

Non-life — Assignment NL1

Niels Keizer* and Robert Jan Sopers†

September 17, 2016

Generating multinormal and multi-student r.v.'s

Q1

First we run the following script:

```
> set.seed(1)
> sum(duplicated(runif(1e6))) ## = 120
[1] 120
> sum(duplicated(rnorm(1e8))) ## = 0
[1] 0
```

The function `duplicated` returns a logical array where unique numbers are marked with 0 and duplicates are marked with 1 (the first occurrence of the number is marked with a 0). Summing this array thus gives the total number of duplicates. The uniform distribution gives 120 duplicates in a much smaller sample size than the normal distribution, which gives 0 duplicates.

In the assignment, the expected number of different numbers is derived to be

$$\mathbb{E}[N_n] = \frac{1 - f^n}{1 - f}$$

The number of duplicates is then given by $n - \mathbb{E}[N_n]$. We run the following script:

```
> m <- 2^32; n <- 1e6
> f <- 1 - 1/m
> num_dup_unif <- n - (1 - f^n)/(1 - f)
> num_dup_unif
[1] 116.3988
```

The expected result of 116.4 is quite close to the generated result. The outcome of 120 is therefore consistent with the assumption that `runif` produces different values uniformly.

*Student number: 10910492

†Student number: 99999999

The resolution of `n_norm` is somewhere in the 2^{50} 's. Directly calculating $1 - f^n$ will give 1, because f is so close to 1. First, we use to approximation given in the assignment.

$$f^n = \left(1 - \frac{1}{m}\right)^n \approx 1 - \frac{n}{m} + \frac{n^2}{2m^2}$$

Inserting this into our equation for the number of different numbers gives

$$\frac{1 - f^n}{1 - f} \approx \frac{1 - 1 + \frac{n}{m} - \frac{n^2}{2m^2}}{1 - \left(1 - \frac{1}{m}\right)} = \frac{\frac{n}{m} - \frac{n^2}{2m^2}}{\frac{1}{m}} = n - \frac{n^2}{2m}$$

This results in the following equation for the expected number of duplicates.

$$n - \left(n - \frac{n^2}{2m}\right) = \frac{n^2}{2m}$$

Next we check in R if the number of duplicates is consistent with values for m of 10^{15} , 10^{16} , 10^{17} or 10^{18} , when n is 10^8

```
> n_norm <- 1e8
> m_norm <- c(1e15, 1e16, 1e17, 1e18)
> num_dup_norm <- n_norm^2 / (2 * m_norm)
> num_dup_norm
[1] 5.000 0.500 0.050 0.005
```

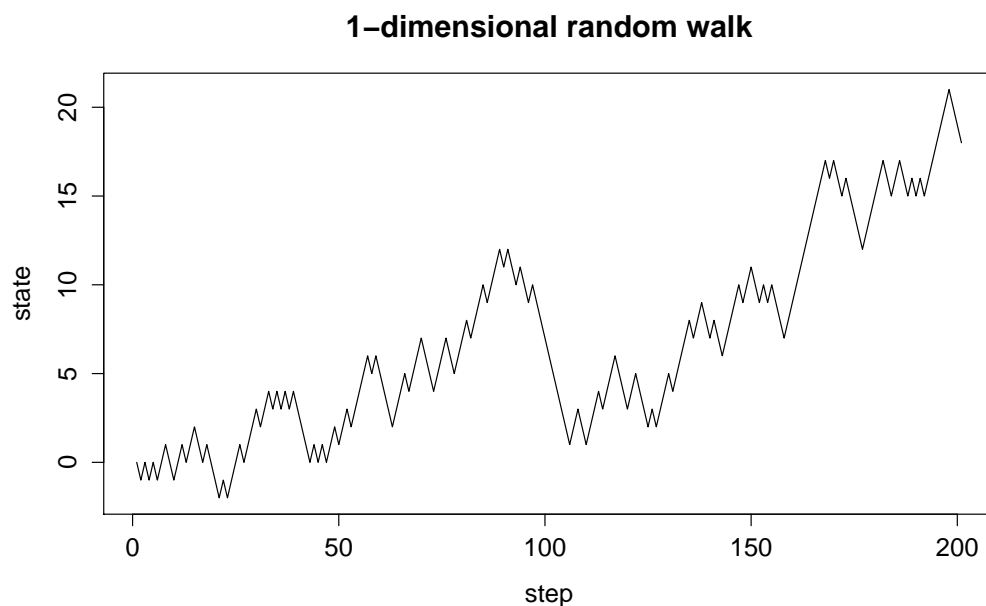
The obtained result seems to be consistent with a resolution of 10^{16} or higher.

