Non-Life Insurance — Assignment 1

AJ*

Autumn 2021

Question 1

To illustrate the difference in resolution between `runif` and `rnorm`, we run the following code:

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

We have selected a point in base `R`'s vector of pseudo-random numbers using `set.seed` and then generated a vector of length 1 million containing uniform random variables on the interval $[0, 1]$, as well as a vector of length 100 million containing $N(0, 1)$ random variables. By the inversion method, the simulated random variables are equivalent. Clearly, the added resolution of `rnorm` makes a significant impact on performance: it has 100 times more elements and produces 120 fewer duplicates.

*Student number: ∞

Question 2

(a) Define two functions:

```
ENsubk <- function(m, k) {
  f <- 1 - 1/m
  (1-f^k) / (1-f)
}

ENapprox <- function(m,k) {
  k - (k^2 / (2*m))
}
```

We then calculate the expected number of duplicates that `runif` will generate:

```
> 1e6 - ENsubk(2^32, 1e6)
[1] 116.4062
```

Hence, using the formula for $\mathrm{E}[N_k]$, we see that the outcome of 120 is consistent with the assumption that `runif` randomly draws from $m = 2^{32}$ distinct numbers.

(b) Let's generate some approximations using the binomial expansion of f^k as implemented by the function `ENapprox`:

```
> 1e8 - ENapprox(10^15, 1e8)
[1] 5
> 1e8 - ENapprox(10^16, 1e8)
[1] 0.5
> 1e8 - ENapprox(10^17, 1e8)
[1] 0.05
```

Again, this is consistent with the stated assumptions regarding `rnorm`.

Question 3

The following code generates three realisations of the specified *random walk*:

```
n <- 200; set.seed(7);
y <- cumsum(c(0, sample(c(1,-1), n, replace = TRUE,
                        prob = c(.55, .45))));
plot(y, col="blue", type="l", ylab="State",
     xlab="Sample size", main="Random Walk", ylim = c(-15,50));

set.seed(17);
y <- cumsum(c(0, sample(c(1,-1), n, replace = TRUE,
                        prob = c(.55, .45))));
lines(y, col="forestgreen");

set.seed(117);
y <- cumsum(c(0, sample(c(1,-1), n, replace = TRUE,
                        prob = c(.55, .45))));
lines(y, col="red");
```

Question 4

(a) We have: $X, V \sim N(0,1)$, such that X, V independent.
Consider $Y = aX + bV$ for chosen constants a, b such that $\text{Var}[Y] = 1$ and $\text{corr}(X,Y) = 0.8$.

We have:

$$\begin{aligned} \text{E}[Y] &= \text{E}[aX + bV] \\ &= a\,\text{E}[X] + b\,\text{E}[V] \\ &= 0 \end{aligned}$$

and

$$\begin{aligned} \text{Var}[Y] &= \text{Var}[aX + bV] \\ &= a^2\,\text{Var}[X] + b^2\,\text{Var}(V) \\ &= a^2 + b^2 := 1 \end{aligned}$$

Now,

$$\begin{aligned} \text{corr}(X,Y) &= \frac{\text{E}[XY] - \text{E}[X]\text{E}[Y]}{\sqrt{\text{Var}[X].\text{Var}[Y]}} \\ &= \frac{\text{E}[XY] - \text{E}[X]\text{E}[Y]}{\sigma_x \sigma_y} \\ &= \text{E}[XY] \end{aligned}$$

Therefore,

$$\begin{aligned} 0.8 = \text{corr}(X,Y) &= \text{E}[XY] \\ &= \text{E}[X(aX + bV)] \\ &= \text{E}[aX^2 + bXV] \\ &= a\,\text{E}[X^2] + b\,E[X]E[V] \\ &= a\,\text{E}[X^2] \\ &= a\,\text{Var}[X] \\ &= a \end{aligned}$$

