An R package for Non-Normal Multivariate Distributions: Simulation and Probability Calculations from Multivariate Lomax (Pareto Type II) and Other Related Distributions

by Zhixin Lun and Ravindra Khattree

Abstract Convenient and easy-to-use programs are readily available in R to simulate data from and probability calculations for several common multivariate distributions such as normal and t. However, functions for doing so from other less common multivariate distributions, especially those which are asymmetric, are not as readily available, either in R or otherwise. We introduce the R package NonNorMvtDist to generate random numbers from multivariate Lomax distribution, which constitutes a very flexible family of skewed multivariate distributions. Further, by applying certain useful properties of multivariate Lomax distribution, multivariate cases of generalized Lomax, Mardia’s Pareto of Type I, Logistic, Burr, Cook-Johnson’s uniform, F, and inverted beta can be also considered, and random numbers from these distributions can be generated. Methods for the probability and the equicoordinate quantile calculations for all these distributions are then provided. This work substantially enriches the existing R toolbox for nonnormal or nonsymmetric multivariate probability distributions.

Introduction

A k-dimensional multivariate Lomax (Pareto Type II) probability distribution was first introduced by (Nayak, 1987) as a joint distribution of k skewed nonnegative random variables X₁, · · · , Xₖ with joint probability density function given by

\[
f(x₁, . . . , xₖ) = \frac{[\prod_{i=1}^{k} \theta_i] a(a + 1) \cdots (a + k - 1)}{\left(1 + \sum_{i=1}^{k} \theta_i x_i\right)^{a+k}}, \quad x_i > 0, a, \theta_i > 0, i = 1, . . . , k.
\]  

We will denote above density function by MLₖ(a; θ₁, . . . , θₖ). Prior to Nayak (1987), the bivariate case of multivariate Lomax distribution was studied by Lindley and Singpurwalla (1986). Nayak (1987) indicated that the k-dimensional multivariate Lomax distribution could be obtained by mixing k independent univariate exponential distributions with different failure rates with the mixing parameter \(\eta\) that has a gamma distribution with certain shape parameter a and the scale parameter 1. This fact readily provides an approach to simulate the multidimensional random vectors from the multivariate Lomax distribution. The multivariate Lomax distribution is also transformable to many other useful multivariate distributions, and therefore, simulations from these distributions are also easily accomplished. Similarly, with appropriate transformations or reparameterizations (or otherwise directly from the probability density function (pdf)), we can also accomplish the cumulative probability calculations as well as the calculation of equicoordinate quantiles. The objective of this work is to formalize all of the above and to provide a ready-to-use R package titled NonNorMvtDist for practitioners to efficiently execute the same. See Lun and Khattree (2020).

The objective of our work is to enrich the existing R packages for supporting simulation and computations for nonnormal continuous multivariate distributions. To the best of our knowledge and also from the list of packages for multivariate distributions in CRAN (https://cran.r-project.org/web/views/Distributions.html), there is no package that provides both simulation and probability computations for multivariate Pareto distribution. In our package, we provide functions for doing so to both Lomax (Pareto type II) and Mardia’s Pareto type I distributions. For multivariate logistic distribution, package VGAM (Yee, 2019) implements the bivariate logistic distribution while we support p-variate logistic distribution for \(p > 2\). Moreover, multivariate Burr, F, and inverted beta distributions had not been implemented in R until we included them in the package NonNorMvtDist.

This paper is organized as follows. In Section Multivariate Lomax and related distributions, we provide simulation algorithms for generating data from k-dimensional multivariate Lomax and generalized Lomax (to be defined later) distributions. Through transformations, the tasks of random
numbers generation from Mardia’s multivariate Pareto Type I, Logistic, Burr, Cook-Johnson’s uniform, and F are achieved. Based on the remarks from Balakrishnan and Lai (2009), we then extend Nayak’s work to simulate data from the multivariate inverted beta distribution. In Section Probability computations, we discuss numerical computations of cumulative distribution functions of equicoordinate quantiles and survival functions for the above distributions. In Section Illustrations of simulations and probability calculations, we illustrate the use of respective functions as implemented in R for each of the above distributions for the bivariate (k = 2) case. In Section Computation times, we provide a run-time study to assess the computation times for functions as the dimension of data increases. In Section Maximum likelihood estimation of parameters, we implement maximum likelihood estimation of parameters for these distributions. In Section Two applications, we give two applications of package NonNorMvtDist, namely, (i) data generation from certain nonelliptical symmetric multivariate distributions with univariate normal marginals and (ii) computation of critical values of the multivariate F distribution. Section Concluding remarks includes some concluding remarks, pointing out other applications.

**Multivariate Lomax and related distributions**

Multivariate Lomax distribution can be derived as the probability distribution of a k-component system where k independent exponential random variables have a common environment or mixing parameter following a gamma distribution with shape parameter a and scale parameter b. Let the corresponding random vector be X = (X1, · · · , Xk)′. The probability density function of X is given by (1), and the joint survival function of X is

\[
S(x_1, \ldots, x_k) = \left(1 + \sum_{i=1}^{k} \theta_i x_i\right)^{-a}, \quad x_i > 0, a > 0, \theta_i > 0, i = 1, \ldots, k. \tag{2}
\]

Specifically, the pivotal result that we use is given by the following theorem (see Nayak (1987)),

**Theorem 2.1**: Conditioned on fixed mixing parameter η, representing the environment effect, let X1, · · · , Xk be independent exponentially distributed random variables with failure rates ηλ1, · · · , ηλk, respectively. Let the environment effect η be distributed as a Gamma random variate with probability density

\[
g(\eta) = b^a \exp(-\eta b)\eta^{a-1}/\Gamma(a), \eta > 0, a, b > 0.
\]

Then, the unconditional joint density of X1, · · · , Xk is given by (1), where \(\theta_i = \lambda_i / b, i = 1, \ldots, k\). Clearly, without loss of generality, b can be taken as 1, in which case \(\theta_i = \lambda_i, i = 1, \ldots, k\).

In view of the above result, we implement the simulation from k-dimensional multivariate Lomax distribution by adopting the following algorithm.

**Algorithm-MLk(a; \theta_1, \ldots, \theta_k)**:

1. Generate a random number η from Gamma(a, 1) distribution;
2. With η as generated in Step 1, generate k-independent random variables \(X_i, i = 1, \ldots, k\), each from exponential distribution with parameter \(\eta \theta_i, i = 1, \ldots, k\), respectively. Let X = (X1, X2, · · · , Xk)′;
3. To obtain a random sample of size n, repeat the Steps 1 and 2 n times.

