US House Listings

cleaned by:

Mohamed Ali Taher Essa and Hassan Hamadi Hassan

Work and steps:

-We call the Pandas library and then call the data file, which is usually in the format(Microsoft Excel Comma Separated Values File (.csv)) Which we upload to the COLAB files and then review it briefly

```
import pandas as pd
df = pd.read_csv('/content/US House Listings.csv')
df.head(10)
```

	State	City	Street	Zipcode	Bedroom	Bathroom	Area	PPSq	LotArea
0	AL	Saraland	Scott Dr	36571.0	4.0	2.0	1614.0	148.636927	0.38050(
1	AL	Robertsdale	Cowpen Creek Rd	36567.0	3.0	2.0	1800.0	144.388889	3.20000(
2	AL	Gulf Shores	Spinnaker Dr #201	36542.0	2.0	2.0	1250.0	274.000000	NaN
3	AL	Chelsea	Mallet Way	35043.0	3.0	3.0	2224.0	150.629496	0.26000(
4	AL	Huntsville	Turtlebrook Ct	35811.0	3.0	2.0	1225.0	204.081633	NaN
5	AL	Montgomery	Brampton Ln	36117.0	3.0	2.0	1564.0	96.547315	0.20000(
6	AL	Boaz	Greenwood Ave	35957.0	3.0	2.0	1717.0	139.196273	0.38000(

We display all columns using the following code:

```
pd.set_option('display.max_columns',None)
df.head()
```

	State	City	Street	Zipcode	Bedroom	Bathroom	Area	PPSq	LotArea
0	AL	Saraland	Scott Dr	36571.0	4.0	2.0	1614.0	148.636927	0.3805
1	AL	Robertsdale	Cowpen Creek Rd	36567.0	3.0	2.0	1800.0	144.388889	3.2000
2	AL	Gulf Shores	Spinnaker Dr #201	36542.0	2.0	2.0	1250.0	274.000000	NaN
3	AL	Chelsea	Mallet Way	35043.0	3.0	3.0	2224.0	150.629496	0.2600

Understanding the Columns:

State: The state in which the property is located (AL:Alabama) . Includes all US states except Hawaii.

City: The city where the property is situated.

Street: The street address of the property.

Zipcode: The postal code associated with the property.

Bedroom: The number of bedrooms in the house.

Bathroom: The number of bathrooms in the house.

Area(sqft): The total area of the house.

PPSq(Price Per Square Foot): The cost per square foot of the property.

LotArea(acres): The total land area associated with the property.

MarketEstimate(Dollars $): Estimated market value of the property. This value is estimated using Zillow's own algorithm.

RentEstimate:(Dollars $) Estimated rental value of the property. This value is estimated using Zillow's own algorithm.

Latitude: The latitude coordinates of the property.

Longitude: The longitude coordinates of the property.

ListedPrice:(Dollars $) The listed price of the property.

Then we show the total number of rows and columns:

```
df.shape
```

```
(22681, 14)
```

After that, we identify the general data:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22681 entries, 0 to 22680
Data columns (total 14 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   State           22681 non-null  object 
 1   City            22681 non-null  object 
 2   Street          22681 non-null  object 
 3   Zipcode         22681 non-null  float64
 4   Bedroom         22667 non-null  float64
 5   Bathroom        22647 non-null  float64
 6   Area            22681 non-null  float64
 7   PPSq            22681 non-null  float64
 8   LotArea         21779 non-null  float64
 9   MarketEstimate  15445 non-null  float64
 10  RentEstimate    16705 non-null  float64
 11  Latitude        22681 non-null  float64
 12  Longitude       22681 non-null  float64
 13  ListedPrice     22681 non-null  float64
dtypes: float64(11), object(3)
memory usage: 2.4+ MB
```

Then we know which columns contain missing data and the number of missing data and Percentage of missing values using the following code:

```
missing_values = df.isna().sum()

missing_percentage = (df.isna().mean() * 100).round(2)

missing_info = pd.DataFrame({
    'Missing Values': missing_values,
    'Percentage%': missing_percentage
})

print(missing_info)
```

```
                Missing Values  Percentage%
State                        0         0.00
City                         0         0.00
Street                       0         0.00
Zipcode                      0         0.00
Bedroom                     14         0.06
Bathroom                    34         0.15
Area                         0         0.00
PPSq                         0         0.00
LotArea                    902         3.98
MarketEstimate            7236        31.90
RentEstimate              5976        26.35
Latitude                     0         0.00
Longitude                    0         0.00
ListedPrice                  0         0.00
```

Now we only have the Country column that contains missing data, but it is text data, so we will try to find out what data it contains:

```
df['MarketEstimate'].value_counts()
```

```
340900.0    12
248000.0    11
329500.0    10
399000.0    10
298900.0    10
            ..
103600.0     1
82000.0      1
185500.0     1
24531.0      1
193400.0     1
Name: MarketEstimate, Length: 6860, dtype: int64
```

```
df['RentEstimate'].value_counts()
```

```
2500.0    415
1999.0    388
2199.0    361
1800.0    358
1500.0    304
          ...
2031.0      1
3543.0      1
6384.0      1
4060.0      1
4777.0      1
Name: RentEstimate, Length: 3313, dtype: int64
```

We notice that there are more than two types of data with varying frequencies, so we will remove the rows that contain missing information.

```
df.dropna(subset=['MarketEstimate'], inplace=True)
```

```
df.dropna(subset=['RentEstimate'], inplace=True)
```

Now we get rid of the missing data in the LotEria column, as it is digital data and its number is very small, so we will replace any NaN (Not a Number) values in the 'LotEria' column with 0.

```
df['LotArea'] = df['LotArea'].fillna(0)
```

This line calculates the median value of the 'Bedroom' and 'Bathroom' columns in the DataFrame (df). The median is a measure of central tendency that represents the middle value of a dataset. It's used here to impute missing values in the 'Bedroom' and 'Bathroom' columns.

```
median_bedroom = df['Bedroom'].median()
df['Bedroom'].fillna(median_bedroom, inplace=True)

median_bathroom = df['Bathroom'].median()
df['Bathroom'].fillna(median_bathroom, inplace=True)
```

We notice now that we got rid of the missing data and now we can do our calculations on the data.

```
df.isnull().sum()
```

```
State             0
City              0
Street            0
Zipcode           0
Bedroom           0
Bathroom          0
Area              0
PPSq              0
LotArea           0
MarketEstimate    0
RentEstimate      0
Latitude          0
Longitude         0
```

```
ListedPrice       0
dtype: int64
```