Q2

The following code is executed in R.

```
> n <- 200; p <- 0.52
> x <- c(0, cumsum(2 * rbinom(n, 1, p) - 1))
> plot(x, type="l", lwd=1, ylab="state", xlab="step", main="1-dimensional random walk")
```

This gives the following biased random walk:



Q3

Given that $X, Y \sim N(0, 1)$, we want to transform (X, Y) into (X, Y^*) , with $Y^* = aX + bY$, with a, b chosen in such a way that $\text{Var}[Y^*] = 1$ and $r(X, Y^*) = 0.8$.

$$\begin{aligned} r(X, Y^*) &= \frac{\mathbb{E}[XY^*] - \mathbb{E}[X] \mathbb{E}[Y^*]}{\sqrt{\text{Var}[X] \text{Var}[Y^*]}} = \frac{\mathbb{E}[aX^2 + bXY] - 0 \cdot \mathbb{E}[Y^*]}{\sqrt{1 \cdot 1}} \\ &= a * \mathbb{E}[X^2] + b * \mathbb{E}[XY] = a(\text{Var}[X] - \mathbb{E}[X]^2) = a \end{aligned}$$

Here we use that $\mathbb{E}[X] = 0$, $\text{Var}[X] = \text{Var}[Y] = \text{Var}[Y^*] = 1$. Also $\mathbb{E}[XY] = \mathbb{E}[X] \mathbb{E}[Y] = 0$, because X and Y are independent. Next we use the condition that the variance of Y^* must also be 1.

$$\text{Var}[Y^*] = \text{Var}[aX + bY] = a^2 \text{Var}[X] + b^2 \text{Var}[Y] + 2ab \text{Cov}[X, Y] = a^2 + b^2 = 1$$

We can conclude from this that $a = 0.8$ and $b = \sqrt{1 - a^2} = 0.6$.

In R code:

```
set.seed(2004); options(digits=2)
X <- rnorm(1000); Y <- rnorm(1000)
a <- .8; b <- sqrt(1 - a^2); Y <- a*X + b*Y
```

Q4

The variance-covariance matrix Σ of the random vector (X, Y^*) is equal to the correlation matrix because $\text{Var}[X] = \text{Var}[Y^*] = 1$. It is given by the following expression.

$$\Sigma = \begin{pmatrix} \text{Cov}[X, X] & \text{Cov}[X, Y^*] \\ \text{Cov}[Y^*, X] & \text{Cov}[Y^*, Y^*] \end{pmatrix} = \begin{pmatrix} \text{Var}[X] & r(X, Y^*) \\ r(X, Y^*) & \text{Var}[Y^*] \end{pmatrix} = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$$

Q5

If A is to be the Cholesky decomposition of Σ , it should be a lower triangular matrix with real and positive entries and $AA^* = \Sigma$, where A^* is the conjugate transpose of A . Checking this gives

$$\begin{pmatrix} 1 & 0 \\ a & b \end{pmatrix} \begin{pmatrix} 1 & a \\ 0 & b \end{pmatrix} = \begin{pmatrix} 1 & a \\ a & a^2 + b^2 \end{pmatrix} = \begin{pmatrix} 1 & a \\ a & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix} = \Sigma$$

Q6

We execute the following R code.

```
> c(mean(X), var(X), mean(Y), var(Y), cor(X,Y))
[1] 0.051 0.983 0.070 0.994 0.796
```

The means of X and Y^* are close to 0. The variances close to 1 and the correlation is close to 0.8. This resembles the theoretical values quite close.

