

LAB 2

TASK 1

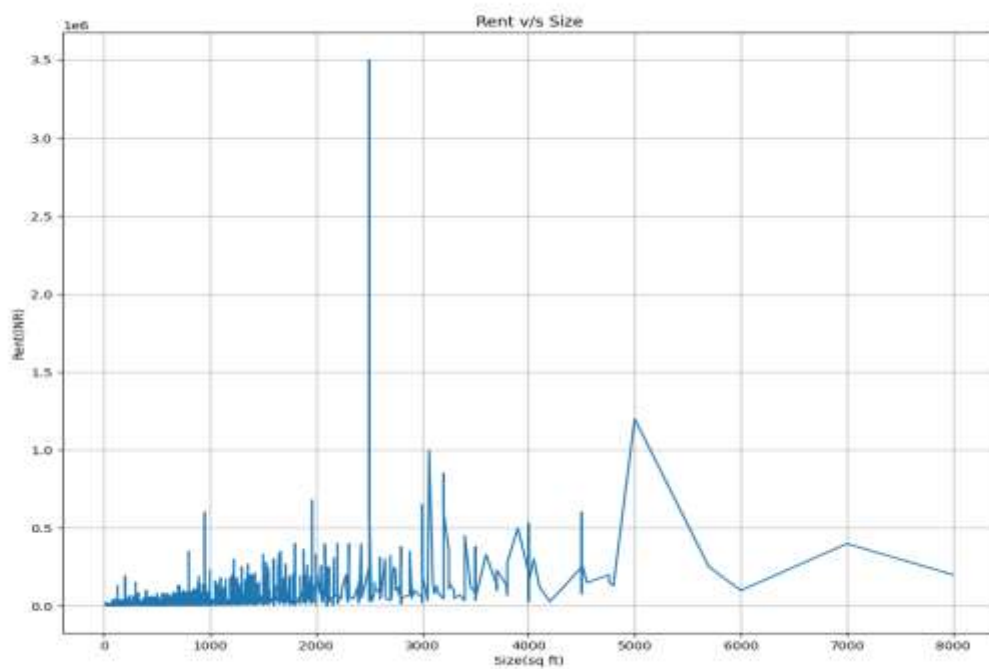
Line Plot

- Plot the rent ("Rent" column) against the size ("Size" column).
- Use different line styles (solid, dashed, dotted) for better visualization.
- Add gridlines to the plot.
- Title: "Rent vs. Size"
- Label the x-axis as "Size (sq ft)" and the y-axis as "Rent (INR)".
- Set the legend position to the top-right corner

INPUT

```
avg = data.groupby('Furnishing Status')['Rent'].mean()
fig, ax = plt.subplots(figsize = (12, 10))
avg.plot(kind = 'bar', color = ['red', 'blue', 'green'], title="Average Rent by Furnishing Status" )
ax.set(xlabel='Furnishing Status', ylabel='Average Rent (INR)')
fig.savefig("Tasks2.png")
plt.show()
```

