# DSA 8070 R Session 4: Inference and Comparison of Mean Vectors

Whitney Huang, Clemson University

## Contents

### CIs: Mineral Content Measurements

```
xbar <- c(0.8438, 1.7927)
s <- c(0.1140, 0.2835)
n = 64; p = 2; alpha = 0.05
```

```
# One at a Time
## mu1
(CI1_1 <- xbar[1] + c(-1, 1) * qt(1 - alpha / 2, n - 1) * (s[1] / sqrt(n)))
```

```
## [1] 0.8153236 0.8722764
```

```
## mu2
(CI2_1 <- xbar[2] + c(-1, 1) * qt(1 - alpha / 2, n - 1) * (s[2] / sqrt(n)))
```

```
## [1] 1.721884 1.863516
## Bonferroni Method
## mu1
(CI1_2 <- xbar[1] + c(-1, 1) * qt(1 - alpha / (2 * p), n - 1) * (s[1] / sqrt(n)))
```

```
## [1] 0.8110786 0.8765214
```

```
## mu2
(CI2_2 <- xbar[2] + c(-1, 1) * qt(1 - alpha / (2 * p), n - 1) * (s[2] / sqrt(n)))
```

```
## [1] 1.711327 1.874073
```

```
# Simultaneous CIs
## mu1
multiplier <- sqrt((p * (n - 1) / (n - p)) * qf(1 - alpha, p, n - p))
(CI1_3 <- xbar[1] + c(-1, 1) * multiplier * (s[1] / sqrt(n)))
```

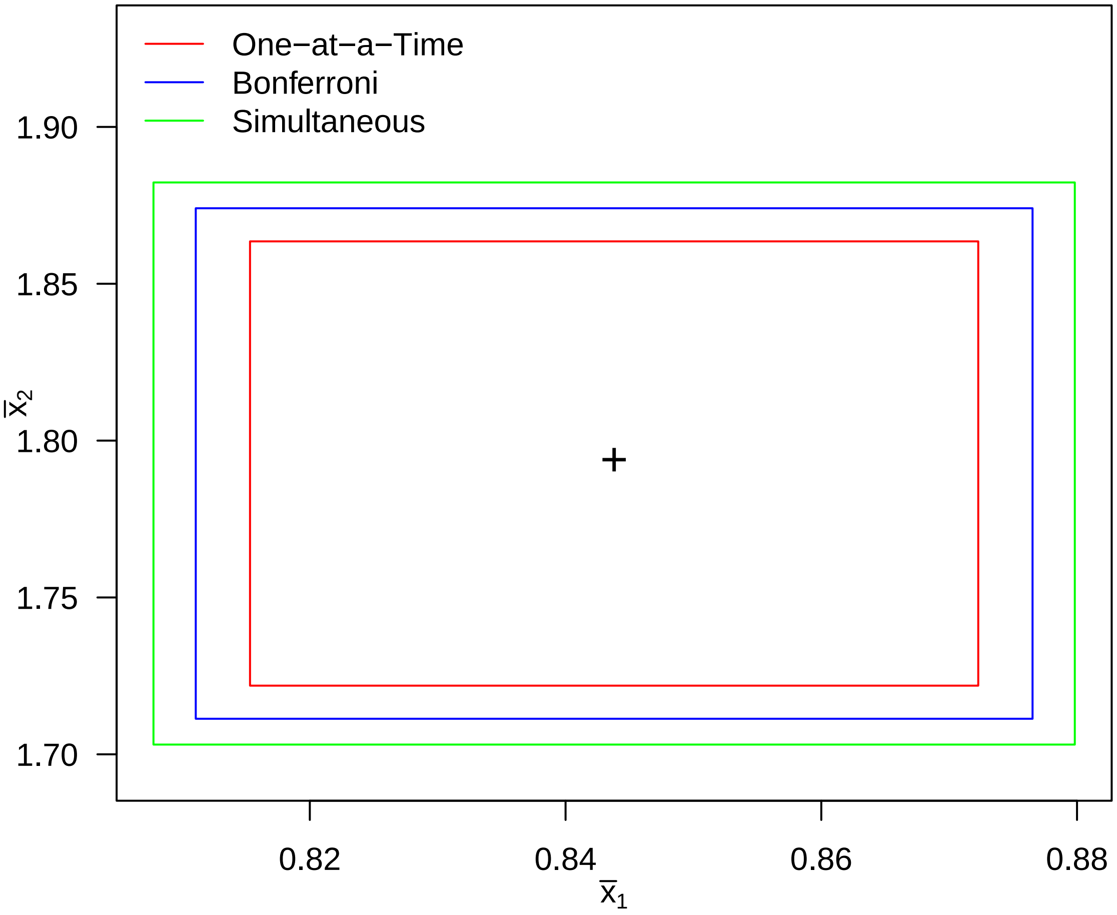
```
## [1] 0.8077726 0.8798274
```

```
## mu2
(CI2_3 <- xbar[2] + c(-1, 1) * multiplier * (s[2] / sqrt(n)))
```

```
## [1] 1.703106 1.882294
```

