Louie Balderrama
Springboard Data Science Career Track, January 2019 cohort

Capstone Project I

Problem Statement: Predicting the nightly rate of Airbnb listings in Austin, TX

Background

Airbnb has become the biggest disrupter of the hospitality and tourism industry. It offers a global platform for private individuals willing to share a space to transact with would-be guests at a nightly rate pre-set by the host. The study will take a look at the different aspects of Airbnb listings and see what combination of features will most accurately predict the nightly rate of a given listing. The study will focus on the city of Austin, the tech-capital of Texas, and will make use of the features already in the Airbnb listings dataset. As a supplement, the study will also incorporate data on the crimes reported within the areas covered by the Airbnb dataset.

Impact

A reliable model can benefit both hosts and guests because the model would be able to provide a standard fair price that is appropriate for what the host can offer. Hosts would have a good rule of thumb in pricing a nightly stay given the nature of the space (lodging type, number of rooms, amenities, etc.), the location (number of crimes reported in the area, serious crimes prevalent in the area), the nature of the booking offered (cancellation policy, fees for extra guests), and the details about the hosts themselves ("superhost" status, types of verifications provided). A host would therefore minimize opportunity cost by not missing out on bookings from guests who would have otherwise been offered a fair price. On the same token, guests would have a reliable indicator of what a fair deal would look like. Even the industry itself could see a positive impact from an optimized, competitive host-guest marketplace. The limitation of the study, however, other than a limited geographical scope, is that the surge in demand from popular events and its effect on price will not be covered. Conventional rates are assumed.

Datasets

a. Airbnb Listings

The dataset comes from **Inside Airbnb**, an independent, third-party website that has its own set of open-source tools that encourage users to explore the negative and positive impacts of Airbnb. They scrape Airbnb listings from different metropolitan areas around the world, strip personal identification (like the full name of hosts and guests), and share the raw data on the site. The dataset used in this study is the "listings.csv" for Austin, TX as compiled on November 2018. There are 11,919 entries which represent all the Airbnb listings in the city at the time. The dataset is both rich with information and quite raw and messy, having a total of 96 columns. Some of the useful details include the listing price, the listing type (private bedroom, shared bedroom, or entire home), the location of the listing, review score of the listing (aggregated guest ratings), review score of the host, etc.

b. Austin, TX Crime

The crime dataset was retrieved from the City of Austin's **official data portal**. This dataset details all the reported crimes that occurred in Austin during 2018 – the timeframe that coincides with the Airbnb dataset. Though there are a total of 102,663 entries, only around 36,000 crimes have real values (i.e. non NaNs) under their "Category Description" series. This classifies major crimes according to the following seven categories recommended by the FBI's Uniform Crime Reporting program: Theft, Auto Theft, Burglary, Robbery, Rape, Aggravated Assault, and Murder. This categorization will then help in getting a sense of what major crimes occur in which areas, while the dataset as a whole will provide the aggregate statistics of all crimes (both petty and serious) in Austin.

c. Zip Codes GeoJSON

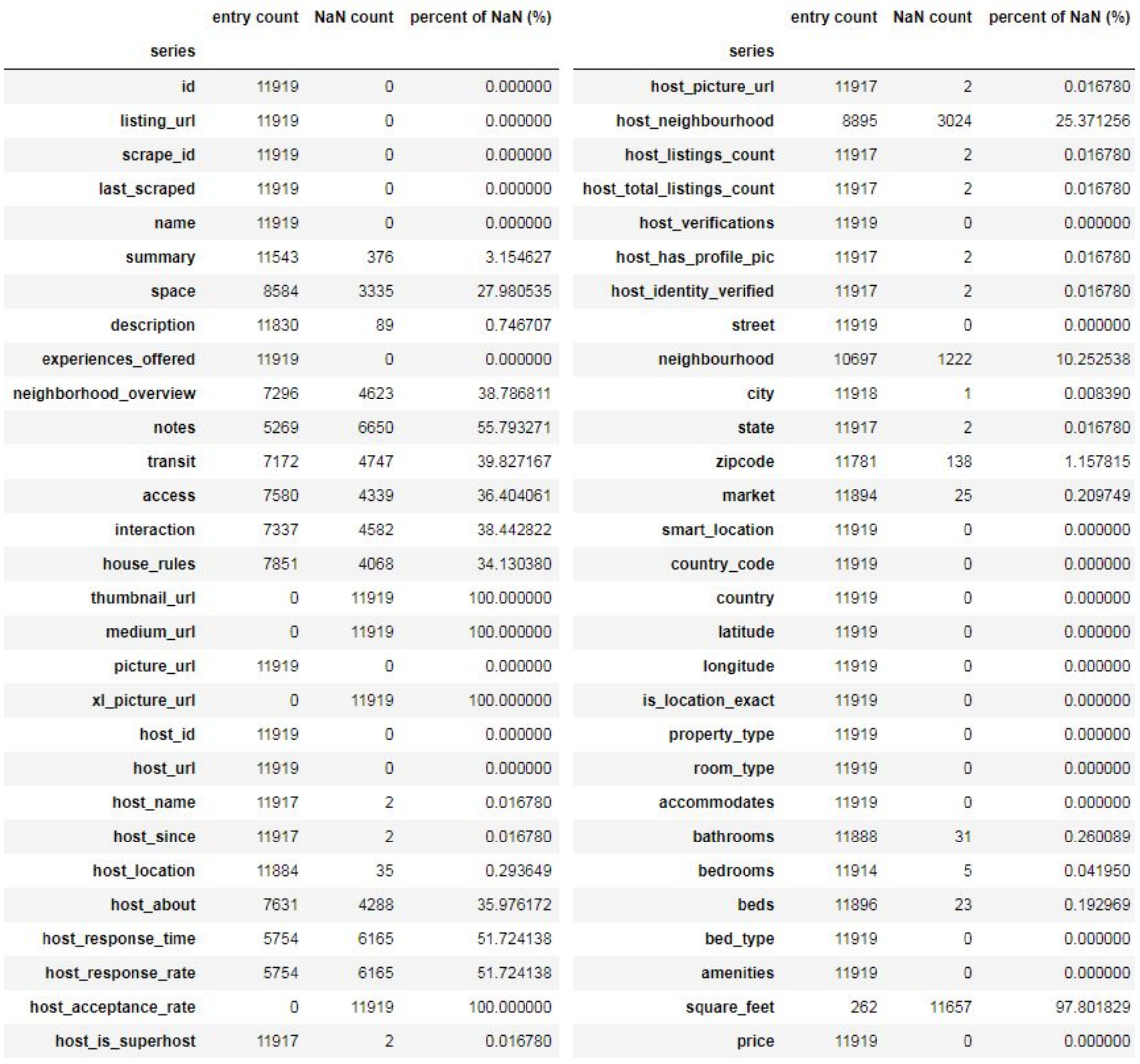
This geospatial JSON file, also retrieved from Austin's official data portal, contains the boundary coordinates of every zip code within the city. When combined with the findings of the other datasets, this file will aid in the visualization of trends by location, including where the most expensive Airbnb listings are and where crimes are reported the most.

Data Wrangling

Sub Dataframe: Listings

The Airbnb dataset in its original form has 96 columns. However, two of these are not directly from Airbnb and were appended by the poster after scraping. These are the neighbourhood_cleansed and neighbourhood_group_cleansed features which are the *cleaned* derivatives of existing columns. These two will be dropped to maintain fidelity with the original dataset.

The tables below describe the NaN count and the percentage of NaN values against the total number of entries. A dataframe built from a dictionary was used to make the table easy-to-read. The snippet is below:



series	entry count	NaN count	percent of NaN (%)
id	11919	0	0.000000
listing_url	11919	0	0.000000
scrape_id	11919	0	0.000000
last_scraped	11919	0	0.000000
name	11919	0	0.000000
summary	11543	376	3.154627
space	8584	3335	27.980535
description	11830	89	0.746707
experiences_offered	11919	0	0.000000
neighborhood_overview	7296	4623	38.786811
notes	5269	6650	55.793271
transit	7172	4747	39.827167
access	7580	4339	36.404061
interaction	7337	4582	38.442822
house_rules	7851	4068	34.130380
thumbnail_url	0	11919	100.000000
medium_url	0	11919	100.000000
picture_url	11919	0	0.000000
xl_picture_url	0	11919	100.000000
host_id	11919	0	0.000000
host_url	11919	0	0.000000
host_name	11917	2	0.016780
host_since	11917	2	0.016780
host_location	11884	35	0.293649
host_about	7631	4288	35.976172
host_response_time	5754	6165	51.724138
host_response_rate	5754	6165	51.724138
host_acceptance_rate	0	11919	100.000000
host_is_superhost	11917	2	0.016780

series	entry count	NaN count	percent of NaN (%)
host_picture_url	11917	2	0.016780
host_neighbourhood	8895	3024	25.371256
host_listings_count	11917	2	0.016780
host_total_listings_count	11917	2	0.016780
host_verifications	11919	0	0.000000
host_has_profile_pic	11917	2	0.016780
host_identity_verified	11917	2	0.016780
street	11919	0	0.000000
neighbourhood	10697	1222	10.252538
city	11918	1	0.008390
state	11917	2	0.016780
zipcode	11781	138	1.157815
market	11894	25	0.209749
smart_location	11919	0	0.000000
country_code	11919	0	0.000000
country	11919	0	0.000000
latitude	11919	0	0.000000
longitude	11919	0	0.000000
is_location_exact	11919	0	0.000000
property_type	11919	0	0.000000
room_type	11919	0	0.000000
accommodates	11919	0	0.000000
bathrooms	11888	31	0.260089
bedrooms	11914	5	0.041950
beds	11896	23	0.192969
bed_type	11919	0	0.000000
amenities	11919	0	0.000000
square_feet	262	11657	97.801829
price	11919	0	0.000000

The zipcode feature will serve as the basis for the location of the listings. However, there appears to be 138 entries that have no zip code. To remedy this, we'll make use of MapQuest API to fill the missing values. All 11,919 entries have non-NaN latitude and longitude values so we will employ this to our API request.

The **MapQuest API** features both forward and reverse geocoding and allows up to 15,000 monthly requests per API Key for non-commercial use. In our case, reverse geocoding will be employed to obtain the spelled out physical address given a pair of coordinates. From there, the zip code will be extracted and will then be used to fill the NaNs according to the index.

The url is the string that incorporates our coordinates, latitude and longitude, to the API using as our API_KEY. The response from the GET request is implemented into a JSON format, which we access like a Pythonic dictionary by retrieving keys with values.

MapQuest breaks down a physical address by street name, city, state, etc., and the zip code is expressed as postalCode. For each of the element in the no_zip list (which corresponds to the index of the entry with missing zip code), the postalCode is retrieved and stored as the value to the no_zip element's key of the dictionary zipcodes.

The zipcode series is converted to integers. However, there are entries that are in a dashed format so for these cases, the first part of the string is kept and the second part is discarded.

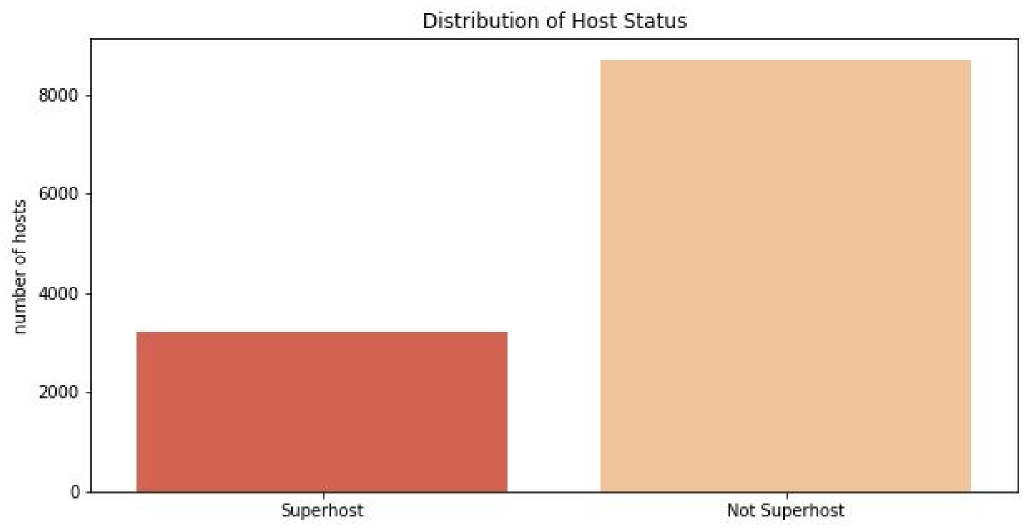
Listings Features Regarding Host

There are three columns we'll be able to use in the listings dataframe that concern the host. These are the host_is_superhost, host_verifications_count, host_identity_verified columns.

The first one is boolean and identifies whether the host is a *Superhost* or a typical host. Per Airbnb:

> *Superhosts are experienced hosts who provide a shining example for other hosts, and extraordinary experiences for their guests. Once a host reaches Superhost status, a badge will automatically appear on their listing and profile to help you identify them.*

As expected, there are significantly more typical hosts than Superhosts. We see this on the plot below.



Host status is actually expressed by a string *"t"* or *"f"*. We convert these to boolean by using the apply function. At the same time, the two NaNs are converted to a False – this is because we'd expect a host to be more likely to be a host than a Superhost given the statistics.

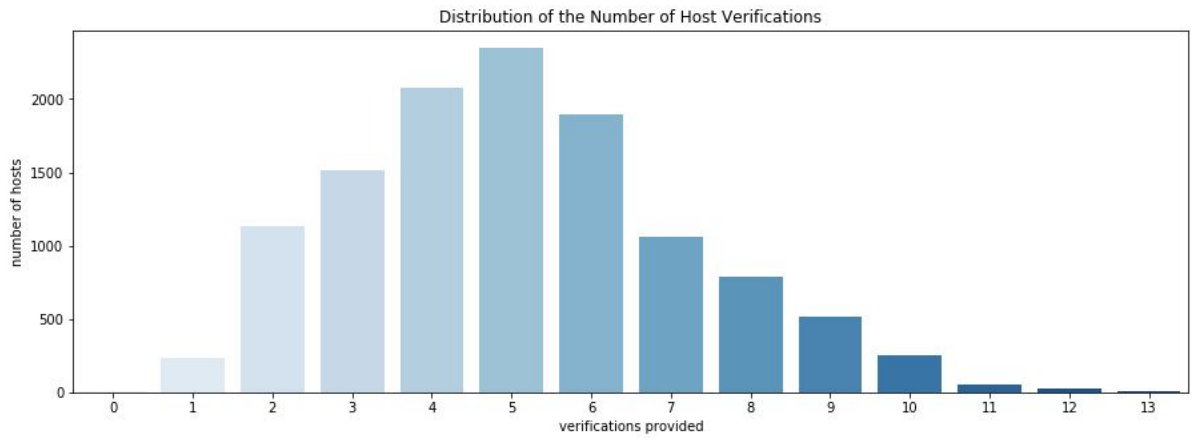
The host_verifications lists the verification methods performed by the host to prove online identity. An example would be *email* verification, wherein a host receives an email from Airbnb to confirm that the email is valid. Another would be *phone* verification, wherein Airbnb calls the phone number given and, if answered, will confirm that the phone number is indeed the host's. The more verifications a host undergoes, the more the host will appear trustworthy to a prospect guest.

This feature is meant to be a list of the verifications provided, but Pandas treated this column as a string. Since these entries are formatted as a list by having elements enclosed in a pair of brackets (and each element separated by comma), the entries can be interpreted as lists by associating Python's `eval` function with Pandas' `apply`. We then count the number of elements per entry and plug the totals into a new feature called `host_verifications_count`.

The heads of the original `host_verifications` feature and the newly created `host_verifications_count` are below.

	host_verifications_count	host_verifications
0	5	['email', 'phone', 'facebook', 'reviews', 'kba']
1	5	['email', 'phone', 'facebook', 'reviews', 'kba']
2	4	['email', 'phone', 'reviews', 'kba']
3	6	['email', 'phone', 'reviews', 'jumio', 'govern...

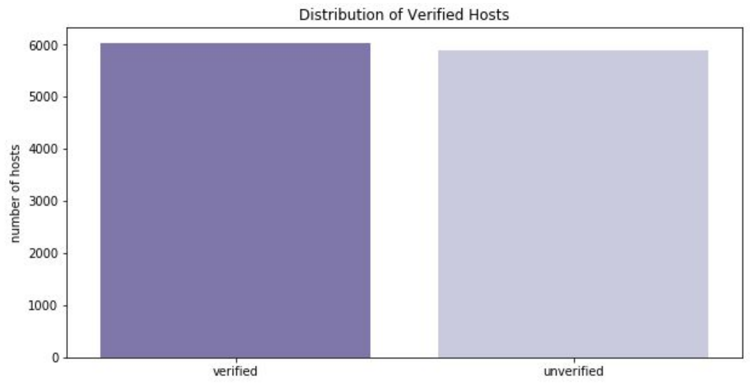
The chart showing distribution of hosts according to the `host_verifications_count` tells us that there are a maximum of 13 verifications. Only four hosts went through all of them.



