 MVN package: Multivariate Normality Tests

Selcuk Korkmaz\textsuperscript{1} and Dincer Goksuluk

Hacettepe University, Faculty of Medicine, Department of Biostatistics, Ankara, TURKEY
\textsuperscript{1}selcuk.korkmaz@hacettepe.edu.tr

\texttt{MVN} version 3.1 (Last revision 2014-03-15)

Abstract

Assessing the assumption of multivariate normality is required by many parametric multivariate statistical methods, such as discriminant analysis, principal component analysis, MANOVA, etc. Here, we present an R package to asses multivariate normality. The \texttt{MVN} package contains three most widely used multivariate normality tests, including Mardia’s, Henze-Zirkler’s and Royston’s multivariate normality tests.

Contents

\begin{itemize}
\item[1] Preparation of input data \hfill 2
\item[2] Multivariate Normality Tests \hfill 3
  \begin{itemize}
  \item[2.1] Mardia’s Multivariate Normality Test \hfill 3
  \item[2.2] Henze-Zirkler’s Multivariate Normality Test \hfill 3
  \item[2.3] Royston’s Multivariate Normality Test \hfill 4
  \end{itemize}
\item[3] Multivariate Normality Plots \hfill 4
  \begin{itemize}
  \item[3.1] Q-Q Plot \hfill 5
  \item[3.2] Perspective and Contour Plots \hfill 6
  \end{itemize}
\item[4] Session info \hfill 8
\end{itemize}
1 Preparation of input data

MVN package expects a numeric matrix or a data frame that contains minimum two variables. In this vignette, we will work with the Iris data set. This data set is a multivariate data set introduced by Ronald A. Fisher (1936) as an application of discriminant analysis [1]. It is also called Anderson’s Iris data set because Edgar Anderson collected the data to measure the morphologic variation of Iris flowers of three related species [2]. The data set consists of 50 samples from each of three species of Iris including setosa, virginica and versicolor. For each sample, four variables were measured including the length and the width of the sepals and petals, in centimeters. We will check the multivariate normality of the Iris data set by using three multivariate normality tests, including Mardia’s, Royston’s and Henze-Zirkler’s multivariate normality tests.

First, we can call our data set using data function and display it using head function as follows:

```r
data(iris)
head(iris)
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1 5.1 3.5 1.4 0.2 setosa
## 2 4.9 3.0 1.4 0.2 setosa
## 3 4.7 3.2 1.3 0.2 setosa
## 4 4.6 3.1 1.5 0.2 setosa
## 5 5.0 3.6 1.4 0.2 setosa
## 6 5.4 3.9 1.7 0.4 setosa
```

The Iris data is in data.frame format which consists of 5 variables (Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, and Species) and 150 samples.

```r
class(iris)
```

```
## [1] "data.frame"
```

```r
dim(iris)
```

```
## [1] 150  5
```

For simplicity, we will work with a subset of the Iris data with first 50 samples without class label.

```r
Iris = iris[1:50, 1:4]
head(Iris)
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1 5.1 3.5 1.4 0.2
## 2 4.9 3.0 1.4 0.2
## 3 4.7 3.2 1.3 0.2
## 4 4.6 3.1 1.5 0.2
## 5 5.0 3.6 1.4 0.2
## 6 5.4 3.9 1.7 0.4
```
2 Multivariate Normality Tests

We will introduce three multivariate normality tests below, including Mardia’s, Henze-Zirkler’s and Royston’s Multivariate Normality Tests.

Before using our multivariate normality tests, we need to load our MVN package as follows:

```
library(MVN)
```

2.1 Mardia’s Multivariate Normality Test

Mardia’s test is based on multivariate extensions of skewness and kurtosis measures \[3\]. Now, we will check the multivariate normality of the Iris data using `mardiaTest` function in the MVN package. This function calculates the Mardia’s multivariate skewness and kurtosis coefficients as well as their corresponding statistical tests. For large sample size the multivariate skewness is asymptotically distributed as a chi-square random variable; here it is corrected for small sample size. Likewise, the multivariate kurtosis is distributed as a unit-normal \[4-6\].

```
result <- mardiaTest(Iris, cov = TRUE, qqplot = FALSE)
result
```

```
## Mardia's Multivariate Normality Test
## ---------------------------------------
## data : Iris
##
## g1p : 3.08
## skew : 25.66
## p.value.skew : 0.1772
##
## g2p : 26.54
## kurtosis : 1.295
## p.value.kurt : 0.1953
##
## small.skew : 27.86
## p.value.small : 0.1128
## ---------------------------------------
```


As seen above results, both skewness \((p = 0.1772)\) and kurtosis \((p = 0.1953)\) values indicate multivariate normality.

