Article

MaxSkew and MultiSkew: Two \( R \) Packages for Detecting, Measuring and Removing Multivariate Skewness

Cinzia Franceschini \(^{1,\dagger} \) and Nicola Loperfido \(^{2,*,\dagger} \)

\(^{1}\) Dipartimento di Scienze Agrarie e Forestali (DAFNE), Università degli Studi della Tuscia, Via San Camillo de Lellis snc, 01100 Viterbo, Italy
\(^{2}\) Dipartimento di Economia, Società e Politica (DESP), Università degli Studi di Urbino “Carlo Bo”, Via Saffi 42, 61029 Urbino, Italy
* Correspondence: nicola.loperfido@uniurb.it
\( \dagger \) These authors contributed equally to this work.

Received: 6 July 2019; Accepted: 22 July 2019; Published: 1 August 2019

Abstract: The R packages MaxSkew and MultiSkew measure, test and remove skewness from multivariate data using their third-order standardized moments. Skewness is measured by scalar functions of the third standardized moment matrix. Skewness is tested with either the bootstrap or under normality. Skewness is removed by appropriate linear projections. The packages might be used to recover data features, as for example clusters and outliers. They are also helpful in improving the performances of statistical methods, as for example the Hotelling’s one-sample test. The Iris dataset illustrates the usages of MaxSkew and MultiSkew.

Keywords: asymmetry; bootstrap; projection pursuit; symmetrization; third cumulant

1. Introduction

The skewness of a random variable \( X \) satisfying \( E \left( |X|^3 \right) < +\infty \) is often measured by its third standardized cumulant

\[
\gamma_1 (X) = \frac{E \left[ (X - \mu)^3 \right]}{\sigma^3},
\]

where \( \mu \) and \( \sigma \) are the mean and the standard deviation of \( X \). The squared third standardized cumulant \( \beta_1 (X) = \gamma_1^2 (X) \), known as Pearson’s skewness, is also used. The numerator of \( \gamma_1 (X) \), that is

\[
\kappa_3 (X) = E \left[ (X - \mu)^3 \right],
\]

is the third cumulant (i.e., the third central moment) of \( X \). Similarly, the third moment (cumulant) of a random vector is a matrix containing all moments (cumulants) of order three which can be obtained from the random vector itself.

Statistical applications of the third moment include, but are not limited to: factor analysis ([1,2]), density approximation ([3–5]), independent component analysis ([6]), financial econometrics ([7,8]), cluster analysis ([9–12]), Edgeworth expansions ([13], page 189), portfolio theory ([14]), linear models ([15,16]), likelihood inference ([17]), projection pursuit ([18]), time series ([7,19]), spatial statistics ([20–22]) and nonrandom sampling ([23]).

A \( d \)-dimensional random vector might have up to \( d (d + 1) (d + 2) / 6 \) distinct third-order moments (cumulants). It is therefore convenient to summarize its skewness with a scalar function of its third-order standardized moments. Default choices are Mardia’s skewness ([24]),
partial skewness ([25–27]) and directional skewness ([28]). These measures are similar in several ways (see, for example, [29]). First, they are invariant with respect to one-to-one affine transformations. Second, they coincide with Pearson’s skewness for univariate distributions. Third, their properties have been mainly investigated within the framework of normality testing. Skewness might hamper the performance of several multivariate statistical methods, as for example the Hotelling’s one-sample test ([24,30,31]).

The default approach to symmetrization relies on power transformations. Unfortunately, the transformed variables might not allow for meaningful interpretation and might not follow a multivariate normal distribution. On top of that, they are neither affinely invariant nor robust ([32,33]). Refs. [4,34] addressed these problems with symmetrizing linear transformations.

The R packages MaxSkew and MultiSkew provide an unified treatment of multivariate skewness by detecting, measuring and alleviating skewness from multivariate data. Data visualization and significance levels are used to assess the hypothesis of symmetry. Default measures of multivariate skewness depend on third-order moments of the standardized distribution. Linear projections might remove or at least alleviate multivariate skewness. To the best of our knowledge, no statistical packages compute their bootstrap estimates, the third cumulant and linear projections alleviating skewness. MaxSkew and MultiSkew were released in 2017. Since then, there has been a lot of research on flexible multivariate distributions purported to model nonnormal features such as skewness. Their third cumulants do not necessarily have simple analytical expressions and might need to be investigated by means of simulations studies based on MaxSkew and MultiSkew. The remainder of the paper is organized as follows. Section 2 reviews the basic concepts of multivariate skewness within the frameworks of third moments and projection pursuit. It also describes some skewness-related features of the Iris dataset. Section 3 illustrates the package MaxSkew. Section 4 describes the functions of MultiSkew related to symmetrization, third moments and skewness measures. Section 5 contains some concluding remarks and hints for improving the packages.

