

trunmnt: An R package for calculating moments in a truncated multivariate normal distribution

Seung-Chun Lee^{1,a}

^aDepartment of Applied Statistics, Hanshin University, Korea

Abstract

The moment calculation in a truncated multivariate normal distribution is a long-standing problem in statistical computation. Recently, Kan and Robotti (2017) developed an algorithm able to calculate all orders of moment under different types of truncation. This result was implemented in an R package **MomTrunc** by Galarza *et al.* (2021); however, it is difficult to use the package in practical statistical problems because the computational burden increases exponentially as the order of the moment or the dimension of the random vector increases. Meanwhile, Lee (2021) presented an efficient numerical method in both accuracy and computational burden using Gauss-Hermit quadrature. This article introduces **trunmnt** implementation of Lee's work as an R package. The Package is believed to be useful for moment calculations in most practical statistical problems.

Keywords: moment of truncated multivariate normal, MomTrunc, Gauss-Hermite quadrature

1. Introduction

The package **trunmnt** deals with the computation of $E(Y_1^{\kappa_1} \cdots Y_n^{\kappa_n} | a_i < Y_i < b_i, i = 1, \dots, n)$, where κ_i 's are nonnegative integers and $\mathbf{Y} = (Y_1, \dots, Y_n)' \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. That is, it computes arbitrary p^{th} ($p = \sum_{i=1}^n \kappa_i$) order of product moments in a truncated multivariate normal distribution with Y_i truncated at the lower limit a_i and the upper limit b_i . In this truncation, some or all of the a_i can be $-\infty$ and some or all of the b_i can be ∞ . When all the b_i 's are ∞ , the random vector \mathbf{Y} ($> \mathbf{a}$) is called lower truncated, whereas \mathbf{Y} ($< \mathbf{b}$) is called upper truncated when all the a_i 's are $-\infty$. The other type is double truncation $\mathbf{a} < \mathbf{Y} < \mathbf{b}$, which can have both lower and upper truncation points.

A difficulty in calculating the moments of a multivariate truncated normal distribution is that the marginal distributions from a truncated normal distribution are not truncated normal. Numerous studies have been performed for the calculations under different conditions, including the type of truncation (one-sided or double-sided) and number of variables (univariate, bivariate, or multivariate). However, as of this writing, there are two R packages related to the moment calculation. The first is based on the work of Manjunath and Wilhelm (2012). They provided explicit representations for the mean and variance of a multivariate normal distribution with arbitrary double truncation using the moment generating function. These results were implemented in **tmvtnorm** (Wilhelm and Manjunath, 2010). Therefore, the **tmvtnorm** can calculate up to the second-order moments. However, it is worth noting that likelihood-based inferences for some limited dependent variable mixed-effects models require fourth-order moments. Another package is **MomTrunc** (Galarza *et al.*, 2021), which uses

This work was supported by Hanshin University research grant.

¹ Department of Applied Statistics, Hanshin University, 137 Hanshindae-gil, Osan, Gyunggi-Do 18101, Korea.
E-mail: seung@hs.ac.kr

Published 30 November 2021 / journal homepage: <http://csam.or.kr>

© 2021 The Korean Statistical Society, and Korean International Statistical Society. All rights reserved.

the iterative relationship developed by Kan and Robotti (2017) for integrals involving the density of multivariate normal distributions. In general, algorithms that use recursive relationships are known to be slow. In fact, the **MomTrunc** is quite slow to compute the moment even for moderately large n . On the other hand, the algorithm proposed by Lee (2021) does not depend heavily on n . The effectiveness of his algorithm depends primarily on the covariance structure of the variables rather than the dimensions of the variables.

One advantage of the **tmvtnorm** and the **MomTrunc** is that they can compute moments without making assumptions about the variance-covariance matrix of the parent multivariate normal distribution. Note that, the variance-covariance matrix of an n -dimensional random vector consists of $n(n+1)/2$ elements. In statistical models, the variance-covariance matrix is generally unknown and it is impossible to estimate the matrix without some assumptions about the its structure. The **trunmnt** computes the moment under some variance-covariance matrices which can be found in most statistical models.

