DATA 3300

Final Project - Unit 1: Data Preparation

Final Project Description

The Final Project in this course is broken into three units, corresponding to the three units in the course. By the end of the course, each student will have completed a comprehensive final project on data preparation, data understanding, and data modeling. The final portion of the project will include an executive summary on the comprehensive final project you will have completed.

Introduction

For this final project, we will take on the role of consultants for Aggie Investments, a Real Estate Investment Firm. In recent years, there has been a significant trend among investment firms to acquire properties for use as rental assets. While various geographies have been proposed, our focus is to assess the opportunities within a specific, rapidly growing market: Nashville, Tennessee.

Our task is to analyze a provided dataset containing information on current Airbnb listings in the Nashville area. The objective is to explore the data comprehensively and provide informed recommendations to Aggie Investments regarding the potential of entering this market, the types of listings they should acquire, and how they should manage those listings. The project will involve data preparation, exploration, and the application of unsupervised machine learning models to uncover deeper insights and patterns within the data. These findings will guide our final recommendations to the firm.

Part 1: Data Types

1 - Import the data3300_airbnb_data_raw_nashville.csv dataset into Python, explore the data to ensure we understand the data types that are present within the data.

REMEMBER THE CODE CHEAT SHEET!

In [117…
```
# replace with code to import the libraries and packages required to import data, manipulate dataframes, and produce
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [136…
```
# replace with code to import dataset
# replace with code to change display_option to display max columns in the dataframe
df = pd.read_csv("data3300_airbnb_data_raw_nashville.csv", dtype={'id':'int64'})
pd.set_option('display.max_columns', None)
df.info()
# df.head()
# df[df['id'] % 1 != 0]
# id is converting to a float, some weird issue wizardy but it works if I just convert it to an int it fixes it
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8808 entries, 0 to 8807
Data columns (total 30 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              8808 non-null   int64  
 1   name                            8808 non-null   object 
 2   host_id                         8808 non-null   int64  
 3   host_name                       8808 non-null   object 
 4   host_since                      8808 non-null   object 
 5   host_is_superhost               8558 non-null   object 
 6   calculated_host_listings_count  8808 non-null   int64  
 7   host_has_profile_pic            8808 non-null   object 
 8   host_identity_verified          8808 non-null   object 
 9   host_listings_count             8808 non-null   int64  
 10  neighbourhood_cleansed          8808 non-null   object 
 11  latitude                        8808 non-null   float64
 12  longitude                       8808 non-null   float64
 13  availability_365                8808 non-null   int64  
 14  minimum_nights                  8808 non-null   int64  
 15  room_type                       8808 non-null   object 
 16  accommodates                    8808 non-null   int64  
 17  bathrooms_text                  8807 non-null   object 
 18  bedrooms                        8647 non-null   float64
 19  beds                            6588 non-null   float64
 20  price                           6589 non-null   object 
 21  number_of_reviews               8808 non-null   int64  
 22  reviews_per_month               7889 non-null   float64
 23  review_scores_rating            7889 non-null   float64
 24  review_scores_accuracy          7889 non-null   float64
 25  review_scores_cleanliness       7889 non-null   float64
 26  review_scores_checkin           7889 non-null   float64
 27  review_scores_communication     7889 non-null   float64
 28  review_scores_location          7889 non-null   float64
 29  review_scores_value             7889 non-null   float64
dtypes: float64(12), int64(8), object(10)
memory usage: 2.0+ MB
```

In [137…