2. Third Moment

The third multivariate moment, that is the third moment of a random vector, naturally generalizes to the multivariate case the third moment $E (X^3)$ of a random variable $X$ whose third absolute moment is finite. It is defined as follows, for a $d$-dimensional random vector $x = (X_1, \ldots, X_d)^\top$ satisfying $E (|X_i^3|) < + \varepsilon$, for $i = 1, \ldots, d$. The third moment of $x$ is the $d^2 \times d$ matrix

$$
M_{3x} = E \left( x \otimes x^\top \otimes x \right),
$$

where “$\otimes$” denotes the Kronecker product (see, for example, [29,35]). In the following, when referring to the third moment of a random vector, we shall implicitly assume that all appropriate moments exist.

The third moment $M_{3x}$ of $x = (X_1, \ldots, X_d)^\top$ contains $d^3$ elements of the form $\mu_{ijh} = E (X_iX_jX_h)$, where $i, j, h = 1, \ldots, d$. Many elements are equal to each other, due to the identities

$$
\mu_{ijh} = \mu_{ijh} = \mu_{ijh} = \mu_{ijh} = \mu_{ijh}.
$$

First, there are at most $d$ distinct elements $\mu_{ijh}$ where the three indices are equal to each other: $\mu_{i} = E (X_i^3)$, for $i = 1, \ldots, d$. Second, there are at most $d(d-1)$ distinct elements $\mu_{ijh}$ where only two indices are equal to each other: $\mu_{ij} = E (X_i^2X_j)$, for $i, j = 1, \ldots, d$ and $i \neq j$. Third, there are at most $d(d-1)(d-2)/6$ distinct elements $\mu_{ijh}$ where the three indices differ from each other: $\mu_{ijh} = E (X_iX_jX_h)$, for $i, j, h = 1, \ldots, d$ and $i \neq j \neq h$. Hence $M_{3x}$ contains at most $d(d+1)(d+2)/6$ distinct elements.

Invariance of $\mu_{ijh} = E (X_iX_jX_h)$ with respect to indices permutations implies several symmetries in the structure of $M_{3x}$. First, $M_{3x}$ might be regarded as $d$ matrices $B_i = E (X_iX_i^\top)$, $i = 1, \ldots, n$ stacked on top of each other. Hence $\mu_{ijh}$ is the element in the $j$-th row and in the $h$-th column of the
The third standardized moment (or cumulant) of the random vector $x$ is the third moment of $x$, $K_{3,x} = E [(x - \mu) \otimes (x - \mu) \otimes (x - \mu)]$. (5)

It is related to the third moment via the identity

$$K_{3,x} = M_{3,x} - M_{2,x} \otimes \mu - \mu \otimes M_{2,x} - vec(M_{2,x}) \mu^\top + 2\mu \otimes \mu^\top \otimes \mu.$$ (6)

The third cumulant allows for a better assessment of skewness by removing the effect of location on third-order moments. It becomes a null matrix under central symmetry, that is when $x - \mu$ and $\mu - x$ are identically distributed.

The third standardized moment (or cumulant) of the random vector $x$ is the third moment of $z = (Z_1, \ldots, Z_d)^\top = \Sigma^{-1/2} (x - \mu)$, where $\Sigma^{-1/2}$ is the inverse of the positive definite square root $\Sigma^{1/2}$ of $\Sigma = \text{cov}(x)$, which is assumed to be positive definite:

$$\Sigma^{1/2} = \left(\Sigma^{1/2}\right)^\top, \Sigma^{1/2} > 0 \text{ and } \Sigma^{1/2} \Sigma^{1/2} = \Sigma.$$ (7)

It is often denoted by $K_{3,z}$ and is related to $K_{3,x}$ via the identity

$$K_{3,z} = \left(\Sigma^{-1/2} \otimes \Sigma^{-1/2}\right) K_{3,x} \Sigma^{-1/2}.$$ (8)

The third standardized cumulant is particularly useful for removing the effects of location, scale and correlations on third order moments. The mean and the variance of $z$ are invariant with respect to orthogonal transformations, but the same does not hold for third moments: $M_{3,z}$ and $M_{3,w}$ may differ, if $w = Uz$ and $U$ is a $d \times d$ orthogonal matrix.

Projection pursuit is a multivariate statistical technique aimed at finding interesting low-dimensional data projections. It looks for the data projections which maximize the projection pursuit index, that is a measure of interestingness. \cite{18} reviews the merits of skewness (i.e., the third standardized cumulant) as a projection pursuit index. Skewness-based projection pursuit is based on the multivariate skewness measure in \cite{28}. They defined the directional skewness of a random vector $x$ as the maximum value $\beta^D_{1,d}(x)$ attainable by $\beta_1(c^\top x)$, where $c$ is a nonnull, $d$-dimensional real vector and $\beta_1(Y)$ is the Pearson’s skewness of the random variable $Y$:

$$\beta^D_{1,d}(x) = \max_{c \in S^{d-1}} \frac{E^2 \left( (c^\top x - c^\top \mu)^3 \right)}{(c^\top \Sigma c)^3},$$ (9)
with \( S^{d-1} \) being the set of \( d \)-dimensional real vectors of unit length. The name directional skewness reminds that \( \beta_{1,d}^M(x) \) is the maximum skewness attainable by a projection of the random vector \( x \) onto a direction. It admits a simple representation in terms of the third standardized cumulant:

\[
\max_{e \in S^{d-1}} \left( (c^T \otimes c^T) K_{3,x} e \right)^2 \leq \max_{e \in S^{d-1}} \sum_{f,g,i,j,h,k} c_f c_g c_i c_j c_k f g i j h k.
\]  

(10)

Statistical applications of directional skewness include normality testing ([28]), point estimation ([36]), independent component analysis ([6,29]) and cluster analysis ([4,9–12,18]).

