

Shiny Interface to the RobStat™ Package

Gregory Brownson

April 22, 2019

1 Introduction

This document describes a Shiny UI to the **RobStat™** R package companion to the *Robust Statistics: Theory and Methods* book (Maronna et al. 2019 second edition). The shiny application provides an interface with **RobStat™** without needing to set many R function arguments or understanding the complexities of the underlying functions. The simplicity of the UI will allow students and practitioners who are new to robust statistical methods to easily use them and compare robust results with classical methods. Within the Shiny UI, you will be able to load a data set either from an existing R package or a local .csv and then run the data through different methods of analysis. Currently, the UI provides an interface to robust location and scale, robust linear regression, robust covariance estimation, and robust principal component analysis.

Package Dependencies

The Shiny UI depends on the following packages:

- **DT**
- **fit.models**
- **ggplot2**
- **grid**
- **gridExtra**
- **gtools**
- **PerformanceAnalytics**
- **robust**
- **robustbase**
- **shiny**
- **shinyjs**
- **xts**

With the above packages all installed, just use the command `ShinyUI()` at the prompt, to start the Shiny UI application. Then you will be ready to start using the Shiny UI.

Figure 1 displays the initial screen you encounter in the Shiny UI application. This page is where you would select or upload a data set, and then navigate to one of the model tabs for further analysis. The subsequent sections contain detailed instructions for each component of the Shiny UI.

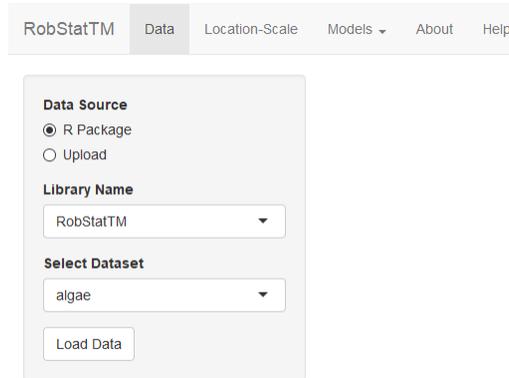


Figure 1: Initial screen after starting the Shiny UI

2 Loading Data

Data may be loaded into the Shiny UI from an existing R package or upload a csv file to the application by selecting either “R Package” or “Upload” as shown in Figure 2. Once you select a package from the list under **Library Name**, the **Select Dataset** input will be populated by the names of existing of datasets in that package. After selecting a dataset, you should click on the **Load Data** button to load the data into the application for further analysis, and the data will be displayed in a table on the right. Figure 2 shows the result for the data set mineral in RobStatTM.

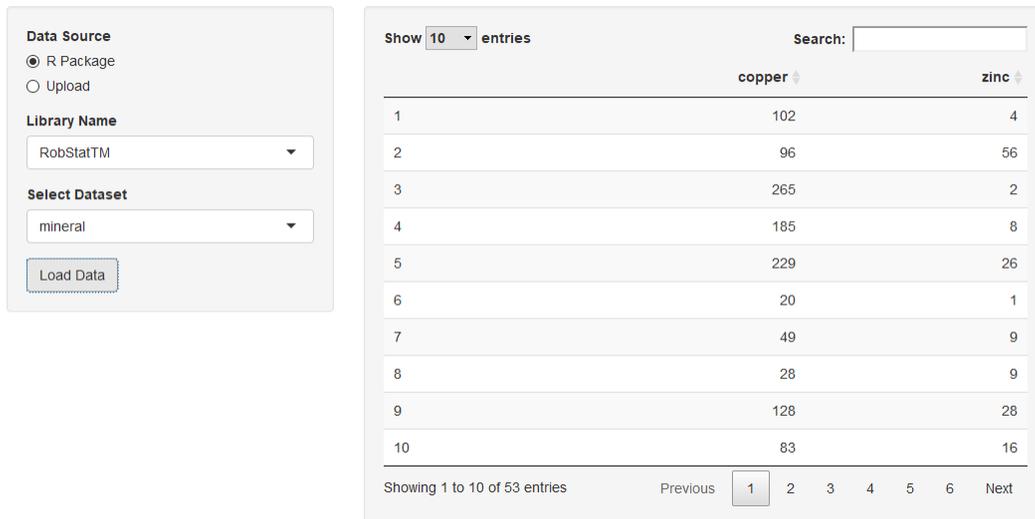


Figure 2: Interface to load a data set from an existing R package

Figure 3 shows the options for uploading a csv file to the application. Since some csv files do not have the default format (comma-delimited values, double quote escape character), advanced options are available to specify the sep-

arator or escape character. If the data is a time series, then you must check the corresponding box as shown in the example. The first column of data must contain the date or time indices to be read as a time series. In the example, we use the data set **hfunds.ts.csv** which contains returns for five hedge funds, and upload it as a time series. The window in the main (right) panel displays the data in a table like format, where you may view the returns for each date. Each column corresponds to one of the five different types of hedge funds: emerging markets (**EM**), Private Equity (**PE**), U.S. High Yield (**USHY**), Alternative Investments (**AI**), and Bond (**BND**). To find the **hfunds** dataset, first you need to type `find.package('RobStatTM')` in the command line to get the path to the **RobStatTM** package. From that location, the file is located in *etc* subdirectory. For datasets with a large number of variables, we suggest viewing the application in **Full Screen**.

