

```

# =====
# Install required libraries
# =====
!pip install openai plotly pandas kaleido
! pip install -U kaleido

# =====
# Imports
# =====
import os
import pandas as pd
import plotly.express as px
from openai import OpenAI
from google.colab import files
import getpass
import uuid

# =====
# Step 1: Enter OpenAI API key securely
# =====
api_key = getpass.getpass("🔑 Enter your OpenAI API key: ")
client = OpenAI(api_key=api_key)

# =====
# Step 2: Upload dataset (CSV)
# =====
print("\n📁 Please upload your CSV dataset:")
uploaded = files.upload()
filename = list(uploaded.keys())[0]
df = pd.read_csv(filename)

print("\n✅ Dataset loaded successfully!")
print("Shape:", df.shape)
print("Columns:", list(df.columns))
display(df.head())

# =====
# Step 3: Interactive query loop
# =====
while True:
    user_query = input("\n💬 Enter your visualization query (or type 'exit' to quit): ")
    if user_query.lower() in ["exit", "quit"]:
        print("👋 Exiting...")
        break

    # Prompt to GenAI
    prompt = f"""
    You are a Python data visualization assistant.
    Generate valid Plotly code (only the code, no explanation).
    Dataset is already loaded into a pandas DataFrame called df.
    Columns available: {list(df.columns)}.
    User query: {user_query}
    """

    try:

```

```
response = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role": "user", "content": prompt}]
)

generated_code = response.choices[0].message.content

# 🟢 FIX: remove ```python ... ``` wrappers
generated_code = generated_code.replace("```python", "").replace("```", "").strip()

print("\n⚡ Generated Code:\n")
print(generated_code)

# Execute the cleaned code
exec(generated_code)

# Try to save the figure if 'fig' exists
if "fig" in locals():
    file_id = uuid.uuid4().hex[:6]
    png_file = f"chart_{file_id}.png"
    html_file = f"chart_{file_id}.html"

    fig.write_image(png_file)  # Save as PNG
    fig.write_html(html_file)  # Save as interactive HTML

    print(f"📊 Chart saved as {png_file} and {html_file}")

except Exception as e:
    print("❌ Error:", e)
```



```
Requirement already satisfied: openai in /usr/local/lib/python3.12/dist-packages (1.107.0)
Requirement already satisfied: plotly in /usr/local/lib/python3.12/dist-packages (5.24.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: kaleido in /usr/local/lib/python3.12/dist-packages (1.1.0)
Requirement already satisfied: anyio<5,>=3.5.0 in /usr/local/lib/python3.12/dist-packages (from openai) (4.10.0)
Requirement already satisfied: distro<2,>=1.7.0 in /usr/local/lib/python3.12/dist-packages (from openai) (1.9.0)
Requirement already satisfied: httpx<1,>=0.23.0 in /usr/local/lib/python3.12/dist-packages (from openai) (0.28.1)
Requirement already satisfied: jiter<1,>=0.4.0 in /usr/local/lib/python3.12/dist-packages (from openai) (0.10.0)
Requirement already satisfied: pydantic<3,>=1.9.0 in /usr/local/lib/python3.12/dist-packages (from openai) (2.11.7)
Requirement already satisfied: sniffio in /usr/local/lib/python3.12/dist-packages (from openai) (1.3.1)
Requirement already satisfied: tqdm>4 in /usr/local/lib/python3.12/dist-packages (from openai) (4.67.1)
Requirement already satisfied: typing-extensions<5,>=4.11 in /usr/local/lib/python3.12/dist-packages (from openai) (4.15.0)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.12/dist-packages (from plotly) (8.5.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from plotly) (25.0)
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: choreographer>=1.0.10 in /usr/local/lib/python3.12/dist-packages (from kaleido) (1.1.0)
Requirement already satisfied: logistro>=1.0.8 in /usr/local/lib/python3.12/dist-packages (from kaleido) (1.1.0)
Requirement already satisfied: orjson>=3.10.15 in /usr/local/lib/python3.12/dist-packages (from kaleido) (3.11.3)
Requirement already satisfied: pytest-timeout>=2.4.0 in /usr/local/lib/python3.12/dist-packages (from kaleido) (2.4.0)
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.12/dist-packages (from anyio<5,>=3.5.0->openai) (3.10)
Requirement already satisfied: simplejson>=3.19.3 in /usr/local/lib/python3.12/dist-packages (from choreographer>=1.0.10->kaleido) (3.20.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from httpx<1,>=0.23.0->openai) (2025.8.3)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.12/dist-packages (from httpx<1,>=0.23.0->openai) (1.0.9)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.12/dist-packages (from httpcore==1.*->httpx<1,>=0.23.0->openai) (0.16.0)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic<3,>=1.9.0->openai) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.12/dist-packages (from pydantic<3,>=1.9.0->openai) (2.33.2)
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.12/dist-packages (from pydantic<3,>=1.9.0->openai) (0.4.1)
Requirement already satisfied: pytest>=7.0.0 in /usr/local/lib/python3.12/dist-packages (from pytest-timeout>=2.4.0->kaleido) (8.4.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Requirement already satisfied: iniconfig>=1 in /usr/local/lib/python3.12/dist-packages (from pytest>=7.0.0->pytest-timeout>=2.4.0->kaleido) (2.1.0)
Requirement already satisfied: pluggy<2,>=1.5 in /usr/local/lib/python3.12/dist-packages (from pytest>=7.0.0->pytest-timeout>=2.4.0->kaleido) (1.6.0)
Requirement already satisfied: pygments>=2.7.2 in /usr/local/lib/python3.12/dist-packages (from pytest>=7.0.0->pytest-timeout>=2.4.0->kaleido) (2.19.2)
Requirement already satisfied: kaleido in /usr/local/lib/python3.12/dist-packages (1.1.0)
Requirement already satisfied: choreographer>=1.0.10 in /usr/local/lib/python3.12/dist-packages (from kaleido) (1.1.0)
Requirement already satisfied: logistro>=1.0.8 in /usr/local/lib/python3.12/dist-packages (from kaleido) (1.1.0)
Requirement already satisfied: orjson>=3.10.15 in /usr/local/lib/python3.12/dist-packages (from kaleido) (3.11.3)
Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from kaleido) (25.0)
Requirement already satisfied: pytest-timeout>=2.4.0 in /usr/local/lib/python3.12/dist-packages (from kaleido) (2.4.0)
Requirement already satisfied: simplejson>=3.19.3 in /usr/local/lib/python3.12/dist-packages (from choreographer>=1.0.10->kaleido) (3.20.1)
Requirement already satisfied: pytest>=7.0.0 in /usr/local/lib/python3.12/dist-packages (from pytest-timeout>=2.4.0->kaleido) (8.4.2)
Requirement already satisfied: iniconfig>=1 in /usr/local/lib/python3.12/dist-packages (from pytest>=7.0.0->pytest-timeout>=2.4.0->kaleido) (2.1.0)
Requirement already satisfied: pluggy<2,>=1.5 in /usr/local/lib/python3.12/dist-packages (from pytest>=7.0.0->pytest-timeout>=2.4.0->kaleido) (1.6.0)
Requirement already satisfied: pygments>=2.7.2 in /usr/local/lib/python3.12/dist-packages (from pytest>=7.0.0->pytest-timeout>=2.4.0->kaleido) (2.19.2)
```

🔑 Enter your OpenAI API key: .....

📁 Please upload your CSV dataset:

No file chosen      Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.  
Saving airbnb\_sample\_dataset.csv to airbnb\_sample\_dataset (1).csv

✅ Dataset loaded successfully!