There is a general consensus that an interesting feature, once found, should be removed ([37,38]).

The appropriateness of the linear transformation purported to remove or alleviate symmetry might be assessed with measures of multivariate skewness, which should be significantly smaller in the transformed data than in the original ones. Ref. [24] summarized the multivariate skewness of the random vector \( x \) with the scalar measure

\[
\beta_{1,d}^M(x) = E \left\{ \left( (x - \mu)^T \Sigma^{-1} (y - \mu) \right)^3 \right\},
\]  

(11)

where \( x \) and \( y \) are two \( d \)-dimensional, independent and identically distributed random vectors with mean \( \mu \) and covariance \( \Sigma \). It might be represented as the squared norm of the third standardized cumulant:

\[
\beta_{1,d}^M(x) = tr \left( K_{3,x} K_{3,x} \right) = \sum_{i,j,k} \kappa_{ijk}^2.
\]  

(12)

It is invariant with respect to one-to-one affine transformations:

\[
\beta_{1,d}^M(x) = \beta_{1,d}^M(Ax + b), \text{ where } b \in \mathbb{R}^d, A \in \mathbb{R}^{d \times d}, \det(A) \neq 0.
\]  

(13)

Mardia’s skewness is by far the most popular measure of multivariate skewness. Its statistical applications include multivariate normality testing ([24]) and the robustness of MANOVA statistics ([25]).

Another scalar measure of multivariate skewness is

\[
\beta_{1,d}^P(x) = E \left\{ (x - \mu)^T \Sigma^{-1} (x - \mu)(x - \mu)^T \Sigma^{-1} (y - \mu)(y - \mu)^T \Sigma^{-1} (y - \mu) \right\},
\]  

(14)

where \( x \) and \( y \) are the same as above. It has been independently proposed by several authors ([25–27]). Ref. [29] named it partial skewness to remind that \( \beta_{1,d}^P(x) \) does not depend on moments of the form \( E(Z_i Z_j Z_k) \) when \( i, j, k \) differ from each other, as it becomes apparent when representing it as a function of the third standardized cumulant:

\[
\beta_{1,d}^P(x) = vec^T(I_d) K_{3,z} vec(I_d) = \sum_{i,j,k} \kappa_{ijk} K_{hkk}.
\]  

(15)

Partial skewness is by far less popular than Mardia’s skewness. Like the latter measure, however, it has been applied to multivariate normality testing ([39–41]) and for assessing the robustness of MANOVA statistics ([25]).

Ref. [27] proposed to measure the skewness of the \( d \)-dimensional random vector \( x \) with the vector \( \gamma_{1,d}(x) = E(zz), \) where \( z \) is the standardized version of \( x \). This vector-valued measure of skewness might be regarded as weighted average of the standardized vector \( z \), with more weight
placed on the outcomes furthest away from the sample mean. It is location invariant, admits the representation $\gamma_1(x) = \mathbf{K}_{1\times d} \text{vec}(\mathbf{L}_d)$ and its squared norm is the partial skewness of $x$. It coincides with the third standardized cumulant of a random variable in the univariate case, and with the null $d$-dimensional vector when the underlying distribution is centrally symmetric. [11] applied it to model-based clustering.

We shall illustrate the role of skewness with the Iris dataset contained in the R package datasets: iris(datasets). We shall first download the data: R> data(iris). Then use the help command R> help(iris) and the structure command R> str(iris) to obtain the following information about this dataset. It contains the measurements in centimeters of four variables on 150 iris flowers: sepal length, sepal width, petal length and petal width. There is also a factor variable (Species) with three levels: setosa, versicolor and virginica. There are 50 rows for each species. The output of the previous code shows that the dataset is a data frame object containing 150 units and five variables:

```
'data.frame': 150 obs. of 5 variables:
$ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 ... 
$ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 ... 
$ Petal.Length : num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 ... 
$ Petal.Width : num 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.2 ... 
$ Species : Factor w/ 3 levels "setosa","versicolor",...: 1 1 ... 
```

With the command

R> pairs(iris[,1:4],col=c("red","green","blue")[as.numeric(Species)]) we obtain the multiple scatterplot of the Iris dataset (Figure 1).
The groups “versicolor” and “virginica” are quite close to each other and well separated from “setosa”. The data are markedly skewed, but skewness reduces within each group, as exemplified by the histograms of petal length in all flowers (Figure 2a) and in those belonging to the “setosa” group (Figure 2b). Both facts motivated researchers to model the dataset with mixtures of three normal components (see, for example [42], Section 6.4.3). However, [43] showed that the normality hypothesis should be rejected at the 0.05 level in the “setosa” group, while nonnormality is undetected by Mardia’s skewness.

