[redacted]

Question 1: What data would you exclude from analysis for being unreliable or potentially a block instead of an actual booking?

When determining whether an unavailable booked listing is false, the columns of data I would exclude include all but price, scraping_date, date, and availability -- for more advanced analysis we could probably take into account the cleaning fee. After viewing Airbnb's website and trying to view the price of a listing on a booked date, it isn't available. So, it leads me to think that any prices extracted from a date that is unavailable are unreliable.

Scraped dates, price, and date of the listing are the columns that would most likely lead to greater insight into whether or not a user's listing is a "block." By block, I'm defining it as a user is intentionally setting the listing to unavailable but no one paid for the stay (revenue = 0, availability = 0). None of the columns give us an indicator on generated revenue.

Question 2: What is a good approach to estimate occupancy and revenue per unit?

Occupancy of a unit and revenue generated per unit are dependent on multiple features of the provided dataset. There are numerous approaches (ranging from simple to more complex) to estimate these numbers. Occupancy and revenue per unit are continuous values and if we are aiming to predict them, we need to define a metric of evaluating this prediction. In this case, the mean-squared error term of fitting a linear equation to these target variables is a good start. In the case of our problem and the provided datasets, we could do this in a few ways:

To predict occupancy, we could:

- build a linear regression model using the bathroom, bedroom, **price**, is_superhost, city (after one-hot encoding categorical values), has_pool, and the cleaning_fee feature columns that shows when a property was booked.
- We could then train the model to correlate price of a listing to capacity, and if the listing was booked (available = 0), we can deduce the occupancy.

To predict revenue per unit, we could:

- build a linear regression model using the bathroom, bedroom, **capacity**, is_superhost, city (after one-hot encoding categorical values), has_pool, and the cleaning_fee as input data feature columns on training data that shows when a property was booked.

[redacted]

Columns that are less necessary for predicting occupancy and revenue per unit are: (i) listing (URL to the airbnb posting), (ii) mapped_location, (iii) name (posting name of the listing), host_name.

The following columns are more important:

- Scraping_id: used to identify unique samples.
- City: after processing the categories and removing noisy data, city is a strong proxy for location and can be useful in determining price (San Francisco rent is likely higher than Orlando, Florida).
- Lon: Although it is a float32 value, we could normalize the values and set the means of this column to 0. That way, we can see if there are particularly values of Longitude that yield larger variances in price.
- Lat: Although it is a float32 value, we could normalize the values and set the means of this column to 0. That way, we can see if there are particularly values of Latitude that yield larger variances in price.
- Capacity: great way to determine price as it is likely that larger capacity is associated with maybe square footage.
- Bathrooms: self-explanatory; this is a common metric to determine the value of a home.
- Bedrooms: self-explanatory; this is a common metric to determine the value of a home.
- Has_pool: categorical variable that could determine a change in price.
- Cleaning_fee: it is likely that higher cleaning fees are associated with larger homes and/or larger capacity buildings.

To quantify these results, I would train a linear regression model with the cleaned data and post their impact on the model’s slope (i.e., how a change in a given feature column impacts the price).

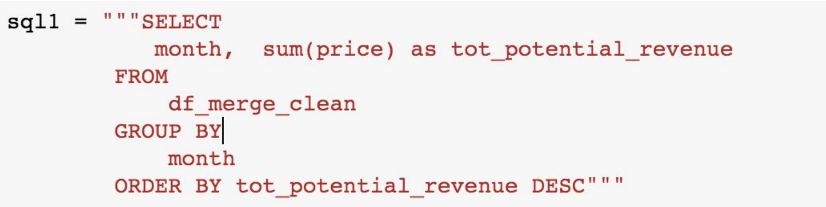
All real booked listings over the period of time provide insight into the total amount of revenue generated.

Question 3: In which month do properties appear to generate more revenue? April or May?

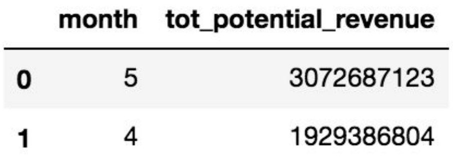
May appears to generate more revenue than April. I’m intentionally not filtering by whether or not a listing is listed as unavailable. Revenue isn’t defined in the columns, so “appearing” to generate revenue could mean a few things. I listed it as the total potential revenue, assuming all listings were booked, the properties could generate.

SQL Query:

[redacted]



```
sql1 = """SELECT
           month,  sum(price) as tot_potential_revenue
        FROM
            df_merge_clean
        GROUP BY
            month
        ORDER BY tot_potential_revenue DESC"""
```



	month	tot_potential_revenue
0	5	3072687123
1	4	1929386804

If we're defining "appearing to generate more revenue" as the total booked listings of all the properties (meaning, aggregating across the properties when availability == 0), I'd add the WHERE clause to filter for the results.