Although a host can undergo all the verifications stated above, it does not mean a host's identity is genuine. Providing identification does not mean that a host is *verified*, per Airbnb's standards. In 2013, Airbnb rolled out a tool that distinguished a host with a *verified* badge on their profile if the host has proven that his or her online identity matches their *offline* identification. The following steps listed on Airbnb's press release described the verification process:

1. Confirm their established online identity (through pre-existing Airbnb reviews, LinkedIn, or Facebook);
2. Provide proof of their offline identity (by either confirming historical personal information or by scanning their photo ID); and
3. The two names (online and offline) must match.

This verification process is not required, although it seems likely that it heightens perceived trustworthiness. The `host_identity_verified` distribution table below shows that there are actually only several hosts that are verified than there are unverified ones.



Listings Features Regarding Property

The listings dataset has multiple columns describing the nature of the properties themselves. We'll begin with the three categorical features: the property_type, room_type, and cancellation_policy. Below are their unique values.

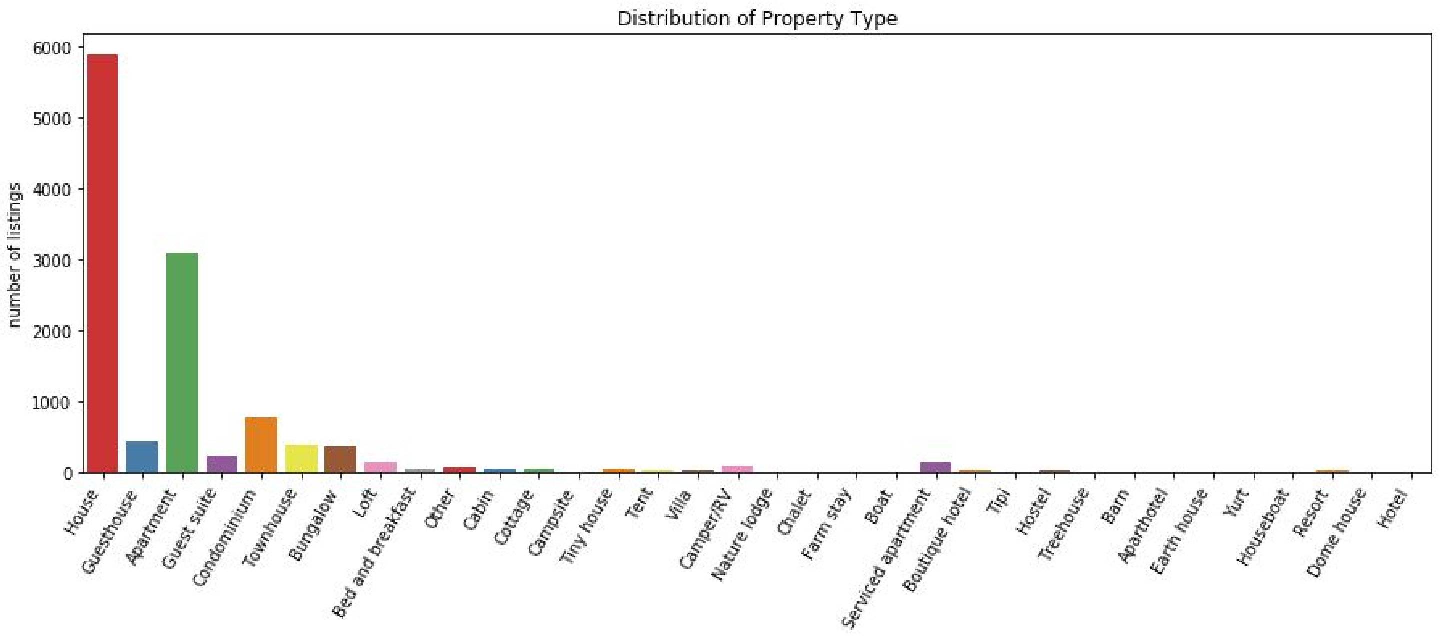
```
Unique 'property_type' values:
 ['House' 'Guesthouse' 'Apartment' 'Guest suite' 'Condominium' 'Townhouse'
 'Bungalow' 'Loft' 'Bed and breakfast' 'Other' 'Cabin' 'Cottage'
 'Campsite' 'Tiny house' 'Tent' 'Villa' 'Camper/RV' 'Nature lodge'
 'Chalet' 'Farm stay' 'Boat' 'Serviced apartment' 'Boutique hotel' 'Tipi'
 'Hostel' 'Treehouse' 'Barn' 'Aparthotel' 'Earth house' 'Yurt' 'Houseboat'
 'Resort' 'Dome house' 'Hotel']

Unique 'room_type' values:
 ['Entire home/apt' 'Private room' 'Shared room']

Unique 'cancellation_policy' values:
 ['strict_14_with_grace_period' 'moderate' 'flexible' 'super_strict_30'
 'super_strict_60']
```

There are too many categorizations for property_type. However, the distribution below shows that we can actually re-classify to just three types by bundling:

1. House
2. Apartment
3. Others

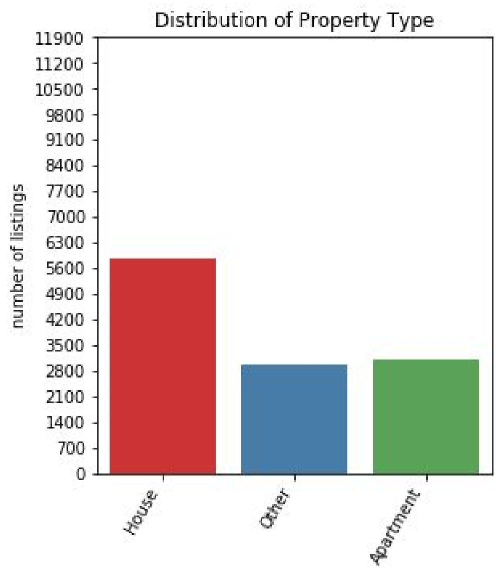


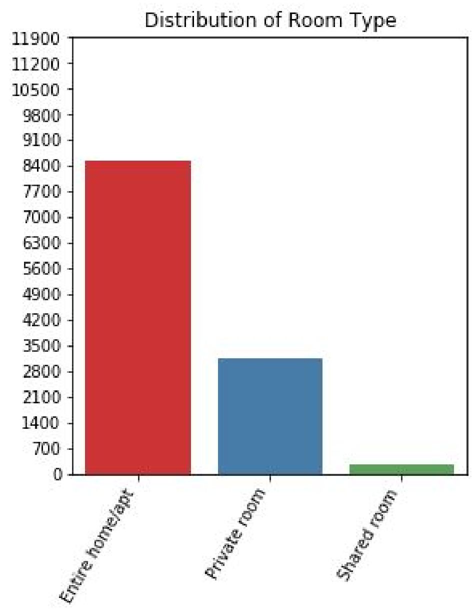
We simplify to keep only the House and Apartment entries while the rest are converted to Others. To compare, the head of the property_type and the newly generated property_type (simple) are shown.

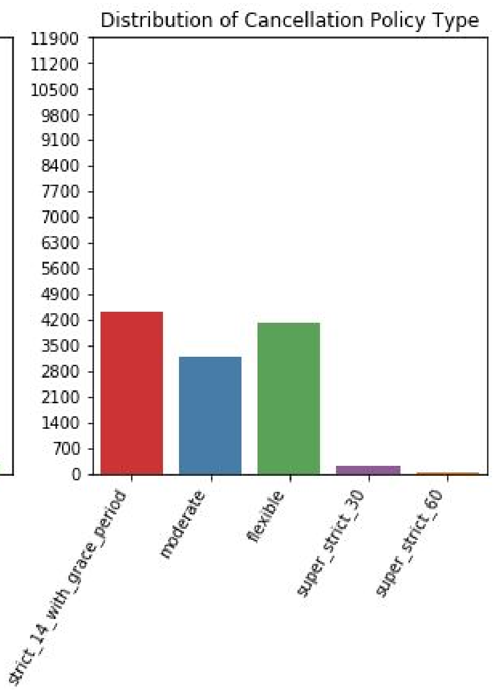
	property_type	property_type (simple)
0	House	House
1	House	House
2	Guesthouse	Other

	property_type	property_type (simple)
4	Apartment	Apartment
5	Guesthouse	Other
6	Guest suite	Other

The property_type (simple), room_type, and cancellation_policy have minimal categories yet succinctly describe the entries accordingly. Their distributions are below.



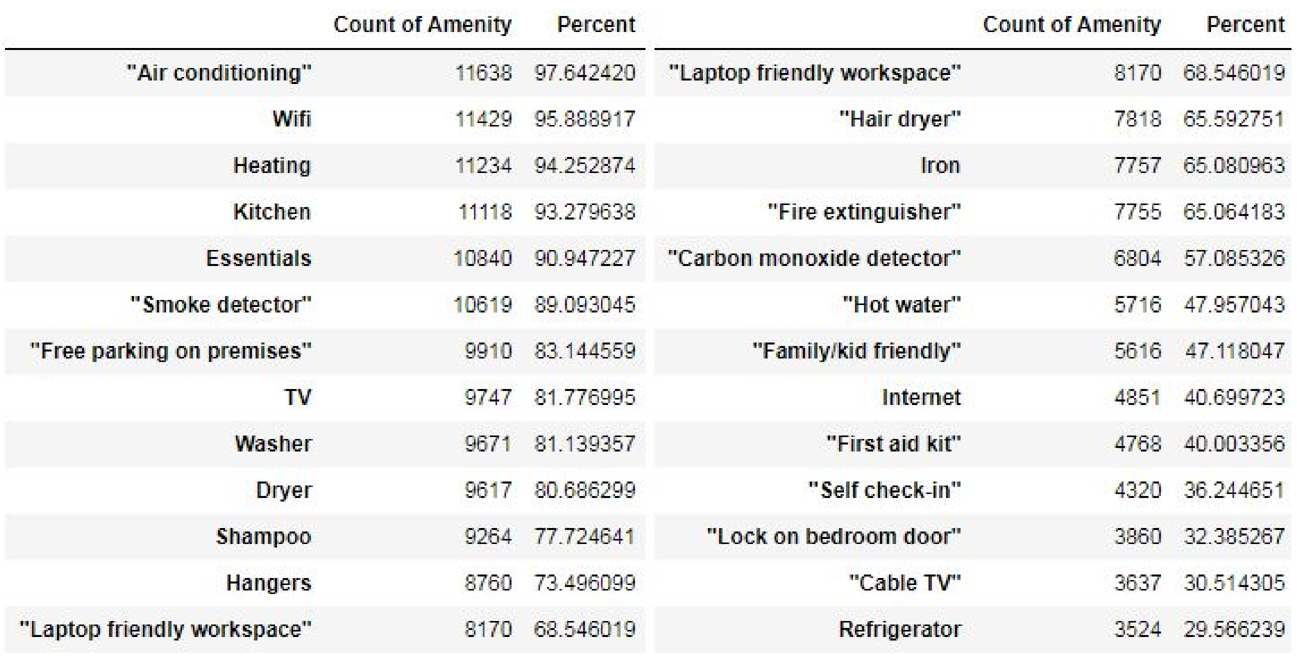




Listings Features Regarding Amenities

The amenities column of the listings dataset lists all the attributes a host claims a property has. This includes *TV*, *Wi-Fi*, or *hair dryer*. They could also be descriptions of the listing, including *family-friendly*, *24-hour check-in*, or if *pets* are allowed.

The entries to this column are elements enclosed in brackets but since curly brackets are used, Pandas interpreted this column as of the string datatype. So for every entry, the brackets are removed by slicing, and each row is split by comma. The amenities_dict is a Pythonic defaultdict that was used to tally each of the attribute entry and see how many listings made use of the attribute. There are a total of 193 amenities counted. The percentage of listings that has a given amenity is shown on the snippet below.



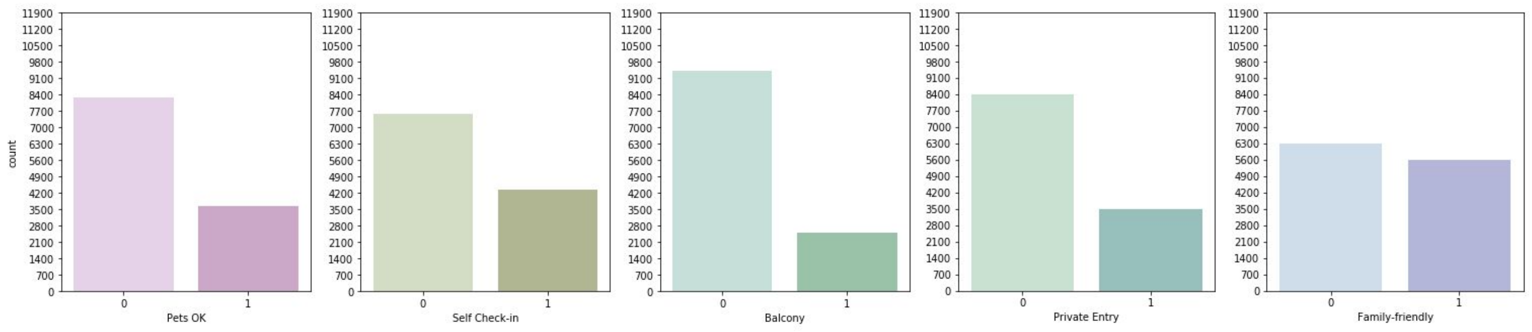
	Count of Amenity	Percent		Count of Amenity	Percent
"Air conditioning"	11638	97.642420	"Laptop friendly workspace"	8170	68.546019
Wifi	11429	95.888917	"Hair dryer"	7818	65.592751
Heating	11234	94.252874	Iron	7757	65.080963
Kitchen	11118	93.279638	"Fire extinguisher"	7755	65.064183
Essentials	10840	90.947227	"Carbon monoxide detector"	6804	57.085326
"Smoke detector"	10619	89.093045	"Hot water"	5716	47.957043
"Free parking on premises"	9910	83.144559	"Family/kid friendly"	5616	47.118047
TV	9747	81.776995	Internet	4851	40.699723
Washer	9671	81.139357	"First aid kit"	4768	40.003356
Dryer	9617	80.686299	"Self check-in"	4320	36.244651
Shampoo	9264	77.724641	"Lock on bedroom door"	3860	32.385267
Hangers	8760	73.496099	"Cable TV"	3637	30.514305
"Laptop friendly workspace"	8170	68.546019	Refrigerator	3524	29.566239

Given that Austin is known for dry heat, almost all the listings are shown to have air-conditioning. The same is true for Wi-Fi and/or Internet. These are almost standard features that would not be helpful to our model because they are not distinct.

The following amenities are chosen because they have varying distributions that do not represent *all* or *none* of the listings:

1. **Private Entry**
   The `Private entrance` is straightforward and doesn't have any variations in spelling or denotation.
2. **Family-friendly**
   This is drawn from the `Family/kid friendly` amenity and is also straightforward.
3. **Pets OK**
   There are four variations of this amenity. This feature will be True if any of the following appears in the listing's amenity: `Pets allowed`, `Cat(s)`, `Dog(s)`, or `Other Pet(s)`.
4. **Self Check-in**
   This is True when the attribute `Self check-in` is among a listing's amenity.
5. **Balcony**
   There are two variations of this amenity, either `Balcony, Patio` or simply `balcony`. So the requirement for this feature to be True is when a lower-cased *balcony* appears as its attribute.

The distribution for the five columns are shown below.



Continuous Listings Features

There are five candidate features for our model that are continuous (or actually are discrete but *appears* continuous): `accommodates`, `guests_included`, `extra_people`, `minimum_nights`, `number_of_reviews`.

The `accommodates` feature is the total number of people the listing can hold. This can also be referred to as the capacity of the property. There are too many outliers spread out beyond the 75th percentile so to simplify this feature, anything above 20 is truncated to `20`.

The `guests_included` column identifies how many people are included in the quoted nightly rate. Again, there are numerous outliers so the values above 20 are truncated to `20`.

The `extra_people` is the additional cost per night for every additional guest beyond the `guests_included` of a listing. The `listings` dataset denotes entries to this column in a monetary format. So a hundred dollars is expressed as `$100.00`, therefore a string. The dollar sign is then removed from each entry before the entire column is converted to float using the `astype` function. NaNs are filled with zeroes since this means there just isn't an extra charge for additional guests. There is an outlier here that requests $500 for an extra guest. This value is changed to match the next maximum value, `300`. This narrowed the distribution a bit.

Some listings do not allow for single-night stays and may require up to two or more nights, hence the feature `minimum_nights`. Most listings actually do not require minimums. Though there are entries that require a year's worth of stay to book, it can be said that a week is enough to be considered *long-term* given that Airbnb is primarily a service for a few days' stay. Therefore the listings that require more than 7 days for their `minimum_nights` are truncated to `7`.

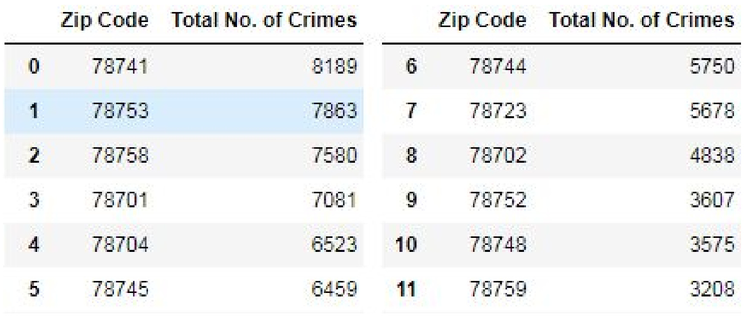
The `number_of_reviews` may indicate to some degree the popularity of the listing. This does not reflect good or bad reviews, however. A listing with more than 75 reviews or more can definitely be said to be popular. This value is also the 75th percentile. We then truncate those with higher number of reviews to a value of `75`.

