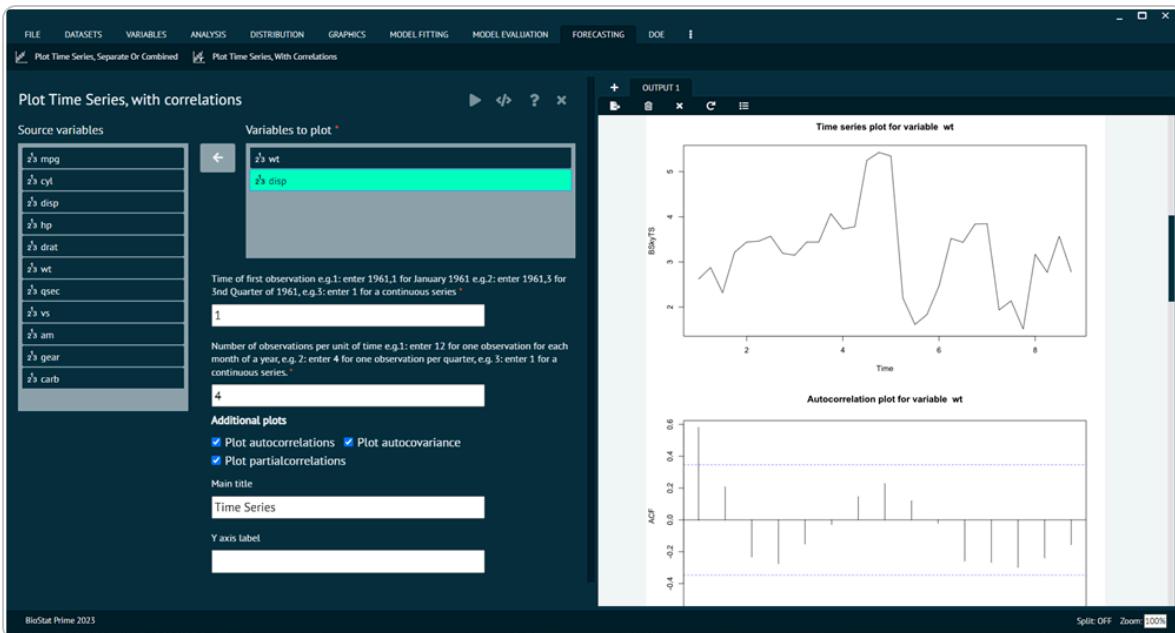
To analyse it in BioStat Prime user must follow the steps as given.

Steps

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.



Plot Time Series with Correlations

Arguments

vars

selected variables to plot

start

Time of first observation should be entered in the format year,month or year,quarter e.g.(if your data is organized in months the 1992,1 for Jan 1992 or if your data is organized in quarters then 1992,1 refers to the first quarter of 1992.

frequency

Number of observations in unit time. Example: for monthly there are 12 observation in a year. For quarterly there are 4 observation in a year.

autocorrelation

if TRUE an autocorrelation plot will also be generated.

autocovariance

if TRUE an autocovariance plot will also be generated.

partialautocorrelations

if TRUE an partial autocorrelations plot will also be generated.

plot.type

"multiple" for separate and "single" for combined plot.

main

main title of the plot

ylab

title for the y axis

dataset

the name of the dataset from which the vars has been picked.

Design of Experiment-Quality Control

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.

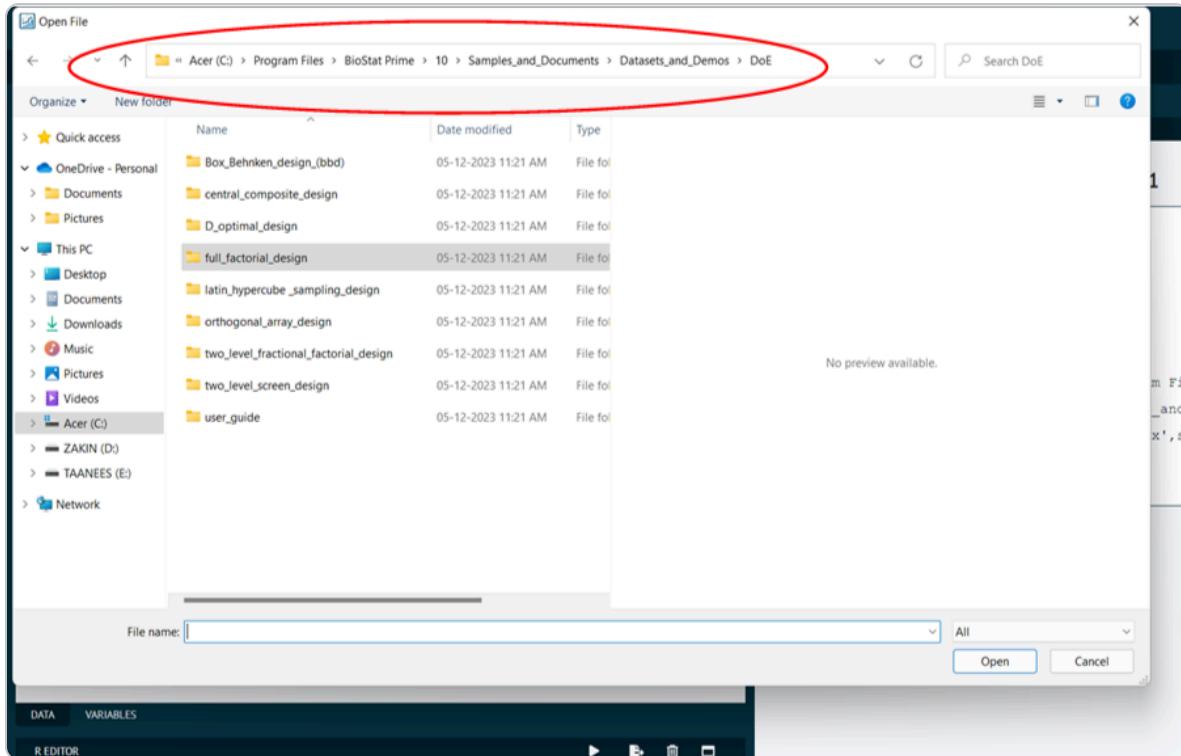
- ⚠** The primary goal of DOE is to optimize processes, improve product or system performance, and understand the relationship between input variables and the output response.

- ⚠** 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.



DOE1

- 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.

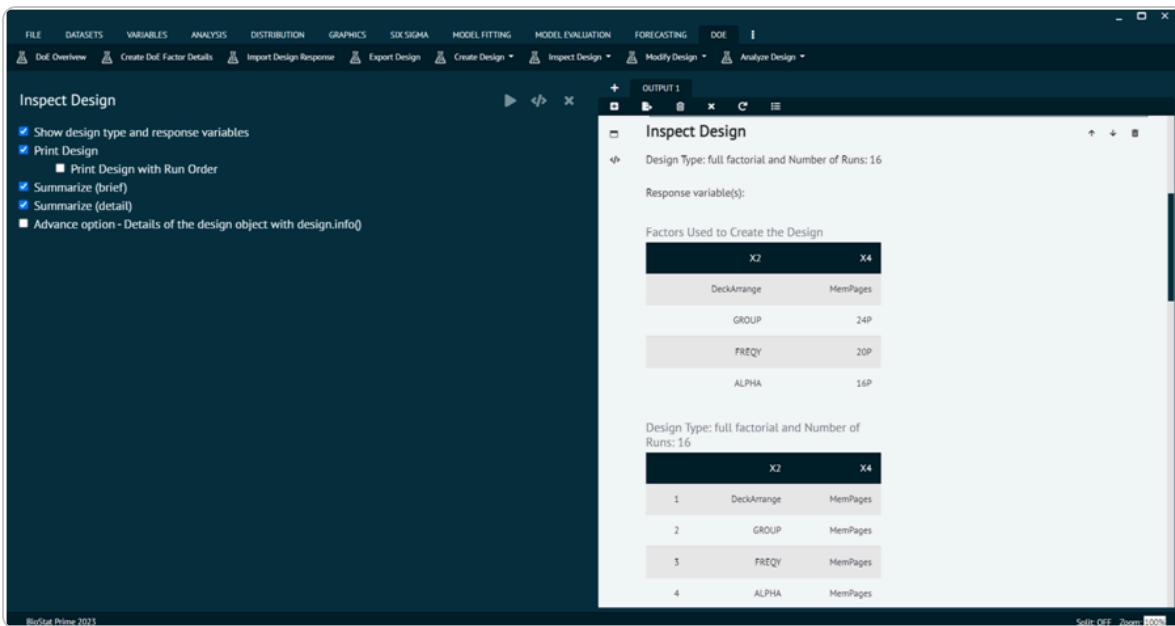
```

New names:
# ... -> ...
# ... -> ...
# ... -> ...
# ... -> ...

Successfully opened using:
[1] "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)"
  
```

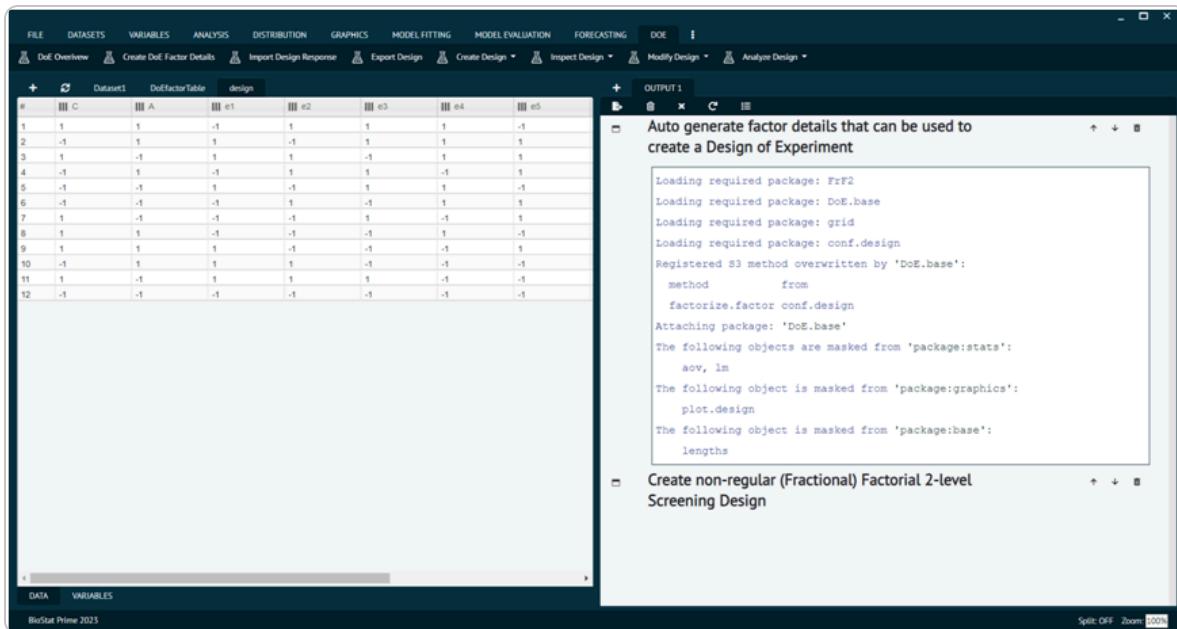
DOE2

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.



