Homework 03

PSTAT 100: Spring 2024 (Instructor: Ethan P. Marzban)

MEMBER 1 (NetID 1) MEMBER 2 (NetID 2)
MEMBER 3 (NetID 3)

! Important Instructions

- This document contains **all** of the problems for Homework 3.
- In Part 1 of this homework, I introduce to you some basics of incorporating equations in QMDs using LaTeX. **There are no steps for you to complete in this part**; rather, you will need to use what is discussed in that part to typeset equations for Part 2.
  - I will also expect you to know LaTeX syntax for ICA02, as there may be a handful of questions pertaining to this.
- Part 2 of this homework is a mini regression project.
- Please write all of your answers in the spaces provided below; for the regression project (Part 2) I expect you to use LaTeX to format your equations.

Part 1: LaTeX in Quarto

For those unaware, LaTeX (often pronounced either like "lay-tech" or "lah-tech") is a typesetting software designed to render mathematical equations. One of the benefits of using QMD (Quarto Markdown Documents) is that they come equipped with LaTeX editors and LaTeX capabilities. In this part of the homework, we will cover some of the basics of using LaTeX to include equations in our final documents.

There are two main types of equations: **inline** equations, like $f(x, y) = x^2 + y^2$, and **display style** equations, like

$$\overline{x}_n := \frac{1}{n}\sum_{i=1}^{n} x_i$$

In LaTeX, inline equations are specified by single dollar signs and display-style equations are specified by double-dollar signs. For instance:

- `$x + y$` will display as $x + y$
- `$$x + y$$` will display as

$$x + y$$

Inside either an inline or display-style equation environment, we can use LaTeX-specific syntax to generate equations.

Important Information

- Standard mathematical symbols display in LaTeX as they would in text: this includes addition (`+`), subtraction (`-`), division (`/`), and multiplication (`*`).
- The symbol for "less than or equal to" ( ) is typeset as `\leq`; the symbol for "greater than or equal to" ( ) is typeset as `\geq`.
- Exponents are generated using the caret (`^`): for example, `$x^2$` displays as x^2
- Subscripts are generated using the underscore (`_`): for example, `$x_2$` displays as x_2

With the basics down, we can also consider typesetting more advanced formulas and symbols:

Important Information

- Sums (using sigma-notation) are generated using the `\sum` command: for example, `$\sum_{x = 0}^{1} x$` displays as $\sum_{x=0}^{1} x$
- Integrals are generated using the `\int` command: for example, `$\int_{a}^{b} f(x) \ dx$` displays as $\int_a^b f(x)\ dx$
- Fractions are generated using the `\frac` command: for example, `$\frac{1}{2}$` displays as $\frac{1}{2}$
- Square-roots are generated using the `\sqrt` command: for example, `$\sqrt{2}$` displays as $\sqrt{2}$
- The symbol π is generated using `$\pi$`; typically, lower case greek letters are generated using a backslash followed by the name of the letter (e.g. `$\beta$` for β).

- Adding a bar on top of a variable can be done using `\overline{}`; e.g. `$\overline{y}$` typesets as $\overline{y}$
- Adding a had on top of a variable can be done using `\hat{}` or `\widehat{}`; e.g. `$\widehat{y}$` typesets as $\widehat{y}$
- The probability symbol can be typeset using `$\mathbb{P}$`; e.g. $\mathbb{P}$.
- The symbol for Expected Value can be typeset using `$\mathbb{E}$`; e.g. $\mathbb{E}$.

Note

In LaTeX, curly braces (`{`, `}`) are used to delineate "chunks" of code/text. For example, if you just write `$x^-2$`, this displays as x^-2. If we want to display x^{-2}, we need to use `$x^{-2}$`.

Note

When writing fractions inside parentheses, it is important to use `\left(` and `\right)` to ensure the size of the parentheses scale with the fractions. For example,

$$(\frac{1}{2})$$

which was generated using `$$ (\frac{1}{2}) $$`, looks a bit worse than

$$\left(\frac{1}{2}\right)$$

which was generated using `$$ \left(\frac{1}{2}\right) $$`.

