Classwork 10: Multicollinearity

In this classwork, we'll work on building a model of the prices of houses. You'll find that the number of bedrooms appears to be very important in the simple regression context, but when you add all the other variables in the dataset, `bedrooms` loses a lot of statistical significance and you may wonder whether it even belongs in the full model at all.

Multicollinearity is the culprit: `bedrooms` is highly correlated with several of the other variables, so when we don't have tons of data and when there isn't much variation in the number of bedrooms different houses have, that slope estimate won't be biased, but it will be measured imprecisely and it will have large standard errors. At the end of this classwork we'll think about some possible fixes.

```
library(tidyverse)
houses <- read_csv("https://raw.githubusercontent.com/cobriant/320data/master/Housing.csv")
```

1. Dplyr questions

1.1 Describe the cheapest house in the dataset; describe the most expensive house in the dataset.

1.2 How many houses in the dataset have one bedroom? How about two, three, etc bedrooms?

1.3 What's the average price for a house with one bedroom versus two, three, etc?

1.4 How many houses are unfurnished?

2. Draw a Scatterplot

First, visualize the relationship between `price` and `bedrooms` with a scatterplot (use `geom_jitter` since bedrooms is discrete). Add a line of best fit.

3. Log-linear plot

Note that when you transform the variable `price` into `log(price)`, a linear model seems like a better fit. You can either use log(price) in the aes() call, or add the layer `scale_y_log10()`. The only difference is the way the y-axis is labeled.

4. Model

Fit the model $log(price) = \beta_0 + \beta_1 bedrooms + u$ and interpret the slope coefficient's estimate and statistical significance. You should find that `bedrooms` seems to be an important factor in house price. The interpretation of a slope coefficient on a log-linear model is this: one extra bedroom can be associated with a (beta1 * 100)% increase in price. We'll talk more about these kinds of transformations next week.

5. Multiple Regression

Here I fit a full model of `log(price)`. Notice in the lm formula, I can use a `.` to indicate I want to include all the other variables in the dataset. You could equivalently use the formula `log(price) ~ area + bedrooms + bathrooms + stories + mainroad + guestroom + basement + hotwaterheating + airconditioning + parking + prefarea + furnishingstatus`

Interpret the slope coefficient on `bedrooms` and its statistical significance. You should find that, compared to the simple regression, when we include the other variables in the dataset, `bedrooms` starts to lose significance.

```
houses %>%
  lm(log(price) ~ ., data = .) %>%
  broom::tidy()
```

6. Multicollinearity

The reason that `bedrooms` starts to lose significance when we add more variables is that `bedrooms` is highly correlated with many of the other explanatory variables. Show that this is true by coercing variables to be numeric (use `mutate` with `if_else` or `case_when`) and then calling `cor` on the resulting dataset to calculate the correlation matrix. What are the 3 variables `bedrooms` is most correlated with?

7. Frisch-Waugh-Lovell Theorem

Multiple regression lets us describe the relationship between two variables, *all others held constant*. We can use the FWL theorem to visualize that relationship, and here's how: Since we want to hold everything else constant, we'll clean `bedrooms` of its correlations with the other explanatory variables by using `mutate` to transform `bedrooms` to be the residual of this regression: `bedrooms ~ area + bathrooms + stories + mainroad + ...`: everything except for price. (Hint: you might want to use the `.` trick in the regression above paired with `select(-price)` to drop the price variable). Do the same thing for `price`: clean `log(price)` of its correlations with all the other explanatory variables except for `bedrooms`. Then plot those `bedroom_residuals` on the x-axis against the `log_price_residuals` on the y-axis. The slope of the line of best fit will be the same as the slope coefficient in the multiple regression from question 2. This lets you see, all other variables held constant, whether or not `bedrooms` seems to have a strong linear relationship with `log(price)`.

```
# houses %>%
#   mutate(
#     bedrooms_resid = __,
#     log_price_resid = __
#   ) %>%
#   ggplot(aes(x = bedrooms_resid, y = log_price_resid)) +
#   geom_point() +
#   geom_smooth(method = lm)
```

8. Conclusion

We've shown that `bedrooms` suffers from multicollinearity because it is highly correlated with other explanatory variables in the regression. As a result, the estimator for its effect on price will be unbiased, but imprecise (standard errors are large and the estimate may be statistically insignificant when in reality it's likely an important variable in the data generating process).

There are a couple of potential fixes for multicollinearity: 1. We can increase the precision of all of the estimates by adding more data to increase the sample size, or by including more explanatory variables to soak up more of the variation in u. 2. Combine the correlated variables into an overall index (example: a price index combines prices of all kinds of consumer goods and services). 3. Drop some of the correlated variables if they have insignificant coefficients. The problem here is that we could be dropping variables which are important factors determining price.

A question for you: Which of these fixes, if any, could be possible? Why?