DOE3

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.



DOE4

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 (?)**.

i 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.

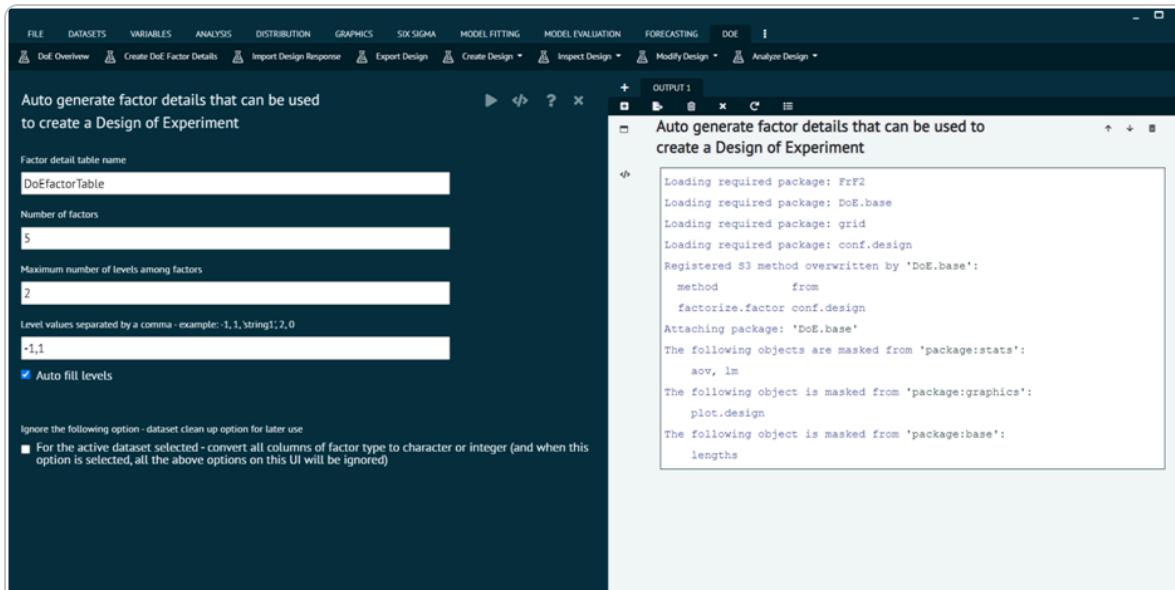
Create DoE Factor Details

Automatically generates a table with factor details based on the parameter specified - number of factors and the maximum levels for the factors along with the default values for the factor to be used.

- ⚠ If non numeric values are used for factor, specify within single quote for example 1,-1,'name1',0,'string2'

Once the factor details are automatically generated and the factor table shows up on the data grid UI, you can change the values and/or remove some values from the grid to manipulate the table as you choose including factors with different number of values

- ⚠ If the number of factors are ≤ 26 , the factors are named with upper case alphabets as A, B, C, .., otherwise, named as F1, F2, ..., F27, F28, ..



Create DoE Factor Details

DoE Overview Create DoE Factor Details Import Design Response Export Design Create Design Inspect D

+ Dataset1 mtcars DoEfactorTable

#	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

Create DoE Factor

Import Design Response

Import the Design csv file with added response columns i.e. results recoded from the experiments conducted from the original design previously exported

The purpose of importing the csv file against a design already open in the dataset UI grid is to add the response columns automatically from the csv file to the design in the UI grid and create a new design with the name specified

Once the new design with the response columns is created - this can be analyzed with various analysis methods under **DOE -> Analyze Design** menu

In addition, designating/un-designating one or more response column(s) can be performed by **DOE -> Modify Design -> Add/Remove Response** menu

A If the required design is not already opened on the dataset UI grid, using the file Open menu, load the design R object (.rda file) that was previously exported with "Export Design" menu into a directory on the file system.

i For additional information - use R help(add.response, package = DoE.base)

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

Import Design Response

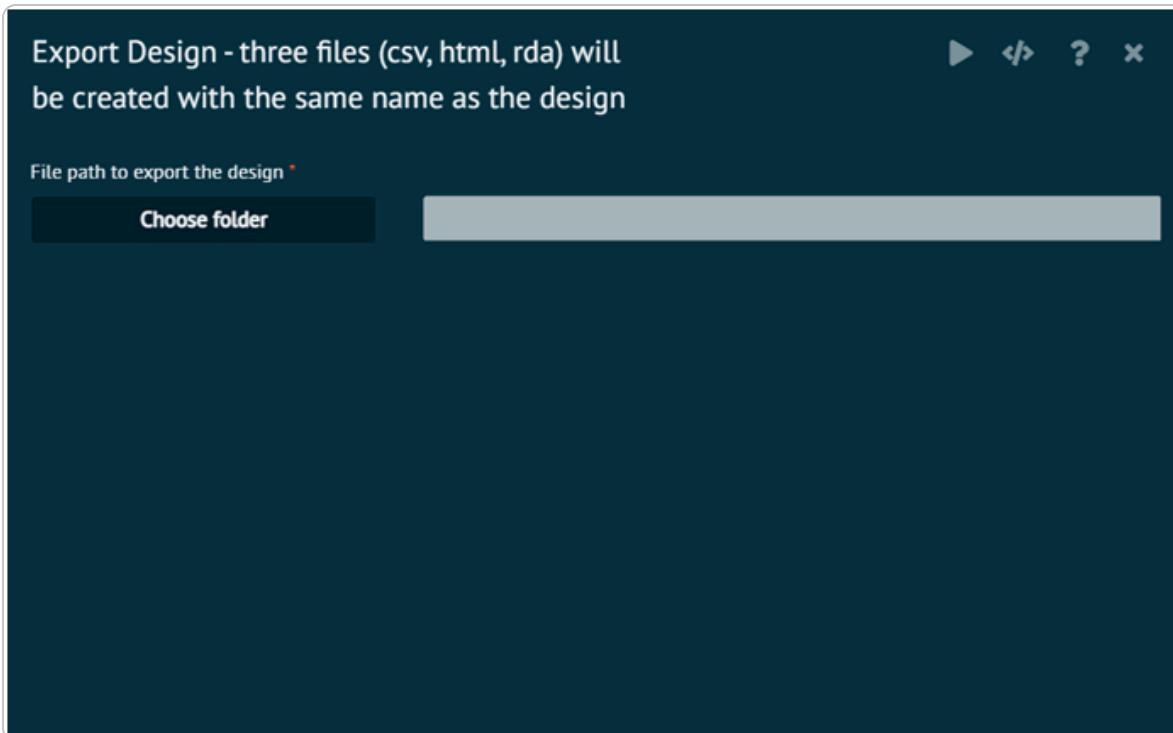
Export Design Response

Exporting the Design will create three files (**csv, html, rda**) with the same name as the design in the directory path chosen

- i** The purpose of exporting the design is to set up and conduct the experiments per the design in the real world and collect the results.

Add column(s) that are compliant with R variable naming convention to the csv file to record the results. These additional columns are known as the response(s) per the vocabulary of Design of Experiments.

Once the response column(s) are added to the csv file, **DOE -> Import Design** menu can be used to load the csv file against the correct design (already open) on the dataset UI grid.



Export Design Response

Create Design

Create 2-level Screening Design

Create 2-level Screening Design

Create Regular (Fractional) Factorial 2-Level Design

Create Regular (Fractional) Factorial 2-level

Design

Source variables

2 ⁹ mpg
2 ⁹ cyl
2 ⁹ disp
2 ⁹ hp
2 ⁹ drat
2 ⁹ wt
2 ⁹ qsec
2 ⁹ vs
2 ⁹ am
2 ⁹ gear
2 ⁹ carb

Design name *

Select variables *

→

Size and randomization

Number of runs (if specified, it must be a power of 2 otherwise make it 0)

8

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) *

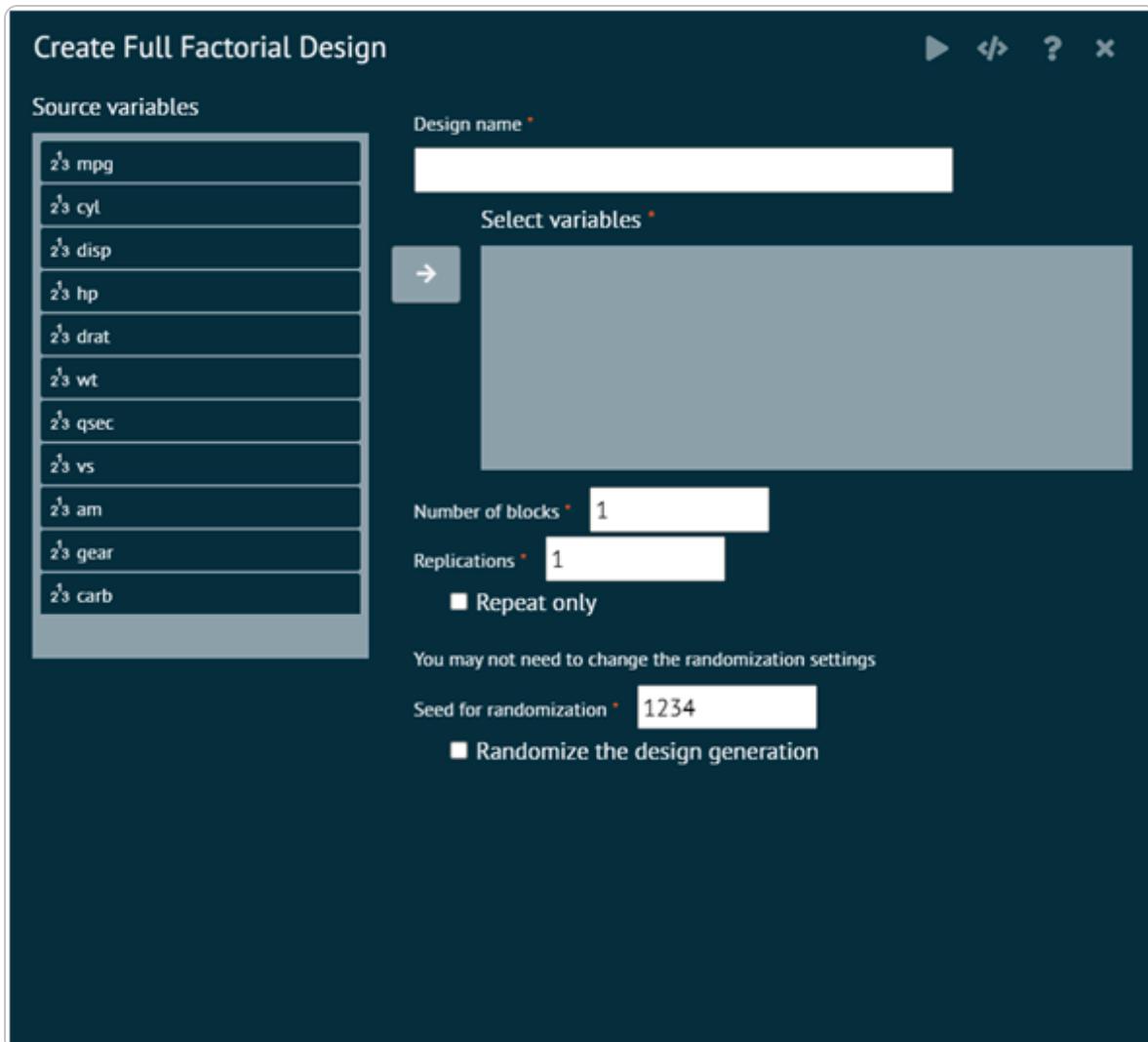
2

Replications * 1

Repeat only

Create Regular (Fractional) Factorial 2-Level Design

Create Full Factorial Design



Create Full Factorial Design

Create Orthogonal Array Design

Create Orthogonal Array Design

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

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)

