# On Sampling from the Multivariate *t* Distribution

*by Marius Hofert*

**Abstract** The multivariate normal and the multivariate *t* distributions belong to the most widely used multivariate distributions in statistics, quantitative risk management, and insurance. In contrast to the multivariate normal distribution, the parameterization of the multivariate *t* distribution does not correspond to its moments. This, paired with a non-standard implementation in the R package **mvtnorm**, provides traps for working with the multivariate *t* distribution. In this paper, common traps are clarified and corresponding recent changes to **mvtnorm** are presented.

## Introduction

A supposedly simple task in statistics courses and related applications is to generate random variates from a multivariate *t* distribution in R. When teaching such courses, we found several fallacies one might encounter when sampling multivariate *t* distributions with the well-known R package **mvtnorm**; see Genz et al. (2013). These fallacies have recently led to improvements of the package ($\geq$ 0.9-9996) which we present in this paper[1]. To put them in the correct context, we first address the multivariate normal distribution.

## The multivariate normal distribution

The *multivariate normal distribution* can be defined in various ways, one is with its stochastic representation

$$X = \mu + A\mathbf{Z}, \tag{1}$$

where $\mathbf{Z} = (Z_1, \ldots, Z_k)$ is a $k$-dimensional random vector with $Z_i$, $i \in \{1, \ldots, k\}$, being independent standard normal random variables, $A \in \mathbb{R}^{d \times k}$ is an $(d, k)$-matrix, and $\mu \in \mathbb{R}^d$ is the mean vector. The covariance matrix of $X$ is $\Sigma = AA^\top$ and the distribution of $X$ (that is, the $d$-dimensional *multivariate normal distribution*) is determined solely by the mean vector $\mu$ and the covariance matrix $\Sigma$; we can thus write $X \sim \mathrm{N}_d(\mu, \Sigma)$.

In what follows, we assume $k = d$. If $\Sigma$ is positive definite (thus has full rank and is therefore invertible), $X$ has density

$$f_X(x) = \frac{1}{(2\pi)^{d/2}\sqrt{\det \Sigma}} \exp\left(-\frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu)\right), \quad x \in \mathbb{R}^d.$$

Contours of equal density are ellipsoids; all so-called elliptical distributions which admit a density share this property.

A positive definite (semi-definite) matrix $\Sigma \in \mathbb{R}^{d \times d}$ can be written as

$$\Sigma = LL^\top \tag{2}$$

for a lower triangular matrix $L$ with $L_{jj} > 0$ ($L_{jj} \geq 0$) for all $j \in \{1, \ldots, d\}$. $L$ is known as the *Cholesky factor* of the *Cholesky decomposition* (2).

The stochastic representation (1), together with the Cholesky decomposition of $\Sigma$, allows for a direct sampling strategy of multivariate normal distributions $\mathrm{N}_d(\mu, \Sigma)$, which can easily be implemented in R as follows.

```
> ## Setup
> mu <- 1:2 # mean vector of X
> Sigma <- matrix(c(4, 2, 2, 3), ncol=2) # covariance matrix of X
> n <- 1000 # sample size
> d <- 2 # dimension
> ## Step 1: Compute the Cholesky factor of Sigma
> L <- t(chol(Sigma)) # t() as chol() returns an upper triangular matrix
> ## Step 2: Generate iid standard normal random variates
```

---

[1]The accompanying R script may be obtained from the author upon request.

```
> set.seed(271) # set seed for reproducibility
> Z <- matrix(rnorm(n*d), nrow=d, ncol=n) # (d,n)-matrix
> ## Step 3: Reconstruct the stochastic representation
> X <- mu + L %*% Z # (d,n)-matrix of realizations N_d(mu, Sigma)
```

This idea for sampling $X \sim N_d(\mu, \Sigma)$ is available in the R package **mvtnorm** as follows:

```
> require(mvtnorm)
> set.seed(271)
> X. <- rmvnorm(n, mean=mu, sigma=Sigma, method="chol") # (n,d)-matrix
> stopifnot(identical(t(X), X.)) # check equality of the random numbers
```