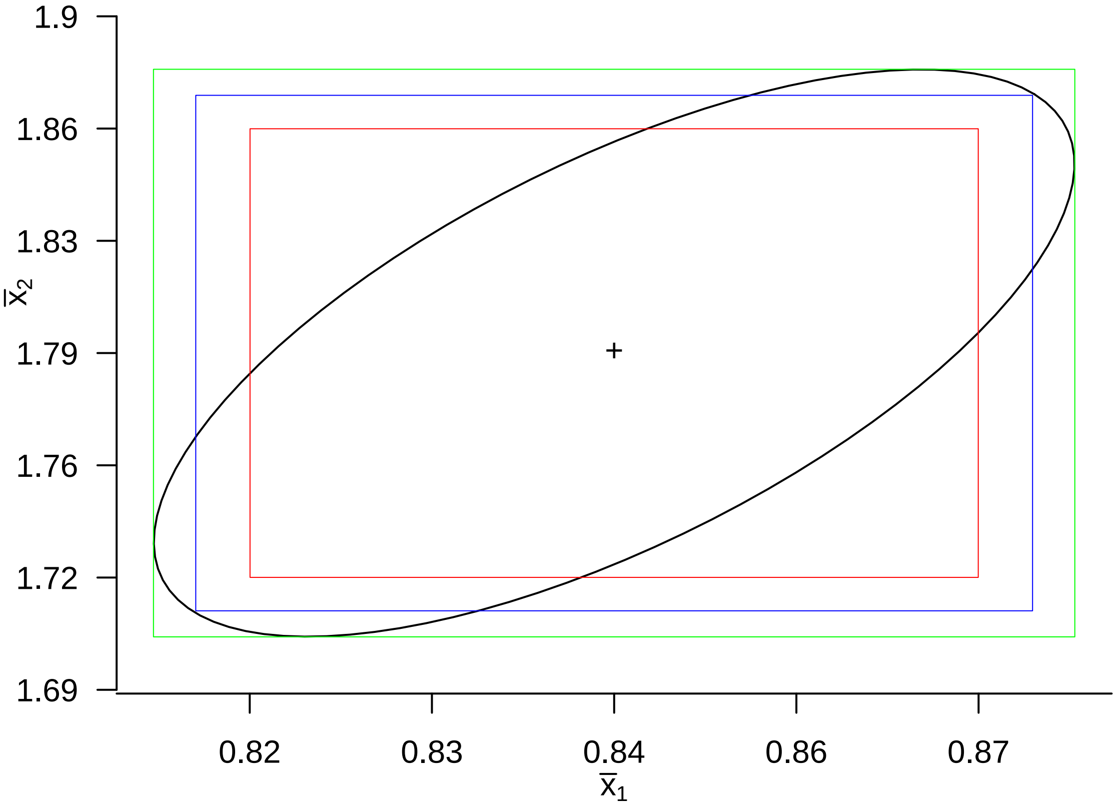
Let's plot the CIs

```
par(las = 1, mgp = c(2, 1, 0), mar = c(3.5, 3.5, 0.8, 0.6))
plot(xbar[1], xbar[2], pch = "+", cex = 1.5,
     xlim = range(CI1_3),
     ylim = range(CI2_3) * c(0.995, 1.025),
     xlab = expression(bar(x)[1]),
     ylab = expression(bar(x)[2]))
rect(CI1_1[1], CI2_1[1], CI1_1[2], CI2_1[2], border = "red")
rect(CI1_2[1], CI2_2[1], CI1_2[2], CI2_2[2], border = "blue")
rect(CI1_3[1], CI2_3[1], CI1_3[2], CI2_3[2], border = "green")
legend("topleft", legend = c("One-at-a-Time", "Bonferroni", "Simultaneous"),
       col = c("red", "blue", "green"), lty = 1, bty = "n")
```



