Workshop 7

- Correlation coefficient
  - Population correlation in bivariate case
  - Sample correlation distribution in bivariate case
- Multiple correlation coefficient
  - Population multiple correlation
  - Sample multiple correlation
  - Sample multiple correlation distribution
- High-dimensional regime
  - Over-estimation of R^2
  - Central limit theorem

Correlation coefficient

Population correlation in bivariate case

```
Sigma <- matrix(c(5,4,4,5), ncol=2)
Sigma
```

```
##      [,1] [,2]
## [1,]    5    4
## [2,]    4    5
```

Use `cov2cor` to scale the covariance matrix Σ by its diagonal to become the correlation matrix

$$R = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

```
R <- cov2cor(Sigma)
R
```

```
##      [,1] [,2]
## [1,]  1.0  0.8
## [2,]  0.8  1.0
```

Alternatively, we can compute ρ explicitly in the bivariate case from the formula

$$\rho = \frac{\sigma_{12}}{\sigma_1 \sigma_2}.$$

```
rho <- Sigma[1,2]/(sqrt(Sigma[1,1]*Sigma[2,2]))
```

Sample correlation distribution in bivariate case

We load the `mvtnorm` library for sampling from the multivariate normal distribution.

```
library(mvtnorm)
```

We sample N observations from a $\mathcal{N}(\mu, \Sigma)$ distribution.

```
N <- 100
mu <- c(0,0)
Sigma <- matrix(c(5,4,4,5), ncol=2)
X <- rmvnorm(N, mean=mu, sigma=Sigma)
```

We can calculate the sample covariance matrix “by hand”.

```
S <- t(X) %*% X / (N-1)
S
```

```
##          [,1]     [,2]
## [1,] 4.859903 3.905533
## [2,] 3.905533 4.881909
```

Or use the in-built function (which may be a bit more accurate).

```
S <- cov(X)
S
```

```
##          [,1]     [,2]
## [1,] 4.843735 3.912750
## [2,] 3.912750 4.878687
```

Then the sample correlation matrix can be obtained by converting the sample covariance.

```
cov2cor(S)
```

```
##           [,1]      [,2]
## [1,] 1.0000000 0.8048973
## [2,] 0.8048973 1.0000000
```

Or alternatively, by calculating it directly from the observations.

```
cor(X)
```

```
##           [,1]      [,2]
## [1,] 1.0000000 0.8048973
## [2,] 0.8048973 1.0000000
```

The sample correlation between the coordinates X_1 and X_2 is

$$R = \frac{S_{12}}{\sqrt{S_{11}S_{22}}}$$

```
R <- S[1,2]/sqrt(S[1,1]*S[2,2])
R
```

```
## [1] 0.8048973
```

We now look at the distribution in the case that $\rho = 0$ by sample N observations from a $\mathcal{N}(\mu, I)$ distribution.

```
p <- 2
Sigma <- diag(1, p, p)
mu <- rep(0, p)

M <- 5000 # number of MC simulations
n <- 1000 # number of observations

MC <- numeric(M)
for (i in 1:M) {
  X <- rmvnorm(n+1, mean=mu, sigma=Sigma)
  R <- cor(X)[1,2]
  MC[i] <- sqrt(n-1)*(R/sqrt(1-R**2))
}
```

Generate histogram of MC simulations.

```
hist(MC, breaks="FD", plot=F) -> h
```

