Tutorial - Week 6

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

Question 1

The Fisher limiting spectral distribution (LSD), denoted $F_{s,t}$, has the density function

$$f_{s,t}(x) := \frac{1-t}{2\pi x(s+tx)}\sqrt{(b-x)(x-a)}, \quad a \le x \le b,$$

where

$$a := a(s,t) := \frac{(1-h)^2}{(1-t)^2}, \quad b := b(s,t) := \frac{(1+h)^2}{(1-t)^2}, \quad h := h(s,t) := (s+t-st)^{1/2}.$$

Suppose we had two independent p-dimensional vector samples $\mathbb{X} := \{\mathbb{x}_1, \ldots, \mathbb{x}_{n_1}\}$ and $\mathbb{Y} := \{\mathbb{y}_1, \ldots, \mathbb{y}_{n_2}\}$ where $p \le n_2$. We assume that each sample comes from a (possibly different) population distribution with i.i.d. components and finite second moment.

(a) How is the Fisher LSD related to the two vector samples $\mathbb{X}$ and $\mathbb{Y}$? What is the relationship between the two parameters (s,t) of $F_{s,t}$ and the three values (p, n_1, n_2) describing the dimensionality and sizes of $\mathbb{X}$ and $\mathbb{Y}$?

> **Solution:** Let $\mathbb{S}_1$ be the sample covariance matrix of $\mathbb{X}$ and $\mathbb{S}_2$ be the sample covariance of $\mathbb{Y}$. Then we can define the random matrix $\mathbb{F} = \mathbb{S}_1\mathbb{S}_2^{-1}$. Let $p/n_1 \to y_1 > 0$ and $p/n_2 \to y_2 \in (0,1)$ then the empirical spectral distribution (ESD) of $\mathbb{F}$ coverges to the Fisher LSD F_{y_1,y_2}, i.e., $s = y_1$ and $t = y_2$.

(b) Now that you explained the relationship between $\mathbb{X}$, $\mathbb{Y}$, the values (p, n_1, n_2) and the parameters (s,t) of $F_{s,t}$ in part (a), what would you expect the empirical density of eigenvalues be for the following three choices of triplets (p, n_1, n_2):

$$(50, 100, 100), \qquad (75, 100, 200), \qquad (25, 100, 200).$$

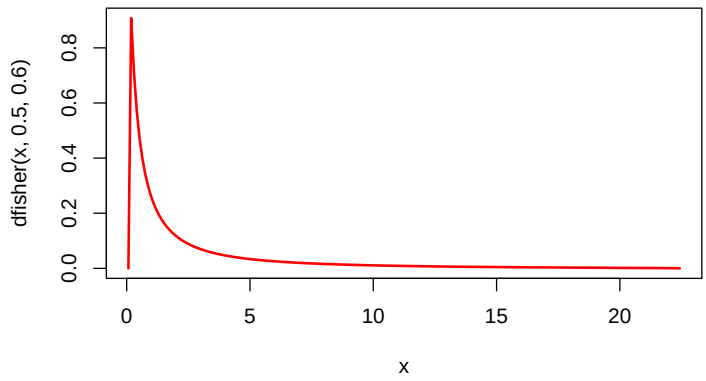
Plot the three densities on the same figure with an appropriate legend.

> **Solution:** We start by implementing the Fisher density.
>
> ```
> dfisher = Vectorize(function(x, s, t) {
>   h = sqrt(s+t-s*t)
>   a = (1-h)^2/(1-t)^2
>   b = (1+h)^2/(1-t)^2
>   ifelse(x <= a | x >= b, 0, suppressWarnings(sqrt((x - a) * (b - x))*(1-t)/(2 * pi * x *(s+t*x))))
> },"x")
> ```
>
> To make plotting easier, we define a function that returns the support of the density for a given choice of parameter values.

```
fisher.support = function(s, t) {
  h = sqrt(s+t-s*t)
  a = (1-h)^2 / (1-t)^2
  b = (1+h)^2 / (1-t)^2
  list("a"=a, "b"=b)
}
```

We can now plot the density

```
supp = fisher.support(0.5, 0.6)
x = seq(supp$a, supp$b, length.out=200)
plot(x, dfisher(x, 0.5, 0.6), type='l', lwd=2, col="red")
```



We now plot each of the densities, but first we determine the best support to plot over. We convert the parameters to values of y_1 and y_2. We do this as vectors.