Create D-Optional Design

Create D-Optimal Design

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

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:

Create D-Optimal Design

Create Central Composite (Quantitative) Design

Create Central Composite (Quantitative) Design
from an existing FrF2 Design

Source Datasets

mtcars
abbey

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

Create Central Composite (Quantitative) Design

Create Box-Behnken (Quantitative) Design

Create Box-Behnken (Quantitative) 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 *

→

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

Create Box-Behnken (Quantitative) Design

Create Latin Hypercube(Quantitative) Design

Create Latin Hypercube Design (for Quantitative Factors)

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

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

The screenshot shows a software window titled "Create Latin Hypercube Design (for Quantitative Factors)". On the left, there's a sidebar labeled "Source variables" containing ten entries: 2³ mpg, 2³ cyl, 2³ disp, 2³ hp, 2³ drat, 2³ wt, 2³ qsec, 2³ vs, 2³ am, 2³ gear, and 2³ carb. To the right of this is a "Design name" field with a placeholder box. Below it is a "Select variables" section with a large gray box and a right-pointing arrow button. Under "Size and randomization", there are fields for "Number of runs" (set to 20) and "Number of decimal places" (set to 2). A note says "You may not need to change the randomization settings" with a seed value of 1234. There's also a checkbox for "Randomize the design generation" which is checked. At the bottom, there's a note about "latin hypercube sampling designs" and a dropdown menu currently set to "optimum".

Create Latin Hypercube(Quantitative) Design

Create Taguchi Parameter Design

Create Taguchi Style Inner-Outer Parameter Design

Source Datasets

mtcars
abey

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)

Create Taguchi Parameter Design

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 'Factors Used to Create the Design' table. The table has columns for 'wt', 'qsec', and 'am'. The data rows are:

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

Inspect design

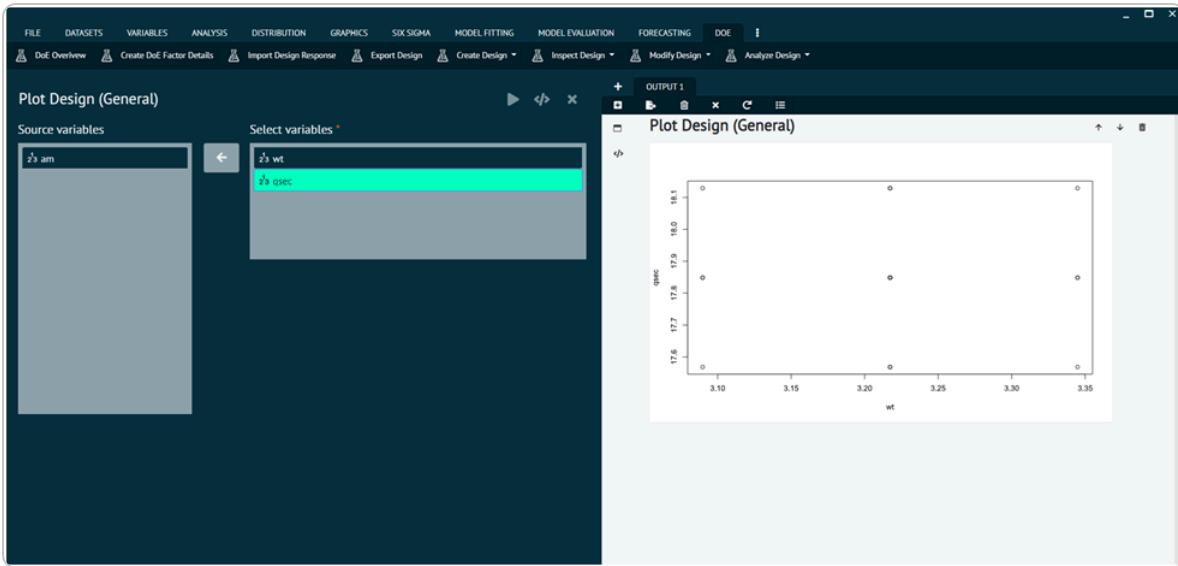
OUTPUT 1

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

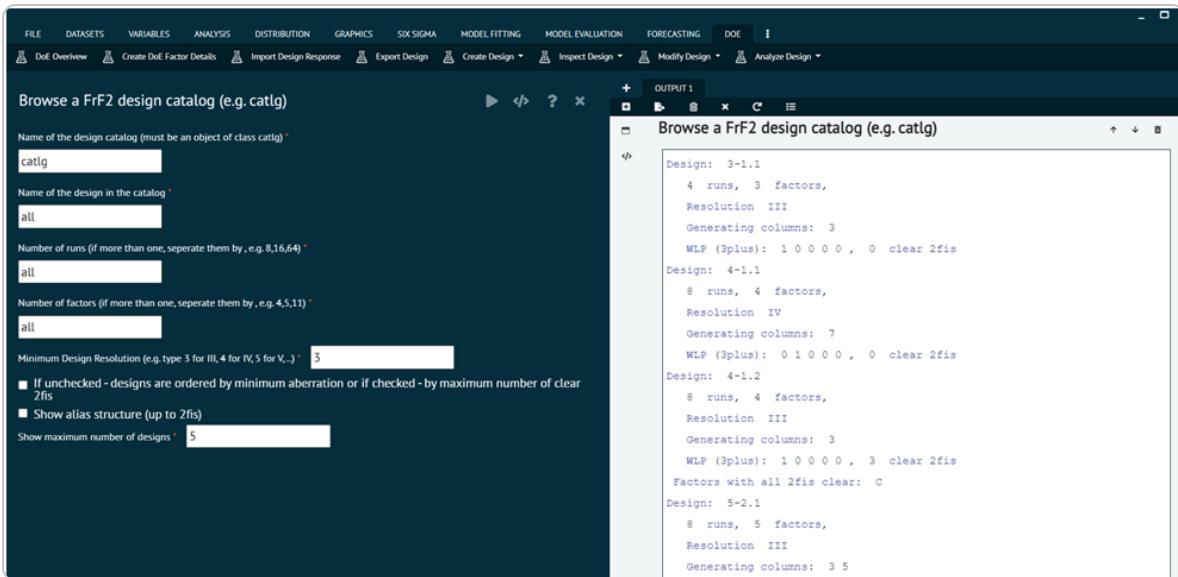
Inspect design output

Plot Design



Plot Design

Browse FrF2 Design Catalog



Browse FrF2 Design Catalog

Browse Orthogonal Design Catalog

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

- Top Menu Bar:** FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, MODEL EVALUATION, FORECASTING, DOE, etc.
- Toolbar:** DoE Overview, Create DoE Factor Details, Import Design Response, Export Design, Create Design, Inspect Design, Modify Design, Analyze Design.
- Left Panel:** Title "Browse the Orthogonal Array (oacat) design catalog". Input fields for:
 - Name of the Orthogonal Array design in the catalog (e.g. L18.3.6.6.1): "all"
 - Number of runs or a 2-element vector e.g. 4,16 with a minimum and maximum for the number of runs: "all"
 - Number of levels (separate them by , e.g. 3,2,5): "all"
 - Number of factors (separate them by , e.g. 4,2,1): "all"
 - Show all array quality metrics with the resulting arrays:
 - Show maximum number of designs: 5
- Right Panel:** Title "Browse the Orthogonal Array (oacat) design catalog". Output window showing the results of the search:


```
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
```

Browse Orthogonal Design Catalog

Modify Design

Add/Remove Response

This sub function of BioStat Prime designate one or more variables to be response variable(s) for the design

⚠ For the detail help - use R help(response.names, package = DoE.base)

{style"note"}

The screenshot shows a dialog box titled 'Add/Remove Response Variable(s) - first check the list of existing response variables with the DOE -> Inspect Design menu'. The interface includes a toolbar with icons for back, forward, help, and close. On the left, a list of 'Source variables' is shown, each preceded by a radio button and a question mark icon. The variables listed are: mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, and carb. An arrow button points from the source list to a larger, empty rectangular area labeled 'Select response variables *'. At the bottom of the dialog is a 'Save' button.

Add/Remove Response

Add Centerpoint 2-Level Design (Quantitative)

This sub function adds centerpoint to a 2-level Design (with no prior centerpoint). The selected design is the active design on the UI grid.

add.center function to add center points to a 2-level fractional factorial design. All factors must be quantitative

- For the detail help - use R help(add.center, package = FrF2)

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)

Number of positions for center point distribution (must be >1)

Add Centerpoint 2-Level Design (Quantitative)

Analyse Design

Design Analysis-Linear Model

Builds a linear regression model for the design to analyze the response (i.e. the results recorded/collected from the experiments). Internally calls function lm in stats package. Displays a summary of the model, coefficient table, Anova table and sum of squares table and plots the following residuals vs. fitted, normal Q-Q, theoretical quantiles, residuals vs. leverage. You can optionally specify a variable with weights and choose to ignore the intercept.

- For more details, see R help for the following

The screenshot shows the 'Design of Experiments analysis with Linear Model' interface. On the left, under 'Source variables', a list includes mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, carb. In the center, 'Enter Model Name' is set to 'LinearRegModel1'. 'Response (dependent) variable' is 'hp'. 'Independent variable(s)' are 'am' and 'vs'. Below these, there's a note about degree (leave blank or type 2), and checkboxes for 'Ignore intercept', 'All effects plot', and 'Plot residuals vs fitted, normal Q-Q, scale-location and residuals vs leverage'. At the bottom, there's a section for 'Specify a variable with weights'. On the right, the 'OUTPUT 1' tab shows the R code used:

```
hp = alpha + beta1(am) + beta2(vs) + epsilon
```

 and the resulting **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 ***

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

Below the summary are tables for **Residuals** and **Coefficients**.

Design Analysis-Linear Model

Design Analysis-Response Surface Model

Response-surface regression - fit a linear model with a response-surface component, and produce appropriate analyses and summaries

In the formula builder, you can type SO(var1, var2, var3, ...) or FO(var1, ..) or other function from the rsm package - help(SO, package =rsm)

Design of Experiments analysis with Response Surface Model (Quantitative)

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

Enter Response Surface model name *

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.

