Lab 3: An Overview of Linear Regression and Data Generation

EDP 380C.28: Simulation in R

Fall 2022 – Unique: 12104

Brian T. Keller, Ph.D.
bk@utexas.edu
University of Texas at Austin
September 18, 2022

Objectives

- Learn about the OLS solution to linear regression in scalar and matrix form.
- Learn to generate data for regression models based on chosen constraints.
- Practice implementing matrix equations into R .

---

1 Foundations of Ordinary Least Squares Regression

One way to view regression is as a prediction problem. Ultimately, we are trying to fit some line, be it linear or curvilinear, to predict or explain the values of our outcome, Y. To accomplish this prediction problem, we create a composite predicted score for Y, denoted as $\hat{Y}$, by weighting our predictors with regression coefficients (β, our parameters) and summing them up. As mentioned previously, we are creating a linear combination of our predictors and parameters to act as a proxy for the value of Y for a given observation. In its most basic scalar form, we can express a linear regression for observation as follows

$$y_i = \beta_0 + \beta_1 x_i + e_i$$

where y_i is the observed value of Y (our criterion) for observation i, x_i is the observed value of X (our predictor) for observation i, and e_i is the residual of our prediction ($y_i - \hat{y}_i$) for observation i. In addition, we have two parameters, β_0, the intercept of our line, and β_1, the weight for our predictor X. The ultimate goal of our prediction problem is to solve for the values of our weights (β_0 and β_1) that maximize our prediction.

1.1 Solving OLS Regression with One Predictor

While we have set up the prediction problem, we have not discussed how to 'maximize' the prediction. As with many things, ultimately, there has to be a definition of criteria to represent a 'good' prediction. In our case, a perfect prediction would represent the best prediction (i.e., $Y - \hat{Y} = 0$). By expressing our criteria in terms of a function, we can use mathematical optimization theory to determine the parameters that get us the closest to our function. This function is often referred to as a ***loss function*** or ***cost function***. Ultimately with optimization, it is often easier to express our

problem in terms of minimization. Therefore, we generally are concerned with only the magnitude of our misprediction for a given observation (i.e., the residual e_i), not the direction (negative or positive). To deal with the magnitude, we could take the absolute value of the deviation score and minimize that:

$$\min\left\{\left|Y-\hat{Y}\right|\right\}$$

Unfortunately, this is a bit tricky to deal with and does not offer easy solutions. A second alternative would be to square the deviations and minimize those.

$$\min\left\{\left(Y-\hat{Y}\right)^2\right\}$$

The above equation is known as the ***least squares*** estimator. In its simplest form (above), it is often called the ***ordinary least squares*** (OLS) estimator. More specifically, because we have n observations, we must minimize the ***sum of squared residuals***.

$$\min\left\{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2\right\}$$

Thus far, we have just set up the prediction problem and the mathematical optimization required to solve it. To find the solution, there are three general steps:

(1) State the function to be minimized in terms of the unknown values.

(2) Differentiate the function in Step 1 with respect to the unknown values.

(3) Set the derivatives from Step 2 to zero and solve. These will give you the ***normal equations***.

Below, we will look at these steps in more detail for the simple linear regression.

Step 1 – State the function to be minimized in terms of the unknown values.

Substitute regression formula for $\hat{Y}$ and then simplify to remove the square.

$$\begin{aligned}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 &= \sum_{i=1}^{n}\left(y_i-[\beta_0+\beta_1x_i]\right)^2\\ &= \sum_{i=1}^{n}(y_i-\beta_0-\beta_1x_i)^2\\ &= \sum_{i=1}^{n}\left(y_i^2+\beta_0^2+\beta_1^2x_i^2-2\beta_0y_i-2\beta_1x_iy_i+2\beta_0\beta_1x_i\right)\end{aligned}$$

Step 2 – Differentiate the function in Step 1 with respect to the unknown values.

Partial derivative for β_0:

$$\begin{aligned}\frac{\partial\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\partial\beta_0} &= \frac{\partial}{\partial\beta_0}\sum_{i=1}^{n}\left(y_i^2+\beta_0^2+\beta_1^2x_i^2-2\beta_0y_i-2\beta_1x_iy_i+2\beta_0\beta_1x_i\right)\\ &= \sum_{i=1}^{n}(2\beta_0-2y_i+2\beta_1x_i)\\ &= 2n\beta_0-2\sum_{i=1}^{n}(y_i)+2\beta_1\sum_{i=1}^{n}(x_i)\end{aligned}$$

Partial derivative for β_1:

$$\begin{aligned}\frac{\partial \sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\partial \beta_1} &= \frac{\partial}{\partial \beta_1}\sum_{i=1}^{n}\left(y_i^2+\beta_0^2+\beta_1^2x_i^2-2\beta_0y_i-2\beta_1x_iy_i+2\beta_0\beta_1x_i\right)\\ &= \sum_{i=1}^{n}\left(2\beta_1x_i^2-2x_iy_i+2\beta_0x_i\right)\\ &= 2\beta_1\sum_{i=1}^{n}\left(x_i^2\right)-2\sum_{i=1}^{n}\left(x_iy_i\right)+2\beta_0\sum_{i=1}^{n}\left(x_i\right)\end{aligned}$$

Step 3 – Set the derivatives from Step 2 to zero and solve.