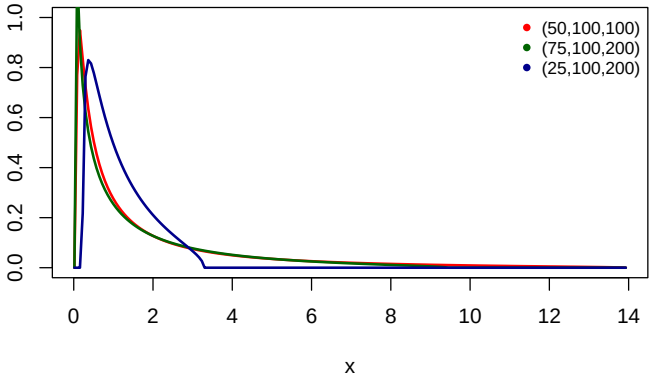
```
p = c(50, 75, 25)
n1 = c(100, 100, 100)
n2 = c(100, 200, 200)
y1 = p/n1
y2 = p/n2
```

We now find the largest support needed based on the minimum a value and maximum b value.

```
supp = mapply(fisher.support, y1, y2)
a = min(as.numeric(supp["a",]))
b = max(as.numeric(supp["b",]))
```

We now plot the figure.

```
x = seq(a, b, length.out=200)
plot(x, dfisher(x, y1[1], y2[1]), type='l', lwd=2, col="red", ylab="", ylim=c(0,1))
lines(x, dfisher(x, y1[2], y2[2]), type='l', lwd=2, col="darkgreen", ylab="")
lines(x, dfisher(x, y1[3], y2[3]), type='l', lwd=2, col="darkblue", ylab="")
legend("topright", c("(50,100,100)", "(75,100,200)","(25,100,200)"), col=c("red", "darkgreen","darkblue"),
```



(c) Perform a simulation study for the same choices of the triplets (p, n_1, n_2) that are given in part (b). Setup your experiment correctly to demonstrate a histogram of eigenvalues and compare them to the appropriately parametrised densities from part (b). For each triplet (p, n_1, n_2), plot the histogram of eigenvalues, overlay the appropriate density, and ensure the plot is appropriately titled. Show the code for your simulation study.

Solution: We will simulate multivariate normal data, generate sample covariances, and then generate the sample Fisher matrix FF. Then we calculate the eigenvalues of FF.

```
library(mvnfast)
```

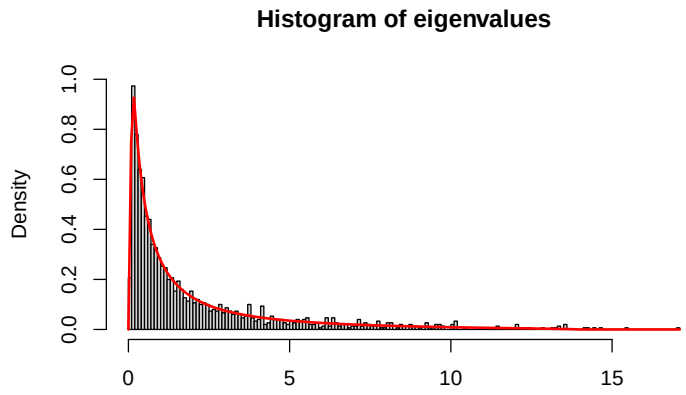
We write a function that samples the eigenvalues.

```
sim.eigenvalues = function(p, n1, n2, m=1) {
  evalues = c()
  for (sim in 1:m) {
    Sigma1 = diag(p)
    Sigma2 = diag(p)
    mu = rep(0, p)
    X1 = rmvn(n1, mu, Sigma1)
    X2 = rmvn(n2, mu, Sigma2)
    S1 = cov(X1)
    S2 = cov(X2)
    FF = S1 %*% solve(S2)
    evalues = c(evalues, eigen(FF, only.values = TRUE)$values)
  }
  evalues
}
```

