MAST30025: Linear Statistical Models

Week 5 Lab

1. Consider the dataset from the Week 4 lab. As before, do the following using both matrix calculations and R's `lm` commands.
   (a) Calculate 95% confidence intervals for the model parameters.
   (b) Calculate a 95% confidence interval for the average income of a person who has had 18 years of formal education.
   (c) Calculate a 95% prediction interval for the income of a single person who has had 18 years of formal education.

2. For simple linear regression, $y = \beta_0 + \beta_1 x + \varepsilon$, show that a $100(1-\alpha)\%$ confidence interval for the mean response when $x = x^*$ can be written as

$$b_0 + b_1 x^* \pm t_{\alpha/2} s \sqrt{\frac{1}{n} + \frac{(x^* - \bar{x})^2}{s_{xx}}}$$

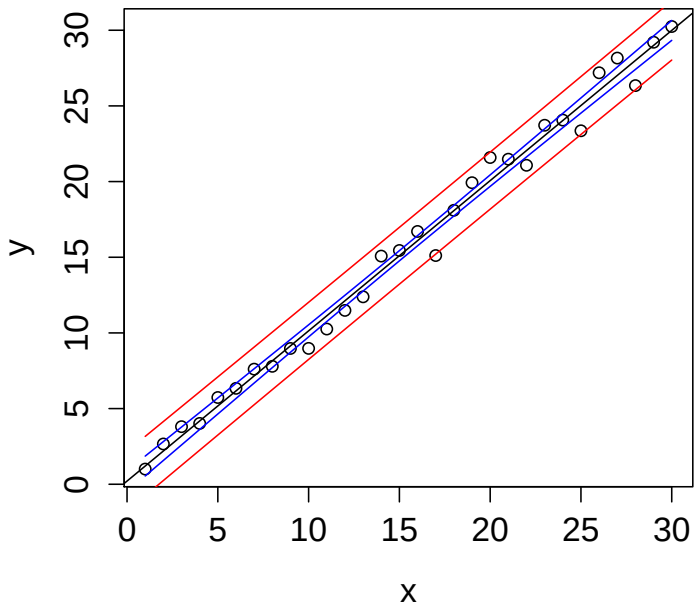
   where $s_{xx} = \sum_{i=1}^n x_i^2 - n\bar{x}^2$.
   Similarly, show that a $100(1-\alpha)\%$ prediction interval for a new response when $x = x^*$ can be written as

$$b_0 + b_1 x^* \pm t_{\alpha/2} s \sqrt{1 + \frac{1}{n} + \frac{(x^* - \bar{x})^2}{s_{xx}}}.$$

3. We can generate some data for a simple linear regression as follows:

```
> n <- 30
> x <- 1:n
> y <- x + rnorm(n)
```

   Construct 95% CI's for $\mathbb{E}y$ and 95% PI's for y, when $x = 1, 2, \ldots, n$. Join them up and plot them on a graph of y against x. Your plot should look something like this:



What proportion of the y's should you expect to lie beyond the outer lines?

4. In generalised least squares we have $\mathbf{y} \sim N(X\boldsymbol{\beta}, V)$.

   Let R be the principal square root of V^{-1} (why does R exist?). Put $\mathbf{v} = R\mathbf{y}$, and show that $\mathbf{v} \sim MVN(Z\boldsymbol{\beta}, I)$ for some Z. From the usual least squares estimate of $\boldsymbol{\beta}$ using $\mathbf{v}$, obtain the generalised least squares estimate of $\boldsymbol{\beta}$ using $\mathbf{y}$.

5. In this exercise, we look at the dangers of overfitting. Generate some observations from a simple linear regression:

```
> set.seed(3)
> X <- cbind(rep(1,100), 1:100)
> beta <- c(0, 1)
> y <- X %*% beta + rnorm(100)
```

   Put aside some of the data for testing and some for fitting:

```
> Xfit <- X[1:50,]
> yfit <- y[1:50]
> Xtest <- X[51:100,]
> ytest <- y[51:100]
```

   (a) Using only the fitting data, estimate $\boldsymbol{\beta}$ and hence the residual sum of squares. Also calculate the residual sum of squares for the test data, that is, $\sum_{i=51}^{100}(y_i - b_0 - b_1 x_i)^2$.

   Now add 10 extra predictor variables which we know have nothing to do with the response:

```
> X <- cbind(X, matrix(runif(1000), 100, 10))
> Xtest <- X[51:100,]
> Xfit <- X[1:50,]
```

   Again using only the fitting data, fit the linear model $\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, and show that the residual sum of squares has reduced (this has to happen). Then show that the residual sum of squares for the test data has gone up (this happens most of the time).

   Explain what is going on.

   (b) Repeat the above, but this time add x^2, x^3 and x^4 terms:

```
> X <- cbind(X[, 1:2], (1:100)^2, (1:100)^3, (1:100)^4)
```

Programming questions

1. What will be the output of the following code? Try to answer this without typing it up.

```
fb <- function(n) {
    if (n == 1 || n == 2) {
        return(1)
    } else {
        return(fb(n - 1) + fb(n - 2))
    }
}
fb(8)
```

2. Suppose you needed all 2^n binary sequences of length n. One way of generating them is with nested for loops. For example, the following code generates a matrix `binseq`, where each row is a different binary sequence of length three.

```
> binseq <- matrix(nrow = 8, ncol = 3)
> r <- 0 # current row of binseq
> for (i in 0:1) {
+   for (j in 0:1) {
+     for (k in 0:1) {
```

```
+       r <- r + 1
+       binseq[r,] <- c(i, j, k)
+     }
+   }
+ }
> binseq

     [,1] [,2] [,3]
[1,]    0    0    0
[2,]    0    0    1
[3,]    0    1    0
[4,]    0    1    1
[5,]    1    0    0
[6,]    1    0    1
[7,]    1    1    0
[8,]    1    1    1
```

Clearly this approach will get a little tedious for large n. An alternative is to use recursion. Suppose that A is a matrix of size $2^n \times n$, where each row is a different binary sequence of length n. Then the following matrix contains all binary sequences of length $n+1$:

$$\left(\begin{array}{c|c} \mathbf{0} & A \\ \hline \mathbf{1} & A \end{array}\right).$$

Here $\mathbf{0}$ is a vector of zeros and $\mathbf{1}$ is a vector of ones.