The default method (method="eigen") utilizes the eigendecomposition of $\Sigma$, which also applies if some eigenvalues are 0. The function mvrnorm() of the recommended R package **MASS** provides the same approach; see Venables and Ripley (2013) or (Ripley, 1987, pp. 98). Note however, that although the internally drawn independent standard normal random variates are identical, the two algorithms compute different matrices $A$ such that $AA^\top = \Sigma$ and thus do not lead to identical $N_d(\mu, \Sigma)$ random variates.

```
> require(MASS)
> X.. <- mvrnorm(n, mu=mu, Sigma=Sigma) # (n,d)-matrix
```

## The multivariate $t$ distribution

The left-hand side of Figure 1 displays 1000 log-returns of daily stock prices for BMW and Siemens in the decade from 1985-01-02 to 1994-12-30; the data is available in the R package **evir** (Pfaff, 2012). This time period includes various historically "extreme" events such as the stock market crash 1987 known as "Black Monday" (1987-10-19), one of the Monday demonstrations in Leipzig (1989-10-16), and the August Putsch attempt against Mikhail Gorbachev (1991-08-19).
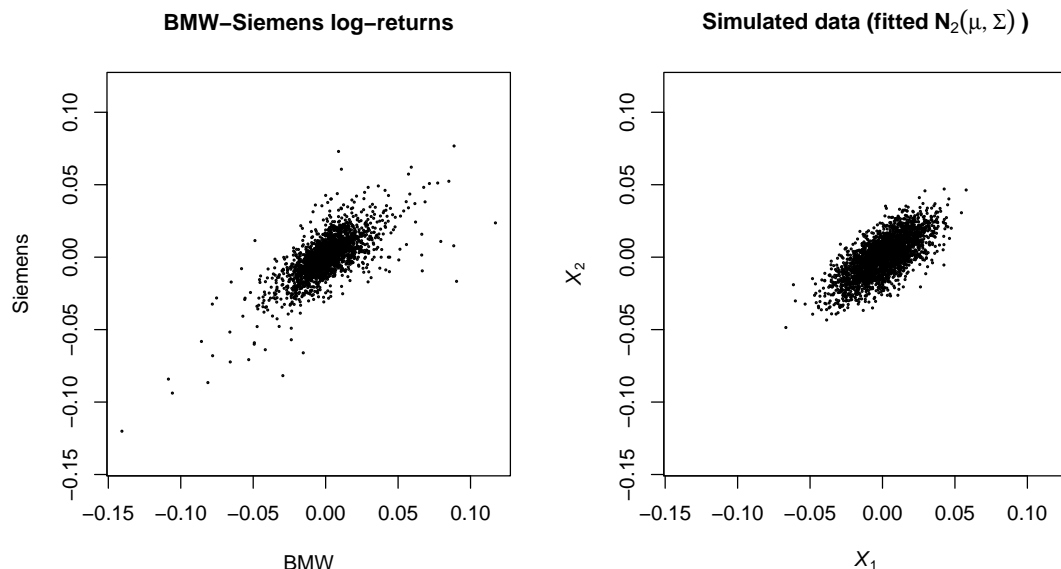


**Figure 1:** Log-returns for daily (BMW, Siemens) data from 1985-01-02 to 1994-12-30 (left). Correspondingly many simulated data points from a fitted bivariate normal distribution (right).

A comparison with simulated data of the same sample size from a fitted bivariate normal distribution on the right-hand side of Figure 1 shows that the bivariate normal distribution is not an adequate model here to account for such (joint) "extreme" events (in the upper right or lower left corner of the bivariate distribution); this can also be checked with formal statistical or graphical goodness-of-fit tests. The bivariate $t$ distribution typically captures such events better (mathematically speaking, it is able to capture "tail dependence") and has gained popularity in modeling such events, for example, in quantitative risk management applications.

### Definition and density

The *multivariate t distribution with ν degrees of freedom* can be defined by the stochastic representation

$$X = \mu + \sqrt{W}AZ, \tag{3}$$

where $W = \nu/\chi_\nu^2$ ($\chi_\nu^2$ is informally used here to denote a random variable following a chi-squared distribution with $\nu > 0$ degrees of freedom) is independent of $Z$ and all other quantities are as in (1).