For example, to typeset the density of the standard normal distribution, we can write

$$\phi(z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

It may also be useful to learn how to make piecewise-defined equations using LaTeX. To typeset piecewise-defined functions, we use the `cases` environment:

$$\begin{cases} a & \text{if condition 1} \\ b & \text{if condition 2} \\ c & \text{if condition 3} \end{cases}$$

was generated using the code

```
$$ \begin{cases}
    a   & \text{if condition 1}   \\
    b   & \text{if condition 2}   \\
    c   & \text{if condition 3}   \\
\end{cases} $$
```

Part 2: Regression Project

Data Background

The dataset we'll explore in this homework was collected by Grete Heinz and Louis J. Peterson at San Jose State University and at the U.S. Naval Postgraduate School in Monterey, California, back in 2003. (Though the dataset is quite old at this point, it is nevertheless a useful dataset to consider for regression purposes.) A full data dictionary can be found at this link. Our primary goal will be to investigate the effects of various predictors on `height` (i.e. we will be treating `height` as our primary response variable.)

Simple Linear Regressions

Question 1

It's always a good idea to get to know the dataset a bit first before modeling. Produce appropriate plots to visualize the distributions of both the `height` variable, as well as two other variables of your choice.

ANSWERS TO QUESTION 1:

Replace this line with your answers; add code chunks and text/equations as necessary.

Question 2

We'll start off simple and consider a few simple linear regressions. Let y_i denote the i^{th} `height` measurement, and let x_i denote the i^{th} weight measurement. Write down the simple linear regression model in this case (**use LaTeX!**), and generate a scatterplot of `height` vs `weight` to visually asses how well you think a linear model might fit.

ANSWERS TO QUESTION 2:

Replace this line with your answers; add code chunks and text/equations as necessary.

Question 3

Fit the model you wrote down in Question 2 using the method of Ordinary Least Squares (OLS). Feel free to use the `lm()` function. Produce a regression table, and interpret the coefficient and p-values in the context of the model (i.e. describe things in terms of the variables `height` and `weight`).

ANSWERS TO QUESTION 3:

Replace this line with your answers; add code chunks and text/equations as necessary.

Question 4

Produce **both** a residuals plot as well as a QQ-plot of the residuals. Comment on whether or not the linear model appears to be fitting well; also comment on whether or not you believe the assumptions of normality and homoskedasticity are met.

ANSWERS TO QUESTION 4:

Replace this line with your answers; add code chunks and text/equations as necessary.

Question 5

Now, let's consider a slightly different regression model. Let `height` denote our response variable, and let `gender` denote our predictor. (Please note: though we now recognize that more than two genders exist, the original experimenters only encoded gender using male and female.) Consider the model

$$y_i = \beta_0 + \beta_1 \mathbb{1}\{x_i = \texttt{male}\} + \varepsilon_i$$

Use the method of OLS to fit this model, produce a regression table, and interpret these results in the context of the model (i.e. in terms of `height` and `gender`).

ANSWERS TO QUESTION 5:

Replace this line with your answers; add code chunks and text/equations as necessary.

Question 6

Note that, in the original dataset, a `gender` value of `male` has been recorded as 1 and a `gender` value of `female` has been recorded as 0. Let's check that `R` still produces the same results, even if we had used more descriptive encodings for the values of `gender`.

Create a copy of your data frame (the one containing your data); in this copy, use `mutate()` and `ifelse()` to convert the values of `gender` to `male` and `female` (instead of `0` and `1`). Then, use `lm()` to re-do the fit you performed in Quesrtion 5 above and check that the regression table output contains the same values as the one you produced in Question 5.

ANSWERS TO QUESTION 6:

Replace this line with your answers; add code chunks and text/equations as necessary.

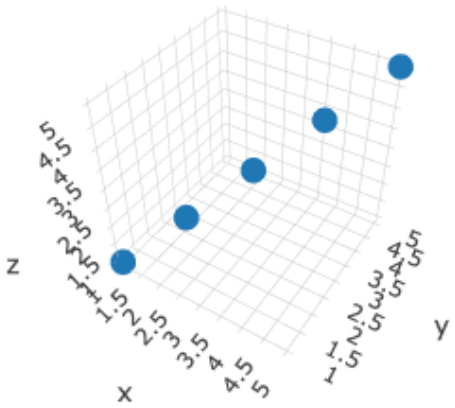
Multiple Linear Regressions

Until now, we've been solely considering simple linear regression models (i.e. models that contain only one explanatory variable). Let's start examining some multiple linear regression models (i.e. models containing 2 or more predictors).