OUTPUT



TASK 2

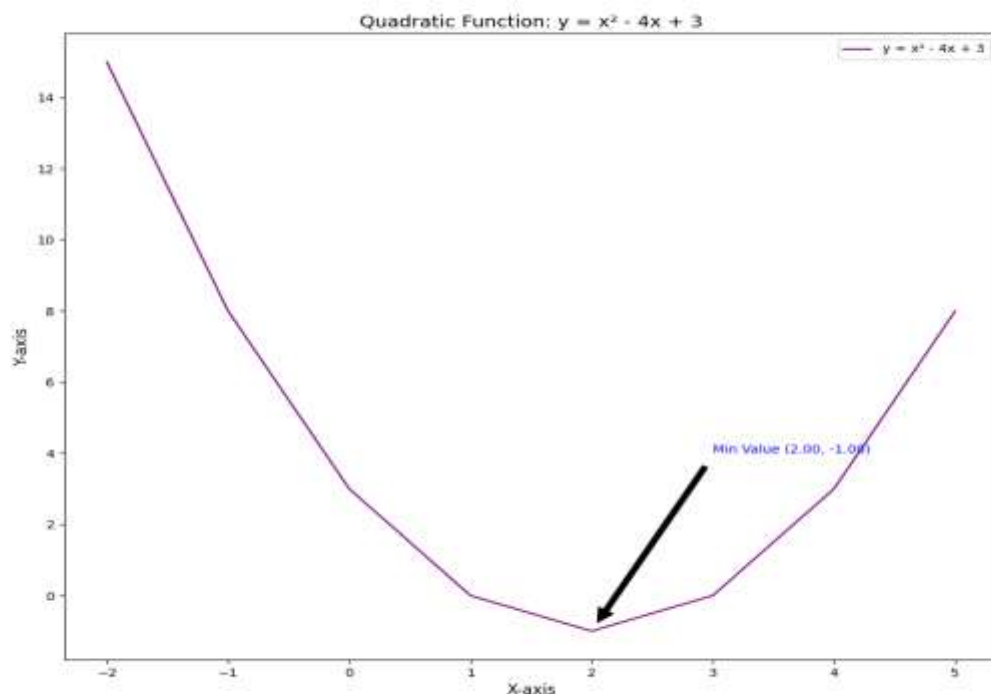
BarChart

- Create a bar chart to display the average rent for each furnishing status
- Use different colors for each bar.
- Title: "Average Rent by Furnishing Status"
- Label the x-axis as "Furnishing Status" and the y-axis as "Average Rent (INR)"

INPUT

```
avg = data.groupby('Furnishing Status')['Rent'].mean()
fig, ax = plt.subplots(figsize = (12, 10))
avg.plot(kind = 'bar', color = ['red', 'blue', 'green'], title="Average Rent by Furnishing Status" )
ax.set(xlabel='Furnishing Status', ylabel='Average Rent (INR)')
fig.savefig("Tasks2.png")
plt.show()
```

OUTPUT



TASK 3

Scatter Plot

- Plot the number of bathrooms ("Bathroom" column) against the rent ("Rent" column).
- Use different marker styles for points representing flats in different cities ("City" column).
- Title: "Bathrooms vs. Rent"
- Label the x-axis as "Number of Bathrooms" and the y-axis as "Rent (INR)".
- Use a legend to indicate cities and place it below the plot.

INPUT

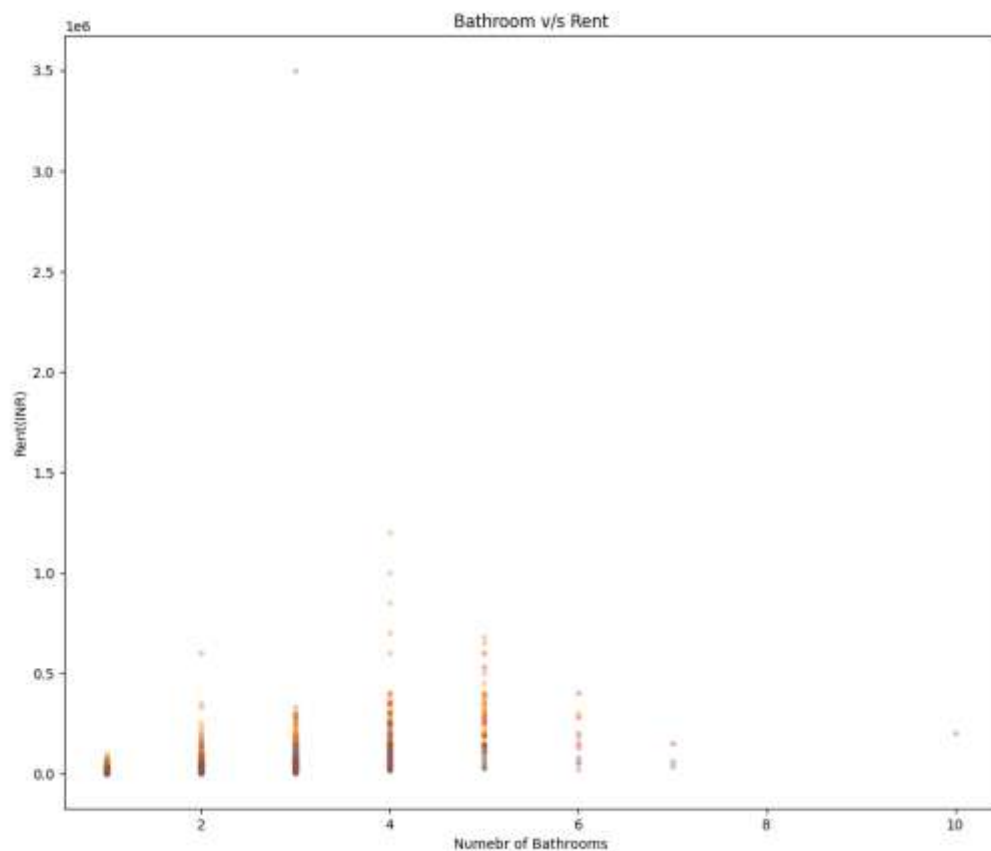
```
# Plot the number of bathrooms ("Bathroom" column) against the rent ("Rent" column)
fig, axes = plt.subplots(figsize=(12, 10))

for city in data['City'].unique():
    new_data = data[data['City'] == city]

    axes.scatter(x = new_data['Bathroom'], y = new_data['Rent'], marker = 'H', label=city, s=12,
alpha = 0.3)

axes.set(xlabel='Numebr of Bathrooms', ylabel='Rent(INR)', title='Bathroom v/s Rent')
ax.legend()
ax.grid(True)
fig.savefig("Task3.png")
plt.show()
```