```
# replace with code to preview the df.
df.head()
```

Out[137…

	id	name	host_id	host_name	host_since	host_is_superhost	calculated_host_listings_count	host_has_profile_p
0	6422	Nashville Charm	12172	Michele	4/3/09	f	1	
1	39870	Close to Vanderbilt 2	171184	Evelyn	7/18/10	t	1	
2	59576	Large Main Suite near Lake *ladies only NS plz	812128	Patricia And John	7/12/11	t	9	
3	72906	Vandy/Belmont/10 mins to Broadway - Sunny 800 ...	176117	Richard	7/21/10	t	1	
4	258817	ButterflyRoom-queen room, private bath	22296	Diana	6/19/09	t	6	

Variable And Data Types

2 - For each of variables listed below, identify both the data type and the variable type.

The field `id` has already been filled in to provide an example.

""""

ID is technically incorrect in the dataset but we have fixed that to an int on ingestion

""""

- **id**
  - **Data Type:** int. this should be a int not a float as was originally specified
  - **Variable Type:** Discrete numerical
- **host_since**

  - **Data Type:** object -> int
  - **Variable Type:** Date, categorical date -> discrete numerical
- **host_is_superhost**
  - **Data Type:** object -> could be changed to an int if we want 0/1 flags
  - **Variable Type:** boolean, dichotomous, categorical
- **availability_365**
  - **Data Type:** int
  - **Variable Type:** discrete numerical
- **accommodates**
  - **Data Type:** int
  - **Variable Type:** discrete numerical
- **price**
  - **Data Type:** object -> float (want to remove $$ sign)
  - **Variable Type:** continous numerical (dollars)
- **reviews_per_month**
  - **Data Type:** float64
  - **Variable Type:** continous numerical

Additional Questions to Answer:

- 3. What is the primary key in our dataset? What is the function of the primary key?
  - Answer: id -> to provide a unique identifier for each record in the dataset
- 4. What is the difference between a continuous and discrete variable? List examples of each in the dataset.
  - Answer: continous measures are generally measured, heigh, weight, temp are all examples. discrete measures are countin numbers.

  continous -> latitude, longitude, and price. discrete -> id, host_id, minimum_nights, bedroom, etc..
- 5. What types of variables are considered quantitative (Numerical)? List examples in the dataset.
  - Answer: Can only take specific values, and are measured usually.

  latitude, lonitude, price, beds, etc...

6 - Create a Bar chart of a qualitative variable where the descriptive stat displayed the is count. What does this show us?

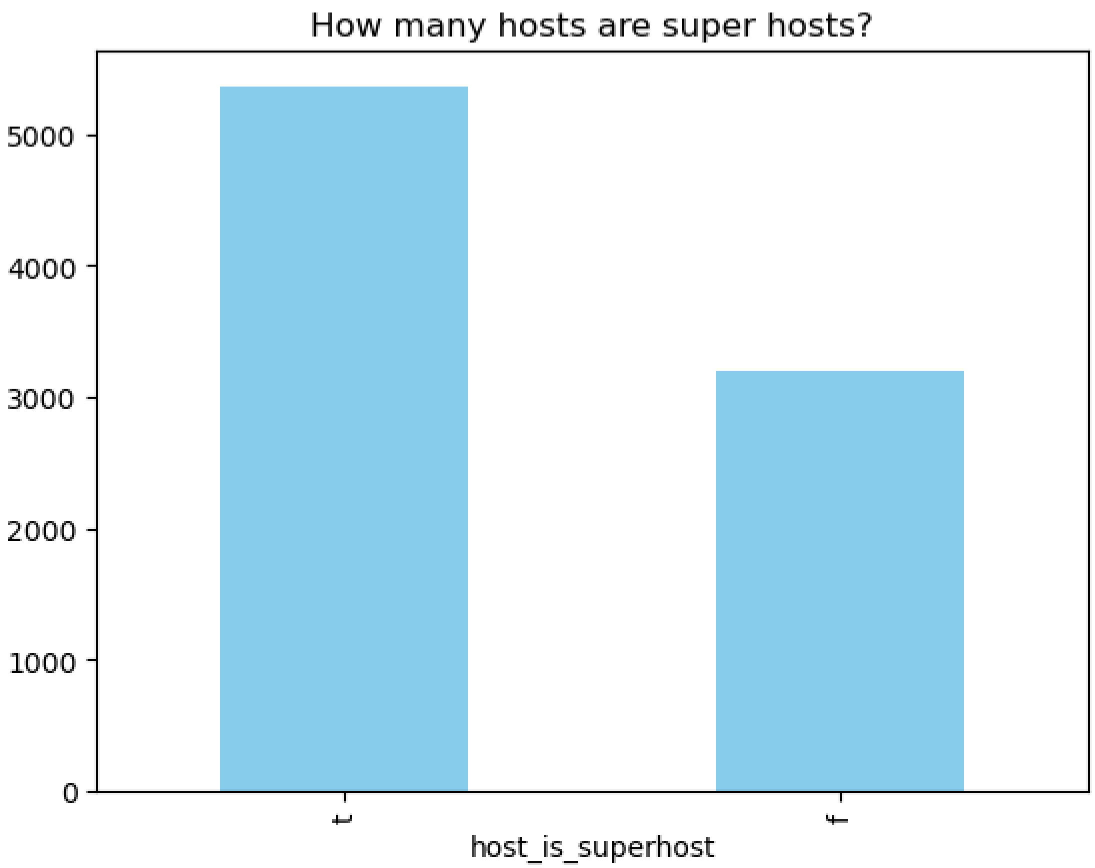
In [138…