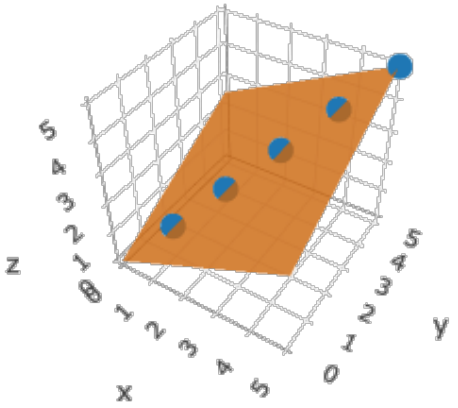
One concept I'd like to mention that I didn't get a chance to talk about in lecture is that of **multicollinearity**. The notion of multicollinearity really only becomes applicable in a multiple regression setting, when we have two or more predictors. Specifically, multicollinearity refers to a setting in which we are including two variables that are highly associated with one another.

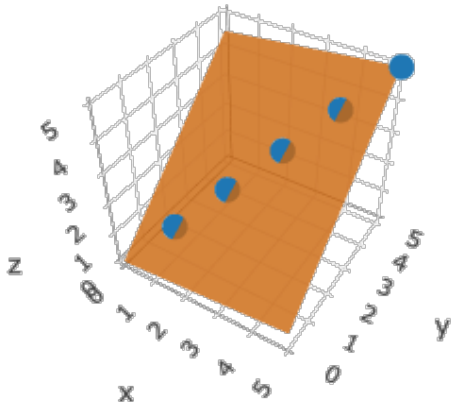
Multicollinearity is actually a problem in regression, and should be avoided. Why? Well, I like to think of this in terms of the graphical visualization of what regression seeks to do. In an SLR, we try to fit a *line* to our data. In MLR (multiple linear regression), we actually try to fit a *hyperplane* to our data. So, if our datapoints are roughly linear in 2 or more dimensions, it becomes increasingly difficult to fit a plane to the data.

Maybe a picture will help. Suppose we have points that fall perfectly along a line in 3 dimensions; i.e. the `x`, `y`, and `z` coordinates are all equal (or scaled/shifted versions of one another):



A MLR would seek to fit a plane to these points. However, notice that both of the planes below "fit" (i.e. contain) the points perfectly:





Indeed, once we have a plane that contains all 5 points we can simply rotate the plane about the axis spanned by these points to obtain an infinite number of planes that contain the data.

This is, loosely speaking, why multicollinearity is a bad idea. If we treated `z` as a response variable and `x` and `y` as explanatory variables, we would not be able to fit an optimal plane using OLS. (Mathematically, multicollinearity is linked with a noninvertible data matrix.)

Of course, perfect multicollinearity is rare in actual datasets. More often than not we have something *close to* perfect multicollinearity. In such a case, the design matrix is not totally noninvertible, but very nearly singular. This leads to values of $(\mathbf{X}^\top\mathbf{X})^{-1}$ that are very large, and hence leads to high variability in our OLS estimates.

All in all, the moral of the story is to avoid multicollinearity!

Question 7

Suppose we want to regress `height` onto both `weight` and `shoulder_girth`. Is this a good idea? Why or why not? (Hint: think about multicollinearity, and how you might be able to detect it in this situation). Regardless of how you answer this, you do **NOT** need to actually perform the fit.

ANSWERS TO QUESTION 7:

Replace this line with your answers; add code chunks and text/equations as necessary.

Question 8

Now, suppose we want to regress `height` onto both `ankle_diam` and `shoulder_girth`. Is this a good idea? Why or why not? (Hint: think about multicollinearity, and how you might be able to detect it in this situation). Regardless of how you answer this, you **SHOULD** perform the fit.

ANSWERS TO QUESTION 8:

Replace this line with your answers; add code chunks and text/equations as necessary.

Question 9

Finally, let's consider a model that contains both a numerical and a categorical response variable. (Such a model is sometimes referred to as an **ANCOVA** (Analysis of Covariance) model). Specifically, regress `height` onto `weight` and `gender`. Produce a regression table, and interpret the results.

ANSWERS TO QUESTION 9:

Replace this line with your answers; add code chunks and text/equations as necessary.

Question 10

Plot `height` against `weight`, and facet by gender (remember this from week 2?). Use `geom_smooth()` to superimpose a linear fit; comment on the fit, and how it compares with your answers to Question 9.

ANSWERS TO QUESTION 10:

Replace this line with your answers; add code chunks and text/equations as necessary.