Inferences About a Mean Vector

Blake Pappas

December 17, 2023

CIs: Mineral Content Measurements

```
xbar <- c(0.8438, 1.7927) # Sample means
s <- c(0.1140, 0.2835) # Sample standard deviations
n = 64 # Sample sizes
p = 2
alpha = 0.05 # Confidence levels


# 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
```

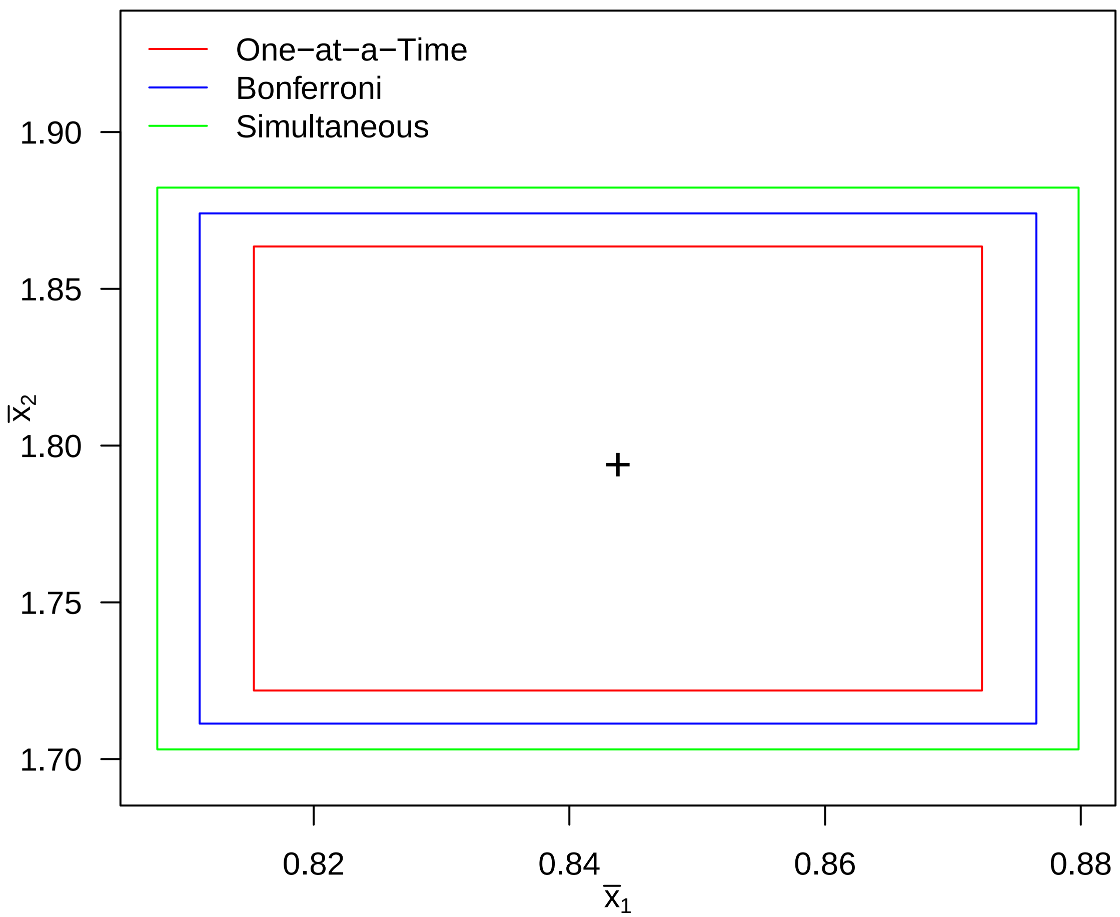
```
# Simultaneous CIs

## 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
```

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

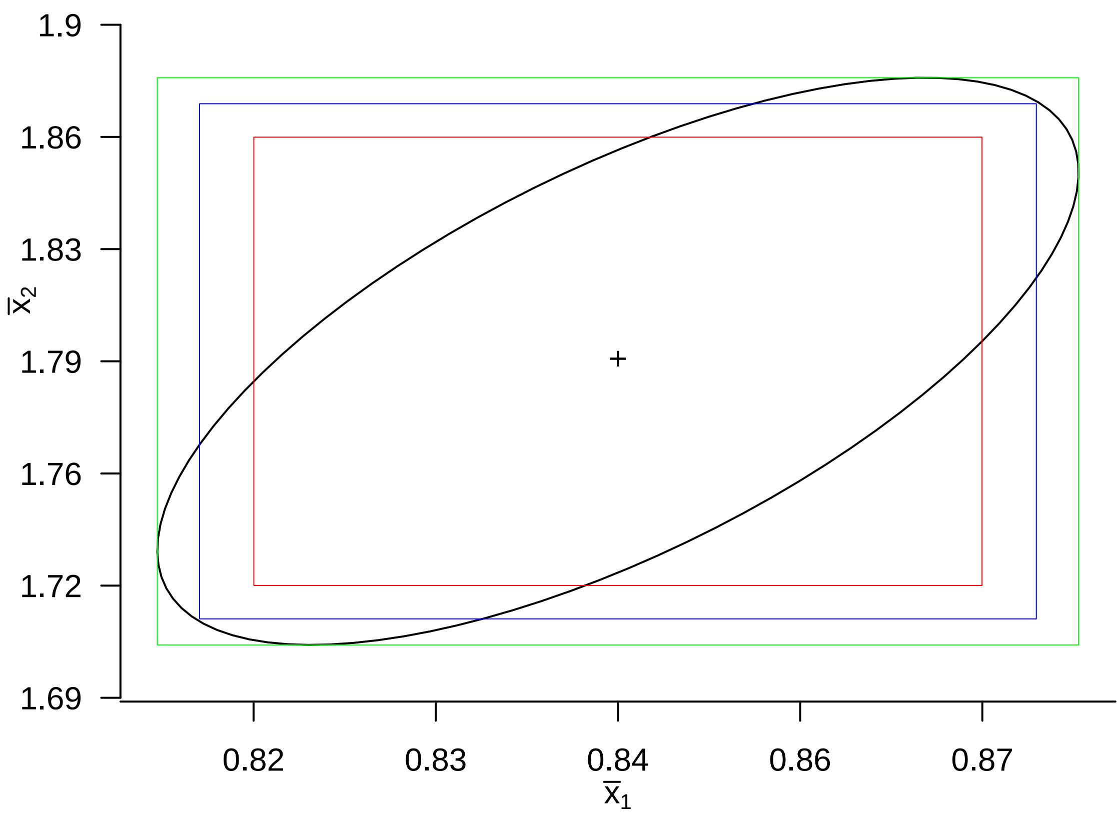
```
## [1] 1.703106 1.882294
```

```r
# 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
r_corr <- sqrt(((n - 1) * p / (n - p)) * qf(0.95, p, n - p) / qchisq(0.95, p))
rho = 2 / 3
par(las = 1, mgp = c(2, 1, 0), mar = c(3.5, 3.5, 0.6, 0.6))
library(ellipse)
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