We have the following system of two equations with two unknowns.

$$\begin{aligned}2n\beta_0-2\sum_{i=1}^{n}\left(y_i\right)+2\beta_1\sum_{i=1}^{n}\left(x_i\right)&=0\\ 2\beta_1\sum_{i=1}^{n}\left(x_i^2\right)-2\sum_{i=1}^{n}\left(x_iy_i\right)+2\beta_0\sum_{i=1}^{n}\left(x_i\right)&=0\end{aligned}$$

First, we will simplify the equations and rearrange them.

$$\begin{aligned}n\beta_0+\beta_1\sum_{i=1}^{n}\left(x_i\right)&=\sum_{i=1}^{n}\left(y_i\right)\\ \beta_0\sum_{i=1}^{n}\left(x_i\right)+\beta_1\sum_{i=1}^{n}\left(x_i^2\right)&=\sum_{i=1}^{n}\left(x_iy_i\right)\end{aligned}$$

Next, we set up a matrix representation to obtain the solutions.

$$\begin{bmatrix} n & \sum_{i=1}^{n}\left(x_i\right)\\ \sum_{i=1}^{n}\left(x_i\right) & \sum_{i=1}^{n}\left(x_i^2\right)\end{bmatrix}\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix}=\begin{bmatrix}\sum_{i=1}^{n}\left(y_i\right)\\ \sum_{i=1}^{n}\left(x_iy_i\right)\end{bmatrix}$$

We can use a matrix inverse to solve for β_0 and β_1.

$$\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix}=\begin{bmatrix} n & \sum_{i=1}^{n}\left(x_i\right)\\ \sum_{i=1}^{n}\left(x_i\right) & \sum_{i=1}^{n}\left(x_i^2\right)\end{bmatrix}^{-1}\begin{bmatrix}\sum_{i=1}^{n}\left(y_i\right)\\ \sum_{i=1}^{n}\left(x_iy_i\right)\end{bmatrix} \tag{1}$$

Although Equation (1) provides us with a solution to the problem that we could use R to solve, let us continue to simplify the solution by applying the matrix inverse of a 2×2 matrix ourselves. The inverse of a 2×2 matrix follows:

$$\begin{bmatrix}a & b\\ c & d\end{bmatrix}^{-1}=\frac{1}{ad-bc}\begin{bmatrix}d & -b\\ -c & a\end{bmatrix}$$

Therefore, using the inverse formula, we obtain the following solution

$$\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix}=\frac{1}{n\sum_{i=1}^{n}\left(x_i^2\right)-\left(\sum_{i=1}^{n}x_i\right)^2}\begin{bmatrix}\sum_{i=1}^{n}\left(x_i^2\right) & -\sum_{i=1}^{n}\left(x_i\right)\\ -\sum_{i=1}^{n}\left(x_i\right) & n\end{bmatrix}\begin{bmatrix}\sum_{i=1}^{n}\left(y_i\right)\\ \sum_{i=1}^{n}\left(x_iy_i\right)\end{bmatrix},$$

and we can further simplify by multiplying the two matrices together.

$$\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix} = \frac{1}{n\sum_{i=1}^{n}(x_i^2) - \left(\sum_{i=1}^{n} x_i\right)^2}\begin{bmatrix}\sum_{i=1}^{n}(x_i^2)\sum_{i=1}^{n}(y_i) - \sum_{i=1}^{n}(x_i)\sum_{i=1}^{n}(x_i y_i)\\ n\sum_{i=1}^{n}(x_i y_i) - \sum_{i=1}^{n}(x_i)\sum_{i=1}^{n}(y_i)\end{bmatrix} \tag{2}$$

Next, we can take the denominator of Equation (2) and multiply it by $\frac{1}{n}$ to obtain the sum of squares formula.

$$\begin{aligned}\frac{1}{n}\left[n\sum_{i=1}^{n}(x_i^2) - \left(\sum_{i=1}^{n} x_i\right)^2\right] &= \sum_{i=1}^{n}(x_i^2) - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}\\ &= \sum_{i=1}^{n}(x_i - \overline{x})^2\end{aligned}$$

Therefore, the expression is equal to $n\sum_{i=1}^{n}(x_i - \overline{x})^2$ because we must multiply by $\frac{n}{n}$ instead. Plugging this result into Equation (2) results in the following equation

$$\begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix} = \frac{1}{n\sum_{i=1}^{n}(x_i - \overline{x})^2}\begin{bmatrix}\sum_{i=1}^{n}(x_i^2)\sum_{i=1}^{n}(y_i) - \sum_{i=1}^{n}(x_i)\sum_{i=1}^{n}(x_i y_i)\\ n\sum_{i=1}^{n}(x_i y_i) - \sum_{i=1}^{n}(x_i)\sum_{i=1}^{n}(y_i)\end{bmatrix},$$

and we can now use it to provide a compact solutions to both parameters.

$$\beta_0 = \frac{\sum_{i=1}^{n}(x_i^2)\sum_{i=1}^{n}(y_i) - \sum_{i=1}^{n}(x_i)\sum_{i=1}^{n}(x_i y_i)}{n\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

$$\beta_1 = \frac{n\sum_{i=1}^{n}(x_i y_i) - \sum_{i=1}^{n}(x_i)\sum_{i=1}^{n}(y_i)}{n\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

Focusing on β_1 first, if we divide the numerator by n (i.e., distribution n in the denominator to the numerator) we obtain the following expression in the numerator.