Combining results yields:

$$b^2 = 1 - a^2 = 0.36$$

So,

$$b = \pm 0.6$$

We then run the following code:

```
set.seed(92020);
X <- rnorm(1000); V <- rnorm(1000)
a <- .8 ; b <- .6 ; Y <- a*X + b*V
rm(V);
```

(b) Comparing the sample and theoretical values can be done easily using:

```
> empirical <- c(mean(X), mean(Y), var(X), var(Y), cor(X,Y));
> theoretical <- c(0, 0, 1, 1, 0.8);
> error.term <- abs(empirical-theoretical)
> tabled.vals <- round(rbind(empirical, theoretical, error.term), 3)
> print(tabled.vals)
             [,1]  [,2] [,3]  [,4]  [,5]
empirical   0.037 0.007 1.03 1.033 0.798
theoretical 0.000 0.000 1.00 1.000 0.800
error.term  0.037 0.007 0.03 0.033 0.002
```

Indeed, they are close.

Question 5

(a)

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathrm{N}(\mathbf{0}, \mathbf{\Sigma})$$

where

$$\mathbf{\Sigma} = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$$

(b)

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \mathbf{A} \begin{pmatrix} X \\ V \end{pmatrix}$$

Consider

$$\mathbf{A} = \begin{pmatrix} 1 & 0 \\ a & b \end{pmatrix}$$

$\mathbf{A}$ is certainly lower triangular. Now,

$$\begin{aligned} \mathbf{A}.\mathbf{A}^T &= \begin{pmatrix} 1 & 0 \\ a & b \end{pmatrix} \begin{pmatrix} 1 & a \\ 0 & b \end{pmatrix} \\ &= \begin{pmatrix} 1 & a \\ a & a^2 + b^2 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix} \end{aligned}$$

Question 6

(a)

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathrm{N}(\mathbf{0}, \mathbf{\Sigma})$$

where

$$\mathbf{\Sigma} = \begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix}$$

and W independent of (X, Y).

To show:

$$\mathrm{corr}(XW, YW) = r$$

By definition,

$$\mathrm{corr}(XW, YW) = \frac{\mathrm{Cov}(XW, YW)}{\sigma_{xw}\sigma_{yw}}$$

Now,

$$\begin{aligned} \mathrm{Cov}(XW, YW) &= \mathrm{E}[XW.YW] - \mathrm{E}[XW]\mathrm{E}[YW] \\ &= \mathrm{E}[W^2]\ \mathrm{E}[XY] - \mathrm{E}[W^2]\ \mathrm{E}[X]\mathrm{E}[Y] \\ &= \mathrm{E}[W^2] \cdot \mathrm{Cov}(X, Y) \\ &= \mathrm{E}[W^2] \cdot r \end{aligned}$$

and

$$\begin{aligned} \mathrm{Var}(XW) &= \mathrm{E}[W^2] \cdot \mathrm{Var}[X] \\ &= \mathrm{E}[W^2] = \mathrm{Var}(YW) \end{aligned}$$

Combining these, we see:

$$\mathrm{corr}(XW, YW) = \frac{r \cdot \mathrm{E}[W^2]}{\sqrt{(\mathrm{E}[W^2])^2}} = r$$

(b) We have (X, Y) as defined in part (a). Now, $V \sim \chi^2_k$ with k ≥ 3, independent of (X, Y). Define,

$$\tilde{X} := \frac{X}{\sqrt{\frac{V}{k}}}$$

and

$$\tilde{Y} := \frac{Y}{\sqrt{\frac{V}{k}}}$$

then,

$$\begin{aligned} \mathrm{E}[\tilde{X}] &= \mathrm{E}\left[\frac{X}{\sqrt{\frac{V}{k}}}\right] \\ &= \mathrm{E}\left[\frac{1}{\sqrt{\frac{V}{k}}}\right] \cdot \mathrm{E}[X] \\ &= 0 = \mathrm{E}[\tilde{Y}] \end{aligned}$$

