housing-eda

August 22, 2025

HOUSING (TYPE OF HOUSES) - EDA

[1]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns # visualisation of cleaned data using various plots
import matplotlib.pyplot as plt # plotting graphs

print("Successfully imported all necessary libraries")
```

```
Successfully imported all necessary libraries
```

[3]:
```
import os

print(os.listdir("C:/Users/Tanya Raj/OneDrive/Desktop"))
```

```
['02_numpy(linear algebra).ipynb', '1.xlsx', '2(wine).xlsx', '3(Housing).csv',
'ADHAAR CARD.pdf', 'aditya', 'Arduino IDE.lnk', 'Canva.lnk', 'desktop.ini',
'FITA', 'major_project doc', 'myself info', 'python basics', 'python
function-2']
```

[8]:
```
pip install openpyxl
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: openpyxl in c:\users\tanya
raj\appdata\roaming\python\python313\site-packages (3.1.5)
Requirement already satisfied: et-xmlfile in c:\users\tanya
raj\appdata\roaming\python\python313\site-packages (from openpyxl) (2.0.0)
Note: you may need to restart the kernel to use updated packages.


[notice] A new release of pip is available: 25.1.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

[14]:
```
df = pd.read_csv(r"C:\Users\Tanya Raj\OneDrive\Desktop/3(Housing).csv")
print(pd.options.display.max_rows) # Showing the maximum number of rows which␣
 ↪can be displayed

df.head() # Displaying first set of datapoints
# print(df.to_string()) => Displaying entire dataset
```

60

[14]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	\
0	13300000	7420	4	2	3	yes	no	no	
1	12250000	8960	4	4	4	yes	no	no	
2	12250000	9960	3	2	2	yes	no	yes	
3	12215000	7500	4	2	2	yes	no	yes	
4	11410000	7420	4	1	2	yes	yes	yes	

	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	no	yes	2	yes	furnished
1	no	yes	3	no	furnished
2	no	no	2	yes	semi-furnished
3	no	yes	3	yes	furnished
4	no	yes	2	no	furnished

[10]:
```
df = pd.read_csv(r"C:\Users\Tanya Raj\OneDrive\Desktop\3(Housing).csv")
```

[ ]:
```
print(pd.options.display.max_rows)
df.head()
```

60

[ ]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	\
0	13300000	7420	4	2	3	yes	no	no	
1	12250000	8960	4	4	4	yes	no	no	
2	12250000	9960	3	2	2	yes	no	yes	
3	12215000	7500	4	2	2	yes	no	yes	
4	11410000	7420	4	1	2	yes	yes	yes	

	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	no	yes	2	yes	furnished
1	no	yes	3	no	furnished
2	no	no	2	yes	semi-furnished
3	no	yes	3	yes	furnished
4	no	yes	2	no	furnished

[15]:
```
#Step 2:Understand the nature of the data values

df.describe()
df.describe(include = 'object')
```

[15]:

	mainroad	guestroom	basement	hotwaterheating	airconditioning	prefarea	\
count	545	545	545	545	545	545	
unique	2	2	2	2	2	2	
top	yes	no	no	no	no	no	
freq	468	448	354	520	373	417	

```
       furnishingstatus
count               545
unique                3
top      semi-furnished
freq                227
```