Sub Dataframe: Crimes

The `crimes` dataset contains all the reported crimes in Austin, TX that occurred in 2018, the same timeframe as the `listings` dataset. This dataset's `Zip Code` is what we'll use to relate the relevant `crimes` features to the `listings` dataframe. In all, there are 948 missing entries for this column, or about 1%. Out of these, there are 181 entries that has valid entries for `Location`, which is a combination of `Latitude` and `Longitude`.

Again, we'll make use of the MapQuest API in order to give these 181 entries their correct `Zip Code`. From 948, we have reduced the NaN count for `Zip Code` to just 768. Since reports that have unknown `Zip Code` are useless to our study because they will not be able to contribute to our Airbnb data, we'll proceed to drop them.

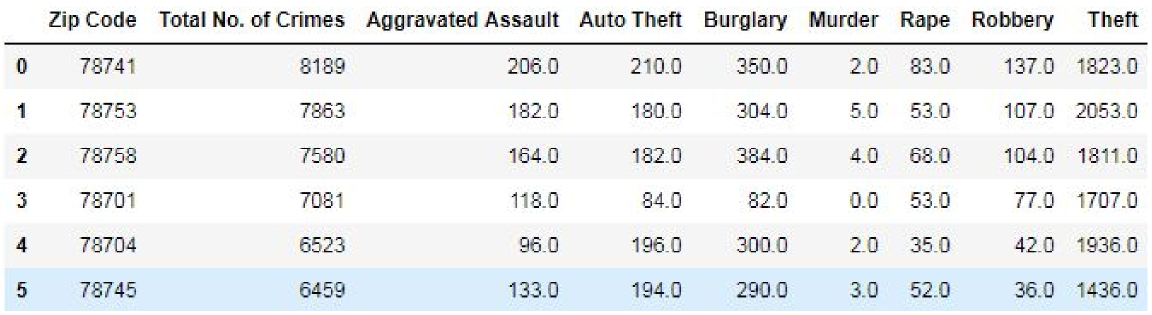
From the `crimes` dataframe, the `crimes_per_zip` is generated by grouping entries by `Zip Code` and aggregating by count. This gives us an overall picture of the total number of crimes in a given area.



	Zip Code	Total No. of Crimes
0	78741	8189
1	78753	7863
2	78758	7580
3	78701	7081
4	78704	6523
5	78745	6459
6	78744	5750
7	78723	5678
8	78702	4838
9	78752	3607
10	78748	3575
11	78759	3208

Crimes are categorized on a top-level according to the FBI designation mentioned in the *Datasets* section. However, only serious crimes have this `Category Description`. The `major_crimes` is a dataframe derived from a pivot_table that aggregated the number of reports by `Category Description` and indexed by `Zip Code`. The NaNs are rightly filled with zeroes.

The `crimes_per_zip` dataframe, which looks at the totals of *all* crimes, and the `major_crimes` dataframe, which details only *serious* crimes, are merged inclusively (outer-wise). The resulting dataframe is `crimes_clean` and its NaNs are filled with zeroes. This is because missing values occur when there is no crime report that exists.



	Zip Code	Total No. of Crimes	Aggravated Assault	Auto Theft	Burglary	Murder	Rape	Robbery	Theft
0	78741	8189	206.0	210.0	350.0	2.0	83.0	137.0	1823.0
1	78753	7863	182.0	180.0	304.0	5.0	53.0	107.0	2053.0
2	78758	7580	164.0	182.0	384.0	4.0	68.0	104.0	1811.0
3	78701	7081	118.0	84.0	82.0	0.0	53.0	77.0	1707.0
4	78704	6523	96.0	196.0	300.0	2.0	35.0	42.0	1936.0
5	78745	6459	133.0	194.0	290.0	3.0	52.0	36.0	1436.0

Putting It All Together

The unique values of the zip codes for the listings dataset are compared with the unique values of the zip codes for the crime dataset. By turning the lists into sets, what is present in one and not in the other can easily be determined.

There are three zip codes that are in the Airbnb dataset that are *not* in the crime data. This is because the crime data focuses in Austin, TX alone, whereas the Airbnb dataset referred to Austin, TX in terms of both the city *and* a few of its neighboring suburbs that are most often associated with the Texan capital.

```
Zip codes present in Airbnb listings but not in crimes data: {78619, 78620, 78669}
```

An interim dataframe listings_clean is created by merging the listings and the crimes_clean dataframes. From the listings_clean, the final dataframe df is generated by filtering the 27 necessary features for our model.

A preview of df is below.

	id	price	zipcode	host_is_superhost	host_verifications_count	host_identity_verified	property_type (simple)	room_type	cancellation_policy	Private Entry
0	2265	$225.00	78702	1	5	1	House	Entire home/apt	strict_14_with_grace_period	1
1	5245	$125.00	78702	1	5	1	House	Private room	strict_14_with_grace_period	0
2	5456	$95.00	78702	1	4	1	Other	Entire home/apt	strict_14_with_grace_period	1
3	5769	$40.00	78729	1	6	1	House	Private room	moderate	0

	Family-friendly	Pets OK	Self Check-in	Balcony	accommodates	guests_included	extra_people	minimum_nights	number_of_reviews	Total No. of Crimes
0	1	1	1	0	4	4	30.0	2	23	4838.0
1	1	1	0	0	2	2	25.0	2	9	4838.0
2	1	1	0	1	3	2	45.0	2	75	4838.0
3	1	1	0	0	2	2	0.0	1	75	1768.0

	Aggravated Assault	Auto Theft	Burglary	Murder	Rape	Robbery	Theft
0	106.0	120.0	211.0	1.0	28.0	72.0	1190.0
1	106.0	120.0	211.0	1.0	28.0	72.0	1190.0
2	106.0	120.0	211.0	1.0	28.0	72.0	1190.0
3	30.0	45.0	56.0	1.0	7.0	10.0	475.0

There are a total of 4 entries that have NaNs in any of the columns of df. These refer to Airbnb listings that *do not* have crime data associated with them because they are in a zip code outside the scope of the crimes dataset. Since it will be wrong to assume that any crime or no crime at all occurred in these areas, and since there are only four of these listings in total, we'll opt to drop these instead.

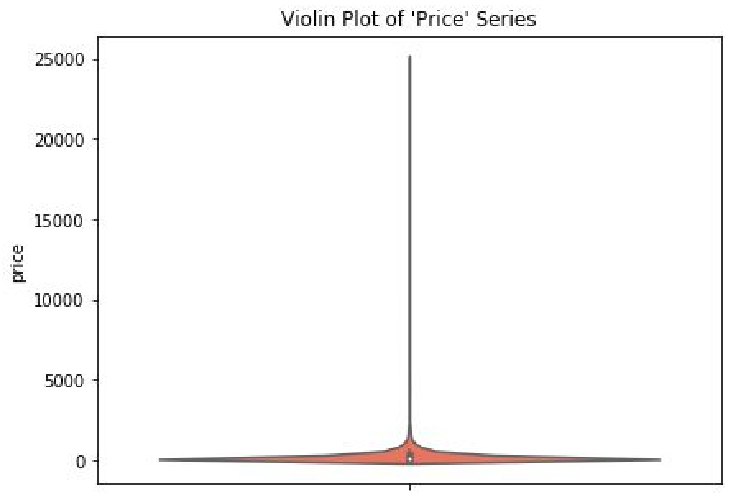
```
Total 'df' entries prior to dropping NaNs: 11919
Total 'df' entries after dropping NaNs: 11916
```

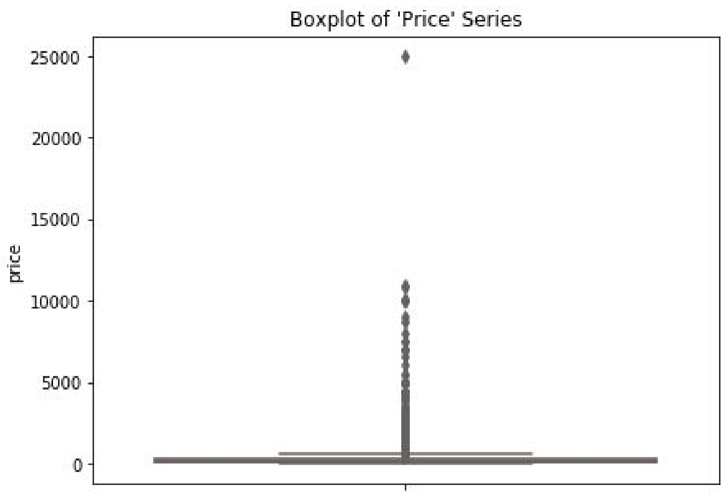
Price: The Main Variable

The dependent variable in our study is of course the nightly rate of the Airbnb listing. This is the price feature of our df. Again, its entries are expressed like monetary values, e.g. $1,000.00. The dollar sign is then removed and the string is split by comma then re-joined before being converted to a float.

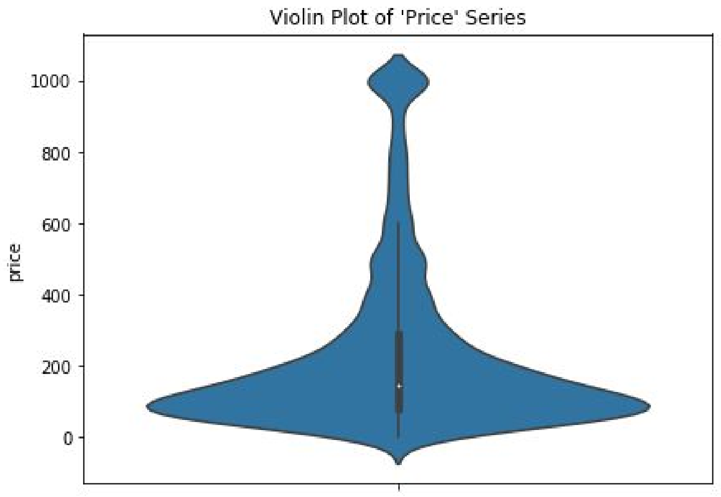
There are actually 8 entries that have price of 0. These listings can be viewed online but are unavailable due to being booked indefinitely, either because the host put it off the market or placed a temporary hold. Since these 8 listings have a price that do not reflect their actual value, we will update df without these entries.

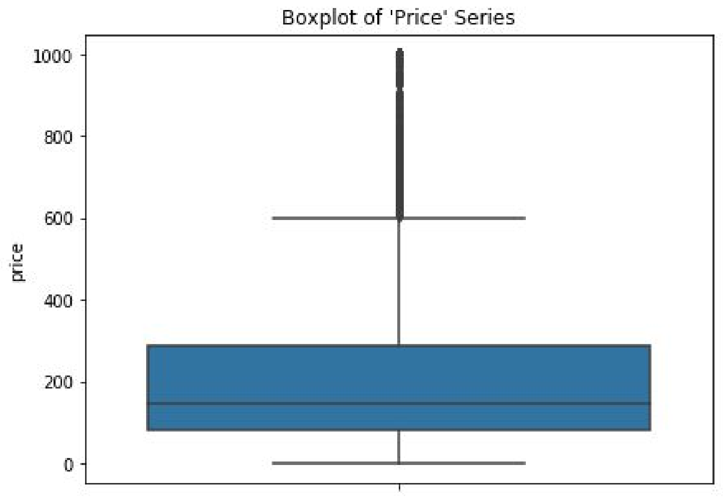
The price for a nightly stay varies very widely by listing, as shown by the boxplot and violin plot below.





By sorting the series by price, we get to see the most expensive listings. If entries are limited to $1,000 per night max, which is already a ludicrous rate, then we get a price that has better distribution. We see that rates above the 75th percentile are denoted appropriately as outliers.





Data Story

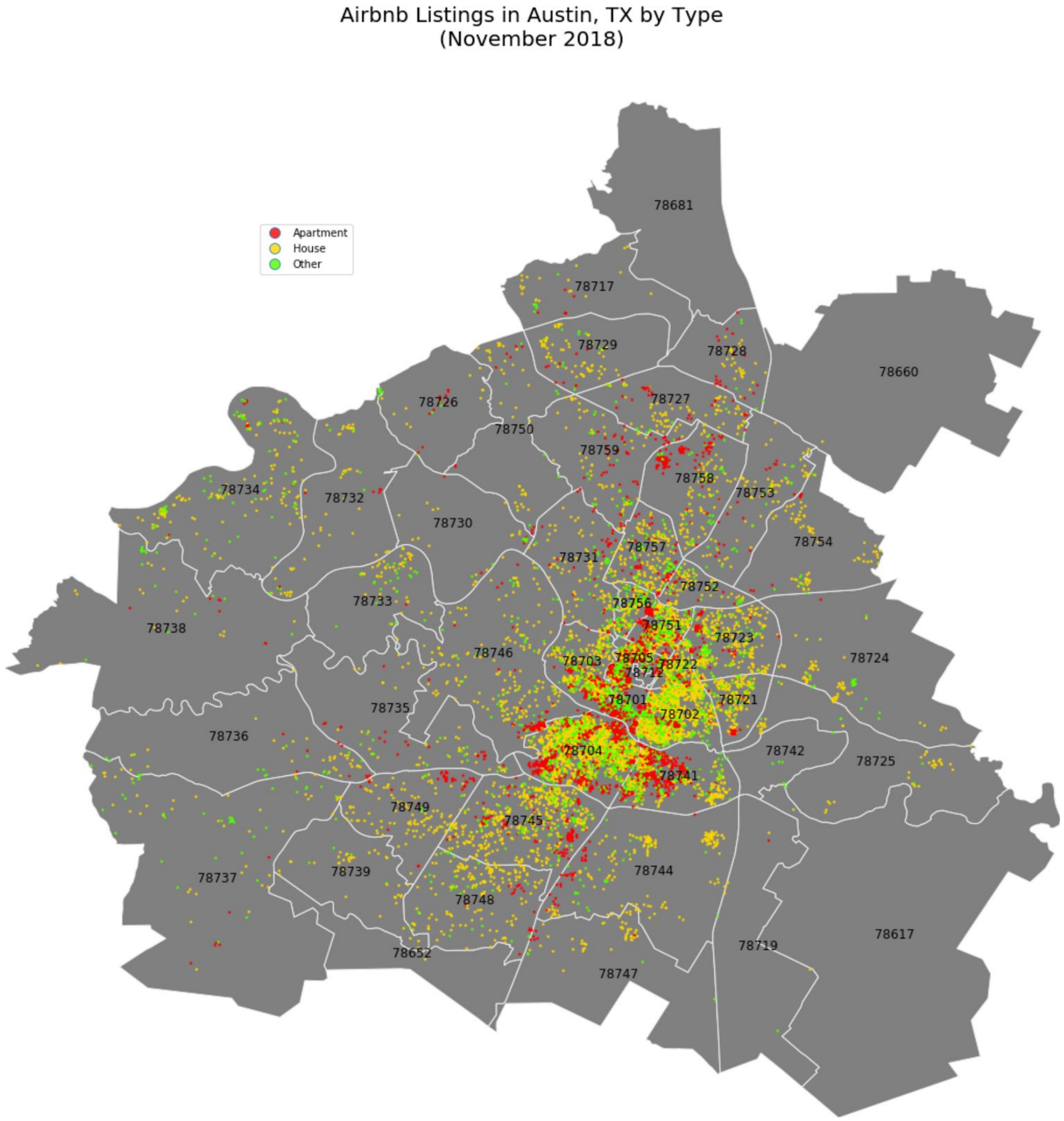
Austin, TX Map

The Airbnb listings dataset contains the global coordinate locations of every listing in Austin, TX. Likewise, the crimes dataset specifies the location of all the crimes that occurred (and were reported) in Austin, TX in the year 2018. These information aren't part of our final df dataset because we have opted to make use of zip codes to categorize the listings' locations. Still, we'll make use of these geographical points to explore our data further through the visualization of maps. We'll make use of the austin_map dataframe as the backdrop of our plots. This was derived from the Zip Codes GeoJSON taken from the City of Austin data portal.

Location and Price

We've simplified the classification of Airbnb listings into three types: *Apartment*, *House*, and *Other* – which is the category for the miscellaneous and even quirky listings including RV's, yurts, treehouses, and more.