```
# replace with code to create a bar chart
count_data = df['host_is_superhost'].value_counts()
count_data.plot(kind='bar', color='skyblue', title="How many hosts are super hosts?")

# how many hosts are super hosts
```

Out[138…

```
<Axes: title={'center': 'How many hosts are super hosts?'}, xlabel='host_is_superhost'>
```



How many hosts are super hosts

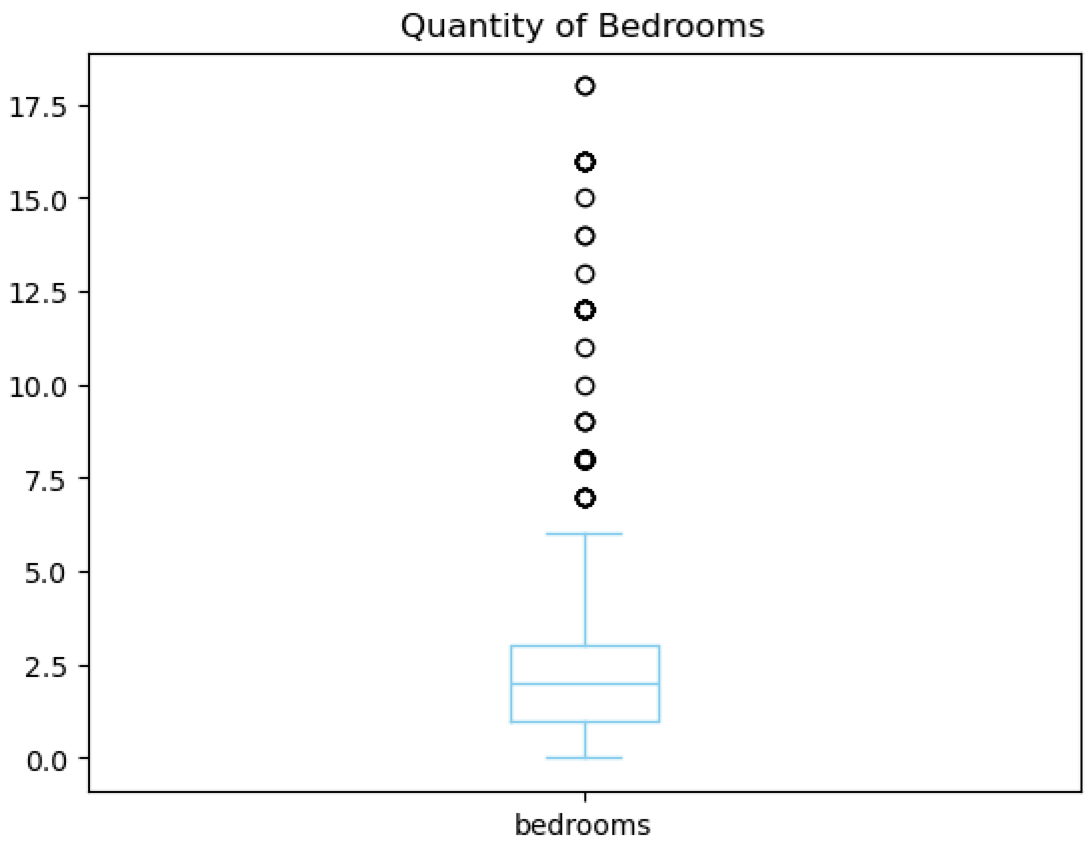
7 - Create a boxplot of a quantitative variable. What does this boxplot tell us about the variable?

In [139…