$$\sum_{i=1}^{n}(x_i y_i) - \frac{\sum_{i=1}^{n}(x_i)\sum_{i=1}^{n}(y_i)}{n} = \sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})$$

This expression is equal to the sum of products; thus, by distributing n in the denominator we obtain the familiar solution for β_1:

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2} = \frac{SP_{xy}}{SS_x} \tag{3}$$

Now that we have solved β_1, for simplicity let's returning to the derivative for β_0 which required β_1 to solve.

$$\begin{aligned}2n\beta_0 - 2\sum_{i=1}^{n}(y_i) + 2\beta_1\sum_{i=1}^{n}(x_i) &= 0\\ 2n\beta_0 &= 2\sum_{i=1}^{n}(y_i) - 2\beta_1\sum_{i=1}^{n}(x_i)\\ \beta_0 &= \frac{2\sum_{i=1}^{n}(y_i) - 2\beta_1\sum_{i=1}^{n}(x_i)}{2n}\\ &= \frac{\sum_{i=1}^{n}(y_i)}{n} - \beta_1\left(\frac{\sum_{i=1}^{n} x_i}{n}\right)\\ &= \overline{y} - \beta_1\overline{x}\end{aligned}$$

Therefore the standard closed-form solutions to the ordinary least squares regression coefficients are as follows.

$$\beta_0 = \overline{y} - \beta_1 \overline{x} = \overline{y} - \left(\frac{SP_{xy}}{SS_x} \right) \overline{x} \tag{4}$$

$$\beta_1 = \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n} (x_i - \overline{x})^2} = \frac{SP_{xy}}{SS_x} \tag{5}$$

Note that the above result is also equal to the following (as a sample or population statistic),

$$\begin{aligned} \beta_1 &= \frac{SP_{xy}}{SS_x} \times \frac{(n-1)^{-1}}{(n-1)^{-1}} = \frac{s_{xy}}{s_x^2} \\ &= \frac{SP_{xy}}{SS_x} \times \frac{(N)^{-1}}{(N)^{-1}} = \frac{\sigma_{xy}}{\sigma_x^2} \end{aligned} \tag{6}$$

The results from Equation (6) should seem familiar because they have already shown up in various data generating processes. For example, in Lab 1, we generated bivariate normally distributed data by regressing Y onto X via the following equation (i.e., Equation 10 in Lab 1).

$$f\left(Y \mid X, U \right) \Rightarrow \quad Y = \mu_y + \left(\frac{\rho \sigma_y}{\sigma_x} \right) (X - \mu_x) + U \sigma_y \sqrt{1 - \rho^2}$$

Now let's rewrite the above equation to map onto the parameters of the simple linear regression.

$$\begin{aligned} Y &= \mu_y + \left(\frac{\rho \sigma_y}{\sigma_x} \right) (X - \mu_x) + U \sigma_y \sqrt{1 - \rho^2} \\ &= \mu_y + \left(\frac{\sigma_x \sigma_y}{\sigma_x \sigma_y} \right) \left(\frac{\rho \sigma_y}{\sigma_x} \right) (X - \mu_x) + E \\ &= \mu_y + \left(\frac{\sigma_{xy}}{\sigma_x^2} \right) (X - \mu_x) + E \\ &= \mu_y - \left(\frac{\sigma_{xy}}{\sigma_x^2} \right) (\mu_x) + \left(\frac{\sigma_{xy}}{\sigma_x^2} \right) (X) + E \\ Y &= \beta_0 + \beta_1 (X) + E \end{aligned}$$

Although this is nothing profound, the illustration should give an intuition about how data generation processes often work. Furthermore, this relationship of reparameterizing a regression equation to the population means, variances, and covariances will prove to be helpful for deriving the true population parameters in methodological simulations.

Similarly, the regression represents a model for the conditional distribution of our outcome given our predictors; our functional notation defines this as $f\left(Y \mid X \right)$. Under typical assumptions (i.e., independent and identically distributed residuals, etc.), the regression model implies a mean and variance structure for the conditional distribution.

$$\begin{aligned} \mathrm{E}(Y) &= \mathrm{E}\left(Y \mid X \right) + \mathrm{E}(E) \\ \mu_y &= \beta_0 + \beta_1 \mu_x \end{aligned} \tag{7}$$

$$\begin{aligned} \mathrm{Var}(Y) &= \mathrm{Var}\left(Y \mid X \right) + \mathrm{Var}(E) \\ \sigma_y^2 &= \beta_1^2 \sigma_x^2 + \sigma_e^2 \end{aligned} \tag{8}$$

The model implied mean structure ought to be unsurprising because it can be directly derived from the solution to the intercept (i.e., solving for μ_y in Equation 4). If we turn to the model implied

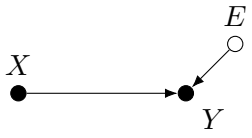

Figure 1: DAG For a Single Predictor Regression

variance, the residual score is uncorrelated with the predictors, thus leaving an additive effect. Var $\left(Y \mid X \right)$ represents the variance explained by X. This directly maps onto what we have already derived in Lab 1 (i.e., Equation 8 in Lab 1):

$$\text{Var}\left(Y \mid X \right) = \rho^2 \left(\frac{\sigma_y^2}{\sigma_x^2} \right).$$

