Corey Cherrington

(she/her)

LIS 511

June 3, 2021

LIS 511 - Final Project

Introduction

For my final project in LIS 511, I chose to analyze the "Kaggle King County Housing" data set (Option 1). I picked this data set because I am interested in the real estate market and was curious about what I could glean from the data set. Generally, the data centers on details about homes in King County that, in my opinion, might be relevant to potential home buyers or real estate agents. Specifically, when doing a quick look at the top five rows of the columns in this dataset via the `.head(5)` command in Jupyter Notebook, I found that the relevant columns in this data set include the following:

- unique ids for all homes listed in the data set,
- a date of the sale of the house,
- the price of the house when it was sold,
- number of bedrooms,
- number of bathrooms,
- number of square feet of the home,
- number of square feet of the property that the home was built on,
- how many floors the home has,
- "whether the apartment was overlooking the waterfront or not" (Murillo, 2016),
- the quality of the view the home has,
- year built,
- "the year of the house's last renovation" (Murillo, 2016),
- zip code of the house,
- latitude and longitude of the home,
- and other details.

I was able to double-check my understanding of the columns by viewing the web page "King County Home Sales: Analysis and the limitations of a multiple regression model" (Murillo, 2016). I was also able to fill in some of my gaps in understanding of the columns by consulting the aforementioned source. Overall, each row in the data set appears to represent a single home that was sold in King County.

Part 1 – The Python Script & Explanations

```
# Setting up the csv and the imports needed for the project
```

```
# Referenced this page for more information about the data set: https://rstudio-
pubs-static.s3.amazonaws.com/155304_cc51f448116744069664b35e7762999f.html
import pandas as pd
import numpy as np

kc_house_data = pd.read_csv("kc_house_data.csv") # Please note: I changed the nam
e of the csv to eliminate the need for spaces in the document's title
kc_house_data.head(5) # using this command to view the column names before diving
 into the questions

# Begin work for each of the questions here:
# For all of the problems, I referenced the following: https://docs.google.com/do
cument/d/1gXe5EihuuXvxtaVqa1DNLbcMdEfODUVWn2K2H8rbrbQ/edit
# https://docs.google.com/document/d/15EII2bg-
ePbPaQe8dlMJhJZeUJ6JIRIlKwv_FJq5zMM/edit
# https://docs.google.com/document/d/1ybiO57YCCFe1IikjtqioffPScYWUXcJEUUqhgQRnLsw
/edit

#Question 1: What is the mean home price in King County?
mean_price = kc_house_data["price"].mean()
print(mean_price)
```

Question 1 Explanation: The correct mean calculated with the .mean() function is: 540088.1417665294. To find this value, I made a variable for mean_price, which would store the calculated mean price of all the houses in the data set. Then, to display this variable, I used the print() command.

```
# Question 2: What is the total number of bathrooms in the zip code 98178?
specific_zip = kc_house_data[kc_house_data["zipcode"] == 98178]
specific_zip["bathrooms"].sum()
```

Question 2 Explanation: As with the mean calculation, we need to set a variable for a specific category, in this case the zip code of 98178. We have the program search through our data set for this zip code and store this in the specific_zip variable. Then, using this limitation on our data set, we can proceed to add up all of the total bathrooms in the zip code with the .sum() function. Using this code, the correct answer comes out to 453.75 total bathrooms in the zip code 98178.

```
# Question 3: What was the zip code with the most expensive home?
max_price = kc_house_data["price"].max()
zip_most_expensive_house = kc_house_data[kc_house_data["price"] == max_price]
zip_most_expensive_house["zipcode"]
```

Question 3 Explanation: To find the answer to this question, we must first find the home in the data set that has the max "price." To find this, I set a variable called max_price and found the

price of the most costly home with the .max() function. After finding this specific price, we have to figure out which home this price applies to, which is why I set the variable zip_most_expensive_house. Because I had already found the max price beforehand, I was able to use my max_price variable to locate the relevant row in the dataset. Next, because we are just looking for the zip code, it is necessary to write the code to reveal only the zip code of the most expensive home found with the preceding code. The correct answer comes out to be the zip code 98102.

```
# Question 4: What is the year built for the home with id 2414600126?
specific_id = kc_house_data[kc_house_data["id"] == 2414600126]
specific_id["yr_built"]
```