Shape: (8, 7)

Columns: ['Listing\_ID', 'City', 'Room\_Type', 'Price\_per\_Night', 'Number\_of\_Reviews', 'Rating', 'Availability\_365']

	Listing_ID	City	Room_Type	Price_per_Night	Number_of_Reviews	Rating	Availability_365
0	101	Mumbai	Entire home/apt	3500	120	4.5	200
1	102	Pune	Private room	1200	45	4.2	150
2	103	Delhi	Entire home/apt	2800	98	4.7	300

	Room Type	Price per Night	Average Rating
3	104 Bangalore Shared room	800	30 3.9
4	105 Hyderabad Entire home/apt	2500	76 4.6

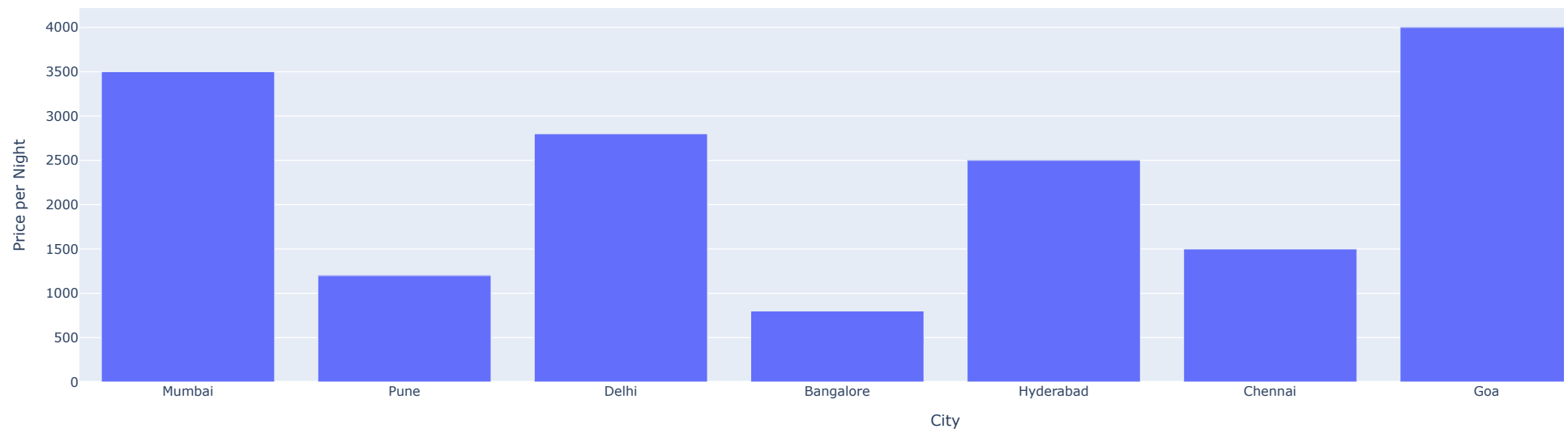
Enter your visualization query (or type 'exit' to quit): bar chart City vs Price\_per\_Night

Generated Code:

```
import plotly.express as px

fig = px.bar(df, x='City', y='Price_per_Night', title='City vs Price per Night',
             labels={'Price_per_Night': 'Price per Night', 'City': 'City'})
fig.show()
```

City vs Price per Night



**Error:**  
Image export using the "kaleido" engine requires the kaleido package,  
which can be installed using pip:  
\$ pip install -U kaleido

Enter your visualization query (or type 'exit' to quit): barchart Average Rating vs Room Type in yellow theme