2. Computation of moments

The general form of the model equation for a mixed model is,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \boldsymbol{\epsilon},$$

where $\mathbf{X}_{n \times p}$ and $\mathbf{Z}_{n \times q}$ are the design matrices corresponding to $\boldsymbol{\beta}$ and \mathbf{u} representing fixed and random effects, respectively, and $\boldsymbol{\epsilon}$ is the vector of errors. Customary, it is assumed that the random effects \mathbf{u} follows a multivariate normal distribution with mean $\mathbf{0}$ and symmetric positive definite variance-covariance matrix \mathbf{D} . As usual, the distribution of $\boldsymbol{\epsilon}$ is assumed to be a multivariate normal with mean $\mathbf{0}$ and variance-covariance matrix $\sigma_\epsilon^2 \mathbf{I}$, but for more flexibility, it is assumed that the error terms are independent, but do not have equal variance. That is, it is assumed $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{E})$ where \mathbf{E} is a diagonal matrix with elements $(\sigma_1^2, \dots, \sigma_n^2)$. Then, writing $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$, we have,

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{Z}\mathbf{D}\mathbf{Z}' + \mathbf{E}).$$

The **trunmnt** concerns the calculation of the truncated moment under these assumptions.

Let $m_{(a,b)}^k(\boldsymbol{\mu}, \sigma^2) = E(Y^k | a < Y < b)$ and $\Phi(a, b; \boldsymbol{\mu}, \sigma^2) = \Pr[a < Y < b]$ where $Y \sim N(\boldsymbol{\mu}, \sigma^2)$. Also, let $\phi_n(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denote the joint density of a n -variate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Using the Theorem 1 of Lee (2021), it can be shown that $E(Y_1^{k_1} \cdots Y_n^{k_n} | a_i < Y_i < b_i, i = 1, \dots, n)$ is equal to

$$\frac{1}{\Pr[\mathbf{a} < \mathbf{Y} < \mathbf{b}]} \int_{-\infty}^{\infty} \left(\prod_{i=1}^n m_{(a_i, b_i)}^{k_i}(\tilde{\boldsymbol{\mu}}_i, \sigma_i^2) \Phi(a_i, b_i; \tilde{\boldsymbol{\mu}}_i, \sigma_i^2) \right) \phi_q(\mathbf{u}; \mathbf{0}, \mathbf{D}) \mathbf{d}\mathbf{u}, \quad (2.1)$$

where $\tilde{\boldsymbol{\mu}}_i = \boldsymbol{\mu}_i + \mathbf{z}'_i \mathbf{u}$, and \mathbf{z}'_i is the i^{th} row of \mathbf{Z} . Equation (2.1) makes a n -dimensional integral problem into a q -dimensional integral, where q is much less than n in most cases. In practice, q is 1 or 2 in many statistical models, and these low-dimensional integrals can be evaluated efficiently with the Gauss-Hermit quadrature method. The **trunmnt** uses the multivariate Gaussian-Hermite quadrature described in Jäckel (2005). Also here is one result of the theorem.

$$\Pr[\mathbf{a} < \mathbf{Y} < \mathbf{b}] = \int_{-\infty}^{\infty} \left(\prod_{i=1}^n \Phi(a_i, b_i; \tilde{\boldsymbol{\mu}}_i, \sigma_i^2) \right) \phi_q(\mathbf{u}; \mathbf{0}, \mathbf{D}) \mathbf{d}\mathbf{u}. \quad (2.2)$$

Table 1: Summary of the functions in the **trunmnt** package

Function	Description
<code>meanvar</code>	Calculate the mean and variance
<code>mtrunmnt</code>	Create a <code>mtrunmnt</code> object with given parameters
<code>prodmnt</code>	Calculate the product moment
<code>probntrun</code>	Calculate the probability of not being truncated
<code>utrunmnt</code>	Calculate the moment of univariate truncated normal distribution

