MGS 616: Predictive Analytics

Homework 1 – Visualizations

Khansa Khanam Umar Sultan	*khansakh*	*50560596*

(Link to Google Colab: MGS 616_HW1_Group23)

The Goal of the Assignment:

The data set for this assignment is made available by Airbnb. It contains data about listings in the Boston, MA area. There are 3,583 listings in the data set. (The number of columns is reduced significantly from the original data set). Each row represents a single listing and contains information about the host of property and the property's characteristics. The goal is to visualize the input variables to get familiar with the data set.

Step 1: Importing the required libraries

```
# importing the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Step 2: Data Cleaning

Initial shape of the dataset

```
(3583, 33)
```

Imputing missing values with mean and mode values

```
mean_reviews_per_month = housing['reviews_per_month'].mean()
mean_review_scores_rating = housing['review_scores_rating'].mean()
mean_review_scores_accuracy =
housing['review_scores_accuracy'].mean()
mean_review_scores_cleanliness =
housing['review_scores_cleanliness'].mean()
mean_review_scores_checkin =
housing['review_scores_checkin'].mean()
```

```
mean_review_scores_communication =
housing['review_scores_communication'].mean()
mean_review_scores_location =
housing['review_scores_location'].mean()
mean_review_scores_value = housing['review_scores_value'].mean()

housing['reviews_per_month'].fillna(mean_reviews_per_month,
inplace=True)
housing['review_scores_value'].fillna(mean_review_scores_value,
inplace=True)
housing['review_scores_rating'].fillna(mean_review_scores_rating,
inplace=True)
housing['review_scores_accuracy'].fillna(mean_review_scores_accura
cy, inplace=True)
housing['review_scores_cleanliness'].fillna(mean_review_scores_cle
anliness, inplace=True)
housing['review_scores_checkin'].fillna(mean_review_scores_checkin
, inplace=True)
housing['review_scores_communication'].fillna(mean_review_scores_c
ommunication, inplace=True)
housing['review_scores_location'].fillna(mean_review_scores_locati
on, inplace=True)
```

```
mode_host_response_rate = housing['host_response_rate'].mode()
housing['host_response_rate'].fillna(mode_host_response_rate,
inplace=True)
```

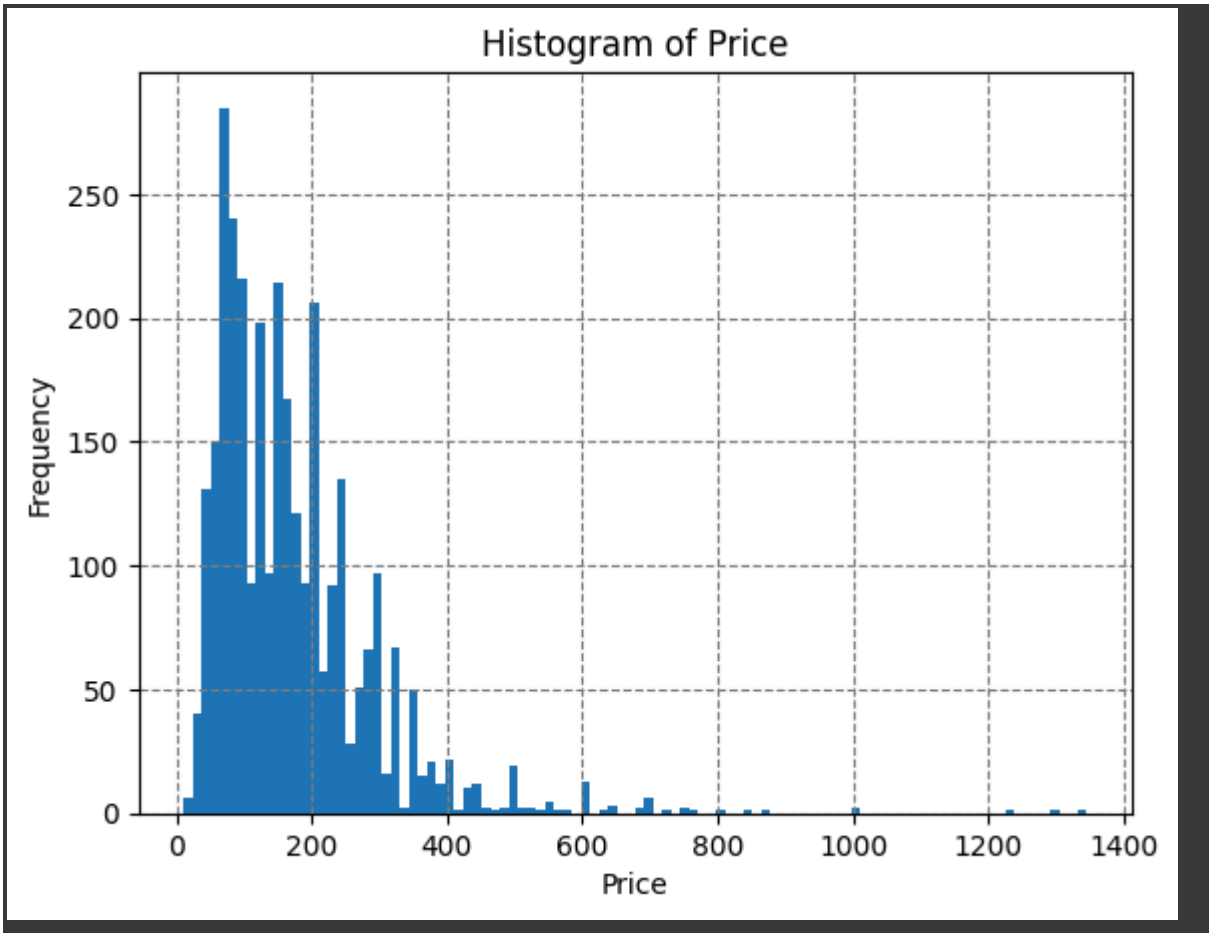
Dropping rows with missing values

