

# MVN package: Multivariate Normality Tests

Selcuk Korkmaz<sup>1</sup> and Dincer Goksuluk

Hacettepe University, Faculty of Medicine, Department of Biostatistics, Ankara, TURKEY

<sup>1</sup>`selcuk.korkmaz@hacettepe.edu.tr`

MVN version 3.7 (Last revision 2014-10-15)

## 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 assess multivariate normality. The MVN package contains three most widely used multivariate normality tests, including Mardia's, Henze-Zirkler's and Royston's multivariate normality tests.

## Contents

<b>1</b>	<b>Preparation of input data</b>	<b>2</b>
<b>2</b>	<b>Multivariate Normality Tests</b>	<b>3</b>
2.1	Mardia's Multivariate Normality Test . . . . .	3
2.2	Henze-Zirkler's Multivariate Normality Test . . . . .	4
2.3	Royston's Multivariate Normality Test . . . . .	5
<b>3</b>	<b>Multivariate Normality Plots</b>	<b>6</b>
3.1	Q-Q Plot . . . . .	6
3.2	Perspective and Contour Plots . . . . .	6
<b>4</b>	<b>Multivariate Outlier Detection</b>	<b>6</b>
4.1	Mahalanobis Distance . . . . .	6
4.2	Adjusted Mahalanobis Distance . . . . .	10
<b>5</b>	<b>Session info</b>	<b>11</b>

# 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:

```
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.

```
class(iris)

## [1] "data.frame"

dim(iris)

## [1] 150   5
```

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

```
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
##      chi.skew       : 25.66
##      p.value.skew   : 0.1772
##
##      g2p           : 26.54
##      z.kurtosis     : 1.295
##      p.value.kurt   : 0.1953
##
##      chi.small.skew : 27.86
##      p.value.small  : 0.1128
##
##      Result        : Data is multivariate normal.
## -----
```

Here:

`g1p`: Mardia's estimation of multivariate skewness,

`chi.skew`: Chi-square value of the skewness statistic,

`p.value.skew`: Significance value of skewness statistic,

`g2p`: Mardia's estimation of multivariate kurtosis,

`z.kurtosis`: z value of the kurtosis statistic,

`p.value.kurt`: Significance value of kurtosis statistic,

`chi.small.skew`: Chi-square value of the small sample skewness statistic,

`p.value.small`: Significance value of small sample skewness statistic.

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

`mardiaTest` function has an S4 class called `mardia`. We can use `getSlots` function in order to get the slots in this S4 class.

```
getSlots("mardia")

##           g1p      chi.skew  p.value.skew  chi.small.skew  p.value.small
##    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"
##           g2p      z.kurtosis  p.value.kurt          dname      dataframe
##    "numeric"    "numeric"    "numeric"    "character"    "data.frame"
```

To access the informations which are stored in these slots, we can use `@` operator as follow:

```
result@p.value.skew

## [1] 0.1772

result@p.value.kurt

## [1] 0.1953
```

## 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 `hzTest` function in the `MVN` package to calculate the Henze-Zirkler's Multivariate Normality Test [7–11].

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

##    Henze-Zirkler's Multivariate Normality Test
##    -----
##    data : Iris
##
##    HZ      : 0.9488
##    p-value : 0.04995
##
##    Result  : Data is not multivariate normal.
##    -----
```

Here, HZ is the value of Henze-Zirkler statistic at significance level 0.05 and `p-value` is a p-value for the Henze-Zirkler's Multivariate Normality Test.

Since the p-value, which optain from the `hzTest`, lower than 0.05, one can conclude that this multivariate data set deviates from multivariate normality.

