Description:

Data cleaning is the process of fixing or removing incorrect, corrupted, duplicate, or incomplete data within a dataset. Messy data leads to unreliable outcomes. Cleaning data is an essential part of data analysis, and demonstrating your data cleaning skills is key to landing a job. Here are some projects to test out your data cleaning skills:

Key Concepts and Challenges:

Data Integrity: Ensuring the accuracy, consistency, and reliability of data throughout the cleaning process.
Missing Data Handling: Dealing with missing values by either imputing them or making informed decisions on how to handle gaps in the dataset.
Duplicate Removal: Identifying and eliminating duplicate records to maintain data uniqueness.
Standardization: Consistent formatting and units across the dataset for accurate analysis.
Outlier Detection: Identifying and addressing outliers that may skew analysis or model performance.

About this file

Suggest Edits "AB_NYC_2019" - Summary information and metrics for listings in New York City. It is good for exploration, visualizations and predictions.

Import library

```python
import pandas as pd
import numpy as np

# Load the dataset
df = pd.read_csv('AB_NYC_2019.csv')

# Display basic info and first few rows
print(df.info())
print(df.head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              48895 non-null  int64
 1   name                            48879 non-null  object
 2   host_id                         48895 non-null  int64
 3   host_name                       48874 non-null  object 
 4   neighbourhood_group             48895 non-null  object 
 5   neighbourhood                   48895 non-null  object 
 6   latitude                        48895 non-null  float64
 7   longitude                       48895 non-null  float64
 8   room_type                       48895 non-null  object 
 9   price                           48895 non-null  int64  
 10  minimum_nights                  48895 non-null  int64  
 11  number_of_reviews               48895 non-null  int64  
 12  last_review                     38843 non-null  object 
 13  reviews_per_month               38843 non-null  float64
 14  calculated_host_listings_count  48895 non-null  int64  
 15  availability_365                48895 non-null  int64  
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
None
     id                                              name  host_id  \
0  2539                Clean & quiet apt home by the park     2787   
1  2595                             Skylit Midtown Castle     2845   
2  3647               THE VILLAGE OF HARLEM....NEW YORK !     4632   
3  3831                   Cozy Entire Floor of Brownstone     4869   
4  5022  Entire Apt: Spacious Studio/Loft by central park     7192   

     host_name neighbourhood_group neighbourhood  latitude  longitude 
\
0         John            Brooklyn    Kensington  40.64749  -73.97237 

1     Jennifer           Manhattan       Midtown  40.75362  -73.98377 

2    Elisabeth           Manhattan        Harlem  40.80902  -73.94190 

3  LisaRoxanne            Brooklyn  Clinton Hill  40.68514  -73.95976 

4        Laura           Manhattan   East Harlem  40.79851  -73.94399 


         room_type  price  minimum_nights  number_of_reviews 
last_review  \
0     Private room    149               1                  9  2018-10-
19
1  Entire home/apt    225               1                 45  2019-05-
21
2     Private room    150               3                  0        
NaN
3  Entire home/apt     89               1                270  2019-07-
05
4  Entire home/apt     80              10                  9  2018-11-
19

   reviews_per_month  calculated_host_listings_count  availability_365
```

```
0            0.21                               6          365
1            0.38                               2          355
2             NaN                               1          365
3            4.64                               1          194
4            0.10                               1            0
```

1. Handling Missing Data

```
print("\nMissing values before handling:")
print(df.isnull().sum())


Missing values before handling:
id                                    0
name                                 16
host_id                               0
host_name                            21
neighbourhood_group                   0
neighbourhood                         0
latitude                              0
longitude                             0
room_type                             0
price                                 0
minimum_nights                        0
number_of_reviews                     0
last_review                       10052
reviews_per_month                 10052
calculated_host_listings_count        0
availability_365                      0
dtype: int64
```

Impute 'reviews_per_month' with median and fill other missing values accordingly

```
df['reviews_per_month'].fillna(df['reviews_per_month'].median(),
inplace=True)
df.fillna(method='ffill', inplace=True)

C:\Users\Admin\AppData\Local\Temp\ipykernel_9036\4187586147.py:1:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
```

```
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  df['reviews_per_month'].fillna(df['reviews_per_month'].median(),
inplace=True)
C:\Users\Admin\AppData\Local\Temp\ipykernel_9036\4187586147.py:2:
FutureWarning: DataFrame.fillna with 'method' is deprecated and will
raise in a future version. Use obj.ffill() or obj.bfill() instead.
  df.fillna(method='ffill', inplace=True)
```

2. Removing Duplicates

```
df.drop_duplicates(inplace=True)
```

3. Standardization

Convert column names to lowercase

```
df.columns = df.columns.str.lower()
```

Standardize price to 2 decimal places

```
df['price'] = df['price'].round(2)
```

4. Outlier Detection and Removal

Remove outliers based on price

```
q1 = df['price'].quantile(0.25)
q3 = df['price'].quantile(0.75)
iqr = q3 - q1
```

```python
lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr
df = df[(df['price'] >= lower_bound) & (df['price'] <= upper_bound)]
```

Save cleaned data to a new CSV

```python
cleaned_file = 'AB_NYC_2019_cleaned.csv'
df.to_csv(cleaned_file, index=False)

print("\nData cleaning complete. Cleaned data saved as
AB_NYC_2019_cleaned.csv")
```

```
Data cleaning complete. Cleaned data saved as AB_NYC_2019_cleaned.csv
```