Lab 11: LA Homes

If you've spent any time on either *zillow.com* or *redfin.com*, you have likely seen the price estimate that they give for any home. This prediction is generated by a regression model that uses covariates on the house - square footage, number of bedrooms, neighborhood, lot size, etc - to estimate its price. The models that they use are non-linear, but you can fairly easily construct a linear model that will produce similar predictions.

That is your big-picture goal for this lab: to fit a multiple linear regression model to a housing data set with the goal of predicting home prices. Your main Python resource will be the `statsmodels` package, which has very good documentation. The data that you'll use to fit your model come from four neighborhoods and cities in the Los Angeles area. It lists all of the homes that were sold in a one month interval. The data set is stored on the course website as a .csv.

`https://raw.githubusercontent.com/andrewpbray/python-for-r-users/master/data/la-homes.csv`

Data Import, Wrangling, and EDA

1. Load the data set into Python. How many observations are there? How many variables?
2. Before you get to building your model, there are several data wrangling tasks that you need to take care of first. Calculate the number of missing values (`NaN`) using the `.isnull()` data frame method combined with the `.sum()` method. If any variables are entire null, drop them from the data frame.
3. How many different unique values are there in the `type` column? How many observations of each one? Filter the data set so that it only includes `SFR` (single family residences).
4. Plot the distribution of the number of bedrooms.

5. Plot the distribution of `garage`. What type of data is this being stored as? Coerce it into a Series of type integer.
6. Create a scatterplot of the relationship between square footage and home price. How would you describe this relationship? For this plot, you can either use `seaborn` or experiment with `plotly`.

Modelling Take I

7. Use `statsmodels` to fit a simple linear regression model that predicts price as a function of square footage. Add the regression line to your scatterplot. Does your model do a good job of capturing the mean function of the data (does it go through the center of the cloud of points)?
8. Print the summary table of the regression model. How many observations was this model fit to? Compare this to the size of the data set - what does this suggest about how this model handles `NaN` values that might appear in certain columns?
9. Create a plot of the residuals against `x`. Does it appear that the residuals have equal variance or does the variance appear to be a function of the `x` variable?
10. Create a plot of the Cook's Distance, a measure of how influential a single observation is to the coefficient estimates. Which homes does it flag as the most influential?

Modelling Take II

11. Remove those most influential observations and make one other structural change to your model: create two columns that are the natural log of square footage and price. Refit your model to this smaller data set and using these new variables. How do the residual plot and the influence plot change?
12. Augment this model by adding a categorical predictor: city. Based on their estimated coefficients, list the cities from most to least expensive (controlling for the size of the house). Are these coefficients found to be statistically significantly different than zero.
13. Add `bed` to your model. What is the sign of its estimated coefficient? Speculate as to why this structure might exist using your knowledge of the way that people value homes.
14. Which is the most undervalued house in this data set according to your model?
15. Use the `.predict()` method to estimate the price of a 1550 square foot house in Long Beach with 3 bedrooms (see the documentation: https://www.statsmodels.org/stable/examples/notebooks/generated/predict.html). Spot check this by hand by plugging those x values into your regression equation using the coefficient estimates.