```
# replace with code to create boxplot
df['bedrooms'].plot(kind='box', color='skyblue', title="Quantity of Bedrooms")
```

Out[139…

```
<Axes: title={'center': 'Quantity of Bedrooms'}>
```



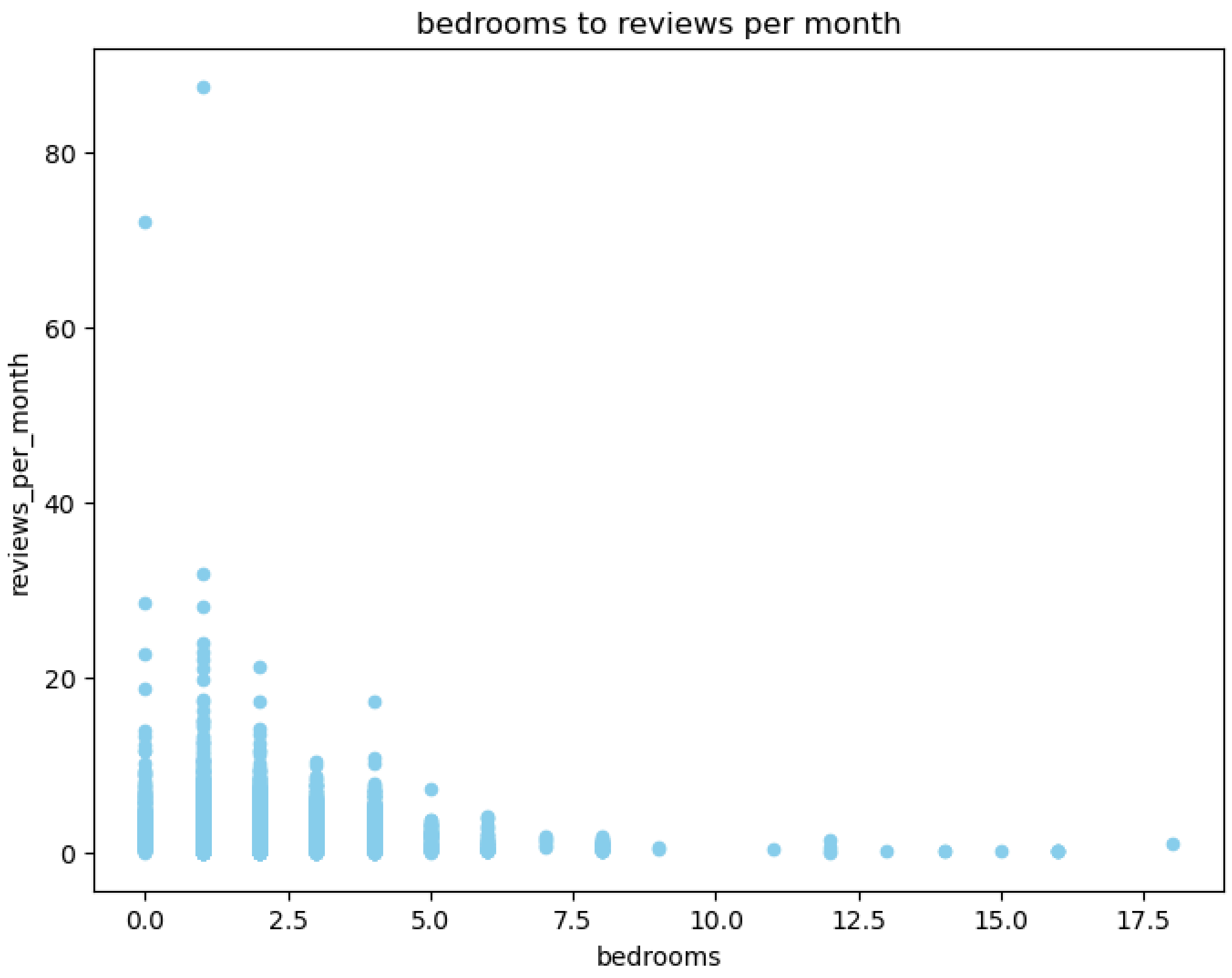
We can see that the majority of our properties are under ~6 bedrooms whlie most are between 3 and 1

8 - Create a scatterplot of 2 continuous variables. What do we learn from this plot?

In [140…