### Confidence Ellipsoid

```
r_corr <- sqrt(((n - 1) * p / (n - p)) * qf(0.95, p, n) / qchisq(0.95, p))
par(las = 1, mgp = c(2, 1, 0), mar = c(3.5, 3.5, 0.6, 0.6))
library(ellipse)
rho = 2 / 3
plot(ellipse(rho, scale = r_corr * s / sqrt(n), centre = xbar), type = 'l',
las = 1, bty = "n", xaxt = "n", yaxt = "n",
xlim = range(CI1_3),
ylim = range(CI2_3) * c(0.995, 1.025), xlab = expression(bar(x)[1]),
ylab = expression(bar(x)[2]))
points(xbar[1], xbar[2], pch = "+")
xg <- seq(xbar[1] - 3 * (s[1] / sqrt(n)), xbar[1] + 3 * (s[1] / sqrt(n)), s[1] / sqrt(n))
yg <- seq(xbar[2] - 3 * (s[2] / sqrt(n)), xbar[2] + 3 * (s[2] / sqrt(n)), s[2] / sqrt(n))
axis(1, at = xg, labels = round(xg, 2))
axis(2, at = yg, labels = round(yg, 2))
rect(CI1_1[1], CI2_1[1], CI1_1[2], CI2_1[2], border = "red", lwd = 0.5)
rect(CI1_2[1], CI2_2[1], CI1_2[2], CI2_2[2], border = "blue", lwd = 0.5)
rect(CI1_3[1], CI2_3[1], CI1_3[2], CI2_3[2], border = "green", lwd = 0.5)
```



### Example: Women's Survey Data

```
dat <- read.table("nutrient.txt")
dat <- dat[, -1]
vars <- c("Calcium", "Iron", "Protein", "Vitamin A", "Vitamin C")
names(dat) <- vars
(xbar <- apply(dat, 2, mean))
```

```
##   Calcium      Iron   Protein Vitamin A Vitamin C
## 624.04925  11.12990  65.80344 839.63535  78.92845
```

```
(colMeans(dat))
```

```
##   Calcium      Iron   Protein Vitamin A Vitamin C
## 624.04925  11.12990  65.80344 839.63535  78.92845
```

```
(S <- cov(dat))
```

```
##                Calcium       Iron   Protein   Vitamin A  Vitamin C
## Calcium   157829.4439  940.08944 6075.8163  102411.127  6701.6160
## Iron         940.0894   35.81054  114.0580    2383.153   137.6720
## Protein     6075.8163  114.05803  934.8769    7330.052   477.1998
## Vitamin A 102411.1266 2383.15341 7330.0515 2668452.371 22063.2486
## Vitamin C   6701.6160  137.67199  477.1998   22063.249  5416.2641
```

```r
n <- dim(dat)[1]; p <- dim(dat)[2]

mu0 <- c(1000, 15, 60, 800, 75)

T.squared <- as.numeric(n * t(xbar - mu0) %*% solve(S) %*% (xbar - mu0))
# test statistic
Fobs <- T.squared * ((n - p) / ((n - 1) * p))
# p-value
pf(Fobs, p, n - p, lower.tail = F)
```

```
## [1] 2.988651e-191
```

## Profile Plots

```r
dat_normalized <- array(dim = dim(dat))
for (i in 1:p){
  dat_normalized[, i] <- dat[, i] / mu0[i]
}

(xbar <- apply(dat_normalized, 2, mean))
```

```
## [1] 0.6240493 0.7419933 1.0967240 1.0495442 1.0523793
```