By introducing the additional random factor $\sqrt{W}$, the multivariate $t$ distribution with $\nu$ degrees of freedom (denoted by $t_\nu(\mu, \Sigma)$) is more flexible than the multivariate normal distribution (which can be recovered by taking the limit $\nu \to \infty$) especially in the tails which are heavier for $t_\nu(\mu, \Sigma)$ than for $N_d(\mu, \Sigma)$. The density of $X \sim t_\nu(\mu, \Sigma)$ is given by

$$f_X(x) = \frac{\Gamma\left(\frac{\nu+d}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)(\pi\nu)^{d/2}\sqrt{\det\Sigma}} \left(1 + \frac{(x-\mu)^\top\Sigma^{-1}(x-\mu)}{\nu}\right)^{-\frac{\nu+d}{2}}, \quad x \in \mathbb{R}^d. \tag{4}$$

As for the multivariate normal distribution, the density (4) has ellipsoidal level sets and thus belongs to the class of elliptical distributions.

### In R

As the CRAN Task View "Distributions" reveals, the R packages **mvtnorm** and **mnormt** (see Azzalini, 2012, for the latter) provide functions for drawing random variates from the multivariate normal and $t$ distributions. The former is the most widely used among all non-recommended packages (measured by Reverse Depends as of August 5, 2012; see Eddelbuettel, 2012). In what follows, we present and discuss changes to **mvtnorm** ($\geq$ 0.9-9996) which were inspired by the corresponding fallacies. Afterwards, we will briefly address **mnormt**.

### Fallacies when sampling the multivariate $t$ distribution

The left-hand side of Figure 2 shows 2608 simulated vectors from a bivariate $t$ distribution fitted to the BMW–Siemens log-return data. The parameters are estimated as follows, utilizing the R package **QRM** (Pfaff and McNeil, 2012).

```
> require(QRM)
> fit <- fit.mst(X, method = "BFGS") # fit a multivariate t distribution
> mu <- fit$mu # estimated location vector
> Sigma <- as.matrix(fit$Sigma) # estimated scale matrix
> nu <- fit$df # estimated degrees of freedom
```

In comparison to the sample from the multivariate normal distribution on the right-hand side of Figure 1, the multivariate $t$ distribution shows significantly heavier tails. This is also indicated by the estimated degrees of freedom parameter $\nu \approx 3.02$.

### The task

We now turn to the task of generating vectors of random variates from a multivariate $t$ distribution with $\nu$ degrees of freedom, that is, generating samples such as shown on the left-hand side of Figure 2. We assume

$$\mu = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 4 & 2 \\ 2 & 3 \end{pmatrix}, \quad \nu = 3, \tag{5}$$

and try to generate $n = 1000$ samples.

### Fallacy 1: Assuming it to work as expected

The obvious generalization of drawing $n$ samples from $N_d(\mu, \Sigma)$ via rmvnorm(n,mean=mu,sigma=Sigma) (with n being $n$, mu being $\mu$, and Sigma being $\Sigma$) to $t_\nu(\mu, \Sigma)$ would be to use rmvt(n,mean=mu,sigma=Sigma,df=nu) (with nu being $\nu$):

```
> require(mvtnorm)
> n <- 1000
```