From the above expression, we can easily obtain the result in Equation (8) because we have already showed that

$$\frac{\rho\sigma_y}{\sigma_x} = \frac{\sigma_{xy}}{\sigma_x^2} = \beta_1.$$

1.2 Generating Data for Simple Regression

Figure 1 presents the data generating process for a simple linear regression and is similar to a DAG from Lab 1. Instead of U, we will use E to map onto our regression notation, and in many cases, we will drop it from our discussion and focus on the parameterization of Y. By assuming normal distributions, we obtain the following two steps using the functional notation to trace the steps for generating Y and X.

$$\begin{aligned} f\left(X \mid \mu_x, \sigma_x \right) \Rightarrow &\quad X \sim \mathcal{N}\left(\mu_x, \sigma_x \right) \\ f\left(Y \mid X, \beta_0, \beta_1, \sigma_e \right) \Rightarrow &\quad Y \sim \mathcal{N}\left(\beta_0 + \beta_1 X, \sigma_e \right) \end{aligned} \tag{9}$$

1.a In R:

(1) Predefine the parameters for X and Y's distributions. Do this via two `list()` for X and Y, respectively. Set $\mu_x = -2$, $\sigma_x = 3$, $\beta_0 = 12$, $\beta_1 = 4$, and $\sigma_e = 7$.

(2) Create a function that generates X and Y based on Equation (9). The inputs should have the following form:

```
function(n, p_x, p_y)
```

and the output should be a $n \times 2$ column matrix (i.e., `cbind(x = x, y = y)`). As a reminder, follow the Style Guide with good commenting practices to document both functions' inputs and outputs.

(3) Create an analyze function that takes one data set and outputs a one-dimensional vector (i.e., `c()`) with the following labels and ordering:

`m_x`, `m_y`, `s_x`, `s_y`, `cor`, `b0`, `b1`, and `s_e`.

Hint: You can use the `lm()` function to obtain the regression coefficients and residual standard deviation.

(4) Set the seed to `17290`.

(5) Set up and run a simulation to generate 500 reps of $n = 100$ using the replicate function.

(6) Use the analyze function to analyze and compute the empirical mean and standard deviation of the parameter estimates.

Answer:

Create a `list(mean, sd)` that holds the mean and standard deviation of the parameter estimates from the simulation. Make sure to label each item of the list. Print the list and include its output as a comment in your code.

2 Multiple Regression with Matrices

Multiple regression falls compactly into matrices that simplify the representation and the solution. To illustrate, suppose we have the following multiple regression with p predictors.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + e_i$$

We can compactly write this in matrix notation for all n participants as follows.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \left[\begin{array}{c|ccc} 1 & x_{11} & \ldots & x_{1p} \\ 1 & x_{21} & \ldots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \ldots & x_{np} \end{array}\right] \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

More formally, we can write the above expression as labeled matrices as follows

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e} \tag{10}$$

with the matrices defined below.

$$\underset{n\times 1}{\mathbf{y}} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad \underset{n\times(p+1)}{\mathbf{X}} = \left[\begin{array}{c|ccc} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \dots & x_{np} \end{array}\right] \qquad \underset{(p+1)\times 1}{\boldsymbol{\beta}} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} \qquad \underset{n\times 1}{\mathbf{e}} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

Notably, the **X** matrix is an augmented data matrix (referred to as the ***design matrix***) that includes a column vector of ones. This vector of ones represents the intercept and allows us to include β_0 in the column vector of regression coefficients ($\boldsymbol{\beta}$).

2.1 Matrix Solution to Multiple Regression

Recall the three steps we used to find the solution in the single predictor case. We can follow those same steps to obtain the matrix solution to multiple regression.

(1) State the function to be minimized in terms of the unknown values.

$$\min\left\{\left(\mathbf{y}-\hat{\mathbf{y}}\right)'\left(\mathbf{y}-\hat{\mathbf{y}}\right)\right\} = \min\left\{\mathbf{e}'\mathbf{e}\right\}$$

(2) Differentiate the function in Step 1 with respect to the unknown values.

$$\frac{\partial \mathbf{e}'\mathbf{e}}{\partial \boldsymbol{\beta}} = -2\mathbf{X}'\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)$$

(3) Set the derivatives from Step 2 to zero and solve.

$$\begin{aligned} 0 &= -2\mathbf{X}'\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right) \\ 0 &= \mathbf{X}'\mathbf{y} - \mathbf{X}'\mathbf{X}\boldsymbol{\beta} \\ \mathbf{X}'\mathbf{y} &= \mathbf{X}'\mathbf{X}\boldsymbol{\beta} \end{aligned}$$

The above result is considered the ***normal equations*** for the least squares solution. We can then solve for $\boldsymbol{\beta}$ to obtain the solution:

$$\boldsymbol{\beta} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{y} \tag{11}$$

Despite the simplicity of this solution, this is nothing new or magical. We have, in fact, already done this solution. To illustrate, let's look at the parts of the solution for a single predictor in matrix form.

$$\mathbf{X}'\mathbf{X} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \end{bmatrix}\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} = \begin{bmatrix} n & \sum_{i=1}^{n}(x_i) \\ \sum_{i=1}^{n}(x_i) & \sum_{i=1}^{n}(x_i^2) \end{bmatrix}$$

$$\mathbf{X}'\mathbf{y} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n}(y_i) \\ \sum_{i=1}^{n}(x_i y_i) \end{bmatrix}$$

These expressions should be familiar because they appear in Equation (1); however, they are now more general and work for any number of predictors.