[16]:
```
df.columns
```

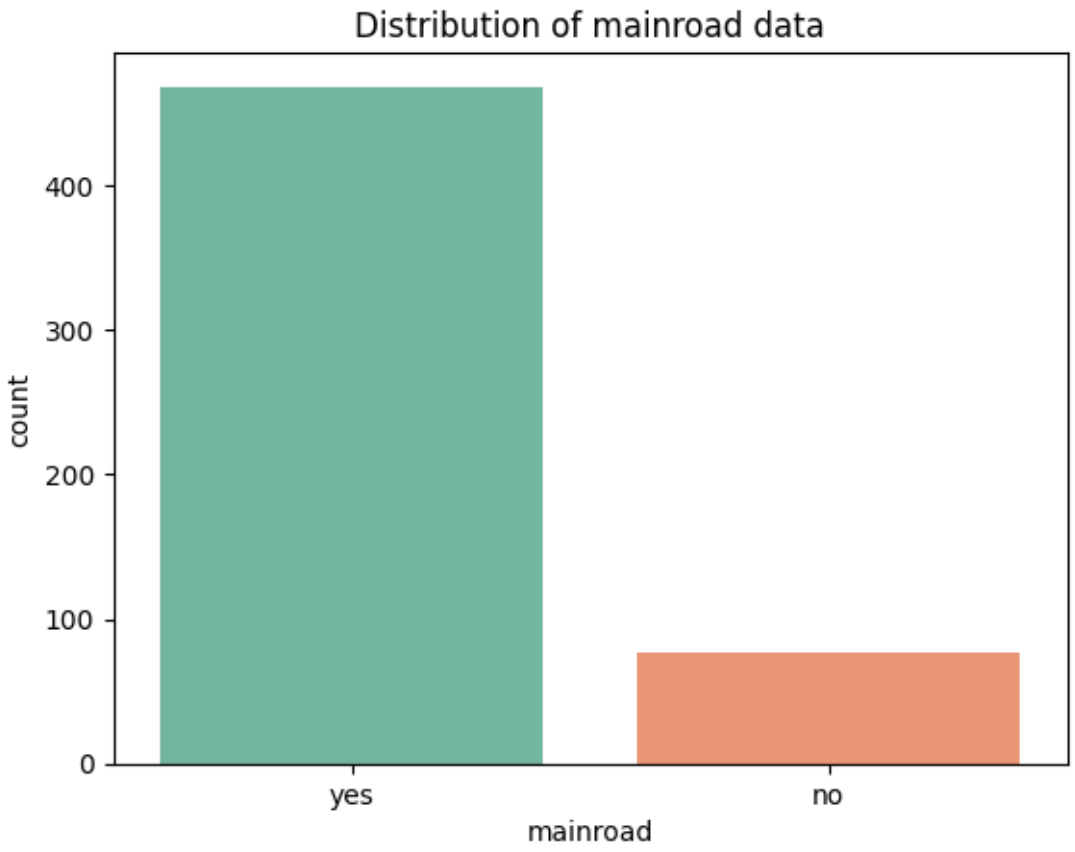
[16]:
```
Index(['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroad',
       'guestroom', 'basement', 'hotwaterheating', 'airconditioning',
       'parking', 'prefarea', 'furnishingstatus'],
      dtype='object')
```

[17]:
```
df.columns.isnull()
```

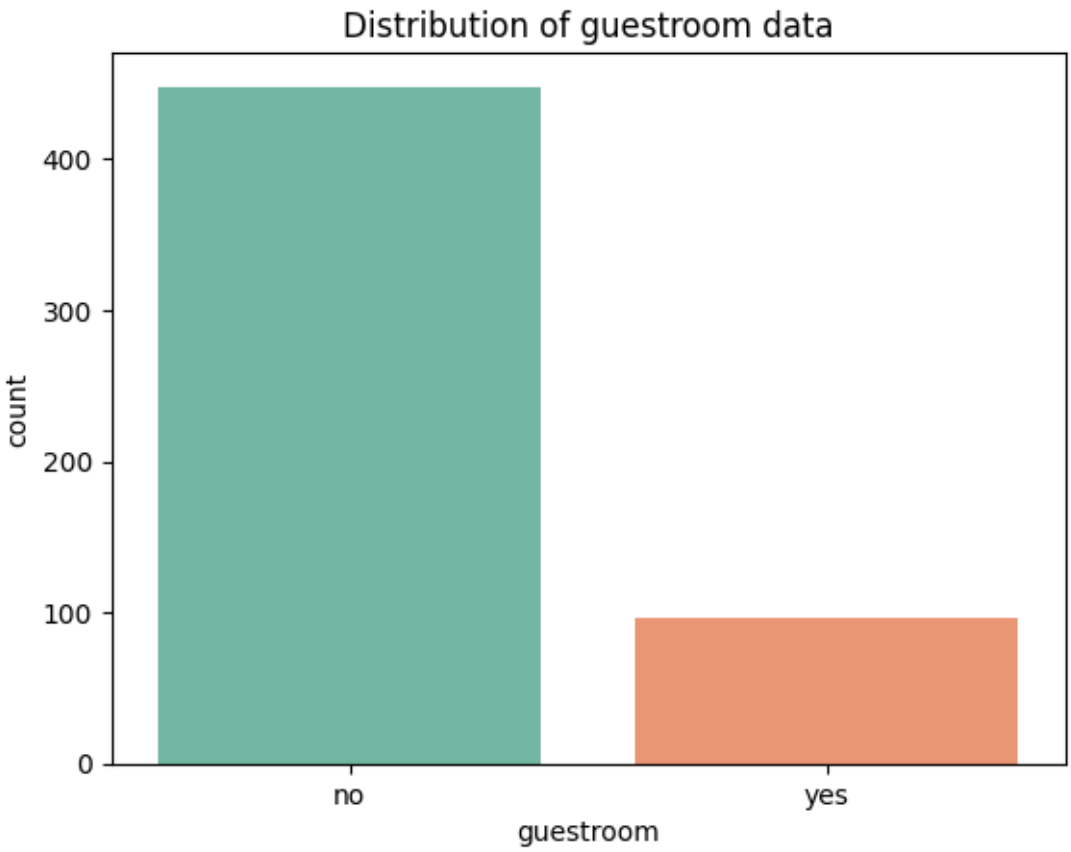
[17]:
```
array([False, False, False, False, False, False, False, False, False,
       False, False, False, False])
```