```
# replace with code to create scatterplot
df.plot(kind='scatter',
        x='bedrooms',
         y='reviews_per_month',
         color='skyblue',
         figsize=(8, 6),
         title="bedrooms to reviews per month"
         )
```

```
Out[140…  <Axes: title={'center': 'bedrooms to reviews per month'}, xlabel='bedrooms', ylabel='reviews_per_month'>
```



The more rooms you have the fewer reviews you will get. There some outliers in our data with 80ish reviews per month.

Part 2: Data Sources

9 - Import the air_quality_dataframe.csv dataset into Python, and join this dataset with our listings dataset. You haven't joined two datasets in this class, so this template will help you!

Our business partners at Aggie investments believe that adding in the Average Air Quality for listings could potentially add value to our analysis. We have utilized the following code to create a new dataset called `air_quality_dataframe.csv` via the OpenWeather Air Quality API. This dataframe has a corresponding listing `id` field as well as the Average Air quality for that listing.

In [102…

```
# replace with code to import the airquality dataset, name dataframe aq_df
aq_df = pd.read_csv("air_quality_dataframe.csv")
```

In [103…

```
# replace with code to join the airquality dataset with our listings dataframe
df = df.merge(aq_df, on='id', how='left')
df.head()
```

Out[103…

	id	name	host_id	host_name	host_since	host_is_superhost	calculated_host_listings_count	host_has_profile_p
0	6422	Nashville Charm	12172	Michele	4/3/09	f	1	
1	39870	Close to Vanderbilt 2	171184	Evelyn	7/18/10	t	1	
2	59576	Large Main Suite near Lake *ladies only NS plz	812128	Patricia And John	7/12/11	t	9	
3	72906	Vandy/Belmont/10 mins to Broadway - Sunny 800 ...	176117	Richard	7/21/10	t	1	
4	258817	ButterflyRoom-queen room, private bath	22296	Diana	6/19/09	t	6	

Questions to Answer:

- 10. Discuss the ethical considerations for our entire dataset (both Airbnb listings and AirQuality) -- Consider things like whether there is any personally identifiable (PI) data in our dataset, bias inherent in the sample of our data, ethical considerations of the impact of this task/analysis, etc.
    - Answer: Overall it is not to bad, we do have host names which likely dont add value but could add bias. Reviews is likely biases due to extremes, if people are going to review they definetly will if its bad and may if its really good. If it's average they likely won't. The name of the facility could be bad but it could add value to the dataset. Air quality could also be skewed since it is an average, using a median value likely would be better.
- 11. Are our data obtained from Primary or Secondary data sources?
    - Answer: Secondary Datasource

Part 3: Data Cleaning

Clean and transform our Airbnb listing data set. If you need some reminders about how to do this, revisit the data cleaning module!

12 - Think about any ethical concerns regarding this dataset. Remove any columns that personally identify hosts

```
In [104…
# replace with code to remove any columns that personally identify hosts
df.drop(columns="host_name", inplace=True)
```

13 - Go through each attribute column and perform various data transformations necessary to cleanse the dataset. For each attribute/column, report each data cleansing step performed and the underlying assumption as to why the data cleansing action was performed.

- Do not simply state that “all columns were trimmed” or restate the cleansing action itself.
- State the assumption (e.g., “M” was changed to “Male” because it was assumed that “M” indicated “Male” in this dataset.).
- Also, if no data transformations were made, state your assumption here as well (all data were assumed to be correct/clean).
- **WE WILL ADDRESS MISSING DATA IN UNIT 2, do NOT fill in or drop missing data** unless specifically instructed to.