```
housing['host_response_time'].dropna(inplace = True)
housing['city'].dropna(inplace = True)
housing['property_type'].dropna(inplace = True)
housing['bathrooms'].dropna(inplace = True)
housing['bedrooms'].dropna(inplace = True)
housing['beds'].dropna(inplace = True)
housing.dropna(inplace = True)
```

Final shape after cleaning processes

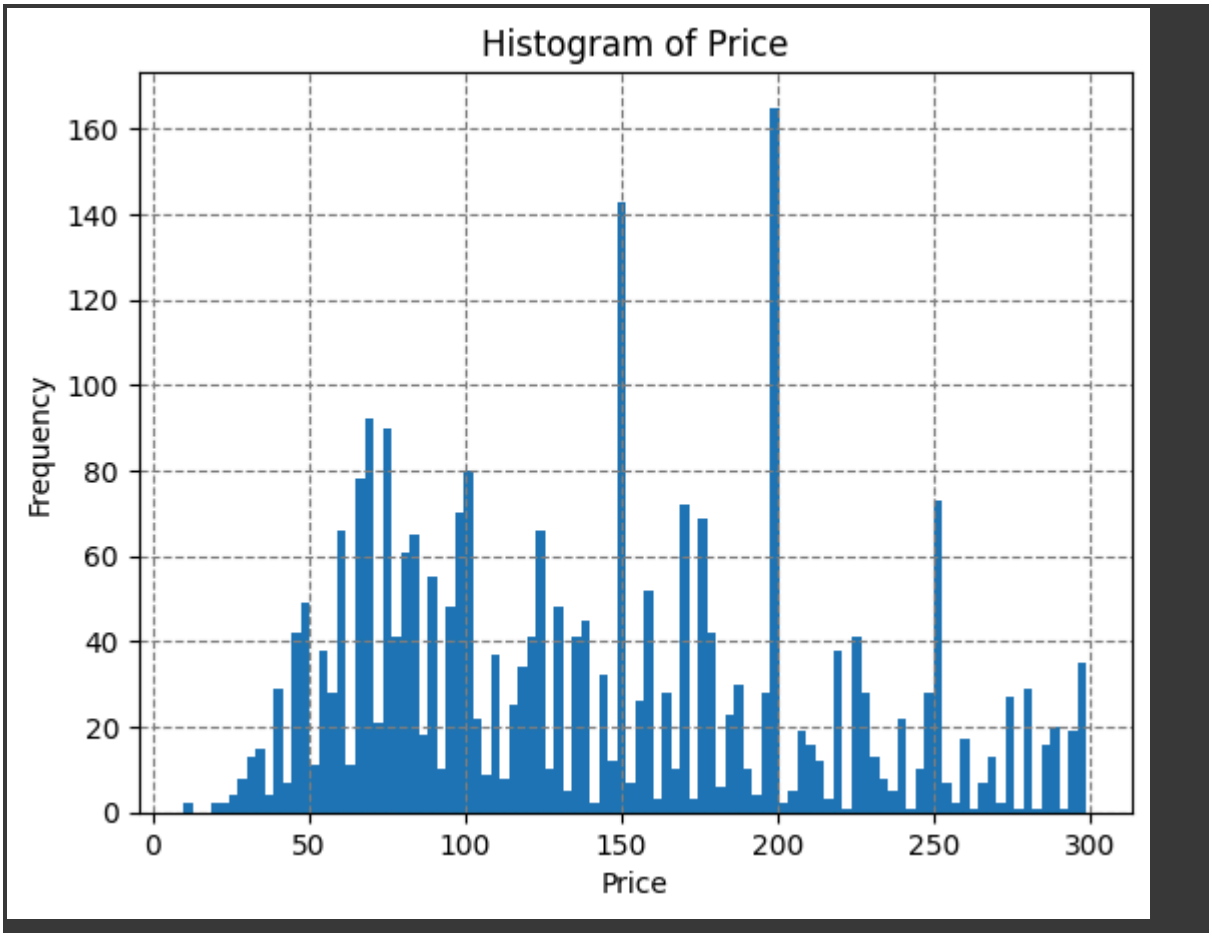
```
(3084, 34)
```

1. **Create a histogram for the PRICE variable:**



A close look at the more crowded region