Q7

Let (X, Y) be bivariate Normal with $\mathbb{E}[X] = \mathbb{E}[Y] = 0$, $\text{Var}[X] = \text{Var}[Y] = 1$ and $r(X, Y) = r$. W is independent of (X, Y) . Then

$$\begin{aligned} r(XW, YW) &= \frac{\mathbb{E}[XWYW] - \mathbb{E}[XW]\mathbb{E}[YW]}{\sqrt{\text{Var}[XW]\text{Var}[YW]}} \\ &= \frac{\mathbb{E}[W^2]\mathbb{E}[XY] - \mathbb{E}[W]^2\mathbb{E}[X]\mathbb{E}[Y]}{\sqrt{(\mathbb{E}[W^2]\mathbb{E}[X^2] - \mathbb{E}[W]^2\mathbb{E}[X]^2)(\mathbb{E}[W^2]\mathbb{E}[Y^2] - \mathbb{E}[W]^2\mathbb{E}[Y]^2)}} \\ &= \frac{\mathbb{E}[W]^2\mathbb{E}[XY] - 0}{\sqrt{(\mathbb{E}[W^2]\mathbb{E}[X^2] - 0)(\mathbb{E}[W^2]\mathbb{E}[Y^2] - 0)}} \\ &= \frac{\mathbb{E}[W^2]}{\sqrt{\mathbb{E}[W^2]^2}} \frac{\mathbb{E}[XY]}{\text{Var}[X]\text{Var}[Y]} = r \cdot \frac{\mathbb{E}[W^2]}{\sqrt{\mathbb{E}[W^2]^2}} = r \end{aligned}$$

When $\mathbb{E}[W^2]$ is finite, the final step in the derivation is allowed. This is equivalent with demanding $\text{Var}[W]$ to be finite.

Q8

We take X and Y^* as defined earlier. $V \sim \chi_k^2$ and $W = \sqrt{k/V}$ with $k = 5$. The population mean of XW is 0 because $\mathbb{E}[XW] = \mathbb{E}[X]\mathbb{E}[W] = 0$. The same holds for Y^*W . $r(XW, Y^*W) = 0.8$ has been proven in question 7. Next we determine the variance of XW .

$$\begin{aligned} \text{Var}[XW] &= \mathbb{E}[X]^2 \text{Var}[W] + \text{Var}[X] \mathbb{E}[W]^2 + \text{Var}[X] \text{Var}[W] \\ &= 0 + \text{Var}[X](\mathbb{E}[W]^2 + \text{Var}[W]) = \text{Var}[X] \mathbb{E}[W^2] \\ &= 1 \cdot \mathbb{E}[k/V] = k \mathbb{E}[1/V] = \frac{k}{k-2} \end{aligned}$$

Here we use that $\mathbb{E}[1/V] = 1/(k-2)$, which we will derive below, $f_{\chi^2}(x; k)$ is the probability density function of the chi-squared distribution with k degrees of freedom.

$$\begin{aligned} \mathbb{E}[1/V] &= \int_0^\infty \frac{1}{x} f_{\chi^2}(x; k) dx = \int_0^\infty \frac{1}{x} \frac{x^{(k/2-1)} e^{-x/2}}{2^{k/2} \Gamma(k/2)} dx = \int_0^\infty \frac{x^{(k/2-2)} e^{-x/2}}{2^{k/2} \Gamma(k/2)} dx \\ &= \int_0^\infty \frac{x^{((k-2)/2-1)} e^{-x/2}}{2^{k/2} \Gamma(k/2)} dx = \int_0^\infty \frac{x^{((k-2)/2-1)} e^{-x/2}}{2 \cdot 2^{(k-2)/2} \frac{k-2}{2} \Gamma(\frac{k-2}{2})} dx \\ &= \frac{1}{k-2} \int_0^\infty f_{\chi^2}(x; k-2) dx = \frac{1}{k-2} \end{aligned}$$

In the derivation, we use $\Gamma(k/2) = \Gamma(\frac{k-2}{2} + 1) = \frac{k-2}{2} \Gamma(\frac{k-2}{2})$.

Now we execute the following R code.

