question 2

Sridhar Adhikarla (sriad858)

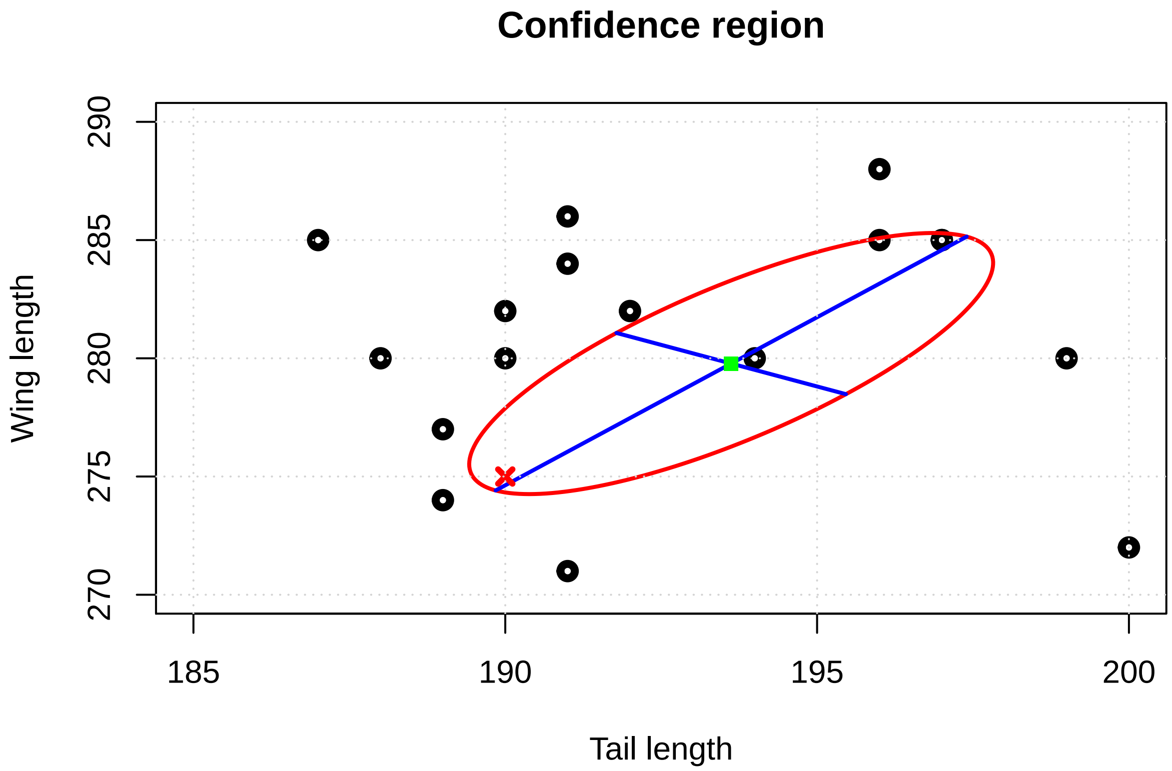
December 1, 2019

QUestion 2

Look at the bird data in file T5-12.dat and solve Exercise 5:20 of Johnson, Wichern. Do not use any extra R package or built{in test but code all required matrix calculations. You MAYNOT use loops!

a

```r
bird_data <- read.table("T5-12.dat")
mu <- c(190, 275) #given mu values
x_bar <- colMeans(bird_data)

calcReqVals = function(data){
  n <- nrow(data)
  p <- ncol(data)
  conf <- 0.05
  S <- cov(data)
  eig <- eigen(S)
  x_bar <- colMeans(data)

  quantile <- qf(1 - conf, df1=p, df2=n - p)
  scale <- sqrt(eig$values * p * (n - 1) * quantile / (n * (n - p)))
  scaled <- eig$vector %*% diag(scale) # scale eigenvectors to length = square-root
  xMat <- rbind(x_bar[1] + scaled[1, ], x_bar[1] - scaled[1, ])
  yMat <- rbind(x_bar[2] + scaled[2, ], x_bar[2] - scaled[2, ])

  angles <- seq(0, 2 * pi, length.out=200)
  ellBase <- cbind(scale[1]*cos(angles), scale[2]*sin(angles)) # making a circle base...
  ellax <- eig$vector %*% t(ellBase)

  return(list("ellax"=ellax, "xMat"=xMat, "yMat"=yMat))
}

out = calcReqVals(bird_data)

#Plotting the confidence region
plot(bird_data, lwd="4", xlab="Tail length", ylab="Wing length",
xlim=c(185, 200), ylim=c(270, 290), main="Confidence region")
lines((out$ellax + x_bar)[1, ], (out$ellax + x_bar)[2, ], asp=1, type="l", lwd=2, col="red")
matlines(out$xMat, out$yMat, lty=1, lwd=2, col="blue") #
points(mu[1], mu[2], pch=4, col="red", lwd=3)
grid()
points(x_bar[1],x_bar[2], type="p", col="green", pch=15)
```