OUTPUT



TASK 4

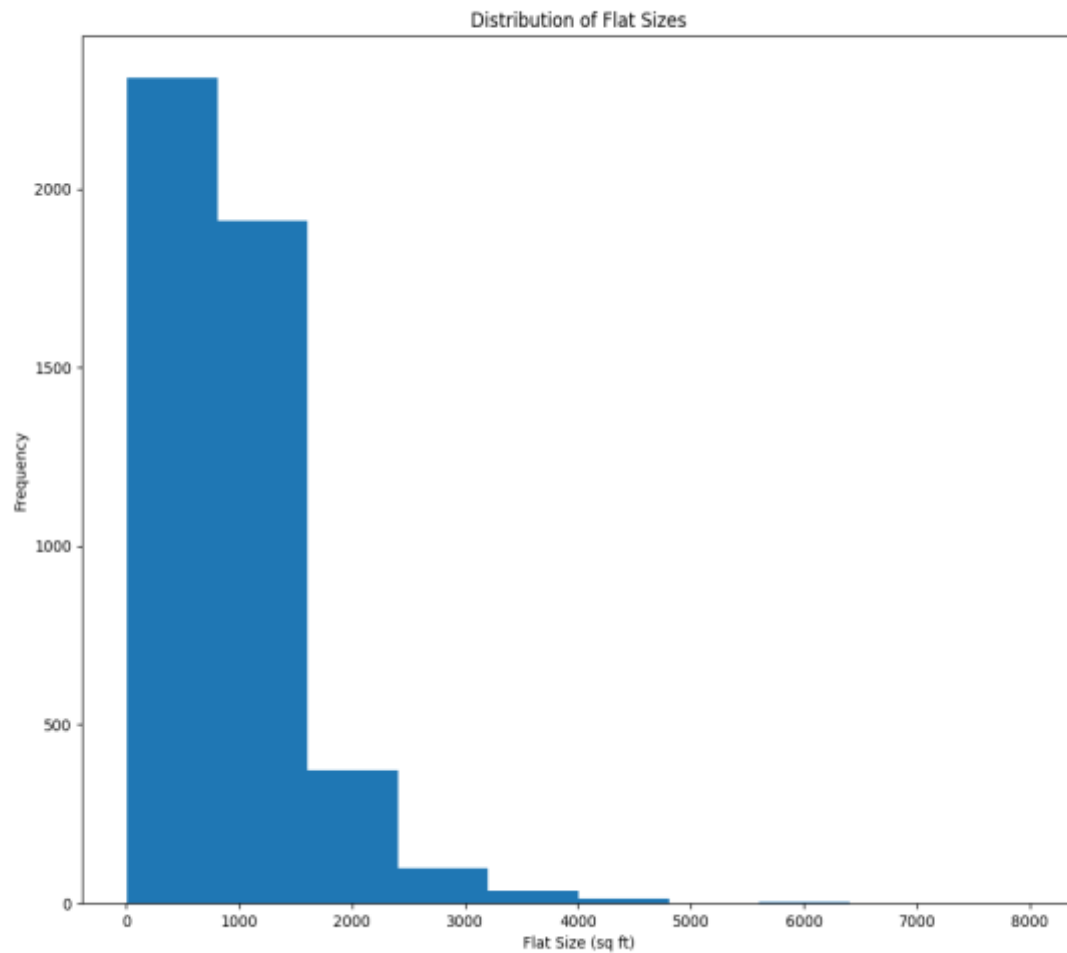
Histogram

- Create a histogram to show the distribution of flat sizes ("Size" column).
- Use 10 bins for the histogram.
- Add title: "Distribution of Flat Sizes".
- Label the x-axis as "Size (sq ft)" and the y-axis as "Frequency"

INPUT

```
fig, axes = plt.subplots(figsize = (12, 10))
axes.hist(x=data['Size'], bins=10, range=None, histtype='bar')
axes.set(xlabel='Flat Size (sq ft)', ylabel='Frequency', title='Distribution of Flat Sizes')
fig.savefig('Task4.png')
```

OUTPUT



TASK 5

Stacked Bar Chart

- o Create stacked bar chart to display the count of flats by furnishing status for each city.
- o Use different colours for each furnishing status.
- o Title: "Flat Count by Furnishing Status and City"
- o Label the x-axis as "City" and the y-axis as "Flat Count".

