Tutorial - Week 8

Please read the related material and attempt these questions before attending your allocated tutorial. Solutions are released on Friday 4pm.

Question 1

(a) Show that sampling n observations from a multivariate normal $N_p(\mu, \Sigma)$ distribution and then calculating a sample covariance matrix is equivalent to sampling from a Wishart distribution using the 'rWishart' function. Consider the distribution of the eigenvalues to show this. How are the function parameters 'df' and 'Sigma' related to the multivariate normal distribution?

Solution: We load a library to simulate from the multivariate normal distribution.

```
#install.packages('mvnfast')
library(mvnfast)
```

We sample from a multivariate normal, calculate the sample covariance, and then the eigenvalues of the

```
p = 500
n = 1000
mu = rep(0, p)
Sigma = diag(p)
X = rmvn(n, mu, Sigma)
S = cov(X)
L = eigen(S, only.values = TRUE)$values
```

We will use a kernel density estimation instead of a histogram.

```
f = density(L)
```

We now sample using a Wishart, notice that we need to scale the random matrix A appropriately to obtain S. See Lecture Week 8, page 6.

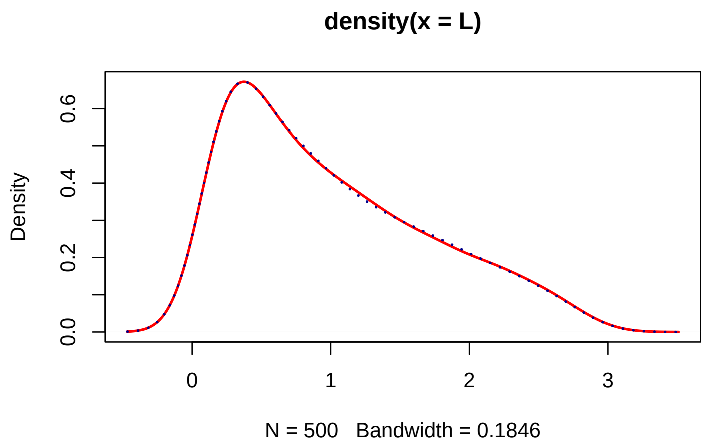
```
A = rWishart(1, n, Sigma)[,,1]
S = A/(n-1)
M = eigen(S, only.values = TRUE)$values
```

We now generate a density for the eigenvalues of the (scaled) Wishart matrix.

```
g = density(M)
```

We plot both eigenvalue densities and we get a good match.

```
plot(f, col="red", lwd=2)
lines(g, col="darkblue", lwd=2, lty=3)
```



We see that `df` should be set to the number n and $Sigma$ to Σ.

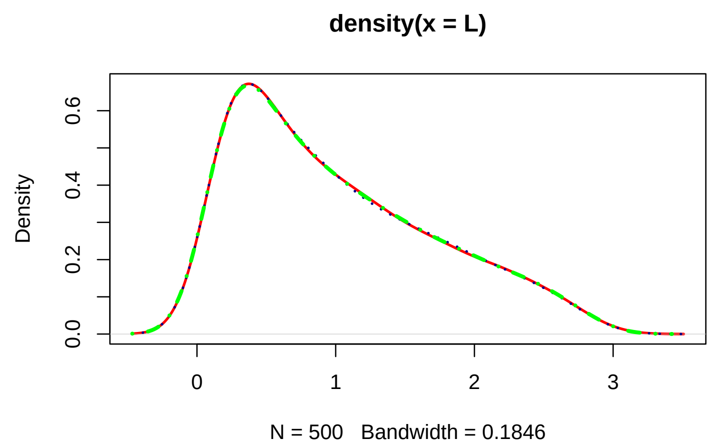
(b) Write a function that samples from the Wishart distribution $W_p(n, I_p)$ using the Bartlett decomposition. That is, sample a lower-triangular matrix T where the off-diagonals are distributed like $T_{ij} \sim N(0, 1)$ and the diagonal entries T_{ii} satisfy $T_{ii}^2 \sim \chi^2_{n-i+1}$. Then $U = TT' \sim W_p(n, I_p)$.

Solution: There are $p(p-1)/2$ lower triangular entries so we can use the `lower.tri` function and generate $p(p-1)/2$ standard normals to fill those entries. Then, we sample p samples from χ^2 for the diagonal.

```
p = 500
n = 1000

rwishart = function(n, p) {
  TT = matrix(0, p, p)
  TT[lower.tri(TT)] = rnorm(p * (p - 1) / 2)
  diag(TT) = sqrt(rchisq(p, n-1:p+1))
  TT %*% t(TT)
}

S = rwishart(n, p) / (n-1)
N = eigen(S, only.values = TRUE)$values

h = density(N)
```

We compare it to the results of the last question.

```
plot(f, col="red", lwd=2)
lines(g, col="darkblue", lwd=2, lty=3)
lines(h, col="green", lwd=3, lty=4)
```



(c) Write a more general function that handles the case where the population matrix Σ is not the identity. Do this by setting $W = \Sigma^{1/2}U(\Sigma^{1/2})'$ where $\Sigma^{1/2}(\Sigma^{1/2})' = \Sigma$, to obtain $W \sim W_p(n, \Sigma)$. Test in the case where $\Sigma = I_p$ and the case where Σ has a compound symmetric covariance structure (see last week's tutorial).