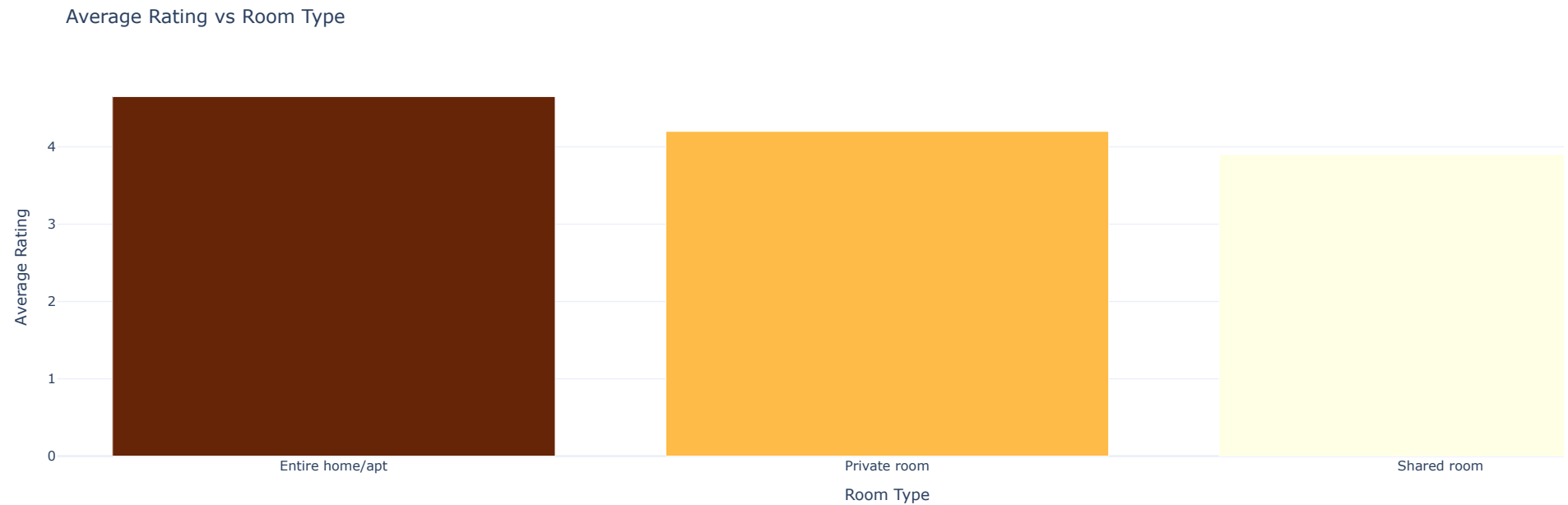
Generated Code:

```
import plotly.express as px

fig = px.bar(df.groupby('Room_Type')['Rating'].mean().reset_index(),
             x='Room_Type',
             y='Rating',
             color='Rating',
             color_continuous_scale='YlOrBr',
             title='Average Rating vs Room Type')
```

```
title= 'Average Rating vs Room Type',
labels={'Rating': 'Average Rating'},
template='plotly_white')
```

```
fig.update_layout(yaxis_title='Average Rating', xaxis_title='Room Type')
fig.show()
```



✖ Error:  
Image export using the "kaleido" engine requires the kaleido package,  
which can be installed using pip:  
\$ pip install -U kaleido

🔗 Enter your visualization query (or type 'exit' to quit): Area Charts Average Reviews per City over Availability

⚡ Generated Code:

