Workshop 12

Two sample test of mean

Consider two sets of observations, the first set has (population) mean μ_1 and the second has μ_2. We are going to look at a couple of hypothesis tests for determining if the mean of each set is equal (or not). That is, the null hypothesis is given by $\mu_1 = \mu_2$.

Test data

Generate some test data for this workshop. We use the library MASS to sample from multivariate Normal distribution.

```
library(MASS)
set.seed(1234) # set seed for reproducibility
```

Set the size of the two data sets equal.

```
n1 <- n2 <- 50
```

Set the dimensionality of the data.

```
p <- 200
```

Set the true population means of the two samples.

```
mu1 <- rep(0, p)
mu2 <- mu1
mu2[1:10] <- 0.2
```

We use a fancy technique to generate our standard (correlated) covariance matrix.

```
true.cov <- 0.4^(abs(outer(1:p, 1:p, "-"))) # AR1 covariance
```

Generate the two samples. We assume both sets have the same covariance.

```
sam1 <- mvrnorm(n = n1, mu = mu1, Sigma = true.cov)
sam2 <- mvrnorm(n = n2, mu = mu2, Sigma = true.cov)
```

Hotelling T^2

Implement the Hotelling T2 (https://en.wikipedia.org/wiki/Hotelling%27s_T-squared_distribution) approach to test the hypothesis that $\mu_1 = \mu_2$. What is the p-value?

Bai and Saranadasa (1996)

We are now going to look at the result proposed in the paper:

Bai ZD and Saranadasa H (1996). "Effect of high dimension: by an example of a two sample problem." Statistica Sinica, 6(2), 311–329.

Obtain the number of samples in each set.

```
n1 <- dim(sam1)[1]
n2 <- dim(sam2)[1]
```

Calculate the parameters, etc.

```
tau <- (n1 * n2)/(n1 + n2)
n <- n1 + n2 - 2
p <- dim(sam1)[2]
```

Calculate the sum of square difference of means.

```
diff <- colMeans(sam1) - colMeans(sam2)
XX <- sum(diff^2)
```

The sample covariance.

```
sam.cov <- ((n1 - 1) * cov(sam1) + (n2 - 1) * cov(sam2))/n
```

Calculate the proposed test statistic.

```
trS <- sum(diag(sam.cov))
tr.cov2 <- n^2/((n + 2) * (n - 1)) * (sum(sam.cov^2) - trS^2/n)
test.stat <- as.numeric((tau * XX - trS)/sqrt(2 * (n + 1)/n * tr.cov2))
```

By Eq. (4.5) in the their paper, this is $N(0, 1)$ so we use this information to get the p-value.

```
pval <- 1 - pnorm(test.stat)
print(pval)
```

```
## [1] 0.4937037
```

Chen and Qin (2010)

Get the sample sizes.

```
n1 <- dim(sam1)[1]
n2 <- dim(sam2)[1]
```

Dimensionality and degrees of freedom.

```
p <- dim(sam1)[2]
n <- n1 + n2 - 2
```

Calculate the test statistic.

```
sam.cov <- ((n1 - 1)*cov(sam1) + (n2 - 1)*cov(sam2))/n
trS <- sum(diag(sam.cov))
tr.cov2 <- n^2/((n + 2)*(n - 1))*(sum(sam.cov^2) - trS^2/n)
T1 <- sam1 %*% t(sam1)
T2 <- sam2 %*% t(sam2)
P1 <- (sum(T1) - sum(diag(T1)))/(n1*(n1 - 1))
P2 <- (sum(T2) - sum(diag(T2)))/(n2*(n2 - 1))
P3 <- -2*sum(sam1 %*% t(sam2))/(n1*n2)
T <- P1 + P2 + P3
test.stat <- as.numeric(T/sqrt((2/(n1*(n1 - 1)) + 2/(n2*(n2 - 1)) + 4/(n1*n2))*
tr.cov2))
```

From Theorem 1, the asymptotic test statistic is $N(0, 1)$. We use this to get the p-value.

```
pval <- 1 - pnorm(test.stat)
print(pval)
```

```
## [1] 0.4937037
```

Chen and Qin (2010) also proposes a test when the two sets of observations have different population covariances, see p.814. This could be implemented as follows:

```r
n1 <- dim(sam1)[1]
n2 <- dim(sam2)[1]
p <- dim(sam1)[2]
T1 <- sam1 %*% t(sam1)
T2 <- sam2 %*% t(sam2)
P1 <- (sum(T1) - sum(diag(T1)))/(n1*(n1 - 1))
P2 <- (sum(T2) - sum(diag(T2)))/(n2*(n2 - 1))
P3 <- -2*sum(sam1 %*% t(sam2))/(n1*n2)
T <- P1 + P2 + P3

tr.cov1.sq <- tr.cov2.sq <- tr.cov1.cov2 <- 0
for(j in 1:n1){
    for(k in 1:n1){
        if(j != k){
            tempmean <- (colSums(sam1) - sam1[j,] - sam1[k,])/(n1 - 2)
            P1 <- sum(sam1[j,]*(sam1[k,] - tempmean))
            P2 <- sum(sam1[k,]*(sam1[j,] - tempmean))
            tr.cov1.sq <- tr.cov1.sq + P1*P2
        }
    }
}

tr.cov1.sq <- tr.cov1.sq/(n1*(n1 - 1))
for(j in 1:n2){
    for(k in 1:n2){
        if(j != k){
            tempmean <- (colSums(sam2) - sam2[j,] - sam2[k,])/(n2 - 2)
            P1 <- sum(sam2[j,]*(sam2[k,] - tempmean))
            P2 <- sum(sam2[k,]*(sam2[j,] - tempmean))
            tr.cov2.sq <- tr.cov2.sq + P1*P2
        }
    }
}

tr.cov2.sq <- tr.cov2.sq/(n2*(n2 - 1))
for(j in 1:n1){
    for(k in 1:n2){
        tempmean1 <- (colSums(sam1) - sam1[j,])/(n1 - 1)
        tempmean2 <- (colSums(sam2) - sam2[k,])/(n2 - 1)
        P1 <- sum(sam1[j,]*(sam2[k,] - tempmean2))
        P2 <- sum(sam2[k,]*(sam1[j,] - tempmean1))
        tr.cov1.cov2 <- tr.cov1.cov2 + P1*P2
    }
}
tr.cov1.cov2 <- tr.cov1.cov2/(n1*n2)
```

```
test.stat <- T/sqrt(2/(n1*(n1 - 1))*tr.cov1.sq + 2/(n2*(n2 - 1))*tr.cov2.sq + 4
/(n1*n2)*tr.cov1.cov2)
test.stat <- as.numeric(test.stat)
pval <- 1 - pnorm(test.stat)
print(pval)
```

```
## [1] 0.4937049
```

Generate two sets of observations with different covariances and test the above.