2.2 Other Regression Quantities in Matrix Form

Fitted Values

The fitted values (i.e., predicted scores, $\hat{y}_i$) can also be represented in matrix form.

$$\underset{n\times 1}{\hat{\mathbf{y}}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \mathbf{X}\boldsymbol{\beta}$$

Hat Matrix

The ***hat matrix*** (**H**) is a symmetric projection matrix that transforms the outcome vector into the predicted score. We can derive it as follows

$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{X}\boldsymbol{\beta} \\ &= \mathbf{X}\left[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}\right] \\ &= \mathbf{H}\mathbf{y} \end{aligned}$$

where

$$\underset{n\times n}{\mathbf{H}} = \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'.$$

Notably, the diagonal elements are measures of ***leverage***. The sum of all of the diagonal elements of the Hat matrix is equal to $p + 1$ (i.e., the number of predictors plus the intercept).

Residuals

We can compute the residual column vector (**e**) in multiple ways.

$$\begin{aligned} \mathbf{e} &= \mathbf{y} - \hat{\mathbf{y}} \\ &= \mathbf{y} - \mathbf{H}\mathbf{y} \\ &= (\mathbf{I} - \mathbf{H})\,\mathbf{y} \end{aligned}$$

Standard Deviation of the Residuals

Another essential quantity in regression is the standard deviation of the residuals (σ_e or s_e in the sample). We can estimate this by dividing the cross products matrix by $n - p - 1$ (i.e., sample size minus the number of regression coefficients, including the intercept).

$$\sigma_e = \sqrt{\frac{\mathbf{e}'\mathbf{e}}{N}} = \sqrt{\frac{\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)'\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)}{N}} \tag{12}$$

$$s_e = \sqrt{\frac{\mathbf{e}'\mathbf{e}}{n - p - 1}} = \sqrt{\frac{\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)'\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)}{n - p - 1}} \tag{13}$$

Model Implied Mean and Variance

The model implied means and variances are a direct extension of everything we have already done. As already discussed, the predicted score represents a linear combination of the predictors to create a composite that minimizes the residual variance. Thus, we can use everything from Lab 2 about computing statistical properties of linear combinations. We can obtain the model implied mean and variance with the following matrix expressions.

$$\mathrm{E}\left(Y\right) = \boldsymbol{\mu}_{\mathbf{X}}'\boldsymbol{\beta} \tag{14}$$

$$\mathrm{Var}\left(Y\right) = \boldsymbol{\beta}'\boldsymbol{\Sigma}_{\mathbf{X}}\boldsymbol{\beta} + \sigma_e^2 \tag{15}$$

As a reminder, **X** includes a vector of ones to represent the intercept; thus, the mean vector would also have a one for the mean of that vector of ones. Similarly, we can obtain the proportion of variance explained (i.e, R^2) with the following expression.

$$R^2 = \frac{\boldsymbol{\beta}'\boldsymbol{\Sigma}_{\mathbf{X}}\boldsymbol{\beta}}{\boldsymbol{\beta}'\boldsymbol{\Sigma}_{\mathbf{X}}\boldsymbol{\beta} + \sigma_e^2} \tag{16}$$

2.3 Statistical Inference

Thus far, we have avoided discussing any statistical inference about the regression coefficients. Under the standard assumption of normally distributed residuals, we write this out in the following matrix form.

$$\mathbf{e} \sim \mathcal{N}_n\left(\mathbf{0}, \sigma_e^2\mathbf{I}\right)$$

Note $\mathcal{N}_n\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ is a multivariate normal distribution with n dimensions and a $n \times 1$ mean vector ($\boldsymbol{\mu}$) and $n \times n$ covariance matrix ($\boldsymbol{\Sigma}$). As illustrated in the previous lab, we can alternatively specify this distributional assumption directly on the outcome of our model.

$$\mathbf{y} \sim \mathcal{N}_n\left(\mathbf{X}\boldsymbol{\beta}, \sigma_e^2\mathbf{I}\right)$$

To obtain the standard errors of $\boldsymbol{\beta}$, we apply a known theorem for a multivariate normal distribution.

> *Suppose* $\mathbf{U} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ *and we have a linear transformation,* $\mathbf{V} = \mathbf{c} + \mathbf{D}\mathbf{U}$. *The linear transformation is then* $\mathbf{V} \sim \mathcal{N}\left(\mathbf{c} + \mathbf{D}\boldsymbol{\mu}, \mathbf{D}\boldsymbol{\Sigma}\mathbf{D}'\right)$.

We can apply the above theorem by setting the following

$$\mathbf{U} = \mathbf{y},\ \boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta},\ \boldsymbol{\Sigma} = \sigma_e^2\mathbf{I},\ \mathbf{V} = \boldsymbol{\beta},\ \mathbf{c} = \mathbf{0},\ \text{and}\ \mathbf{D} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'$$

to obtain the following result.

$$\begin{array}{rcccccc} \mathbf{V} & = & \mathbf{c} & + & \mathbf{D} & \cdot & \mathbf{U} \\ \boldsymbol{\beta} & = & \mathbf{0} & + & \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}' & \cdot & \mathbf{y} \end{array}$$

Therefore, it follows:

$$\boldsymbol{\beta} \sim \mathcal{N}_p\left(\left(\mathbf{X}'\mathbf{X}\right)^{-1}\left(\mathbf{X}'\mathbf{X}\right)\boldsymbol{\beta}, \sigma_e^2\left[\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}\right]\mathbf{I}\left[\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}\right]'\right)$$