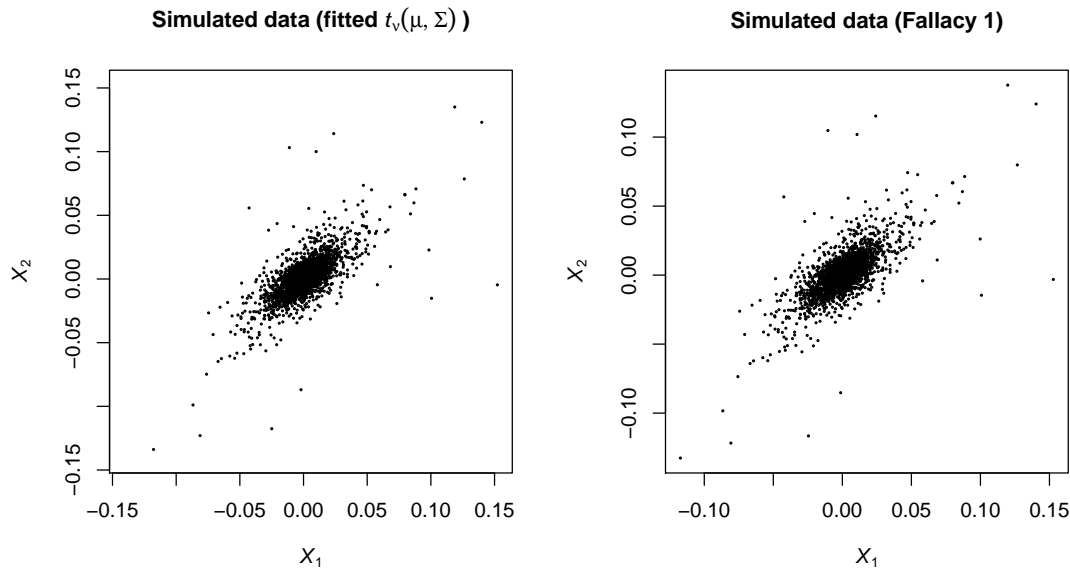**Simulated data (fitted $t_\nu(\mu, \Sigma)$ )**  **Simulated data (Fallacy 1)**



**Figure 2:** Simulated data (sample size 2608) from a fitted bivariate $t$ distribution ($\nu \approx 3.02$) (left). Simulated data (sample size 2608) from (5) with the approach described in Fallacy 1 (7) (right).

```
> mu <- 1:2
> Sigma <- matrix(c(4, 2, 2, 3), ncol=2)
> nu <- 3
> set.seed(271)
> X1 <- try(rmvt(n, mean=mu, sigma=Sigma, df=nu)) # error; 'mean' not allowed anymore
```

In **mvtnorm** ($\geq$ 0.9-9996), this generates an error and is thus not allowed.

To understand why it is dangerous not to throw an error in this situation, let us look at the mean. Theoretically, the mean of $X \sim t_\nu(\mu, \Sigma)$ is given by

$$\mathbb{E}[X] = \mathbb{E}[\mu + \sqrt{W}AZ] = \mu + \mathbb{E}[\sqrt{W}]A\mathbb{E}[Z] = \mu, \tag{6}$$

where we used that $W$ and $Z$ are independent, $\mathbb{E}[Z] = 0$, and that $\nu > 1$ (so that $\mathbb{E}[\sqrt{W}]$ exists). In previous versions of rmvt(), a specified argument mean was passed to the internal call of rmvnorm() via the ellipsis "...". This implies that one actually generated a sample from

$$X = \sqrt{W}Y, \tag{7}$$

where $Y \sim N_d(\mu, \Sigma)$. The expectation of $X$ in this case is

$$\mathbb{E}[X] = \mathbb{E}[\sqrt{W}]\mu$$

instead of $\mu$. $\sqrt{W}$ has distribution function $F_{\sqrt{W}}(x) = 1 - F_{\chi_\nu^2}(\nu/x^2)$ with density $f_{\sqrt{W}}(x) = 2\nu f_{\chi_\nu^2}(\nu/x^2)/x^3$. Using the fact that a $\chi_\nu^2$ distribution is a $\Gamma(\nu/2, 1/2)$ distribution with density $f_{\Gamma(\nu/2,1/2)}(x) = (1/2)^{\nu/2}x^{\nu/2-1}\exp(-x/2)/\Gamma(\nu/2)$, a rather tedious than complicated calculation shows that

$$\mathbb{E}[\sqrt{W}] = \sqrt{\frac{\nu}{2}}\frac{\Gamma((\nu-1)/2)}{\Gamma(\nu/2)}$$

and thus that $\mathbb{E}[X] \approx (1.38, 2.76)$ rather than the required $\mu = (1, 2)$. Table 1 gives an intuition for how fast $\mathbb{E}[\sqrt{W}]$ converges to 1 for large $\nu$. In financial applications, one typically has $\nu$ values between 3 and 5 which implies values of $\mathbb{E}[\sqrt{W}]$ between 1.3820 and 1.1894, respectively.