![Histogram of iris[, 3]](image1)

![Histogram of iris[1:50, 3]](image2)

Figure 2. (a) Histogram of petal length in the Iris dataset, (b) Histogram of petal length in the “setosa” group.

We shall use MaxSkew and MultiSkew to answer the following questions about the Iris dataset.

1. Is skewness really inept at detecting nonnormality of the variables in the “setosa” group?
2. Does skewness help in recovering the cluster structure?
3. Can skewness be removed via meaningful linear projections?

The above questions will be addressed within the frameworks of projection pursuit, normality testing, cluster analysis and data exploration.

### 3. MaxSkew

The package MaxSkew [44] is written in the R programming language [45]. It is available for download on the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/package=MaxSkew.

The package MaxSkew uses several R functions. The first one is eigen \{base\} [45]. It computes eigenvalues and eigenvectors of real or complex matrices. Its usage, as described by the command help(eigen) is

R> eigen(x, symmetric, only.values = FALSE, EISPACK = FALSE).

The output of the function are values (a vector containing the eigenvalues of x, sorted in decreasing order according to their modulus) and vectors (either a matrix whose columns contain the normalized eigenvectors of x, or a null matrix if only.values is set equal to TRUE). A second R function used in MaxSkew is polyroot \{base\} [45] which finds the zeros of a real or complex polynomial. Its usage is polyroot(z), where z is the vector of polynomial coefficients arranged in increasing order. The Value in output is a complex vector of length n - 1, where n is the position of the largest non-zero element of z. MaxSkew also uses the R function svd \{base\} [45] which computes the singular value decomposition of a matrix. Its usage is...
The last R function used in \texttt{MaxSkew} is \texttt{kronecker} \cite{matrix} which computes the Kronecker product of two arrays, \texttt{X} and \texttt{Y}. The Value in output is an array with dimensions $\dim(X) \times \dim(Y)$.

The \texttt{MaxSkew} package finds orthogonal data projections with maximal skewness. The first data projection in the output is the most skewed among all data projections. The second data projection in the output is the most skewed among all data projections orthogonal to the first one, and so on. Ref. \cite{maxskew} motivates this method within the framework of model-based clustering. The package implements the algorithm described in \cite{maxskew} and is downloaded with the command \texttt{R> install.packages("MaxSkew")}. The package is attached with the command \texttt{R> library(MaxSkew)}.

The packages \texttt{MaxSkew} and \texttt{MultiSkew} require the dataset to be a data matrix object, so we transform the \texttt{iris} data frame accordingly: \texttt{R> iris.m<-data.matrix(iris)}. We check that we have a data matrix object with the command \texttt{R> str(iris.m)}:

\begin{verbatim}
R> str(iris.m)
num [1:150, 1:5] 5.1 4.9 4.7 4.6 5 5.4 ...
   - attr(*, "dimnames")= List of 2
     ..$ : NULL
\end{verbatim}

The help command shows the basic informations about the package and the functions it contains: \texttt{R> help(MaxSkew)}.

The package \texttt{MaxSkew} has three functions, two of which are internal. The usage of the main function is

\begin{verbatim}
R> MaxSkew(data, iterations, components, plot),
\end{verbatim}

where \texttt{data} is a data matrix object, \texttt{iterations} (the number of required iterations) is a positive integer, \texttt{components} (the number of orthogonal projections maximizing skewness) is a positive integer smaller than the number of variables, and \texttt{plot} is a dichotomous variable: TRUE/FALSE. If \texttt{plot} is set equal to TRUE (FALSE) the scatterplot of the projections maximizing skewness appears (does not appear) in the output. The output includes a matrix of projected data, whose $i$-th row represents the $i$-th unit, while the $j$-th column represents the $j$-th projection. The output also includes the multiple scatterplot of the projections maximizing skewness. As an example, we call the function

\begin{verbatim}
R> MaxSkew(iris.m[,1:4], 50, 2, TRUE).
\end{verbatim}

We have used only the first four columns of the \texttt{iris.m} data matrix object, because the last column is a label. As a result, we obtain a matrix with 150 rows and 2 columns containing the projected data and a multiple scatterplot. The structure of the resulting matrix is

\begin{verbatim}
R> str(MaxSkew(iris.m[,1:4], 50, 2, TRUE))
\end{verbatim}

For the sake of brevity we show only the first three rows of the projected data matrix:

\begin{verbatim}
R> iris.projections <-MaxSkew(iris.m[,1:4], 50,2,TRUE)
R> iris.projections[1:3,]
   [,1]        [,2]
[1,]  -2.631244186  -0.817635353
[2,]  -2.620071890  -1.033692782
[3,]  -2.376652037  -1.311616693
\end{verbatim}