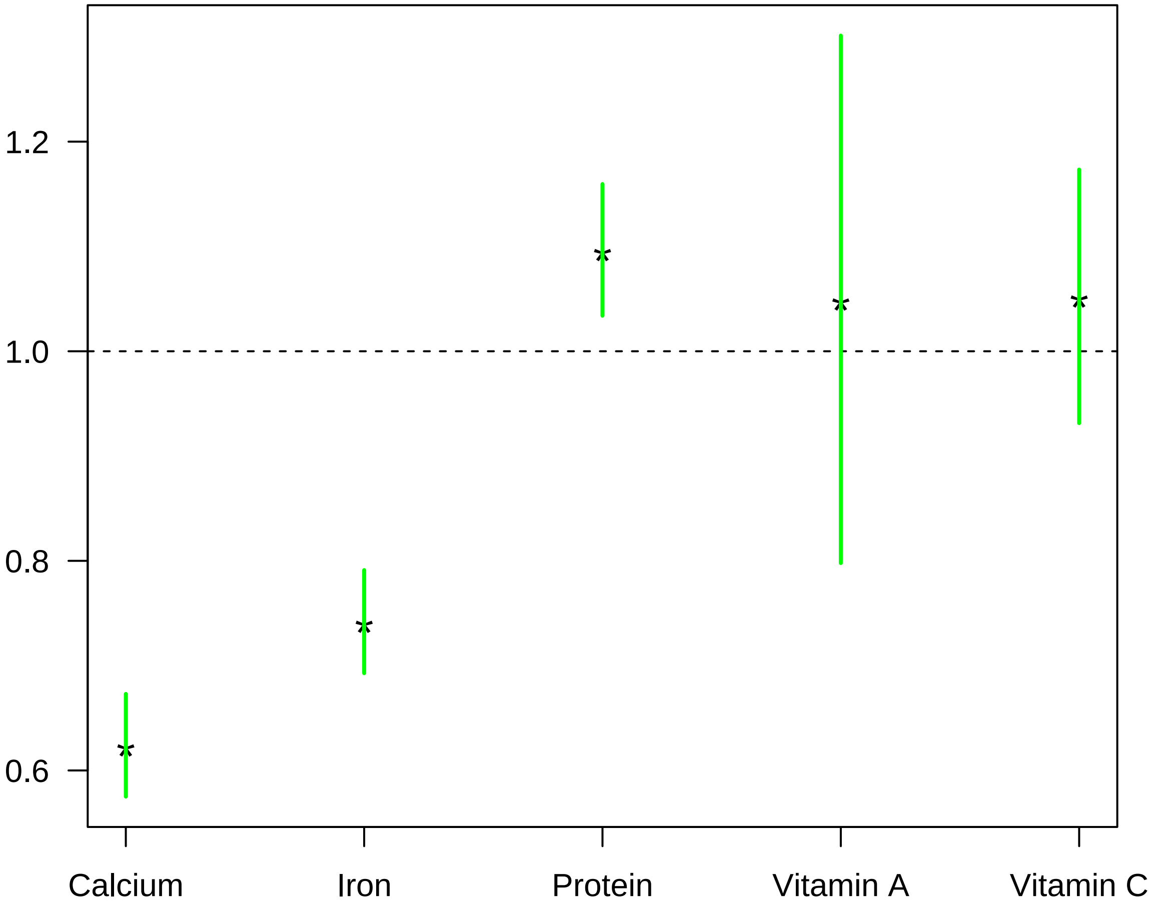
```r
(xbar <- colMeans(dat_normalized))
```

```
## [1] 0.6240493 0.7419933 1.0967240 1.0495442 1.0523793
```

```r
(sd <- apply(dat_normalized, 2, sd))
```

```
## [1] 0.3972775 0.3989460 0.5095959 2.0419248 0.9812703
```

```r
# Simultaneous CIs
CIs <- array(dim = c(p, 2))
multiplier <- sqrt((p * (n - 1) / (n - p)) * qf(1 - alpha, p, n - p))
for (j in 1:p){
  CIs[j,] <- xbar[j] + c(-1, 1) * multiplier * (sd[j] / sqrt(n))
}
#  Profile Plot
par(las = 1, mgp = c(2, 1, 0), mar = c(3, 2.4, 0.6, 0.8))
plot(1:p, xbar, ylim = range(CIs), xaxt = "n", pch = "*",
     xlab = "", ylab = "", cex = 1.5)
abline(h = 1, lty = 2)
for (j in 1:p) segments(x0 = j, y0 = CIs[j, 1], y1 = CIs[j, 2], col = "green", lwd = 2)
axis(1, at = 1:p, labels = vars)
```