[21]:
```
categorical_labels = ['mainroad', 'guestroom', 'basement',
                      'hotwaterheating', 'airconditioning',
                      'prefarea', 'furnishingstatus']

for label in categorical_labels:
    sns.countplot(x=label, data=df, palette="Set2")  #  use palette

    plt.title(f'Distribution of {label} data')
    plt.show()
```

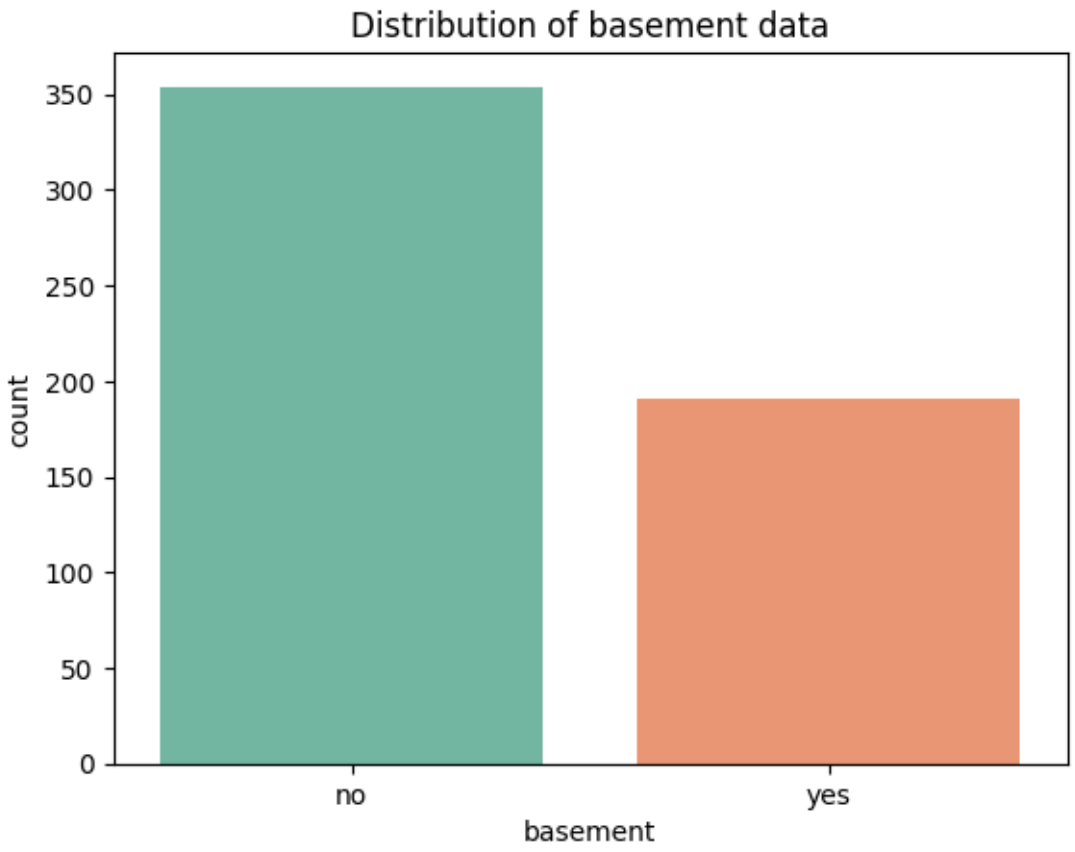
```
C:\Users\Tanya Raj\AppData\Local\Temp\ipykernel_22784\211982551.py:6:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(x=label, data=df, palette="Set2")  #  use palette
```