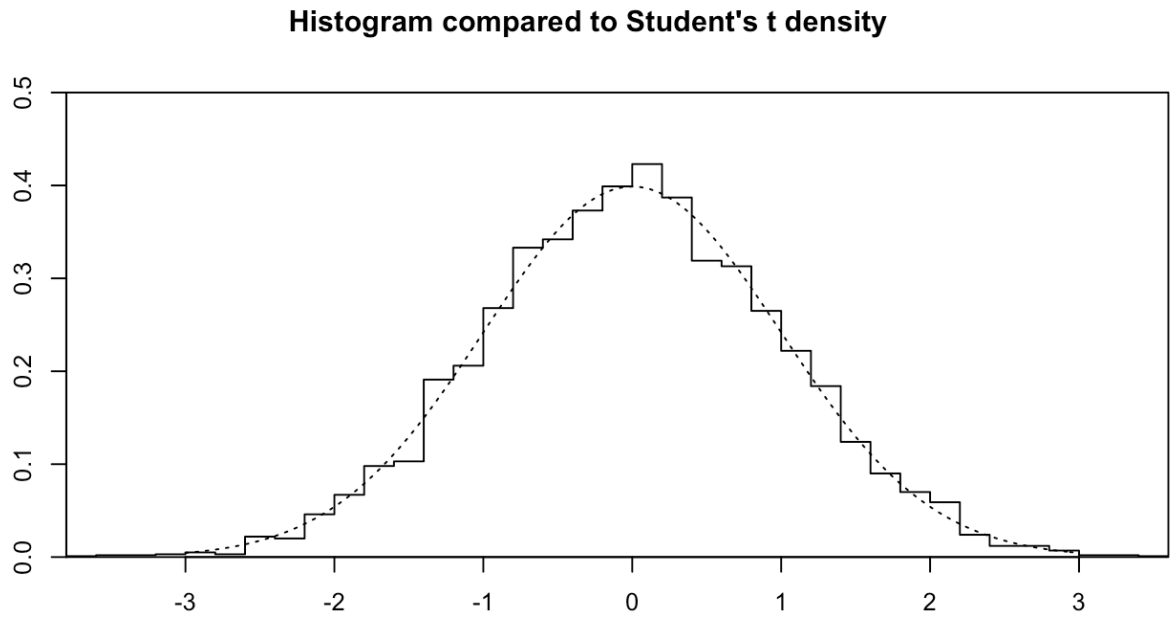
Create a custom histogram plot that looks nice.

```
par(xaxs="i", yaxs="i", cex=0.8, cex.axis=1.0) -> opar

# theoretical density
f <- function(x) dt(x, df=n-1)

plot(h$breaks, c(h$density, 0), type="s",
     xlab="", ylab="", ylim=c(0, 0.5))
curve(f, -3, 3, lwd=1, lty=3, add=TRUE)

title(main="Histogram compared to Student's t density", outer=T, line=-2)
```




```
par(opar)
```

Multiple correlation coefficient

Population multiple correlation

We look at the population multiple correlation coefficient between X_1 and $\mathbf{X}_2 = (X_2, X_3, \ldots, X_p)'$. Let's look at a simple example where $p = 4$. Generate a random covariance matrix.

```
X <- matrix(runif(25), ncol=5)
Sigma <-  t(X) %*% X
Sigma
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]
## [1,] 2.120765 1.618218 2.036706 1.745503 1.641680
## [2,] 1.618218 1.660239 1.742019 1.123014 1.249280
## [3,] 2.036706 1.742019 2.656212 1.794636 1.780071
## [4,] 1.745503 1.123014 1.794636 1.946383 1.395896
## [5,] 1.641680 1.249280 1.780071 1.395896 1.673934
```

Partitioning the covariance matrix Σ as

$$\Sigma = \begin{pmatrix} \sigma_{11} & \bar{\sigma}_{21}' \\ \bar{\sigma}_{21} & \Sigma_{22} \end{pmatrix}.$$

The multiple correlation coefficient between X_1 and $\mathbf{X}_2$ is

$$\rho = \sqrt{\frac{\bar{\sigma}_{21}' \Sigma_{22}^{-1} \bar{\sigma}_{21}}{\sigma_{11}}}$$

```
p <- ncol(Sigma)
rho <- sqrt(t(Sigma[2:p,1]) %*% solve(Sigma[2:p,2:p]) %*% Sigma[2:p,1] / Sigma[
1,1])
rho
```

```
##          [,1]
## [1,] 0.963838
```

Sample multiple correlation

We sample N observations from a $\mathcal{N}(\mu, \Sigma)$ distribution.

```
N <- 100
mu <- c(0,0,0,0,0)
X <- rmvnorm(N, mean=mu, sigma=Sigma)
```

The sample MCC is given by

```
S <- cov(X)
p <- ncol(S)
R <- sqrt(t(S[2:p,1]) %*% solve(S[2:p,2:p]) %*% S[2:p,1] / S[1,1])
rho
```

```
##          [,1]
## [1,] 0.963838
```

Sample multiple correlation distribution

We perform a simulation.

```
p <- 5
Sigma <- diag(1, p, p)
mu <- rep(0, p)

M <- 5000 # number of MC simulations
n <- 1000 # number of observations

MC <- numeric(M)
for (i in 1:M) {
  X <- rmvnorm(n+1, mean=mu, sigma=Sigma)
  S <- cov(X)
  R <- sqrt(t(S[2:p,1]) %*% solve(S[2:p,2:p]) %*% S[2:p,1] / S[1,1])
  Rsq <- R**2
  MC[i] <- (n-(p-1))/(p-1) * Rsq/(1-Rsq)
}
```

