Homework 2

ST557

Nick Sun

October 16, 2019

Question 1

X_1 and X_2 are not independent since they have nonzero covariance.

X_2 and X_3 are independent since they have zero covariance and are multivariate normal. They are also not functions of each other.

$Y = [X_1, X_2]$ and X_3 are independent since Y is a function of X_1 and X_2 which are both independent from X_3.

$Y = \frac{X_1+X_2}{2}$ and X_3 are independent since Y is a function of X_1 and X_2 which are both independent from X_3.

X_2 and $Y = X_2 - \frac{5}{2}X_1 - X_3$ are not independent since knowing about X_2 tells us something about Y.

Question 2

The `EuclideanDistance()` function in R can calculate the Euclidean distance.

```
mu_1 <- c(0, 0 ,0)
mu_2 <- c(3, 4, -3.5)
EuclideanDistance(c(1, 2, -2), mu_1)
```

```
## [1] 3
```

```
EuclideanDistance(c(1, 2, -2), mu_2)
```

```
## [1] 3.201562
```

The `mahalanobis()` function computes the squared Mahalanobis distance between X and μ with covariance matrix Σ

```
Sigma <- matrix(c(9.0, 8.1, -3.6, 8.1, 9.0, -4.8, -3.6, -4.8, 4.0), nrow=3, ncol=3)

sqrt(mahalanobis(c(1, 2, -2), mu_1, Sigma))
```

```
## [1] 1.063808
```

```
sqrt(mahalanobis(c(1, 2, -2), mu_2, Sigma))
```

```
## [1] 0.8387172
```

part c.

If $\bar{X}$ is the sample of n multivariate normal random vectors, then $\bar{X} \sim MVN(\mu, \frac{\Sigma}{n})$ Then $n(\bar{X}-\mu)^T\Sigma^{-1}(\bar{X}-\mu) = (\bar{X}-\mu)^T\left(\frac{\Sigma}{n}\right)^{-1}(\bar{X}-\mu) \sim \chi^2_p$. This quantity is also the squared Mahalanobis distance between $\bar{X}$ and μ

part d.

μ_2 has a small Mahalanobis distance from $\bar{X}$, so it would be more plausible of a population mean than μ_2. While μ_1 has a smaller Euclidean distance, it does not factor in the covariance of the data so we should opt to use Mahalanobis distance instead.

Question 3

$Y_1 = \frac{1}{5}X_1 + \frac{1}{5}X_2 + \frac{1}{5}X_3 + \frac{1}{5}X_4 + \frac{1}{5}X_5$ is equivalent to $\bar{X}$. As we discussed above, $\bar{X} \sim MVN(\mu, \frac{\Sigma}{n})$
$Y_2 = X_1 - X_2 + X_3 - X_4 + X_5 \sim MVN(\mu, 5\Sigma)$ since $\sum_{i=1}^n c_i X_i \sim MVN(\sum c_i\mu_i, \left(\sum c_i^2\right)\Sigma)$

Question 4

The MLE for the mean vector is just the sample mean vector

$$\begin{pmatrix} 4 \\ 6 \end{pmatrix}$$

```
mean(c(3, 4, 5, 4))
```

```
## [1] 4
```

```
mean(c(6, 4, 7, 7))
```

```
## [1] 6
```

The MLE for the covariance matrix is $\frac{n-1}{n}S$. We can calculate this in R using a few different methods, one of which is inputting the data vectors as a list and then performing matrix multiplication and summation over that list.

```
mat <- list(c(3,6),
            c(4,4),
            c(5,7),
            c(4,7))

(1/4) * Reduce('+', lapply(mat, function(x) (x - c(4,6)) %*% t(x - c(4, 6))))
```

```
##      [,1] [,2]
## [1,] 0.50 0.25
## [2,] 0.25 1.50
```

Question 5

Testing performance of the correlation normality test

Part a.

For data drawn from a standard Uniform distribution, we get the following simulation:

```
corr_stat <- function(n) {

    data <- runif(n)
    sample_quantiles <- sort(data)
    theoretical_quantiles <- qnorm(c(1:n - .5)/n)
    xbar <- mean(sample_quantiles)
    qbar <- mean(theoretical_quantiles)

    numerator <- sum((sample_quantiles - xbar)*(theoretical_quantiles - qbar))
    sample_denom <- sqrt(sum((sample_quantiles - xbar)^2))
    theory_denom <- sqrt(sum((theoretical_quantiles - qbar)^2))
    r_Q <- numerator/(sample_denom * theory_denom)
    return(r_Q)

}

pvals <- vector(length = 10000, mode = "numeric")
start <- Sys.time()
for (i in 1:10000) {
  pvals[i] <- corr_stat(10)
}
end <- Sys.time()
end - start
```

```
## Time difference of 0.899482 secs
```

```
sum(pvals < .9198)/10000
```

```
## [1] 0.058
```

This is an awful level of rejection! For this sample size, the power of the test is quite low.

Part b.

```
corr_stat_chisq <- function(n, df) {

    data <- rchisq(n, df = df)
    sample_quantiles <- sort(data)
    theoretical_quantiles <- qnorm(c(1:n - .5)/n)
    xbar <- mean(sample_quantiles)
    qbar <- mean(theoretical_quantiles)

    numerator <- sum((sample_quantiles - xbar)*(theoretical_quantiles - qbar))
    sample_denom <- sqrt(sum((sample_quantiles - xbar)^2))
    theory_denom <- sqrt(sum((theoretical_quantiles - qbar)^2))
    r_Q <- numerator/(sample_denom * theory_denom)
    return(r_Q)
```

```
}

pvals <- vector(length = 10000, mode = "numeric")
start <- Sys.time()
for (i in 1:10000) {
  pvals[i] <- corr_stat_chisq(5, 5)
}
end <- Sys.time()

sum(pvals < .8788)/10000
```

```
## [1] 0.086
```

This is still a pretty poor level of rejection!

Part c.

```
pvals <- vector(length = 10000, mode = "numeric")
start <- Sys.time()
for (i in 1:10000) {
  pvals[i] <- corr_stat_chisq(20, 2)
}
end <- Sys.time()

sum(pvals < .9508)/10000
```

```
## [1] 0.8008
```

This level of rejection is *somewhat decent*. I would say this test is basically powerless *unless* there is a decent sample size of approximately 20 observations.