We can simplify the above distribution to have a mean vector of $\boldsymbol{\beta}$ and covariance matrix of

$$\begin{aligned} \sigma_e^2\left[\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}\right]\mathbf{I}\left[\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}\right]' &= \sigma_e^2\left(\mathbf{X}'\mathbf{X}\right)^{-1}\left(\mathbf{X}'\mathbf{X}\right)\left[\left(\mathbf{X}'\mathbf{X}\right)^{-1}\right]' \\ &= \sigma_e^2\left(\mathbf{X}'\mathbf{X}\right)^{-1} \end{aligned} \tag{17}$$

by using the fact that $\mathbf{X}'\mathbf{X}$ and its inverse are symmetrical, letting us drop the transpose from the expression. Thus, we can obtain the standard errors for the regression coefficients by taking the square root of the main diagonal from $\sigma_e^2\left(\mathbf{X}'\mathbf{X}\right)^{-1}$.

2.4 Building Your Own OLS Regression in R

2.a In R:

Construct a function that produces a solution to OLS regression. The function should map onto the following inputs:

```
function(y, X)
```

where `y` is a column vector and `X` is a design matrix that may or may not include a one for the intercept. The output of the function ought to be a $(p + 2) \times 4$ matrix with the following labels in the column and the empty blanks (—) filled in:

	Estimate	SE	t value	Pr(>|t|)
b0	—	—	—	—
b1	—	—	—	—
⋮	⋮	⋮	⋮	⋮
bp	—	—	—	—
SD(e)	—	NA	NA	NA
R2	—	NA	NA	NA

Hint: To compute the last column you can use the following code in R :

```
2 * pt(abs(value), dfs_residual, lower.tail = FALSE)
```

where `values` is the t value and `dfs_residual` is the degrees of freedom residual (i.e., $n - p$).

Answer:

Using the `mtcars` data set that is included in R , fit the following model using `lm()`:

```
summary(lm(mpg ~ wt + cyl + gear, data = mtcars))
```

Construct the same model using the OLS function you created and compare the results. Include the output from your function in the comments of the code.

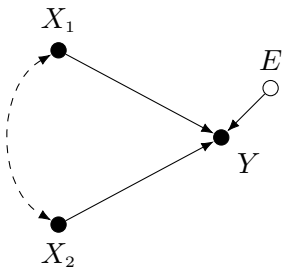

Figure 2: Graph of Data Generating Process for Two Predictors

3 Generating Data for a Multiple Regression

This section will discuss generating data for a multiple regression model. We will begin by discussing a two-predictor model and move to any number of predictors. Consider Figure 2, a DAG for a two-predictor data generating process. This section will discuss generating data for a multiple regression model. We will begin by discussing a two-predictor model and move to any number of predictors. Consider Figure 2, a DAG for a two-predictor data generating process. The easiest way to approach this DAG is in two steps: (1) generating the data for the predictors and (2) generating the data for the outcome. By assuming normal distributions, we obtain the following two steps using the functional notation to trace the steps for generating Y, X_1, and X_2.

$$f\left(X_1, X_2\right) \Rightarrow \quad X_1, X_2 \sim \mathcal{N}_2\left(\boldsymbol{\mu}_{\mathbf{X}}, \boldsymbol{\Sigma}_{\mathbf{X}}\right) \tag{18}$$

$$f\left(Y \mid X_1, X_2\right) \Rightarrow \quad Y \sim \mathcal{N}\left(\beta_0 + \mathbf{X}\boldsymbol{\beta}, \sigma_e\right) \tag{19}$$

3.1 Generating Data for X_1 and X_2 from $f\left(X_1, X_2\right)$

Generating X_1 and X_2 is a straightforward process that we discussed in previous labs. The key here is we will discuss what parameters we must as we begin to approach the problem of a Monte Carlo simulation study. To break down the issue, let us look at the distribution implied by Equation (18) in more detail. Below, we have a single row from a non-augmented data matrix.

$$\mathbf{X}_{i\bullet} = \begin{bmatrix} x_{i1} & x_{i2} \end{bmatrix} \sim \mathcal{N}_2\left(\boldsymbol{\mu}_{\mathbf{X}} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \boldsymbol{\Sigma}_{\mathbf{X}} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}\right) \tag{20}$$

The above distribution contains five unique parameters (two means, two variances, and a correlation). Because the correlation is not shown, let us reparameterize the covariance matrix $\boldsymbol{\Sigma}$ in terms of a correlation matrix.

$$\begin{aligned} \boldsymbol{\Sigma}_{\mathbf{X}} &= \operatorname{diag}\left(\boldsymbol{\sigma}\right) \cdot \mathbf{R}_{\mathbf{X}} \cdot \operatorname{diag}\left(\boldsymbol{\sigma}\right) \\ &= \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \cdot \begin{bmatrix} 1 & \rho_{12} \\ \rho_{12} & 1 \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \end{aligned} \tag{21}$$

Note that $\boldsymbol{\sigma}$ is the vector of standard deviations for X_1 and X_2.

3.a In R:

(1) Look in the directory labeled `scripts` for the file labeled `rmvnorm.R`. In this file, add the `rmvnorm()` function you implemented in Lab 2.

Make sure to fill out the documentation based on the skeleton provided in the comments.

(2) Set the seed to `21389`.