Similarly,

$$\begin{aligned}
\mathrm{Var}[\tilde{X}] &= \mathrm{E}[\tilde{X}^2] - \mathrm{E}[\tilde{X}]^2 \\
&= \mathrm{E}\left[\left(\frac{X}{\sqrt{\frac{V}{k}}}\right)^2\right] \\
&= \mathrm{E}\left[\frac{X^2}{\frac{V}{K}}\right] \\
&= \mathrm{E}\left[k \cdot X^2\right] \cdot \mathrm{E}\left[\frac{1}{V}\right] \\
&= k \cdot \mathrm{Var}[X] \cdot \frac{1}{k-2} \\
&= \frac{k}{k-2} = \mathrm{Var}[\tilde{Y}]
\end{aligned}$$

Now, define

$$W := \frac{1}{\sqrt{\frac{V}{k}}}$$

then

$$\mathrm{corr}[\tilde{X}, \tilde{Y}] \equiv \mathrm{corr}\left[\frac{X}{\sqrt{\frac{V}{k}}}, \frac{Y}{\sqrt{\frac{V}{k}}}\right]$$

Combining this with part (a), we see:

$$\mathrm{corr}[\tilde{X}, \tilde{Y}] = r$$

Question 7

We run the following code to set the scene:

```
set.seed(92020);
X <- rnorm(1000); V <- rnorm(1000)
a <- .8 ; b <- .6 ; Y <- a*X + b*V
chi5 <- sqrt(rchisq(1000, df=5)/5)
X. <- X/chi5; Y. <- Y/chi5
```

To verify our sample results versus their theoretical counterparts, run:

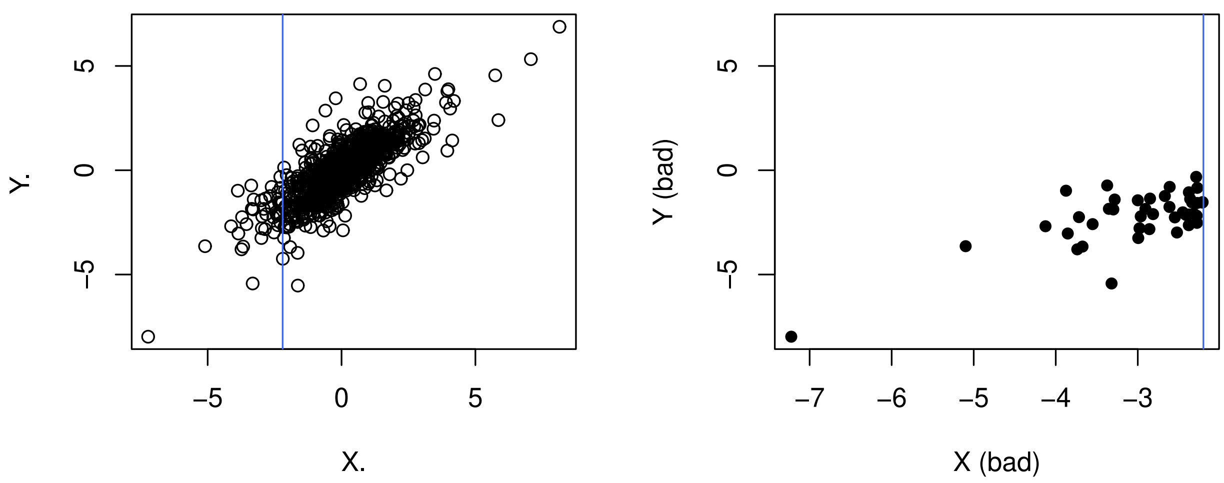
```
empirical <- c(mean(X.), mean(Y.), var(X.), var(Y.), cor(X.,Y.));
theoretical <- c(0, 0, 5/3, 5/3, 0.8);
error.term <- abs(empirical-theoretical)
tabled.vals <- round(rbind(empirical, theoretical, error.term), 3)
print(tabled.vals)
             [,1]  [,2]  [,3]  [,4]  [,5]
empirical   0.039 0.007 1.791 1.760 0.804
theoretical 0.000 0.000 1.667 1.667 0.800
error.term  0.039 0.007 0.125 0.093 0.004
```

Once again, the results are close.