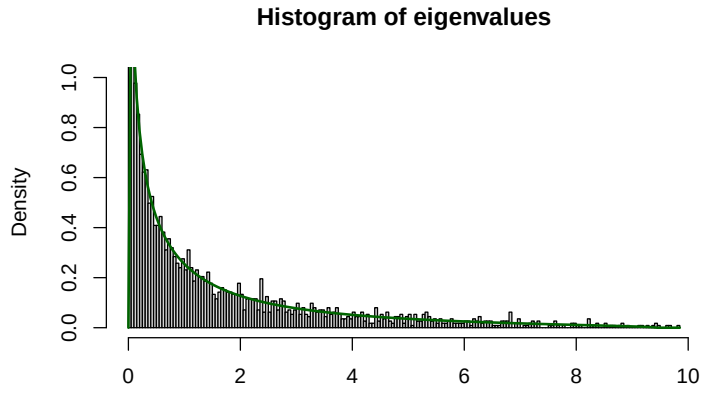
The parameter m in the above code allows us to simulate the eigenvalues m times and add them

all to the histogram. When the `hist` function shows a density (`freq=FALSE`) it takes the number of values (frequency) in each bucket and then divides by the total number. Therefore, if we simulate m times we get $m * p$ eigenvalues but then they are normalised by dividing through by $m * p$. The result is that, when $m > 1$, we get a sort of "averaging" over the m histograms (you can argue this by looking at each histogram bucket). This allows us to show a nicer histogram picture.

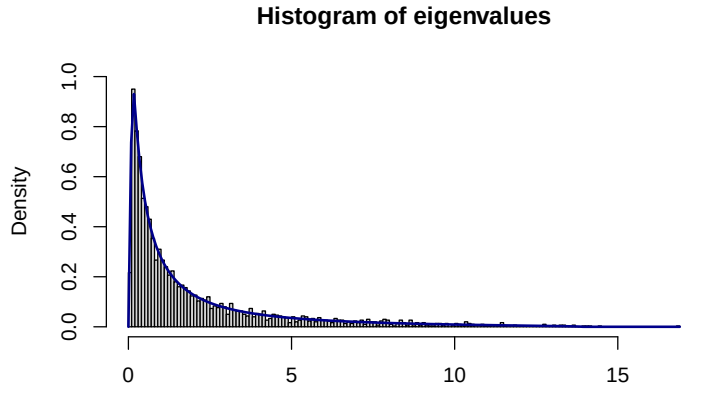
```
p = 50
n1 = 100
n2 = 100
hh = hist(sim.eigenvalues(p, n1, n2, 30), 200, freq=FALSE,
          ylim=c(0,1), main="Histogram of eigenvalues", xlab="")
x = seq(hh$breaks[1], hh$breaks[length(hh$breaks)], length.out=200)
lines(x, dfisher(x, p/n1, p/n2), type='l', lwd=2, col="red", ylab="")
```




```
p = 75
n1 = 100
n2 = 200
hh = hist(sim.eigenvalues(p, n1, n2, 30), 200, freq=FALSE,
          ylim=c(0,1), main="Histogram of eigenvalues", xlab="")
x = seq(hh$breaks[1], hh$breaks[length(hh$breaks)], length.out=200)
lines(x, dfisher(x, p/n1, p/n2), type='l', lwd=2, col="darkgreen", ylab="")
```




```
p = 100
n1 = 200
n2 = 200
hh = hist(sim.eigenvalues(p, n1, n2, 30), 200, freq=FALSE,
          ylim=c(0,1), main="Histogram of eigenvalues", xlab="")
x = seq(hh$breaks[1], hh$breaks[length(hh$breaks)], length.out=200)
lines(x, dfisher(x, p/n1, p/n2), type='l', lwd=2, col="darkblue", ylab="")
```



(d) The first moment of the Fisher LSD in terms of its parameters s and t is given by

$$\int_a^b x\, f_{s,t}(x)\, dx = \frac{1}{1-t}.$$

Perform a numerical experiment to confirm this formula. That is, choose 3 values of (s, t) then use your simulation study code (from part c) to generate empirical eigenvalues for those values and calculate their sample mean. Compare the sample means to the formula.

Solution: We perform the simulation for 3 values and compare to the mean $1/(1-t)$.

```
p = 50
n1 = 1000
n2 = 1000
t = p/n2 # y2
c(mean(sim.eigenvalues(p, n1, n2)), 1/(1-t))
```

```
## [1] 1.050041 1.052632
```

```
p = 500
n1 = 1000
n2 = 1000
t = p/n2 # y2
c(mean(sim.eigenvalues(p, n1, n2)), 1/(1-t))
```

```
## [1] 1.994263 2.000000
```

```
p = 750
n1 = 1000
n2 = 1000
t = p/n2 # y2
c(mean(sim.eigenvalues(p, n1, n2)), 1/(1-t))
```

```
## [1] 4.073784 4.000000
```

(e) Describe what happens to the first moment when $t \to 0$ and $t \to 1$. Explain the $t \to 1$ case in terms of p, n_1, and n_2.

