PDAT 610: Module 5 Homework

Introduction

1. This first problem will have you explore the AmesHousing data set in order to discover why we had to limit our use of data to homes built on or after 1950.

    a. First, let's look at what we would have found if we hadn't looked only at homes built after 1950. Write some code to

        - load the `AmesHousing` data set into R and create the data frame `ames`,
        - create a new column called `Remodeled`, as we did in the lecture notes, that will be true if `Year_Built != Year_Remod_Add`,
        - sample 500 data points from the `ames` data frame, using `set.seed(248)` to make sure we all get the same sample, and
        - run an independent-sample *t*-test to test whether the means of the remodeled and non-remodeled groups are different.

    b. Compare the results of this *t*-test to those shown in the lecture notes. There should be one aspect that's markedly different.

    c. What's going on? Make a plot that helps you explore the data. Then write a brief explanation of why the results differ, depending on whether homes built before 1950 are included in the data set. [Hint: There might be lots of ways to do this, but a scatter plot involving `Year_Remod_Add` and `Year_Built` might work well.]

2. In this problem, we'll look at simulating the results of an independent-sample t-test in order to gain some insight about sample sizes.

    a. Complete the code stub below to write code to generate two sets of random, normally-distributed data, with the sample sizes, means and standard deviations as specified in the code. Then perform an independent-sample *t*-test using your two samples as the two groups to be compared. [Note: we're purposefully **not** going to use `set.seed` here.]

```
n1 <- 50
n2 <- 50
mu1 <- 15
mu2 <- 17
sigma1 <- 8
sigma2 <- 8
```

    b. Suppose your simulated data represents the results of a trial of a new blood pressure medication. Group 1 represents the decrease in blood pressure for a group taking the current "best in class" medication, while Group 2 represents the decrease in blood pressure for a group taking a new competitor's drug. Interpret the results of the test you did in a way that communicates accurately, but without unnecessary complexity. Make sure that you refer to the specifics of this real-world scenario. [Assume $\alpha = .05$ so that you can get a specific Yes/No answer to the test.]

3. This question picks up where the last left off. If you run your code from Question 2 several times (click the green arrow on the code chunk several times), you should notice that sometimes you get a *p*-value that would reject the null hypothesis, and sometimes you don't. Furthermore, remember that there are two ways that a hypothesis test might be in error:

   - Type I Error: Rejecting H_0 when H_0 is really true. Prob(Type I) $= \alpha$.
   - Type II Error: Failing to reject H_0 when H_0 is really false. Prob(Type II) $= \beta$.

   We can control the probability of committing a Type I error by controlling the significance level, α, that we choose for the test. On the other hand, calculating and controlling β is harder. Even calculating β requires an assumption about what we think the *true* alternative really is. Not just $\mu_2 - \mu_1 \neq 0$, but, for example, $\mu_2 - \mu_1 = 2$. With that assumption, it is possible to calculate β. This can be done theoretically, or—as we'll see—through simulation.

   a. By setting `mu1 <- 15` and `mu2 <- 17` in the code above, we've actually made the assumption in this simulation that the true population means are not equal—in fact, they differ by 2. Therefore, to get an estimate of β, all we need to do is to run the simulations and hypothesis test many times, and look at the percentage of times that we *fail to reject* the null hypothesis, even though we know that the null hypotheses is false in our simulation.

      Run your simulation and *t*-test code many times (maybe at least 20 times?), and calculate the percentage of those runs where you fail to reject the null hypothesis. Then fill in the sentence below:

      "Under the assumption that the new drug lowers blood pressure by 2 mmHg more than its competitor, I estimate the the probability of failing to detect that difference is _____."

      [Note: You can either accomplish this task by clicking the green arrow to run you code chunk many times and then recording the results on a piece of paper, or you could write slick code to loop through the simulation many times automatically.]

   b. Suppose you find this probability of Type II error to be too high in the scenario outlined above. One way to decrease it would be to raise your sample size. By experimenting with changing `n1` and `n2`, find the minimum sample sizes necessary to reduce your probability of Type II error to 5%. [You might try $n = 100$, $n = 200$, etc. as a start. **Note:** The *power* of a test is defined to be $1 - \beta$, so we're really finding the sample size that would give a *power* of 95% for this test.]

4. We saw that taking the log transform of `Sale_Price` was one way to try to get control of the variance of residuals in our simple linear regression. Another way to try to improve the model's fit and the model assumptions is to add further important variables to the model. Often bad fit is the result of confounding from lurking variables.

   In this problem, start with the first multiple linear regression model we looked at in the lecture. It's named `fit.original` in the code chunk below. Then do the following:

   a. Write code to plot residuals vs. fit for `fit.original` (try to find the right plot option to print out *only* that graph).

   ```
   # Uncomment after you've loaded the AmesHousing library and done Q1.
   # fit.original <- lm(Sale_Price ~ Year_Built + First_Flr_SF, data=ames)
   ```

   b. Write code to output the R^2-adjusted for `fit.original`.

   c. Think about which variables which variables might also be highly related to `Sale_Price`. Then create a new model using those variables that has a higher R^2-adjusted and seems to satisfy the linearity and constant variance conditions better. Output the plot of fit vs. residuals and the adjusted r-squared for this new model as well. There's no single right answer here—for now you're just playing around to see what you'll find.