Question 4 Explanation: Because each house has a unique id, there should only be one home with the requested id. To find this house, we can search for our specified "id" in the dataset, here called kc_house_data. After we locate this house using this method, we can reveal the yr_built of that house, which comes out to be 1960.

```
#Question 5: What is the correlation between sqft_living and price?
# referenced this for info on correlation coefficients: https://magoosh.com/statistics/pearson-correlation-coefficient/
# also referenced: https://www.westga.edu/academics/research/vrc/assets/docs/scatterplots_and_correlation_notes.pdf
kc_house_data[['sqft_living', 'price']].corr()
```

Question 5 Explanation: To find the correlation between sqft_living and price, we can use these column labels to find the relevant data in the data set and then we can use .corr() to find the correlation between these two entities. When we calculate these values, we get the following chart from our program:

	sqft_living	price
sqft_living	1.000000	0.702035
price	0.702035	1.000000

Figure 1 Table generated by my code written in Jupyter Notebook

According to the sources cited in my code for Question 5, the correlation qualifies as “strong” (see links above).

My Two Original Questions for the Data Set

```
# My original Question (1): Which houses have a view of 1 or
more, sqft_living greater than or equal to 1300, >= 3 bathrooms, and are on the w
aterfront?
# Find out the answer to the above question and create a scatter plot with the yr
_built and zipcode as the x and y axes respectively.
kc_view = kc_house_data[kc_house_data["view"] >= 1]
kc_sqft = kc_view[kc_view["sqft_living"] >= 1300]
kc_br = kc_sqft[kc_sqft["bathrooms"] >= 3]
kc_wf = kc_br[kc_br["waterfront"] == 1]

kc_wf.plot.scatter(x = "yr_built", y = "zipcode")
```

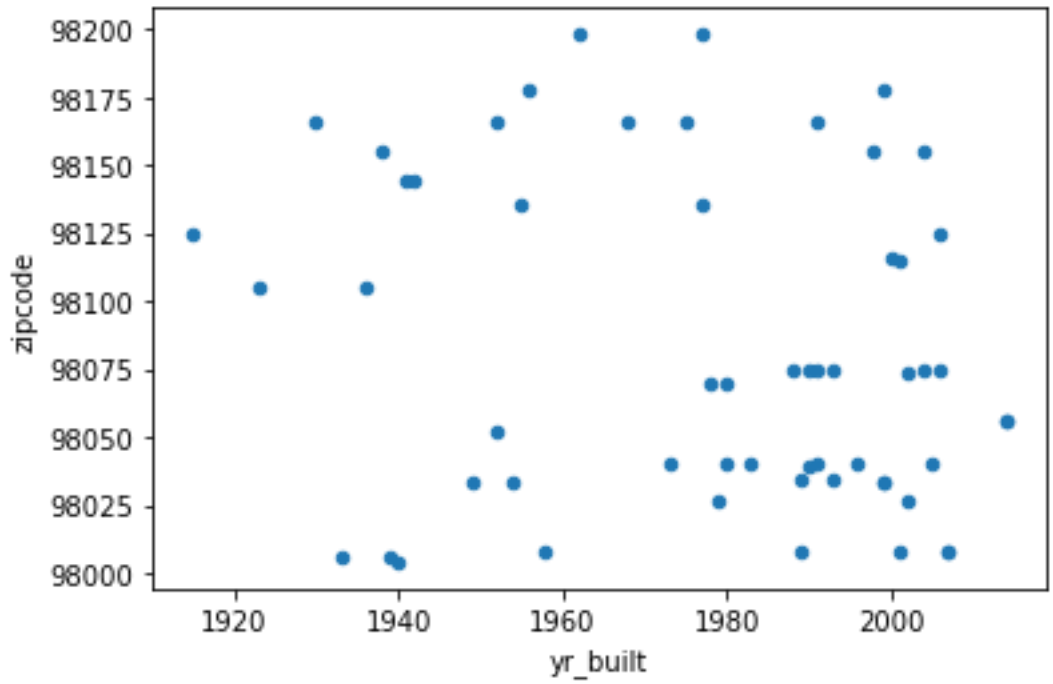

Figure 2 The result for the code I wrote in response to my "Original Question (1)"