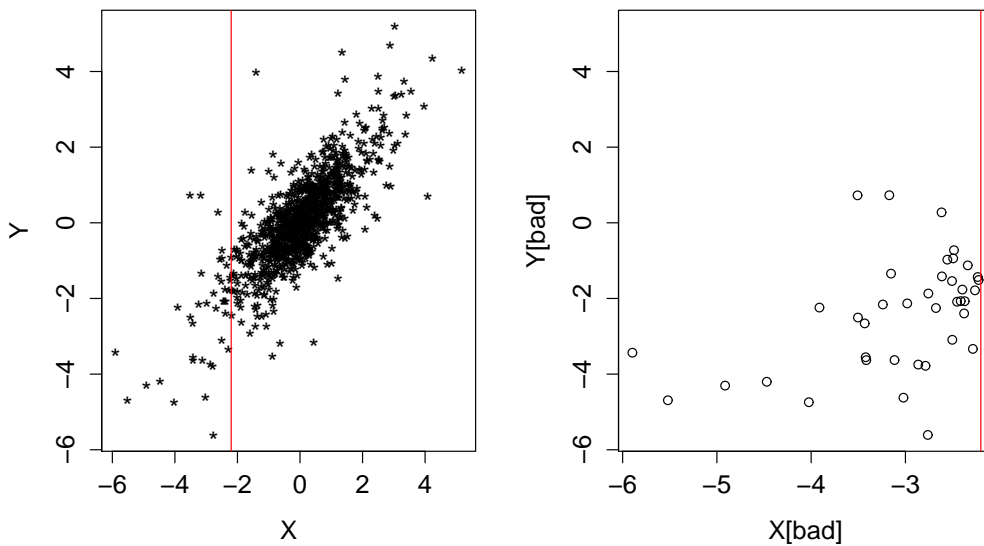
```
> chi5 <- sqrt(rchisq(1000, df=5)/5)
> X <- X/chi5; Y <- Y/chi5
> c(mean(X), var(X), mean(Y), var(Y), cor(X,Y))
[1] 0.038 1.525 0.068 1.528 0.786
```

We see that the means are close to 0, as expected, the variances differ by about 0.1 and the correlation is quite close to 0.8.

Q8

As instructed in an earlier example, we execute the following R code to obtain side by side scatterplots.

```
> par(mfrow=c(1,2))
> plot(X,Y, pch="*")
> d <- -2.2
> abline(v=d, col="red")
>
> bad <- (X < d)
> plot(X[bad], Y[bad], ylim=range(Y))
> abline(v=d, col="red")
> cor(X[bad],Y[bad])
[1] 0.44
```



We can see a tail dependence for the multiStudent distribution in the plot on the right hand side. The tail correlation of 0.44 is lower than 0.8, but is a lot larger than 0. This means that there is a tail dependence.

Q10

Let \vec{Z} be a correlated multinormal random vector with mean \vec{m} and covariance matrix Σ . We denote ρ as the correlation matrix and we use shorthand notation $\rho(Z_i, Z_j) = \rho_{i,j}$ and $\text{Cov}(Z_i, Z_j) = \Sigma_{i,j}$.

By definition, $\rho(Z_i, Z_j) = \frac{\text{Cov}(Z_i, Z_j)}{\sqrt{\text{Var}[Z_i] \text{Var}[Z_j]}}$. Using the shorthand, we derive

$$\begin{aligned}
\Sigma_{i,j} &= \rho_{i,j} \sqrt{\text{Var}[\vec{Z}_i] \text{Var}[\vec{Z}]_j} \\
&= \rho_{i,j} \sqrt{(\text{Var}[\vec{Z}] \text{Var}[\vec{Z}]^T)_{i,j}} \\
\Sigma &= \rho \sqrt{\text{Var}[\vec{Z}] \text{Var}[\vec{Z}]^T} \\
&= \rho \sqrt{\text{Var}[\vec{Z}] \otimes \text{Var}[\vec{Z}]}
\end{aligned}$$

In the first step, we rewrite from index notation to vector notation as can be seen on the wikipedia page of the outer product. We then drop the indexes and use that the outer product can be written as a matrix multiplication.

We then execute the following code:

```

> library(MASS)
> mu <- c(1,3,5); sig2 <- c(1,2,5)
> Corrmat <- rbind(c(1., .3, .3),
+                  c(.3, 1., .4),
+                  c(.3, .4, 1.))
> Varmat <- Corrmat * sqrt(sig2 %*% t(sig2))
> Z <- mvrnorm(100, mu, Varmat)
> options(digits=7)
> colMeans(Z); diag(cov(Z)); cor(Z)
[1] 0.9497085 2.8927698 4.7217666
[1] 0.9276677 2.1010010 5.4895002
[,1]      [,2]      [,3]
[1,] 1.0000000 0.1696436 0.1025938
[2,] 0.1696436 1.0000000 0.4265725
[3,] 0.1025938 0.4265725 1.0000000

```

`colMeans(Z)` estimates the mean of each column. Therefore `colMeans` should be close to the mean vector $\vec{\mu}$. `diag(cov(Z))` gives the diagonal elements of the covariance matrix. The diagonal elements of the covariance matrix contains the estimated variance of each element of Z . This should be close to $\text{Var}[\vec{Z}]$ (or `sig2`). `cor(Z)` contains the estimated correlation between the three elements of Z .