```
Text(0.5, 1.0, 'Histogram of Price')
```



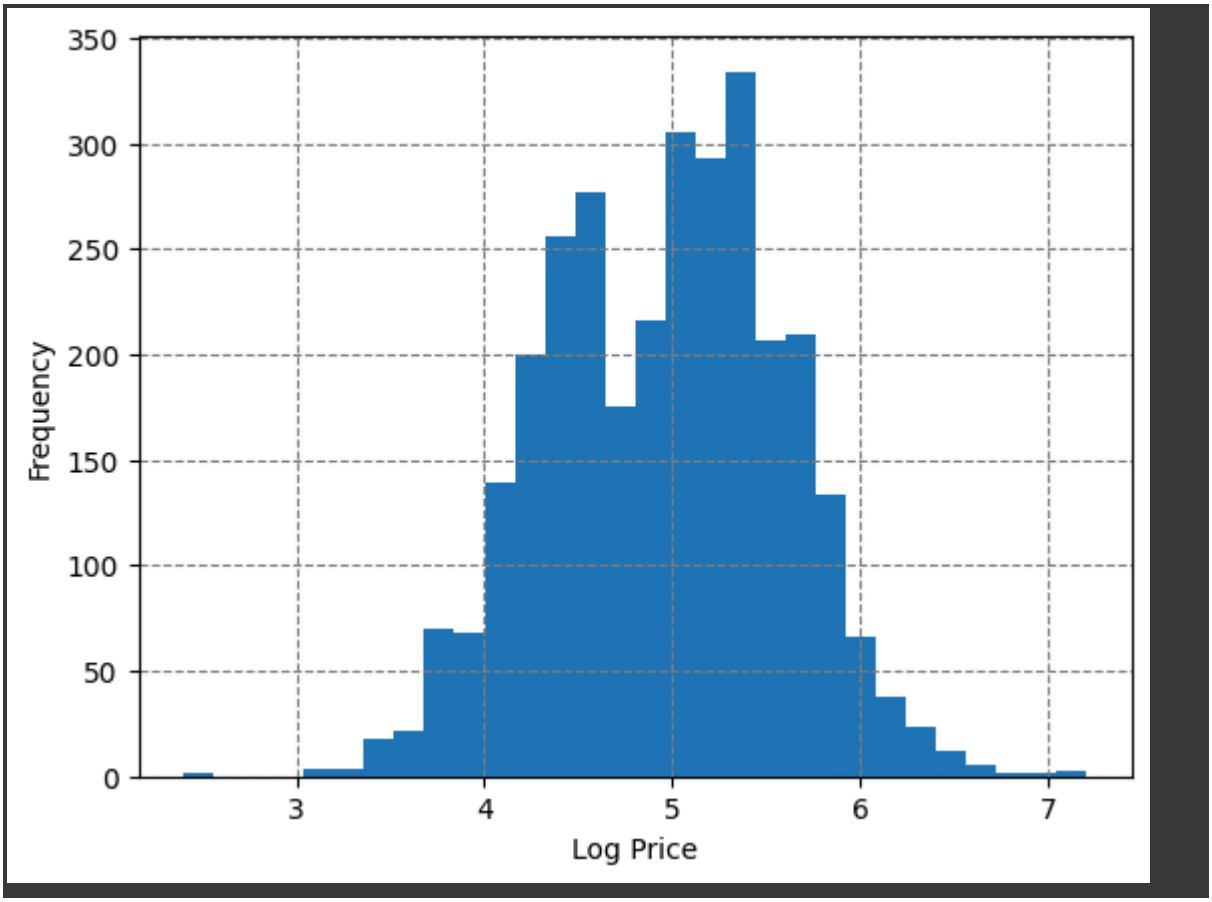
A higher number of properties are either prices at 150K or 200K. Since most of the properties are priced in the lower range (150K-200K), and there are fewer properties at higher prices, the right tail is long. This skewness can cause issues in statistical models that assume normality (such as linear regression) or models sensitive to outliers.

Logarithmic Transformation:

This is the most common transformation for right-skewed data. It compresses the higher end of the distribution and can make the data closer to normal.

If we're simply planning to use this variable in a model that assumes normality or where skewness affects performance, applying a transformation would be beneficial. However, if our goal is only visualization or using models robust to skewness, we might leave the data as is.

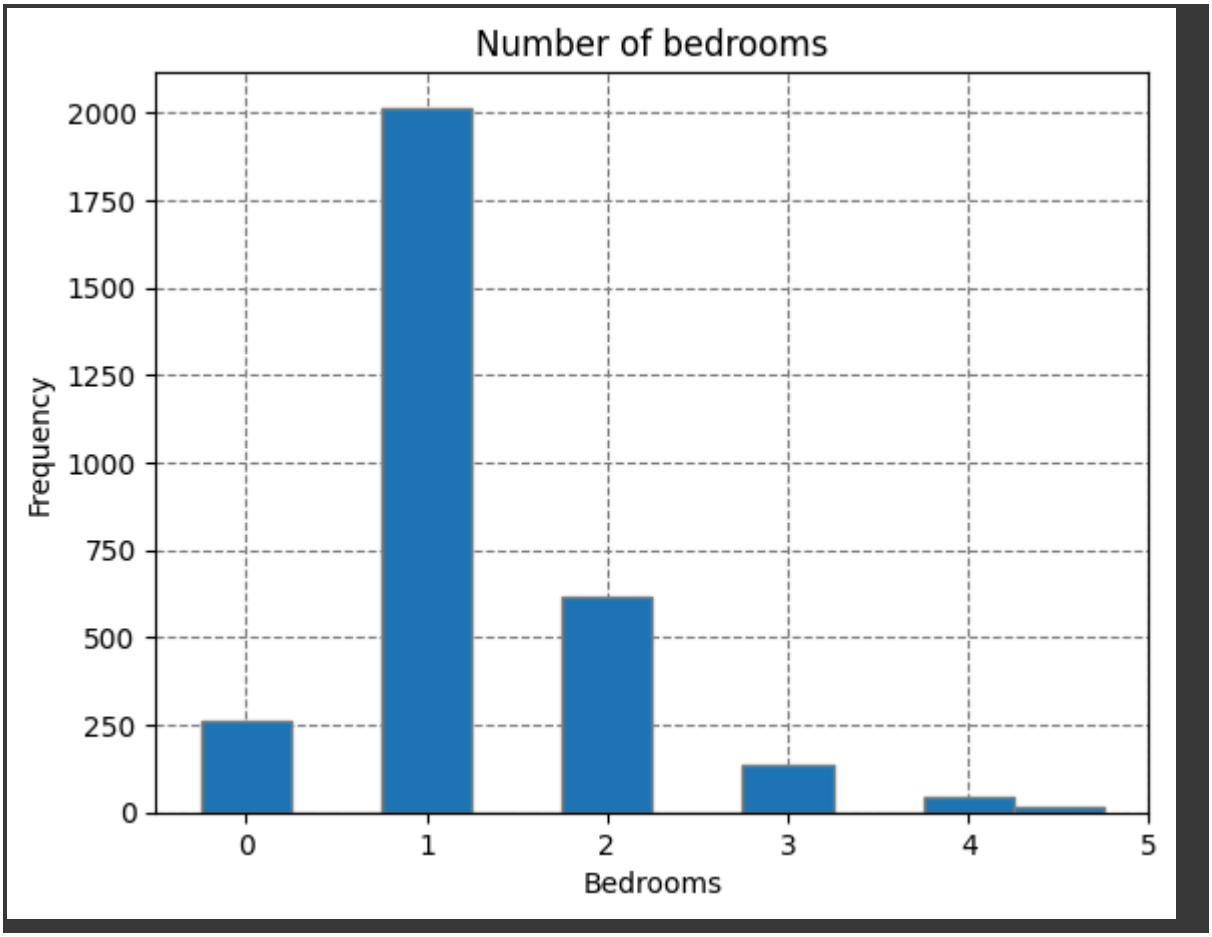
```
<bound method Axes.set of <Axes: xlabel='Log Price', ylabel='Frequency'>>
```



2. Create a histogram for the BEDROOMS variable:

```
ax = housing['bedrooms'].hist(edgecolor = 'grey', grid = True, align =
"left")
ax.grid(which = 'major', linestyle = '--', color = 'grey')
ax.set_axisbelow(True)
ax.set_xlabel("Bedrooms")
ax.set_ylabel("Frequency")
ax.set_title("Number of bedrooms")
```

```
Text(0.5, 1.0, 'Number of bedrooms')
```

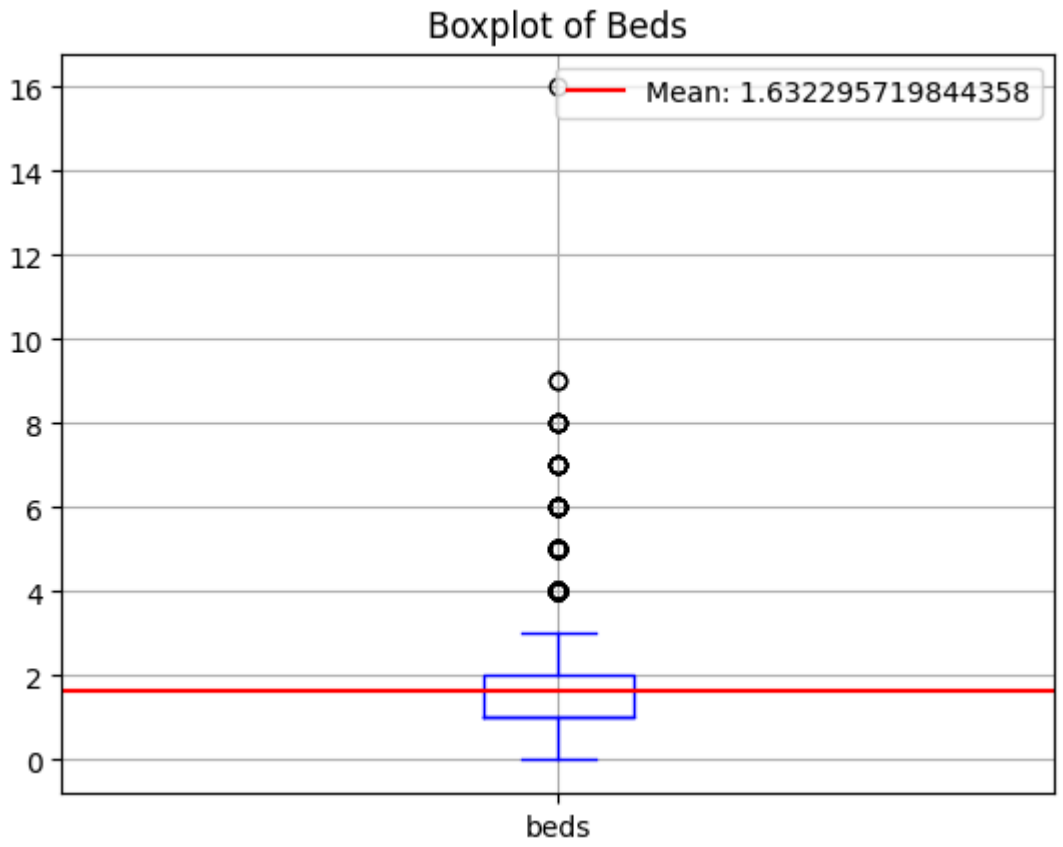


A transformation is required for the data type of the column. It would be more appropriate as a categorical variable due to the limited, defined set of possible values. However, addressing skewness is unnecessary since the variable is discrete.

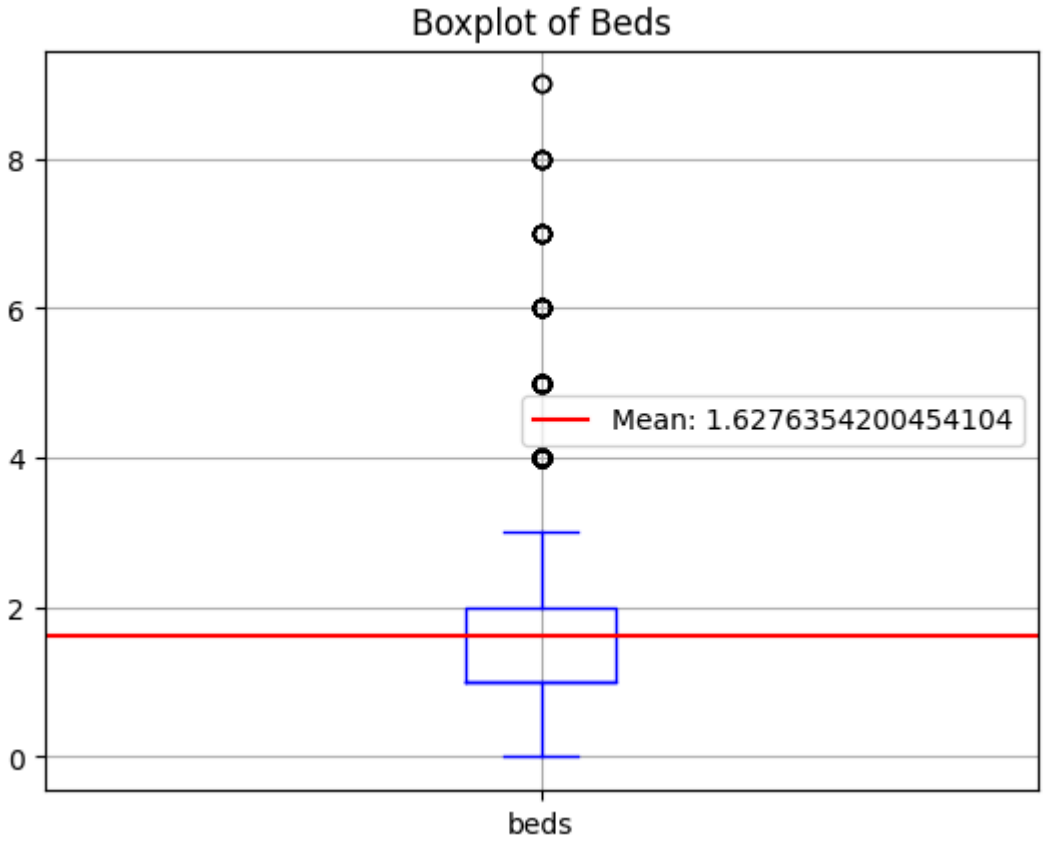
Inference drawn from the graph:

It appears that the majority of properties have 1 bedroom, followed by those with 2 bedrooms. Properties with 4 or 5 bedrooms are quite uncommon!

3. Create a boxplot for the BEDS variable:



A close look at the more crowded region