Nayak (1987) also generalized this distribution by mixing conditionally independent \(X_i\) having the Gamma(\(l_i, \eta \theta_i\)) distribution, with mixing variable \(\eta \sim \text{Gamma}(a, 1), i = 1, \ldots, k\). This is termed as generalized multivariate Lomax distribution denoted by GML(a; \(\theta_1, \ldots, \theta_k, l_1, \ldots, l_k\)) and has the probability density function

\[
f(x_1, \ldots, x_k) = \frac{\prod_{i=1}^{k} l_i^{\theta_i} \Gamma \left( \sum_{i=1}^{k} l_i + a \right) \prod_{i=1}^{k} x_i^{l_i - 1}}{\Gamma(a) \left[ \prod_{i=1}^{k} \Gamma(l_i) \right] \left( 1 + \sum_{i=1}^{k} \theta_i x_i \right)^{\sum_{i=1}^{k} l_i + a}}, \quad x_i > 0, a, \theta_i, l_i > 0, i = 1, \ldots, k \tag{3}
\]

Accordingly, we perform the corresponding simulation by implementing the suitable changes in the above algorithm. The algorithm is given below.

**Algorithm-GML(a; \theta_1, \ldots, \theta_k, l_1, \ldots, l_k)**:

1. Generate a random number η from Gamma(a, 1) distribution;
2. With η as generated in Step 1, generate k-independent $X_i$, each following $\text{Gamma}(l_i, \eta \theta_i)$, $i = 1, \ldots, k$, respectively;
3. To obtain a random sample of size n, repeat the Steps 1 and 2 n times.

Both algorithms are easily implemented using R stats (R Core Team, 2019) functions rexp() and rgamma(), respectively, for generating univariate exponential and gamma random variates. In the following, we describe approaches to generate other distributions related to multivariate Lomax and generalized multivariate Lomax distributions.

Nayak (1987) has also discussed the inter-relationships between many other multivariate distributions and generalized multivariate Lomax distribution. In view of these inter-relationships, the above algorithm can accordingly be amended to simulate data from these distributions - a task which can be quite difficult to accomplish directly. These inter-relationships are described in Table 1. For convenience, we assume $X = (X_1, \ldots, X_k)' \sim \text{MLM}(a; \theta_1, \ldots, \theta_k)$ and $T = (T_1, \ldots, T_k)' \sim \text{GMLM}(a; \theta_1, \ldots, \theta_k)$.

| Multivariate Distribution | Transformation/Parameter Substitutions | Probability Density Function |
|---------------------------|----------------------------------------|-----------------------------|
| Lomax                     | $l_i = 1$, $i = 1, \ldots, k$          | $f(x_1, \ldots, x_k) = \frac{[\prod_{i=1}^{k} \theta_i a_i^{a_i-1}(a_i+1)^{a_i+1-k}]}{(1-a_i+1)!} x_i^{a_i-1} \theta_i^a$ |
| Mardia’s                  | $l_i = 1$, $i = 1, \ldots, k$          | $f(y_1, \ldots, y_k) = \frac{[\prod_{i=1}^{k} \theta_i a_i^{a_i-1}(a_i+1)^{a_i+1-k}]}{(1-a_i+1)!} y_i^{a_i-1} \theta_i^a$ |
| Pareto Type I             | $l_i = 1$ and $a = 1$, $i = 1, \ldots, k$ | $f(w_1, \ldots, w_k) = \frac{\mu \exp(-\sum_{i=1}^{k} \frac{x_i}{\mu_1})}{\Gamma(k) \Gamma(1)} \frac{1}{\prod_{i=1}^{k} \theta_i a_i^{a_i-1}(a_i+1)^{a_i+1-k}}$ |
| Logistic                  | $l_i = 1$, $i = 1, \ldots, k$          | $f(b_1, \ldots, b_k) = \frac{\mu \exp(-\sum_{i=1}^{k} \frac{x_i}{\mu_1})}{\Gamma(k) \Gamma(1)} \frac{1}{\prod_{i=1}^{k} \theta_i a_i^{a_i-1}(a_i+1)^{a_i+1-k}}$ |
| Burr                      | $l_i = 1$, $i = 1, \ldots, k$          | $f(v_1, \ldots, v_k) = \frac{\mu \exp(-\sum_{i=1}^{k} \frac{x_i}{\mu_1})}{\Gamma(k) \Gamma(1)} \frac{1}{\prod_{i=1}^{k} \theta_i a_i^{a_i-1}(a_i+1)^{a_i+1-k}}$ |
| Cook-Johnson’s uniform    | $l_i = 1$, $i = 1, \ldots, k$          | $f(t_1, \ldots, t_k) = \frac{\Gamma(a+k \mu \sum_{i=1}^{k} x_i) \Gamma(k) \Gamma(1)}{\prod_{i=1}^{k} \theta_i a_i^{a_i-1}(a_i+1)^{a_i+1-k}}$ |

The multivariate F distribution can also be obtained by considering

$$T_i = \frac{S_i / (2l_i)}{S_0 / (2a)} \quad i = 1, \ldots, k,$$

where $S_0, S_1, \ldots, S_k$ are independent Chi-square variables with $2a, 2l_1, \ldots, 2l_k$ degrees of freedom respectively; see Johnson and Kotz (1972). It is the joint distribution of the ratios of mean squares under certain linear hypotheses on treatments as discussed in Krishnaiah (1965) in the context of simultaneous ANOVA and MANOVA tests where $S_0$ is a residual sum of squares and $S_i$'s are various effect sums of squares. The density given in Table 1 is a special case of generalized multivariate F distribution defined by Krishnaiah (1965) when $S_i$'s, $i = 1, \ldots, k$, are all independent. This fact is useful in that the Tables given by Armitage and Krishnaiah (1964) make use of this in constructing the statistical tables for certain linear hypotheses. We use these tables to confirm our calculation as done by our R programs. The multivariate Inverted Beta distribution also called the multivariate inverted Dirichlet distribution, is essentially a special case of multivariate F distribution when $l_1 = l_2 = \cdots = l_k = a$.  