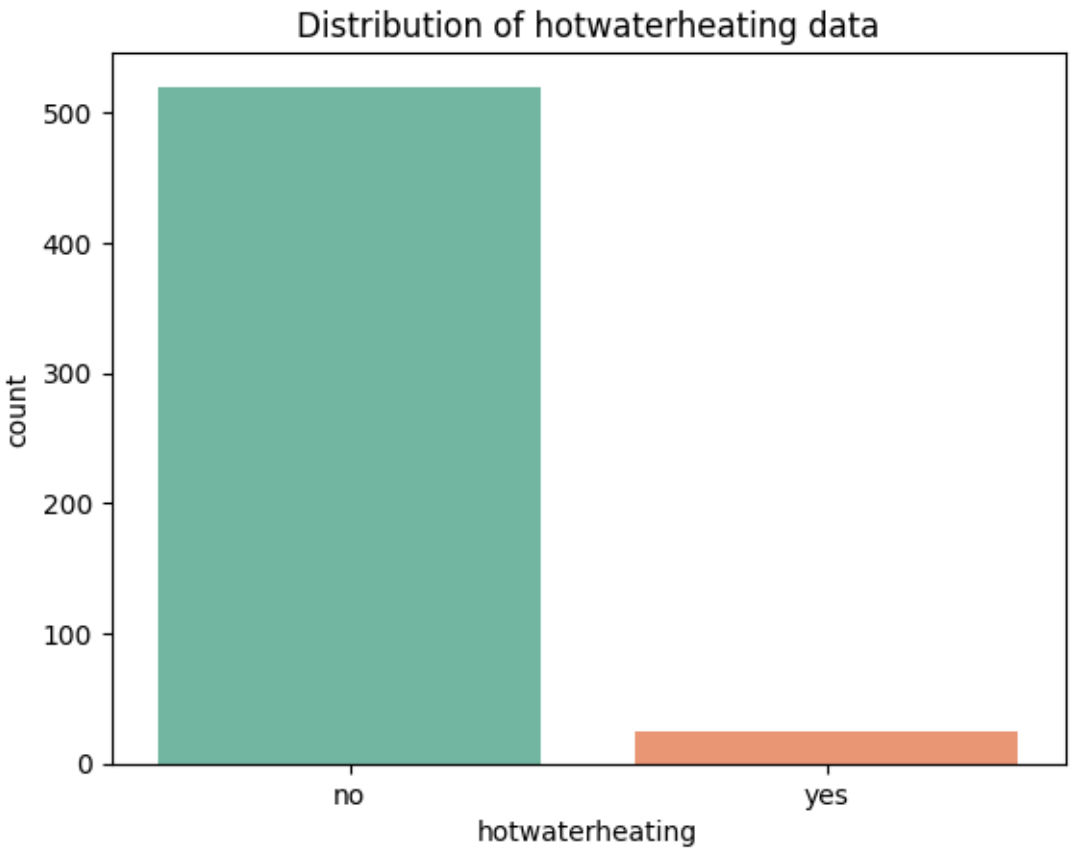

```
C:\Users\Tanya Raj\AppData\Local\Temp\ipykernel_22784\211982551.py:6:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(x=label, data=df, palette="Set2")  #   use palette
```