Q11

We execute the following code in R:

```

> no_sims = 1e6
> VaR <- rep(0,10)
> for (i in (1:10)){
+   Z <- mvrnorm(no_sims, mu, Varmat)
+   VaR[i] <- quantile(rowSums(Z),0.9999)
+ }
> VaR
[1] 22.34890 22.27627 22.24395 22.40076 22.29522 22.23880 22.33798 22.17733 22.32926 22.31303
> c(mean(VaR),sd(VaR))
[1] 22.29615014 0.06445194

```

We are asked to determine the $F_Z^{-1}(0.9999)$, where $Z = Z_1 + Z_2 + Z_3$, with Z_i as defined in question 10. Because Z_i are random normally distributed, so is the sum. The mean of Z is given by the following equation.

$$\mathbb{E}[Z] = \mathbb{E}[Z_1] + \mathbb{E}[Z_2] + \mathbb{E}[Z_3] = 1 + 3 + 5 = 9$$

The variance is given by the following formula:

$$\text{Var}[Z] = \text{Var}[Z_1] + \text{Var}[Z_2] + \text{Var}[Z_3] + 2(\text{Cov}(Z_1, Z_2) + \text{Cov}(Z_1, Z_3) + \text{Cov}(Z_2, Z_3))$$

The right hand side of this equation is equal to the sum of the elements of covariance matrix from the previous question. The theoretical value is then computed by executing the following code in R.

```
> Var_t <- qnorm(0.9999, sum(mu), sqrt(sum(Varmat)))
> Var_t
[1] 22.26391}
```

The estimated value of 22.296 is close to the theoretical value of 22.264.

Q12

We execute the following code in R.

```
> n <- 1e6
> mu <- c(0,0,0)
> sigma <- rbind(c(1,1/6,1/6),
+               c(1/6,1,1/6),
+               c(1/6,1/6,1))
> Z <- mvrnorm(n, mu,sigma)
```

Q13

We execute the following code in R.

```
> V <- rowSums(Z)
```

Q14

We execute the following code in R.

```
> d <- quantile(V,0.975)
> d
97.5%
3.918903
```

The estimate for the 97,5% quantile of V is 3.92.

Q15

We execute the following code in R.

```
> stoploss_premium <- mean(pmax(V-d,0))
> stoploss_premium
[1] 0.01884366
```

The estimated stoploss premium is 0.0188.

Q16

By definition, the random variables X_i , Y_i and Z_i are normally distributed. The sum of normally distributed variables is also normally distributed, therefore V_i is also normally distributed. The mean and variance of V_i can be obtained in the same manner as in question 11.

We run the following code in R to obtain the mean and standard deviation of V_i .

```
> mean <- sum(mu); sd <- sqrt(sum(sigma))
> c(mean, sd)
[1] 0 2
```

Q17

First, we calculate the theoretical value of d , after which we use formula (3.104) from MART.

```
> d_new <- qnorm(0.975, mean, sd)
> stoploss_premium_mart <- sd * dnorm((d_new - mean)/sd) - (d_new - mean)*(1 - pnorm((d_new - mean)/sd))
> stoploss_premium_mart
[1] 0.01889194
```

The theoretical stoploss premium of 0.01889 is really close to its estimated value of 0.01884366.

Q18

```
> n <- 1e6
> k <- 5
> mu <- c(0,0,0)
> sigma <- rbind(c(1,1/6,1/6),
+               c(1/6,1,1/6),
+               c(1/6,1/6,1))
> Z <- mvrnorm(n, mu,sigma)
> chi5 <- sqrt(rchisq(n, df=5)/5)
> Z_prime <- Z/chi5
> V_prime <- rowSums(Z_prime)
> d <- quantile(V_prime,0.975)
> stoploss_premium <- mean(pmax(V_prime-d,0))
```



```
> stoploss_premium  
[1] 0.04818821
```