```
print(f"Min whisker: {housing['beds'].quantile(0.25)}")

print(f"Max whisker: {housing['beds'].quantile(0.75)}")

print(f"Median: {housing['beds'].median()}")

print(f"Mean: {mean_beds}")

print(f"Farthest outlier: {housing['beds'].max()}")
```

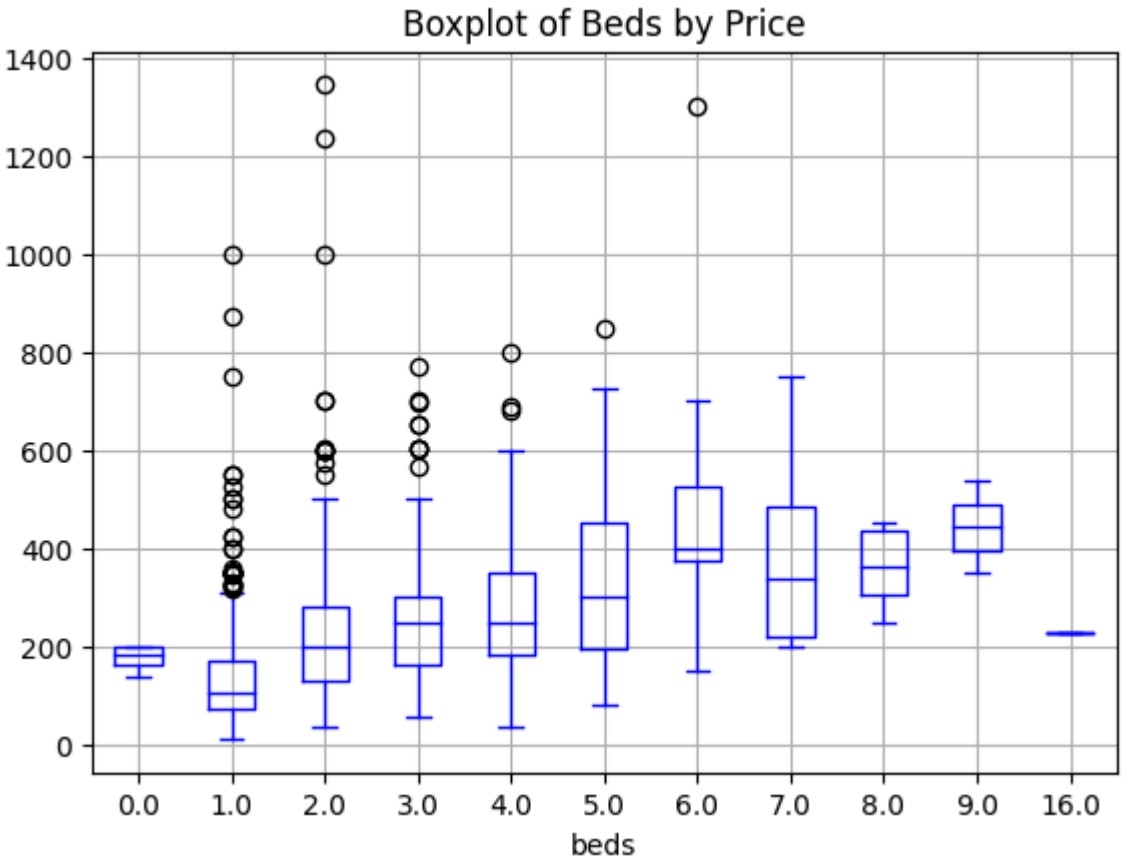
Min whisker: 1.0
Max whisker: 2.0
Median: 1.0
Mean: 1.6276354200454104
Farthest outlier: 16.0

Key Insights from the Graph:

The minimum whisker, maximum whisker, median, and mean values of the beds column are displayed above, along with the value of the farthest outlier.

There are outliers present in the data, particularly concentrated around the values of 4, 5, 6, 7, and 8 beds. The farthest outlier is observed at 16 beds.

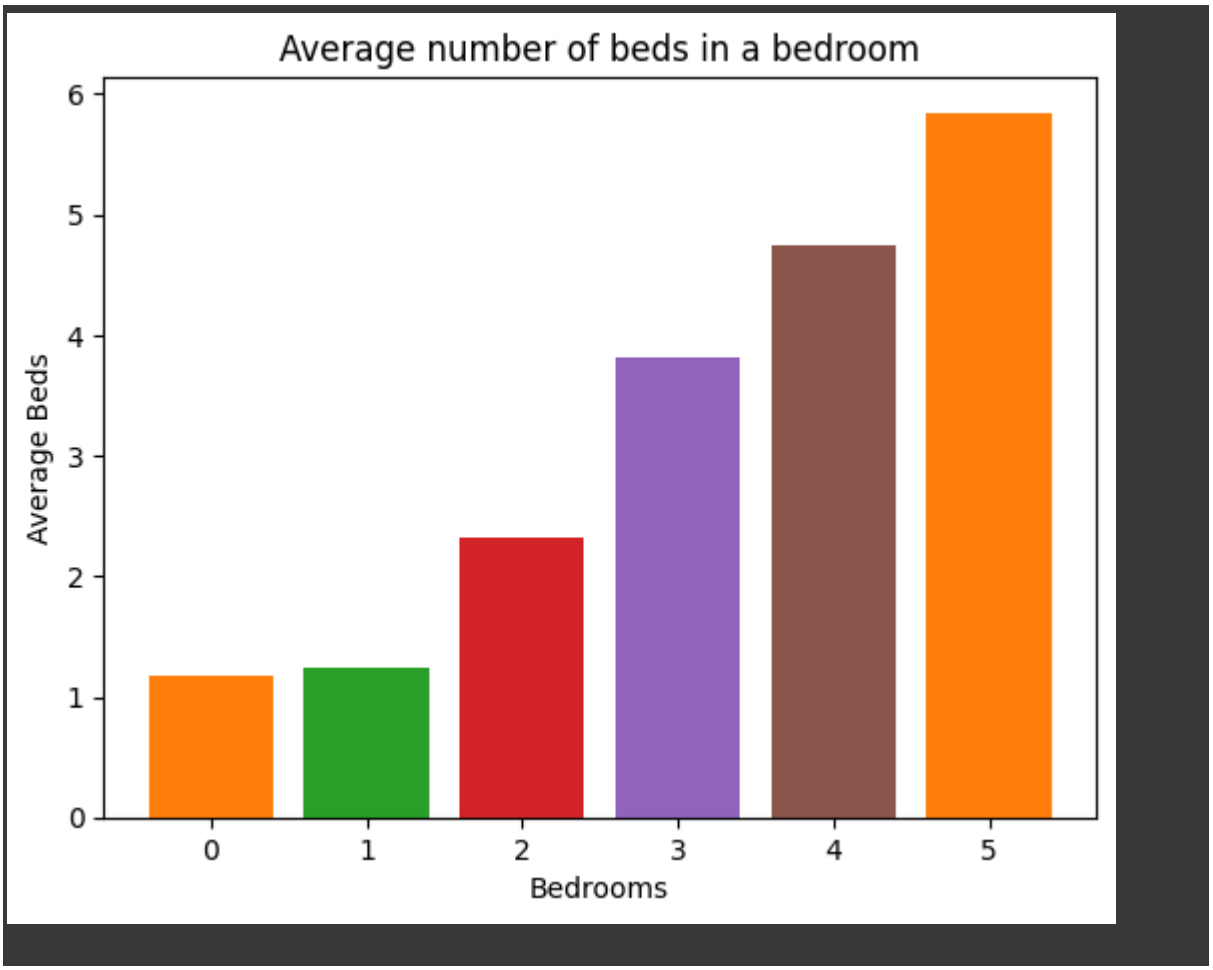
4. Create a boxplot for BEDS (as X) and PRICE (as Y):



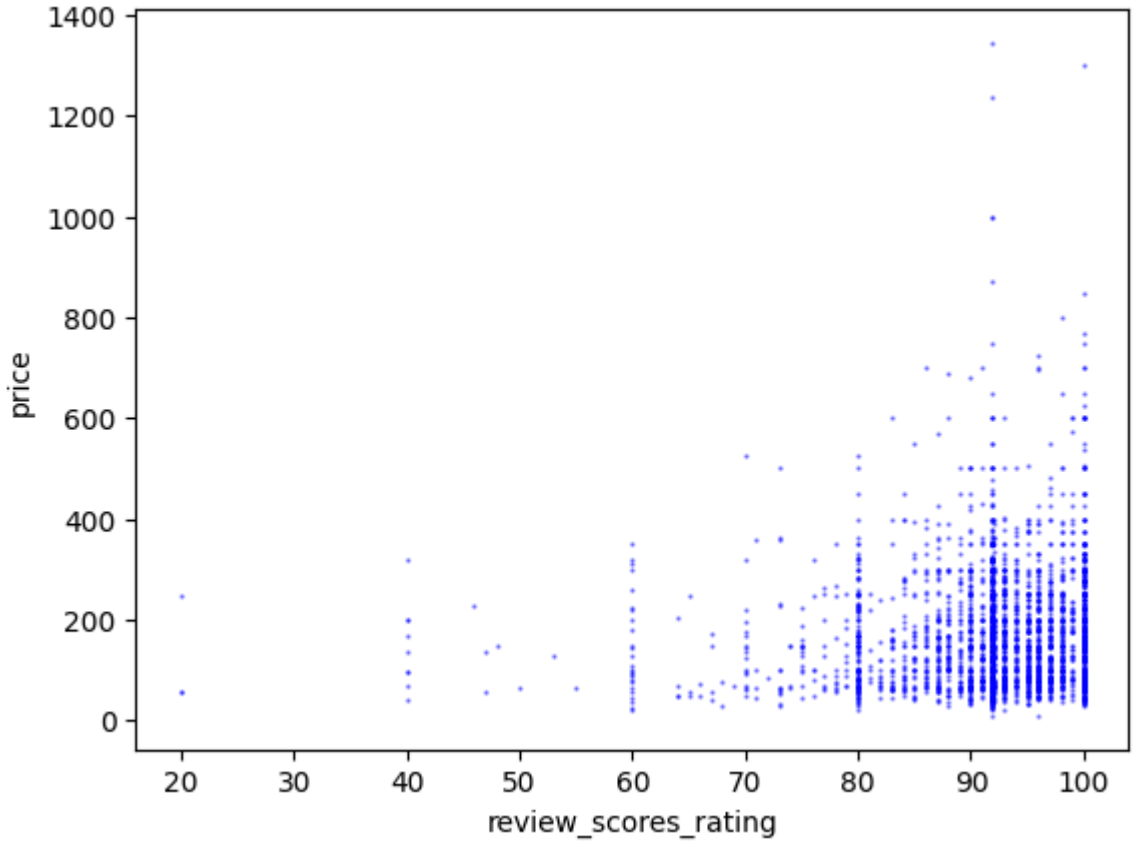
Key Insights from the Graph:

1. The median price generally increases with the number of beds, but there is little to no price difference between properties with 3 and 4 beds. This is likely because most 3-bed properties are in 2-bedroom spaces, and many 4-bed properties are also in 2-bedroom spaces, leading to similar median prices.
2. Properties with 7 or 8 beds show comparatively lower prices, as these beds are often spread across 3 or 4 bedrooms, contributing to the reduced price.
3. Although properties with 9 beds would typically be expected to have higher prices, they are shared across 3 bedrooms and can accommodate up to 10 people. As a result, their price is similar to those with 6 beds.
4. Properties with 16 beds have unusually low prices, as all 16 beds are located in a single bedroom.
5. Properties with 1 bed priced over $300 show a significant number of outliers compared to other properties. However, most of these high-priced listings are entire apartment rentals.

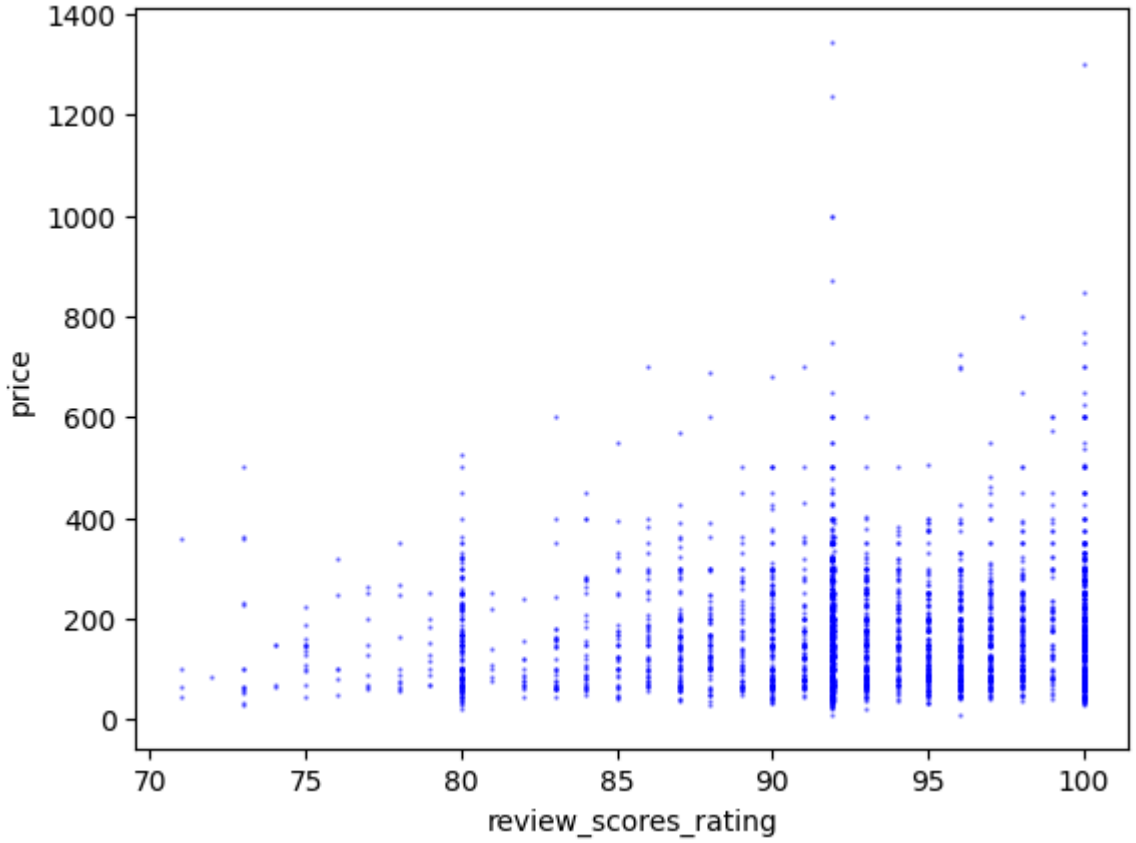
6. It is very rare for properties to exceed a price of $1,000.