Generate histogram of MC simulations.

```
hist(MC, breaks="FD", plot=F) -> h
```

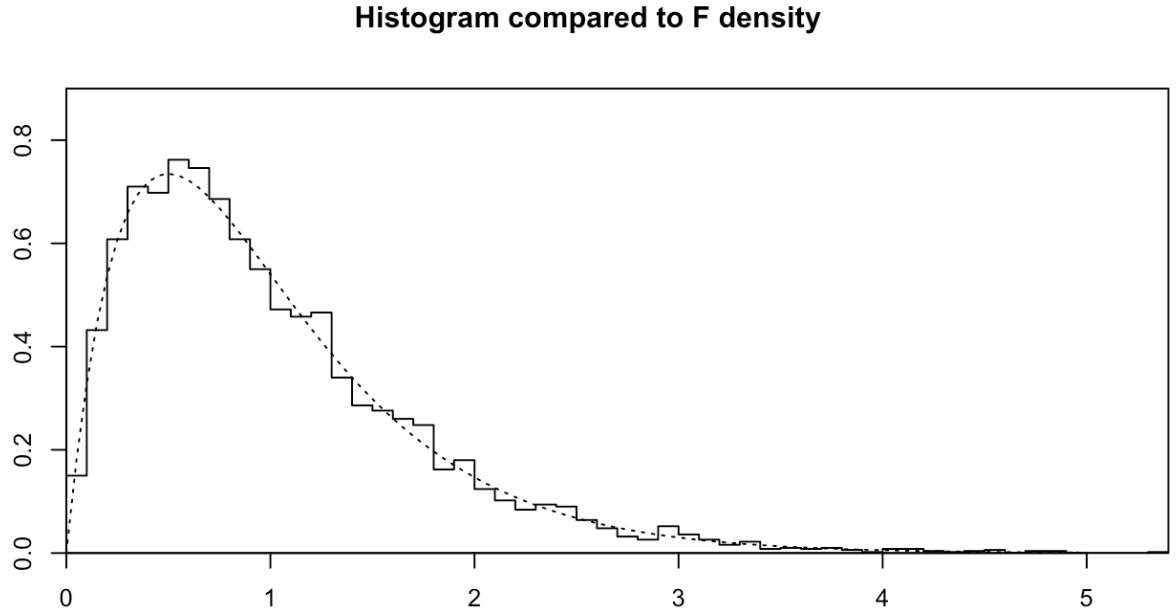
Create a custom histogram plot that looks nice.

```
par(xaxs="i", yaxs="i", cex=0.8, cex.axis=1.0) -> opar

# theoretical density
f <- function(x) df(x, df1=p-1, df2=n-(p-1))

plot(h$breaks, c(h$density, 0), type="s",
     xlab="", ylab="", ylim=c(0, 0.9))
curve(f, 0, 5, lwd=1, lty=3, add=TRUE)

title(main="Histogram compared to F density", outer=T, line=-2)
```




```
par(opar)
```

High-dimensional regime

Over-estimation of R^2

Choose a large p.

```
p <- 500
```

Generate a population covariance matrix.

```
pcor <- function(rho, p) {
  Tn <- matrix(0, p, p)
  for (i in 1:p) {
    for (j in 1:p) {
      Tn[i,j] <- rho^abs(i-j)
    }
  }
  return(Tn)
}

Sigma <- pcor(0.6, p)
```

Calculate the population MCC.

```
p <- ncol(Sigma)
rho <- sqrt(t(Sigma[2:p,1]) %*% solve(Sigma[2:p,2:p]) %*% Sigma[2:p,1] / Sigma[
1,1])
rho
```

```
##      [,1]
## [1,]  0.6
```

```
mu <- rep(0, p)

M <- 500 # number of MC simulations
n <- p # number of observations
y <- p/n

MC <- numeric(M)
for (i in 1:M) {
  X <- rmvnorm(n+1, mean=mu, sigma=Sigma)
  S <- cov(X)
  R <- sqrt(t(S[2:p,1]) %*% solve(S[2:p,2:p]) %*% S[2:p,1] / S[1,1])
  Rsq <- R**2
  MC[i] <- Rsq - (1-y)*rho**2-y
}
```

We can see that the quantity $R^2 - (1 - y)\rho^2 - y$ is closely distributed around zero, or in other words, R^2 does not converge to ρ^2 as desired!

```
mean(MC)
```

```
## [1] -0.001279824
```

```
sd(MC)
```

```
## [1] 0.001678863
```