The R Journal Vol. 13/2, December 2021
ISSN 2073-4859
Figure 1 pictorially summarizes the relationships between (generalized) Lomax and other distributions.

\[
\begin{align*}
F(2a, 2l) & \quad \text{Generalized Lomax} \quad (a, \theta, l) \\
\theta_i &= l_i / a \\
Lomax(\alpha, \theta) & \quad \text{Logistic} \quad (\mu, \sigma) \\
\mu_i - \sigma_i \ln(\theta_i x_i) & \\
Mardia’s \text{ Pareto Type I} (a, \theta) & \quad \text{Inverted Beta} \quad (a, l) \\
(1 + \theta_i x_i)^{-a} & \\
\text{Cook-Johnson’s Uniform} (a) & \quad \text{Burr} \quad (a, d, c) \\
(\theta_i x_i / d_i)^{1/c_i} & \\
\text{(Clayton copula)} & \\
\end{align*}
\]

Figure 1: Relationships among (generalized) multivariate Lomax and other distributions. Solid lines represent transformations and dashed dot lines represent reparameterization (parameter substitutions).

Probability computations

Here, we give details of computations of cumulative probability distribution function (cdf), survival function, equicoordinate quantile function for each distribution introduced in Section Multivariate Lomax and related distributions. Depending on the situation, the calculation may sometimes be simpler for joint cdf or joint survival function.

Distributions transformable from Lomax distribution

The multivariate Lomax distribution has an explicit closed-form expression for the joint survival function given by (2). The survival or cumulative distribution function of other related distributions can be obtained either directly or through appropriate transformations. We summarize these explicit expressions of cumulative distribution function \(F(\cdot)\) and survival function \(S(\cdot)\) in Table 2.

For the cumulative distribution functions or survival functions with no closed-form expressions, we rely on the following useful formulas (Joe, 1997):

\[
\begin{align*}
S(\mathbf{x}) &= 1 + \sum_{\mathbf{C} \subseteq C} (-1)^{|\mathbf{C}|} F_C(x_j, j \in \mathbf{C}), \quad (5) \\
F(\mathbf{x}) &= 1 + \sum_{\mathbf{C} \subseteq C} (-1)^{|\mathbf{C}|} S_C(x_j, j \in \mathbf{C}), \quad (6)
\end{align*}
\]

where \(F_C(x_j, j \in \mathbf{C}) \) (\(S_C(x_j, j \in \mathbf{C}) \)) is the joint cdf (joint survival function) of \(x_j \) where the subscripts belong to the set \(C \), which is a subset of \(\{1, 2, \cdots, k\} \). Clearly, \(C \subseteq \mathbb{C} \) where \(\mathbb{C} \) is the powerset of \(\{1, 2, \cdots, k\} \). Also, \(|C|\) represents the cardinality of \(C \).

The equation

\[
P[x_i \leq q_i, i = 1, \cdots, k] = \int_0^{q_1} \cdots \int_0^{q_k} f(x_1, \cdots, x_k) dx_k \cdots dx_1 = p
\]
for a given \( p \) does not have a unique solution \( (q_1, \ldots, q_k) \). We thus provide the quantile computations only for the equicoordinate quantile, obtained by solving the following equation for \( q_i \):

\[
P[X_i \leq q_i, i = 1, \cdots, k] = \int_0^q \cdots \int_0^q f(x_1, \ldots, x_k)dx_k \cdots dx_1 = p,
\]

where \( 0 < p < 1 \) is a (given) cumulative probability. We make use of the R \texttt{stats} function \texttt{uniroot()}, which is used for finding one dimensional root.

**Distributions related to generalized multivariate Lomax distribution**

For generalized multivariate Lomax distribution and its related distributions, explicit expressions of the cumulative distribution function and survival function are not available. Thus, we obtain the cumulative distributions and survival functions of multivariate Lomax and related distributions.

\[
S(x_1, \ldots, x_k) = \left(1 + \sum_{i=1}^k \theta_i x_i\right)^{-d}
\]

\[
y_i > 0, \quad a, \theta_i > 0
\]

\[
S(y_1, \ldots, y_k) = \left(\sum_{i=1}^k \theta_i y_i - k + 1\right)^{-d}
\]

\[
y_i > 0, \quad a, \theta_i > 0
\]

Table 2: Cumulative distribution functions and survival functions of multivariate Lomax and related distributions.

| Multivariate Distribution | Cumulative Distribution Function | Survival Function |
|---------------------------|----------------------------------|------------------|
| Lomax                     | No closed form                   | \( S(x_1, \ldots, x_k) = \left(1 + \sum_{i=1}^k \theta_i x_i\right)^{-d} \) |
| Mardia’s Pareto Type I    | No closed form                   | \( S(y_1, \ldots, y_k) = \left(\sum_{i=1}^k \theta_i y_i - k + 1\right)^{-d} \) |
| Logistic                  | No closed form                   | \( S(x_1, \ldots, x_k) = \left(\sum_{i=1}^k \theta_i x_i\right)^{-d} \) |
| Burr                      | No closed form                   | \( S(b_1, \ldots, b_k) = \left(1 + \sum_{i=1}^k d_i b_i^d\right)^{-d} \) |
| Cook-Johnson’s Uniform    | No closed form                   | \( S(b_1, \ldots, b_k) = \left(1 + \sum_{i=1}^k d_i b_i^d\right)^{-d} \) |

\[
F(v_1, \ldots, v_k) = \left[\sum_{i=1}^k v_i^{-1/a} - k + 1\right]^{-d}
\]

\[
o < v_i \leq 1, \quad a > 0
\]

Property 3.1: Let \( T_1, \cdots, T_k \) be \( k \) continuous random variables that jointly follow the GML\(_k\)\(\{a; \theta_1, \ldots, \theta_k, l_1, \ldots, l_k\}\) distribution as given in (3). Then, the cumulative distribution function of \( T_1, \cdots, T_k \) can be computed as

\[
F(x_1, \ldots, x_k) = P(T_1 \leq x_1, \ldots, T_k \leq x_k) = P(U_1 \leq 1, \ldots, U_k \leq 1),
\]

where \( U_1, \ldots, U_k \) jointly follow GML\(_k\)\(\{a; \theta_1 x_1, \ldots, \theta_k x_k, l_1, \ldots, l_k\}\) distribution.