Use this idea to write a recursive function `binseq`, which takes as input an integer n and returns a matrix containing all binary sequences of length n, as rows of the matrix. You should find the functions `cbind` and `rbind` particularly useful.

3. Let $A = (a_{i,j})_{i,j=1}^n$ be a square matrix, and denote by $A_{(-i,-j)}$ the matrix with row i and column j removed. If A is a 1×1 matrix then $det(A)$, the determinant of A, is just $a_{1,1}$. For $n \times n$ matrices we have, for any i,

$$det(A) = \sum_{j=1}^{n} (-1)^{i+j} \, a_{i,j} \, det(A_{(-i,-j)}).$$

Use this to write a *recursive* function to calculate $det(A)$.

Question 2

Claim: $b_0 + b_1 x^* \pm t_{\frac{\alpha}{2}} S \sqrt{\frac{1}{n} + \frac{(x^* - \bar{x})^2}{S_{xx}}}$ where $S_{xx} = \sum_{i=1}^{n} x_i^2 - n\bar{x}^2$

is the Confidence Interval.

* For linear regression, we have that

$$X^T X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & \vdots \\ 1 & x_n \end{bmatrix} = \begin{bmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix}$$

$$(X^T X)^{-1} = \frac{1}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \begin{bmatrix} \sum_{i=1}^{n} x_i^2 & -\sum_{i=1}^{n} x_i \\ -\sum_{i=1}^{n} x_i & n \end{bmatrix}$$

Let $t = [1 \quad x^*]^T$

* Based from the lecture slide, we have that CI for $t^T\beta$ is given by

$$t^T b \pm t_{\frac{\alpha}{2}} S \sqrt{t^T (X^T X)^{-1} t} = b_0 + b_1 x^* \pm t_{\frac{\alpha}{2}} S \sqrt{t^T (X^T X)^{-1} t}$$

$\Rightarrow$ Calculate $t^T (X^T X)^{-1} t$

$$= [1 \quad x^*] \frac{1}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \left[\begin{array}{c|c} \sum_{i=1}^{n} x_i^2 & -\sum_{i=1}^{n} x_i \\ -\sum_{i=1}^{n} x_i & n \end{array}\right] \begin{bmatrix} 1 \\ x^* \end{bmatrix}$$

$$= \frac{1}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \left[\sum_{i=1}^{n} x_i^2 - x^* \sum_{i=1}^{n} x_i, \; -\sum_{i=1}^{n} x_i + n x^*\right] \left[\begin{array}{c} 1 \\ \hline x \end{array}\right]$$

$$= \frac{1}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \sum_{i=1}^{n} x_i^2 - x^* \sum_{i=1}^{n} x_i - x^* \sum_{i=1}^{n} x_i + n(x^*)^2 \underbrace{- 2x^* \sum_{i=1}^{n} x_i}$$

$$= \frac{1}{n S_{xx}} \left[\sum_{i=1}^{n} x_i^2 - n\bar{x}^2 + n\bar{x}^2 - 2n x^* \bar{x} + n(x^*)^2\right] \quad (n\bar{x}^2 - n\bar{x}^2 = 0)$$

$$= \frac{1}{n S_{xx}} \left[S_{xx} + n(x^* - \bar{x})^2\right]$$

$$= \frac{1}{n} + \frac{(x^* - \bar{x})^2}{S_{xx}} \quad \text{as required}$$

QED

* Similar proof for Prediction interval

Question 4

$y \sim MVN(X\beta, V)$

R be principal square root of V^{-1} which is pos definite and symmetric

Let $V = Ry$

Claim: Show that $V \sim MVN(Z\beta, I)$ for some Z

Answer

V is symmetric and positive definite, thus $V = P \Lambda P^T$ (can be diagonalized)

$\Rightarrow V^{-1} = P \Lambda^{-1} P^T$

$\Rightarrow R = P \Lambda^{-\frac{1}{2}} P^T$ (principal root of V^{-1})

* we know that $y \sim MVN(X\beta, V)$

$\Rightarrow V = Ry$ follows MVN with

$$E[Ry] = R\,E[y]$$
$$= P^T \Lambda^{-\frac{1}{2}} P\, E[y]$$
$$= RX\beta$$

$$Var[Ry] = R V R^T$$
$$= P \Lambda^{\frac{1}{2}} P^T V\, P \Lambda^{-\frac{1}{2}} P^T$$

($V = P \Lambda P^T$)

$$= P \Lambda^{\frac{1}{2}} \Lambda^{-\frac{1}{2}} P^T$$
$$= I$$

$\therefore Ry \sim MVN(RX\beta, I)$

* So $Z = RX$, we have that the LS notes,

$$(Z^T Z)^{-1} Z^T z = (X^T \underbrace{R^T R}_{V^{-1}} X)^{-1} X^T R^T Ry$$

$$= \boxed{(X^T V^{-1} X)^{-1} X^T V^{-1} y}$$