QUIZ 1 - QUESTION 3

Amanjit Gill

October 31, 2022

```
library("ellipse")
```

```
##
## Attaching package: 'ellipse'
## The following object is masked from 'package:graphics':
##
##     pairs
```

```
par(mar = c(4, 4, 0.5, 0.5))
dev.new(width = 2.5, height = 2.5, unit = "in")

n <- 12
p = 2
xbar <- cbind(c(-1.2, -3.9))
S <- rbind(c(6,1), c(1,6))
Sinv <- solve(S)

# PART A - confidence ellipsoid at level 90%

alpha <- 0.1
S.eig <- eigen(S)
scale <- p*(n-1)*qf(1-alpha, p, n-p)/(n*(n-p))

# major axis endpoints
(major_pt1 <- xbar - S.eig$vectors[,1] * sqrt(S.eig$values[1]*scale))
```

```
##           [,1]
## [1,] -2.569866
## [2,] -5.269866
```

```
(major_pt2 <- xbar + S.eig$vectors[,1] * sqrt(S.eig$values[1]*scale))
```

```
##            [,1]
## [1,]  0.1698658
## [2,] -2.5301342
```

```
# minor axis endpoints
(minor_pt1 <- xbar - S.eig$vectors[,2] * sqrt(S.eig$values[2]*scale))
```

```
##             [,1]
## [1,] -0.04225209
## [2,] -5.05774791
```

```
(minor_pt2 <- xbar + S.eig$vectors[,2] * sqrt(S.eig$values[2]*scale))
```

```
##            [,1]
## [1,] -2.357748
## [2,] -2.742252
```

```
# vector of x-coordinates for major axis
major_xvals <- c(major_pt1[1], major_pt2[1])

# vector of y-coordinates for major axis
major_yvals <- c(major_pt1[2], major_pt2[2])

# vector of x-coordinates for minor axis
minor_xvals <- c(minor_pt1[1], minor_pt2[1])

# vector of y-coordinates for minor axis
minor_yvals <- c(minor_pt1[2], minor_pt2[2])

# plot ellipse and major and minor axes
plot(ellipse(cov2cor(S), centre=c(-1.2, -3.9), t=qt(1-alpha/2, n-1), npoints=1000),
    pch=".", cex.lab=.8, cex.axis=.7, xlim=c(-8,6), ylim=c(-10,2),
    xlab=expression(mu*"1"), ylab=expression(mu*"2"))
points(c(major_xvals,minor_xvals), c(major_yvals,minor_yvals),
     pch=4, col="blue", lwd=2)
lines(major_xvals, major_yvals)
lines(minor_xvals, minor_yvals)
points(xbar[1], xbar[2], pch=16, col="blue")

# PART B - confidence ellipsoid for known variance

# everything stays the same except for the scale
scale <- qchisq(1-alpha, p)

# major axis endpoints
(major_pt1 <- xbar - S.eig$vectors[,1] * sqrt(S.eig$values[1]*scale))
```

```
##            [,1]
## [1,] -5.214735
## [2,] -7.914735
```

```
(major_pt2 <- xbar + S.eig$vectors[,1] * sqrt(S.eig$values[1]*scale))
```

```
##            [,1]
## [1,] 2.8147348
## [2,] 0.1147348
```

```
# minor axis endpoints
(minor_pt1 <- xbar - S.eig$vectors[,2] * sqrt(S.eig$values[2]*scale))
```

```
##           [,1]
## [1,]  2.19307
## [2,] -7.29307
```

```
(minor_pt2 <- xbar + S.eig$vectors[,2] * sqrt(S.eig$values[2]*scale))
```

```
##             [,1]
## [1,] -4.5930702
## [2,] -0.5069298
```

```r
# vector of x-coordinates for major axis
major_xvals <- c(major_pt1[1], major_pt2[1])