Proof of the above is straightforward by making the substitutions

\[
u_i = \frac{t_i}{x_i}, \quad i = 1, \cdots, k.
\]

Through parameter substitutions, the cumulative distribution functions of multivariate \( F \) and the inverted beta distribution can be found. These are summarized in Table 3.
Contributed Research Articles

We will illustrate here the functions and corresponding arguments for the R Journal Vol. 13/2, December 2021 ISSN 2073-4859

Consequently, for large dimensions, the approach requires more computing power and time (in fact, to simulate data from the normal distribution, many programs themselves first require random number generations from uniform distributions, from which normal random numbers are obtained).

| Distribution | Parameter Substitutions | Cumulative Distribution Function |
|--------------|-------------------------|----------------------------------|
| Multivariate F with degrees of freedom | \( \theta_i = i \alpha_i \), \( i = 1, \ldots, k \) | \( F(x_1, \ldots, x_k) = P(U_1 \leq x_1, \ldots, U_k \leq x_k) \), \( (U_1, \ldots, U_k) \) follows \( \text{GML}_k(\alpha_1/a_1, \ldots, \alpha_k/a_k, \theta_1, \ldots, \theta_k) \) |

Table 3: Cumulative distribution functions of related multivariate distributions to generalized multivariate Lomax.

Algorithm - CDF Computation using Monte Carlo Method:

1. Generate \( N \) random vectors \( t^{(i)} = (t_1^{(i)}, \ldots, t_k^{(i)})', i = 1, \ldots, N \) from the desired distribution;
2. Compute

\[
\hat{F}(T_1 \leq x_1, \ldots, T_k \leq x_k) = \frac{1}{N} \sum_{i=1}^{N} I(t^{(i)} \leq x),
\]

where \( x = (x_1, \ldots, x_k)' \) and \( I(\cdot) \) is the zero-one indicator function corresponding to the conditions specified.

Step 1 above is readily carried out by the random numbers generation as described in Section Multivariate Lomax and related distributions using the package NonNorMvtDist. Since \( cdf \) is computable using adaptive multivariate integration over unit cube \([0,1]^k\) or via the Monte Carlo method, it follows that the survival function can also be calculated (by using \( \theta \)). The equicoordinate quantile is computed by using \( \gamma \).

We also add in our package the calculations of joint probability density function - Being self-explanatory with all \( pdfs \) available in closed form, it needs no further elaboration. The corresponding function is \( \text{dmv*}() \).

Illustrations of simulations and probability calculations

We will illustrate here the functions and corresponding arguments for NonNorMvtDist. The calling sequences include probability density calculation (\( \text{dmv*} \)), cumulative distribution calculation (\( \text{pmv*} \)), equicoordinate quantile calculation (\( \text{qmv*} \)), random numbers generation (\( \text{rmv*} \)), and survival function calculation (\( \text{smv*} \)) for each of the multivariate distributions introduced in Section Multivariate Lomax and related distributions. For each distribution, we consider the bivariate case \( (k = 2) \). This choice enables us to also succinctly and graphically present the probability density plots. The detailed description of the calling sequence for each of the several cases has been moved into a digital complement of this paper.

For example, for the bivariate Lomax distribution \( (k = 2) \) with parameters \( a = 5, \theta = (0.5, 1) \), the calling sequences for various functions are

\[
\begin{align*}
d\text{mvloax}(x, \text{parm1} = 5, \text{parm2} = (0.5, 1)) \\
p\text{mvloax}(q, \text{parm1} = 5, \text{parm2} = (0.5, 1)) \\
q\text{mvloax}(p, \text{parm1} = 5, \text{parm2} = (0.5, 1)) \\
r\text{mvloax}(n, \text{parm1} = 5, \text{parm2} = (0.5, 1)) \\
\text{smvloax}(q, \text{parm1} = 5, \text{parm2} = (0.5, 1))
\end{align*}
\]

It may be mentioned that our approach may be more efficient than the NORTA method (Ghosh and Henderson, 2002) for simulation in that NORTA always first requires simulation from multivariate normal, which are then transformed to multivariate uniform. Only often this step, one could subsequently transform the simulated data to the desired distribution. Consequently, for large dimensions, the approach requires more computing power and time (in fact, to simulate data from the normal distribution, many programs themselves first require random number generations from uniform distributions, from which normal random numbers are obtained).
3D bivariate density plot

For each bivariate distribution, we provide the density surface plots along with contours using the function `persp3D()` from the package `plot3D` (Soetaert, 2017). To illustrate, we define function `dplot2()`, and we pass appropriate density functions to the density function argument `dfun` to create the density surfaces. These are summarized in the digital complement along with corresponding resulting plots. For the Lomax distribution with parameters indicated previously, the statements will be

```r
library(plot3D)
dplot2 <- function(dfun, x1, x2, zlim) {
    zmat <- matrix(0, nrow = length(x1), ncol = length(x2))
    for (i in 1:length(x1)) {
        for (j in 1:length(x2)) {
            zmat[i, j] = dfun(x = c(x1[i], x2[j]))
        }
    }
    persp3D(z = zmat, x = x1, y = x2, theta = -60, phi = 10, ticktype = "detailed",
           zlim = zlim, contour = list(nlevels = 30, col = "red"),
           facets = FALSE, image = list(col = "white", side = "zmin"),
           xlab = "X1", ylab="X2", zlab = "Density", expand = 0.5, d = 2)
}
dplot2(dfun = function(x) dmvlomax(x, parm1 = 5, parm2 = c(0.5, 1)), x1 = seq(0, 4, 0.1),
       x2 = seq(0, 4, 0.1), zlim = c(-5, 13))
```

The plot that results is shown in Figure 2.

![Figure 2: Density surface of bivariate Lomax distribution with parameters \(a = 5, \theta = (0.5,1)\).](image)

Random number generation

The following code illustrates the use of the function `rmvlomax()` with a bivariate sample of size \(n = 2\). Sampling is done by setting `set.seed(2019)` in advance. The digital complement explicitly provides the code as well as output for all of the probability distributions discussed here. In the output, each row represents a bivariate observation.

- **Bivariate Lomax**: \(a = 5, \theta_1 = 0.5, \theta_2 = 1\)

  ```r
  > set.seed(2019)
  > rmvlomax(n = 2, parm1 = 5, parm2 = c(0.5, 1))
  [,1] [,2]
  [1,] 1.0174406 0.7876480
  [2,] 0.3686253 0.7826978
  ```
CDF, survival function and equicoordinate quantile

The applications of \( cd f \) \( pm\mathbf{v^*}() \), survival function \( sm\mathbf{v^*}() \), and equicoordinate quantile function \( q\mathbf{m\mathbf{v^*}}() \) are straightforward and follow the same pattern earlier. See the digital complement for computation details. In the following, we give code as well as output only for Lomax distribution \((a = 5, \theta_1 = 0.5, \theta_2 = 1)\) for specified coordinates \((x_1, x_2)\) and for the cumulative probability \( p = 0.5\).