Figure 3a shows the scatterplot of the projected data. The connections between skewness-based projection pursuit and cluster analysis, as implemented in \texttt{MaxSkew}, have been investigated by several authors \cite{maxskew, multi, projpursuit, cluster, clustering, projection}. For the Iris dataset, it is well illustrated by the scatterplot of the two most skewed, mutually orthogonal projections, with different colors to denote the group memberships (Figure 3b). The plot is obtained with the commands
R> attach(iris)
R> iris.projections <- MaxSkew(iris.m[,1:4], 50,2,TRUE)
R> iris.projections.species <- cbind(iris.projections, iris$Species)
R> pairs(iris.projections.species[,1:2], col=c("red","green","blue") [as.numeric(Species)])
R> detach(iris)

![Projections](image1)

**Figure 3.** (a) Multiple scatterplot of the first two most skewed, mutually orthogonal projections, (b) Scatterplot of the first two most skewed, mutually orthogonal projections of Iris data, with different colors to denote different group memberships.

The scatterplot clearly shows the separation of “setosa” from “virginica” and “versicolor”, whose overlapping is much less marked than in the original variables. The scatterplot is very similar, up to rotation and scaling, to those obtained from the same data by [46,47].

Mardia’s skewness is unable to detect nonnormality in the “setosa” group, thus raising the question of whether any skewness measure is apt at detecting such nonnormality ([43]). We address this question using skewness-based projection pursuit as a visualization tool. Figure 4 contains the scatterplot of the two most skewed, mutually orthogonal projections obtained from the four variables recorded from setosa flowers only. It clearly shows the presence of a dense, elongated cluster, which is inconsistent with the normality assumption. Formal testing for excess skewness in the “setosa” group will be discussed in the following sections.

![Projections](image2)

**Figure 4.** Scatterplot of the two most skewed, mutually orthogonal projections computed from data in the “setosa” group.
4. MultiSkew

The package MultiSkew ([48]) is written in the R programming language [45] and depends on the recommended package MaxSkew ([44]). It is available for download on the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/package=MultiSkew. The MultiSkew package computes the third multivariate cumulant of either the raw, centered or standardized data. It also computes the main measures of multivariate skewness, together with their bootstrap distributions. Finally, it computes the least skewed linear projections of the data. The MultiSkew package contains six different functions. First install it with the command R> install.packages("MultiSkew") and then use the command R> library(MultiSkew) to attach the package. Since the package MultiSkew depends on the package MaxSkew, the latter is loaded together with the former.

4.1. MinSkew

The function R> MinSkew(data, dimension) alleviates sample skewness by projecting the data onto appropriate linear subspaces and implements the method in [34]. It requires two input arguments: data (a data matrix object), and dimension (the number of required projections), which must be an integer between 2 and the number of the variables in the data matrix. The output has two values: Linear (the linear function of the variables) and Projections (the projected data). Linear is a matrix with the number of rows and columns equal to the number of variables and number of projections. Projections is a matrix whose number of rows and columns equal the number of observations and the number of projections. We call the function using our data matrix object: R> MinSkew(iris.m[,1:4],2). We obtain the matrix Linear as a first output. With the commands

R> attach(iris)
R> projections.species<-cbind(Projections,iris$Species)
R> pairs(projections.species[,1:2],col=c("red","green","blue")[,as.numeric(Species)])
R> detach(iris)

we obtain the multiple scatterplot of the two projections (Figure 5). The points clearly remind of bivariate normality and the groups markedly overlap with each other. Hence the projection removes the group structure as well as skewness. This result might be useful for a researcher interested in the features which are common to the three species.

![Figure 5. Scatterplot of two projections obtained with the function Minskew.](image-url)
The histograms of the \texttt{MinSkew} projections remind of univariate normality, as it can be seen from Figure 6a,b, obtained with the code

\begin{verbatim}
R> hist(Projections[,1], freq=FALSE)
R> curve(dnorm, col = 2, add = TRUE)
R> hist(Projections[,2], freq=FALSE)
R> curve(dnorm, col = 2, add = TRUE)
\end{verbatim}

![Histogram of Projections[, 1]](attachment:hist1.png)

![Histogram of Projections[, 2]](attachment:hist2.png)

**Figure 6.** (a) Histogram of the first projection of the Iris dataset obtained with \texttt{MinSkew}, (b) Histogram of the second projection of the Iris dataset obtained with \texttt{MinSkew}.