Since the known mean of the male data lies in the confidence interval for the mean of the female data, we are unable reject the hypothesis that the male and the female have the same mean.

b

```
tsq95_intervals <- function(data) {
  conf = 0.05
  n <- nrow(data)
  p <- ncol(data)
  x_bar <- colMeans(data)
  S <- cov(data)
  offset <- sqrt(p * (n - 1) * qf(1 - conf, df1=p, df2=n - p) / (n - p) * diag(S) / n)
  rbind(x_bar - offset, x_bar + offset)
}

bon95_intervals <- function(data) {
  conf = 0.05
  n <- nrow(data)
  p <- ncol(data)
  x_bar <- colMeans(data)
  S <- cov(data)
  offset <- sqrt(diag(S) / n) * qt(1 - conf / (2 * p), df=n - 1)
  rbind(x_bar - offset, x_bar + offset)
}
```

```
cat("\n95% T-square interval\n")
```

```
##
## 95% T-square interval
```

```
tsq95_intervals(bird_data)
```

```
##            V1       V2
## [1,] 189.4217 274.2564
## [2,] 197.8227 285.2992
```

```
cat("\n95% Bonferroni interval\n")
```

```
##
## 95% Bonferroni interval
```

```
bon95_intervals(bird_data)
```

```
##            V1       V2
## [1,] 189.8216 274.7819
## [2,] 197.4229 284.7736
```

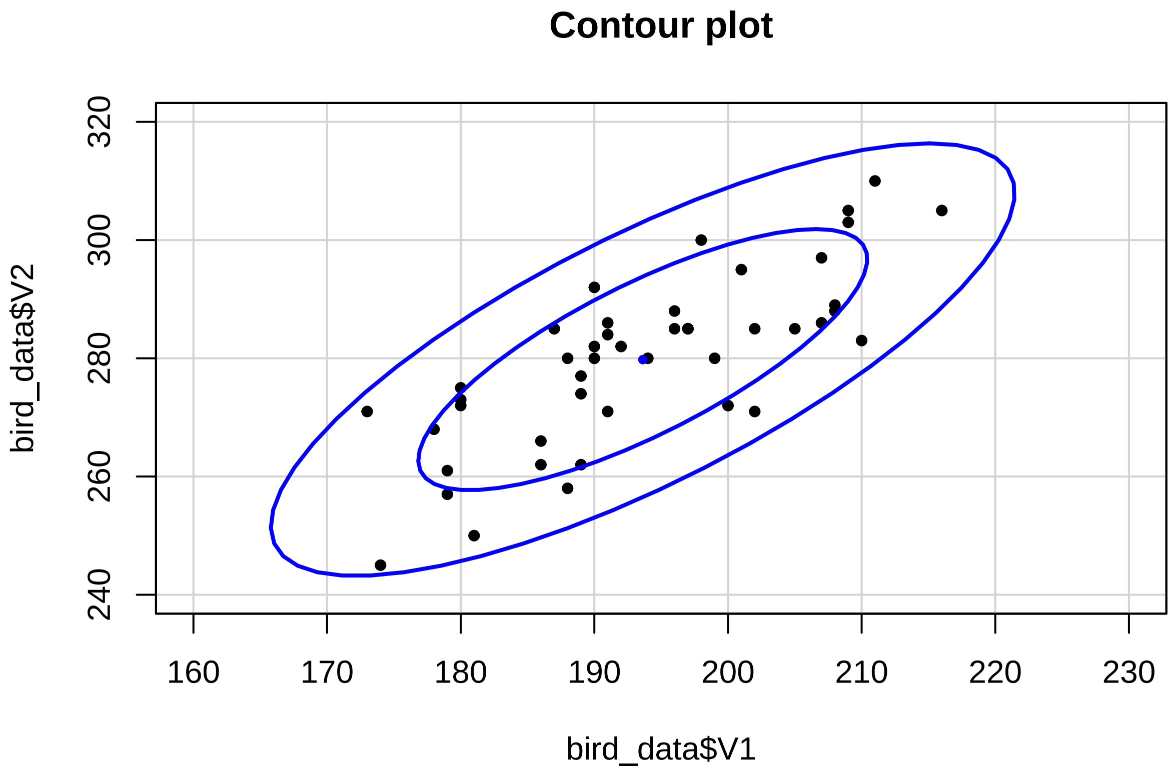
The only advantage we could find for T-squared intervals over Bonferroni interval is that, T-squared intervals account for the correlation between intervals, while Boneferroni does not.