(3) Generate X_1 and X_2 using your `rmvnorm()` function with the following parameters and $n = 100000$.

$$\rho_{12} = 0.3, \boldsymbol{\mu}_X' = \begin{bmatrix} 5 & 10 \end{bmatrix}, \boldsymbol{\sigma}' = \begin{bmatrix} 1 & 2 \end{bmatrix}$$

Make sure to predefine the parameters via a `list()`.

Hint: It will be advantageous to make a new function to translate the parameter list to the inputs expected by `rmvnorm()`.

3.2 Generating Data for Y from $f\left(Y \mid X_1, X_2\right)$

As with any data generation method, we must choose the strength of association among variables. With an outcome with multiple predictors, there are two main methods of making this choice in a standardized metric. The first approach is to fix the regression coefficients to a known weight and then analytically derive the required residual standard deviation that produces a desired R^2. The second approach is to specify the marginal correlation between the predictors and the outcome. We can then analytically derive the required regression coefficients and residual standard deviation. Both methods have their uses and pros and cons; thus, we will discuss both separately.

Method 1: Setting the Total Variance Explained

The key component to this method is to parameterize the total variance explained (i.e., R-squared or R^2). Recall, R^2 is equal to the total variance explained divided by Y's total variance.

$$R^2 = \frac{\text{Var}\left(Y \mid X_1, X_2\right)}{\text{Var}\left(Y\right)} = \frac{\text{Var}\left(Y \mid X_1, X_2\right)}{\text{Var}\left(Y \mid X_1, X_2\right) + \text{Var}\left(E\right)} \tag{22}$$

Therefore, let us first focus on $\text{Var}\left(Y \mid X_1, X_2\right)$ and how to analytically derive $\text{Var}\left(E\right)$.

Determining the variance of Y conditional on X_1 and X_2 is a small extension from previous data generation methods. Previously, we generated an outcome conditional on two uncorrelated variables (i.e., X and U), which is a special case of the more general scenario of correlated predictors. With correlated predictors, we need to account for the correlation between the two predictors with the model implied mean variance of Y.

$$\text{Var}\left(Y \mid X_1, X_2\right) = \beta_1^2\sigma_1^2 + \beta_2^2\sigma_2^2 + 2\beta_1\beta_2\sigma_{12}$$

The third term, $2\beta_1\beta_2\sigma_{12}$, represents the shared variance explained between X_1 and X_2 because both regression coefficients are now partial regression coefficients. Fortunately, the matrix expression

does not need to be modified and can be computed simply as follows.

$$\text{Var}\left(Y \mid \mathbf{X}\right) = \boldsymbol{\beta}'\boldsymbol{\Sigma}_{\text{X}}\boldsymbol{\beta} \tag{23}$$

Now that we have derived the variance explained for Y, we can solve for the unexplained variance by substituting Equation (23) into Equation (22) and solving for Var (E).

$$\text{Var}\left(E\right) = \boldsymbol{\beta}'\boldsymbol{\Sigma}_{\text{X}}\boldsymbol{\beta}\left(\frac{1}{R^2} - 1\right) \tag{24}$$

Therefore, σ_e is equal to the square root of Equation (24), and we must choose a value for R^2 and each regression coefficient. And thus, the con of this method. We must decide the actual values of the regression slopes. This choice isn't arbitrary because they weigh the proportion of how much variance is explained by each predictor. For the two predictor example, if $\beta_1 = \beta_2 = 1$, then each predictor explains an equal amount of unique variance of the outcome. Similarly, if $\beta_1 = -1$ and $\beta_2 = 1$, then each predictor explains the same amount of unique variance, but their correlation are in different directions. Conversely, we could weight the parameters so that one contributes two times the variance as the other (i.e., $\beta_1 = 1$, $\beta_2 = 2$). These choices are ultimately arbitrary and depend very much on what is trying to be accomplished. Still, a good default choice is to have each predictor contribute equal amounts (i.e., all regression coefficients to 1).

With the variance and covariance parameters specified, we must now solve the mean structure of Y. This straightforward procedure requires either setting the mean of Y or a desired β_0. With either set, we can use the analytical solution to the regression intercept to determine the value required.

$$\text{E}\left(Y\right) = \mu_y = \beta_0 + \boldsymbol{\mu}_{\text{X}}'\boldsymbol{\beta} \tag{25}$$

Now that all parameters are either solved for or required to be specified by the user, we can generate the data for the outcome. We can use Equation (19) to generate the outcome Y dependent on the predictors and the derived or chosen parameters.

3.b In R:

(1) Set the seed to `23921`.

(2) Using $\beta_1 = \beta_2 = 1$, $R^2 = 0.6$, $\mu_y = 10$, and the already generated predictors, generate Y based on the appropriate constraints. Make sure to predefine the parameters via a `list()`.

Answer:

Analyze the model with the OLS regression function you created. Create a difference between the population parameters for Y and the results from your generated data. Include the output from the model and differences as a comment in your code.

Method 2: Setting the Marginal Correlations

The previous method has the advantage of specifying the regression coefficients in the population (which will become essential with Monte Carlo simulations) and manipulating the data generation algorithm via the total variance explained parameter. However, we must derive the marginal population correlation between the outcome and predictors (if we desire to know this quantity). The second method is the complete opposite. We specify the marginal correlation matrix for all data

and then derive the required regression coefficients and what the population variance explained is for the outcome. The advantage of this approach is that it allows us to manipulate the correlation between each predictor and outcome directly.