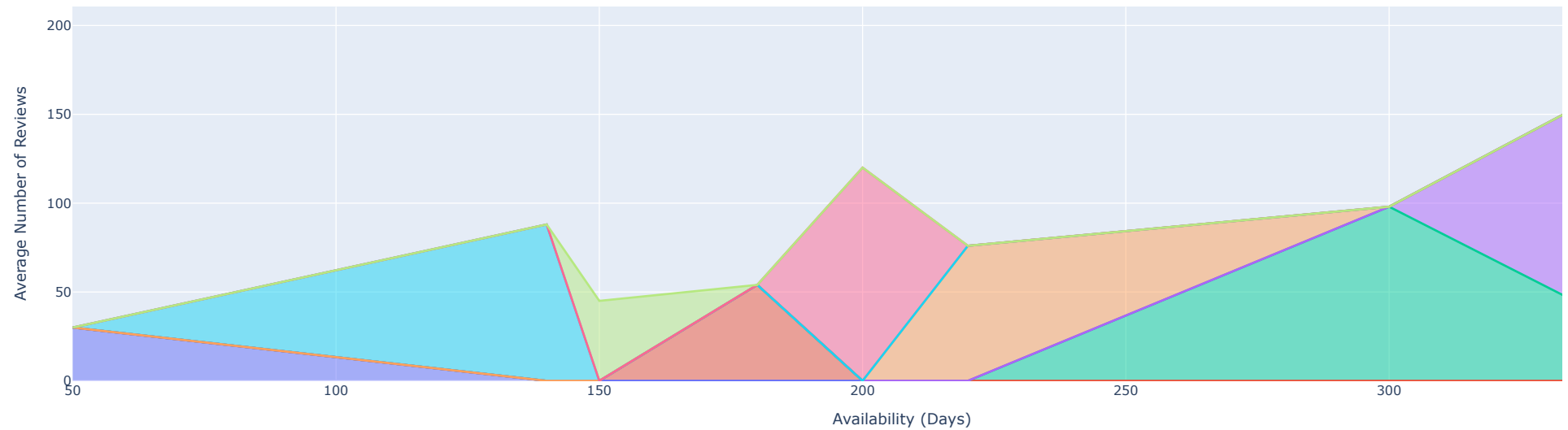
```
import plotly.express as px

# Grouping the data to get average number of reviews per city based on availability
average_reviews = df.groupby('City').agg({'Number_of_Reviews': 'mean', 'Availability_365': 'mean'}).reset_index()

# Creating the area chart
fig = px.area(average_reviews, x='Availability_365', y='Number_of_Reviews',
              title='Average Reviews per City over Availability',
              labels={'Availability_365': 'Availability (Days)', 'Number_of_Reviews': 'Average Number of Reviews'},
              color='City')

fig.show()
```

Average Reviews per City over Availability



✖ Error:  
Image export using the "kaleido" engine requires the kaleido package,  
which can be installed using pip:  
\$ pip install -U kaleido

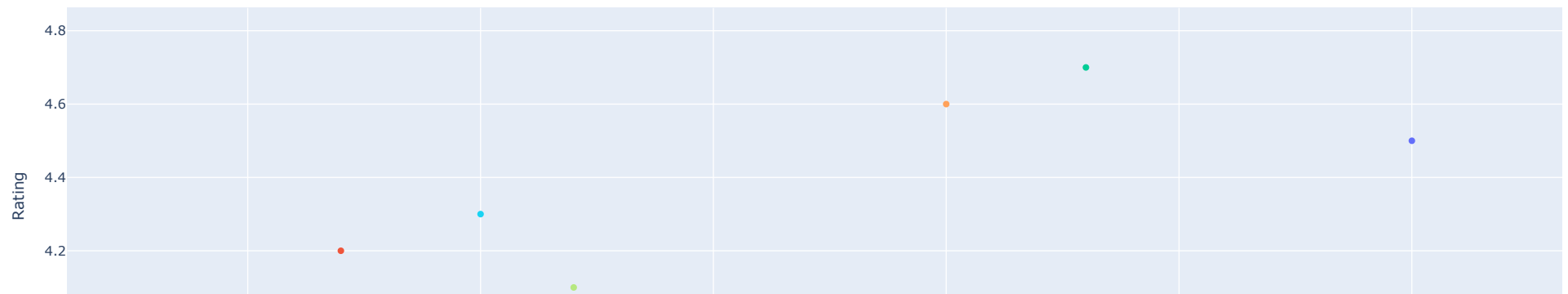
💡 Enter your visualization query (or type 'exit' to quit): Scatter Plots Price per Night vs Rating