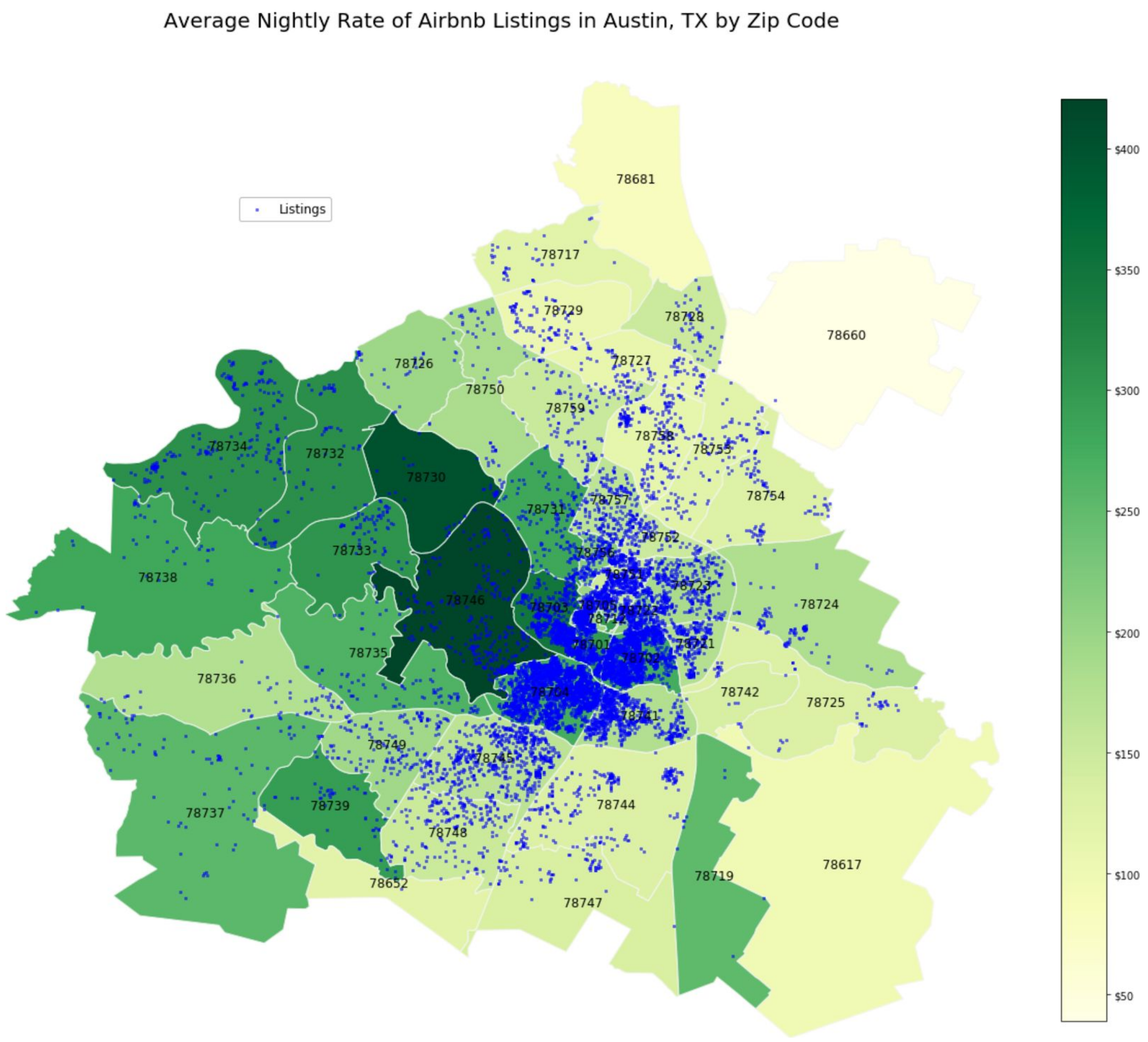
Below is the plot of all the Austin, TX Airbnb listings categorized by type.



From the figure, it was evident that listings are more concentrated within the inner zip codes, and listings begin dwindling and become more scattered as we go outward. We can draw different inferences from this – perhaps more listings suggest a more urban area. Or perhaps the farther out we are, the more regulations are put in place to ward off commercialization of suburban homes.

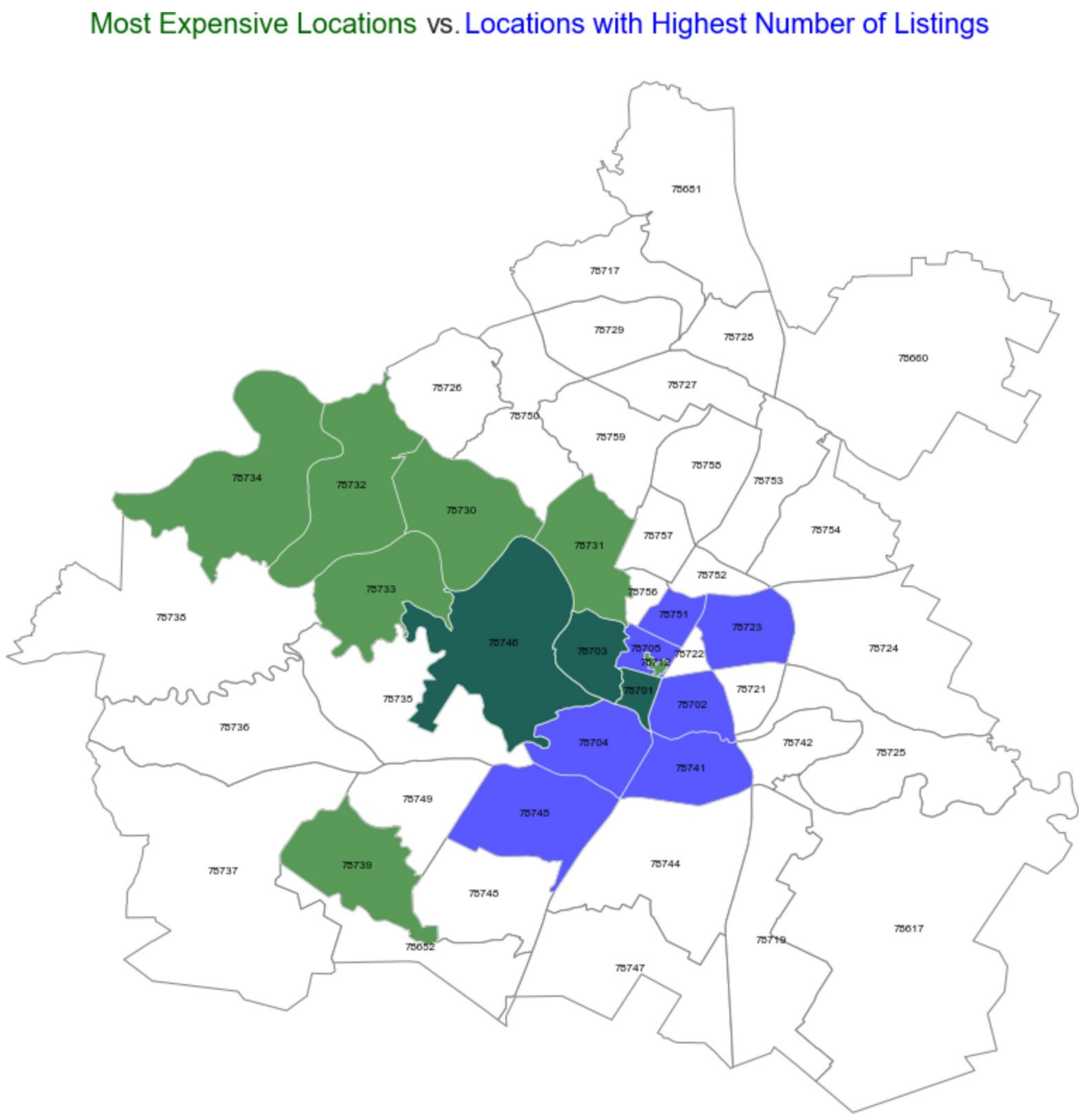
We can associate the cluster of listings with our main variable, *price*. Below is the plot of all the listings again but this time, on the backdrop of how expensive the zip code is in terms of *average* nightly Airbnb rate. The darker the green is, the more expensive, and the lighter the green is, the more affordable its listings are.

A good hypothesis is that prices should be cheaper in places that are more clustered because the competition should have lowered the price down. At the same time, the counterpoint could just be as likely: that more listings mean higher prices because this suggests more attractions and/or businesses are within close proximity.



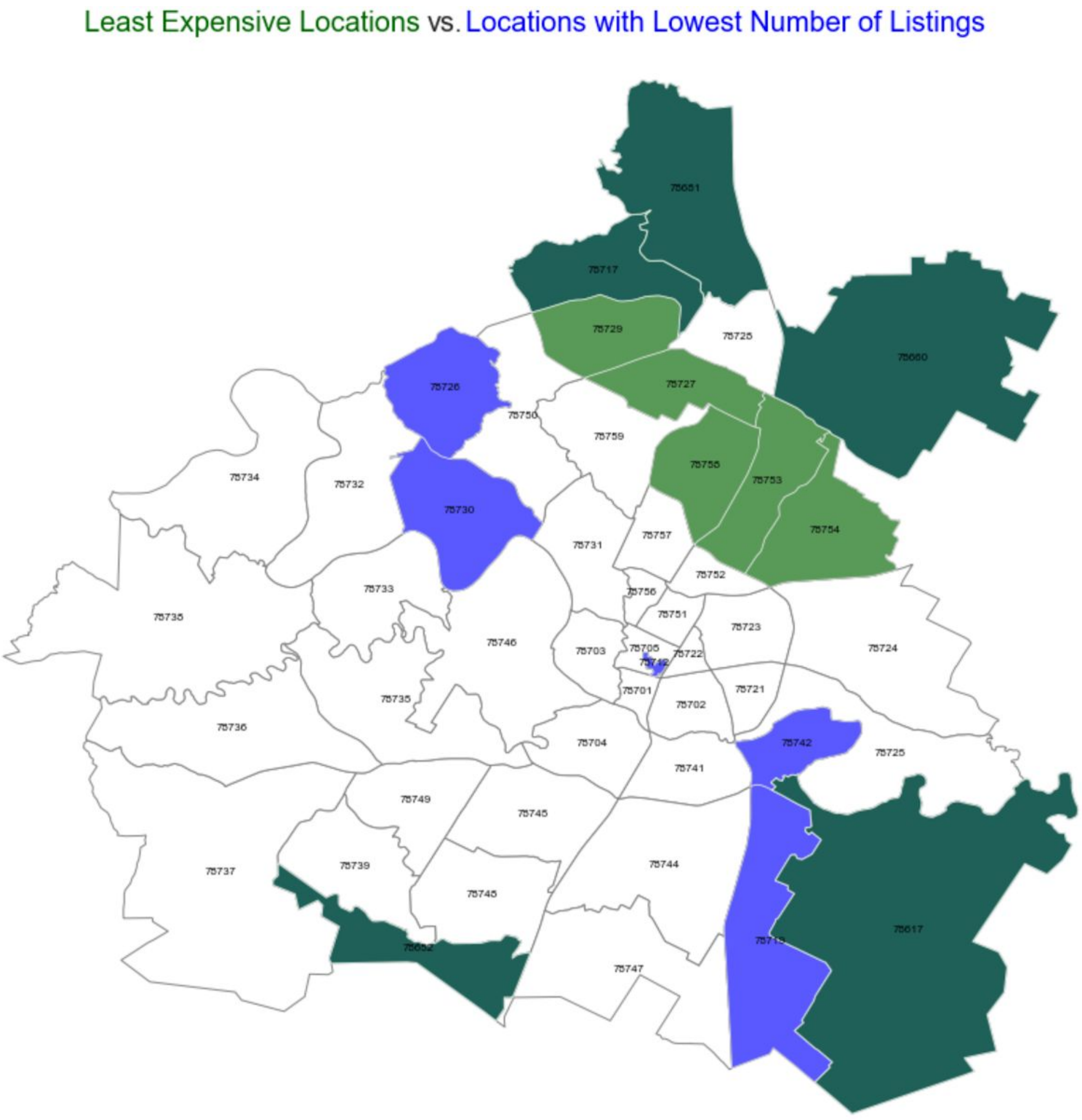
We see that, indeed, prices are higher in a zip code when there are a higher number of listings, though not exactly. Zip code 78730 is the second darkest area (suggesting the second priciest zip code) but there are only a handful of listings within it. And the opposite is true, in some cases, including 78757 which holds a lot of listings – most of which seem to be affordable according to their mean.

To get a better picture, below is the top 10 most expensive Airbnb zip codes (*in green*) and the top 10 places with the most Airbnb listings (*in blue*). If there is high correlation then we should see a lot of intersections (*in blue-green*).



There were three places that happen to be in *both* the top 10 most expensive zip codes and the top 10 most clustered. These are 78746, 78703, and 78701.

Another way to intuitively check correlation is to see the flipside of this: below is the top 10 least expensive zip codes and the top 10 locations that have the lowest number of listings.



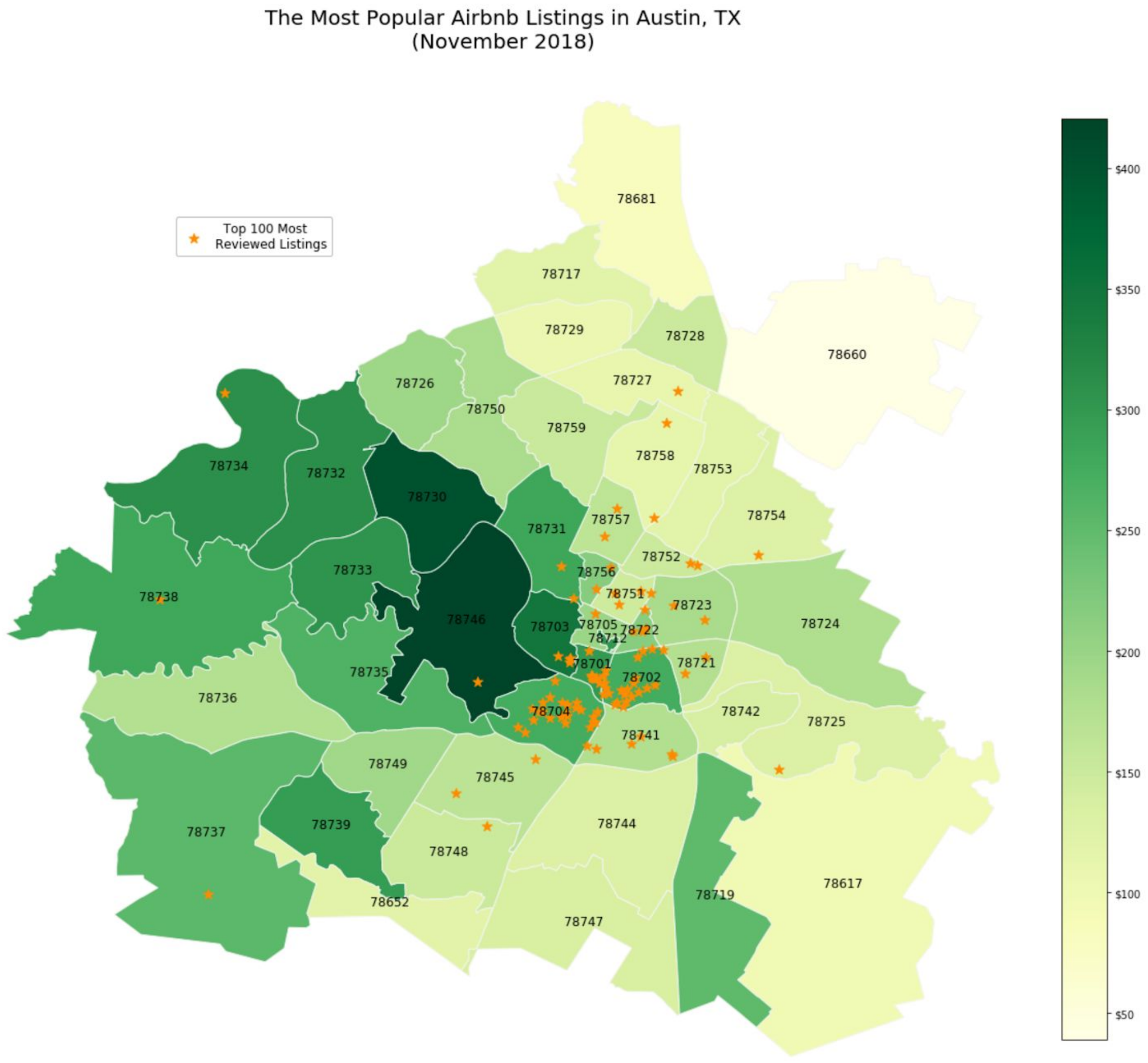
This time they are more aligned, with *half* of the samples that intersect. These are 78681, 78717, 78660, 78617, 78652.

It is not clear-cut, but it's safe to say that in very general terms, price do have a relationship with the number of listings in a given area.