The evaluation of $\Pr[\mathbf{a} < \mathbf{Y} < \mathbf{b}]$ is another n -dimensional integral problem that is analytically impossible. This quantity is also required by the **tmvtnorm** and **MomTrunc**, both of which rely on **mvtnorm** (Genz *et al.*, 2021), an R package for computing multivariate normal and t probabilities, quantiles, random deviates and densities to compute this fraction after truncation. Note that the **mvtnorm** uses the randomized quasi Monte Carlo procedure (Genz, 1992, 1993), whereas the **trunmnt** evaluates equation (2.2) with the multivariate Gauss-Hermite quadrature. The quasi Monte Carlo integration is known to be better than quadrature methods for high-dimensional integral problems, but its convergence properties are very slow. It is also known that quadrature methods are very fast and efficient for low-dimensional integrations. Since the quantity $m_{(a_i, b_i)}^{k_i}(\tilde{\mu}_i, \sigma_i^2)$, moment of a truncated univariate normal random variable can be obtained by an explicit formula (Burkardt, 2014), it is thought that equation (2.1) can be successfully approximated by the Gauss-Hermite quadrature. The formula is implemented in `utrunmnt()` function.

3. The **trunmnt** package

3.1. Structure and functionality

The **trunmnt** package consists of five main functions for calculating the moment of truncated multivariate normal distributions. Most of the functional parts of **trunmnt** were written by RccpAramdillo (Eddelbuettel and Sanderson, 2014) to speed up computation. Table 1 provides a brief summary of the available functions. The functionality will be demonstrated with numerical examples.

The moment calculation for a truncated univariate normal distribution is implemented in the `utrunmnt()` function:

```
utrunmnt(kappa, mu = 0., sd = 1., lower = -Inf, upper = Inf)
```

Here, the `kappa` argument specifies the order of moment and must be provided as a nonnegative integer. The other arguments are clear by name. For example, if a normal distribution with mean 5 and standard deviation 1 is upper truncated at 10, the fourth moment is computed with the command:

```
> utrunmnt(4, mu = 5, sd = 1, upper = 10)
[1] 777.9971
```

The following R (R Core Team, 2019) code mimics the situation of a mixed-effects probit regression model. It generate a 5-dimensional latent variable from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ is a vector of 5 equally spaced values from -1 to 1, and $\boldsymbol{\Sigma} = \mathbf{I}_5 + 2\mathbf{1}_5\mathbf{1}'_5$. It then performs a transformation on the latent variable to make binary observations. That is, if the latent variable is greater than or equal to 0, the observation will be 1, indicating that the latent variable is lower truncated at the lower bound of zero. On the other hand, if the latent variable is less than zero, the binary observation will be zero, meaning that the latent variable is upper truncated at the upper zero.

```
set.seed(123)
```

```

sigma2e <- 1
sigma2a <- 2
n <- 5
mu <- seq(-1,1, length.out = n)
y <- mu + rnorm(1, sd = sqrt(sigma2a)) + rnorm(n, sd = sqrt(sigma2e))
y <- ifelse( y > 0, 1, 0)
a <- rep(-Inf, n)
b <- rep(Inf, n)
a[y >= 0] <- 0
b[y < 0] <- 0

```

To calculate the moment, you must first create an `mtrunmnt` object using `mtrunmnt()`:

```

mtrunmnt(mu, Sigma2 = 1, lower = rep(-Inf, length(mu)), upper = rep(Inf, length(mu)),
        Z = matrix(rep(1, length(mu)), ncol = 1), D = matrix(1, ncol = 1, nrow = 1),
        nGH = 32)

```

The `Sigma2` argument specifies the variances of the error terms and must be specified as a scalar, or a vector of length equal to `mu`. Given a scalar, the variances are assumed to be equal. The design matrix and variance-covariance matrix of the random effect are passed in the `Z` and `D` arguments, respectively. By default, the `Z` argument is provided as a vector of ones of the same length as `mu`, and should be changed accordingly when necessary. If the random components are independent and the variances are the same, you can pass the variance as a scalar to `D`. The `nGH` argument specifying the number of quadrature points is set to 32. The other arguments are self-explanatory.