| $\nu$ | 1.0080 | 1.0794 | 1.7757 | 8.5417 | 76.0418 | 751.0417 |
|---|---|---|---|---|---|---|
| $\mathbb{E}[\sqrt{W}]$ | 101 | 11 | 2 | 1.1 | 1.01 | 1.001 |

**Table 1:** Values of $\nu$ for $\mathbb{E}[\sqrt{W}] \in \{1 + 10^i : i \in \{2, 1, 0, -1, -2, -3\}\}$.

It follows from (1) and (7) that previous versions of `rmvt()` with specified argument mean actually sampled from a random vector $X$ with stochastic representation

$$X = \sqrt{W}\boldsymbol{\mu} + \sqrt{W}A\mathbf{Z}. \qquad (8)$$

The distribution of such $X$ belongs to the class of *normal mean-variance mixtures*; see (McNeil et al., 2005, Section 3.2.2). In general, such distributions are not elliptical distributions anymore. By looking at the source code of `rmvt()`, we can still mimic this previous behavior of `rmvt(n,mean=mu,sigma=Sigma,df= nu)` by

```
> set.seed(271)
> ## exactly the random variates drawn by rmvt(n, mean=mu, sigma=Sigma, df=nu)
> ## in versions of mvtnorm before 0.9-9996:
> X12 <- rmvt(n, sigma=Sigma, df=nu, delta=mu, type="Kshirsagar")
> colMeans(X12) # => wrong (sample) mean

[1] 1.5380 2.7955
```

The result is shown on the right-hand side of Figure 2. In contrast to many other sampling functions in R (even including **mnormt**'s `rmt()`), `rmvt()` does not have an argument mean and previous versions of `rmvt()` thus generated random variates from (8) instead if this argument was provided.

**Remark 1**     • *As we can see from the last chunk,* `rmvt()` *with* `type="Kshirsagar"` *specifically allows to sample* (8). *For other applications of* `type="Kshirsagar"`, *see* `?rmvt`.

• *We saw in* (6) *that* $\boldsymbol{\mu}$ *is only the mean of $X$ if $\nu > 1$. The parameter $\boldsymbol{\mu}$ of $t_\nu(\boldsymbol{\mu}, \Sigma)$ is therefore referred to as* location vector *(as opposed to "mean vector").*

## Fallacy 2: Vector vs matrix

To properly specify the location vector, R users are often tempted to do the following:

```
> X2 <- mu + rmvt(n, sigma=Sigma, df=nu)
```

The problem with this approach is not visible for the human eye here! To make it more pronounced, let us blow up the location vector:

```
> set.seed(271)
> X2 <- 20*mu + rmvt(n, sigma=Sigma, df=nu)
```

The left-hand side of Figure 3 reveals the problem. Indeed, we added the *vector* 20*mu to the *matrix* returned by `rmvt()`. R solves this problem by sufficiently often repeating the vector elements to obtain a matrix of the same size such that addition makes sense.

```
> head(matrix(20*mu, nrow=n, ncol=d))

     [,1] [,2]
[1,]   20   20
[2,]   40   40
[3,]   20   20
[4,]   40   40
[5,]   20   20
[6,]   40   40
```

As we can see (and more experienced R users know this fact well), matrices are filled and worked on column-wise. So every second sample has a different mean vector (alternating between $(20, 20)$ and $(40, 40)$). We thus sampled a mixture of two $t$ distributions and again have left the class of elliptical distributions. The left-hand side of Figure 3 clearly indicates this by showing the two clouds centered around the two mean vectors. This problem is virtually impossible to detect here without the scaling factor (and thus harbors risk of being overlooked).

In order to take care of the correct mean, there are several possibilities, some are:

```
> set.seed(271)
> X21 <- matrix(mu, nrow=n, ncol=d, byrow=TRUE) + rmvt(n, sigma=Sigma, df=nu)
> set.seed(271)
> X22 <- rep(mu, each=n) + rmvt(n, sigma=Sigma, df=nu)
> set.seed(271)
> X23 <- sweep(rmvt(n, sigma=Sigma, df=nu), MARGIN=2, STATS=mu, FUN="+")
> stopifnot(identical(X21, X22),
            identical(X21, X23)) # check equality of the random numbers
```