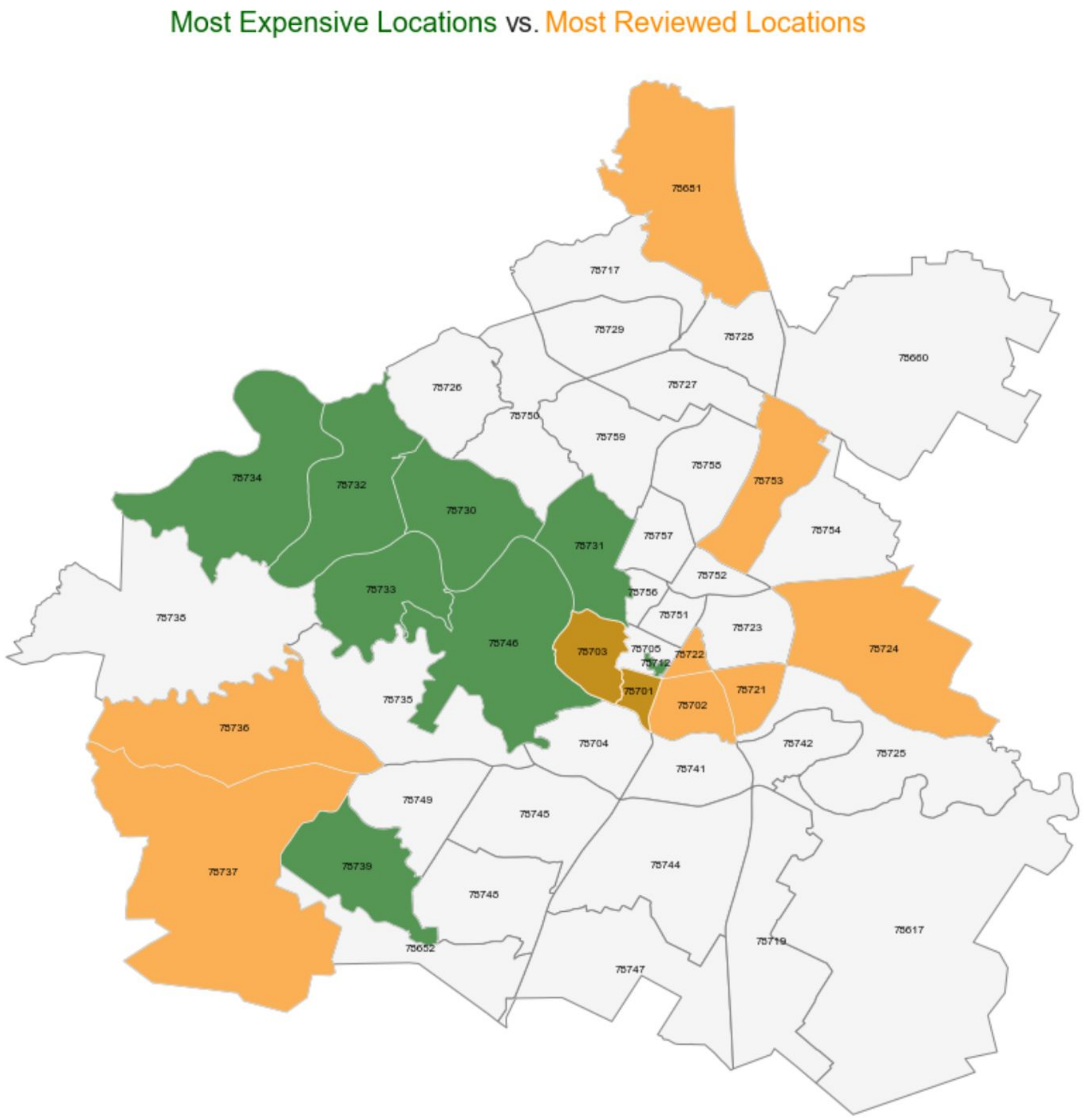
Popularity and Price

We also have data on the number of reviews a listing has. We cannot say matter-of-factly that the more reviews a listing gets, the better rated it is. After all, a listing may have thousands of reviews but a majority could be *negative*. What a high number of reviews suggests instead is *popularity*. Popularity here means the same way as how an online retailer would describe a "best-seller". A popular listing just happens to have been more booked.

Below is the plot of the top 100 most reviewed listings in Austin, TX. A good hypothesis here is that the more reviews a listing has, the pricier it is. If so, this could be because popular listings tend to be more attractive. Or perhaps the more guests a host has had, the more experienced the host becomes in terms of hospitality.

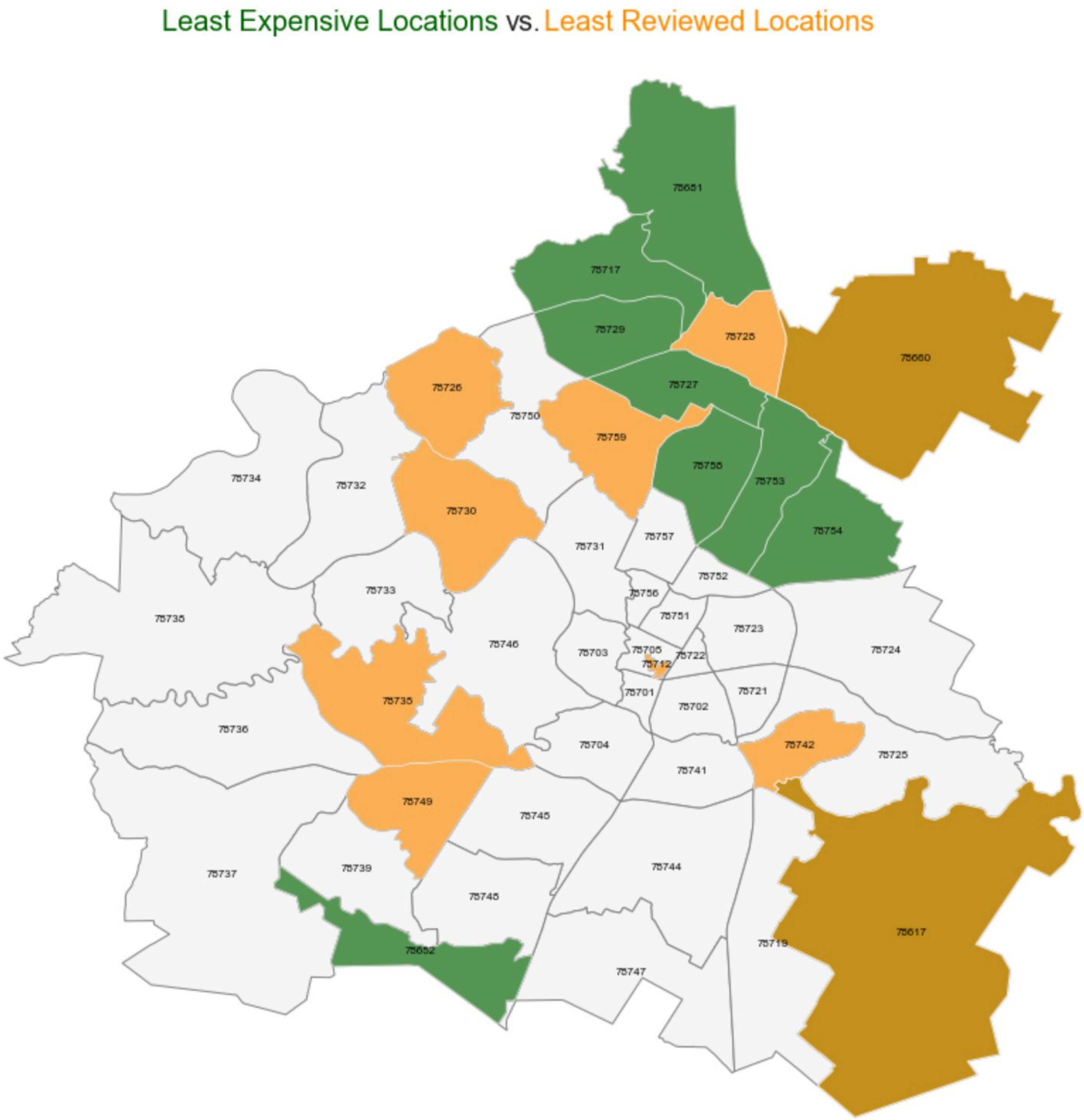


We do see that the most reviewed listings are clustered in areas that tend to be more expensive. But to test this deeper visually, we'll again plot the top 10 most expensive places (*in green*) along with the top 10 most reviewed places (*in orange*) to see if there are any intersections (*in gold*).



We see that only two of the most expensive zip codes happen to also be the places where the most reviewed listings are. These are 78703 and 78701.

Again, we'll do the opposite and see if the top ten least expensive locations coincide with the top ten least reviewed locations.



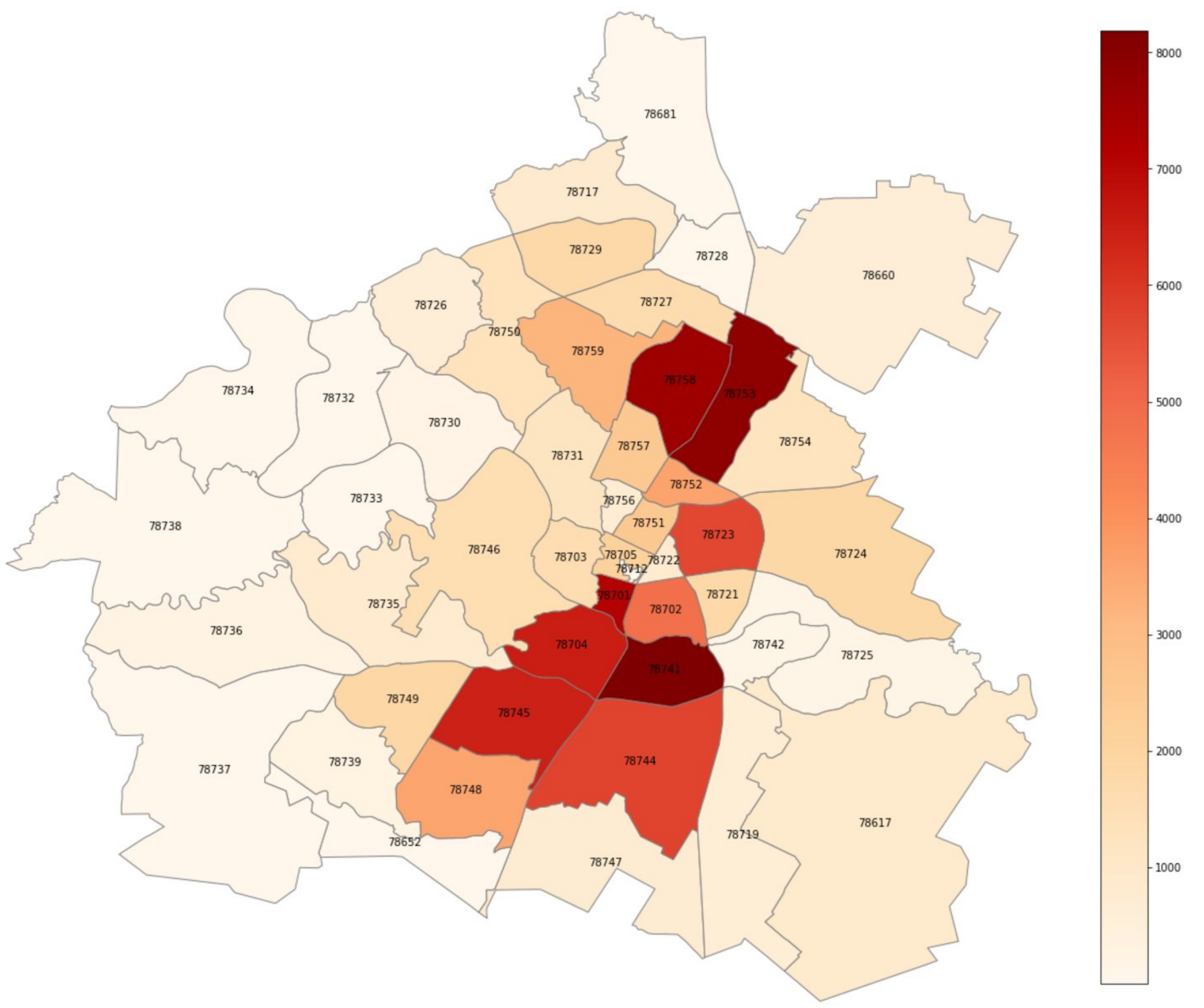
We learn that there are also only two zip codes that happen to be the least expensive and the least reviewed. These are 78660 and 78617.

From the above plots, it can be said that there is a relationship between the number of reviews of a listings and its price – though it's quite weak.

Crimes and Price

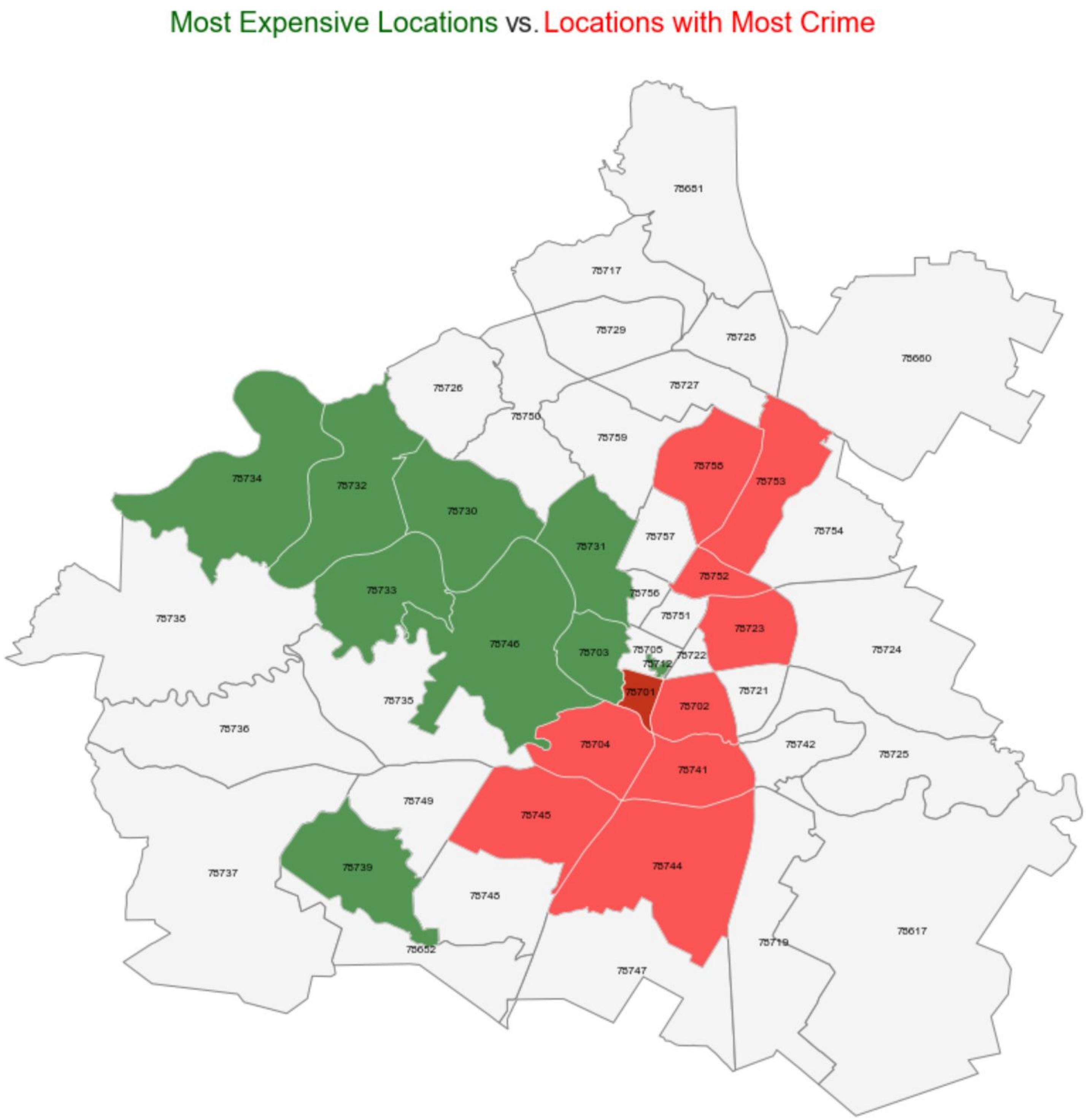
Below is the total number of crimes that occurred in every Austin, TX zip code during 2018. This was plotted after aggregating all the crimes in the crimes dataset in only the zip codes that are also present in our listings dataset.

Since high crime just happens to be an unfortunate byproduct of a bustling city, then a good hypothesis would be: more crime indicates an expensive zip code. Note that we are keeping it simple and are looking at the total number of crimes in *absolute* terms, as a count of how many were reported. We are not looking at *per capita* crime since there are no population data in our datasets.