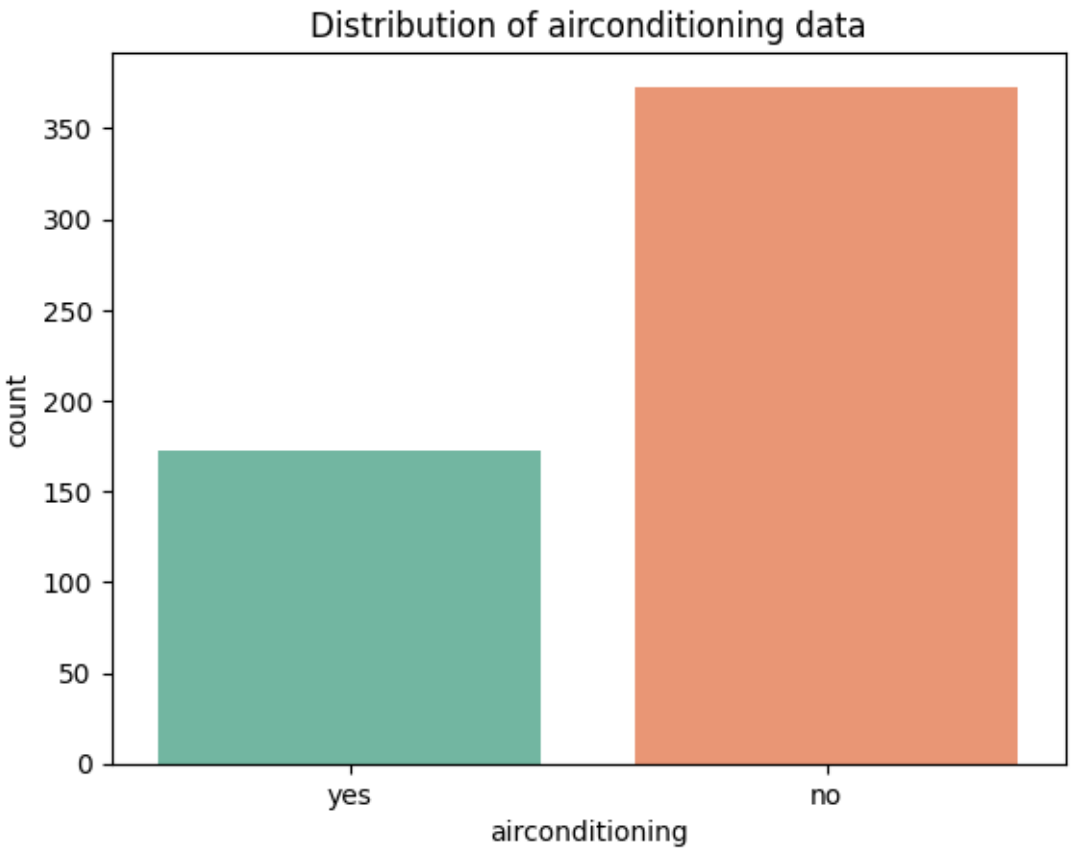

```
C:\Users\Tanya Raj\AppData\Local\Temp\ipykernel_22784\211982551.py:6:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(x=label, data=df, palette="Set2")  #   use palette
```



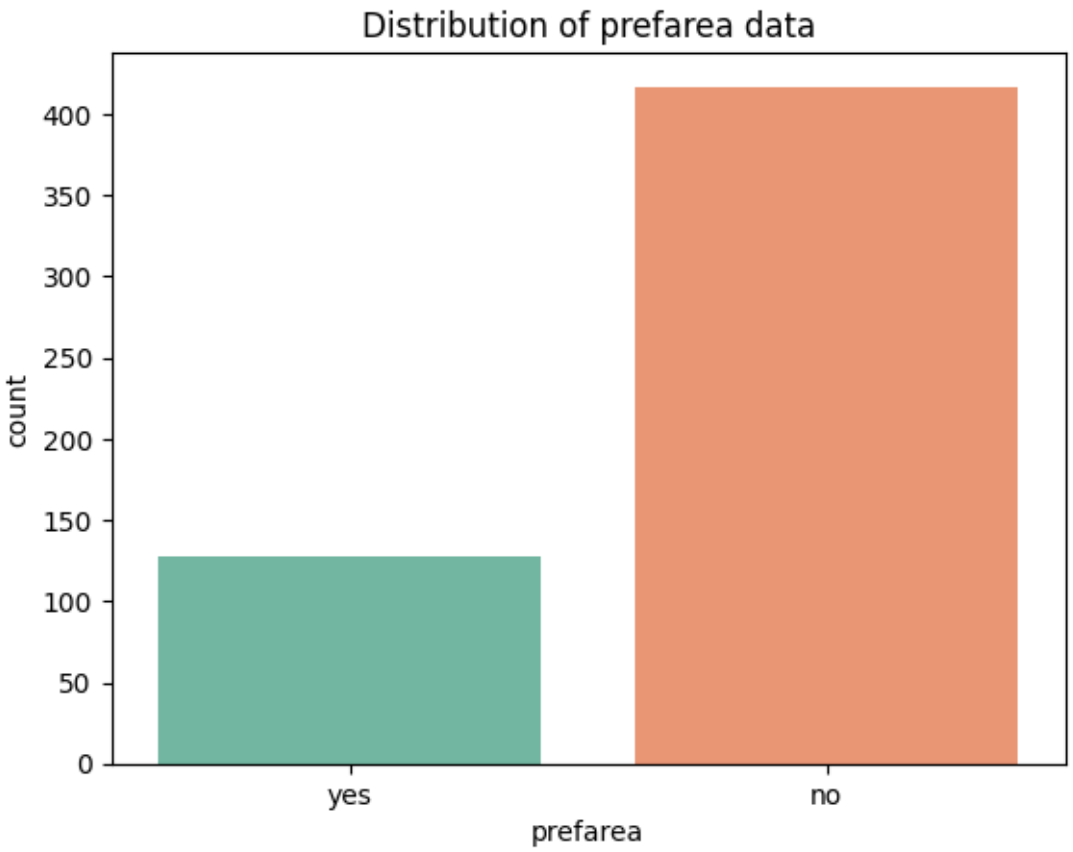

```
C:\Users\Tanya Raj\AppData\Local\Temp\ipykernel_22784\211982551.py:6:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(x=label, data=df, palette="Set2")  #   use palette
```