`hzTest` function has an S4 class called `hz`. We can use `getSlots` function in order to get the slots in this S4 class.

```
getSlots("hz")

##           HZ           p.value           dname           dataframe
##    "numeric"    "numeric"    "character" "data.frame"
```

To access the informations which are stored in these slots, we can use @ operator as follow:

```
result@HZ

## [1] 0.9488

result@p.value

## [1] 0.04995
```

## 2.3 Royston's Multivariate Normality Test

Royston's H test uses Shapiro-Wilk's W statistic for multivariate normality. However, if kurtosis of the data greater than 3 then Shapiro-Francia test is used for leptokurtic samples else Shapiro-Wilk test is used for platykurtic samples [10,12–18].

```
result <- roystonTest(Iris, qqplot = FALSE)
result

##    Royston's Multivariate Normality Test
##    -----
##    data : Iris
##
##    H          : 31.52
##    p-value    : 2.188e-06
##
##    Result     : Data is not multivariate normal.
##    -----
```

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

According to the Royston's Multivariate Normality Test, the *Iris* data set does not appear to follow a multivariate normal distribution ( $p < 0.001$ ).

roystonTest function has an S4 class called royston. We can use getSlots function in order to get the slots in this S4 class.

```
getSlots("hz")

##           HZ           p.value           dname           dataframe
##    "numeric"    "numeric"    "character" "data.frame"
```

To access the informations which are stored in these slots, we can use @ operator as follow:

```
result@H
## [1] 31.52

result@p.value
## [1] 2.188e-06
```

### 3 Multivariate Normality Plots

Our MVN package has ability to create three multivariate plots. We can use `qqplot = TRUE` option in the `mardiaTest`, `hzTest` and `roystonTest` functions to create a chi-square Q-Q plot. Furthermore, we can use `mvnPlot` function in our 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.

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 `mvnPlot` function in the 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 *sepal* measures of *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.

Another alternative is to use 2-dimensional contour graphs. We can use the `mvnPlot` function in the 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 above, we can say that there is a positive correlation among *sepal* measures of flowers since contour lines lie around main diagonal.

## 4 Multivariate Outlier Detection

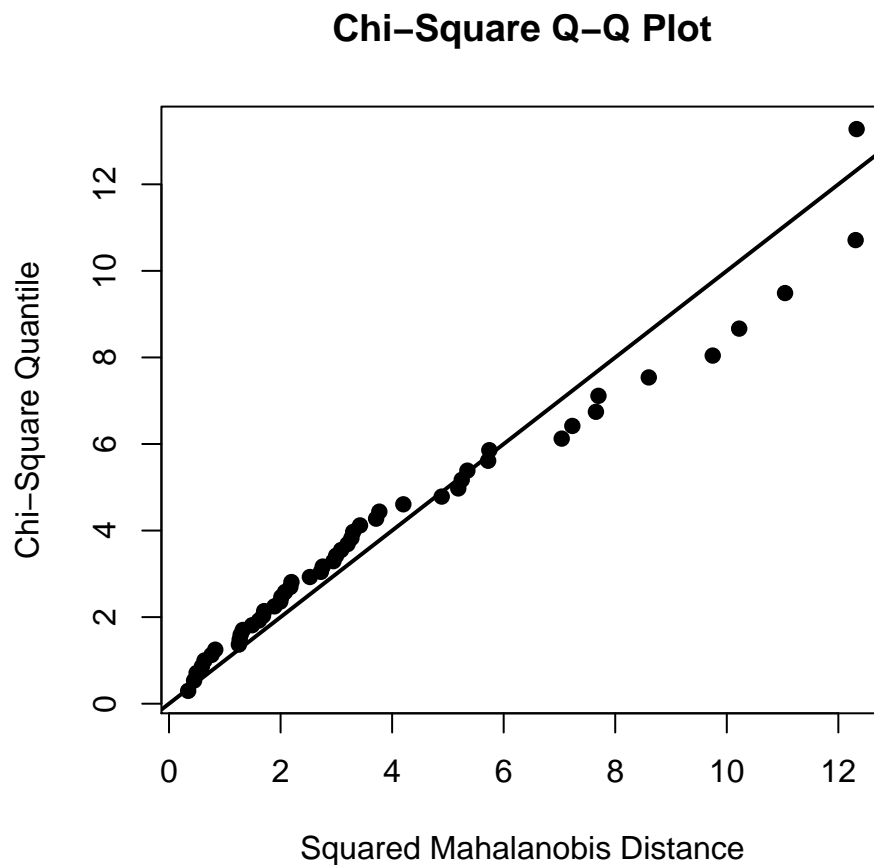
There are two multivariate outlier detection methods which are based on Mahalanobis distance in the MVN package.