The previously generated artificial data can be passed to the `mtrunmnt()` to create a `mtrunmnt` object. Once you have created the object, pass it to `probntrun()`, `prodmnt()` and `meanvar()` as follows:

```

> tobj <- mtrunmnt(mu, Sigma2 = sigma2e, lower = a, upper = b, D = sigma2a)
> probntrun(tobj)
[1] 0.01453587
> prodmnt(tobj, c(2,2,0,0,0))
[1] 1.612259
> c(prodmnt(tobj, c(4,0,0,0,0)), prodmnt(tobj, c(0,4,0,0,0)))
[1] 21.433824 1.482977
>
> meanvar(tobj)
$tmmean
[1] -1.4996487 0.5993896 -0.9354669 -0.7448227 1.1792822

$ tvar
      [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 0.84907923 0.05062909 0.09792397 0.07408044 0.11730096
[2,] 0.05062909 0.26332061 0.03141038 0.02362801 0.04374713
[3,] 0.09792397 0.03141038 0.49954827 0.04749122 0.07292303
[4,] 0.07408044 0.02362801 0.04749122 0.36549384 0.05487201
[5,] 0.11730096 0.04374713 0.07292303 0.05487201 0.66073805

```

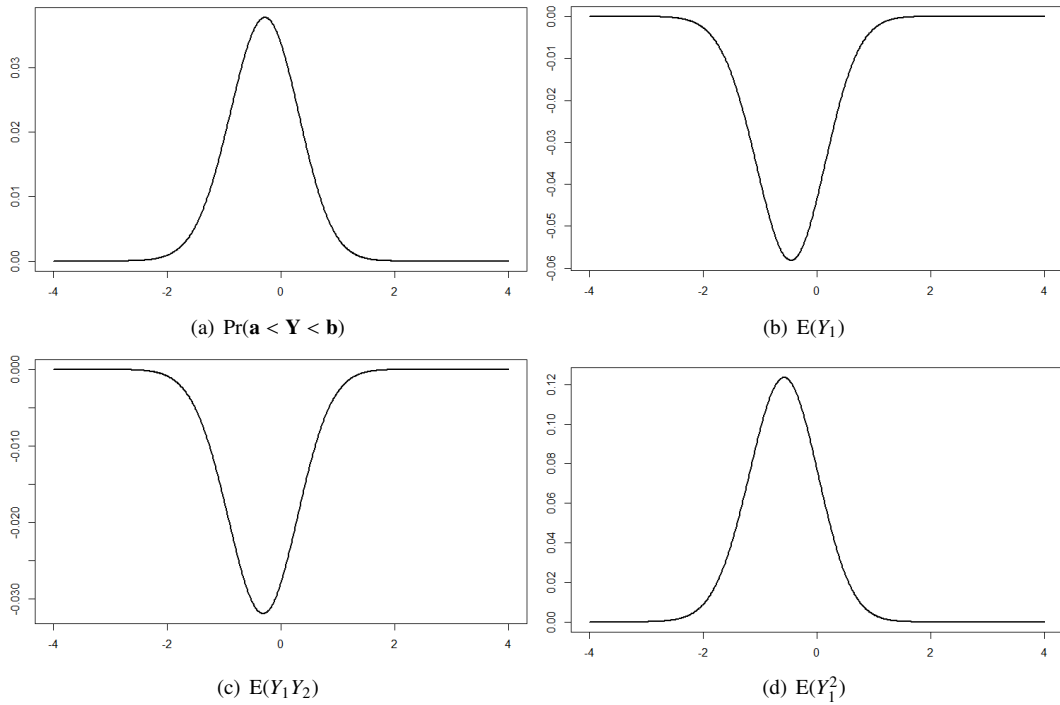


Figure 1: Functions to be integrated by the Gauss-Hermite quadrature.

The `probntrun()` and `meanvar()` have only one argument specifying a `trunmnt` object, and compute $\Pr[\mathbf{a} < \mathbf{Y} < \mathbf{b}]$ of equation (2.2), and the mean and variance-covariance matrix, respectively. The `prodmnt()` computes the product moment of equation (2.1) and takes two arguments: a `mtrunmnt` object and the order of moment. The last argument must be specified as a vector of nonnegative integers of the same length as μ . For example, the third command computes $E(Y_1^2 Y_2^2)$.