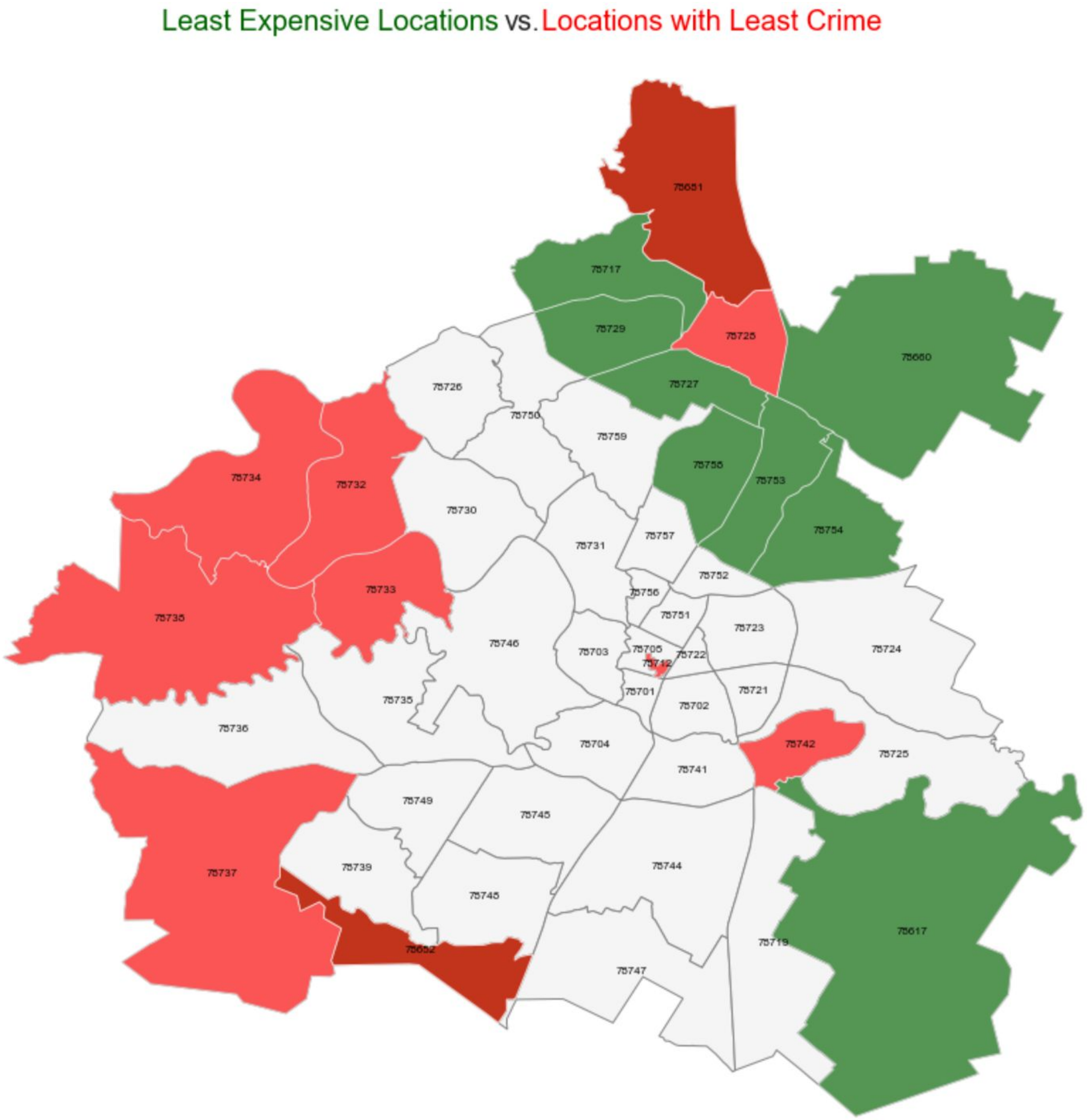


Total Number of Crimes Reported in Austin, TX by Zip Code (2018)

We can test the relationship of high crime and high price by plotting the top 10 most expensive Airbnb zip codes (*in green*) along with the top 10 locations with the most crime (*in red*). We see that there is only one intersection, 78701, in maroon.



The opposite can also test the relationship: this is by plotting the top ten with the least crimes against the top ten least expensive places. Again, there isn't much in terms of intersection, with only two zip codes that coincide: 78681 and 78652.



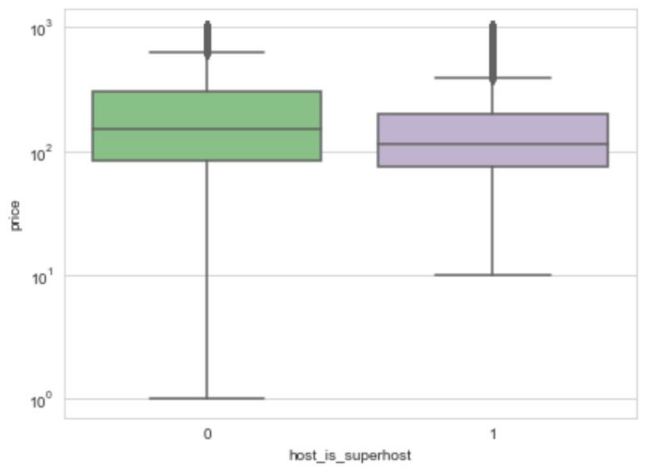
This then tells us that high crime doesn't necessarily mean higher price. Perhaps because crime may not be a concern for a guest that has already decided to book a place in Austin, TX, especially if the crime rate isn't dire enough to stave off tourism.

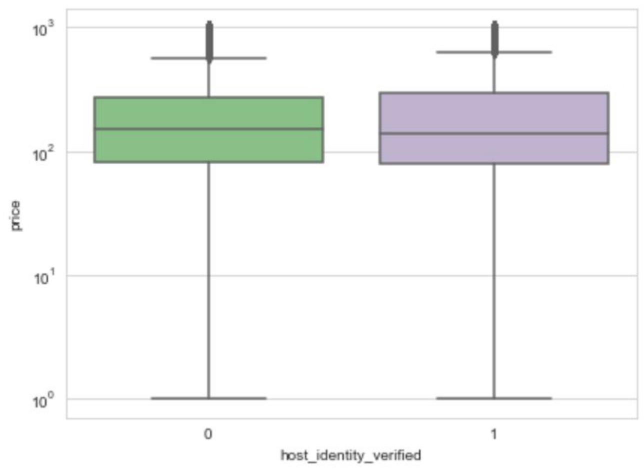
Distribution of Categorical Data

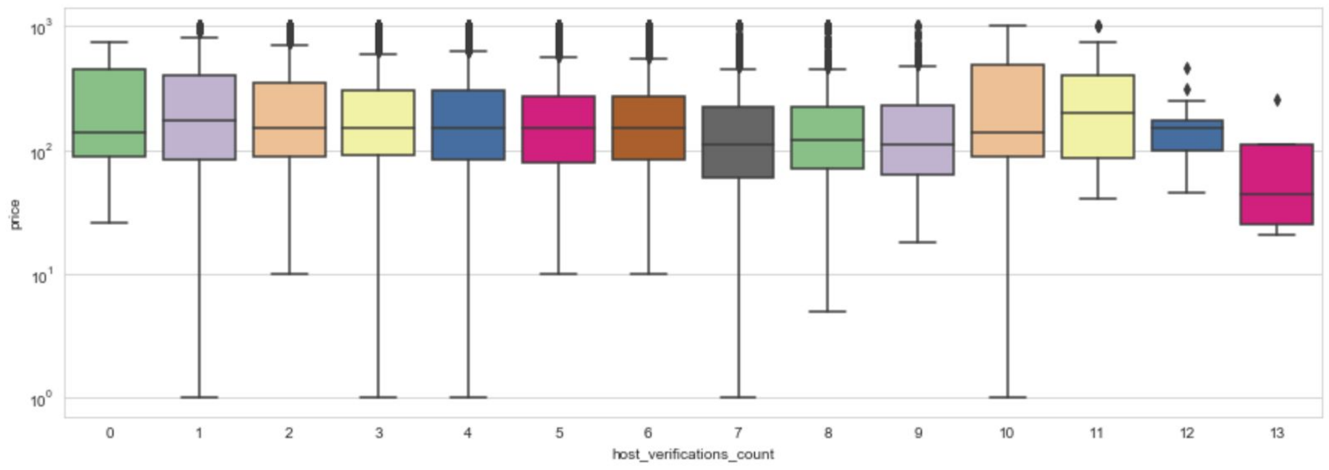
A lot of the features of our final df dataframe are discrete with minimal range of values. If we group the distribution of price per a categorical feature's range of values and plot them in a boxplot, we would be able to visually *infer* if the feature would be able to predict our dependent variable, *price*.

The figures below characterize the listing in terms of its host. As we see, host_identity_verified shows almost no variability across its two categories and that 1 is only *slightly* different from 0 in terms of spread. That tells us that this is not a good predictor for our model.

On the other hand, host_is_superhost is more characterizing as there is a noticeable shift.

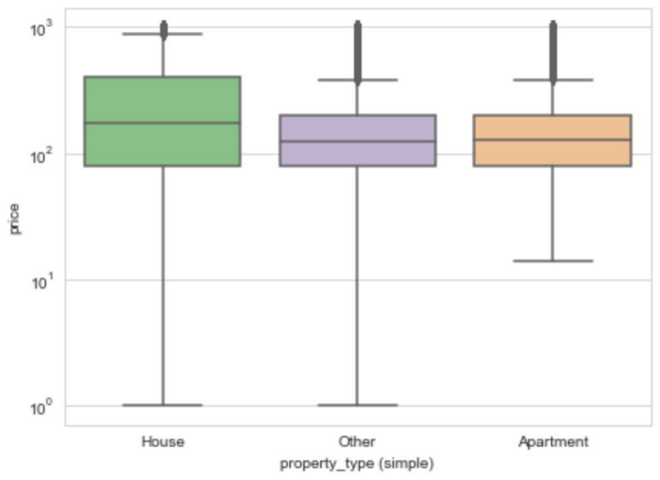


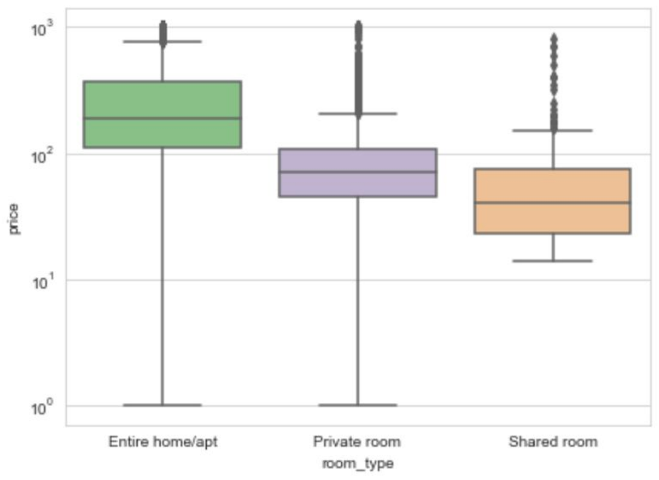


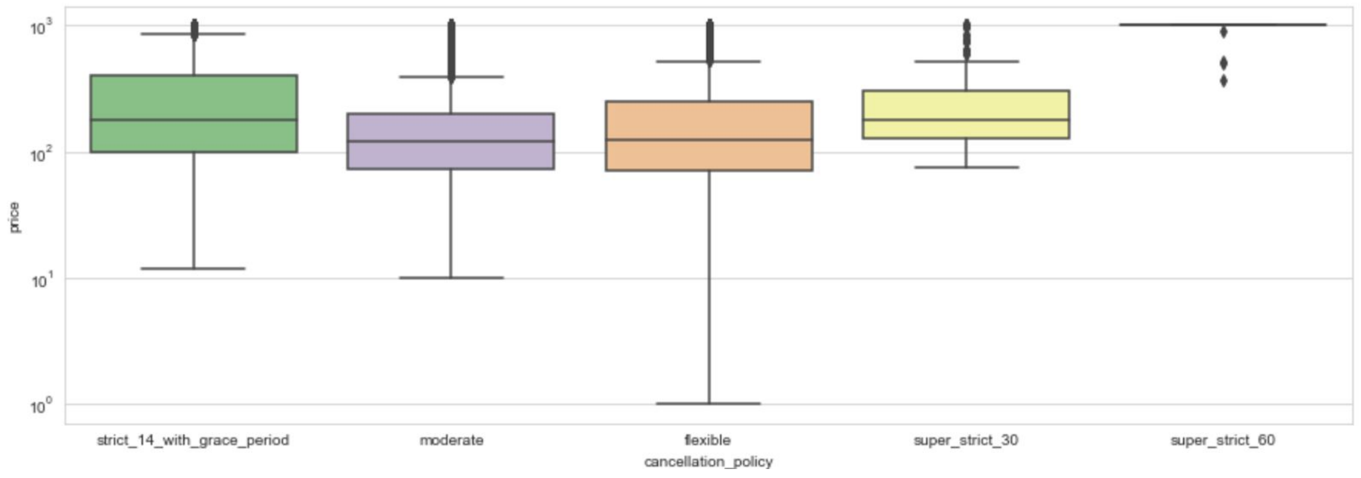


The next set of features look at the categorical columns of listings in terms of their nature. The room_type is a very good candidate for our model because of how distinct each of its value is. It is immediately clear from the figure that entire houses are more affordable than private rooms, and even more so than with shared rooms.

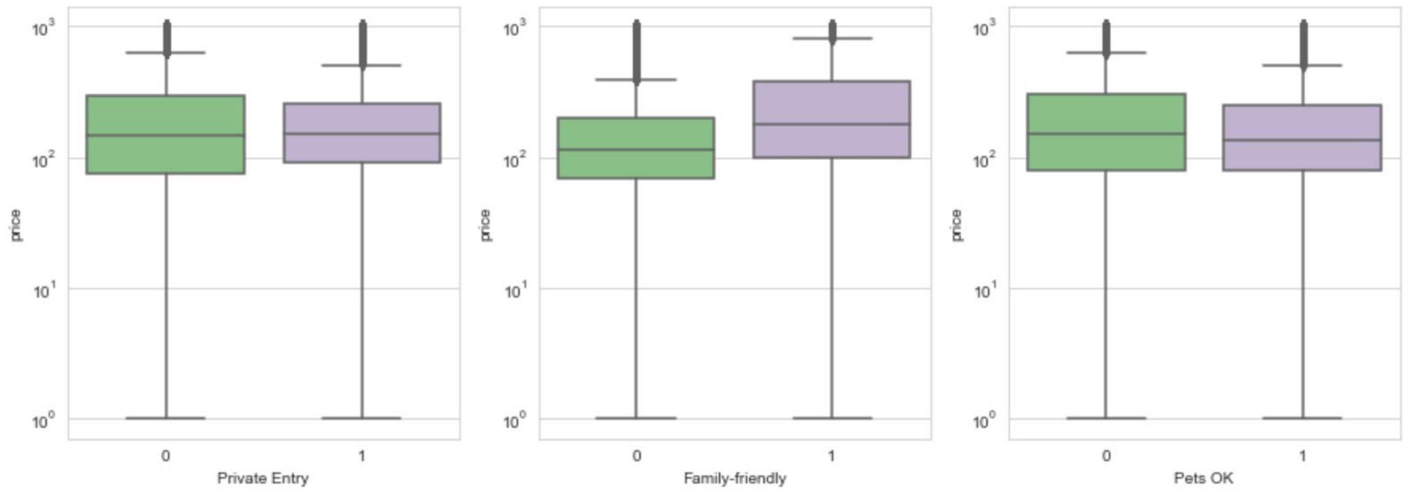
The cancellation_policy boxplots show us that the most expensive places have the strictest booking policies.

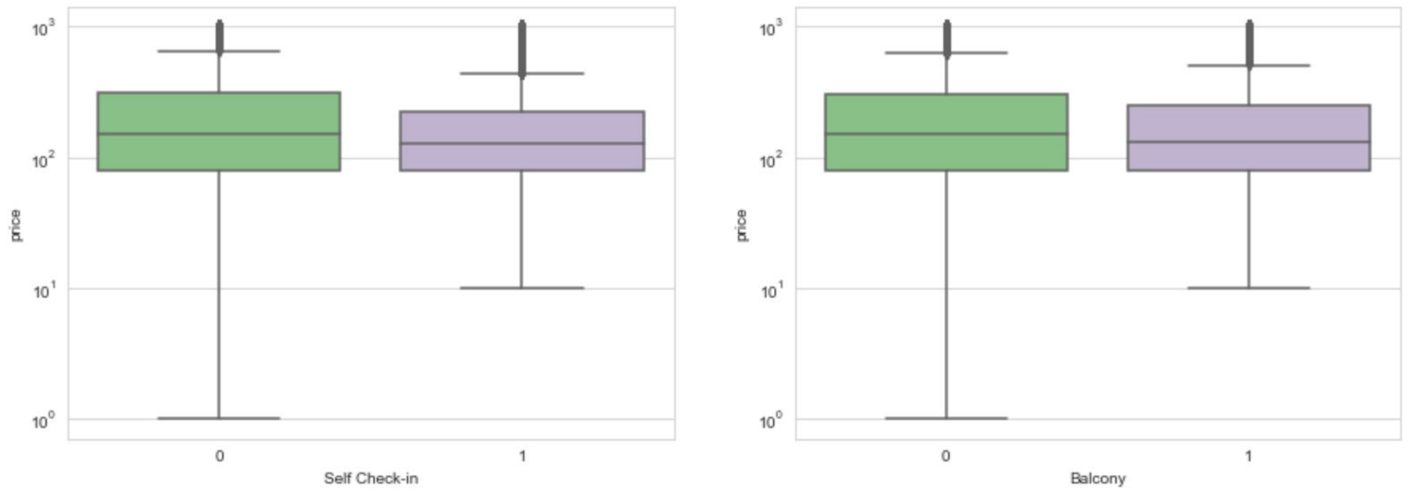




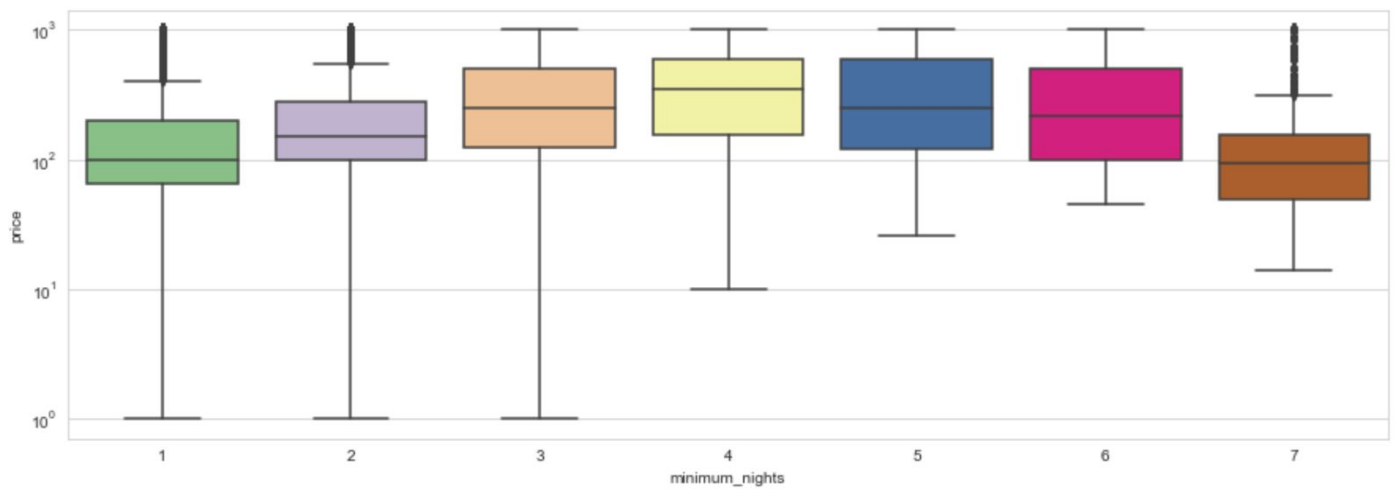


The next batch of features are the categories that were generated from the listings' amenities. Here, it makes it evident that Family-friendly places tend to be costlier than those that are not (or those that do not specify that they are). Perhaps these listings are able to accommodate more headcount or perhaps these tend to be in quieter neighborhoods and hence are generally more expensive.