### 4.1 Mahalanobis Distance

This methodology has following steps:

- 1 Compute robust Mahalanobis distances ( $MD(x_i)$ )
- 2 Compute the 97.5 percent-Quantile  $Q$  of the Chi-Square distribution

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



```
result

##   Royston's Multivariate Normality Test
##   -----
##   data : Iris
##
##   H       : 31.52
##   p-value : 2.188e-06
##
##   Result  : Data is not multivariate normal.
##   -----
```

3 Declare  $MD(x_i) > Q$  as possible outlier

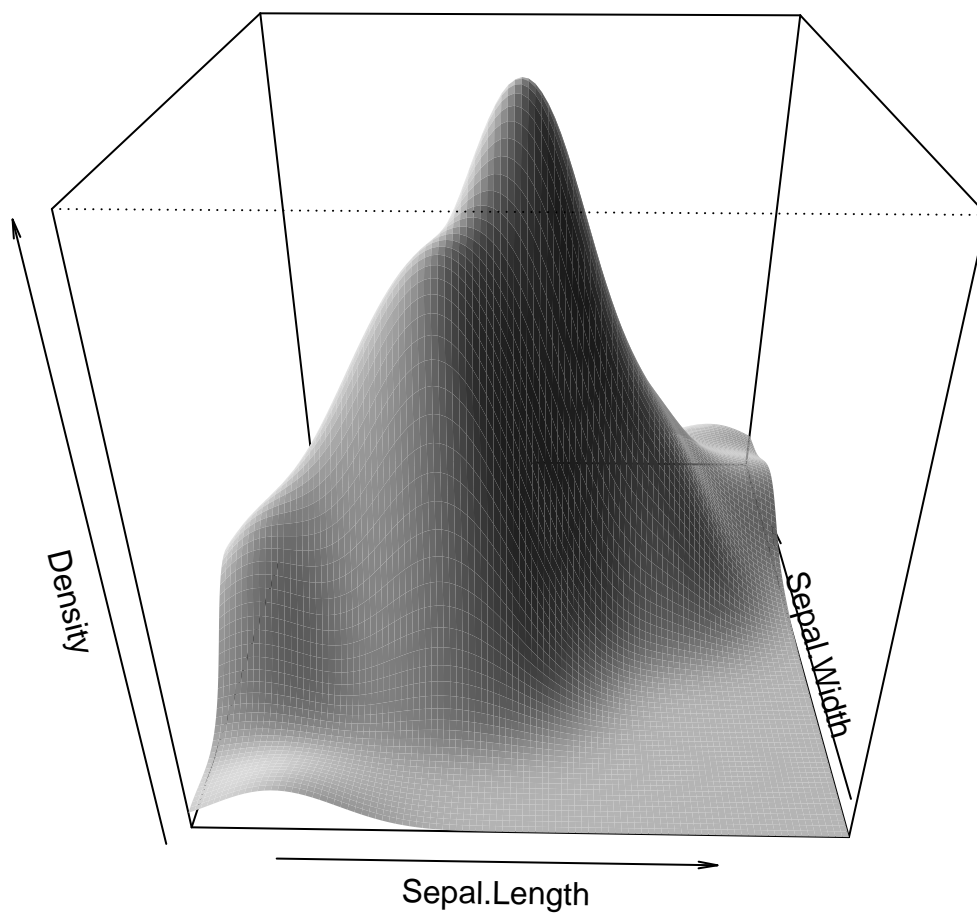
For this task `mvnPlot` function can be used as follows:

```
Iris = iris[1:50, 1:3]
result <- mvnPlot(Iris, qqplot = FALSE, method = "quan")
head(result$outlier)
```

```

Iris = iris[1:50, 1:2]
result = hzTest(Iris)
mvnPlot(result, type = "persp", default = TRUE)