Solution: When t approaches zero, the mean approaches 1. When t approaches 1, then mean approaches infinity. This occurs when p approaches n_2.

Question 2

The Bartlett statistic, see **[A]** page 413 Eq. (10)[1], is for $g = 2$ given by

$$V_1 = \frac{|\mathbb{A}_1|^{N_1/2}|\mathbb{A}_2|^{N_2/2}}{|\mathbb{A}_1 + \mathbb{A}_2|^{N/2}}$$

where $N_g := n_g - 1$ and $N := N_1 + N_2$. Setting $\mathbb{S}_g = \mathbb{A}_g/N$, multiplying through the numerator and denominator by $|\mathbb{S}_2^{-1}|$ and using the fact that $|AB| = |A||B|$ for matrices A and B, we can instead consider

$$V_1^* = \frac{|\mathbb{S}_1\mathbb{S}_2^{-1}|^{N_1/2}}{|c_1\mathbb{S}_1\mathbb{S}_2^{-1} + c_2 I_p|^{N/2}}$$

where $c_g = N_g/N$ and I_p is the identity matrix of size $p \times p$. Notice we are in the Fisher regime $\mathbb{S}_1\mathbb{S}_2^{-1}$.

[1] See `2003-Anderson-Book-Chapter10` in Readings on Wattle.

(a) Sample the distribution of V_1^* for $p = 3$, $n_1 = 100$, $n_2 = 100$. Plot the histogram of the distribution when you sample $m = 500$ times.

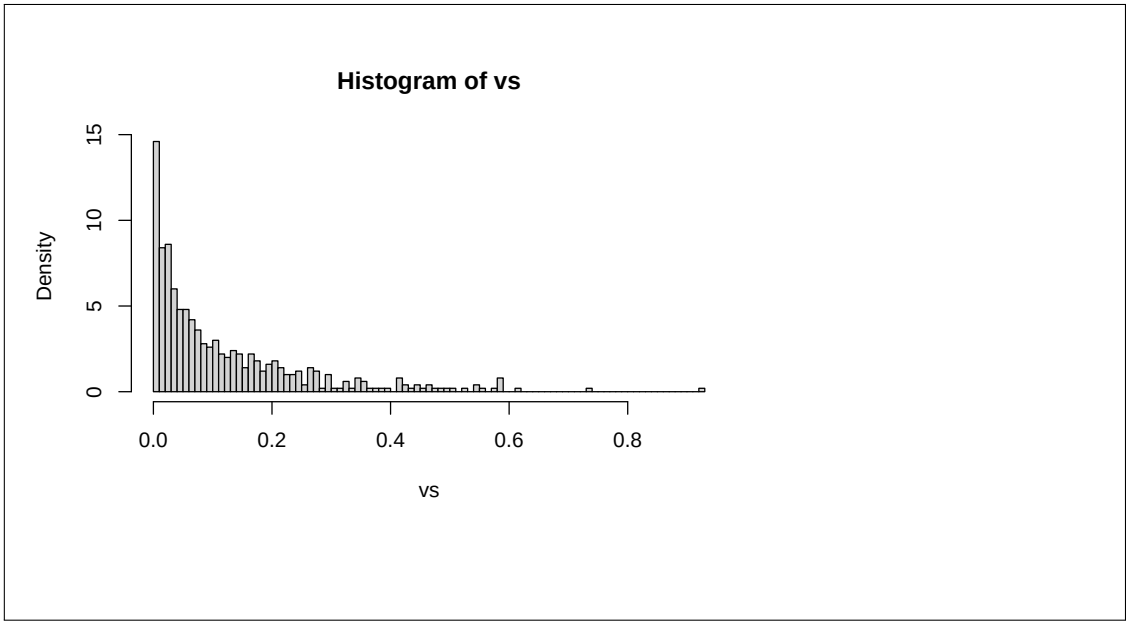
Solution: We modify our code from Question 1 to simulate V_1^*. Here, m is the number of samples we generate.

```
sim.V1.star = function(p, n1, n2, m=100) {
  N1 = n1 - 1
  N2 = n2 - 1
  N = N1 + N2
  c1 = N1/N
  c2 = N2/N
  Ip = diag(p)
  values = c()
  for (sim in 1:m) {
    Sigma1 = diag(p)
    Sigma2 = diag(p)
    mu = rep(0, p)
    X1 = rmvn(n1, mu, Sigma1)
    X2 = rmvn(n2, mu, Sigma2)
    S1 = cov(X1)
    S2 = cov(X2)
    FF = S1 %*% solve(S2)
    numer = det(FF)^(N1/2)
    denom = det(c1*FF + c2*Ip)^(N/2)
    values = c(values, numer/denom)
  }
  values
}
```