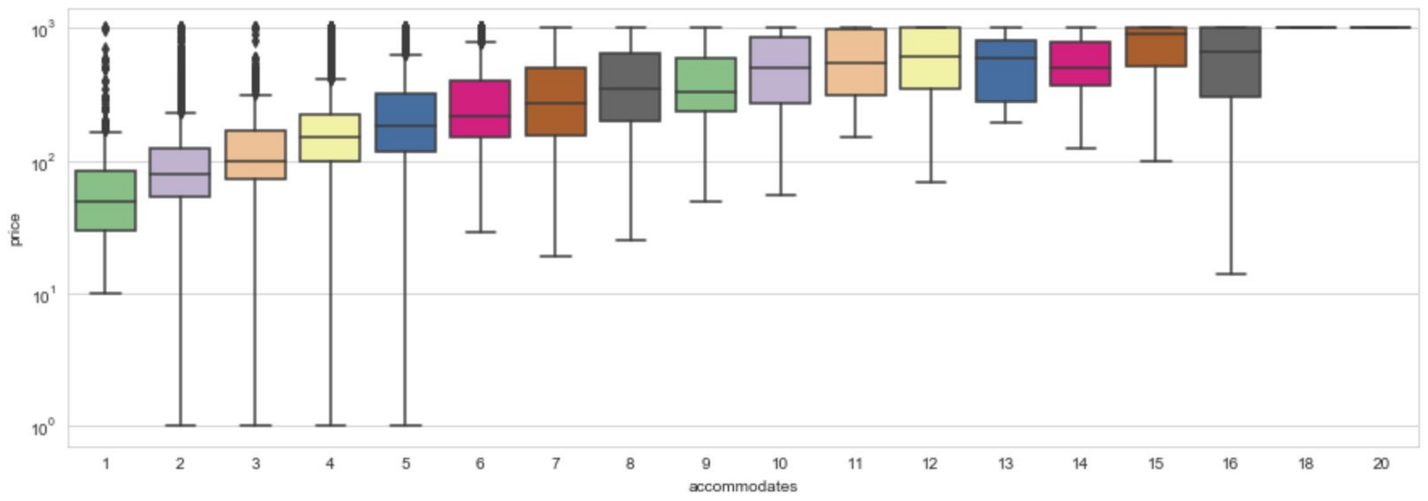


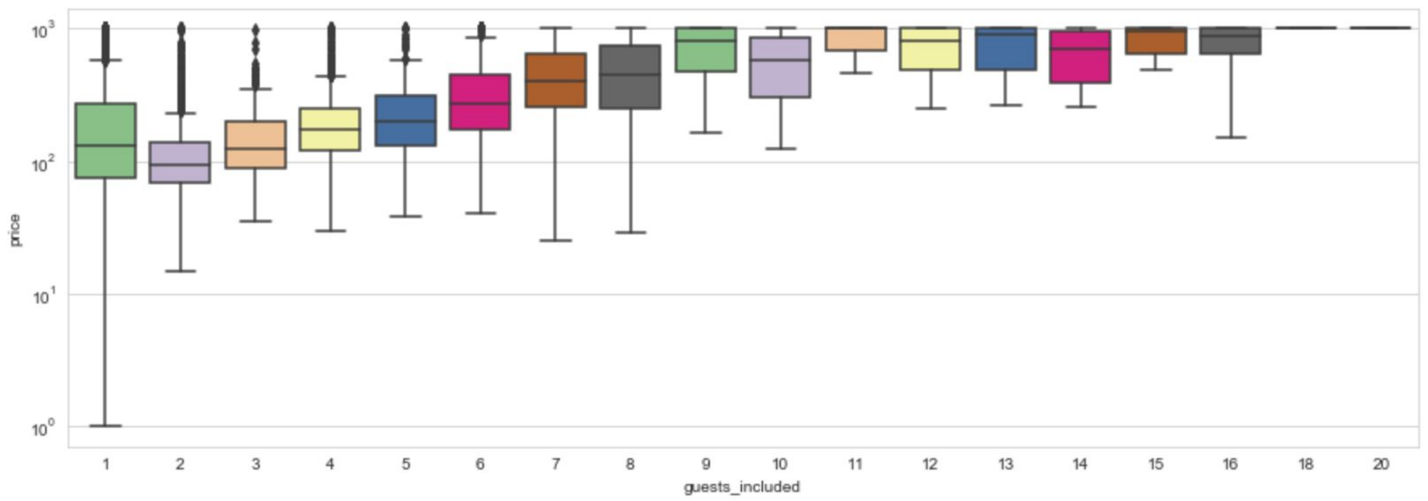
The minimum nights feature also shows distinction per classification. It's quite intuitive to assume that places that allow shorter-term stays would be pricier (like in the terms of lease in most apartments), but somehow this is not the case here. The pricing in terms of this feature is more varied.



The accommodates and the guests_included features both give us a sense of how large a listing is. For the former, a smaller listing would only be able to accommodate fewer people and therefore the lower the number is, the cheaper. For the latter, if more headcount can be accommodated without extra charge, then that suggests these extra people are already *priced in* on the nightly rate, and are therefore more expensive.

These takeaways can be inferred from the plots below.





Inferential Statistics

There are a total of 25 features to our final df dataframe. Of these, we'll formally test each to find out which make good candidates for our model.

Popularity and Price

We've already discussed how number_of_reviews can be a good indicator of a listing's popularity. Now let's test its correlation with price. A reliable indicator of correlation for two variables is the Pearson coefficient. But using this approach will need to meet some assumptions.

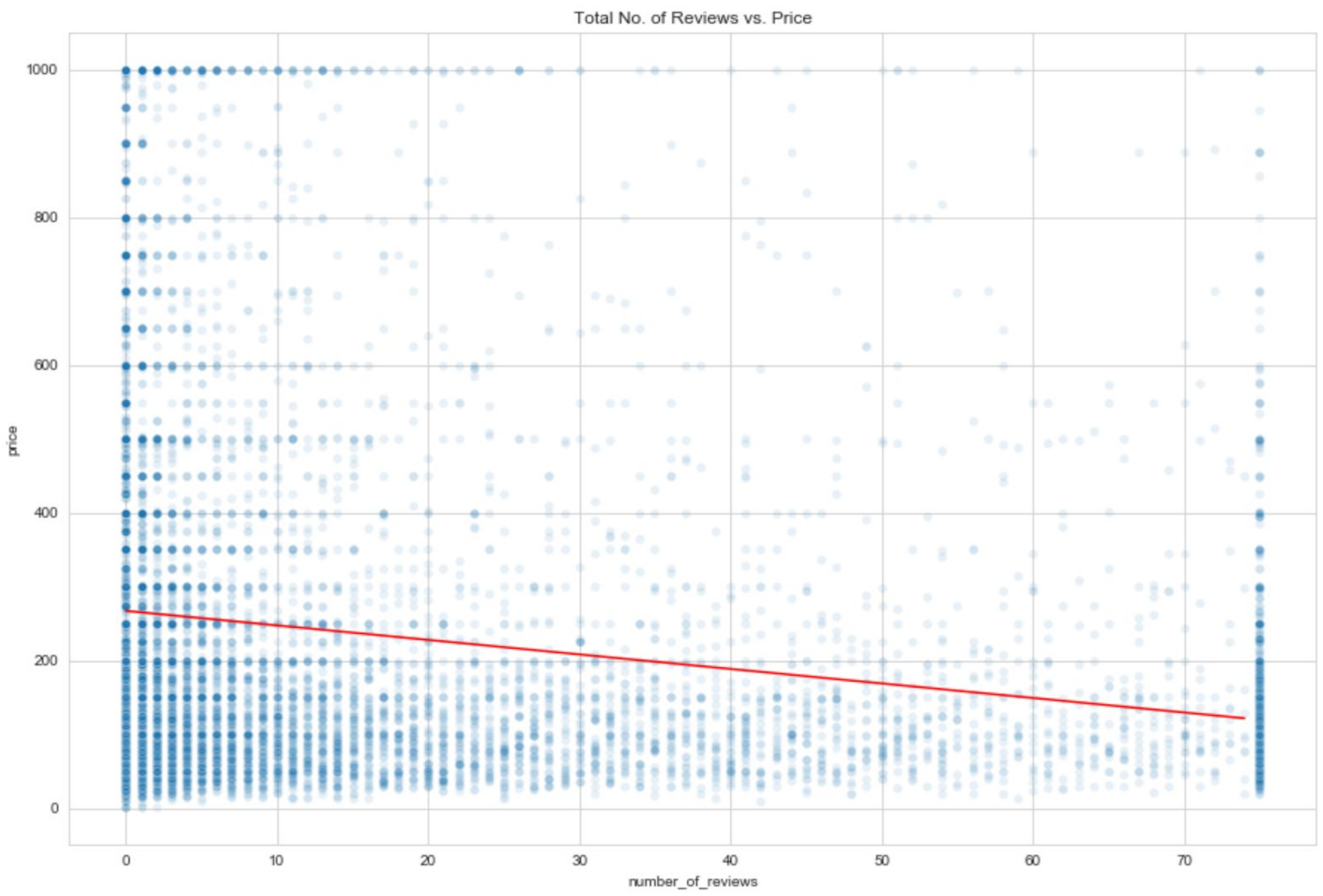
One of these assumptions is homoscedasticity, or if there is homogeneity of variances. We make use of *Levene's test* and set the critical value alpha to 0.05. The hypothesis statements are below:

- **Null Hypothesis**: There is homogeneity of variances between price and number_of_reviews
- **Alternative Hypothesis**: There is no homogeneity of variances between price and number_of_reviews

```
Levene W: 5321.392
P-Value: 0.000000
P-Value larger than alpha? False
Assumption of homoscedasticity IS violated (reject null hypothesis)
```

Since our p-value p is very close to zero, and thus much lower than our alpha, we cannot assume that the variances of the two variables are similar. We therefore cannot use Pearson coefficient to measure correlation. When it fails the variance homogeneity assumption, a good candidate would be the *Spearman rank*. This approach also happens to be a reliable measure for when we do not know whether the distribution is normal or not.

Below is the scatter plot for the price and number_of_reviews variables. First-degree polyfit function was used to produce the line of best fit. Their Spearman rank is also below. At -0.26, the two variables are negatively correlated (hence the negative slope), though the correlation is considered weak to moderate.



Spearman Rank

	price	number_of_reviews
price	1.000000	-0.261526
number_of_reviews	-0.261526	1.000000

Crimes and Price

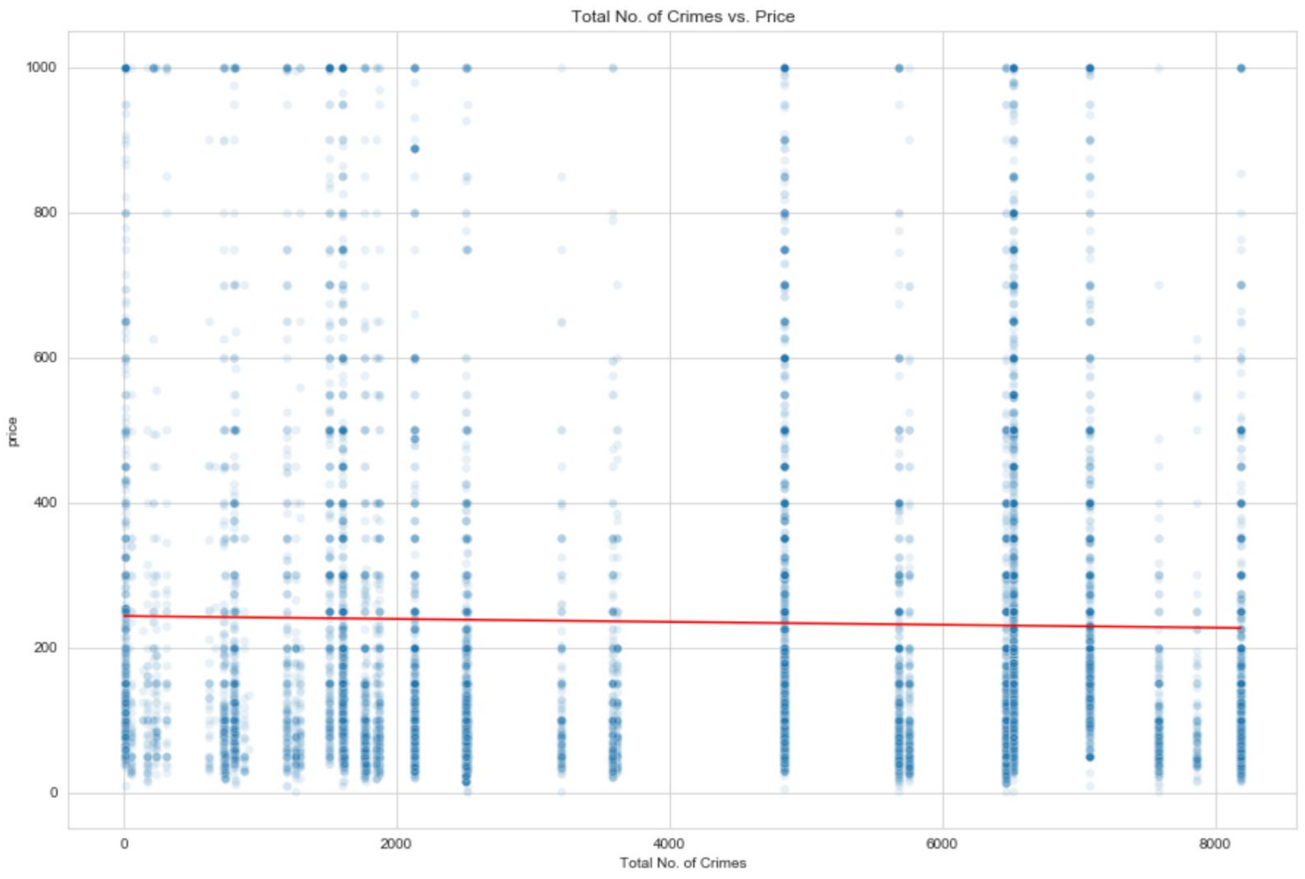
Let's look at the relationship between Total No. of Crimes and price to understand how crime data might affect Airbnb listings. Note that as we've done during *Data Wrangling*, the crimes tally is per zip code and *not* per listing. Therefore all the listings within a zip code have the same value for a given crime tally.

Again we perform Levene's test to see if there is homoscedasticity between the two.

- **Null Hypothesis**: There is homogeneity of variances between price and Total No. of Crimes
- **Alternative Hypothesis**: There is no homogeneity of variances between price and Total No. of Crimes

Once more, we see that the assumption is violated. We then resolve to use the Spearman rank to measure correlation. As the Total No. of Crimes and price plot shows, we see that there is almost no correlation between the two. A Spearman rank of 0.02 is very close to zero and indicates very weak correlation.