```
In [105…
df.head()
```

Out[105…

	id	name	host_id	host_since	host_is_superhost	calculated_host_listings_count	host_has_profile_pic	host_ide
0	6422	Nashville Charm	12172	4/3/09	f	1	t	
1	39870	Close to Vanderbilt 2	171184	7/18/10	t	1	t	
2	59576	Large Main Suite near Lake *ladies only NS plz	812128	7/12/11	t	9	t	
3	72906	Vandy/Belmont/10 mins to Broadway - Sunny 800 ...	176117	7/21/10	t	1	t	
4	258817	ButterflyRoom- queen room, private bath	22296	6/19/09	t	6	t	

Data Transformations & Assumptions

- **id**
  - **Action** technically was converted to an int at the top
  - **Assumption** all values are correct
- **name**
  - **Action** none
  - **Assumption** all values are correct
- **host_id**
  - **Action** none
  - **Assumption** all values are correct
- **host_name**
  - **Action** dropped column
  - **Assumption** un-needed

- **host_since**
  - **Action** create new column called `days_as_host` and drop this column
  - **Assumption**
- **host_is_superhost**
  - **Action** Changed to 0/1
  - **Assumption** t: 1, f: 0
- **host_has_profile_pic**
  - **Action** Changed to 0/1
  - **Assumption** t: 1, f: 0
- **host_identity_verified**
  - **Action** Changed to 0/1
  - **Assumption** t: 1, f: 0
- **neighbourhood_group_cleansed**
  - **Action** Nothing Done
  - **Assumption** All Correct
- **room_type**
  - **Action** left alone
  - **Assumption** all correct
- **bathrooms_text**
  - **Action** Create new column called `bathrooms` and drop this column
  - **Assumption** "1 bathroom means One bathroom",
- **price**
  - **Action** Convert to float
  - **Assumption** removed spaces, $, and commas

In [106…

```
# did all this to see you wrote all the code below
# df['days_as_host'] = (pd.to_datetime('today') - pd.to_datetime(df['host_since'])).dt.days
# df['host_is_superhost'].replace({'t': True, 'f': False}, inplace=True)
# df['host_has_profile_pic'].replace({'t': True, 'f': False}, inplace=True)
# df['host_identity_verified'].replace({'t': True, 'f': False}, inplace=True)
# df['bathrooms'] = df["bathrooms_text"].str.split(' ', expand=True)[0].replace({"One": 1})
# df['price'] = df['price'].str.replace('[\$,]', '', regex=True).astype(float)

# df.drop(columns=["host_since", "bathrooms_text"], inplace=True)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8808 entries, 0 to 8807
Data columns (total 33 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              8808 non-null   int64  
 1   name                            8808 non-null   object 
 2   host_id                         8808 non-null   int64  
 3   host_since                      8808 non-null   object 
 4   host_is_superhost               8558 non-null   object 
 5   calculated_host_listings_count  8808 non-null   int64  
 6   host_has_profile_pic            8808 non-null   object 
 7   host_identity_verified          8808 non-null   object 
 8   host_listings_count             8808 non-null   int64  
 9   neighbourhood_cleansed          8808 non-null   object 
 10  latitude_x                      8808 non-null   float64
 11  longitude_x                     8808 non-null   float64
 12  availability_365                8808 non-null   int64  
 13  minimum_nights                  8808 non-null   int64  
 14  room_type                       8808 non-null   object 
 15  accommodates                    8808 non-null   int64  
 16  bathrooms_text                  8807 non-null   object 
 17  bedrooms                        8647 non-null   float64
 18  beds                            6588 non-null   float64
 19  price                           6589 non-null   object 
 20  number_of_reviews               8808 non-null   int64  
 21  reviews_per_month               7889 non-null   float64
 22  review_scores_rating            7889 non-null   float64
 23  review_scores_accuracy          7889 non-null   float64
 24  review_scores_cleanliness       7889 non-null   float64
 25  review_scores_checkin           7889 non-null   float64
 26  review_scores_communication     7889 non-null   float64
 27  review_scores_location          7889 non-null   float64
 28  review_scores_value             7889 non-null   float64
 29  Unnamed: 0                      8383 non-null   float64
 30  latitude_y                      8383 non-null   float64
 31  longitude_y                     8383 non-null   float64
 32  avg_air_quality                 8383 non-null   float64
dtypes: float64(16), int64(8), object(9)
memory usage: 2.2+ MB
```

There are a few specific transformations you will need to complete as well.

14 - Create a new `days_as_host` column using the following hints:

- Convert the `host_since` column to a `datetime` object
- Create the `days_as_host` column using the logic below (Note: this logic subtracts the host since date from the current date and then we pull the days from that calculation)
- Drop `host_since`

In [107…