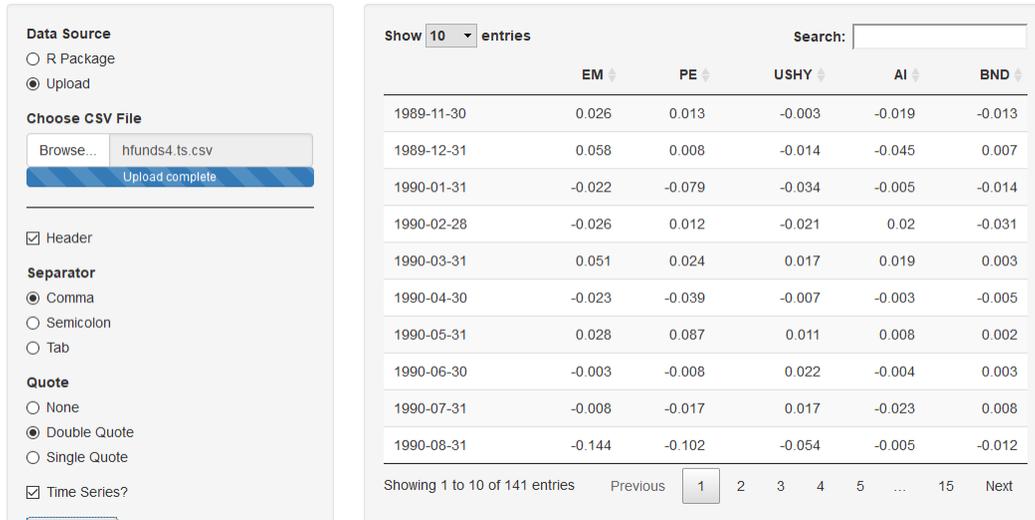


Figure 3: Interface to upload a csv file to the application

3 Robust Location and Scale Estimators

Click on the **Location-Scale** tab to open the dialog which allows you to compute classical estimates for mean and standard deviation and/or robust estimates for location and scale for a single variable, and compare results when you choose to compute both classical and robust estimates.

Computing Location and Scale

The Shiny UI interfaces with the `locScaleM()` function in **RobStatTM** to compute robust estimates of location and scale. Select a **Variable** and choose a **Method** from the options of *Classical*, *Robust*, or *Both*. The classical and robust methods are both computed and then compared when *Both* is chosen. Furthermore, if either *Robust* or *Both* are selected, then additional parameters for tuning the robust method will appear in the bottom of the side panel on the left, when default values **Psi Function** and **Asymptotic Efficiency**. You may choose an alternate psi function from the Psi Function drop-down list, and can choose one of the alternate values for asymptotic efficiency. To compute the estimates and display the summary, click the **Results** button. Figure 4 shows an example comparing the estimates of location and scale for the variable called V1 in the data set **flour** (which you need to load) using for the robust estimate the bisquare score function and an asymptotic efficiency of 90%.

Location-Scale

Calculate the robust location and scale for a single variable in a data set.

Comparison of Location (SE):	
Classical	4.28 (1.08)
Robust	3.15 (0.132)

Comparison of Scale:	
Classical	5.3
Robust	0.678

Figure 4: Comparison of Classical and robust estimates for location and scale for **flour** dataset

Note that the results are displayed in the main panel on the right. This is the layout for each model summary in the Shiny UI. For location and scale, the summary displays estimates for location, the standard error of the location estimate, and the estimate for scale.

4 Robust Linear Regression

The **Robust Linear Regression** tab is located under **Models** tab in the menu as shown in Figure 5 (all subsequent models are located under the **Models** tab). This section allows a simple computation and comparison of classical and robust linear regression methods.

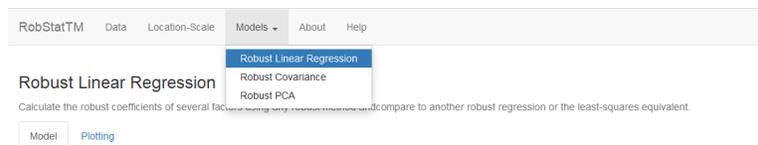


Figure 5: Navigation to **Robust Linear Regression** menu

Fitting Linear Models

There are four linear model fitting methods to choose from in the UI: least-squares (LS), M, MM, and distance constrained maximum-likelihood (DCML). You have the option to compare any two models by checking the **Add Second Method** box and selecting inputs for a second model. In Figure 6, the classical least-squares (LS) and robust MM regression methods are being compared. You must select dependent and independent variables for the model, and then

the regression formula will automatically populate the text box. When comparing regressions, the formulas do not need to be identical. You fit the models and view the summary by pressing the **Results** button at the bottom of the model panel, and then the results will be displayed in the panel that appears at the right.