• **Bivariate Lomax**: \( a = 5, \theta_1 = 0.5, \theta_2 = 1; \) quantiles: \((x_1, x_2) = (1, 0.5)\).

\[
\begin{align*}
\text{> pmvlomax}(q = c(1, 0.5), \text{parm1} = 5, \text{parm2} = c(0.5, 1)) \\
\text{[1]} & 0.7678755 \\
\text{> smvlomax}(q = c(1, 0.5), \text{parm1} = 5, \text{parm2} = c(0.5, 1)) \\
\text{[1]} & 0.03125 \\
\text{> qmvlomax}(p = 0.5, \text{parm1} = 5, \text{parm2} = c(0.5, 1)) \\
\text{[1]} & 0.3928917
\end{align*}
\]

Computation times

We assess the run-times for the computation of probability \( (pm\mathbf{v^*}) \), equicoordinate quantile function \( (q\mathbf{m\mathbf{v^*}}) \) for multivariate Lomax, and generalized multivariate Lomax distributions for reference. The survival function \( (sm\mathbf{v^*}) \) of multivariate Lomax distribution has a closed-form expression, and hence the assessment of computation time is omitted. We have used the computer with Intel Core i5-8250U CPU and 8.00 GB RAM. The results for \( p \)-variate Lomax distribution are summarized in Table 4. As we can see, the run-times for \( pm\mathbf{v\mathbf{l\mathbf{om\mathbf{a\mathbf{x}}}}}(\cdot) \) are quit short, even for the dimension \( p = 20 \). However, \( q\mathbf{m\mathbf{v\mathbf{l\mathbf{om\mathbf{a\mathbf{x}}}}}(\cdot) \) requires a considerable longer time when \( p \geq 17 \), which seems to double for every extra dimension added to the size of the random vector. The computation times in Table 4 can also be used as a reference for the distributions related to multivariate Lomax, which are in Table 2 since we apply the same approach for probability computations there as well.

| \( p \) | \( pm\mathbf{v\mathbf{lo\mathbf{m\mathbf{a\mathbf{x}}}}}(\cdot) \) | \( qm\mathbf{v\mathbf{lo\mathbf{m\mathbf{a\mathbf{x}}}}}(\cdot) \) | \( p \) | \( pm\mathbf{v\mathbf{lo\mathbf{m\mathbf{a\mathbf{x}}}}}(\cdot) \) | \( qm\mathbf{v\mathbf{lo\mathbf{m\mathbf{a\mathbf{x}}}}}(\cdot) \) |
|---|---|---|---|---|---|
| 1 | 0.01695895 | 0.03702188 | 11 | 0.04889703 | 0.7992568 |
| 2 | 0.01795197 | 0.02293706 | 12 | 0.09025002 | 1.879766 |
| 3 | 0.01794696 | 0.02789807 | 13 | 0.1545889 | 3.087925 |
| 4 | 0.01795197 | 0.03290296 | 14 | 0.245369 | 5.950396 |
| 5 | 0.01976085 | 0.04189205 | 15 | 0.4657819 | 11.37162 |
| 6 | 0.01995182 | 0.06582808 | 16 | 0.853502 | 22.04591 |
| 7 | 0.02094102 | 0.098768 | 17 | 1.708418 | 45.51909 |
| 8 | 0.02197218 | 0.146611 | 18 | 2.803703 | 90.70764 |
| 9 | 0.02396512 | 0.2393882 | 19 | 5.453189 | 181.55772 |
| 10 | 0.04587412 | 0.4296861 | 20 | 10.40903 | 351.58896 |

**Table 4:** Runtimes (in seconds) for functions \( pm\mathbf{v\mathbf{l\mathbf{om\mathbf{a\mathbf{x}}}}}(\cdot) \) and \( qm\mathbf{v\mathbf{l\mathbf{om\mathbf{a\mathbf{x}}}}}(\cdot) \) as functions of \( p \).

The results for generalized \( p \)-variate Lomax distribution are summarized in Tables 5 and 6. Both functions \( pm\mathbf{v\mathbf{g\mathbf{l\mathbf{om\mathbf{a\mathbf{x}}}}}(\cdot) \) and \( sm\mathbf{v\mathbf{g\mathbf{l\mathbf{om\mathbf{a\mathbf{x}}}}}(\cdot) \) require relatively much longer time when \( p > 4 \), and \( q\mathbf{m\mathbf{v\mathbf{g\mathbf{l\mathbf{om\mathbf{a\mathbf{x}}}}}(\cdot) \) takes longer when \( p > 2 \). Based on the run-time consumption, we recommend algorithm \( MC \) for larger dimensions (e.g., when \( p > 5 \)). Similarly, this run-time study can also be used as a reference for the related distributions (to generalized Lomax distribution) as listed in Table 3 since we apply the same method for computations.

| \( p \) | \( pm\mathbf{v\mathbf{g\mathbf{l\mathbf{om\mathbf{a\mathbf{x}}}}}(\cdot) \) | \( sm\mathbf{v\mathbf{g\mathbf{l\mathbf{om\mathbf{a\mathbf{x}}}}}(\cdot) \) | \( p \) | \( pm\mathbf{v\mathbf{g\mathbf{l\mathbf{om\mathbf{a\mathbf{x}}}}}(\cdot) \) | \( sm\mathbf{v\mathbf{g\mathbf{l\mathbf{om\mathbf{a\mathbf{x}}}}}(\cdot) \) |
|---|---|---|---|---|---|
| 1 | 0.03395414 | 4.600386 | 0.03084397 | 3.977374 |
| 2 | 0.02397585 | 6.536522 | 0.05378199 | 5.510087 |
| 3 | 0.101758 | 8.396577 | 0.245369 | 5.950396 |
| 4 | 0.9758801 | 10.64278 | 2.13447 | 8.741386 |
| 5 | 16.43411 | 19.30024 | 40.11997 | 9.762063 |
| 6 | 305.50092 | 19.57221 | 134.4908 | 11.63218 |

**Table 5:** Runtimes (in seconds) for functions \( pm\mathbf{v\mathbf{g\mathbf{l\mathbf{om\mathbf{a\mathbf{x}}}}}(\cdot) \) and \( sm\mathbf{v\mathbf{g\mathbf{l\mathbf{om\mathbf{a\mathbf{x}}}}}(\cdot) \) by using algorithms \( \text{numerical} \) and \( \text{MC} \) as functions of \( p \).
Maximum likelihood estimation of parameters