Question 8

Plotting our bivariate student sample:

```
par(mfrow=c(1,2))
plot(X., Y., pch=1)
d <- -2.2
abline(v=d, col="royalblue")
bad <- (X. < d)
plot(X.[bad], Y.[bad], pch=16, ylim=range(Y.),
      xlab = "X (bad)", ylab = "Y (bad)")
abline(v=d, col="royalblue")
cor(X.[bad], Y.[bad])
[1] 0.7108746
```



The `cor(X.[bad], Y.[bad])` figure of 0.71 is much nearer the theoretical value of 0.8, as opposed to the figure of 0.01 observed under the bivariate normal case. Clearly, the bivariate student distribution does display greater degree of tail dependence.

Question 9

We run the following code having filled in the `Covat` matrix:

```
library(MASS)
mu <- c(1,3,5); sig2 <- c(1,2,5);

Corrmat <- rbind(c(1.0, 0.3, 0.3),
                 c(0.3, 1.0, 0.4),
                 c(0.3, 0.4, 1.0));

Covmat <- rbind(c(1.0, 0.3 * sqrt(2), 0.3 * sqrt(5)),
                c(0.3 * sqrt(2), 2.0, 0.4 * sqrt(10)),
                c(0.3 * sqrt(5), 0.4 * sqrt(10), 5.0));

XYZ <- mvrnorm(100, mu, Covmat)
```

We then verify the results using:

```
marg.x <- XYZ[,1]
marg.y <- XYZ[,2]
marg.z <- XYZ[,3]
empirical <- c(mean(marg.x), mean(marg.y), mean(marg.z),
               var(marg.x), var(marg.y), var(marg.z),
               cor(marg.x, marg.y), cor(marg.x, marg.z),
               cor(marg.y, marg.z));
theoretical <- c(1, 3, 5, 1, 2, 5, .3, .3, .4);
error.term <- abs(empirical-theoretical)
tabled.vals <- round(rbind(empirical, theoretical, error.term), 3)
print(tabled.vals)
```

which yields:

```
             [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]
empirical   0.961 2.908 4.589 1.117 2.158 5.654 0.307 0.427 0.346
theoretical 1.000 3.000 5.000 1.000 2.000 5.000 0.300 0.300 0.400
error.term  0.039 0.092 0.411 0.117 0.158 0.654 0.007 0.127 0.054
```

Our sample of size 100 has performed relatively well.

Question 10

(a) We run:

```
set.seed(779);
n <- 1e6
mu <- c(1,3,5); sig2 <- c(1,2,5);
Corrmat <- rbind(c(1.0, 0.3, 0.3),
                 c(0.3, 1.0, 0.4),
                 c(0.3, 0.4, 1.0));

Covmat <- rbind(c(1.0, 0.3 * sqrt(2), 0.3 * sqrt(5)),
                c(0.3 * sqrt(2), 2.0, 0.4 * sqrt(10)),
                c(0.3 * sqrt(5), 0.4 * sqrt(10), 5.0));

XYZ <- mvrnorm(n, mu, Covmat)

linear.xyz <- rowSums(XYZ)

a <- c(1, 1, 1)
y.mu <- t(a) %*% mu
y.sig2 <- t(a) %*% Covmat %*% a
empirical <- c(mean(linear.xyz), var(linear.xyz));
theoretical <- c(y.mu, y.sig2);
error.term <- abs(empirical-theoretical)
tabled.vals <- round(rbind(empirical, theoretical, error.term), 3)
print(tabled.vals)

xyz.at.risk <- quantile(linear.xyz, probs = c(.9999))
xyz.at.risk
```

Note: I have defined `linear.xyz` to be the linear combination $Y := X_1 + X_2 + X_3$ in lieu of $X + Y + Z$.