Central limit theorem

Choose a large p.

```
p <- 250
```

Generate a population covariance matrix.

```
pcor <- function(rho, p) {
  Tn <- matrix(0, p, p)
  for (i in 1:p) {
    for (j in 1:p) {
      Tn[i,j] <- rho^abs(i-j)
    }
  }
  return(Tn)
}

Sigma <- pcor(0.6, p)
```

Calculate the population MCC.

```
p <- ncol(Sigma)
rho <- sqrt(t(Sigma[2:p,1]) %*% solve(Sigma[2:p,2:p]) %*% Sigma[2:p,1] / Sigma[
1,1])
rho
```

```
##      [,1]
## [1,]  0.6
```

```
mu <- rep(0, p)

M <- 1000 # number of MC simulations
n <- 500 # number of observations
y <- p/n

MC <- numeric(M)
for (i in 1:M) {
  X <- rmvnorm(n+1, mean=mu, sigma=Sigma)
  S <- cov(X)
  R <- sqrt(t(S[2:p,1]) %*% solve(S[2:p,2:p]) %*% S[2:p,1] / S[1,1])
  MC[i] <- sqrt(n)*(R^2-y-(1-y)*rho^2)
}
```

Generate histogram of MC simulations.

```
hist(MC, breaks="FD", plot=F) -> h
```

```
sigmasq <- function(t) 2*(y+(1-y)*t)^2 - 2*(-2*(1-y)*t^2 + 4*(1-y)*t+2*y)*(y+(1
-y)*t-0.5)
```

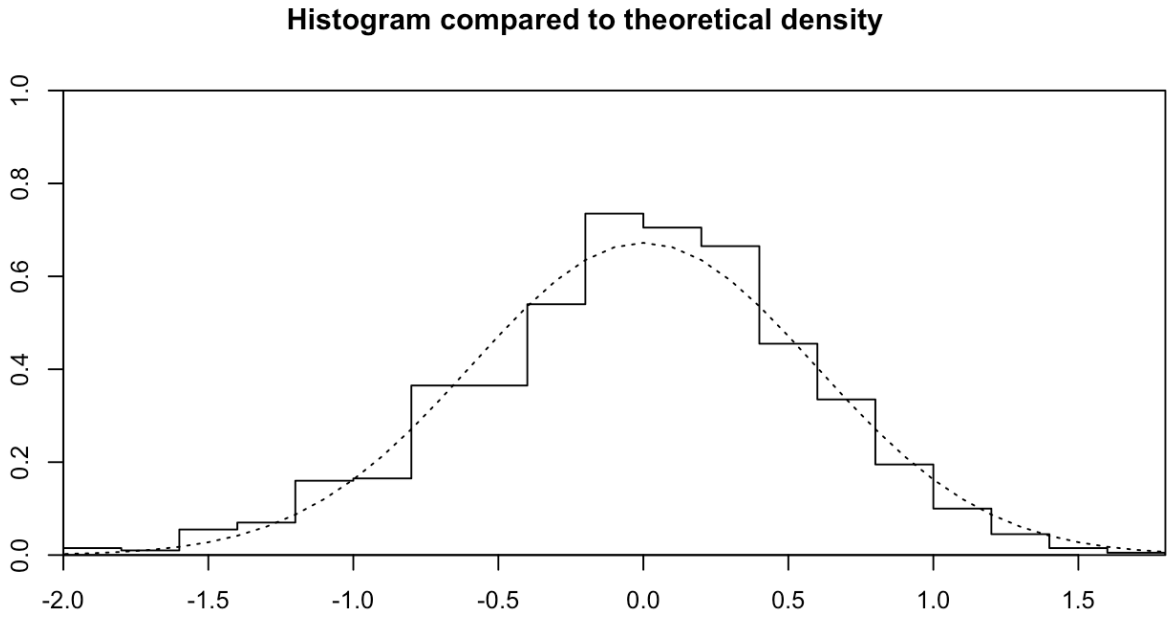
Create a custom histogram plot that looks nice.

```
par(xaxs="i", yaxs="i", cex=0.8, cex.axis=1.0) -> opar

# theoretical density
sd <- sqrt(sigmasq(rho^2))
f <- function(x) dnorm(x, mean=0, sd=sd)

plot(h$breaks, c(h$density, 0), type="s",
     xlab="", ylab="", ylim=c(0, 1.0))
curve(f, -5, 5, lwd=1, lty=3, add=TRUE)

title(main="Histogram compared to theoretical density", outer=T, line=-2)
```


Histogram compared to theoretical density
0.0
0.2
0.4
0.6
0.8
1.0
-2.0
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5

```
par(opar)
```