We also include the maximum likelihood estimation to estimate the parameters for various Lomax related distributions (except for the bivariate $F$ distribution). Although many of the density functions in Section Multivariate Lomax and related distributions have complicated forms, maximum likelihood estimation can be easily accomplished by using the built-in optimization functions in R stats. The log-likelihood function for a given sample $x_1, \ldots, x_n$ is given by

$$
\mathcal{L}(\theta | x_1, \ldots, x_n) = \log \left( \prod_{j=1}^{n} f(x_j; \theta) \right) = \sum_{j=1}^{n} \log \left( f(x_j; \theta) \right), \quad \theta \in \Theta,
$$

where $n$ is the sample size, $\theta$ is a vector of parameters to be estimated, and $x_j = (x_{j1}, \ldots, x_{jk})'$ is the $j$th observation for the random vector $X = (X_1, \ldots, X_k)'$, respectively. The maximizer $\hat{\theta}$ of the log-likelihood function given in (9), namely,

$$
\hat{\theta} = \arg \max_{\theta \in \Theta} \mathcal{L}(\theta | x_1, \ldots, x_n),
$$

is obtained using an appropriate optimization method. The parameter space $\Theta$ in each case must be appropriately constrained, and these constraints must be taken into account during the optimization process. We have thus made use of three R stats functions, namely, `optim()`, `constrOptim()`, and `optimize()` in this work. The functionality of these optimization functions is described in Table 7.

| Function       | Number of parameters | Usage                      |
|----------------|----------------------|----------------------------|
| optim()        | Multiple             | General-purpose Optimization|
| constrOptim()  | Multiple             | Optimization with linear constraints |
| optimize()     | Single               | One Dimensional Optimization|

Table 7: Use of optimization functions in R stats.

By default, all these functions perform the task of minimization of a function. To maximize (9), we only need to add argument `control = list(fnscale = -1)` in functions `optim()` and `constrOptim()`, and set `maximum = TRUE` in function `optimize()`.

For example, for the multivariate Lomax distribution, we define the log-likelihood function `loglik.lomax()` by using the following code:

```r
loglik.lomax <- function(data, par) {
  ll <- sum(dmvlomax(data, parm1 = par[1], parm2 = par[-1], log = TRUE))
  ll
}
```

The R stats function `constrOptim()` is chosen to obtain the maximizer of the log-likelihood function (or equivalently, `loglik.lomax()`). The linear constraints imposed on the parameters $a, \theta_1$, and $\theta_2$ are $a > 0, \theta_1 > 0$, and $\theta_2 > 0$. In matrix notation, it is,

$$
U \theta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ \theta_1 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} a \\ \theta_1 \\ \theta_2 \end{pmatrix} > \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = C.
$$

Thus, in the code that follows, the constraint matrix $u_i$ is set as an identity matrix $I_3$ by using the function `diag(3)`, and constraint vector $c_i$ is set as a zero vector `rep(0, 3)`. For our illustration, let the
data be the one simulated by the methods described earlier, each with \( n = 300 \). It is done by

```r
set.seed(1)
bvtlomax <- rmvlomax(n = 300, parm1 = 5, parm2 = c(0.5, 1))
```

The starting initial values for \( a_1, \theta_1, \) and \( \theta_2 \) are all set as 10 by assigning \( \text{rep}(10, 3) \) to the argument theta in function \( \text{constrOptim()} \). The gradient argument grad is optional, and we have chosen this to be \( \text{NULL} \).

```r
> est = constrOptim(theta = rep(10, 3), f = loglik.lomax, grad = NULL, data = bvtlomax,
+        ui = diag(3), ci = rep(0, 3), control = list(fnscale = -1))
> est$convergence
[1] 0
> est$par
[1] 0
> est$convergence

The output consists of two important pieces of information, (i) whether convergence is successfully achieved (\( \text{est$convergence} = 0 \)) or not (\( \text{est$convergence} = 1 \)) and (ii) the values of the final maximum likelihood estimates. In our illustration, the convergence is successfully achieved, and we have \( \hat{a} = 5.0555691 \) and \( (\hat{\theta}_1, \hat{\theta}_2) = (0.4468724, 0.9036692) \).

We summarize the optimization methods, constraints, and the resulting outputs for all the bivariate distributions (except for bivariate \( F \) distribution) in Table 8. The detailed illustrations and codes for the remaining distributions are included in the digital complement. Observe that these estimates are reasonably close to the true parameter values, thereby confirming that the program is functioning as it is expected to.

| Multivariate Distribution | Parameters                  | Optimization Method | Constraints                          | Estimated Parameters                  |
|---------------------------|-----------------------------|---------------------|--------------------------------------|---------------------------------------|
| Lomax                     | \( a = 5 \)                | \( \theta = (0.5,1)' \) | \( U = I_3 \)                        | \( \hat{a} = 5.05556 \)              |
|                           |                             | \( \theta = (0.5,2)' \) | \( C = (0,0,0)' \)                   | \( \hat{\theta} = (0.44687,0.90366)' \) |
| Mardia Type I             | \( a = 5 \)                | \( \theta = (0.5,2)' \) | \( U = I_3 \)                        | \( \hat{a} = 4.63862 \)              |
|                           |                             | \( \theta = (1,1.5)' \) | \( C = (0,0,0,0,0)' \)              | \( \hat{\theta} = (0.99573,1.55976)' \) |
| Logistic                  | \( \mu = (0.5,1)' \)       | \( \sigma = (1,1.5)' \) | \( \text{N/A} \)                     | \( \hat{\mu} = (0.39199,0.89731)' \) |
|                           |                             |                     |                                      | \( \hat{\sigma} = (0.97573,1.55976)' \) |
| Burr                      | \( a = 3 \)                | \( \theta = (1)' \)  | \( U = I_3 \)                        | \( \hat{a} = 3.91498 \)              |
|                           |                             | \( \sigma = (2.5)' \) | \( C = (0,0,0,0,0)' \)              | \( \hat{\sigma} = (0.64889,2.06858)' \) |
| Cook-Johnson’s Uniform    | \( a = 0.3 \)              | \( \text{optimize()} \) | \( a > 0 \)                          | \( \hat{a} = 0.31064 \)              |
| Generalized               | \( a = 5 \)                | \( \theta = (0.5,1)' \) | \( U = I_3 \)                        | \( \hat{a} = 3.91869 \)              |
|                           |                             | \( \theta = (2,4)' \) | \( C = (0,0,0,0,0)' \)              | \( \hat{\theta} = (0.54674,1.79126)' \) |
| Inverted Beta             | \( a = 4 \)                | \( \theta = (2,6)' \) | \( U = I_3 \)                        | \( \hat{a} = 3.67153 \)              |

|                           |                             | \( C = (0,0,0)' \) | \( I = (1.70310,5.21736)' \)         | \( I = (1.82450,5.46465)' \)          |

Table 8: Comparison between True and Estimated Values of Parameters for each of the Distributions (except for bivariate \( F \) distribution).

### Two applications

In this last section, we give two brief applications, which not only demonstrate the use but also confirm the accuracy and verify the correctness of our work.

#### Generating data from the nonelliptical symmetric distributions with univariate normal marginals

Cook-Johnson\’s multivariate uniform distribution is a family of distributions that can be used for modeling nonelliptical symmetric data. Further, in view of uniform distribution for marginal, it has been as one of the useful choices for modeling through copula (in fact, Cook-Johnson\’s uniform distribution is indeed a Clayton copula (Nelsen, 2006)). The value of parameter \( a \) impacts the common
correlation coefficient $\rho$ among variates in that $\rho \to 0$ as $a \to \infty$, and $\rho \to 1$ as $a \to 0$ (Cook and Johnson, 1981). An interesting application of Cook-Johnson’s multivariate uniform distribution is to obtain new joint distributions by marginal transformation. Specifically, we consider the problem of generating random numbers from a multivariate distribution that is not elliptically symmetric but has univariate normal marginals. Let $U_j's, i=1,2,$ be two random variables corresponding to the Cook-Johnson’s bivariate uniform distribution. The following code yields the pairs of random numbers, each having the normal marginals by the transformation $X_i = \Phi^{-1}(U_i)$, where $\Phi^{-1}(\cdot)$ is the quantile function of a standard normal distribution. Clearly, the joint distribution of $X_1$ and $X_2$ is not bivariate normal. To begin with, the parameter $a$ is taken to be $a = 2$.