The last approach is implemented in `rmvt()` in terms of the argument `delta` (if the argument `type` attains its default `"shifted"`):

```
> set.seed(271)
> X24 <- rmvt(n, sigma=Sigma, df=nu, delta=mu)
> stopifnot(identical(X21, X24))
```

**Fallacy 3: The meaning of $\Sigma$ and `sigma`**

After having taken care of the mean vector $\mu$, let us now consider $\Sigma$.

```
> set.seed(271)
> X3 <- rmvt(n, sigma=Sigma, df=nu, delta=mu)
> cov(X3)

        [,1]   [,2]
[1,] 9.8843 4.9204
[2,] 4.9204 7.6861
```

As we see, the sample covariance matrix is not close to $\Sigma$ as specified in (5).

For $X \sim N_d(\mu, \Sigma)$ it is easy to see that $\mathbb{E}[X] = \mu$ and $\text{Cov}[X] = \Sigma$, so $\mu$ and $\Sigma$ are indeed the mean vector and covariance matrix of $X$, respectively. As we have already seen for $\mu$, this is not necessarily true anymore for $t_\nu(\mu, \Sigma)$ due to the additional random factor $\sqrt{W}$ in the stochastic representation (3). The same applies to the covariance matrix of $X$. It follows from (3) that if $\mathbb{E}[W] < \infty$, we have

$$\text{Cov}[X] = \text{Cov}[\sqrt{W}AZ] = \mathbb{E}[(\sqrt{W}AZ)(\sqrt{W}AZ)^\top] = \mathbb{E}[W]\text{Cov}[AZ] = \mathbb{E}[W]\Sigma.$$

It is a basic exercise to show that $W = \nu/\chi_\nu^2$ implies that $\mathbb{E}[W] = \frac{\nu}{\nu-2}$. Therefore, the covariance matrix of $X$ only exists if $\nu > 2$ in which case it is $\text{Cov}[X] = \frac{\nu}{\nu-2}\Sigma$, not $\Sigma$. For this reason, $\Sigma$ is referred to as *scale* (or *dispersion*) *matrix* in order to distinguish it from the covariance matrix of $X$ (which does not have to exist). In our task (5) the covariance matrix of $X$ is $3\Sigma$ which is roughly what we see above (and which can be confirmed by a larger sample size). X3 (displayed on the right-hand side of Figure 3) therefore shows a sample from the correct distribution as specified by (5); note that the same construction principle has been used to create the left-hand side of Figure 2.
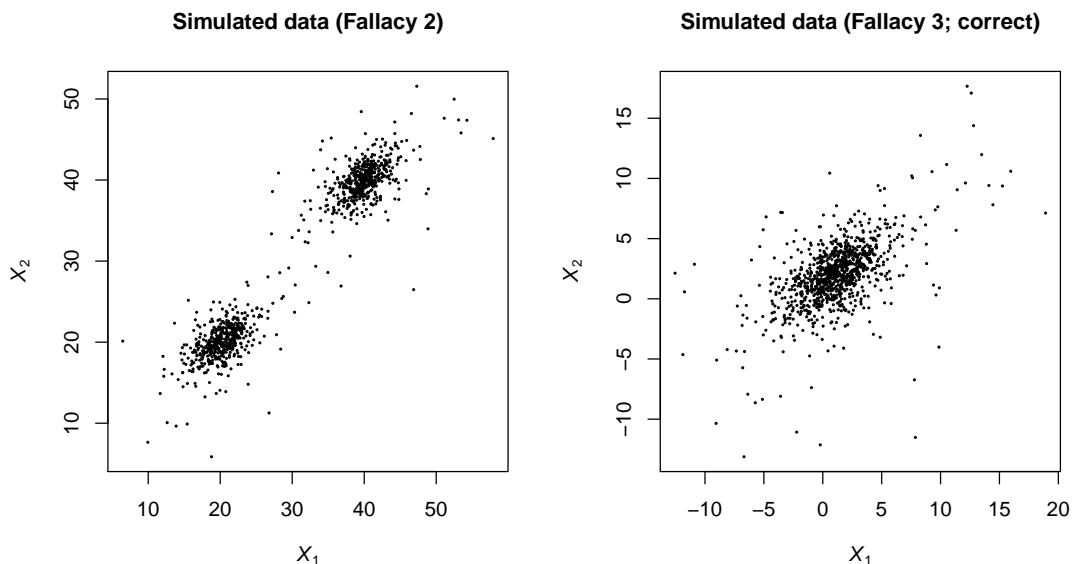


