BlueSky Statistics 7.0 Release Notes

Enhancements - Machine learning

We have expanded the choices under Model Fitting to

1. Train a neural network model using the `neuralnet` package.
2. Train a multi-layer perceptron using the RSNNS package
3. Extreme Gradient Boosting using the `xgboost` package

These dialogs provide direct control over their R functions, providing great flexibility. However, since each function has its own author, their use has unique rules, some of which are described below (also see help files on each dialog box). The Model Tuning menu drives modeling via the caret package, which standardizes the interface to these same functions, but which also offers less detailed controls.

1. Train a multi-layer perceptron using the RSNNS package, see "Model Fitting > Neural Nets > NeuralNets". Sample datasets can be found in the "C:\Program Files\BlueSky Statistics\Sample Datasets and Demos\Neuralnet" folder. The dependent variable can be numeric or factor. If the dependent variable specified is a factor, we automatically dummy code the factor variable using one-hot Encoding using the `decode` function in the RSNNS package.

   Additionally, if you are using one-hot encoding to dummy code a factor variable, you can specify more than one dependent variable in the dialog. In this case, the dependent variables must be of type numeric.
   You can use "Data > Compute dummy variables", choose the “Keep all levels” setting to get one-hot encoding.

   For dependent variables of type factor, we will display a confusion matrix, ROC and model accuracy statistics when scoring a dataset using the model built. The predictions generated are of type factor since we predict the class. These will be saved in the dataset along with the predicted probabilities when scoring.

   When there are dummy coded dependent variables, we will not display a confusion matrix, ROC and model accuracy statistics when scoring a dataset using the model built. However, the predictions will be saved in the dataset when scoring the dataset. The predictions are the probabilities associated with the dummy coded dependent variables.

   It usually best to standardize independent variables (they must be numeric, too) See “Data > Standardize Variables.”
   If you have categorical independent variables, use one-hot encoding to dummy code the factor variables.

2. Train a neural network model using the `neuralnet` package, see “Model Fitting > Neural Nets > Multi-layer Perceptron”. Sample datasets can be found in the “C:\Program Files\BlueSky Statistics\Sample Datasets and Demos\Neuralnet” folder.

The dependent variable can be numeric or factor. If the dependent variable is a factor, we dummy code the factor variable using one-hot encoding using the decode function in the RSNNS package.

Additionally, if you are using one-hot encoding to dummy code a factor variable, you can specify more than one dependent variable in the dialog. In this case, the dependent variables must be of type numeric.
You can use "Data > Compute dummy variables," select the "Keep all levels" option for one-hot encoding.

For dependent variables of type factor, we will display a confusion matrix, ROC and model accuracy statistics when scoring a dataset using the model built. The predictions generated are of type factor since we predict the class. These will be saved in the dataset along with the predicted probabilities when scoring.

When there are dummy coded dependent variables, we will not display a confusion matrix, ROC and model accuracy statistics when scoring a dataset using the model built. However, the predictions will be saved in the dataset when scoring the dataset. The predictions are the probabilities associated with the dummy coded dependent variables.

It usually best to standardize independent variables (they must be numeric, too) See "Data > Standardize Variables."
If you have categorical independent variables, use one-hot encoding to dummy code the factor variables.

3. Extreme Gradient Boosting using the `xgboost` package, see "Model Fitting > Extreme Gradient Boosting." For predicting dependent variable of type factor, you need to recode the dependent variable to a numeric with values starting from 0. For example, if there are 3 levels in the factor variable, the numeric variable must contain the values 0,1,2. See "Data > Recode Variables." Alternately, just convert the factor variable to numeric, typically the levels will get mapped to integers starting from 1, and then subtract 1 from the resulting variable to get numeric values starting with 0. This will give you a numeric variable with values starting from 0.

   You need to dummy code independent factor variables, use one-hot encoding see "Data > Compute Dummy Variables."

4. Support a broader set of models in Model Tuning > Bootstrap Resampling, Model Tuning > k-Fold Cross Validation, Model Tuning > Leave One Out Cross Validation, Model Tuning > Repeated K-Fold Cross Validation. Scoring datasets is supported with these tuned models
   a. Gradient Boosting Machines with the gbm package
   b. Extreme Gradient Boosting with the XGBoost package
   c. Stepwise model selection with AIC with the MASS package

d. Conditional Inference Trees with the party package
e. Robust Linear regression with the MASS package
f. Lasso and Elastic-Net Regularized Generalized Linear Models with the glmnet package
g. Multi-variate Adaptive Regression Spline with the earth package
h. Neural Net (Single hidden layer) with the nnet package
i. Neural Net (Train neural nets using backpropagation, RPROP, GRPROP) with the neuralnet package

5. In the scoring section on the top right-hand part of the main application window, renamed Model Type to Model Class.

6. Added a Help button below Scoring button.

Enhancements – General

1. Added a Font icon on the ribbon bar of the R Syntax editor window to increase the Font of the syntax displayed. This was provided for instructors to easily display syntax during lectures.