So, our sample estimates of μ_y and σ_y^2 are:

```
             [,1]   [,2]
empirical   9.002 12.707
theoretical 9.000 12.720
error.term  0.002  0.013
```

And the quantile estimate (`xyz.at.risk`) for $F_Y^{-1}(0.9999) =$ `22.26805`

(b) Repeating this process for 10 different seeds yields estimates of $F_Y^{-1}(0.9999)$ which I have stored in the vector `quantile.xyz`:

```
quantile.xyz <- c(22.36249, 22.34873, 22.27201,
                  22.19473, 22.30522, 22.29585, 22.29907,
                  22.33885, 22.16892, 22.26805)

table.xyz <- round(rbind(mean(quantile.xyz), var(quantile.xyz)),3)
print(table.xyz)
       [,1]
[1,] 22.285
[2,]  0.004
```

The precision reached is quite remarkable.

(c) The theoretical value of this quantile can be computed with:

```
qnorm(.9999, mean = y.mu, sd = sqrt(y.sig2))
```

which evaluates to: `[1] 22.26391`

This confirms that our estimate in **(b)** is accurate.

Question 11

(a) We run:

```
n <- 10^6
mu <- c(0,0,0);
Covmat <- rbind(c(1.0, 1/6, 1/6),
                c(1/6, 1.0, 1/6),
                c(1/6, 1/6, 1.0));
XYZ <- mvrnorm(n, mu, Covmat)
```

(b) Compute $S_i = X_i + Y_i + Z_i, i = 1, \ldots, n$

We use:

```
S <- rowSums(XYZ)
```

and

```
d <- quantile(S, probs = c(.975))
```

and get $d = F_S^{-1}(0.975) = 3.912093$.

(c) Code:

```
stop.loss <- pmax(S, d) - d
stop.loss <- stop.loss[which(stop.loss!=0)]
premium <- mean(stop.loss)
```

with output `[1] 0.7490712`.

Question 12

(a) Suppose $X_1, \ldots, X_n$ are jointly multivariate normally distributed.

Consider the linear transformation $\mathbf{Y} = \mathbf{a}^T\mathbf{X}$, where $\mathbf{a} = (a_1, \ldots, a_n)^T$, $\mathbf{X} = (X_1, \ldots, X_n)^T$. It can be shown that:

$$\mathbf{Y} = \mathbf{a}^T\mathbf{X} \sim N(\mathbf{a}^T\mu, \mathbf{a}^T\mathbf{\Sigma}\mathbf{a})$$

It follows that $S_i \sim N(0, 4)$.

To verfiy, let's run:

```
a <- c(1, 1, 1)
s.mu <- t(a) %*% mu
s.sig2 <- t(a) %*% Covmat %*% a
empirical <- c(mean(S), var(S));
theoretical <- c(s.mu, s.sig2);
error.term <- abs(empirical-theoretical)
tabled.vals <- round(rbind(empirical, theoretical, error.term), 3)
print(tabled.vals)
```

which shows:

```
              [,1]  [,2]
empirical   -0.003 4.007
theoretical  0.000 4.000
error.term   0.003 0.007
```

so $S_i \sim N(0, 4)$ is correct.

(b) The true value of the stop-loss premium is calculated using:

```
# true quantile
theoretical.q <- qnorm(.975, mean = 0, sd = 2) # [1] 3.919928
# using formula (3.104)
theoretical.prem <- (sqrt(s.sig2) *
  dnorm((theoretical.q-s.mu)/sqrt(s.sig2), mean = 0, sd = 2)) -
  ((theoretical.q - s.mu) *
  (1 - pnorm((theoretical.q-s.mu)/sqrt(s.sig2), mean = 0, sd = 2)))
```

which yields `-0.3942806`.