My Original Question (1) Explanation: There were a lot of criteria to search for in this question, so my approach was to take each parameter one at a time and progressively transfer them from one variable to the next. I started out looking for houses with a view score of 1 or greater and stored this in the variable kc_view. Moving on from there, I used my kc_view variable to search through the data I determined had a view of 1 or better. In this search, I looked for houses with a sqft_living of 1300 or greater and stored this in kc_sqft. Next, I took this data and searched through it for houses having 3 or more bathrooms and stored this in the variable kc_br. From there, I searched through this data and found the houses that had a waterfront value of 1 (assuming this basically means waterfront = yes based on what I learned in LIS 543 – Relational Database Management Systems). Then, we can use the final variable in

this sequence, kc_wf to make a scatter plot of the zip code and year built of each of the homes that meet all of the aforementioned conditions.

```
# My original Question (2): It looks like there may
be an interesting trend in the above graph in the zipcode 98075. Is there a plot-
able relationship between the sqft_lot and sqft_living of the houses that meet th
e criteria from "Original Question (1)?"

# Find out on a scatter plot where each of the houses that meet the above criteri
a from "My original Question (1)" AND are in the zipcode 98075 are in terms of sq
ft_living (x-axis) and sqft_lot (y-axis).

# Posting in the code from the previous variable definitions in (1):
kc_view = kc_house_data[kc_house_data["view"] >= 1]
kc_sqft = kc_view[kc_view["sqft_living"] >= 1300]
kc_br = kc_sqft[kc_sqft["bathrooms"] >= 3]
kc_wf = kc_br[kc_br["waterfront"] == 1]

# Finally, the code to figure out the answer to my 2nd question:
kc_zip3 = kc_wf[kc_wf['zipcode'] == 98075]
kc_zip3.plot.scatter(x = "sqft_living", y = "sqft_lot")
```

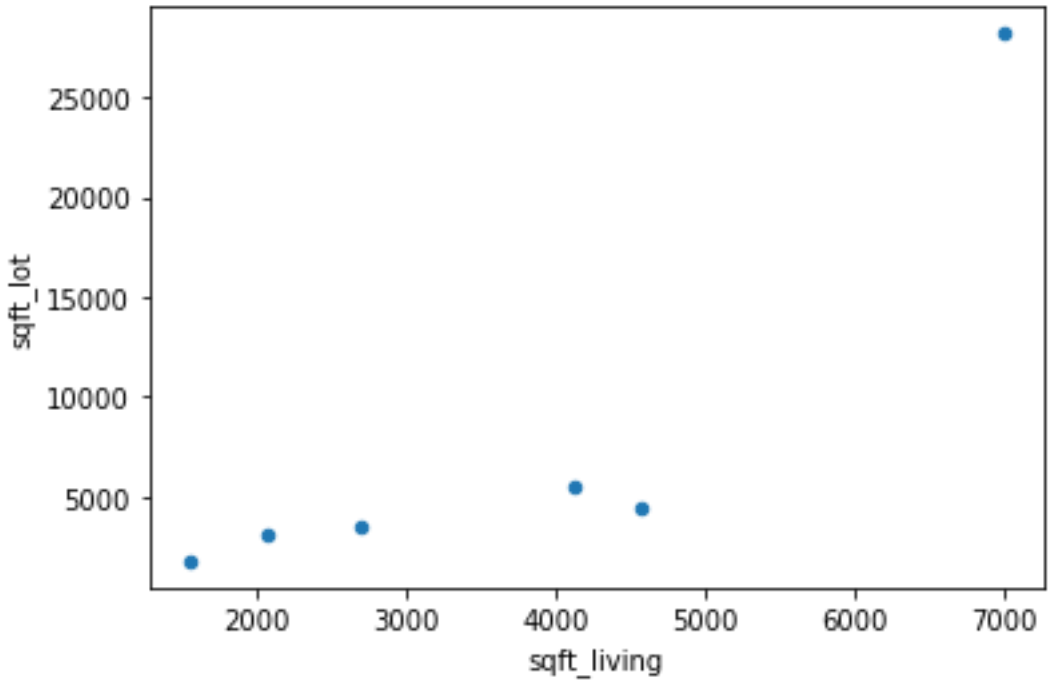

Figure 3 The result of the code written above - a scatter plot.

```
# For a point of Comparison, let's plot the house data for all of the houses list
ed in the data set for the zip code 98075.
kc_zip2 = kc_house_data[kc_house_data["zipcode"] == 98075]
kc_zip2.plot.scatter(x = "sqft_living", y = "sqft_lot")
```