5. <u>Create a scatterplot between PRICE (as Y) and REVIEW_ SCORES_RATING (as X)</u>



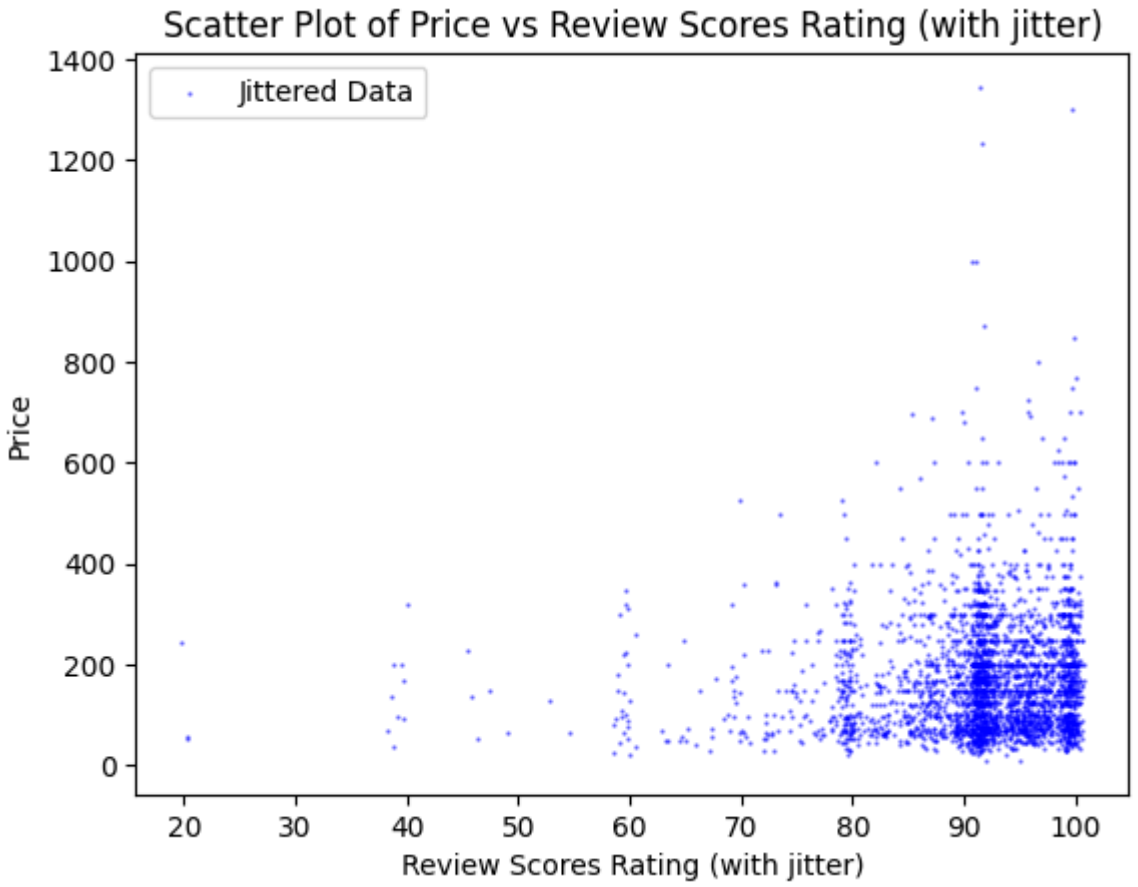
A close look at the more crowded region



Insights from the Graph:

1. Properties with higher review_scores_rating tend to have higher prices.
2. Properties with lower ratings generally do not command high prices.
3. A significant number of properties are rated between 70 and 100, with those rated between 92 and 100 having notably higher prices.
4. The higher ratings are often associated with specific property characteristics, such as property_type and room_type. Additionally, these properties usually score well across multiple metrics, including review_scores_accuracy, review_scores_cleanliness, review_scores_checkin, review_scores_communication, review_scores_location, and review_scores_value.

Scatter Plot with Jitters



The patter for higher rating corresponding to a higher price is now more obvious with jitters in the graph.