### Spouse Survey Data Example

```
dat <- read.table("spouse.txt")
d <- array(dim = c(dim(dat)[1], dim(dat)[2] / 2))
# Calculate the differences
for (i in 1:(dim(dat)[2] / 2)){
  d[, i] <- dat[, i] - dat[, i + dim(dat)[2] / 2]
}

(xbar <- apply(d, 2, mean))
```

```
## [1]  0.06666667 -0.13333333 -0.30000000 -0.13333333
```

```
(S <- cov(d))
```

```
##             [,1]        [,2]       [,3]        [,4]
## [1,]  0.82298851  0.07816092 -0.0137931 -0.05977011
## [2,]  0.07816092  0.80919540 -0.2137931 -0.15632184
## [3,] -0.01379310 -0.21379310  0.5620690  0.51034483
## [4,] -0.05977011 -0.15632184  0.5103448  0.60229885
```

```
n <- dim(d)[1]; p <- dim(d)[2]

mu0 <- rep(0, 4)

T.squared <- as.numeric(n * t(xbar - mu0) %*% solve(S) %*% (xbar - mu0))
# test statistic
Fobs <- T.squared * ((n - p) / ((n - 1) * p))
##p-value
pf(Fobs, p, n - p, lower.tail = F)
```

```
## [1] 0.03936914
```

## Swiss Bank Notes Example

Suppose there are two distinct populations for 1000 franc Swiss Bank Notes:

- The first population is the population of Genuine Bank Notes.
- The second population is the population of Counterfeit Bank Notes.

For both populations, the following measurements were taken:

1. Length of the note
2. Width of the Left-Hand side of the note
3. Width of the Right-Hand side of the note
4. Width of the Bottom Margin
5. Width of the Top Margin
6. Diagonal Length of Printed Area

We want to determine if counterfeit notes can be distinguished from the genuine Swiss bank notes.

### Read the data

```
library(mclust)
data(banknote)
head(banknote)
```

```
##    Status Length  Left Right Bottom  Top Diagonal
## 1 genuine  214.8 131.0 131.1    9.0  9.7    141.0
## 2 genuine  214.6 129.7 129.7    8.1  9.5    141.7
## 3 genuine  214.8 129.7 129.7    8.7  9.6    142.2
## 4 genuine  214.8 129.7 129.6    7.5 10.4    142.0
## 5 genuine  215.0 129.6 129.7   10.4  7.7    141.8
## 6 genuine  215.7 130.8 130.5    9.0 10.1    141.4
```

### Calculate summary statistics

Mean vectors: $\bar{\boldsymbol{X}}_1 = \frac{1}{n_1}\sum_{i=1}^{n_1}\boldsymbol{X}_{1,i}$, $\bar{\boldsymbol{X}}_2 = \frac{1}{n_2}\sum_{j=1}^{n_2}\boldsymbol{X}_{2,i}$

Covariance Matrices: $\boldsymbol{S}_i = \frac{1}{n_i-1}\sum_{j=1}^{n_i}(\boldsymbol{x}_{ij}-\bar{\boldsymbol{x}}_i)(\boldsymbol{x}_{ij}-\bar{\boldsymbol{x}}_i)^T, \quad i = 1, 2$