### 4.2. Third

This section describes the function which computes the third multivariate moment of a data matrix. Some general information about the third multivariate moment of both theoretical and empirical distributions are reviewed in [29]. The name of the function is \texttt{Third}, and its usage is

\begin{verbatim}
R> Third(data,type)
\end{verbatim}

As before, \texttt{data} is a data matrix object while \texttt{type} may be: “raw” (the third raw moment), “central” (the third central moment) and “standardized” (the third standardized moment). The output of the function, called \texttt{ThirdMoment}, is a matrix containing all moments of order three which can be obtained from the variables in \texttt{data}. We compute the third raw moments of the numerical variables in the iris dataset with the command

\begin{verbatim}
R > Third(iris.m[,1:4], "raw")
\end{verbatim}

The matrix \texttt{ThirdMoment} is:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>211.633</td>
<td>106.023</td>
<td>145.811</td>
<td>47.786</td>
</tr>
<tr>
<td>2</td>
<td>106.023</td>
<td>55.427</td>
<td>69.105</td>
<td>22.314</td>
</tr>
<tr>
<td>3</td>
<td>145.811</td>
<td>69.105</td>
<td>109.933</td>
<td>37.197</td>
</tr>
<tr>
<td>4</td>
<td>47.787</td>
<td>22.314</td>
<td>37.197</td>
<td>12.961</td>
</tr>
<tr>
<td>5</td>
<td>106.023</td>
<td>55.427</td>
<td>69.105</td>
<td>22.314</td>
</tr>
<tr>
<td>6</td>
<td>55.427</td>
<td>30.334</td>
<td>33.701</td>
<td>10.639</td>
</tr>
<tr>
<td>7</td>
<td>69.105</td>
<td>33.701</td>
<td>50.774</td>
<td>17.125</td>
</tr>
<tr>
<td>8</td>
<td>22.314</td>
<td>10.639</td>
<td>17.125</td>
<td>5.982</td>
</tr>
<tr>
<td>9</td>
<td>145.811</td>
<td>69.105</td>
<td>109.933</td>
<td>37.197</td>
</tr>
<tr>
<td>10</td>
<td>69.105</td>
<td>33.701</td>
<td>50.774</td>
<td>17.125</td>
</tr>
<tr>
<td>11</td>
<td>109.933</td>
<td>50.774</td>
<td>86.489</td>
<td>29.693</td>
</tr>
<tr>
<td>12</td>
<td>37.197</td>
<td>17.125</td>
<td>29.693</td>
<td>10.446</td>
</tr>
</tbody>
</table>
Similarly, we use "central" instead of "raw" with the command 

```r
R > Third(iris.m[,1:4], "central")
```

The output which appears in console is:

```
[ 1 ] "Third Moment"
```

and the structure of the object `ThirdMoment` is

```r
R > str(ThirdMoment)
num [1:16, 1:4] 0.1752 0.0420 0.1432 0.0259 ...
```

Finally, we set type equal to standardized: 

```r
R > Third(iris.m[,1:4], "standardized")
```

The output is:

```
[ 1 ] "Third Moment"
```

<table>
<thead>
<tr>
<th></th>
<th>[1]</th>
<th>[2]</th>
<th>[3]</th>
<th>[4]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>0.2988</td>
<td>-0.0484</td>
<td>0.3257</td>
<td>0.0034</td>
</tr>
<tr>
<td>[2]</td>
<td>-0.0484</td>
<td>0.0927</td>
<td>-0.0358</td>
<td>-0.0444</td>
</tr>
<tr>
<td>[3]</td>
<td>0.3257</td>
<td>-0.0358</td>
<td>0.0788</td>
<td>-0.2221</td>
</tr>
<tr>
<td>[4]</td>
<td>0.0034</td>
<td>-0.0444</td>
<td>-0.2221</td>
<td>0.0598</td>
</tr>
<tr>
<td>[5]</td>
<td>-0.0484</td>
<td>0.0927</td>
<td>-0.0358</td>
<td>-0.0444</td>
</tr>
<tr>
<td>[6]</td>
<td>0.0927</td>
<td>-0.0331</td>
<td>-0.1166</td>
<td>-0.0844</td>
</tr>
<tr>
<td>[7]</td>
<td>-0.0358</td>
<td>-0.1166</td>
<td>0.2894</td>
<td>0.1572</td>
</tr>
<tr>
<td>[8]</td>
<td>-0.0444</td>
<td>-0.0844</td>
<td>0.1572</td>
<td>0.2276</td>
</tr>
<tr>
<td>[9]</td>
<td>0.3257</td>
<td>-0.0358</td>
<td>0.0788</td>
<td>-0.2221</td>
</tr>
<tr>
<td>[10]</td>
<td>-0.0358</td>
<td>-0.1166</td>
<td>0.2894</td>
<td>0.1572</td>
</tr>
<tr>
<td>[11]</td>
<td>0.0788</td>
<td>0.2894</td>
<td>-0.0995</td>
<td>-0.3317</td>
</tr>
<tr>
<td>[12]</td>
<td>-0.2221</td>
<td>0.1572</td>
<td>-0.3317</td>
<td>0.3009</td>
</tr>
</tbody>
</table>
We show the structure of the matrix ThirdMoment with

\[
\begin{bmatrix}
13, & 0.0034 & -0.0444 & -0.2221 & 0.0598 \\
14, & -0.0444 & -0.0844 & 0.1572 & 0.2276 \\
15, & -0.2221 & 0.1572 & -0.3317 & 0.3009 \\
16, & 0.0598 & 0.2276 & 0.3009 & 0.8259 \\
\end{bmatrix}
\]