```r
set.seed(1)
biv.unif <- rmvunif(8000, parm = 2, dim = 2)
biv.norm <- as.data.frame(apply(biv.unif, 2, qnorm))
```

The sample correlation coefficient $\rho$ of data set `biv.norm` is computed by the following code.

```r
> cor(biv.norm$V1, biv.norm$V2)
[1] 0.3180119
```

We create a bivariate scatter plot using the function `ggplot()` in package `ggplot2` (Wickham, 2016) for data set `biv.norm`. This is shown in Figure 3 (a).

```r
ggplot(biv.norm, aes(x = V1, y = V2)) + xlim(c(-4, 4)) + ylim(c(-4, 4)) + xlab("X1") + ylab("X2") + geom_point()
```

To assess the behavior as a function of $a$, we now decrease the parameter $a$ to 1.0, 0.5, 0.1 resulting in higher correlations $\rho$ (= 0.51, 0.68, 0.93, respectively) between the two variates. The bivariate scatter plots for the four cases that is, when $a = 2.0, 1.0, 0.5, 0.1$ are shown in Figure 3 (a)-(d). It is easy to observe that the generated bivariate data have nonelliptical yet, symmetric contours.

![Figure 3](image)

**Figure 3**: The scatterplots of nonelliptical symmetric normal data generated from transformed Cook-Johnson’s uniform random numbers with $a = 2, 1, 0.5$ and $0.1$. 

The R Journal Vol. 13/2, December 2021 ISSN 2073-4859
Creating tables for simultaneous MANOVA hypothesis tests

Multivariate $F$ distribution arises naturally as the distribution of test statistics in several testing problems in simultaneous MANOVA. Let $s_1^2, \ldots, s_k^2$ be $k$ independent sums of squares, and $s_k^2$ be the sum of squares due to error in the classical ANOVA model. Also, let $H_1, \ldots, H_k$ be certain individual linear hypotheses with the corresponding sum of squares $s_1^2, \ldots, s_k^2$. Assume that under $H_1, \ldots, H_k$, respectively, the sums of squares $s_1^2, \ldots, s_k^2$ are all $\chi^2$ random variables each with $n$ degrees of freedom and $s_k^2$ is a $\chi^2$ random variable with $m$ degrees of freedom and is independent of $s_1^2, \ldots, s_k^2$. Armitage and Krishnaiah (1964) defined the critical values $F_k$ at a level of significance for simultaneously testing hypotheses $H_1, \ldots, H_k$ by the probability statement,

$$P \left[ \frac{\max (s_1^2, \ldots, s_k^2)}{n s_k^2} \leq F_k \bigg| \bigcap_{i=1}^{k} H_i \right] = P \left[ \frac{m s_i^2}{n s_k^2} \leq F_k, i = 1, \ldots, k \bigg| \bigcap_{i=1}^{k} H_i \right] = 1 - \alpha.$$

The quantities $(m s_i^2)/(n s_k^2), i = 1, \ldots, k$ jointly follow multivariate $F$ distribution if the overall null hypothesis $H_0 = \bigcap_{i=1}^{k} H_i$ is true. In this case, the critical value $F_k$ can be readily computed using the equicoordinate quantile function `qmvf()` by setting the argument corresponding to $k + 1$ values of the degrees of freedom as $df = c(m, n, \ldots, n)$. The following code gives $F_{0.05} = 9.551505$ for the bivariate $F$ case when $m = 5$ and $n = 1$ with default algorithm by using adaptive multiple integration over unit cube (argument “numerical”). With Monte Carlo algorithm (argument “MC” with nsim = 1,000,000), we obtain $F_{0.05} = 9.550944$. Note that the Monte Carlo method is seed dependent, so the output from different runs may slightly differ from each other.