```
# convert host_since to datetime object

df['days_as_host'] = (pd.to_datetime('today').normalize() - pd.to_datetime(df['host_since'])).dt.days # create new co

# drop host_since
df.drop(columns='host_since', inplace=True)
```

15 - Create a new column called `bathrooms`

- Begin by examining the `value_counts` of `bathrooms_text`
- Replace any numbers written in word form with the corresponding number (e.g., zero baths --> 0 baths)
- Create a new column called `bathrooms` by splitting the text from bathrooms_text on a space delimiter and extracting the first value
- Fill in missing values with 0 and drop `bathrooms_text`

In [108…

```
# check value_counts of bathrooms_text
# df["bathrooms_text"].value_counts()

# string replace any numbers in word form with the corresponding number form
df['bathrooms_text'] = df['bathrooms_text'].str.replace('One', '1')

# create new bathrooms column by extracting first value from 'bathrooms_text'
df['bathrooms'] =  df['bathrooms_text'].str.split(' ', n=1, expand=True)[0].astype(float)

# fill in missing values with 0
df['bathrooms'] =  df['bathrooms'].fillna('0')
```

```
# drop 'bathrooms_text' from df
df.drop(columns='bathrooms_text', inplace=True)
```

16 - Create a column called `short_term`. This column will be 1 if the `minimum_nights` column is less than 30, and 0 otherwise.

In [109…
```
# replace with code to create short_term column
df["short_term"] = np.where(df['minimum_nights'] < 30, 1, 0)
```

17 - Any columns containing 't' and 'f' as values (True, False), should be converted to 1, 0.

In [111…
```
# replace with code to replace t,f values with 1,0
df = df.applymap(lambda x: 1 if x == 't' else (0 if x == 'f' else x))

for column in df.columns:
    try:
        df[column] = pd.to_numeric(df[column], errors='ignore')
        df[column] = df[column].astype('Int64', errors='ignore')
    except Exception as e:
        print(f"Skipping column {column}: {e}")
        continue
```

In [112…
```
# df[df["host_is_superhost"].isna()]
# df.head()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8808 entries, 0 to 8807
Data columns (total 34 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              8808 non-null   Int64  
 1   name                            8808 non-null   object 
 2   host_id                         8808 non-null   Int64  
 3   host_is_superhost               8558 non-null   Int64  
 4   calculated_host_listings_count  8808 non-null   Int64  
 5   host_has_profile_pic            8808 non-null   Int64  
 6   host_identity_verified          8808 non-null   Int64  
 7   host_listings_count             8808 non-null   Int64  
 8   neighbourhood_cleansed          8808 non-null   object 
 9   latitude_x                      8808 non-null   float64
 10  longitude_x                     8808 non-null   float64
 11  availability_365                8808 non-null   Int64  
 12  minimum_nights                  8808 non-null   Int64  
 13  room_type                       8808 non-null   object 
 14  accommodates                    8808 non-null   Int64  
 15  bedrooms                        8647 non-null   Int64  
 16  beds                            6588 non-null   Int64  
 17  price                           6589 non-null   object 
 18  number_of_reviews               8808 non-null   Int64  
 19  reviews_per_month               7889 non-null   float64
 20  review_scores_rating            7889 non-null   float64
 21  review_scores_accuracy          7889 non-null   float64
 22  review_scores_cleanliness       7889 non-null   float64
 23  review_scores_checkin           7889 non-null   float64
 24  review_scores_communication     7889 non-null   float64
 25  review_scores_location          7889 non-null   float64
 26  review_scores_value             7889 non-null   float64
 27  Unnamed: 0                      8383 non-null   Int64  
 28  latitude_y                      8383 non-null   float64
 29  longitude_y                     8383 non-null   float64
 30  avg_air_quality                 8383 non-null   float64
 31  days_as_host                    8808 non-null   Int64  
 32  bathrooms                       8808 non-null   float64
 33  short_term                      8808 non-null   Int64  
dtypes: Int64(16), float64(14), object(4)
memory usage: 2.4+ MB
```

18 - We want to treat `price` as a float, but it's currently an object. Remove any text characters, then convert to float.

In [113…