We now simulate them and plot a histogram.

```
m = 500
p = 3
n1 = 100
n2 = 100
vs = sim.V1.star(p, n1, n2, m)
```

```
hist(vs, 100, freq=FALSE)
```



(b) Show that if the observations from each $\mathbb{X}_1$ and $\mathbb{X}_2$ are transformed by

$$\mathbb{x}_i^* = C\mathbb{x}_i + \mu_i, \qquad i = 1, 2$$

where $\mu_1, \mu_2 \in \mathbb{R}^p$ and C is a $p \times p$ matrix, the distribution of V_1^* remains unchanged. You can do this with a simulation.

Solution: We modify our simulation code to take μ_1, μ_2 and C as parameters.

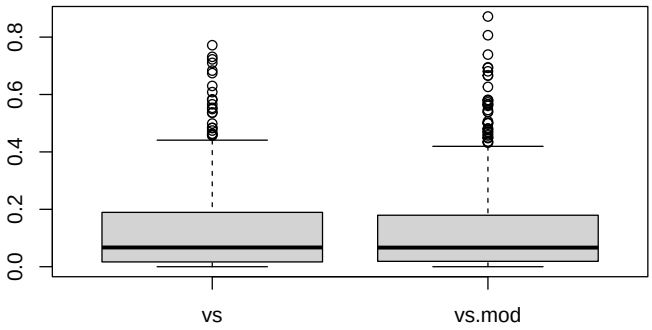
```
sim.V1.star.mod = function(p, n1, n2, mu1, mu2, C, m=100) {
  N1 = n1 - 1
  N2 = n2 - 1
  N = N1 + N2
  c1 = N1/N
  c2 = N2/N
  Ip = diag(p)
  values = c()
  for (sim in 1:m) {
    Sigma1 = diag(p)
    Sigma2 = diag(p)
    X1 = rmvn(n1, mu1, Sigma1) %*% C
    X2 = rmvn(n2, mu2, Sigma2) %*% C
    S1 = cov(X1)
    S2 = cov(X2)
    FF = S1 %*% solve(S2)
    numer = det(FF)^(N1/2)
    denom = det(c1*FF + c2*Ip)^(N/2)
    values = c(values, numer/denom)
  }
  values
}
```

We can choose various parameters here.

```
m = 500
p = 3
n1 = 100
n2 = 100
mu1 = rep(1, p)
mu2 = rep(10, p)
C = 0.2^abs(outer(1:p, 1:p, "-"))
vs = sim.V1.star(p, n1, n2, m)
vs.mod = sim.V1.star.mod(p, n1, n2, mu1, mu2, C, m)
```

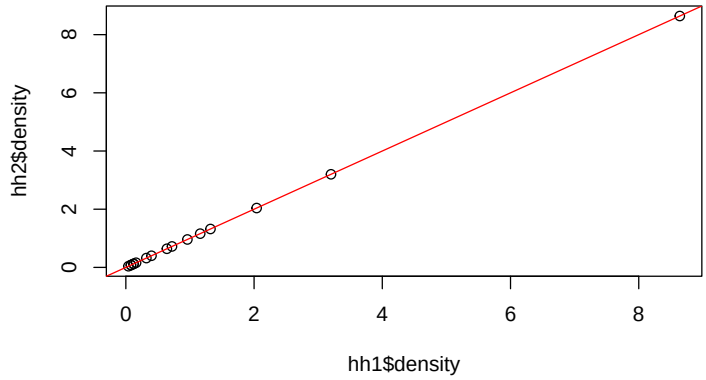
We can compare with a boxplot.

```
boxplot(cbind(vs,vs.mod))
```



We can create an empirical QQ-plot.