```r
> qmvf(0.95, df = c(5, 1, 1))
[1] 9.551505
> qmvf(0.95, df = c(5, 1, 1), algorithm = "MC")
[1] 9.550944
```

For further demonstration and also to further affirm our trust in the calculations, we compare the output of quantile function `qmvf()` using both adaptive multivariate integration and Monte Carlo methods with the values given in Armitage and Krishnaiah (1964). These three calculations are reported in Table 9 for a few choices of $m$ and $n$. The agreement among the three columns shows that the package `NonNorMvtDist` provides a convenient way to obtain percentage points for the hypothesis testing problems considered by Armitage and Krishnaiah (1964) and Krishnaiah (1965). Clearly, unlike the tables in Armitage and Krishnaiah, the choices of $\alpha$ and degrees of freedom are not restricted, and in that sense, our package is very comprehensive and exhaustive in this respect.

| $\alpha$ | $df (m, n, n)$ | `qmvf()` Output (algorithm = "numerical") | `qmvf()` Output (algorithm = "MC") | Tabulated Values in Armitage and Krishnaiah (1964, pp. 33-42) |
|----------|----------------|------------------------------------------|--------------------------------|-----------------------------------------------------------|
| 0.05     | (5, 1, 1)      | 9.551505                                 | 9.550944                       | 9.55                                                      |
|          | (5, 2, 2)      | 7.879999                                 | 7.881698                       | 7.88                                                      |
|          | (5, 3, 3)      | 7.136473                                 | 7.165361                       | 7.14                                                      |
|          | (5, 4, 4)      | 6.702224                                 | 6.715759                       | 6.70                                                      |
|          | (5, 5, 5)      | 6.412372                                 | 6.399167                       | 6.41                                                      |
|          | (10, 6, 6)     | 3.899335                                 | 3.898442                       | 3.90                                                      |
|          | (10, 7, 7)     | 3.768494                                 | 3.769715                       | 3.77                                                      |
|          | (10, 8, 8)     | 3.665646                                 | 3.661366                       | 3.67                                                      |
|          | (10, 9, 9)     | 3.582271                                 | 3.583923                       | 3.58                                                      |
|          | (10, 10, 10)   | 3.513163                                 | 3.514059                       | 3.51                                                      |

Table 9: Comparison between the output of `qmvf()` with the values given in Armitage and Krishnaiah (1964).

Concluding remarks

We have developed a new R package, `NonNorMvtDist`, for generating multivariate random numbers from Lomax (Pareto type II), generalized Lomax, Mardia’s Pareto type I, logistic, Burr, Cook-Johnson’s uniform, $F$, and inverted beta distributions. Detailed examples of each distribution are given to illustrate data simulation, probability calculations, and statistical modeling.
The fact that nonnormal and skewed multivariate distributions are common in the real world but are rarely pursued for analysis due to the lack of ready-to-use computational support underscores the importance of this package. Possibilities of the use of these distributions are practically limitless and yet unforeseen in a variety of areas, starting from the biomedical sciences, reliability, and engineering as well as in statistical finance in the contexts of volatility estimation. Simulations, probability calculations, as well as calculations of quantiles, and the maximum likelihood estimation of parameters are the natural first set of computations in such studies. We have addressed all of these in this work.

The calculations of probabilities of hypercubes (for example, of \( P(a_1 < X_1 < b_1, a_2 < X_2 < b_2, a_3 < X_3 < b_3) \)) can be easily implemented by appropriately combining several \( cdf \) calculations. Alternatively, our codes for \( pmv*() \) can be suitably modified for this purpose. The probability density surface plots for any bivariate marginal can be easily constructed since, for the multivariate Lomax distribution, the marginal distributions of any subset of random variables also follow the multivariate Lomax distribution in the appropriate dimension. Further, our work provides a way to generate data from, probability calculations for, as well as modeling for, the data which are marginally distributed as normal but jointly are not.

Acknowledgements

Content of this article constitutes a part of the PhD dissertation of Zhixin Lun at Oakland University. Financial research support from Oakland University during the summer of Year 2019 is acknowledged. We would like to thank the anonymous referees for their helpful comments and remarks to improve both package and this paper.

Bibliography

J. V. Armitage and P. R. Krishnaiah. *Tables for Studentized Largest Chi-Square and Their Application*. Report ARL 64-188, Aerospace Research Laboratories Wright-Patterson Air Force Base, Ohio, 1964. [p429, 438]

N. Balakrishnan and C.-D. Lai. *Continuous Bivariate Distributions*. Springer-Verlag New York, 2nd edition, 2009. ISBN 978-0-387-09614-8. URL https://doi.org/10.1007/b101765. [p428]

R. E. Cook and M. E. Johnson. A family of distributions for modeling non-elliptically symmetric multivariate data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 43(2):210–218, 1981. URL www.jstor.org/stable/2984851. [p437]

S. Ghosh and S. Henderson. Properties of the norta method in higher dimensions. *Proceedings of the 2002 Winter Simulation Conference*, 1:263–269, 2002. URL https://doi.org/10.1109/WSC.2002.1172894. [p432]

H. Joe. *Multivariate Models and Dependence Concepts*. Chapman and Hall, London, 1997. URL https://doi.org/10.1201/b13158. [p430]

N. L. Johnson and S. Kotz. *Distribution in Statistics: Continuous Multivariate Distributions*. John Wiley & Sons, New York, 1972. [p429]

P. R. Krishnaiah. On the simultaneous anova and manova tests. *Annals of the Institute of Statistical Mathematics*, 17(1):35–53, 1965. URL https://doi.org/10.1007/BF02868151. [p429, 438]

D. V. Lindley and N. D. Singpurwalla. Multivariate distributions for the life lengths of a system sharing a common environment. *Journal of Applied Probability*, 23:418–431, 1986. URL https://doi.org/10.2307/3214184. [p427]

Z. Lun and R. Khattree. *NonNorMvtDist: Multivariate Lomax (Pareto Type II) and Its Related Distributions*, 2020. URL https://CRAN.R-project.org/package=NonNorMvtDist. R package version 1.0.2. [p427]

B. Narasimhan, S. G. Johnson, T. Hahn, A. Bouvier, and K. Kiêu. *cubature: Adaptive Multivariate Integration over Hypercubes*, 2018. URL https://CRAN.R-project.org/package=cubature. R package version 2.0.3. [p431]

T. K. Nayak. Multivariate lomax distribution: Properties and usefulness in reliability theory. *Journal of Applied Probability*, 24(1):170–177, 1987. URL https://doi.org/10.2307/3214066. [p427, 428, 429]

R. B. Nelsen. *An Introduction to Copulas*. Springer-Verlag New York, 2nd edition, 2006. ISBN 978-0-387-28659-4. URL https://doi.org/10.1007/0-387-28678-0. [p436]
R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019. URL https://www.R-project.org/. [p429]

K. Soetaert. *plot3D: Plotting Multi-Dimensional Data*, 2017. URL https://CRAN.R-project.org/package=plot3D. R package version 1.1.1. [p433]

H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org. [p437]

T. Yee. *VGAM: Vector Generalized Linear and Additive Models*, 2019. URL https://CRAN.R-project.org/package=VGAM. R package version 1.1-2. [p427]

Zhixin Lun
Department of Clinical Pharmacy
University of California, San Francisco
San Francisco, CA, USA
ORCID: 0000-0002-8980-1554
zhixin.lun@ucsf.edu

Ravindra Khattree
Department of Mathematics and Statistics
Oakland University
Rochester, MI, USA
ORCID: 0000-0002-9305-2365
khattree@oakland.edu