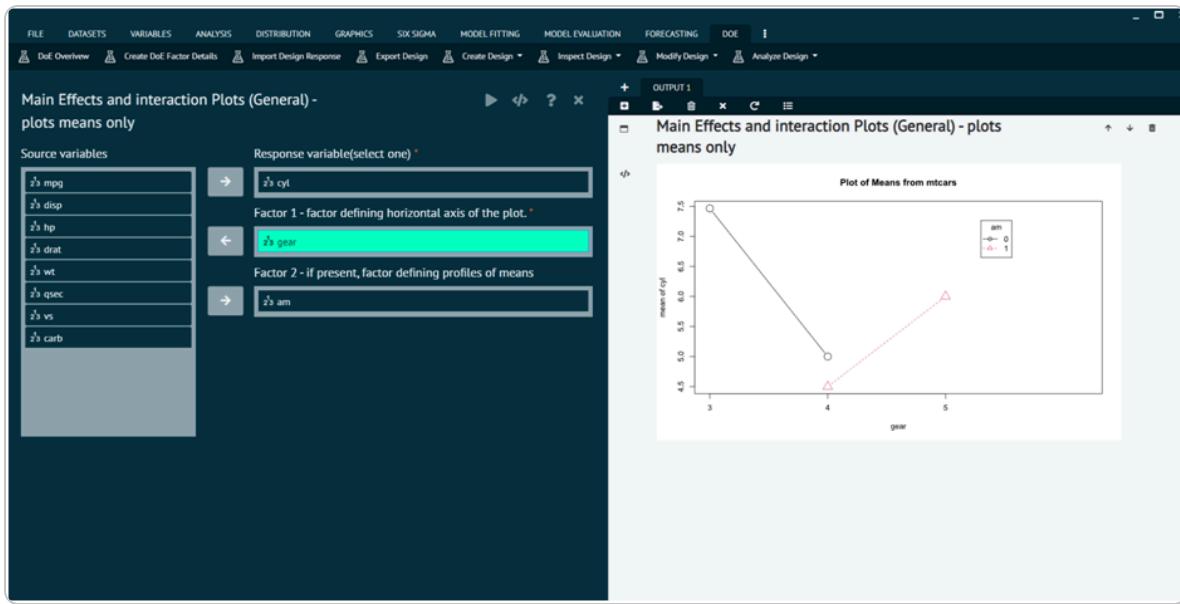
+	-	*	/	
()	%in%	I	:
All 2 ways		Polynomial terms 2		
df for splines 5		Polynomial degree 5		
B-spline		Natural spline		
Orthogonal polynomial		Raw polynomial		

Formula appears here

You can also enter Software var2 = 1 for second-order effects and interactions.

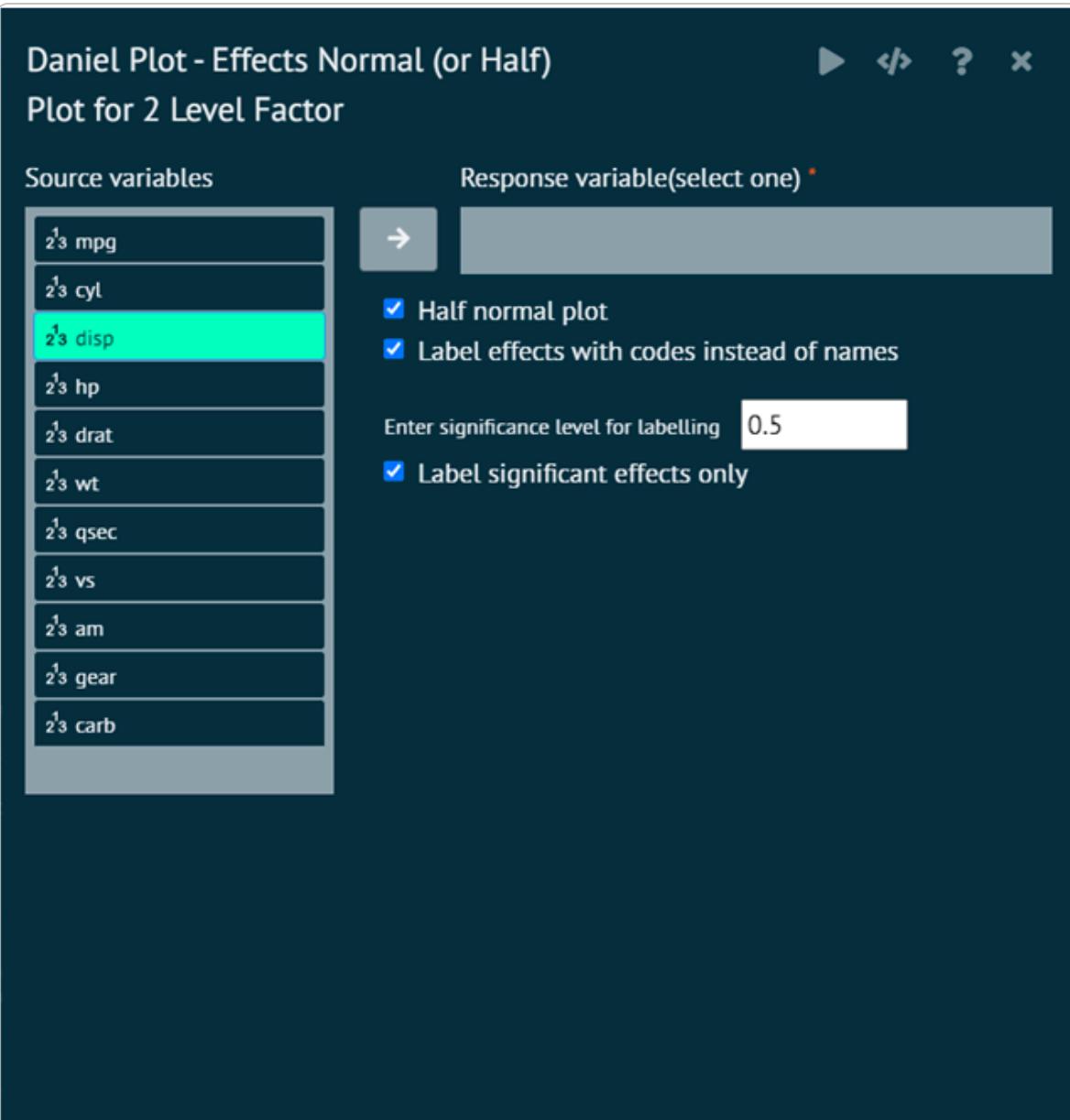
Design Analysis-Response Surface Model

Main Effect And Interaction Plots (General)



Main Effect And Interaction Plots (General)

Half Normal Plot For 2 levels



Half Normal Plot For 2 levels

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 displays the BioStat Prime software interface. The main window shows a data grid titled 'factor_grid_full_factorial_Design_Sheet1' with 16 rows and 4 columns, containing data for factors X2, X3, X4, and X5. The columns are labeled 'X2', 'X3', 'X4', and 'X5'. The data includes various levels such as 'MemPages', '24P', '20P', and '16P'. Below the grid are tabs for 'DATA' and 'VARIABLES'. To the right, an 'OUTPUT' window is open, showing the process of opening a dataset and creating a full factorial design. The output window displays the following text:

```
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_design/factor_grid_full_factorial_Design.xlsx',sheet='Sheet1',
col_names=FALSE)"
```

Below this, under 'Create Full Factorial Design', it says 'creating full factorial with 16 runs ...'

Full Factor Analysis (in detail)

The user can now inspect the design as shown below.

Inspect Design

- Show design type and response variables
- Print Design
 - Print Design with Run Order
- Summarize (brief)
- Summarize (detail)
- Advance option - Details of the design object with design.info()

Design Type: full factorial and Number of Runs: 16

X2	X4
DeckArrange	MemPages
GROUP	24P
FREQY	20P
ALPHA	16P

Design Type: full factorial and Number of Runs: 16

X2	X4
1	DeckArrange MemPages
2	GROUP MemPages
3	FREQY MemPages
4	ALPHA MemPages

Full Factor Analysis (in detail) output1

Inspect Design

- Show design type and response variables
- Print Design
 - Print Design with Run Order
- Summarize (brief)
- Summarize (detail)
- Advance option - Details of the design object with design.info()

```

Call:
FullFactorialDesign <- DoE.base::fac.design(nfactors =
factorParam$nFactors, replications = 1,
repeat.only = FALSE, blocks = 1, randomize = FALSE, seed = 1234,
nlevels = factorParam$nlevels,
factor.names = factorParam$factor.names)
Experimental design of type full factorial
16 runs
Factor settings (scale ends):
      X2      X4
1 DeckArrange MemPages
2      GROUP    24P
3     FREQY    20P
4     ALPHA    16P
Call:
FullFactorialDesign <- DoE.base::fac.design(nfactors =
factorParam$nFactors, replications = 1,
repeat.only = FALSE, blocks = 1, randomize = FALSE, seed = 1234,
nlevels = factorParam$nlevels,
factor.names = factorParam$factor.names)
Experimental design of type full factorial
16 runs
Factor settings (scale ends):
  
```

Full Factor Analysis (in detail) output2

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.

Quality Assurance

Quality control (QC) in biostatistics is critical for ensuring the accuracy, reliability, and validity of data analyses. Methods of QC typically focus on both data integrity and the analytical processes used in biostatistical studies.

In the BioStatistics sector, upholding stringent quality and compliance standards is vital to safeguarding the effectiveness and safety of medicinal products. With the growing complexity of manufacturing workflows and regulatory frameworks, organizations are increasingly adopting sophisticated statistical software to optimize their quality control efforts and simplify the Product Quality Review process. These advanced solutions offer comprehensive data analysis features, empowering manufacturers to process large volumes of information, uncover patterns, and make informed decisions that drive product excellence and ensure adherence to regulatory requirements.

- A** Quality control is not just a compliance measure but a strategic enabler across industries. By integrating QC into their workflows, professionals can drive operational excellence, foster innovation, and deliver superior outcomes, ensuring sustained success in competitive environments.

Quality control (QC) serves as a cornerstone for maintaining consistency, reliability, and accuracy in professional settings, ensuring that processes and outputs meet established standards.

Applications

1. Healthcare and Biostatistics Data Integrity: QC ensures data collected in clinical trials or epidemiological studies is accurate and consistent, supporting valid conclusions.
Clinical Trials: QC validates the adherence to protocols, randomization processes, and data collection standards. Laboratory Processes: Monitoring assay performance, calibration of instruments, and reagent quality using statistical control charts.
2. Manufacturing and Production Process Monitoring: Real-time monitoring of production lines to ensure products meet predefined specifications. Defect Reduction: Identifying and eliminating defects through sampling and analysis. Regulatory Compliance: Meeting standards such as ISO or FDA guidelines.

3. Software Development Bug Detection: Regular code reviews and testing frameworks ensure that applications are free of critical errors. Version Control: Maintaining the integrity of software through systematic updates and error tracking. User Experience (UX): Ensuring features function as intended by testing across multiple environments.
4. Education and Research Research Validation: QC protocols ensure the reproducibility and reliability of experimental results. Assessment Accuracy: Maintaining consistency in grading systems or evaluation metrics. Publication Standards: Adhering to ethical and methodological standards in academic writing.
5. Business and Service Industries Customer Feedback Analysis: Systematic QC of survey data to ensure insights are based on accurate interpretations. Performance Metrics: Regular evaluation of key performance indicators (KPIs) to improve service quality. Supply Chain Management: Ensuring that incoming materials meet quality standards to minimize downstream disruptions.