As we have seen, the `trunmnt` can compute several parameters of a truncated multivariate normal distribution. However, since the parameters cannot be calculated analytically, we do not know the actual value of each parameter, so it is not clear how close the result of `trunmnt` was to the true value.

It is well known that the Gaussian-Hermite quadrature method gives an exact integration if integral function is a polynomial of order less than or equal to $2m - 1$, where m represents the number of quadrature points. The function in parentheses of equation (2.1) is a re-expression of,

$$f(\mathbf{u}) = \prod_{i=1}^n \int_{a_i}^{b_i} y^{k_i} \phi_1(y; \tilde{\mu}_i, \sigma_i^2) dy.$$

Thus, it is easy to see that the functions enclosed in parentheses in equations (2.1) and (2.2) are infinitely differentiable with respect to \mathbf{u} almost everywhere. That is, the function to be integrated by the Gauss-Hermite method is smooth, and can be approximated sufficiently as a polynomial. Figure 1 shows these functions to compute $\Pr(\mathbf{a} < \mathbf{Y} < \mathbf{b})$, $E(Y_1)$, $E(Y_1 Y_2)$, and $E(Y_1^2)$. As a function of \mathbf{u} , we can observe that these functions do not fluctuate too much. Therefore, we can expect that the Gauss-Hermite works well with a moderate number of quadrature points.

We want to verify the accuracy of `trunmnt` via the `tmvtnorm` and the `MomTrunc`. As mentioned

earlier, both packages use deterministic algorithms, but calculate the fraction after truncation given in equation (2.2) using a R package **mvtnorm** (Genz *et al.*, 2021). Since the computation of **mvtnorm** is performed by the random quasi Monte Carlo procedure proposed by Genz (1992, 1993), the probability estimates change slightly from run to run despite the same conditions. As a result, the output of the two packages will not be the same each time you run them. Nevertheless, it was observed that the mean and variance calculated with `meanvar()` match those of the two packages in each run, mostly to 4 decimal places. A match of this degree is considered practically meaningful and concludes that **trunmnt** is indeed usable.

3.2. Numerical example

The Gauss-Hermit quadrature method is known to be efficient for low-dimensional integration. Therefore, the preceding example performed under the best conditions for **trunmnt**. In fact, it works best with random-effects one-way layout models because it's essentially a one-dimensional integration problem because of the independence.

Consider an artificial example. Suppose \mathbf{y}_1 and \mathbf{y}_2 are independent and have the same truncated multivariate normal distribution as in the previous example in Section 3.1. Then, the parent distribution of $\mathbf{y} = (\mathbf{y}'_1, \mathbf{y}'_2)'$ is a multivariate normal with mean $\mathbf{1}_2 \otimes \boldsymbol{\mu}$ and variance-covariance $\mathbf{I}_2 \otimes \boldsymbol{\Sigma}$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are given previously, and \otimes denotes the kronecker product. The following R code computes the mean and variance-covariance of \mathbf{y} :

```
Z <- model.matrix( ~ factor(rep(c(0,1), each = 5)) - 1)
D <- diag(c(sigma2a, sigma2a))
mu2 <- c(mu, mu)
a2 <- c(a, a)
b2 <- c(b, b)
tobj2 <- mtrunmnt(mu2, Sigma2=sigma2e, lower = a2, upper = b2, Z = Z, D = D)
meanvar(tobj2)
```

We do not give the means and variance-covariance matrices of \mathbf{y}_1 and \mathbf{y}_2 here, but the **trunmnt** computes them with exactly the same values as in the previous example. The covariance matrix of \mathbf{y}_1 and \mathbf{y}_2 is shown below:

```
$tvar
      [,1]      [,2]      [,3]      [,4]      [,5]
[6,] 2.664535e-15 -3.330669e-16  8.881784e-16  1.110223e-15 -2.220446e-15
[7,] -1.110223e-15  4.996004e-16 -3.330669e-16 -3.885781e-16  5.551115e-16
[8,]  1.998401e-15 -4.440892e-16  1.110223e-16  3.330669e-16 -8.881784e-16
[9,]  1.998401e-15 -2.775558e-16  8.881784e-16  4.440892e-16 -8.881784e-16
[10,] -1.776357e-15  2.220446e-16 -6.661338e-16 -2.220446e-16  8.881784e-16
```