# vector of y-coordinates for major axis
major_yvals <- c(major_pt1[2], major_pt2[2])

# vector of x-coordinates for minor axis
minor_xvals <- c(minor_pt1[1], minor_pt2[1])

# vector of y-coordinates for minor axis
minor_yvals <- c(minor_pt1[2], minor_pt2[2])

# plot major and minor axes
lines(ellipse(cov2cor(S), centre=c(-1.2, -3.9),
    scale=c(sqrt(qchisq(1-alpha, p)),sqrt(qchisq(1-alpha, p))), npoints=1000),
    pch=".", cex=1, lty="dashed")
points(c(major_xvals,minor_xvals), c(major_yvals,minor_yvals), pch=4, lwd=1.5)
lines(major_xvals, major_yvals, lty="dashed")
lines(minor_xvals, minor_yvals, lty="dashed")
points(xbar[1], xbar[2], pch=16, col="blue")

# PART C - projection to get confidence interval

# change back to Part A scale
scale <- p*(n-1)*qf(1-alpha, p, n-p)/(n*(n-p))

# for variable x1
l_x1 <- c(1,0)
(x1_ci_low <- c(crossprod(l_x1, xbar)) - sqrt(scale * t(l_x1) %*% S %*% l_x1))
```

```
##           [,1]
## [1,] -2.993575
```

```r
(x1_ci_high <- c(crossprod(l_x1, xbar)) + sqrt(scale * t(l_x1) %*% S %*% l_x1))
```

```
##           [,1]
## [1,] 0.5935754
```

```r
# for variable x2
l_x2 <- c(0,1)
(x2_ci_low <- c(crossprod(l_x2, xbar)) - sqrt(scale * t(l_x2) %*% S %*% l_x2))
```

```
##           [,1]
## [1,] -5.693575
```

```r
(x2_ci_high <- c(crossprod(l_x2, xbar)) + sqrt(scale * t(l_x2) %*% S %*% l_x2))
```

```
##           [,1]
## [1,] -2.106425
```

```r
# plot x1 CI as vertical lines
# plot x2 CI as horizontal lines
abline(v=c(x1_ci_low, x1_ci_high), col="blue", lty="dashed")
abline(h=c(x2_ci_low, x2_ci_high), col="blue", lty="dashed")

# PART D - Bonferroni method to get confidence intervals
```

```r
# adapt Part C to do this efficiently

m <- 2*p

scale <- qt(1-alpha/m, n-1)

# for variable x1
l_x1 <- c(1,0)
(x1_ci_low <- c(crossprod(l_x1, xbar)) - scale*sqrt(t(l_x1) %*% S %*% l_x1 / n))
```

```
##           [,1]
## [1,] -2.756332
```

```r
(x1_ci_high <- c(crossprod(l_x1, xbar)) + scale*sqrt(t(l_x1) %*% S %*% l_x1 / n))
```

```
##           [,1]
## [1,] 0.3563315
```

```r
# for variable x2
l_x2 <- c(0,1)
(x2_ci_low <- c(crossprod(l_x2, xbar)) - scale*sqrt(t(l_x2) %*% S %*% l_x2 / n))
```

```
##           [,1]
## [1,] -5.456332
```

```r
(x2_ci_high <- c(crossprod(l_x2, xbar)) + scale*sqrt(t(l_x2) %*% S %*% l_x2 / n))
```

```
##           [,1]
## [1,] -2.343668
```

```r
# plot x1 CI as vertical lines
# plot x2 CI as horizontal lines
abline(v=c(x1_ci_low, x1_ci_high), col="orange", lty="dashed")
abline(h=c(x2_ci_low, x2_ci_high), col="orange", lty="dashed")

# add legend for all 4 parts
legend(
    x="topleft",
    legend=c("Solid ellipse: part A", "Dashed ellipse: Part B",
             "Blue dash: projected CI", "Orange dash: Bonferroni CI"),
    bty='n',
    cex=0.55
)
```