```



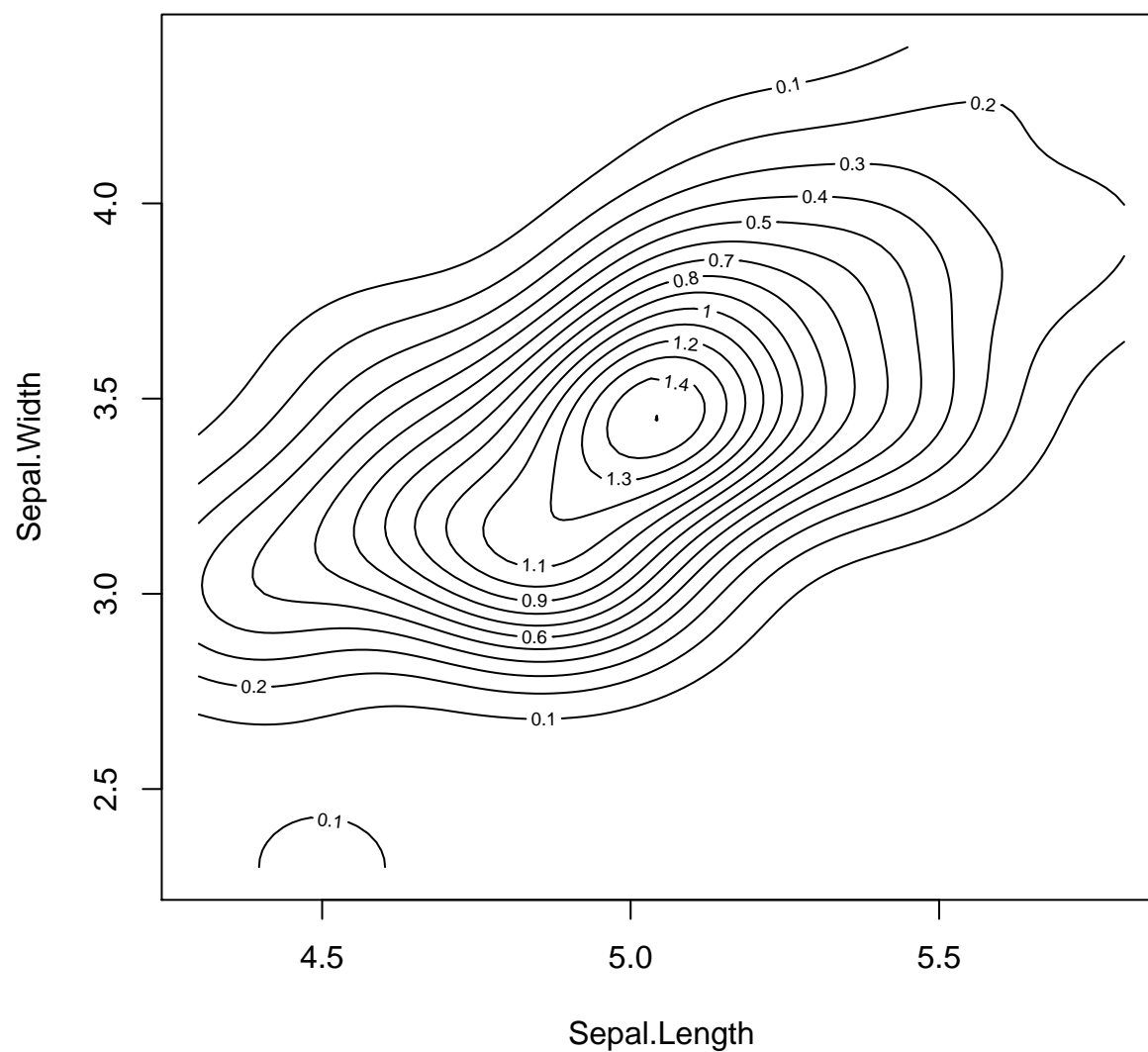
```

##      MD Outlier
## 23 14.961    TRUE
## 25 12.660    TRUE
## 15 11.890    TRUE
## 45 11.586    TRUE
## 14  9.081   FALSE
## 42  7.733   FALSE
head(result$newData)

```



```
mvnPlot(result, type = "contour", default = TRUE)
```



##	Sepal.Length	Sepal.Width	Petal.Length
## 1	5.1	3.5	1.4
## 10	4.9	3.1	1.5
## 11	5.4	3.7	1.5
## 12	4.8	3.4	1.6
## 13	4.8	3.0	1.4
## 14	4.3	3.0	1.1

Here, user can get outlier set based on Mahalanobis distance and data set without outliers.