```r
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
```

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

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

```r
(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

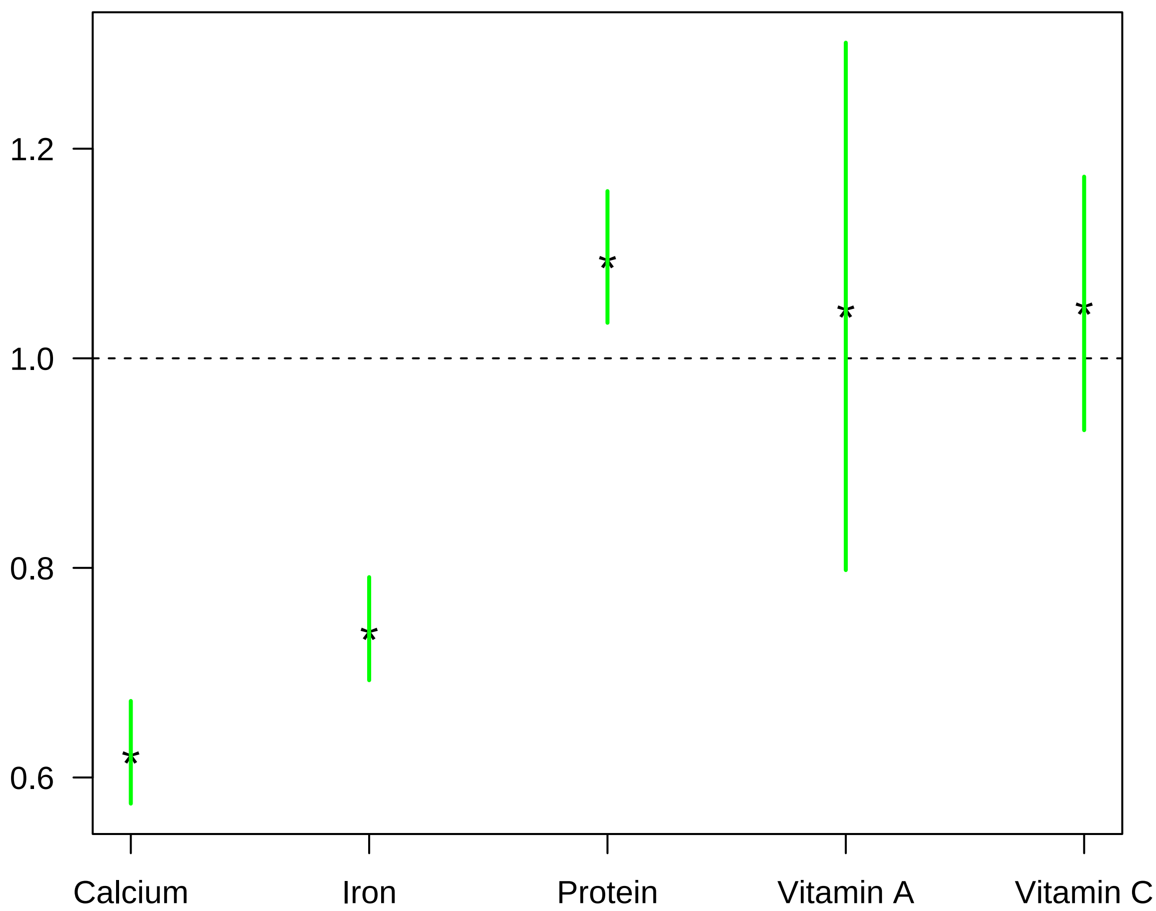
```r
dat_normalized <- array(dim = dim(dat))
for (i in 1:p) {
  dat_normalized[, i] <- dat[, i] / mu0[i]
}

(xbar <- 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

```r
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
```

```r
(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
```

```r
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
```