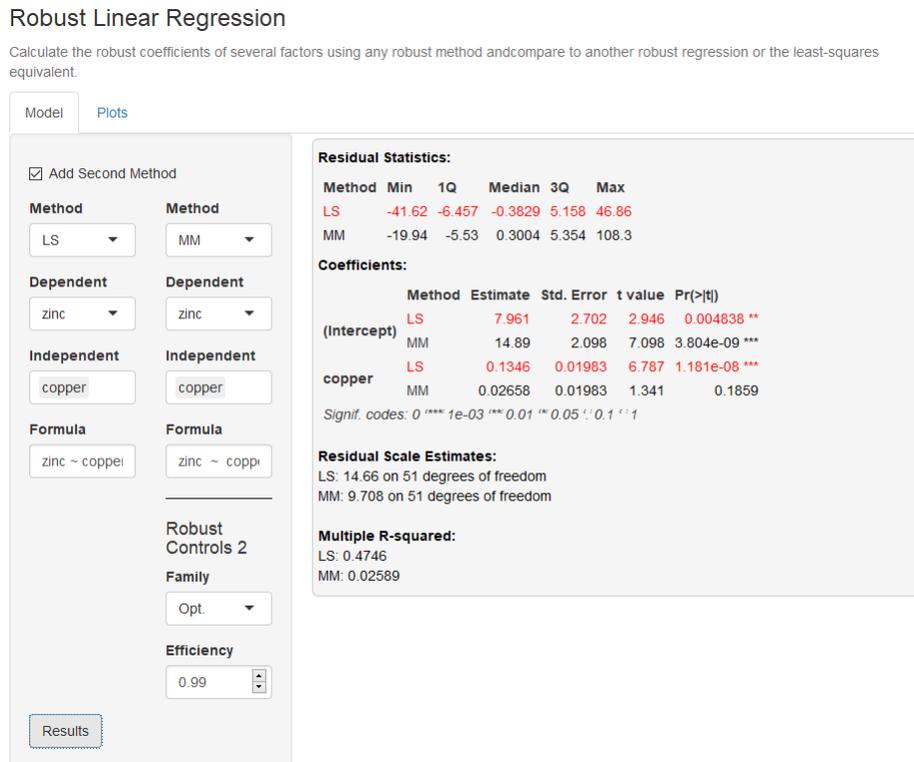


Figure 6: Least-squares and robust MM regression comparison using **mineral** data from **RobStatTM**

The standard errors, t-statistics, and p-values for the robust coefficients for robust fits are computed using a robust covariance matrix for the independent variables, as an important step to ensure that those quantities are robust themselves (see the RobStatTM book Section 5.6 for details). Furthermore, the *proportion of variance explained by the model*, or R^2 , for robust fits is a robust version of classical least-squares R^2 . In the example, zinc is the dependent variable and copper is the independent variable for the LS and MM and for the MM estimator we are using the optimal value for psi with asymptotic efficiency of 99%. The equivalent command-line code is provided below.

```
# Load mineral data set
data(mineral)

# Send fits to fit.models
fm <- fit.models(LS = lm(zinc ~ copper, data = mineral),
                MM = lmrobdetMM(zinc ~ copper, data = mineral,
                               control = lmrobdet.control(family = "optimal",
                                                         eff = 0.99)))

# Comparison of results
summary(fm)
```

Plots

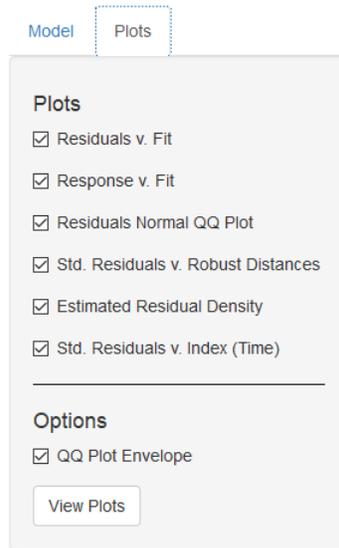


Figure 7: Linear Regression plotting options

After fitting a linear regression model, or a pair of such models, you may graphically analyze the results using the **Plots** tab. In the side-panel, you may choose from the following selection: *Residuals v. Fit*, *Response v. Fit*, *Residuals Normal QQ Plot*, *Estimated Residual Density*, *Std. Residuals v. Robust Distances*, *Std. Residuals v. Index (or Time in the case of time series data)*, and *Scatter with Overlaid Fits*. Note the *Scatter with Overlaid Fits* plot is only available in the case of two univariate regressions. The following are examples of each plot:

Residuals versus Fitted Values

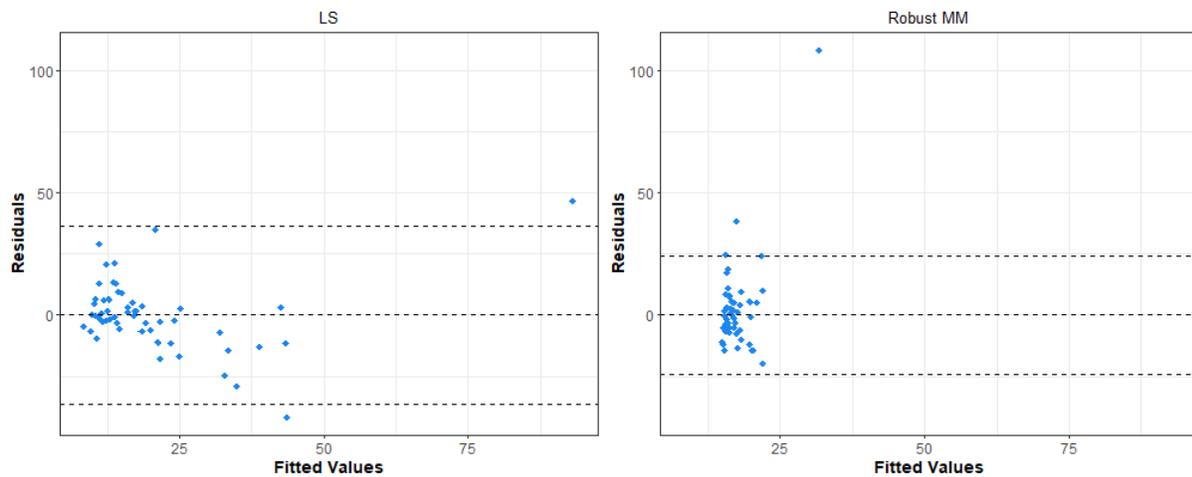


Figure 8: Scatterplot of LS and robust MM residuals versus fitted values: **mineral**

Response versus Fitted Values

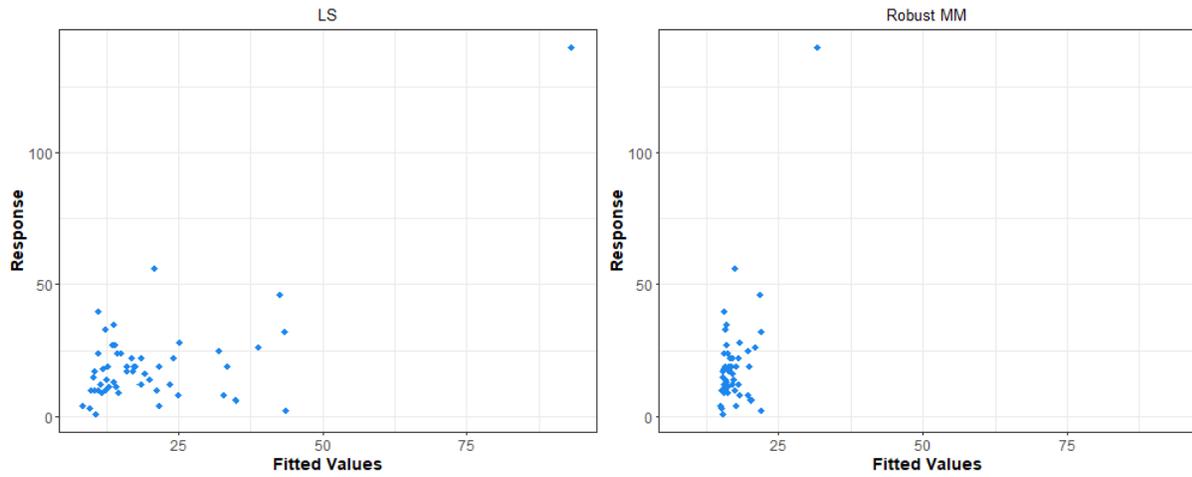


Figure 9: Scatterplot of response values versus LS and robust MM fitted values: **mineral**