Third moments and cumulants might give some insights into the data structure. As a first example, use the command

\[
\text{R> } \text{Third(MaxSkew(iris.m[,1:50,1:4],50,2,TRUE),"standardized")}
\]

to compute the third standardized cumulant of the two most skewed, mutually orthogonal projections obtained from the four variables recorded from setosa flowers only. The resulting matrix is

\[
\begin{bmatrix}
1, & 1.2345 & 0.0918 \\
2, & 0.0918 & -0.0746 \\
3, & 0.0918 & -0.0746 \\
4, & -0.0746 & 0.5936 \\
\end{bmatrix}
\]

The largest entries in the matrix are the first element of the first row and the last element in the last row. This pattern is typical of outcomes from random vectors with skewed, independent components (see, for example, [29]). Hence the two most skewed projections might be mutually independent. As a second example, use the commands

\[
\text{R> MinSkew(iris.m[,1:4],2) and R> Third(Projections,"standardized")},
\]

where Projections is a value in output of the function MinSkew, to compute the third standardized cumulant of the two least skewed projections obtained from the four numerical variables of the Iris dataset. The resulting matrix is

\[
\begin{bmatrix}
1, & -0.0219 & 0.0334 \\
2, & 0.0334 & -0.0151 \\
3, & 0.0334 & -0.0151 \\
4, & -0.0151 & -0.0963 \\
\end{bmatrix}
\]

All elements in the matrix are very close to zero, as it might be better appreciated by comparing them with those in the third standardized cumulant of the original variables. This pattern is typical of outcomes from weakly symmetric random vectors ([34]).

### 4.3. Skewness Measures

The package **MultiSkew** has other four functions. All of them compute skewness measures. The first one is \texttt{R > FisherSkew(data)} and computes Fisher’s measure of skewness, that is the third standardized moment of each variable in the dataset. The usage of the function shows that there is only one input argument: data (a data matrix object). The output of the function is a dataframe, whose name is \texttt{tab}, containing Fisher’s measure of skewness of each variable of the dataset. To illustrate the function, we use the four numerical variables in the Iris dataset with the command \texttt{R > FisherSkew(iris.m[,1:4])} and obtain the output
The table below shows the variables and their corresponding values, as well as the Fisher skewness:

<table>
<thead>
<tr>
<th>Variables</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.0000</td>
<td>2.0000</td>
<td>3.0000</td>
<td>4.0000</td>
</tr>
</tbody>
</table>

in which X1 is the variable Sepal.Length, X2 is the variable Sepal.Width, X3 is the variable Petal.Length, and X4 is the variable Petal.Width. Another function is `PartialSkew`: \( R > \text{PartialSkew}(\text{data}) \). It computes the multivariate skewness measure as defined in [27]. The input is still a data matrix, while the values in output are the objects `Vector`, `Scalar` and `pvalue`. The first is the skewness measure and it has a number of elements equal to the number of the variables in the dataset used as input. The second is the squared norm of `Vector`. The last is the probability of observing a value of `Scalar` greater than the observed one, when the data are normally distributed and the sample size is large enough. We apply this function to our dataset: \( R > \text{PartialSkew}(\text{iris.m[,1:4]}) \) and obtain

\[
\begin{align*}
\text{Vector} & \quad R > [1] \\
& \quad [1,] 0.5301 \\
& \quad [2,] 0.4355 \\
& \quad [3,] 0.4105 \\
& \quad [4,] 0.4131 \\
\text{Scalar} & \quad R > [1] \\
& \quad [1,] 0.8098 \\
\text{pvalue} & \quad R > [1] \\
& \quad [1,] 0.0384
\end{align*}
\]

The function \( R > \text{SkewMardia}(\text{data}) \) computes the multivariate skewness introduced in [24], that is the sum of squared elements in the third standardized cumulant of the data matrix. The output of the function is the squared norm of the third cumulant of the standardized data (\( \text{MardiaSkewness} \)) and the probability of observing a value of \( \text{MardiaSkewness} \) greater than the observed one, when data are normally distributed and the sample size is large enough (\( \text{pvalue} \)). With the command \( R > \text{SkewMardia}(\text{iris.m[,1:4]}) \) we obtain

\[
\begin{align*}
\text{MardiaSkewness} & \quad R > [1] \\
& \quad 2.69722 \\
\text{pvalue} & \quad R > [1] \\
& \quad 0.4757998e-07
\end{align*}
\]