INPUT

```
import matplotlib.pyplot as plt  
import pandas as pd
```

```
# Group and count
```

```
furnishing_by_city = data.groupby(['City', 'Furnishing Status']).size().unstack(fill_value=0)
```

```
# Plot stacked bar chart
```

```
furnishing_by_city.plot(
```

```
    kind='bar',
```

```
    stacked=True,
```

```
    figsize=(10, 6),
```

```
    color=['#FF9999', '#66B2FF', '#99FF99', '#FFD966']
```

```
)
```

```
# Add labels and title
```

```
plt.title("Flat Count by Furnishing Status and City", fontsize=14)
```

```
plt.xlabel("City", fontsize=12)
```

```
plt.ylabel("Flat Count", fontsize=12)
```

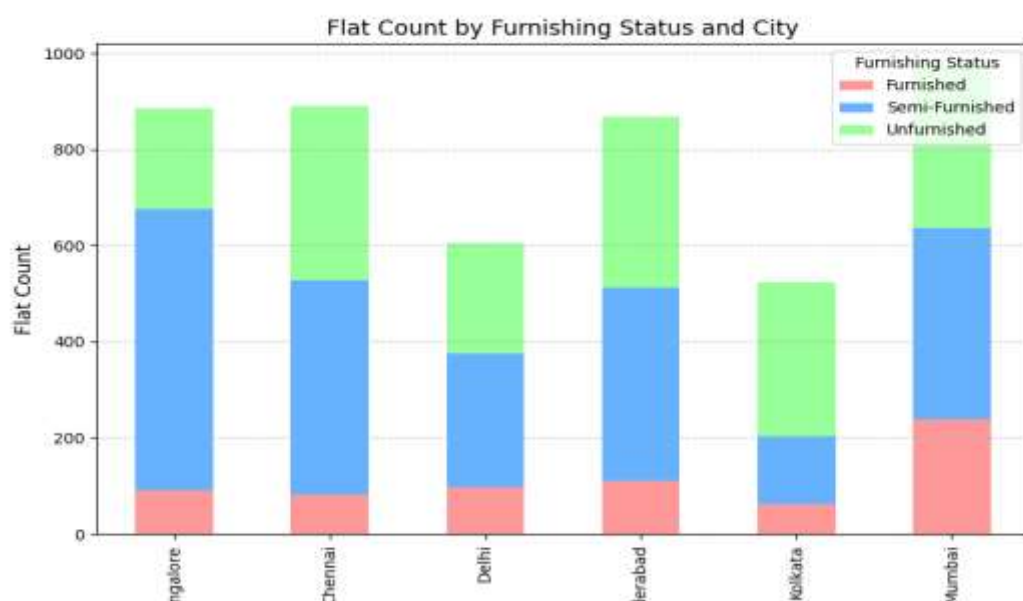
```
plt.legend(title="Furnishing Status")
```

```
plt.grid(axis='y', linestyle='--', alpha=0.7)
```

```
plt.savefig("Task5.png")
```

```
plt.show()
```