It's worth noting that the **trunmnt** provides very close to true values for the covariances. Since the calculation of these covariances involves two-dimensional integrals, it can be inferred that Gaussian-Hermite quadrature works well, at least for two-dimensional integrals. Note also that the **mvtnorm** and **MomTrunc** did not provide such degree of approximation for the covariances. Indeed, we observe that they can approximate the true value up to 2 or 3 decimal place. This magnitude of error can be problematic in real world problems. Another advantage of **trunmnt** is that it can compute moments much faster than the other two packages.

4. Summary

This paper introduces the **trummnt** package, an R implementation for calculating the product moment of a truncated multivariate normal distribution. Unlike the **tmvtnorm** or **MomTrunc** packages, it only works with certain variance-covariance structures. However, it is believed that the package is applicable to the truncated multivariate distributions encountered in most statistical models, and in some cases can give more accurate values than others. More importantly, the **trummnt** is much faster than the other two. For example, on our PC, the **trummnt** was approximately 10x and 74x faster than the **tmvtnorm** and the **MomTrunc**, respectively, when computing the mean and variance-covariance matrices of the example given in Section 3.1. The speed advantage is noticeable when the dimensions of the variable are large. Specifically, in the example in Section 3.2, it is about 200x and 1340x faster. This slow computational speed can be a major impediment to using both packages in real-world problems. The computation of higher order moment, which may be required by some applications, is more problematic in both packages in that **tmvtnorm** lacks functionality and **MomTrunc** is unusably slow. Considering these points, the **trummnt** can be considered as an alternative to both packages in certain problems.

References

- Burkardt J (2014). The truncated normal distribution, *Online document*, Available from: <http://people.sc.fsu.edu/jburkardt/presentations/truncatednormal.pdf>
- Eddelbuettel D and Sanderson C (2014). RcppArmadillo: Accelerating R with high-performance C++ linear algebra, *Computational Statistics & Data Analysis*, **71**, 1054–1063.
- Galarza CE, Kan R, and Lachos VH (2021). MomTrunc: Moments of folded and doubly truncated multivariate distributions, *Online document*, Available from: <https://cran.r-project.org/web/packages/MomTrunc/MomTrunc.pdf>
- Genz A (1992). Numerical computation of multivariate normal probabilities, *Journal of Computational and Graphical Statistics*, **1**, 141–150.
- Genz A (1993). Comparison of methods for the computation of multivariate normal probabilities, *Computing Science and Statistics*, **25**, 400–405.
- Genz *et al.* (2021). mvtnorm: Multivariate Normal and t Distributions, R package version 1.1, Available from: <https://cran.r-project.org/web/packages/mvtnorm/mvtnorm.pdf>
- Jäckel P (2005). A note on multivariate Gauss-Hermite quadrature, *London: ABN-Amro*, Available from: <http://www.pjaeckel.webspace.virginmedia.com/ANoteOnMultivariateGaussHermiteQuadrature.pdf>
- Kan R and Robotti C (2017). On moments of folded and truncated multivariate normal distributions, *Journal of Computational and Graphical Statistics*, **26**, 930–934.
- Lee SC (2021). moments calculation for truncated multivariate normal in nonlinear generalized mixed models, *Communications for Statistical Applications and Methods*, **27**, 377–383.
- Manjunath BG and Wilhelm S (2012). Moments calculation for the doubly truncated multivariate normal density, *SSRN Electronic Journal*, arXiv preprint arXiv:1206.5387.
- R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Wilhelm S and Manjunath BG (2010). tmvtnorm: Truncated multivariate normal and Student t distribution, *The R Journal*, **2**.