2.2 Henze-Zirkler’s Multivariate Normality Test

The Henze-Zirkler test is based on a non-negative functional distance that measures the distance between two distribution functions. If the data is multivariate normal, the test statistic is approximately log-normally distributed. It proceeds to calculate the mean, variance and smoothness parameter. Then, mean and variance are log-normalized and the p-value is estimated. We can
use \texttt{hzTest} function in the \texttt{MVN} package to calculate the Henze-Zirkler’s Multivariate Normality Test \cite{7,11}.

\begin{verbatim}
result <- hzTest(Iris, cov = TRUE, qqplot = FALSE)
result
## Henze-Zirkler's Multivariate Normality Test
## ---------------------------------------------
## data : Iris
## ## HZ : 0.9488
## ## p-value : 0.04995
## ---------------------------------------------
\end{verbatim}

Here, \texttt{HZ} is the value of Henze-Zirkler statistic at significance level 0.05 and \texttt{p-value} is a p-value for the Henze-Zirkler’s Multivariate Normality Test.

Since the p-value, which obtain from the \texttt{hzTest}, lower than 0.05, one can conclude that this multivariate data set deviates from multivariate normality.

### 2.3 Royston’s Multivariate Normality Test

Royston’s H test uses \textit{Shapiro-Wilk’s} \texttt{W} statistic for multivariate normality. However, if kurtosis of the data greater than 3 then \textit{Shapiro-Francia} test is used for leptokurtic samples else \textit{Shapiro-Wilk} test is used for platykurtic samples \cite{10,12,18}.

\begin{verbatim}
result <- roystonTest(Iris, qqplot = FALSE)
result
## Royston's Multivariate Normality Test
## ---------------------------------------------
## data : Iris
## ## H : 31.52
## ## p-value : 2.188e-06
## ---------------------------------------------
\end{verbatim}

Here, \texttt{H} is the value of Royston’s H statistic at significance level 0.05 and \texttt{p-value} is an approximate p-value for the test with respect to equivalent degrees of freedom (\texttt{edf}).

According to the Royston’s Multivariate Normality Test, the \textit{Iris} data set does not appear to follow a multivariate normal distribution (\textit{p} < 0.001).

### 3 Multivariate Normality Plots

Our \texttt{MVN} package has ability to create three multivariate plots. We can use \texttt{qqplot = TRUE} option in the \texttt{mardiaTest}, \texttt{hzTest} and \texttt{roystonTest} functions to create a chi-square Q-Q plot. Furthermore, we can use \texttt{mvnPlot} function in our \texttt{MVN} package to create perspective and contour plots for binary data sets.
3.1 Q-Q Plot

We can create a chi-square Q-Q plot for our Iris data set to see whether there is any deviation from multivariate normality.

```r
result <- roystonTest(Iris, qqplot = TRUE)
```

If the data set follows approximately a multivariate normal distribution, the resulting plot should be roughly straight line. As you can see from the chi-square Q-Q plot above, there are some
deviations from the straight line and this indicates possible departures from a multivariate normal distribution.

3.2 Perspective and Contour Plots

We can use the \texttt{mvnPlot} function in the \texttt{MVN} package to create a perspective plot for a binary data set. In order to get a perspective plot, we should continue with two variables, i.e., bivariate normal distribution. As an example, we subset first 50 rows and \textit{sepal} measures of \textit{Iris} data. Sepal measures of first 50 samples are bivariate normal. We can see that from the perspective plot. Perspective plot produces 3-dimensional bell-shaped graph when data is bivariate normal.

\begin{verbatim}
Iris = iris[1:50, 1:2]
result = hzTest(Iris)
mvnPlot(result, type = "persp", default = TRUE)
\end{verbatim}

Another alternative is to use 2-dimensional contour graphs. We can use the \texttt{mvnPlot} function in the \texttt{MVN} package to create a contour plot for a binary data set. Contour graphs are very useful since it gives information about normality and correlation at the same time. From contour graph
below, we can say that there is a positive correlation among sepal measures of flowers since contour lines lie around main diagonal.

```r
mvnPlot(result, type = "contour", default = TRUE)
```
4 Session info

sessionInfo()

## R version 3.0.3 (2014-03-06)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
## locale:
## [1] C/tr_TR.UTF-8/tr_TR.UTF-8/C/tr_TR.UTF-8/tr_TR.UTF-8
## attached base packages:
## [1] stats graphics grDevices utils datasets methods base
## other attached packages:
## [1] MVN_3.1 MASS_7.3-29 moments_0.13 nortest_1.0-2 knitr_1.5
## loaded via a namespace (and not attached):
## [1] codetools_0.2-8 digest_0.6.4 evaluate_0.5.1 formatR_0.10
## [5] highr_0.3 stringr_0.6.2 tools_3.0.3

References