Normal QQ Plots of Residuals

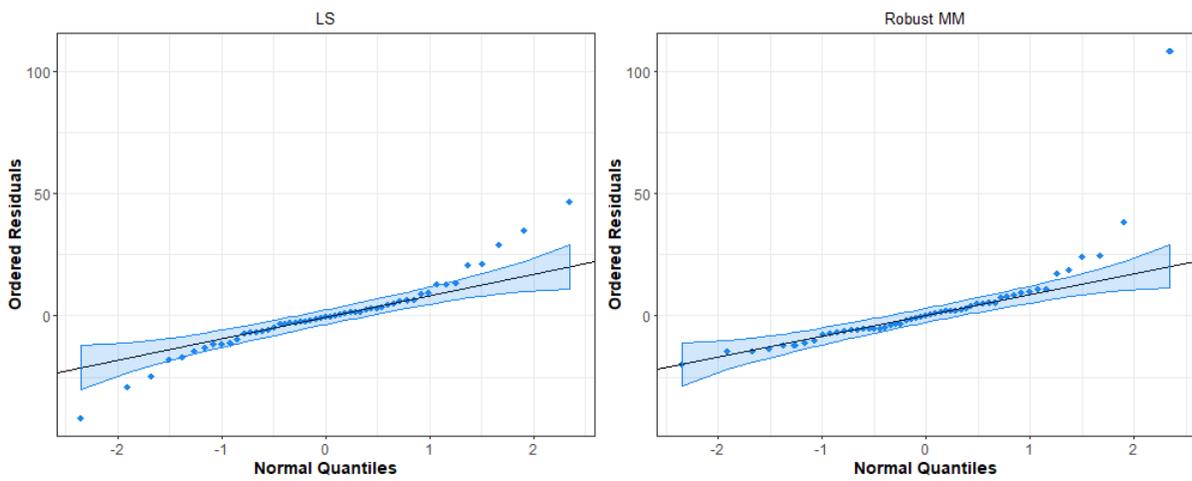


Figure 10: LS and robust MM normal QQ plots of residuals: **mineral**

Standardized Residuals versus Robust Distances

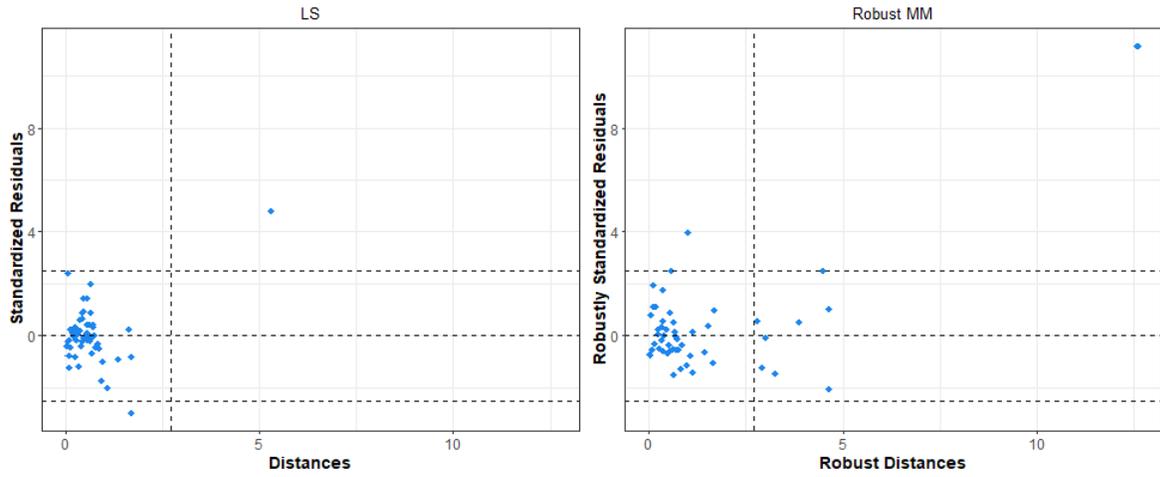


Figure 11: LS and robust MM Plots of standardized residuals versus square-root Mahalanobis distances: **mineral**

Probability Density Estimates of Residuals

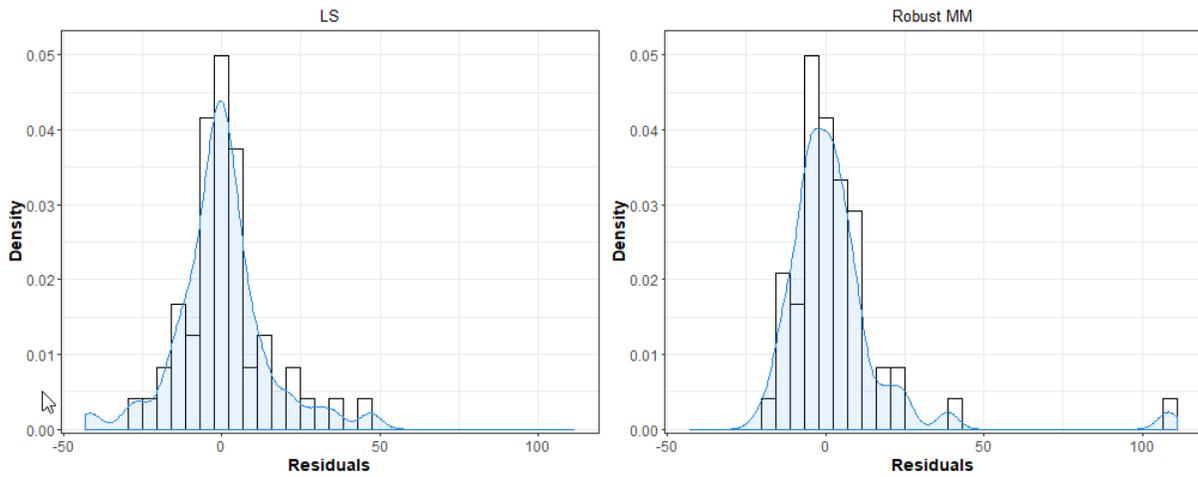


Figure 12: Probability density estimates of LS and robust MM residuals: **mineral**

Standardized Residuals versus Index (Time)