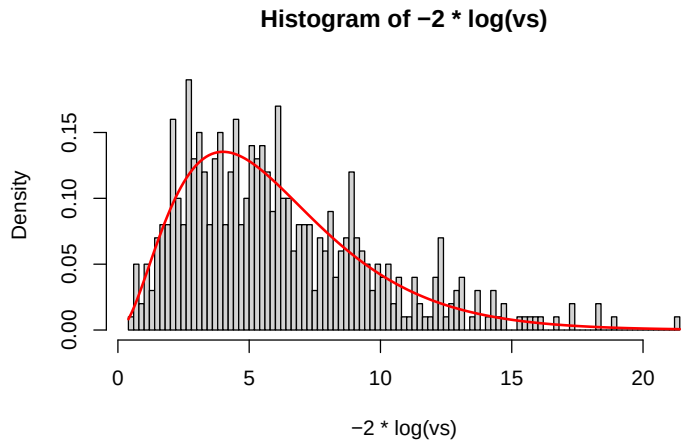
```
hh1 = hist(vs, "FD", plot=FALSE)
hh2 = hist(vs, hh1$breaks, plot=FALSE)
plot(hh1$density, hh2$density)
abline(a=0, b=1, col="red")
```



(c) Now consider the distribution of $-2\log(V_1^*)$ and compare it to the χ^2_ν distribution where $\nu = \frac{1}{2}p(p+1)$. Is it a good fit?

Solution: We transform the distribution of V_1^* and compare it to the χ^2_ν distribution where $\nu = \frac{1}{2}p(p+1)$.

```
hh = hist(-2 * log(vs), 100, freq=FALSE)
x = seq(hh$breaks[1], hh$breaks[length(hh$breaks)], length.out=200)
lines(x, dchisq(x, 1/2*p*(p+1)), lwd=2, col="red")
```

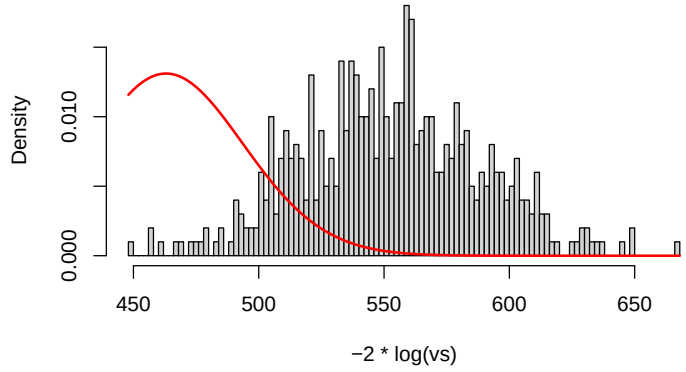


It is a good fit.

(d) What happens if you repeat (c) with $p = 30$? Is it still a good fit?

Solution: We increase p to 30.

```
m = 500
p = 30
n1 = 100
n2 = 100
vs = sim.V1.star(p, n1, n2, m)
hh = hist(-2 * log(vs), 100, freq=FALSE)
x = seq(hh$breaks[1], hh$breaks[length(hh$breaks)], length.out=200)
lines(x, dchisq(x, 1/2*p*(p+1)), lwd=2, col="red")
```



It is no longer a good fit.

(e) In **[B]** they show that[2] in the high-dimensional setting

Theorem. *Assume* $N_1 \to \infty$, $N_2 \to \infty$, *and* $p \to \infty$ *such that* $y_{N_1} = p/N_1 \to y_1 \in (0,1)$ *and* $y_{N_2} = p/N_2 \to y_2 \in (0,1)$. *Then*

$$-\frac{2}{N}\log V_1^* - p\, F_{y_{N_1}, y_{N_2}}(f) \to N(\mu_2, \sigma_2^2),$$

where

$$F_{a,b}(f) := \frac{a+b-ab}{ab}\log\left(\frac{a+b}{a+b-ab}\right) + \frac{a(1-b)}{b(a+b)}\log(1-b) + \frac{b(1-a)}{a(a+b)}\log(1-a),$$

and μ_2 *and* σ_2 *can be determined.*

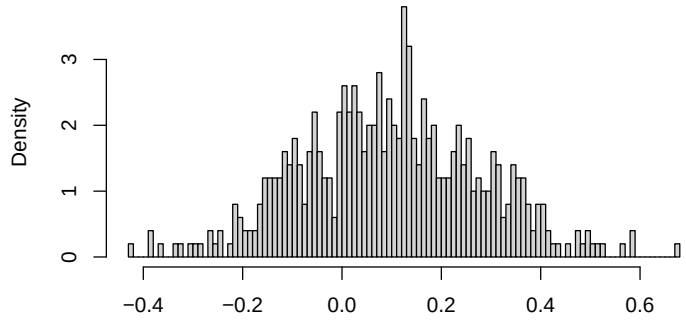
Perform this correction to the distribution of V_1^* in the case where $p = 30$. Does this look like a normal distribution?