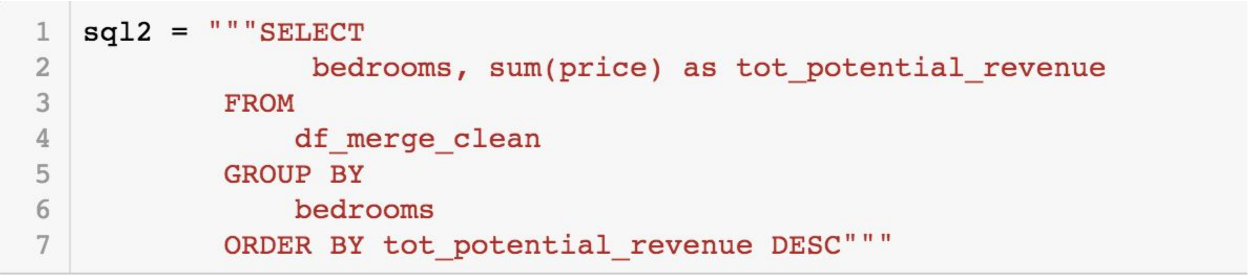
Question 4: How much more revenue do places with 3 bedrooms make vs. places with 2 bedrooms?

Assumptions:

- This revenue is calculated across all dates, and scraped dates, for all listings with 3 or 2 bedrooms.
- I did not filter by whether or not they are available for booking. I didn't do this because if we want to see the impact of having 3 bedrooms versus places with 2 bedrooms, it is important to consider the price even if the listing is not available. For example, I might not be able to book a place on a given day, but that doesn't mean that I shouldn't consider the potential revenue that it could make. Even if it's booked, we would want to use its data to reason about the potential revenue of the properties with 3 or 2-bedrooms.

SQL Query:

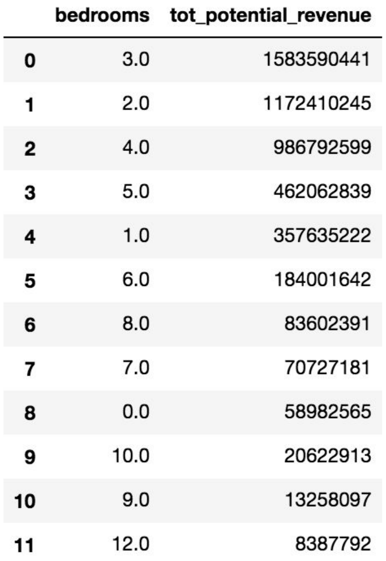
[redacted]



```
sql2 = """SELECT
             bedrooms, sum(price) as tot_potential_revenue
        FROM
            df_merge_clean
        GROUP BY
            bedrooms
        ORDER BY tot_potential_revenue DESC"""
```

Listings with three bedrooms, in aggregate, across the two months of scrapped data, have the potential to generate more than **$411,180,196** in revenue as compared to listings with two bedrooms, holding all other variables constant.

Disclaimer: I'm adding potential revenue from each day of web scraping. For any given day of scraping, there is an associate price per night for 3-bedroom and 2-bedroom listings for the time period between 4/1/2018 - 05/31/2018. Therefore, I wanted to remove the variability of the day of scraping and aggregate across all the listing days. That way, we can say that on average over all the days we scraped data, 3-bedroom apartments generated more potential revenue than 2-bedroom apartments over this two-month period.



	bedrooms	tot_potential_revenue
0	3.0	1583590441
1	2.0	1172410245
2	4.0	986792599
3	5.0	462062839
4	1.0	357635222
5	6.0	184001642
6	8.0	83602391
7	7.0	70727181
8	0.0	58982565
9	10.0	20622913
10	9.0	13258097
11	12.0	8387792

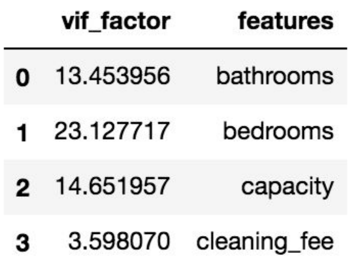
Question 5: What are any other interesting insights you may have found?

I explored the following questions:

[redacted]

Which features exhibit the highest degree of multicollinearity?

Multicollinearity can have a negative impact on any machine learning models we develop. The expected relationship between features and our target variable, price, may not hold when there are many features that are related to each other. This might lead to hypothesis testing results to be unreliable and coefficient estimates from a linear model can be less stable. A good rule of thumb is that features above 10 tend to exhibit a higher degree of multicollinearity [link]. The variance inflation factor for each feature is shown below:

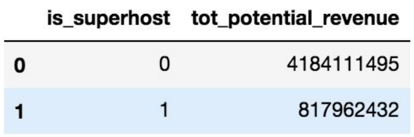


	vif_factor	features
0	13.453956	bathrooms
1	23.127717	bedrooms
2	14.651957	capacity
3	3.598070	cleaning_fee

From initial inspection, the variance inflation factor for bathrooms, bedrooms, and capacity suggest a strong case of interdependence. If we were to build a price prediction algorithm, I would suggest dropping bedrooms as one of the features to obtain better performance (measured by R2 value).