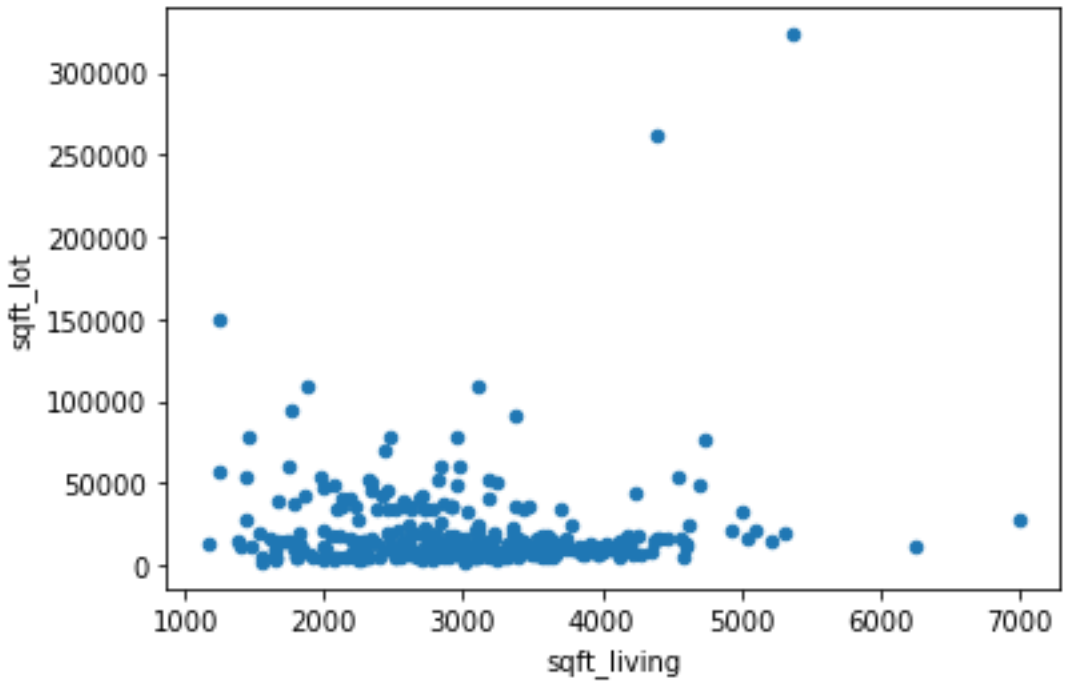

Figure 4 As mentioned above, this scatter plot was generated as a point of comparison for the plot in the preceding figure.

My Original Question (2) Explanation:

For the first scatter plot, we needed to carry over the variables made for the previous question, so this was the first step for "Original Question (2)." From there, I was able to create a variable, kc_zip3, that could find all of the houses that meet the aforementioned conditions and that also are in the zip code of 98075. Because I was curious about where each of these houses lies in terms of square feet of the living space and square feet of the lot that the house is on, I used these as my x and y axes for my scatter plot. This allowed me to dive deeper into the data I found in my first original question to create a visual representing the relationship between sqft_living and sqft_lot for each house meeting all of my stated conditions. Then, as a point of comparison, I produced a scatter plot displaying a similar relationship, but with all of the houses that had the zip code of 98075.

Conclusion

To summarize all my findings from the script I wrote for this project, I discovered the following:

- The mean home price in King County is 540,088.1417665294 (this can be rounded to $540, 088.14)

- The total number of bathrooms in the zip code 98178 is 453.75
- The zip code of the most expensive home in King County is 98102
- The home with the unique id of 2414600126 was built in the year 1960
- The correlation between sqft_living and price is strong (see table above and sources referenced in my comments written for question #5)
- From the scatter plot created for my first original question, it seems that many houses with a view score of 1 or more, sqft_living greater than or equal to 1300, >= 3 bathrooms, and that are on the waterfront were built after 1980, with an interesting relationship occurring between the zip code 98075 and year the house was built.
- Using the conditions set in my first original question, I added the condition that the house also had to be in the zip code 98075. Then, I wanted to make a scatter plot revealing the relationship that each of the houses meting all of my stated conditions had between sqft_lot and sqft_living. This finding revealed what looked like an outlier: only one of these houses had 7000 square feet of living space or more and was on a lot of 25,000 square feet or more. All other houses had 5000 square feet of living space or less and were built on lots of 10,000 square feet or less. For a point of comparison, I presented my second plot for my 2nd original question. The same general trend occurred in these houses, with the number of square feet of living space and square feet of the lot increasing to accommodate more homes on the scatter plot.

I am not an expert on statistical analysis, but the information I was able to discover by using Python code in Jupyter Notebook was clear and interesting. Overall, there seems to be many other possible avenues for research with the "Kaggle King County Housing" data set.

Bibliography

Murillo, J. (2016). "King County Home Sales: Analysis and the limitations of a multiple regression model." https://rstudio-pubs-static.s3.amazonaws.com/155304_cc51f448116744069664b35e7762999f.html.