First, let us think about the entire data in terms of a single combined data matrix. Recall from Lab 2 we looked at partitioned data matrices and covariance matrices. We can construct a similar matrix for our outcome Y, predictors X_1 and X_2.

$$\underset{n\times 3}{\mathbf{M}} = \left[\mathbf{y} \mid \mathbf{X}\right] \equiv \left[\begin{array}{c|cc} y_1 & x_{11} & x_{12} \\ \vdots & \vdots & \vdots \\ y_n & x_{n1} & x_{n2} \end{array}\right]$$

Following the discussion in Lab 2, we can obtain the overall covariance matrix for the combined data set and have the following partitions.

$$\underset{3\times 3}{\mathbf{\Sigma_M}} = \left[\begin{array}{c|c} \mathbf{\Sigma_y} & \mathbf{\Sigma_{yX}} \\ \hline \mathbf{\Sigma_{Xy}} & \mathbf{\Sigma_X} \end{array}\right] = \left[\begin{array}{c|cc} \sigma_y^2 & \sigma_{y1} & \sigma_{y2} \\ \hline \sigma_{y1} & \sigma_1^2 & \sigma_{12} \\ \sigma_{y2} & \sigma_{12} & \sigma_2^2 \end{array}\right]$$

Similarly, we can have an overall correlation matrix for the combined data set.

$$\underset{3\times 3}{\mathbf{R_M}} = \left[\begin{array}{c|c} \mathbf{R_y} & \mathbf{R_{yX}} \\ \hline \mathbf{R_{Xy}} & \mathbf{R_X} \end{array}\right] = \left[\begin{array}{c|cc} 1 & \rho_{y1} & \rho_{y2} \\ \hline \rho_{y1} & 1 & \rho_{12} \\ \rho_{y2} & \rho_{12} & 1 \end{array}\right]$$

By specifying the entire mean vector and covariance matrix, we can obtain the solutions to the linear regression only using these sufficient statistics.

With the relationships, variances, and covariances all specified, we can then solve for the regression coefficients and residual variance. Returning to Equation (11), we can rewrite this expression in terms of the partitioned covariance matrix (and it can be analytically shown to be equivalent).

$$\boldsymbol{\beta} = \mathbf{\Sigma}_\mathbf{X}^{-1}\mathbf{\Sigma}_\mathbf{Xy} \tag{26}$$

In addition, we can derive the residual variance based solely on the partitioned covariance matrix.

$$\sigma_e^2 = \mathbf{\Sigma}_\mathbf{y} - \boldsymbol{\beta}'\mathbf{\Sigma}_\mathbf{Xy} \tag{27}$$

The above has two terms, the total variance of Y and the variance explained by X (written differently from Equation 23 but equivalent). Note that any attempt to try and adjust the residual variance to obtain a specific R^2 value will inevitably not produce the correct marginal correlation matrix. However, one can determine the R^2 based on the underlying correlation matrix and using the following computation.

$$R^2 = \mathbf{R}_\mathbf{yX}\mathbf{R}_\mathbf{X}^{-1}\mathbf{R}_\mathbf{Xy} \tag{28}$$

With the variance and covariance parameters specified, the solution to the mean structure for Y follows the same process as the previous method. First, we specify either the intercept value directly or the mean of Y and then use Equation (25) to obtain the appropriate solutions. Now that all parameters are either solved for or required to be specified by the user, we can generate the data for the outcome. However, it might be tempting to draw the entire **M** data matrix from a multivariate normal distribution, I recommend against it. This approach is correct when assuming a multivariate normal distribution; however, if we include nonlinear relationships (e.g., interactions, random effects, quadratics), it is much easier to extend by following Equation (19)—that is, to generate Y conditional on the predictors.

3.c In R:

(1) Set the seed to `123782`.

(2) Using $\rho_{y1} = 0.3$, $\rho_{y2} = -.4$, $\mu_y = 10$, $\sigma_y = 5$, and the already generated predictors, generate a new Y (make sure to call it something different) based on the appropriate constraints. Make sure to predefine the parameters via a new `list()`.

Answer:

Analyze the model with the OLS regression function you created. Create a difference between the population parameters for Y and the results from your generated data. Include the output from the model and differences as a comment in your code.

3.3 Generating for Any Number of Predictors

3.d In R:

(1) Create two functions that combine the predictor data generation with the outcome data generation for methods 1 and 2. Both functions should implement the general matrix expressions given so that they can incorporate any number of predictors. These functions should map onto the following inputs:

```
function(n, p_x, p_y)
```

Hint: Add as attributes to the output the population parameters using

```
attr(object, 'attribute name') <- value
```

(2) Test the Method 1 function using five predictors with $n = 100000$ that are all correlated 0.15 with variances from 1 to 5 and all means 0. Set all regression coefficients to 1, the $R^2 = 0.5$, and $\mu_y = 25$. Set the seed to `6972`.

(3) Test the Method 1 function using five predictors with $n = 100000$ that are all correlated 0.15 with variances from 1 to 5 and all means 0. The correlations with Y ought to be $-.15$, $-.50$, $.15$, 0.30, and 0.20. The mean and variance of Y are $\mu_y = 10$, $\sigma_y = 4$. Set the seed to `1237`

Hint: Create a function to generate the X covariance matrix for you.

Answer:

Analyze the model with the OLS regression function you created. Create a difference between the population parameters for Y and the results from your generated data. Include the output from the model and differences as a comment in your code.