**Figure 3:** Simulated data from the approach described in Fallacy 2 (left) and Fallacy 3 (right).

Finally, let us mention that if $\nu > 2$, then

$$\text{Cor}[X] = P = (\rho_{ij})_{i,j\in\{1,\dots,d\}}, \quad \text{where } \rho_{ij} = \frac{\text{Cov}[X_i, X_j]}{\sqrt{\text{Var}[X_i]\,\text{Var}[X_j]}}.$$

Let $\Sigma = (\sigma_{ij})_{i,j \in \{1,\dots,d\}}$. Since

$$\text{Cov}[X_i, X_j] = \mathbb{E}[W]\sigma_{ij}, \quad i, j \in \{1,\dots,d\},$$
$$\text{Var}[X_k] = \mathbb{E}[(\sqrt{W}(A\mathbf{Z})_k)^2] = \mathbb{E}[W]\text{Var}[(A\mathbf{Z})_k] = \mathbb{E}[W]\sigma_{kk}, \quad k \in \{1,\dots,d\},$$

where $(A\mathbf{Z})_k$ denotes the $k$th row of $A\mathbf{Z}$, we obtain

$$P_{ij} = \frac{\mathbb{E}[W]\sigma_{ij}}{\sqrt{\mathbb{E}[W]\sigma_{ii}\mathbb{E}[W]\sigma_{jj}}} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}}.$$

If $\nu > 2$, we see that although the covariance matrix $\text{Cov}[X]$ is not equal to the scale matrix $\Sigma$, the correlation matrix $\text{Cor}[X]$ is equal to the correlation matrix $P$ corresponding to the scale matrix $\Sigma$. This can also be verified numerically:

```
> set.seed(271)
> ## sample correlation matrix of a t3 sample with scale matrix Sigma
> cor(rmvt(1e6, sigma=Sigma, df=3, delta=mu))

        [,1]    [,2]
[1,] 1.00000 0.57667
[2,] 0.57667 1.00000

> ## correlation matrix corresponding to Sigma
> cov2cor(Sigma)

        [,1]    [,2]
[1,] 1.00000 0.57735
[2,] 0.57735 1.00000
```

**Remark 2** *The user should also be aware of the fact that* `rmvt(n,delta=mu,sigma=Sigma,df=0,...)` *is equivalent to* `rmvnorm(n,mean=mu,sigma=Sigma,...)`. *This is counter-intuitive as the multivariate normal distribution arises as* $\nu \to \infty$, *not* $\nu \to 0+$. *This might be problematic for very small degrees of freedom parameters* $\nu$ *due to truncation to 0. Note that* **mvtnorm** ($\geq$ 0.9-9996) *now also allows to specify* `df=Inf` *for the (same) limiting multivariate normal distribution. The case* `df=0` *remains for backward compatibility.*

## Comparison with mnormt

The R package **mnormt** provides the function `rmt()` for sampling from the multivariate $t$ distribution. The call `rmt(n,mean=mu,S=Sigma,df=nu)` provides the correct answer to task (5). Note that `rmt()` has an argument `mean`, but actually means the location vector (see `?rmt`). Furthermore, there is a subtle difference between **mnormt** and **mvtnorm**. **mnormt**'s `rmnorm()` uses a Cholesky decomposition of $\Sigma$. Even by starting with the same seed and calling **mvtnorm**'s `rmvt()` with `method="chol"`, the vectors of random variates generated by `rmt(n,mean=mu,S=Sigma,df=nu)` and those by `rmvt(n,sigma=Sigma,df=nu, delta=mu,method="chol")` are not identical. The reason for this is that the order in which the normal and the $\chi_\nu^2$ distributed random variates are generated differs for `rmt()` and `rmvt()`.

## Summary and conclusion

Table 2 collects the various calls of `rmvt()` and corresponding stochastic representations of $X$ we encountered.