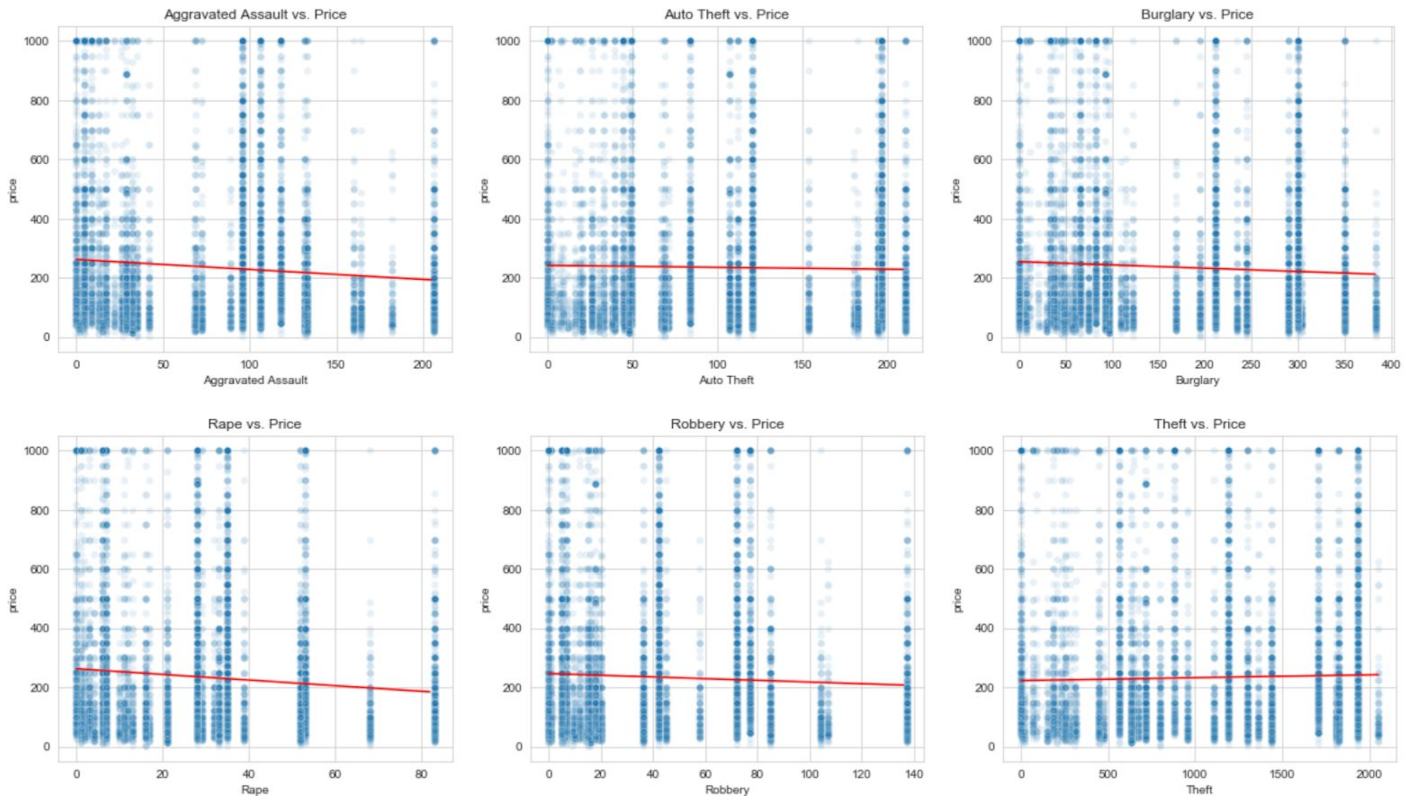
```
Levene W: 29318.802
P-Value: 0.000000
P-Value larger than alpha? False
Assumption of homoscedasticity IS violated (reject null hypothesis)
```



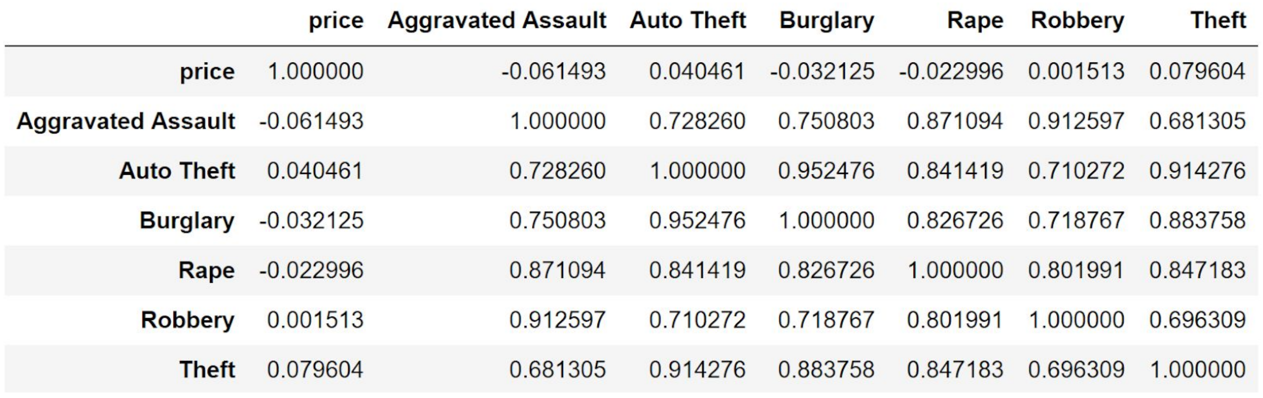
Spearman Rank

	price	Total No. of Crimes
price	1.0000	0.0216
Total No. of Crimes	0.0216	1.0000

In fact, the same is true when it comes to the tally of each major crime occurrence in Austin, TX. The Spearman matrix shows very weak correlation between each tally and price. What the data says, though, is that each major crime is somewhat related to another. The strongest correlation is between Aggravated Assault and Robbery, with a score of 0.91, suggesting that the presence of one in an area would highly likely exhibit the presence of the other.

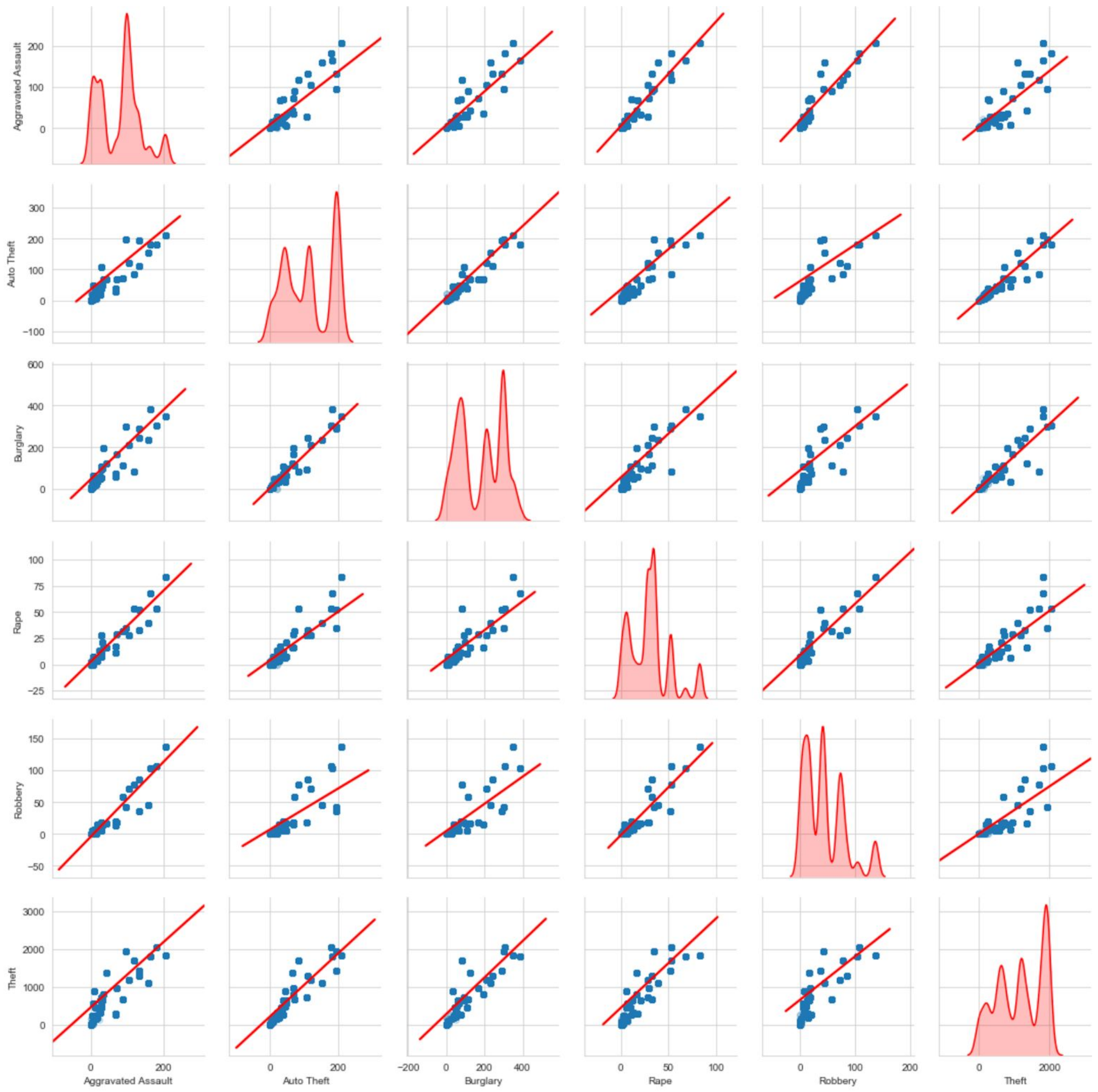


Spearman Matrix



	price	Aggravated Assault	Auto Theft	Burglary	Rape	Robbery	Theft
price	1.000000	-0.061493	0.040461	-0.032125	-0.022996	0.001513	0.079604
Aggravated Assault	-0.061493	1.000000	0.728260	0.750803	0.871094	0.912597	0.681305
Auto Theft	0.040461	0.728260	1.000000	0.952476	0.841419	0.710272	0.914276
Burglary	-0.032125	0.750803	0.952476	1.000000	0.826726	0.718767	0.883758
Rape	-0.022996	0.871094	0.841419	0.826726	1.000000	0.801991	0.847183
Robbery	0.001513	0.912597	0.710272	0.718767	0.801991	1.000000	0.696309
Theft	0.079604	0.681305	0.914276	0.883758	0.847183	0.696309	1.000000

To intuitively see how each major crime is correlated with another, the pair plot is shown below. The slope of each regression line and how each clusters are in step indicate positive correlation among the crime variables.



Location and Price

Locations are denoted in our `df` dataframe by `zipcode`. In order to find out what the appropriate significance test can be used for our categorical features, it's important to first understand the nature of our `price` function. A popular test to check if the distribution is normal is the *Shapiro-Wilk test*. However, the accuracy for its p-value wanes if the data contains 5,000 samples, as in the case of our dataset. So we'll opt to use the *Kolmogorov-Smirnov test* instead.

We want to be able to find out if `price` follows a normal distribution.

- **Null Hypothesis**: The `price` curve is normal
- **Alternative Hypothesis**: The `price` curve is NOT normal

```
KS-Statistic: 0.227
P-Value: 0.000000
P-Value larger than alpha? False
Price Curve does NOT appear Normal (reject null hypothesis)
```

Our p-value `p` is significantly smaller than our critical value of `0.05`, suggesting that it's is extremely unlikely we get a curve like ours if we assumed that our curve is usually normal. We cannot therefore assume normality and have to use nonparametric statistical tests.

Our goal is to identify if our categorizing of `price` by `zipcode` produce the same distribution. If so, then `zipcode` would not make a good feature for our model since it does not discriminate cheap listings from expensive listings. Our hypothesis statements are below.

- **Null Hypothesis**: The `zipcode` distributions are similar
- **Alternative Hypothesis**: The `zipcode` distributions are NOT similar

We'll use *Kruskal-Wallis test* because it's the nonparametric analog to ANOVA testing. This test is suited for variables with multiple discrete categories, as in the case for `zipcode`.

```
Zip Code
--------
Number of variables: 48
Kruskal-Wallis H: 1584.793
P-Value: 0.000000
P-Value larger than alpha? False
Distributions are NOT the same (reject null hypothesis)
```

The p-value is significantly low and so we reject our null hypothesis and conclude that the distributions for `zipcode` are not similar, making `zipcode` a good feature for our predictive model.

Other Categorical Features

The following categories listed below have values with `0` signifying False and `1` signifying True.

- host_is_superhost
- host_identity_verified
- Private Entry
- Family-friendly
- Pets OK
- Self Check-in
- Balcony

The hypothesis statements for these are below:

- **Null Hypothesis**: The distributions for the category are similar
- **Alternative Hypothesis**: The distributions for the category are NOT similar

Because of their nonparametric nature, and because there are only two outcomes, we'll be using *Mann-Whitney test* which is similar to Student's T-test but for distributions that are non-normal.

```
host_is_superhost
--------------------
Mann-Whitney U: 11397718.500
P-Value: 0.000000
P-Value larger than alpha? False
Distribution is NOT the same (reject H₀)

host_identity_verified
--------------------
Mann-Whitney U: 17540981.000
P-Value: 0.166199
P-Value larger than alpha? True
Distribution is the same (do NOT reject H₀)

Private Entry
--------------------
Mann-Whitney U: 13825506.500
P-Value: 0.000000
P-Value larger than alpha? False
Distribution is NOT the same (reject H₀)

Family-friendly
--------------------
Mann-Whitney U: 12680419.000
P-Value: 0.000000
P-Value larger than alpha? False
Distribution is NOT the same (reject H₀)

Pets OK
--------------------
Mann-Whitney U: 14355032.000
P-Value: 0.000050
P-Value larger than alpha? False
Distribution is NOT the same (reject H₀)

Self Check-in
--------------------
Mann-Whitney U: 14716557.000
P-Value: 0.000000
P-Value larger than alpha? False
Distribution is NOT the same (reject H₀)

Balcony
--------------------
Mann-Whitney U: 11234091.000
P-Value: 0.000339
P-Value larger than alpha? False
Distribution is NOT the same (reject H₀)
```

The next batch of features are also categorical but all of which have more than two values.

- host_verifications_count
- property_type (simple)
- room_type
- cancellation_policy
- accommodates
- guests_included
- extra_people
- minimum_nights
- Murde

The hypothesis statements for these are:

- **Null Hypothesis**: The distributions for the category are similar
- **Alternative Hypothesis**: The distributions for the category are NOT similar

Like with the zipcode feature, we'll use the *Kruskal-Wallis* approach to test the hypothesis for each category.

```
host_verifications_count
--------------------
Number of variables: 14
Kruskal-Wallis H: 181.691
P-Value: 0.000000
P-Value larger than alpha? False
Distribution is NOT the same (reject H_0)

property_type (simple)
--------------------
Number of variables: 3
Kruskal-Wallis H: 266.037
P-Value: 0.000000
P-Value larger than alpha? False
Distribution is NOT the same (reject H_0)

room_type
--------------------
Number of variables: 3
Kruskal-Wallis H: 3318.730
P-Value: 0.000000
P-Value larger than alpha? False
Distribution is NOT the same (reject H_0)

cancellation_policy
--------------------
Number of variables: 5
Kruskal-Wallis H: 591.027
P-Value: 0.000000
P-Value larger than alpha? False
Distribution is NOT the same (reject H_0)

accommodates
--------------------
Number of variables: 18
Kruskal-Wallis H: 4821.669
P-Value: 0.000000
P-Value larger than alpha? False
Distribution is NOT the same (reject H_0)

guests_included
--------------------
Number of variables: 18
Kruskal-Wallis H: 1776.368
P-Value: 0.000000
P-Value larger than alpha? False
Distribution is NOT the same (reject H_0)

extra_people
--------------------
Number of variables: 75
Kruskal-Wallis H: 1555.390
P-Value: 0.000000
P-Value larger than alpha? False
Distribution is NOT the same (reject H_0)

minimum_nights
--------------------
Number of variables: 7
Kruskal-Wallis H: 1149.105
P-Value: 0.000000
P-Value larger than alpha? False
Distribution is NOT the same (reject H_0)

Murder
--------------------
Number of variables: 6
Kruskal-Wallis H: 343.466
P-Value: 0.000000
P-Value larger than alpha? False
Distribution is NOT the same (reject H_0)
```

We've now explored all the features of our df dataframe. To summarize, the following variables have proved to be meaningful for our model. The continuous features have slight to moderate correlation with price and the discrete features have constituents that have distributions that are distinct from one another.

1. number_of_reviews
2. zipcode
3. host_is_superhost
4. Private Entry
5. Family-friendly
6. Pets OK
7. Self Check-in
8. Balcony
9. host_verifications_count
10. property_type (simple)
11. room_type
12. cancellation_policy
13. accommodates
14. guests_included
15. extra_people
16. minimum_nights
17. Murder

The following features were identified as variables that may not be helpful in distinguishing the price of a listing. The continuous features have almost no correlation with price and the discrete features have similar distributions.

1. Total No. of Crimes
2. Aggravated Assault
3. Auto Theft
4. Burglary
5. Rape
6. Robbery
7. Theft
8. host_identity_verified

We have now closely inspected every variable of our df dataframe. Now we have a better understanding of how well each feature may predict the nightly rate of an Airbnb listing in Austin, TX. We can now build our predictive model upon these and be able to tackle our problem statement.