2. Integration with QuestionPro Datapad, see https://www.questionpro.com and https://www.questionpro.com/help/datapad.html. This allows you to bring in survey data directly into BlueSky Statistics for analysis, see "File > Open QuestionPro Dataset" and on the Output window, "File > Export Output to QuestionPro" to save the results of the analysis back to the QuestionPro Datapad.

3. When saving a dataset to an RData file, we rename the dataset/data frame to the name of the R data file. This is to prevent overwriting existing datasets in the situation when you create a new dataset via File > New Dataset, call it test1, close the application. Now restart the application, create another dataset from File-> New Dataset, and call it test2. In the prior release, the datasets were called Dataset1 and Dataset1 although they were saved to test1 and test2 respectively. When you opened test2 after opening test1, it would overwrite Dataset1. In the new release you can open files named test1 and test1 (from different folders), since the R objects contained within them have different names, nothing will be over-written.

4. Made improvements to the installation of BlueSky Statistics. We were looking for the path of R using a registry key which was no longer necessary. We have also created a technote at https://www.blueskystatistics.com/v/vspfiles/downloadables/BlueSky-R-Session-Creation-Steps.pdf that lists in detail how we create an R session when the application is launched.

5. Reversed the direction of the navigation tree icon displayed on the top right of the output window to match the way the arrow on the top left works.

6. Wrapped menus in the main application window when shrinking the main application window.

7. A warning message is displayed when a user tries to save a blank dataset.

8. We now support r data files with extension ".rda".

Enhancements – Statistics

1. The interaction plots in Mixed Models now has the “Force Continuous” option.

2. Moved the “Crosstab, Two-way' dialog to Analysis > Contingency Tables > Legacy > Crosstab 2 way.” You should instead use, “Analysis > Contingency Tables >Crosstab, Multi-way.”

3. Changed the captions in the Alternate Hypothesis in the “Analysis> Means> Independent Samples T-Test” to “group1 != group2”, “group1 > group2”...

4. Added summary statistics to “Analysis > Means > T-Test, paired samples.”

5. A sample dataset has been added to test Bland-Altman plots: bland.altman.PEFR.1986.RData located in: “C:\Program Files\BlueSky Statistics\Sample Datasets and Demos\BlandAltman.”

Enhancements – Data Manipulation

1. The placement of the “Compute Dummy Variables” dialog has been changed to “Data > Compute Dummy Variables.”

2. When opening a dataset, we now display the first 40 columns/variables in the first page of the data editor window. You will need page through the dataset using the paging button/controls on the bottom right hand corner of the data editor window only if you have more than 40 columns/ variables in the dataset.

3. Renamed “Data > Merge Dataset (tidy)” to “Data> Merge Datasets.” Moved older “Merge Datasets” dialog to “Data > Legacy” and renamed it to “Merge Datasets (legacy)”

4. A warning message is displayed when a user tries to save a blank dataset

Bug Fixes

1. Fixed an issue when plotting both continuous and discrete distributions where the lower and upper bounds of the sequence was getting generated incorrectly. The parameters entered in the dialog were not getting passed to the quantile function of the desired distribution. This impacted the dialogs under Distribution > Continuous and Distribution > Discrete that plotted distributions.

2. The GroupBy control in "Analysis Proportions > Proportion Test, independent samples" was restricted to factors with 2 levels; this has been fixed.

3. For "Analysis > Means > T-Test, One Sample" and "T-Test, Independent Samples," the column header for the p-value was displaying Sig(2-tail) even when the alternate hypothesis was Population Mean > mu and Population Mean < mu. This is fixed.

4. Analysis functions like Correlation was not working when you split a dataset of class tbl_df. This has been fixed. For example when you converted an existing dataset using Reshape, the resulting dataset was of class tbl_df. When this was split using Data > Split dataset >For Group by analysis, and analysis like Correlation was run, an error was displayed.

5. We were unnecessarily printing a count label in the table with "Analysis > Contingency Tables > Crosstab, multi-way." This has been fixed.

6. When building a model using a formula, for example, if the model for Y is a 2nd order polynomial regression on X, without the intercept (i.e. the formula is lm(Y ~ -1 + X + I(X^2) or lm(Y ~ -1 + poly(X,2,raw=T)), when you score a dataset using the model the "Make Prediction" window would display an error, "The predictor variables that the model requires for scoring are non available in the dataset. [1,poly,X,2, raw = T variables are not found]. This has been fixed.

7. In "Times Series > Plot Time Series (with Correlations)," ACF and PACF have fractional lags, we have replaced these with integer lags (Acf and Pacf functions).

8. Fixed issue with "Analysis > Time Series > Holtwinters" and "Automated Arima" reporting that the function forecast.Holtwinters and forecast.Arima could not be found.

9. Fixed issue with BSOZ (saved output) not displaying contents of empty tables correctly.

10. Fixed an issue where the decision tree diagram was not getting displayed when the model name was changed from the default treeModel1.

11. Deprecated connection method "dbDriver()" has been replaced with the new "RPostgress()" for creating connection to Postgres, for importing data.