Solution: We revise our previous function to take a parameter Σ. We use the results of Workshop week 3 to take the square-root of the matrix Σ.

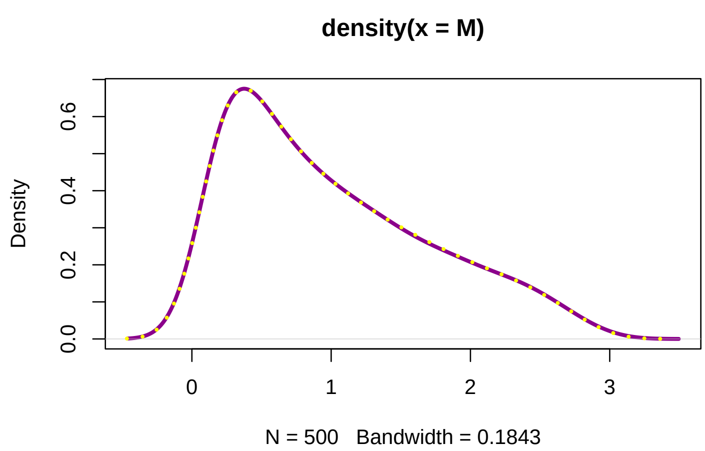
```
rwishart = function(n, p, Sigma) {
  TT = matrix(0, p, p)
  TT[lower.tri(TT)] = rnorm(p * (p - 1) / 2)
  diag(TT) = sqrt(rchisq(p, n-1:p+1))
  U = TT %*% t(TT)

  e = eigen(Sigma)
  P = e$vectors
  L = e$values
  Sigma.sqrt = P %*% diag(sqrt(L)) %*% t(P)
  Sigma.sqrt %*% U %*% t(Sigma.sqrt)
}
```

We first test if we get the correct result in the case where $\Sigma = I_p$.

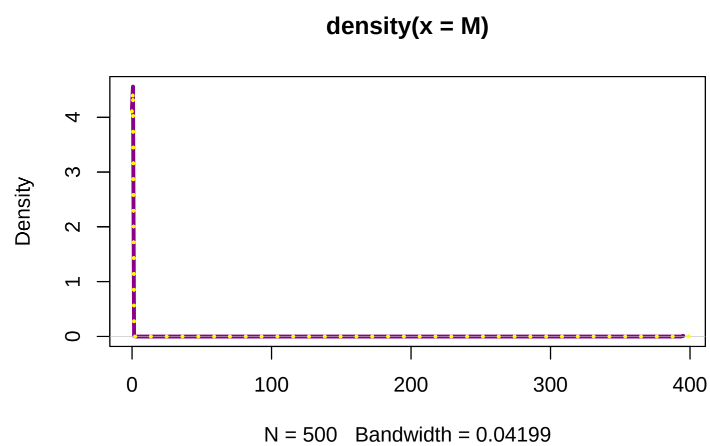
```
p = 500
n = 1000

Sigma = diag(p)

A = rWishart(1, n, Sigma)[,,1]
S = A/(n-1)
M = eigen(S, only.values = TRUE)$values
f = density(M)

S = rwishart(n, p, Sigma) / (n-1)
N = eigen(S, only.values = TRUE)$values
g = density(N)
```

We get the same.

```
plot(f, col="darkmagenta", lwd=3)
lines(g, col="yellow", lwd=3, lty=3)
```



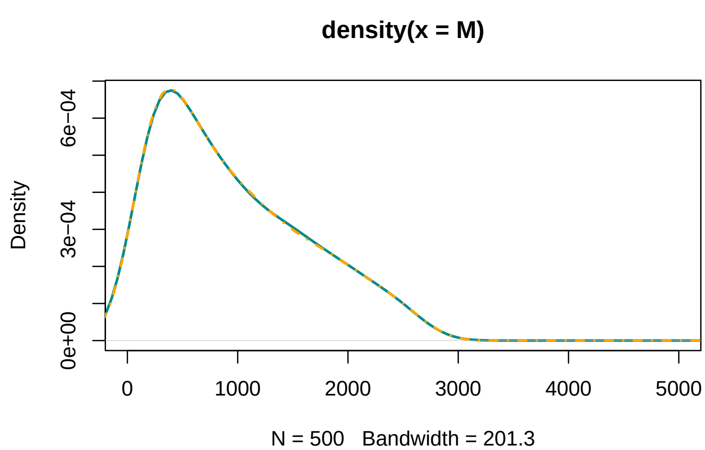
```
p = 500
n = 1000

rho = 0.8
# Compound symmetric covariance structure
Sigma = (1 - rho) * diag(p) + rho * outer(rep(1, p), rep(1, p))

A = rWishart(1, n, Sigma)[,,1]
S = A/(n-1)
M = eigen(S, only.values = TRUE)$values
f = density(M)

S = rwishart(n, p, Sigma) / (n-1)
N = eigen(S, only.values = TRUE)$values
g = density(N)

plot(f, col="darkmagenta", lwd=3)
lines(g, col="yellow", lwd=3, lty=3)
```



(d) Show through a simulation that if $U_1 \sim W_p(n_1, \Sigma)$ and $U_2 \sim W_p(n_2, \Sigma)$, then $U_1 + U_2 \sim W_p(n_1 + n_2, \Sigma)$.

