STAT636 - Exam 1

Daniel Osorio - dcosorioh@tamu.edu
Department of Veterinary Integrative Biosciences
Texas A&M University

1.
```
> print(Sigma <- matrix(c(1,0.125,0.2,
+                          0.125,0.25,0.1,
+                          0.2,0.1,0.64),
+                        ncol = 3,
+                        byrow = TRUE))
```

```
      [,1]  [,2] [,3]
[1,] 1.000 0.125 0.20
[2,] 0.125 0.250 0.10
[3,] 0.200 0.100 0.64
```

   (a) Correlation Value

```
> # cov(x,y)/ sd(x) * sd(y)
> Sigma[2,3] / (sqrt(Sigma[2,2]) * sqrt(Sigma[3,3]))
```

```
[1] 0.25
```

   (b) First eigen value and eigen vector of sigma

```
> eigenDSigma <- eigen(Sigma)
> # First eigen value of Sigma
> eigenDSigma$values[1]
```

```
[1] 1.116835
```

```
> # First eigen vector of Sigma
> eigenDSigma$vectors[,1]
```

```
[1] 0.8939588 0.1764355 0.4119565
```

   (c) Determinant of sigma

```
> det(Sigma)
```

```
[1] 0.135
```

   (d) Is Sigma PD? *Yes, because all of their eigen values are positive*

```
> all(eigenDSigma$values > 0)
```

```
[1] TRUE
```

   (e) What is the inverse of Σ?

```
> solve(Sigma)
```

```
           [,1]       [,2]       [,3]
[1,]  1.1111111 -0.4444444 -0.2777778
[2,] -0.4444444  4.4444444 -0.5555556
[3,] -0.2777778 -0.5555556  1.7361111
```

(f) What is the first eigenvalue and eigenvector of Σ^{-1}

```
> eigenDinvSigma <- eigen(solve(Sigma))
> eigenDinvSigma$values[1]
```

```
[1] 4.59644
```

```
> eigenDinvSigma$vectors[1,]
```

```
[1] 0.1103844 0.4343420 0.8939588
```

2. `> X <- read.csv("exam_1.csv")`

(a) 90 % CI region

```
> n <- nrow(X)
> p <- ncol(X)
> # Center
> print(x_bar <- colMeans(X))
```

```
        V1         V2         V3
0.12691538 0.07514347 0.16364740
```

```
> S <- var(X)
> eigen_S <- eigen(S)
> # Primary axes
> eigen_S$vectors
```

```
           [,1]       [,2]        [,3]
[1,] -0.4667773  0.8133957  0.34714044
[2,] -0.2391548  0.2618053 -0.93502029
[3,] -0.8514247 -0.5194665  0.07232275
```

```
> c2 <- (((n - 1) * p) / (n - p)) * qf(0.90, p, n - p)
> # Half lengths
> sqrt(eigen_S$values / n) * sqrt(c2)
```

```
[1] 0.6070969 0.2042265 0.1092275
```

```
> # 90% Confidence region
> print(CR90 <- sapply(seq_len(p), function(x){
+   x_bar[x] + c(-1, 1) * sqrt(c2 * S[x, x] / n)
+ }))
```

```
           [,1]       [,2]       [,3]
[1,] -0.2037449 -0.1102466 -0.3640836
[2,]  0.4575757  0.2605336  0.6913784
```

(b) Interpretation:*A 90% confidence region is a range of values that you can be certain contains the 90% of the population values.*

(c) Is the given value contained? *No.*

```
> mu_0 <- c(-0.5, 0, 0.5)
> all(CR90[1,] < mu_0 & CR90[2,] > mu_0)
```

```
[1] FALSE
```

(d) 90% Bonferroni simultaneous confidence intervals

```
> # Bonferroni
> sapply(seq_len(p), function(x){
+ x_bar[1] + c(-1, 1) * qt(1 - 0.1 / (2 * p), n - 1) * sqrt(S[1, 1] / n)
+ })
```

```
            [,1]       [,2]       [,3]
[1,] -0.1488520 -0.1488520 -0.1488520
[2,]  0.4026828  0.4026828  0.4026828
```

(e) Interpretation:*A 90% confidence interval is a range of values that you can be 90% certain contains the true mean of the population.*

(f)
```
> S_0 <- (t(X) - mu_0) %*% t(t(X) - mu_0) / (n - 1)
> # T2 Statistic Value
> print(T_2_stat <- n * t(x_bar - mu_0) %*% solve(S) %*% (x_bar - mu_0))
         [,1]
[1,] 90.75469
> # P - Value
> print(p_value <- 1 - pf((n - p) * T_2_stat / ((n - 1) * p), p, n - p))
            [,1]
[1,] 9.10505e-11
> T_2_stat <= c2
      [,1]
[1,] FALSE
```