Under the common covariance assumption we can compute the pooled covariance matrix

$$\boldsymbol{S}_p = \frac{(n_1-1)\boldsymbol{S}_1 + (n_2-1)\boldsymbol{S}_2}{n_1+n_2-2}$$

```
dat <- banknote
real <- which(dat$Status == "genuine")
fake <- which(dat$Status == "counterfeit")
(xbar1 <- colMeans(dat[real, -1]))
```

```
##   Length     Left    Right   Bottom      Top Diagonal
##  214.969  129.943  129.720    8.305   10.168  141.517
```

```
(xbar2 <- colMeans(dat[fake, -1]))
```

```
##   Length     Left    Right   Bottom      Top Diagonal
##  214.823  130.300  130.193   10.530   11.133  139.450
```

```
(Sigma1 <- round(cov(dat[real, -1]), 3))
```

```
##          Length   Left  Right Bottom    Top Diagonal
## Length    0.150  0.058  0.057  0.057  0.014    0.005
## Left      0.058  0.133  0.086  0.057  0.049   -0.043
## Right     0.057  0.086  0.126  0.058  0.031   -0.024
## Bottom    0.057  0.057  0.058  0.413 -0.263    0.000
## Top       0.014  0.049  0.031 -0.263  0.421   -0.075
## Diagonal  0.005 -0.043 -0.024  0.000 -0.075    0.200
```

```
(Sigma2 <- round(cov(dat[fake, -1]), 3))
```

```
##          Length   Left  Right Bottom    Top Diagonal
## Length    0.124  0.032  0.024 -0.101  0.019    0.012
## Left      0.032  0.065  0.047 -0.024 -0.012   -0.005
## Right     0.024  0.047  0.089 -0.019  0.000    0.034
## Bottom   -0.101 -0.024 -0.019  1.281 -0.490    0.238
## Top       0.019 -0.012  0.000 -0.490  0.404   -0.022
## Diagonal  0.012 -0.005  0.034  0.238 -0.022    0.311
```

```
n1 <- length(real); n2 <- length(fake); p <- dim(dat[, -1])[2]
Sp <- ((n1 - 1) * Sigma1 + (n2 - 1) * Sigma2) / (n1 + n2 - 2)
```

### Perform a two-sample Hotelling's T-Square test

$$T^2 = (\bar{\boldsymbol{x}}_1 - \bar{\boldsymbol{x}}_2)^T \left[\boldsymbol{S}_p \left(\frac{1}{n_1} + \frac{1}{n_2}\right)\right]^{-1} (\bar{\boldsymbol{x}}_1 - \bar{\boldsymbol{x}}_2)$$

Under $H_0$, we have

$$F = \frac{n_1 + n_2 - p - 1}{p(n_1 + n_2 - 2)} T^2 \sim F_{p, n_1 + n_2 - p - 1}$$

We can use this result to calculate the p-value to conduct a two-sample Hotelling's T-Square test

```
# Test statistic
T.squared <- as.numeric(t(xbar1 - xbar2) %*% solve(Sp * (1 / n1 + 1 / n2)) %*% (xbar1 - xbar2))
Fobs <- T.squared * ((n1 + n2 - p - 1) / ((n1 + n2 - 2) * p))
# p-value
pf(Fobs, p, n1 + n2 - p - 1, lower.tail = F)
```

```
## [1] 3.332366e-105
```

⇒ We can distinguish counterfeit notes from genuine notes based on at least one of the measurements

### Simultaneous Confidence Intervals

$$\bar{x}_{1k} - \bar{x}_{2k} \pm \sqrt{\frac{p(n_1 + n_2 - 2)}{n_1 + n_2 - p - 1} F_{p, n_1 + n_2 - p - 1, \alpha}} \sqrt{\left(\frac{1}{n_1} + \frac{1}{n_2}\right) s_{k,p}^2},$$

where $s_{k,p}^2$ is the pooled variance for the variable $k$

```
s1 <- diag(Sigma1); s2 <- diag(Sigma2)

xbar_diff <- xbar1 - xbar2
sp_diff <- ((n1 - 1) * s1 + (n2 - 1) * s2) / (n1 + n2 - 2)

multipler <- sqrt((p * (n1 + n2 - 2) / (n1 + n2 - p - 1)) * qf(0.95, p, n1 + n2 - p - 1))

sp <- sqrt((1 / n1 + 1 / n2) * sp_diff)

CIs <- cbind(xbar_diff + -1 * multipler * sp, xbar_diff + 1 * multipler * sp)
CIs
```

```
##                  [,1]       [,2]
## Length   -0.04423903  0.3362390
## Left     -0.51871747 -0.1952825
## Right    -0.64151694 -0.3044831
## Bottom   -2.69802167 -1.7519783
## Top      -1.29510440 -0.6348956
## Diagonal  1.80720261  2.3267974
```