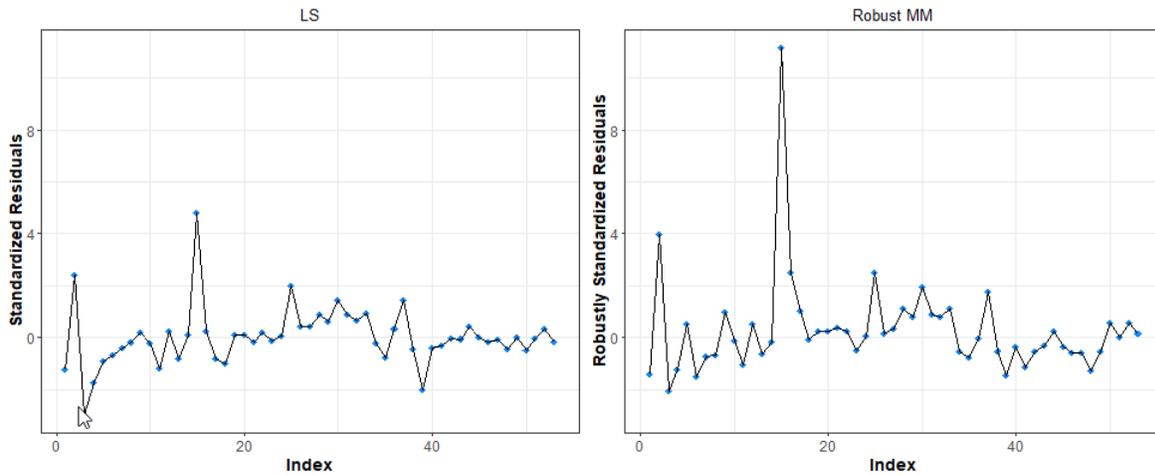


Figure 13: LS and robust MM standardized residuals: **mineral**

Scatterplots with Overlaid Fits

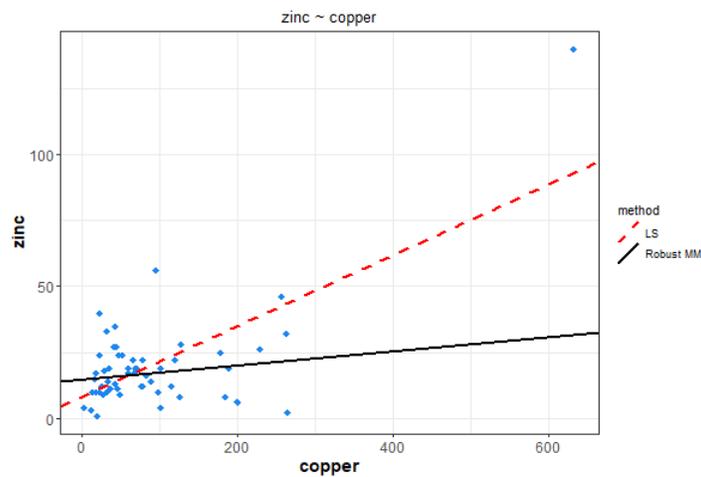


Figure 14: Scatterplot of data with overlaid LS and robust MM fits: **mineral**

5 Robust Covariance

The Shiny UI allows you to use the **RobStatTM** functions `covClassic()` and `covRob()`. First off, since the methods used in **RobStatTM** only allow for numeric data, the options for **Variables** are limited to those with numeric values only. You may choose to compute classical estimates of covariance/correlation, robust estimates of covariance/correlation, or both using the **Method** allows you to choose either the classical method alone, the robust method alone, or use the default of both methods, which is highly recommended. If *Robust*, or *Both* is chosen, then you use the Robust Covariance Estimator drop-down list to make an estimator choice. You may select *MM*, *Rocke*, or *Auto* as the estimator. If *Auto* is selected, then a Rocke-estimator is used if the number of variables is greater than or equal to 10,

and an MM-estimator is used otherwise. Figure 15 displays the comparison of classical and robust covariance matrix and location estimates along with classical and robust eigenvalues associated with those covariance matrix estimate for variables in the **wine** dataset.

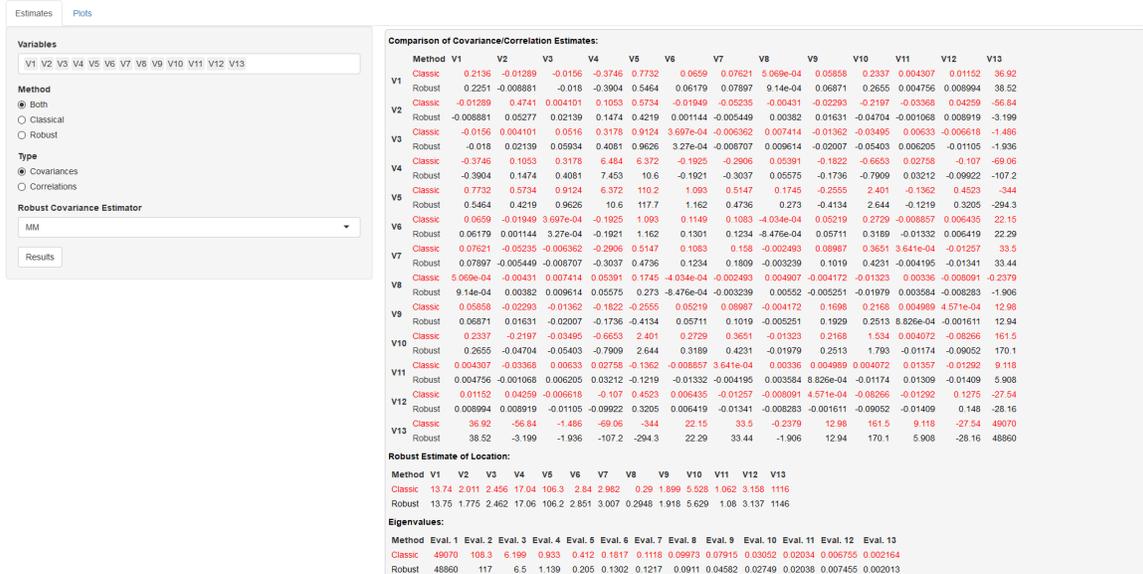


Figure 15: Classical and robust covariance estimates for selected variables in **wine** data

The above, along with figures 17, 18, and 19, correspond to Example 6.2 and Figure 6.3 in Maronna et al. You may find it of interest run the above example with the choice "Covariances" changed to "Correlations". A command-line equivalent of the Shiny UI interface in Figure 15 is shown below

```
# Load wine data
data(wine)

# Use fit.models to estimate centers and covariance matrices
cov.fm <- fit.models(Classic = covClassic(dat),
                     Robust = covRob(wine, type = "auto"))

# Comparison of results
print(summary(cov.fm))
```