How much total potential revenue do non-super hosts and superhosts of all listings have?

From these two months of data, it looks like there is more potential revenue from non_superhosts. This can be for a number of reasons, and the most likely reason is that there are more non_superhosts in the listings dataset.

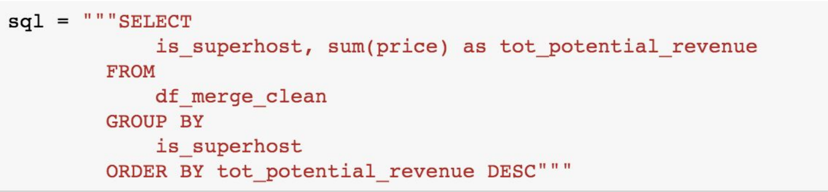


	is_superhost	tot_potential_revenue
0	0	4184111495
1	1	817962432

We have 4,236,395 superhosts and 12,063,780 non_superhosts. So, to determine which type of hosts generates the highest number of potential_revenue normalized by the number of listings over this two month period, we divide and get $346/host for non_superhosts and $193/host for superhosts.

SQL query:

[redacted]

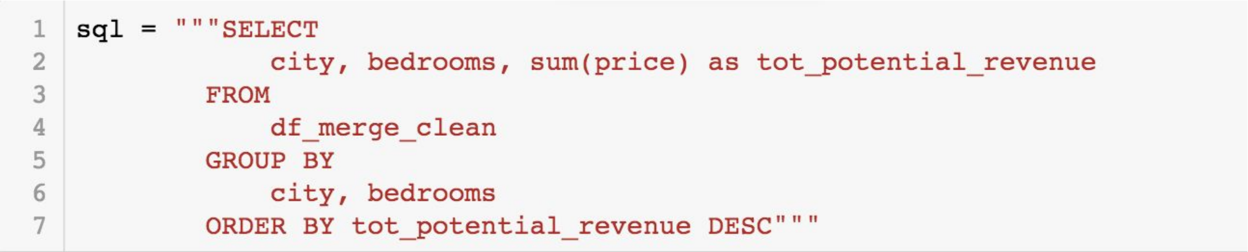


```
sql = """SELECT
            is_superhost, sum(price) as tot_potential_revenue
        FROM
            df_merge_clean
        GROUP BY
            is_superhost
        ORDER BY tot_potential_revenue DESC"""
```

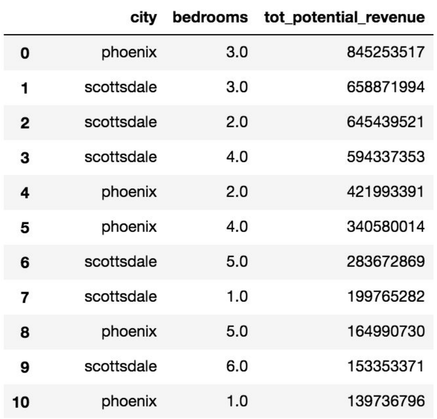
What are the most valuable cities for different number of listings for bedrooms and bathrooms?

In aggregate, including all listings over all scraped dates, **Scottsdale and Phoenix** are the two cities with the highest potential revenue for listings with 1, 2, 3, 4, 5, or 6 bedrooms (by potential revenue, this means I did not add a conditional statement to aggregate across properties that were booked).

SQL Query:



```
sql = """SELECT
            city, bedrooms, sum(price) as tot_potential_revenue
        FROM
            df_merge_clean
        GROUP BY
            city, bedrooms
        ORDER BY tot_potential_revenue DESC"""
```



	city	bedrooms	tot_potential_revenue
0	phoenix	3.0	845253517
1	scottsdale	3.0	658871994
2	scottsdale	2.0	645439521
3	scottsdale	4.0	594337353
4	phoenix	2.0	421993391
5	phoenix	4.0	340580014
6	scottsdale	5.0	283672869
7	scottsdale	1.0	199765282
8	phoenix	5.0	164990730
9	scottsdale	6.0	153353371
10	phoenix	1.0	139736796

The least valuable city in terms of potential revenue is Mesa.

What are some additional sources of data that could be scraped for the pricing algorithm?

Maybe an additional option for scraping short-term rental data would include from websites like Zillow or Sonder. Though Zillow gives monthly rental prices, maybe calculating the daily rate and comparing the margin of difference in pricing between Sonder & Airbnb listings could give

[redacted]

the team a price average across more than just one website. This also reduces the reliance on Airbnb pricing. There’s also AirDNA https://www.airdna.co/.