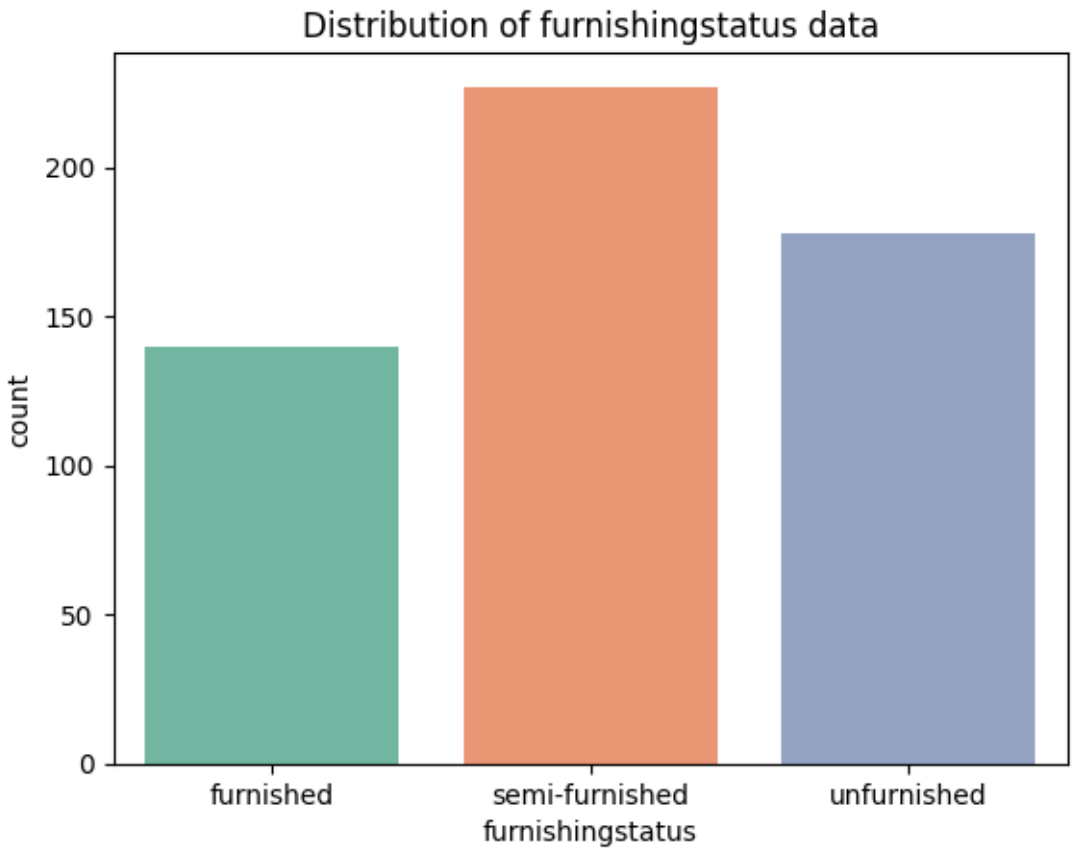

```
C:\Users\Tanya Raj\AppData\Local\Temp\ipykernel_22784\211982551.py:6:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(x=label, data=df, palette="Set2")  #   use palette
```