Plots

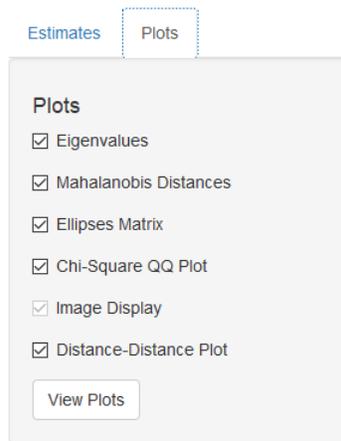


Figure 16: Covariance plotting options

The Shiny UI provides the following plotting options for **Robust Covariance**: *Eigenvalues*, *Mahalanobis Distances*, *Ellipses Matrix*, *Distances Chi-Squared QQ Plots*. Below are examples of each plot:

Mahalanobis Distances

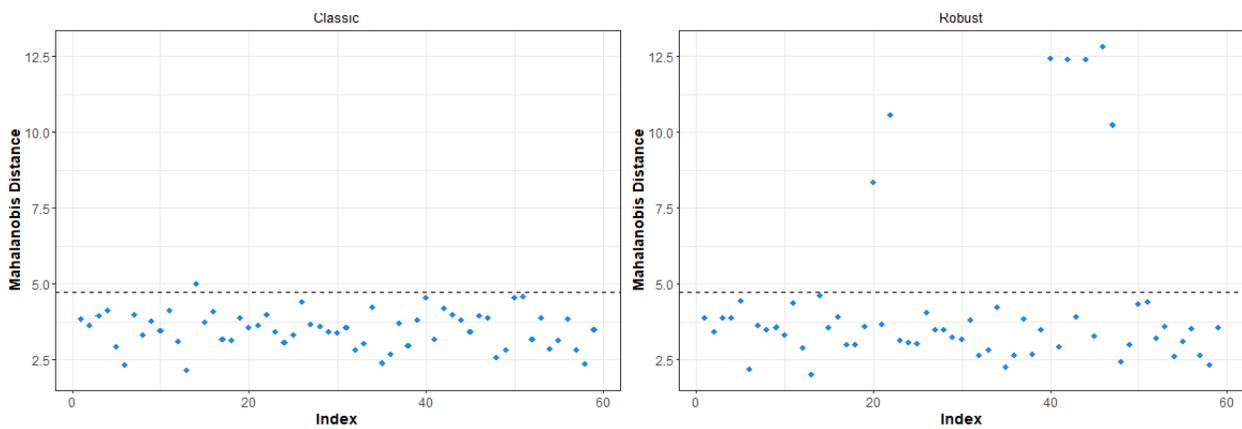


Figure 17: Square-root Mahalanobis distances for classical and robust covariance estimates: **wine**

Distances Chi-Squared QQ Plot

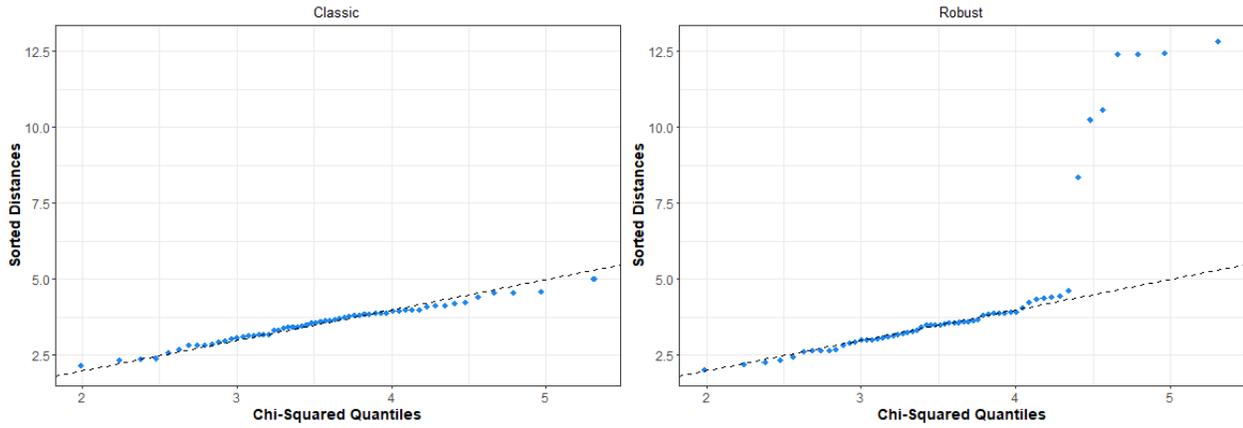


Figure 18: Classical and robust square-root Mahalanobis distances chi-squared QQ plots: **wine**

Distance-Distance scatterplot

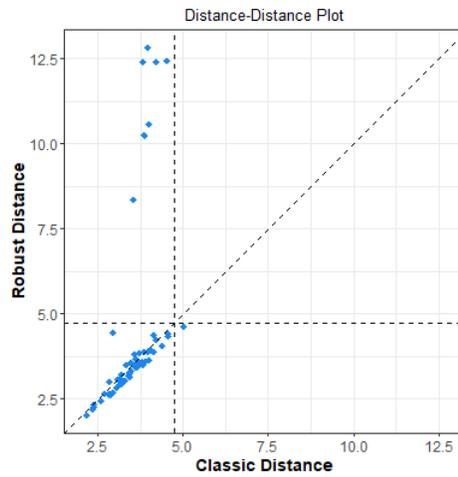


Figure 19: Scatterplot of square-root classical and robust Mahalanobis distances: **wine**

Figures 20 and 21 use the variables V2, V4, V8, V11, and V12 in the **wine** dataset.