## MANOVA: Romano-British Pottery Example

Pottery shards were collected from four sites in the British Isles:

1. Llanedyrn
2. Caldicot
3. Isle Thorns
4. Ashley Rails

The concentrations of five different chemicals were measured:

- Aluminum ($Al$)
- Iron ($Fe$)
- Magnesium ($Mg$)
- Calcium ($Ca$)
- Sodium ($Na$)

Objective: To determine whether the chemical content of the pottery depends on the site where the pottery was obtained.

```
dat <- read.table("pottery.txt", header = F)
head(dat)
```

```
##   V1   V2   V3   V4   V5   V6
## 1  L 14.4 7.00 4.30 0.15 0.51
## 2  L 13.8 7.08 3.43 0.12 0.17
## 3  L 14.6 7.09 3.88 0.13 0.20
## 4  L 11.5 6.37 5.64 0.16 0.14
## 5  L 13.8 7.06 5.34 0.20 0.20
## 6  L 10.9 6.26 3.47 0.17 0.22
```

### MANOVA Calculations and Different Tests

$$
\begin{aligned}
\boldsymbol{T} &= \sum_{i=1}^{g}\sum_{j=1}^{n_i}(\boldsymbol{Y}_{ij} - \boldsymbol{y}_{..})(\boldsymbol{Y}_{ij} - \bar{\boldsymbol{y}})^T \\
&= \sum_{i=1}^{g}\sum_{j=1}^{n_i}\left[(\boldsymbol{Y}_{ij} - \bar{\boldsymbol{y}}_{i.}) + (\bar{\boldsymbol{y}}_{i.} - \bar{\boldsymbol{y}}_{..})\right]\left[(\boldsymbol{Y}_{ij} - \bar{\boldsymbol{y}}_{i.}) + (\bar{\boldsymbol{y}}_{i.} - \bar{\boldsymbol{y}}_{..})\right]^T \\
&= \underbrace{\sum_{i=1}^{g}\sum_{j=1}^{n_i}(\boldsymbol{Y}_{ij} - \bar{\boldsymbol{y}}_{i.})(\boldsymbol{Y}_{ij} - \bar{\boldsymbol{y}}_{i.})^T}_{\boldsymbol{E}} + \underbrace{\sum_{i=1}^{g} n_i(\bar{\boldsymbol{y}}_{i.} - \bar{\boldsymbol{y}}_{..})(\bar{\boldsymbol{y}}_{i.} - \bar{\boldsymbol{y}}_{..})^T}_{\boldsymbol{H}}
\end{aligned}
$$

- **Wilks Lambda**

  $$\Lambda^* = \frac{|\boldsymbol{E}|}{|\boldsymbol{H} + \boldsymbol{E}|}$$

  Reject $H_0$ if $\Lambda^*$ is “small”

- **Hotelling-Lawley Trace**

  $$T_0^2 = trace(\boldsymbol{H}\boldsymbol{E}^{-1})$$

  Reject $H_0$ if $T_0^2$ is "large"

- **Pillai Trace**

  $$V = trace(\boldsymbol{H}(\boldsymbol{H}+\boldsymbol{E})^{-1})$$

  Reject $H_0$ if $V$ is "large"

```
out <- manova(cbind(V2, V3, V4, V5, V6) ~ V1, data = dat)
summary(out, test = "Wilks")
```

```
##           Df    Wilks approx F num Df den Df   Pr(>F)    
## V1         3 0.012301   13.088     15 50.091 1.84e-12 ***
## Residuals 22                                             
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(out)
```

```
##           Df Pillai approx F num Df den Df    Pr(>F)    
## V1         3 1.5539   4.2984     15     60 2.413e-05 ***
## Residuals 22                                            
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```