## 4.2 Adjusted Mahalanobis Distance

This methodology has following steps:

- 1 Compute robust Mahalanobis distances ( $MD(x_i)$ )
- 2 Compute the 97.5 percent Adjusted Quantile (AQ) of the Chi-Square distribution
- 3 Declare  $MD(x_i) > AQ$  as possible outlier

Likewise, `mvnPlot` function can be used as follows:

```
result <- mvOutlier(Iris, qqplot = FALSE, method = "adj.quan")
head(result$outlier)

##           MD Outlier
## 23 14.961    TRUE
## 25 12.660    TRUE
## 15 11.890    TRUE
## 45 11.586    TRUE
## 14  9.081   FALSE
## 42  7.733   FALSE

head(result$newData)

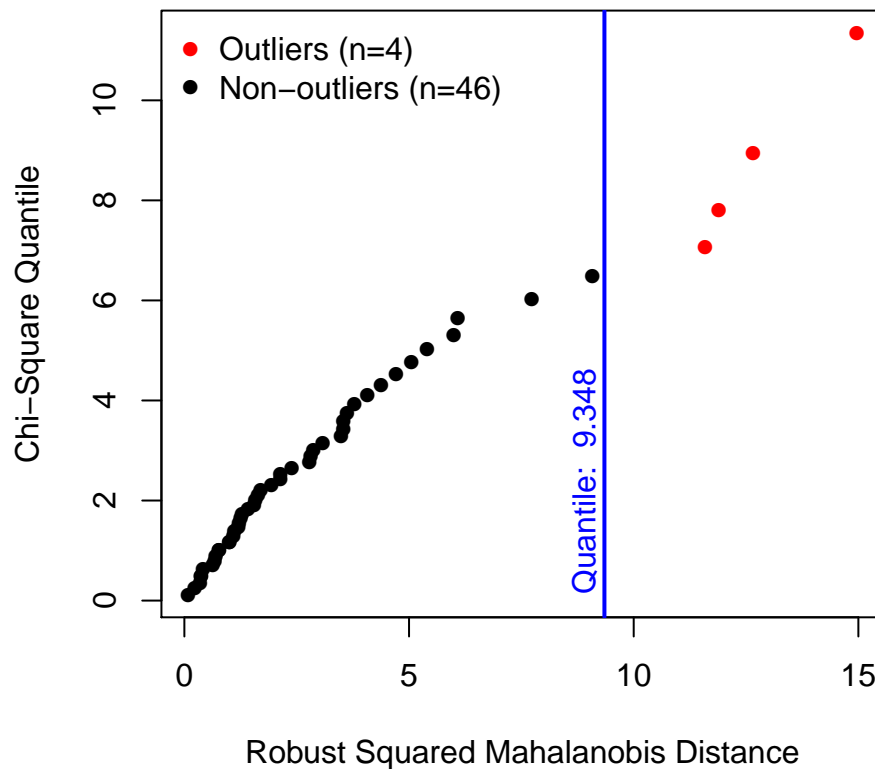
##      Sepal.Length Sepal.Width Petal.Length
## 1           5.1         3.5         1.4
## 10          4.9         3.1         1.5
## 11          5.4         3.7         1.5
## 12          4.8         3.4         1.6
## 13          4.8         3.0         1.4
## 14          4.3         3.0         1.1
```

Here, user can get outlier set based on adjusted Mahalanobis distance and data set without outliers.

A Q-Q plot can be created with using `qqplot = TRUE` option in the `mvOutlier` function for visual inspection.

```
result <- mvOutlier(Iris, qqplot = TRUE, method = "adj.quan")
```