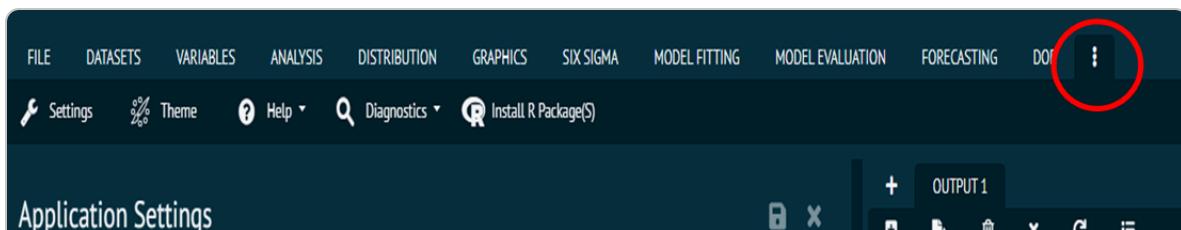
Benefits of Quality Control

1. Improved Reliability QC minimizes errors and variability, leading to consistent outputs. Builds trust with stakeholders by ensuring the dependability of results or products.
2. Cost Efficiency Early identification of defects prevents expensive rework or product recalls. Optimized processes reduce waste and operational costs.
3. Compliance with Standards Helps meet regulatory and industry-specific standards, avoiding legal or financial penalties. Demonstrates commitment to excellence and ethical practices.
4. Enhanced Decision-Making High-quality data provides a solid foundation for strategic decisions. Robust QC processes reduce uncertainty and bias in analytical outcomes.
5. Customer Satisfaction Consistently delivering high-quality products or services improves brand reputation and customer loyalty. Resolving quality issues proactively strengthens client relationships.
6. Operational Excellence Encourages a culture of continuous improvement through systematic monitoring and refinement. Facilitates the identification of bottlenecks and inefficiencies in workflows.

7. Competitive Advantage Quality assurance becomes a differentiator in markets where precision and reliability are critical. Enables organizations to innovate confidently, knowing their foundation is secure.

Settings Menu

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.

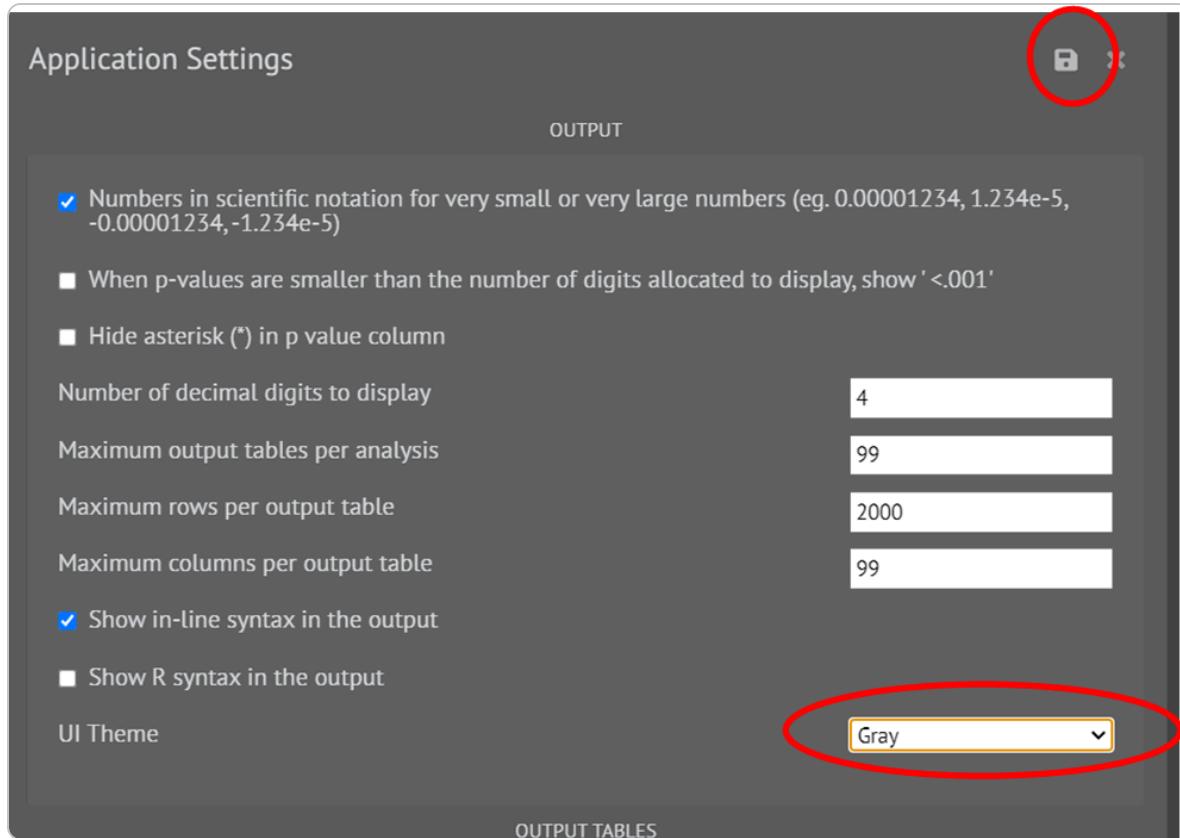


Settings Menu

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.



OUTPUT TABLES

Settings

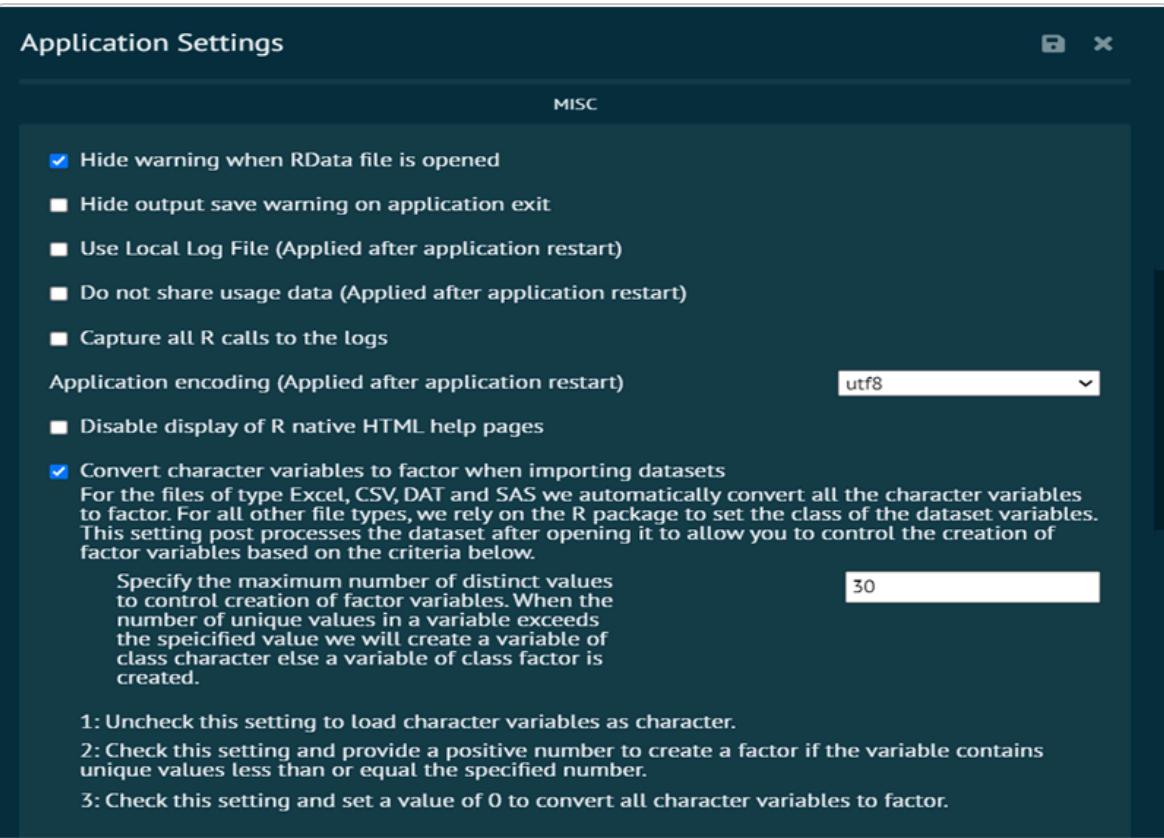
The icon at top right corner is used to save the changes.

OUTPUT TABLES: Used to modify settings related to theme, font, LaTeX of the tables that appear in the output.



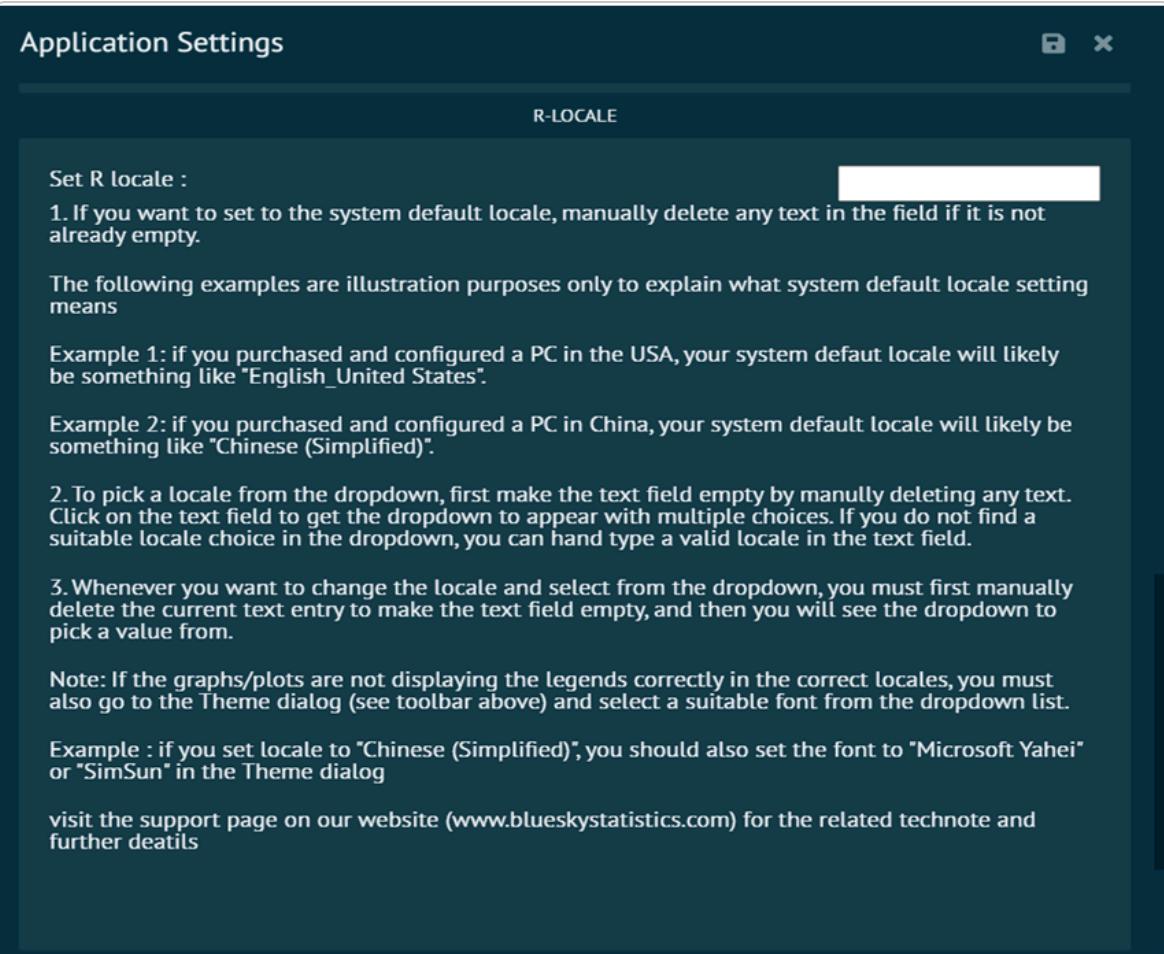
OUTPUT TABLES

MISC: This section of the settings enables the user to modify miscellaneous settings of the application.