Solution: We write it for a general matrix Σ.

```r
p = 500
n1 = 500
n2 = 500

rho = 0.05
# Compound symmetric covariance structure
Sigma = (1 - rho) * diag(p) + rho * outer(rep(1, p), rep(1, p))

U1 = rWishart(1, n1, Sigma)[,,1]
U2 = rWishart(1, n2, Sigma)[,,1]

W = rWishart(1, n1+n2, Sigma)[,,1]

M = eigen(U1+U2, only.values = TRUE)$values
Q = eigen(W, only.values = TRUE)$values

f = density(M)
h = density(Q)

plot(f, col="darkcyan", lwd=2, xlim=c(0,5000))
lines(h, col="orange", lwd=2, lty=4)
```



(e) Show through a simulation that if $U \sim W_p(n, \Sigma)$ and A is a constant $p \times p$ matrix, then $\mathbb{E}[\text{tr}(AU)] = n\,\text{tr}(A\Sigma)$.

Solution: We are going to construct some arbitrary matrix Σ.

```r
rho = 0.05
# Compound symmetric covariance structure
Sigma = (1 - rho) * diag(p) + rho * outer(rep(1, p), rep(1, p))
```

And now construct an arbitrary matrix A.

```r
A = matrix(runif(p*p), p, p) # generate a random matrix A
```

We create a function that calculates the trace of a matrix.

```r
tr = function(X) sum(diag(X))
```

We can now perform a simulation whereby we sample from a Wishart and calculate an empirical version of the LHS of the identity and then compare it to the RHS. We see that the two values are close. The difference is due to the fact that we are approximating the LHS by an empirical approximation.

```r
nsims = 500
p = 500
n = 1000

u = replicate(nsims, {
          U = rWishart(1, n, Sigma)[,,1]
          tr(A %*% U)
})

c(mean(u), n * tr(A %*% Sigma))
```

```
## [1] 6476729 6477175
```

(f) Show through a simulation that if $U \sim W_p(n, \Sigma)$ and A is a constant $p \times p$ matrix, then $\mathbb{E}[\mathrm{tr}(AU^{-1})] = \frac{1}{n-p-1}\mathrm{tr}(A\Sigma^{-1})$ for $n - p - 1 > 0$.

Solution: Again, we are going to construct some arbitrary matrix Σ.

```r
rho = 0.05
# Compound symmetric covariance structure
Sigma = (1 - rho) * diag(p) + rho * outer(rep(1, p), rep(1, p))
```

And now construct an arbitrary matrix A.

```r
A = matrix(runif(p*p), p, p) # generate a random matrix A
```

We perform a similar simulation to the last question but now consider the inverse of the matrix U. Remember the inverse of the matrix is obtained using the function `solve`.

```
nsims = 100
p = 500
n = 2000

u = replicate(nsims, {
          U = rWishart(1, n, Sigma)[,,1]
          tr(A %*% solve(U))
})

c(mean(u), tr(A %*% solve(Sigma))/(n-p-1))
```

```
## [1] 0.001295464 0.002017835
```

(g) Show through a simulation that if $U \sim W_p(n, \Sigma)$ then

$$\mathbb{E}[|U|^h] = 2^{ph}|\Sigma|^h \prod_{i=1}^{p} \frac{\Gamma(\frac{1}{2}(n-i+1)+h)}{\Gamma(\frac{1}{2}(n-i+1))}.$$

Initially take $p = 2$, comment on what happens when p becomes larger.

Solution: We implement a function to calculate the determinant of a matrix.

```
det = function(x) as.numeric(determinant(x, logarithm = FALSE)$modulus)
```

We sample from the Wishart distribution, calculate the determinant of all the samples, then calculate the sample mean of those values, i.e., we do the case $h = 1$.

```
h = 1
p = 2
n = 100
Sigma = diag(p)
A = rWishart(1000, n, Sigma)
mean(apply(A, 3, det))
```

```
## [1] 10028.79
```

We now calculate the formula. We use the shortcut `1:p` to generate a vector of terms and `prod` to take the product of those terms.

```
2^(p * h) * det(Sigma)^h * prod(gamma(0.5*(n-1:p+1) + h) / gamma(0.5*(n-1:p+1)))
```

```
## [1] 9900
```