c

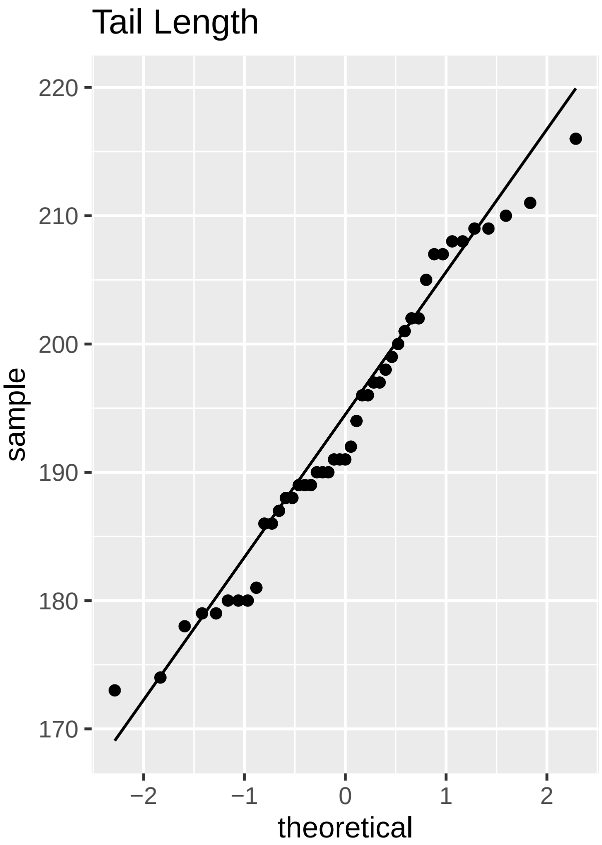
```
library(car)
```

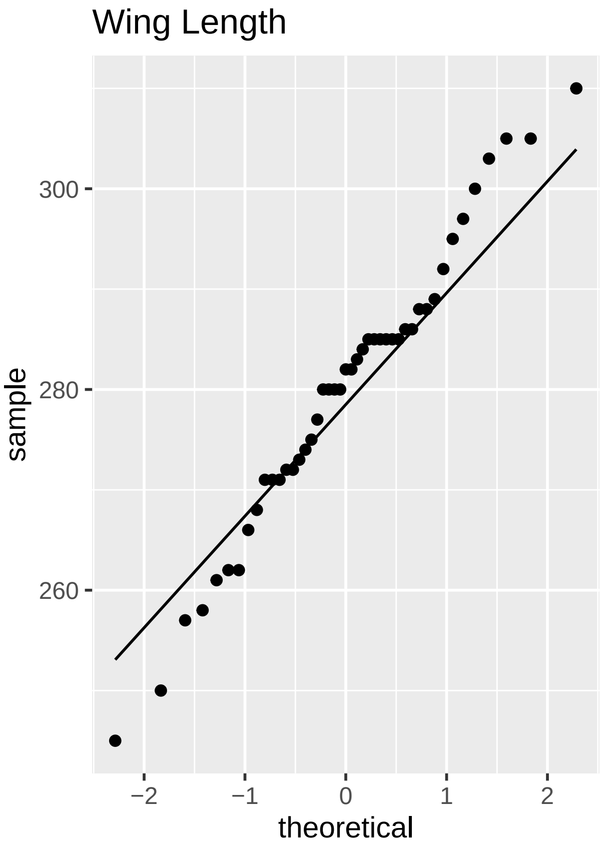
```
## Loading required package: carData
```

```
dataEllipse(x=bird_data$V1, y=bird_data$V2, pch=20, levels=c(0.68, 0.95),
            xlim=c(160, 230), ylim=c(240, 320), center.cex=0.5, main="Contour plot")

library(ggplot2)
```



```
library(gridExtra)
grid.arrange(ggplot(data = bird_data, aes(sample = V1)) + stat_qq() + stat_qq_line() + ggtitle("Tail Le
             ggplot(data = bird_data, aes(sample = V2)) + stat_qq() + stat_qq_line() + ggtitle("Wing Le
             ncol=2)
```





Since the data points go along the line in the qqplots, it shows that the data is normally distributed. A bivariate normal distribution would be a viable population model.