MISC

R-LOCALE: Used for language setting of R engine.



R-LOCALE

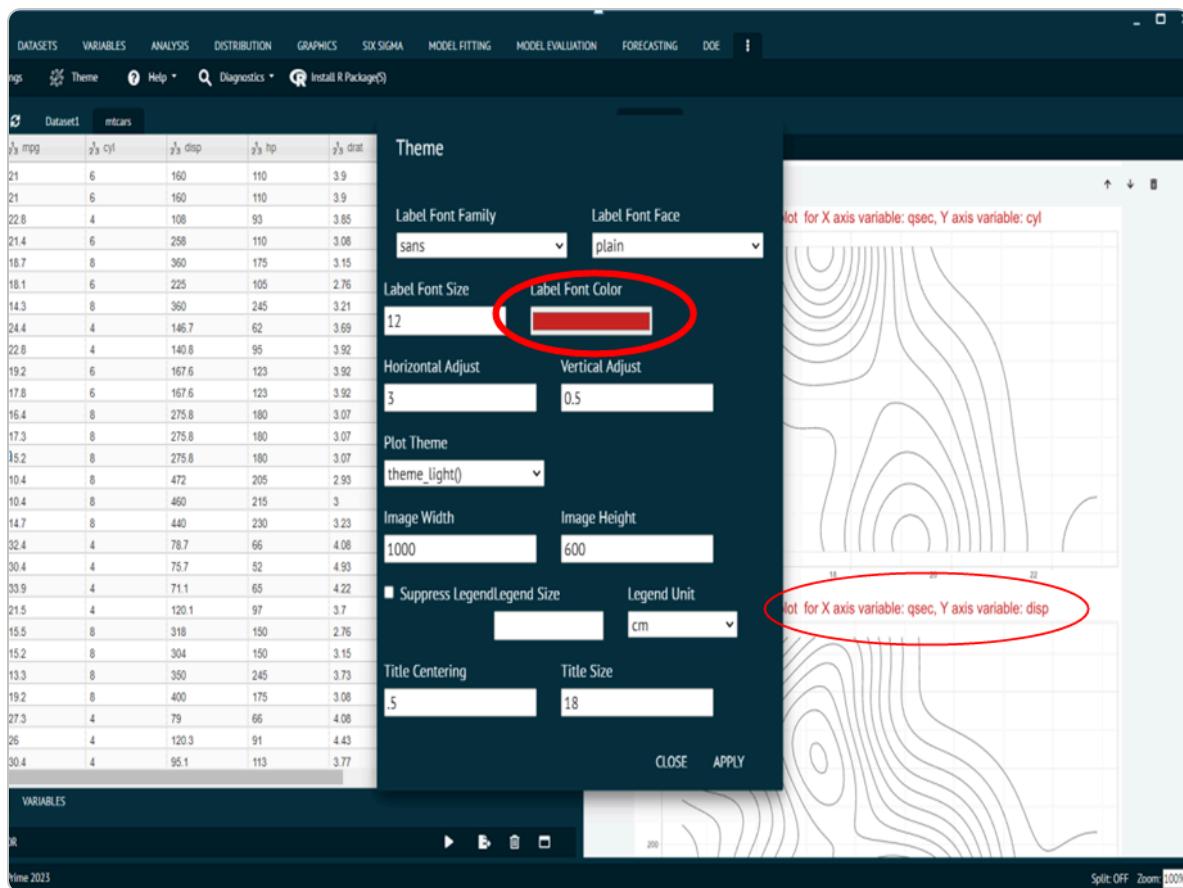
DATABASE: Enables the user to choose the location for database.



DATABASE

Themes

This section of the software provides the user with ability to customize the look of the output labels.

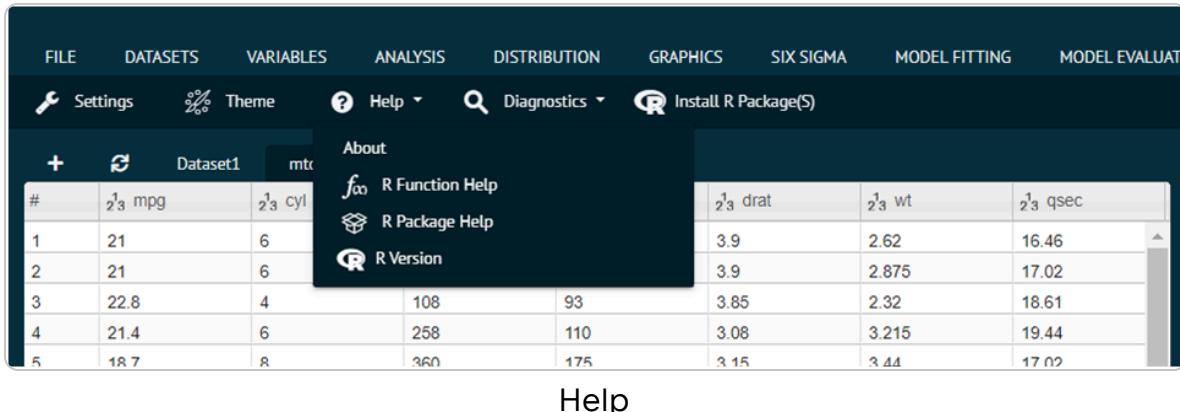


Themes

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.