```
# replace with code to remove text characters from price, then convert to float
df['price'] = df['price'].str.replace('[\$,]', '', regex=True).astype(float)
```

19 - You should drop variables that are not relevant to the analysis in this step (i.e., do we need the lat and long of properties or is that unnecessary info?). We will examine missing data and outliers in the Unit 2 Assessment, so don't worry about `int` or `float` columns for now (unless they should be dropped).

In [114…

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8808 entries, 0 to 8807
Data columns (total 34 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              8808 non-null   Int64  
 1   name                            8808 non-null   object 
 2   host_id                         8808 non-null   Int64  
 3   host_is_superhost               8558 non-null   Int64  
 4   calculated_host_listings_count  8808 non-null   Int64  
 5   host_has_profile_pic            8808 non-null   Int64  
 6   host_identity_verified          8808 non-null   Int64  
 7   host_listings_count             8808 non-null   Int64  
 8   neighbourhood_cleansed          8808 non-null   object 
 9   latitude_x                      8808 non-null   float64
 10  longitude_x                     8808 non-null   float64
 11  availability_365                8808 non-null   Int64  
 12  minimum_nights                  8808 non-null   Int64  
 13  room_type                       8808 non-null   object 
 14  accommodates                    8808 non-null   Int64  
 15  bedrooms                        8647 non-null   Int64  
 16  beds                            6588 non-null   Int64  
 17  price                           6589 non-null   float64
 18  number_of_reviews               8808 non-null   Int64  
 19  reviews_per_month               7889 non-null   float64
 20  review_scores_rating            7889 non-null   float64
 21  review_scores_accuracy          7889 non-null   float64
 22  review_scores_cleanliness       7889 non-null   float64
 23  review_scores_checkin           7889 non-null   float64
 24  review_scores_communication     7889 non-null   float64
 25  review_scores_location          7889 non-null   float64
 26  review_scores_value             7889 non-null   float64
 27  Unnamed: 0                      8383 non-null   Int64  
 28  latitude_y                      8383 non-null   float64
 29  longitude_y                     8383 non-null   float64
 30  avg_air_quality                 8383 non-null   float64
 31  days_as_host                    8808 non-null   Int64  
 32  bathrooms                       8808 non-null   float64
 33  short_term                      8808 non-null   Int64  
dtypes: Int64(16), float64(15), object(3)
memory usage: 2.4+ MB
```

In [115…
```
# remove columns based on your assumptions identified and perform other data cleaning steps as necessary.
# you might consider using a different code cell for each variable/column you make any changes to
df.drop(columns=["host_id", "host_is_superhost", "Unnamed: 0", "latitude_y", "latitude_x"], inplace=True)
"""
I contemplated hard on this, I don't think we need this data but all of the other data
could be useful in some way shape or form.
I don't think I would send all of this into a model at once but there is valuable insight from
all of this.
"""
```

Out[115…
```
"\nI contemplated hard on this, I don't think we need this data but all of the other data \ncould be useful in some way shape or form. \nI don't think I would send all of this into a model at once but there is valuable insight from \nall of this.\n"
```

20 - Display the finalized clean dataset

In [116…
```
# replace with code to display finalized clean dataset
df
```

Out[116…

	id	name	calculated_host_listings_count	host_has_profile_pic	host_identity_verified	host_list
0	6422	Nashville Charm	1	1	1	
1	39870	Close to Vanderbilt 2	1	1	1	
2	59576	Large Main Suite near Lake *ladies only NS plz	9	1	1	
3	72906	Vandy/Belmont/10 mins to Broadway - Sunny 800 ...	1	1	1	
4	258817	ButterflyRoom-queen room, private bath	6	1	1	
...	...	...	...	...	...	
8803	1183419901812352256	The Vinyl Vault - 6 Minutes from Broadway	124	1	1	
8804	1183420167756057088	The Pink Royale - 6 Mins from Broadway	124	1	1	
8805	1183429746281090816	Southern Charm Townhome in East!	6	1	1	
8806	1183565468543983104	Stunning 3Bdr Home in Nashville	1	1	1	
8807	1184189146609359616	1 Mile to Nissan Stadium! 1 King Bed w/city vi...	1	1	0	

8808 rows × 29 columns