The function \( \text{SkewBoot} \) performs bootstrap inference for multivariate skewness measures. It computes the bootstrap distribution, its histogram and the \( p \)-value of the chosen measure of multivariate skewness using a given number of bootstrap replicates. The function calls the function `MaxSkew` contained in the `MaxSkew` package, with the number of iterations required by the function `MaxSkew` set equal to 5. The function’s usage is \( R > \text{SkewBoot}(\text{data}, \text{replicates}, \text{units}, \text{type}) \). It requires four inputs: `data` (the data matrix object), `replicates` (the number of bootstrap replicates), `units` (the number of rows in the data matrices sampled from the original data matrix object) and `type`. The latter may be “Directional”, “Partial” or “Mardia” (three different measures of multivariate skewness). If `type` is set equal to “Directional” or “Mardia”, `units` is an integer greater than the number of variables. If `type` is set equal to “Partial”, `units` is an integer greater than the number of variables augmented...
by one. The values in output are three: histogram (a plot of the above mentioned bootstrap distribution), Pvalue (the p-value of the chosen skewness measure) and Vector (the vector containing the bootstrap replicates of the chosen skewness measure). For the reproducibility of the result, before calling the function SkewBoot, we type \texttt{R> set.seed(101)} and then \texttt{R > SkewBoot(iris.m[,1:4],10,11,"Directional")}. We obtain the output

\begin{verbatim}
[1] "Vector"
[1] 2.0898 1.4443 1.0730 0.7690 0.6914 0.3617 0.2375 0.0241
[9] −0.1033 0.6092
[1] "Pvalue"
[1] 0.7272727
\end{verbatim}

and also the histogram of bootstrapped directional skewness (Figure 7).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{histogram.png}
\caption{Histogram of bootstrapped Directional skewness}
\end{figure}

We call the function \texttt{SkewBoot}, first setting \texttt{type} equal to \texttt{Mardia} and then equal to \texttt{Partial}:

\begin{verbatim}
R> set.seed(101)
R > SkewBoot(iris.m[,1:4],10,11,"Mardia").
We obtain the output
\begin{verbatim}
[1] "Vector"
[1] 1.4768 1.1260 0.8008 0.6164 0.4554 0.1550 0.0856
−0.1394 −0.1857 0.4018
[1] "Pvalue"
[1] 0.6363636
\end{verbatim}
\end{verbatim}

\begin{verbatim}
R> set.seed(101)
R > SkewBoot(iris.m[,1:4],10,11,"Partial")
\end{verbatim}
Figure 8a,b contain the histograms of bootstrapped Mardia’s skewness and bootstrapped partial skewness, respectively.

![Histogram of bootstrapped Mardia skewness](a)

![Histogram of bootstrapped Partial skewness](b)

**Figure 8.** (a) Bootstrapped Mardia’s skewness of Iris dataset, (b) Bootstrapped Partial skewness of Iris dataset.

We shall now compute the Fisher’s skewnesses of the four variables in the “setosa” group with R > FisherSkew(iris.m[1:50,1:4]). The output is the dataframe

R> tab

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variables</strong></td>
<td>1.0000</td>
<td>2.0000</td>
<td>3.0000</td>
<td>4.0000</td>
</tr>
<tr>
<td><strong>Fisher Skewness</strong></td>
<td>0.1165</td>
<td>0.0399</td>
<td>0.1032</td>
<td>1.2159</td>
</tr>
</tbody>
</table>

Ref. [43] showed that nonnormality of variables in the “setosa” group went undetected by Mardia’s skewness calculated on all four of them (the corresponding $p$-value is 0.1772). We compute Mardia’s skewness of the two most skewed variables (sepal length and petal width), with the commands

R>iris.m.mardia<-cbind(iris.m[1:50,1],iris.m[1:50,4])
R>SkewMardia(iris.m.mardia).

We obtain the output

R> MardiaSkewness

[1] 1.641217

R> pvalue

[1] 0.008401288
The $p$-value is highly significant, clearly suggesting the presence of skewness and hence of nonnormality. We conjecture that the normality test based on Mardia’s skewness is less powerful when skewness is present only in a few variables, either original or projected, while the remaining variables might be regarded as background noise. We hope to either prove or disprove this conjecture by both theoretical results and numerical experiments in future works.

5. Conclusions

MaxSkew and MultiSkew are two R packages aimed at detecting, measuring and removing multivariate skewness. They also compute the three main skewness measures. The function SkewBoot computes the bootstrap $p$-value corresponding to the chosen skewness measure. Skewness removal might be achieved with the function MinSkew. The function Third, which computes the third moment, plays a role whenever the researcher compares the third sample moment with the expected third moment under a given model, in order to get a better insight into the model’s fit.

The major drawback of both MaxSkew and Multiskew is that they address skewness by means of third-order moments only. In the first place, they may not exist even if the distribution is skewed, as it happens for the skew-Cauchy distribution. In the second place, the third moment of a random vector may be a null matrix also when the distribution of random vector itself is asymmetric. In the third place, third-order moments are not robust to outliers. We are currently investigating these problems.

Author Contributions: Conceptualization, C.F. and N.L.; methodology, N.L.; software, C.F.; data curation, C.F.; writing, N.L.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

15. Mardia, K. The effect of nonnormality on some multivariate tests and robustness to nonnormality in the linear model. *Biometrika* 1971, 58, 105–121. [CrossRef]
23. Mardia, K. Measures of multivariate skewness and kurtosis with applications. *Biometrika* 1997, 57, 519–530. [CrossRef]


47. Hui, G.; Lindsay, B. Projection pursuit via white noise matrices. *Sankhya B* **2010**, 72, 123–153. [CrossRef]


© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).