Diagnostics

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
```

Diagnostics

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					
List of Installed Packages in path C:/Program Files/BioStat Prime/10/resources/package/R-4.1.3/lib					
	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

Diagnostics

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.

The screenshot shows the BioStat Prime software interface with the 'Install R Package(s)' dialog box open. The menu bar includes FILE, DATASETS, VARIABLES, ANALYSIS, DISTRIBUTION, GRAPHICS, SIX SIGMA, MODEL FITTING, and MODEL EVALUATION. Sub-menu items like Settings, Theme, Help, Diagnostics, and Install R Package(S) are also visible. The dialog box has a title bar 'Install R Package(s)' with icons for back, forward, help, and close. It contains notes about case sensitivity and examples, a text input field for package names separated by commas, a note about write access to the path, a dropdown for selecting the R library path (with options like 'C:/Users/zakin/Documents/R/win-library/4.1' and 'C:/Program Files/BioStat Prime/10/resources/package/R-4.1.3/library'), a URL input field for the CRAN repository (set to 'https://cloud.r-project.org/'), and a note about the R logo. A large 'Install R package(s)' button is at the bottom.

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

🔧 Settings 🌈 Theme 🌐 Help 🔎 Diagnostics 📦 Install R Package(S)

Install R Package(s)

▶ ⌂ ? ✕

NOTE: R package names are case sensitive. You can install multiple package names by using , as a separator
Example 1: foreign
Example 2: foreign, car, MASS

Please enter the name(s) of one or more package(s) separated by comma *

NOTE: You must have write access to the path selected below else relaunch the application by selecting the run as administrator option

Select the R library path where the R package will get installed *

C:/Users/zakin/Documents/R/win-library/4.1
C:/Program Files/BioStat Prime/10/resources/package/R-4.1.3/library

Enter the URL of the CRAN repository, the default <https://cloud.r-project.org/> installs from the CRAN mirror closest to you. *

<https://cloud.r-project.org/>

NOTE: The R logo that displays on the menu bar is the copyright property of the R foundation

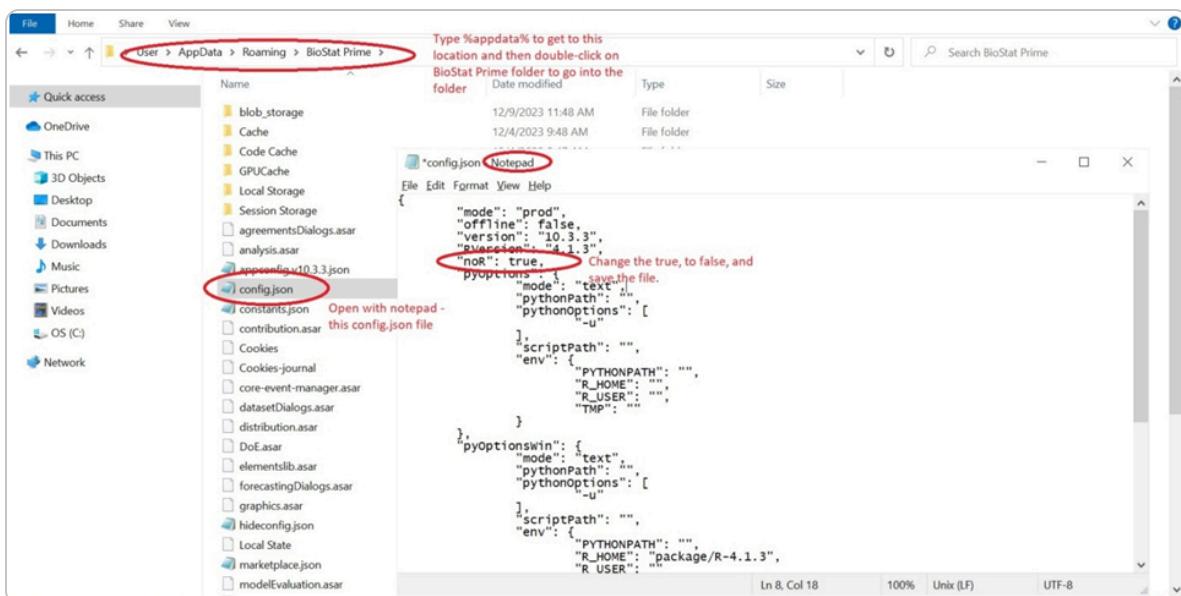
Install R package(s)

Advanced Functionalities

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.

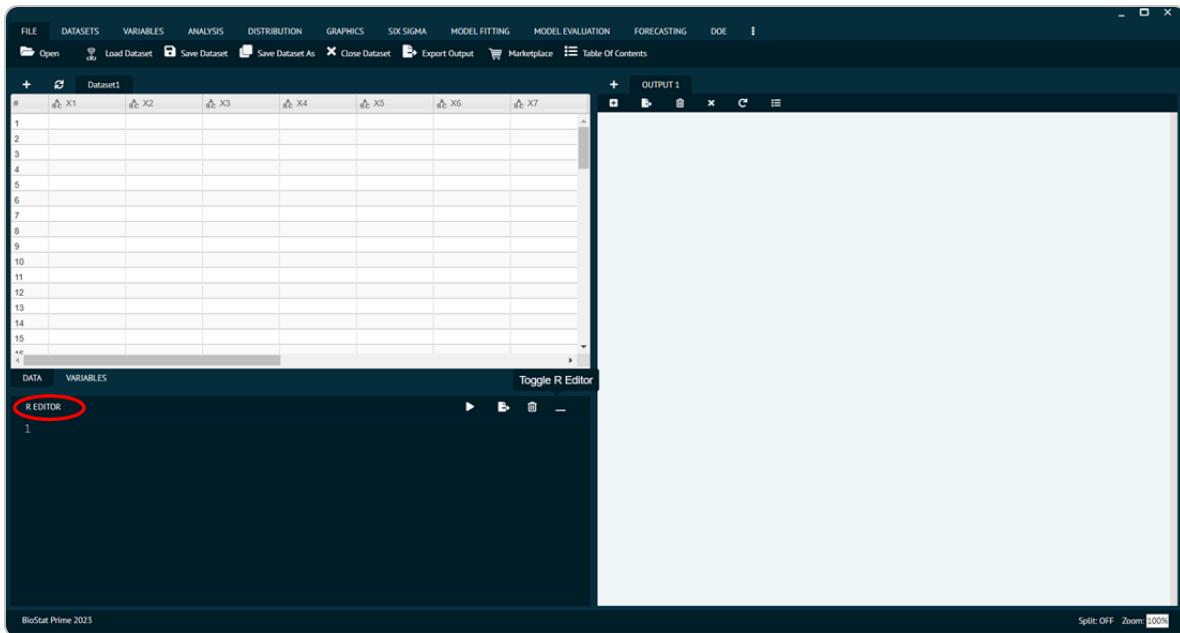