OUTPUT



TASK 6

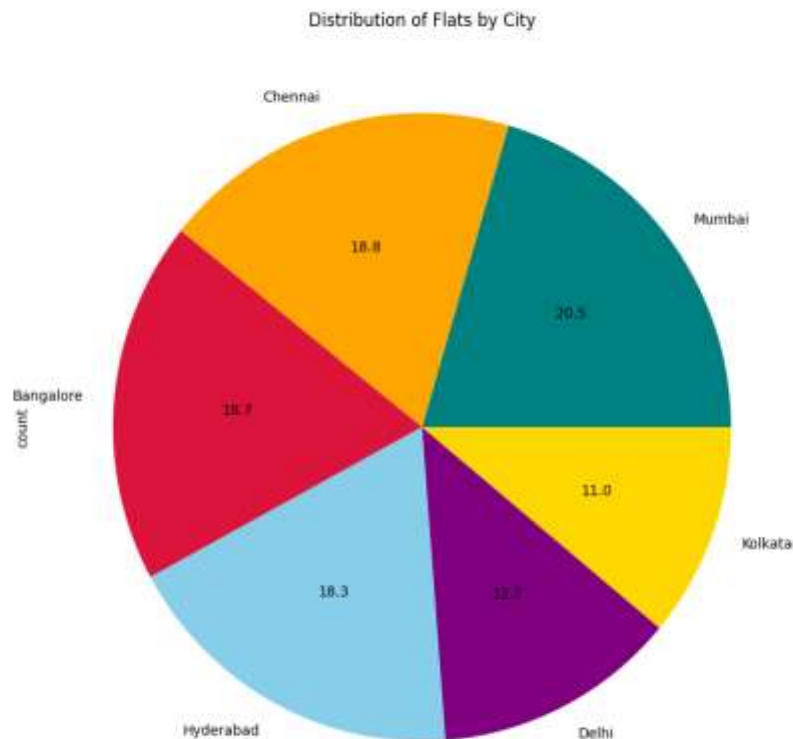
Pie Chart

- Create a pie chart to show the proportion of flats in different cities ("City" column).
- Use a different color for each segment.
- Add percentage labels to each segment.
- Title: "Distribution of Flats by City"

INPUT

```
flats_distribution = data['City'].value_counts()
flats_distribution.plot(kind = 'pie', figsize = (12, 10), autopct = "%.1f",
                        colors=['teal', 'orange', 'crimson', 'skyblue', 'purple', 'gold'])
plt.title("Distribution of Flats by City")
plt.savefig("Task6.png")
plt.show()
```

OUTPUT



TASK 7

Stacked Plot

- Create a stacked plot to visualize rent, size, and number of bathrooms for the first 10 rows of the dataset.
- Title: "Stacked Plot of Rent, Size, and Bathrooms"
- Label the x-axis as "Index" and the y-axis as "Values".
- Use different colors for each variable

INPUT

```
stacked_data = data.head(10)
```

```
plt.figure(figsize = (12, 10))
```

```
plt.stackplot(stacked_data.index, stacked_data['Rent'], stacked_data['Size'],
stacked_data['Bathroom'], labels=['Rent', 'Size', 'Bathroom'], color=['teal', 'crimson',
'orange'])
```



```
plt.title("Stacked Plot of Rent, Size, and Bathrooms")  
plt.xlabel("Index")  
plt.savefig("Task7.png")  
plt.ylabel("Values")
```

OUTPUT