### Adjusted Chi-Square Q-Q Plot



## 5 Session info

```
sessionInfo()

## R version 3.0.2 (2013-09-25)
## Platform: i386-w64-mingw32/i386 (32-bit)
##
## locale:
## [1] LC_COLLATE=C
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
```

```
## [1] MVN_3.7    knitr_1.5
##
## loaded via a namespace (and not attached):
## [1] DEoptimR_1.0-2      GGally_0.4.8        MASS_7.3-29
## [4] RColorBrewer_1.0-5  Rcpp_0.11.0         codetools_0.2-8
## [7] colorspace_1.2-4    dichromat_2.0-0     digest_0.6.4
## [10] evaluate_0.5.1      formatR_0.10        ggplot2_1.0.0
## [13] grid_3.0.2          gtable_0.1.2        highr_0.3
## [16] labeling_0.2        moments_0.13        munsell_0.4.2
## [19] mvoutlier_2.0.5     mvtnorm_0.9-9997    nortest_1.0-2
## [22] pcaPP_1.9-60        pls_2.4-3           plyr_1.8.1
## [25] proto_0.3-10        reshape_0.8.5       reshape2_1.2.2
## [28] robCompositions_1.9.0 robustbase_0.91-1   rrcov_1.3-4
## [31] scales_0.2.3        sgeostat_1.0-25     stats4_3.0.2
## [34] stringr_0.6.2       tools_3.0.2
```

## References

- [1] R. A. Fisher (1936). "The use of multiple measurements in taxonomic problems". *Annals of Eugenics* 7 (2): 179-188. doi:10.1111/j.1469-1809.1936.tb02137.x
- [2] Edgar Anderson (1936). "The species problem in Iris". *Annals of the Missouri Botanical Garden* 23 (3): 457-509. JSTOR 2394164.
- [3] Mardia, K. V. (1970). "Measures of multivariate skewness and kurtosis with applications". *Biometrika* 57 (3): 519-530. doi:10.1093/biomet/57.3.519.
- [4] Mardia, K. V. (1974), Applications of some measures of multivariate skewness and kurtosis for testing normality and robustness studies. *Sankhy A*, 36:115-128.
- [5] Trujillo-Ortiz, A. and R. Hernandez-Walls. (2003). Mskekur: Mardia's multivariate skewness and kurtosis coefficients and its hypotheses testing. A MATLAB file. URL <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=3519>
- [6] Stevens, J. (1992), *Applied Multivariate Statistics for Social Sciences*. 2nd. ed. New-Jersey:Lawrance Erlbaum Associates Publishers. pp. 247-248.
- [7] Trujillo-Ortiz, A., R. Hernandez-Walls, K. Barba-Rojó and L. Cupul-Magana. (2007). HZmvntest:Henze-Zirkler's Multivariate Normality Test. A MATLAB file. URL <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=17931>
- [8] Henze, N. and Zirkler, B. (1990), A Class of Invariant Consistent Tests for Multivariate Normality. *Commun. Statist.-Theor. Meth.*, 19(10): 35953618.
- [9] Henze, N. and Wagner, Th. (1997), A New Approach to the BHEP tests for multivariate normality. *Journal of Multivariate Analysis*, 62:1-23.
- [10] Johnson, R.A. and Wichern, D. W. (1992). *Applied Multivariate Statistical Analysis*. 3rd. ed. New-Jersey:Prentice Hall.

- [11] Mecklin, C. J. and Mundfrom, D. J. (2003), On Using Asymptotic Critical Values in Testing for Multivariate Normality.
- [12] Mecklin, C.J. and Mundfrom, D.J. (2005). A Monte Carlo comparison of the Type I and Type II error rates of tests of multivariate normality. *Journal of Statistical Computation and Simulation*, 75:93-107.
- [13] Royston, J.P. (1982). An Extension of Shapiro and Wilks W Test for Normality to Large Samples. *Applied Statistics*, 31(2):115-124.
- [14] Royston, J.P. (1983). Some Techniques for Assessing Multivariate Normality Based on the Shapiro-Wilk W. *Applied Statistics*, 32(2).
- [15] Royston, J.P. (1992). Approximating the Shapiro-Wilk W-Test for non-normality. *Statistics and Computing*, 2:117-119.121-133.
- [16] Royston, J.P. (1995). Remark AS R94: A remark on Algorithm AS 181: The W test for normality. *Applied Statistics*, 44:547-551.
- [17] Shapiro, S. and Wilk, M. (1965). An analysis of variance test for normality. *Biometrika*, 52:591-611.
- [18] Trujillo-Ortiz, A., R. Hernandez-Walls, K. Barba-Rojo and L. Cupul-Magana. (2007). Roystest:Royston's Multivariate Normality Test. A MATLAB file. URL <http://www.mathworks.com/matlabcentral/fileexchange/17811>