Solution: We implement the function to calculate the correction.

[2] This is Theorem 4.1 in **[B]**.

```
F.correct = function(p, n1, p2) {
  N1 = n1 - 1
  N2 = n2 - 1
  yN1 = p/N1
  yN2 = p/N2
  N = N1 + N2
  a = yN1
  b = yN2
  (a+b-a*b)/(a*b) * log((a+b)/(a+b-a*b)) + (a*(1-b))/(b*(a+b)) * log(1-b) + (b*(1-a))/(a*(a+b)) * log(1-a)
}
```

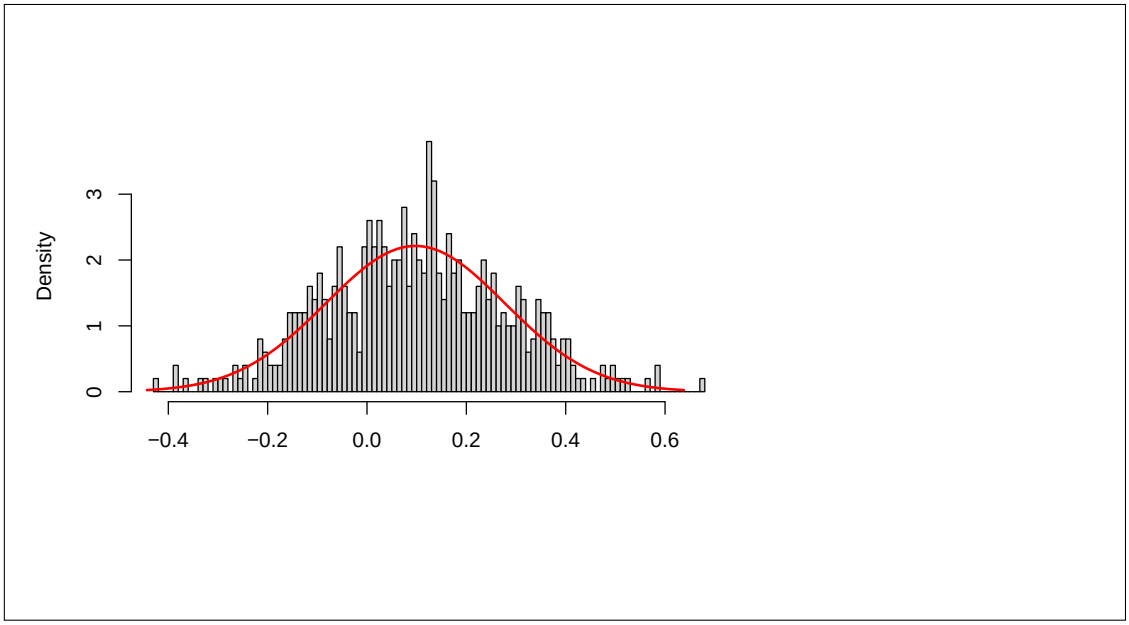
```
N1 = n1 - 1
N2 = n2 - 1
N = N1 + N2
vs.corrected = -2/N*log(vs) - p * F.correct(p, n1, n2)
hist(vs.corrected, 100, freq=FALSE, main='', xlab='')
```



(f) Empirically determine the correct μ_2 and σ_2^2 for the corrected V_1^* and use this to plot a normal distribution over the histogram. Does this fit better?

Solution:

```
mu = mean(vs.corrected)
sd = sd(vs.corrected)
hist(vs.corrected, 100, freq=FALSE, main='', xlab='')
x = seq(mu-3*sd, mu+3*sd, length.out=100)
lines(x, dnorm(x, mu, sd), col="red", lwd=2)
```



References

[A] Anderson (2003). An introduction to Multivariate Statistical Analysis. Wiley.

[B] Bai, Jiang, Yao, Zheng (2009). Corrections to LRT on large-dimensional covariance matrix by RMT. Annals of Statistics Vol 37, No. 6B, 3822–3840.