How to Enable/ Disable R console

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.



How to Enable/ Disable R console

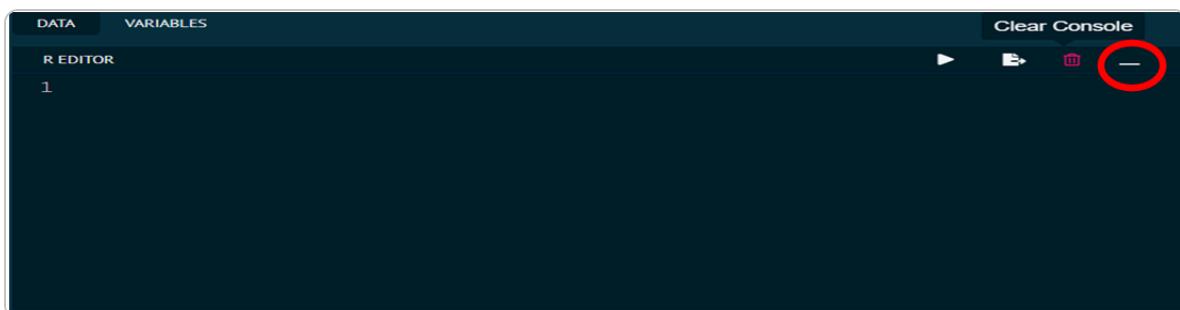
Exit the BioStat Prime app if it is running before making the change to the configuration flag. Once user saves the configuration -> restart the app -> see the R Editor panel. Do the same steps to reverse the configuration to hide the R Editor.



R console

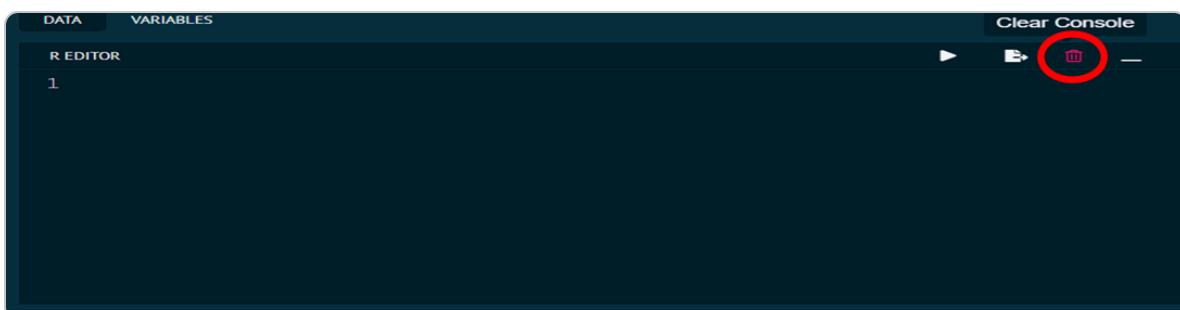
The options at top right corner of the R console are (from right to left).

Toggle R Editor: Used to minimize or maximize the R console.



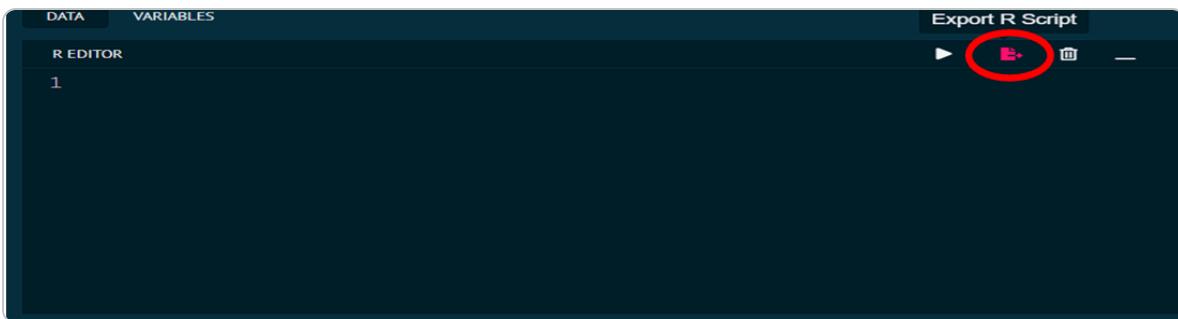
Toggle R Editor

Clear Console: Clears the entire code in R console.



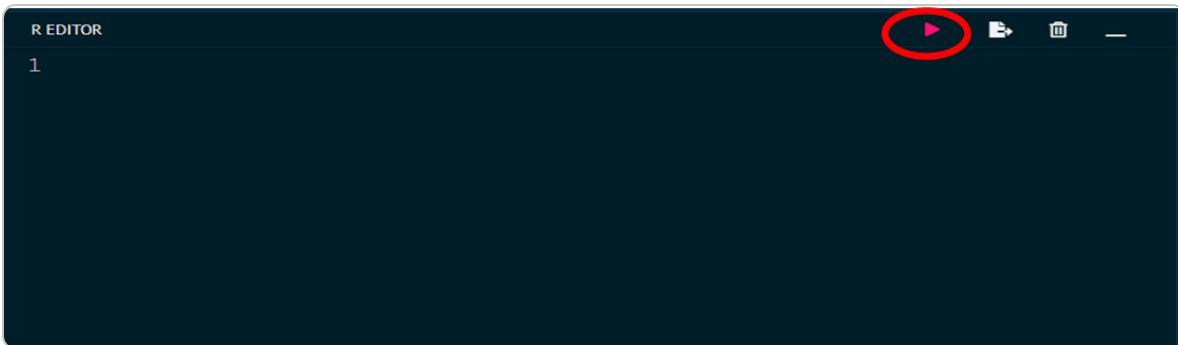
Clear Console

Export R Script: Used to save R script exporting it to the PC/Laptop.



Export R Script

Execute Button: Executes the R script.



Execute

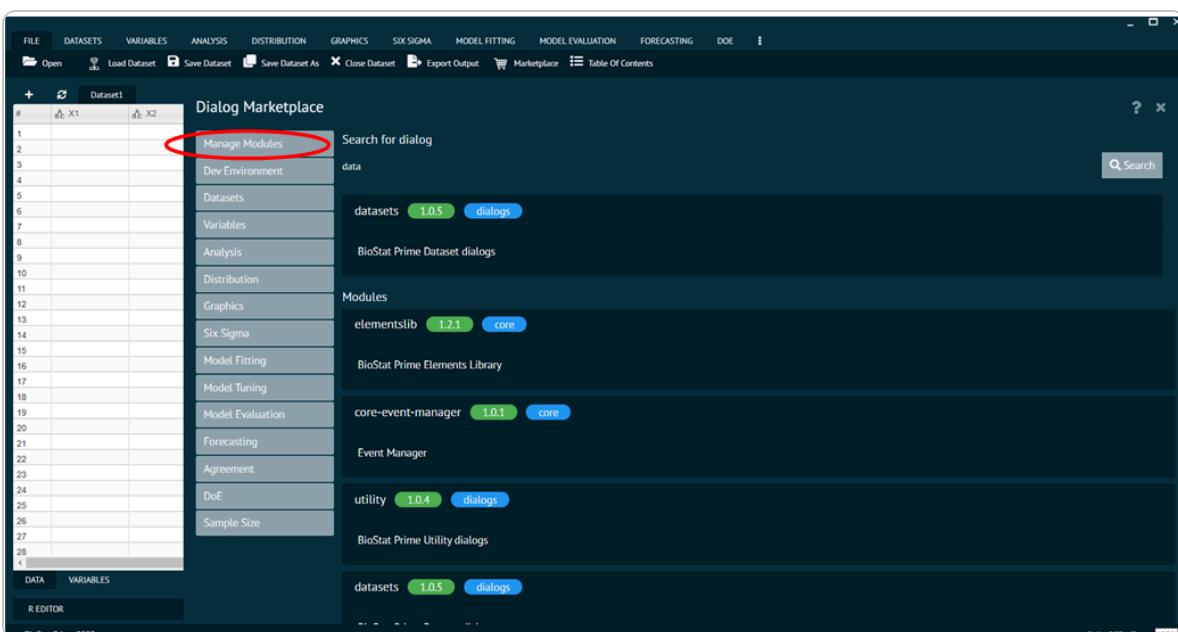
Marketplace

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.

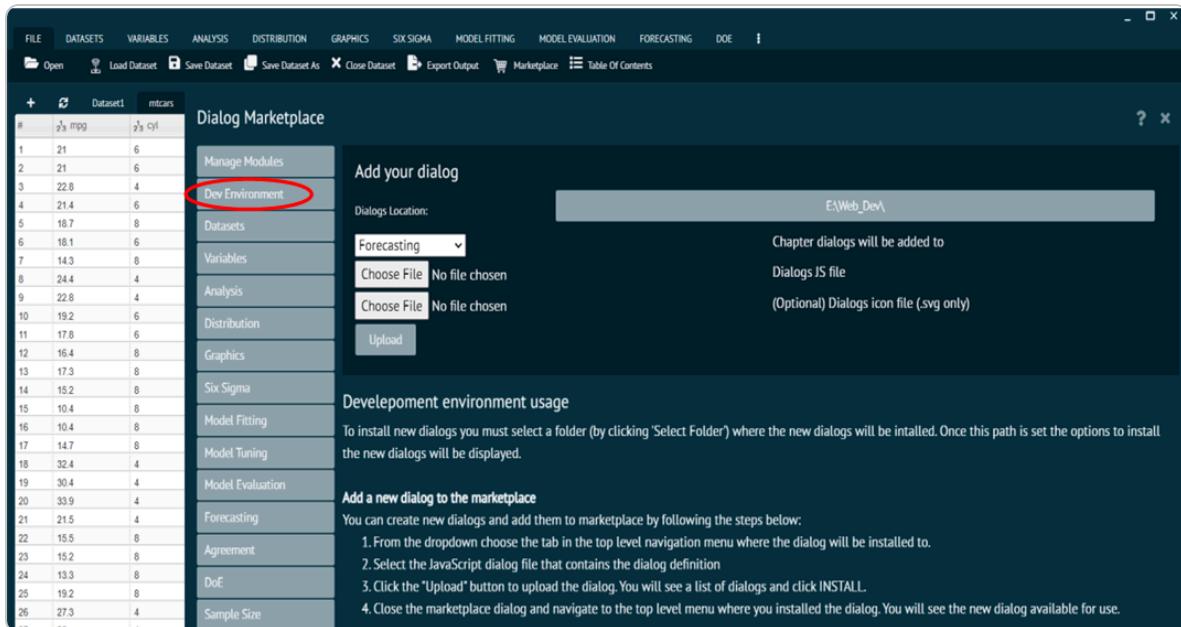


Manage Modules

Dev Environments

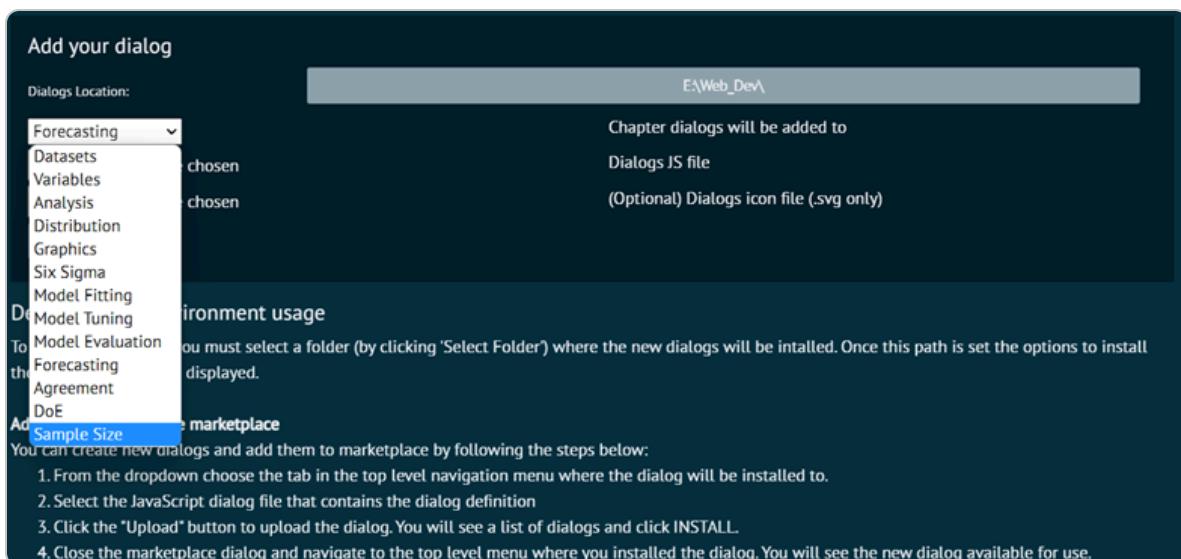
To install the new dialogs, users need to go to the Dev Environment tab in Marketplace section and must pick the location (by clicking "Select Folder") where they want to be installed -> Once the place is specified, the options to install the new dialogs will show up -> Select the tab in the top level 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.



Dev Environments

Selecting the tab where the dialog will be installed.



Dev Environments

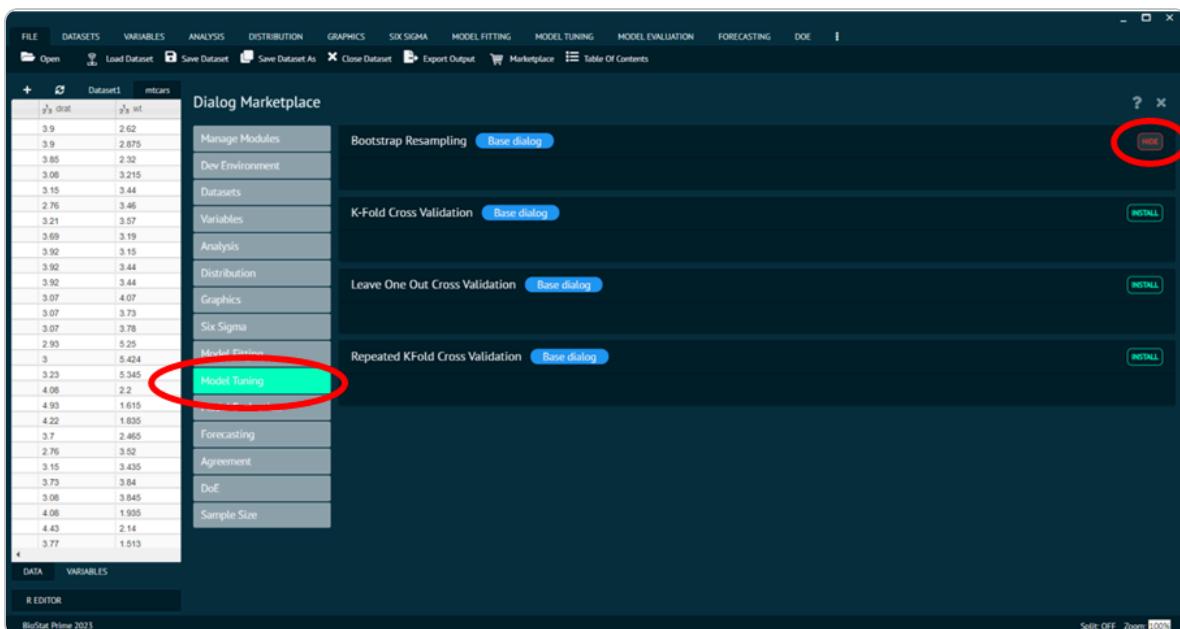
Installing R Libraries from Marketplace

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.

Steps

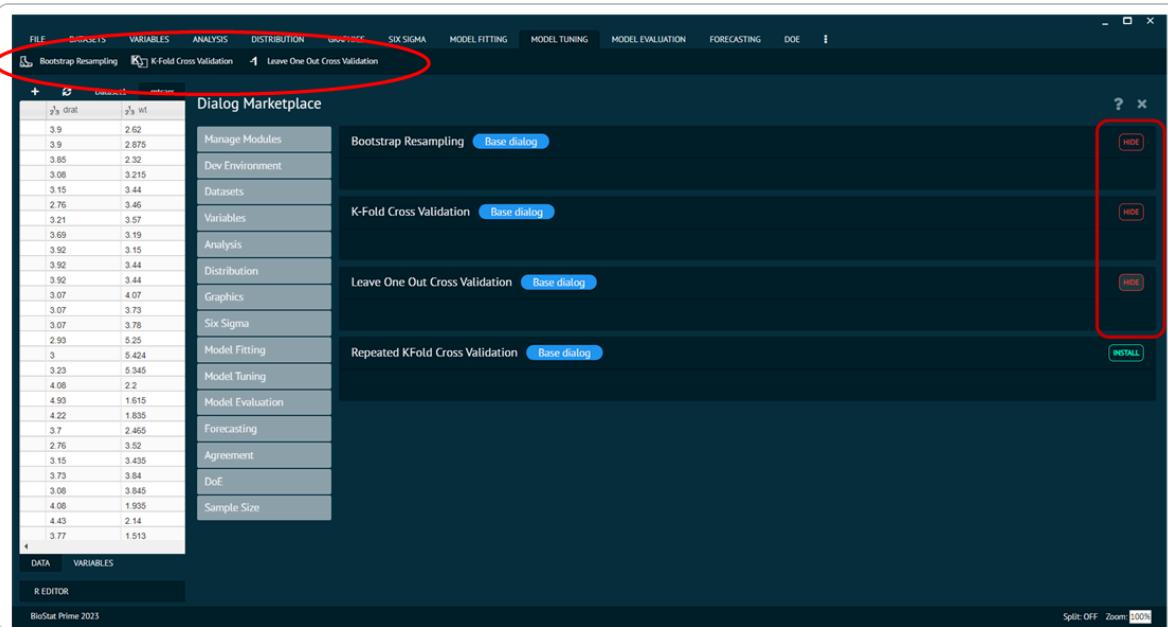
Go to file Menu -> Marketplace -> Choose the package to be installed (say Model Tuning) -> Click install next to respective functions that user wants in the sub menus.

BioStat Prime will add the library in the main menu and its functions in sub menu.



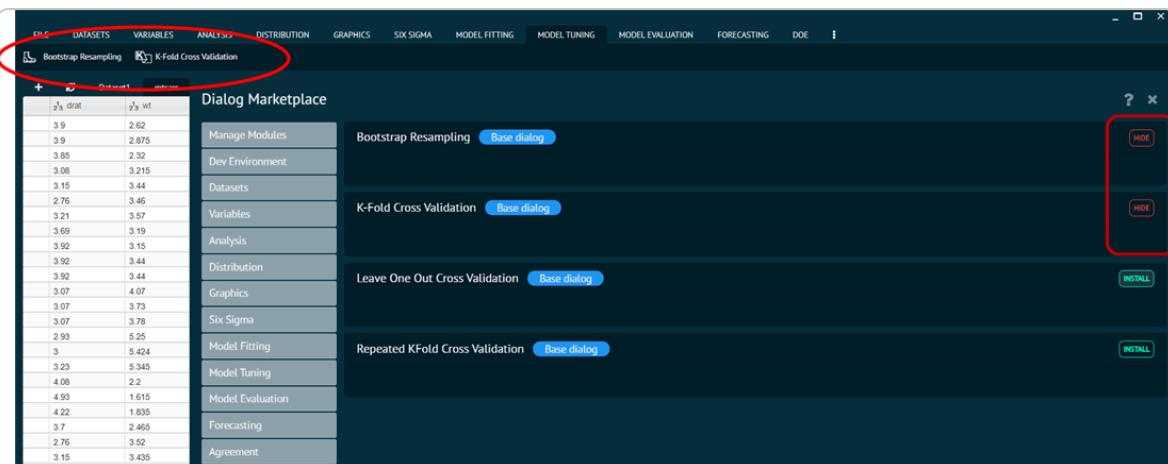
Installing R Libraries from Marketplace

As the user proceeds to install the packages, the functions appear in the main menu and sub menu.



Installing R Libraries from Marketplace

User can hide the sub function whenever needed, by clicking the hide button next to the respective function in marketplace.



Installing R Libraries from Marketplace