To summarize, let $X \sim t_\nu(\boldsymbol{\mu}, \Sigma)$. Then:

1. The location vector $\boldsymbol{\mu}$ is not equal to $\mathbb{E}[X]$ unless $\nu > 1$ and thus $\mathbb{E}[X]$ exists. The scale matrix $\Sigma$ is *a* covariance matrix but not equal to *the* covariance matrix of $X$.

2. $X$ can be sampled via `rmvt(n,sigma=Sigma,df=nu,delta=mu)`, where `n` is the sample size $n$, `mu` the location vector $\boldsymbol{\mu}$, `Sigma` the scale matrix $\Sigma$, and `nu` the degrees of freedom parameter $\nu$; this holds for all $\nu > 0$ (watch out for very small ones, though).

3. The argument `sigma` of `rmvt()` denotes the scale matrix $\Sigma$ of $X$. If the scale matrix $\Sigma$ is standardized, it is *a* correlation matrix, but not necessarily *the* correlation matrix of $X$ (the latter does not have to exist). Only if $\nu > 2$, $\text{Cor}[X]$ exists and equals the correlation matrix corresponding to the scale matrix $\Sigma$ (which can be computed via `cov2cor()`).

| Call | Stochastic representation of $X$ |
|---|---|
| `X1  <- rmvt(n, mean=mu, sigma=Sigma, df=nu);` | gives an error |
| `X12 <- rmvt(n, sigma=Sigma, df=nu, delta=mu, type="Kshirsagar");` | $X = \sqrt{W}(\mu + AZ)$ |
| `X2  <- mu + rmvt(n, sigma=Sigma, df=nu);` | mixture of two $t$ distributions (see text) |
| `X21 <- matrix(mu, nrow=n, ncol=d, byrow=TRUE) + rmvt(n, sigma=Sigma, df=nu);` | as X3 |
| `X22 <- rep(mu, each=n) + rmvt(n, sigma=Sigma, df=nu);` | as X3 |
| `X23 <- sweep(rmvt(n, sigma=Sigma, df=nu), MARGIN=2, STATS=mu, FUN="+");` | as X3 |
| `X24 <- rmvt(n, sigma=Sigma, df=nu, delta=mu);` | as X3 |
| `X3  <- rmvt(n, sigma=Sigma, df=nu, delta=mu);` | $X = \mu + \sqrt{W}AZ \sim t_\nu(\mu, \Sigma)$ |

**Table 2:** Calls of `rmvt()` and corresponding stochastic representations of $X$.

## Acknowledgments

## Bibliography

A. Azzalini. *mnormt: The Multivariate Normal and t Distributions*, 2012. URL http://CRAN.R-project.org/package=mnormt. R package version 1.4.5. [p131]

D. Eddelbuettel. Counting CRAN package Depends, Imports and LinkingTo, 2012. URL http://dirk.eddelbuettel.com/blog/2012/08/05/. [p131]

A. Genz, F. Bretz, T. Miwa, X. Mi, F. Leisch, F. Scheipl, and T. Hothorn. *mvtnorm: Multivariate Normal and t Distributions*, 2013. URL http://CRAN.R-project.org/package=mvtnorm. R package version 0.9-9996. [p129]

A. J. McNeil, R. Frey, and P. Embrechts. *Quantitative Risk Management: Concepts, Techniques, Tools*. Princeton University Press, 2005. [p133]

B. Pfaff. *evir: Extreme Values in R*, 2012. URL http://CRAN.R-project.org/package=evir. R package version 1.7-3. [p130]

B. Pfaff and A. McNeil. *QRM: Provides R-Language Code to Examine Quantitative Risk Management Concepts*, 2012. URL http://CRAN.R-project.org/package=QRM. R package version 0.4-8. [p131]

B. D. Ripley. *Stochastic Simulation*. John Wiley & Sons, 1987. [p130]

W. N. Venables and B. D. Ripley. *MASS: Support Functions and Datasets for Venables and Ripley's MASS*, 2013. URL http://CRAN.R-project.org/package=MASS. R package version 7.3.26. [p130]

*Marius Hofert*
*RiskLab, Department of Mathematics,*
*ETH Zürich, 8092 Zürich,*
*Switzerland.*
marius.hofert@math.ethz.ch