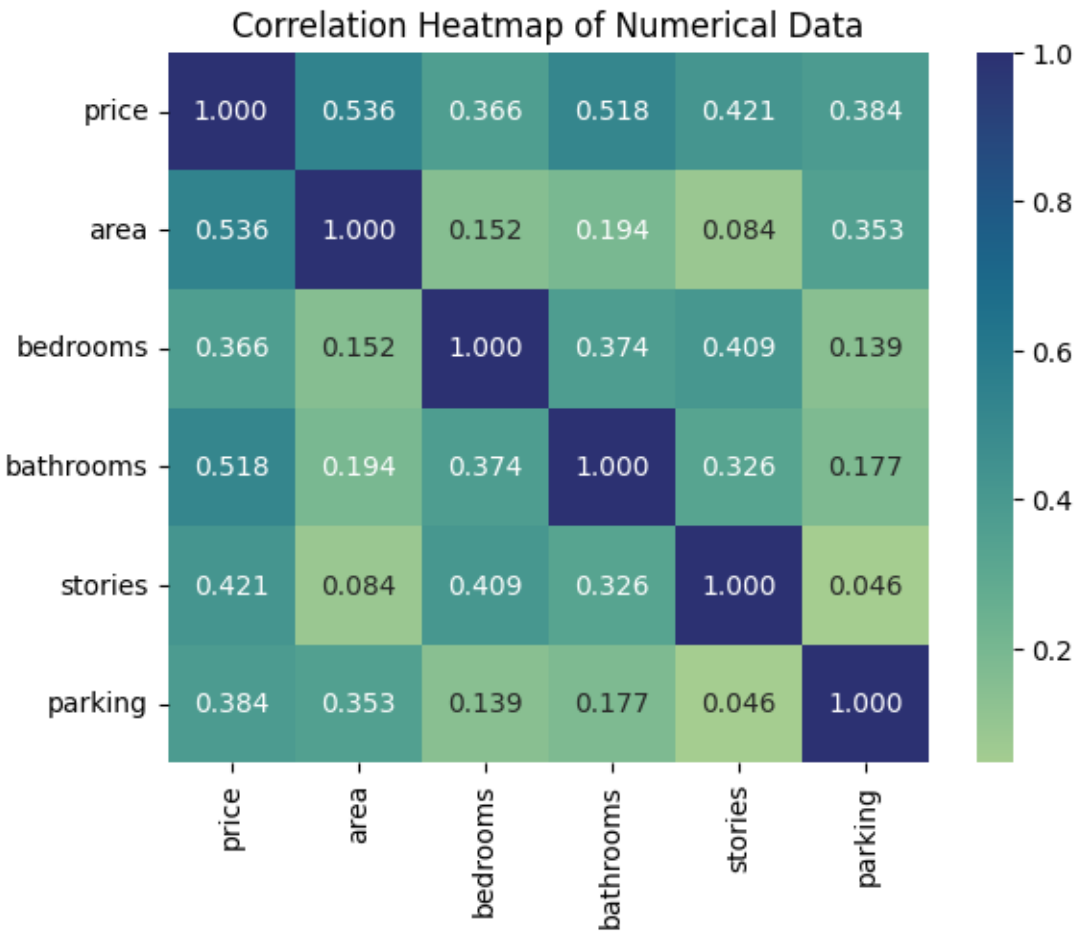

```
C:\Users\Tanya Raj\AppData\Local\Temp\ipykernel_22784\211982551.py:6:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(x=label, data=df, palette="Set2")  #   use palette
```




```
C:\Users\Tanya Raj\AppData\Local\Temp\ipykernel_22784\211982551.py:6:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(x=label, data=df, palette="Set2")  #   use palette
```



[22]:
```
#Correlation analysis of numerical data

correlation_df = df.copy() # make a copy of the dataframe
correlation_analysis = correlation_df.corr(numeric_only=True) # correlation␣
 ↪matrix for numeric data is created

sns.heatmap(correlation_analysis, annot=True, cmap='crest', fmt=".3f") # plots␣
 ↪data with 3-decimal place accuracy
plt.title("Correlation Heatmap of Numerical Data")
plt.show()
```



CONCLUSION:

No extreme correlation between any metrics

(Price,Area) and (Price, Bathrooms) have reasonably good positive correlation.