Eigenvalues

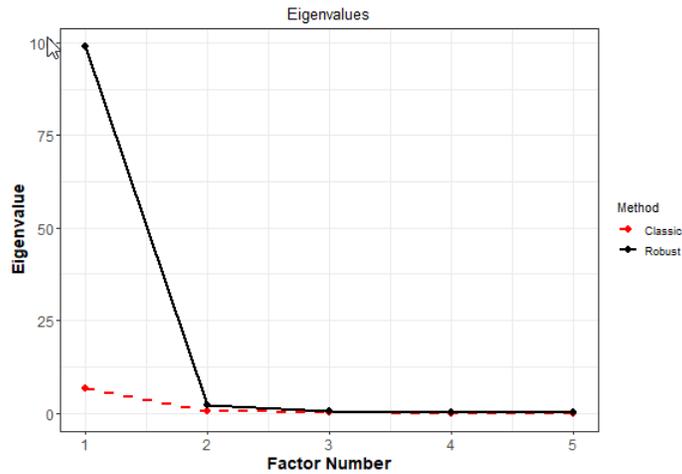


Figure 20: Scree plot for classical and robust covariance estimates: **wine**

Ellipses Matrix

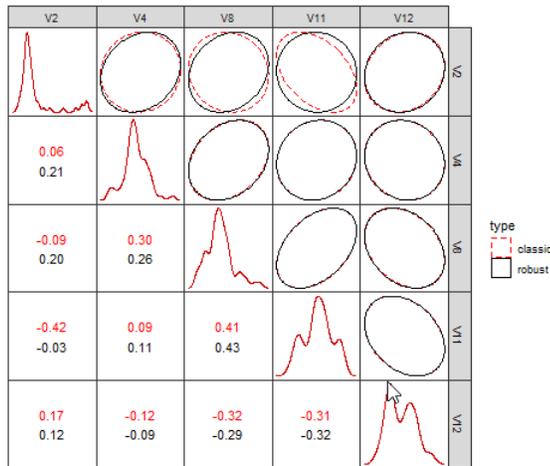


Figure 21: Ellipses matrices with classical and robust correlation estimates: **wine**. If you repeat the analysis with just three (or four) well-chosen variables, then you will get a 3 by 3 (or 4 by 4) display.

6 Robust PCA

This section allows a simple computation and comparison of classical and robust principal component analysis.

Computing Robust Principal Components

The options for computing principal components are similar to the options for computing covariance. Classical principal components are computed using `prcomp()` and robust principal components are computed using the `pcaRobS()`

function in **RobStatTM**. You must select at least three **Variables** for analysis. If the option “Both” is selected for **Method**, then the classical and robust principal components will be computed and compared. Figure 22 contains an example comparing classical and robust principal components of the **bus** data.

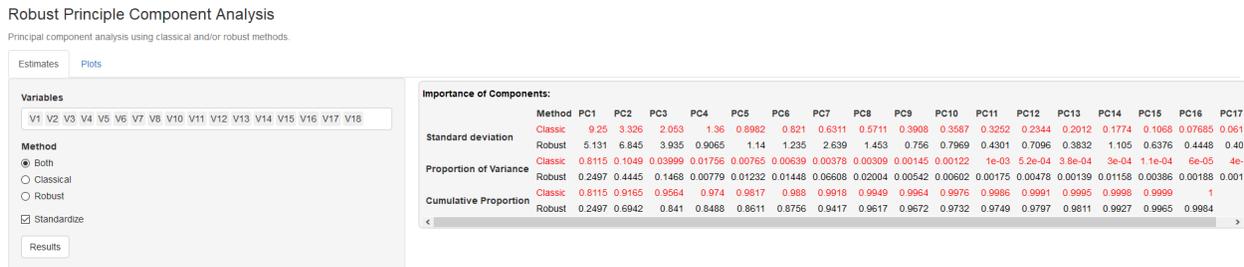


Figure 22: Classical and robust estimates for principal components for first ten variables in the **bus** data set

The command-line equivalent to the example from Figure 22 is:

```
# Load bus data
data(bus)

# Use fit.models to estimate principal components
pca.fm <- fit.models(Classic = prcomp(bus[, 1:10]),
                     Robust = prcompRob(bus[, 1:10]))

# Comparison of results
print(summary(pca.fm))
```

Plots

Plots for principal component analysis are not available in the current version, and we hope to add them in a future update.

7 What's Next?

Improvements to the User Interface

The Shiny UI will continue to evolve as we receive more feedback on the interface and consider new ideas for it. Future versions will generally contain small aesthetic adjustments and not a major overhaul of the current interface.

Additional Modeling Capability

Future versions will also contain additional robust modeling methods. Currently, we plan to add interfaces to robust logistic regression and robust time series analysis as discussed in Chapters 7 and 8 of the book.