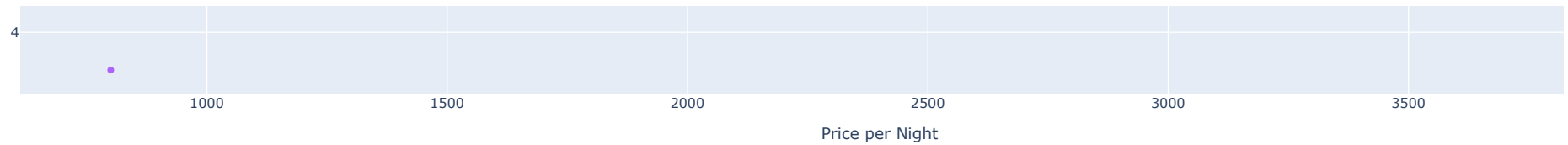
⚡ Generated Code:

```
import plotly.express as px

fig = px.scatter(df, x='Price_per_Night', y='Rating', color='City', title='Price per Night vs Rating')
fig.update_layout(xaxis_title='Price per Night', yaxis_title='Rating')
fig.show()
```

Price per Night vs Rating





✖ Error:  
Image export using the "kaleido" engine requires the kaleido package,  
which can be installed using pip:  
\$ pip install -U kaleido

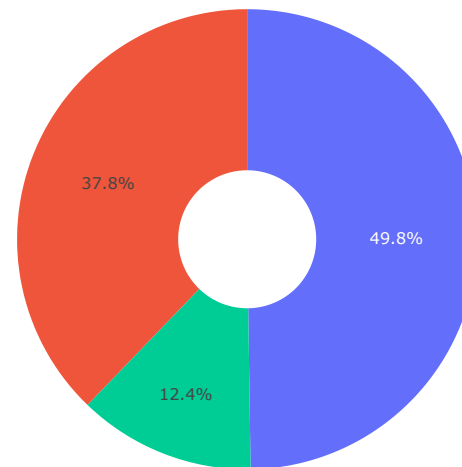
💡 Enter your visualization query (or type 'exit' to quit): Pie chart Proportion of Listings by Room Type

⚡ Generated Code:

```
import plotly.express as px
```

```
fig = px.pie(df, names='Room_Type', values='Listing_ID', title='Proportion of Listings by Room Type', hole=0.3)
fig.show()
```

Proportion of Listings by Room Type



✖ Error:  
Image export using the "kaleido" engine requires the kaleido package,  
which can be installed using pip:  
\$ pip install -U kaleido

💡 Enter your visualization query (or type 'exit' to quit): Proportion of Listings by City

⚡ Generated Code:

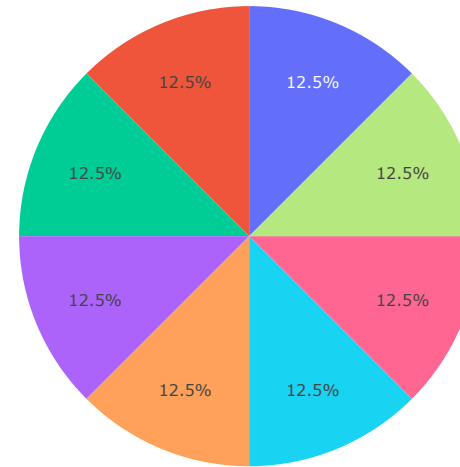


```
import plotly.express as px

city_counts = df['City'].value_counts().reset_index()
city_counts.columns = ['City', 'Count']

fig = px.pie(city_counts, values='Count', names='City', title='Proportion of Listings by City')
fig.show()
```

Proportion of Listings by City



✖ Error:  
Image export using the "kaleido" engine requires the kaleido package,  
which can be installed using pip:  
\$ pip install -U kaleido

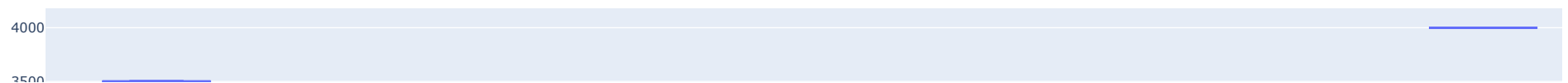
💡 Enter your visualization query (or type 'exit' to quit): Price per Night distribution per City

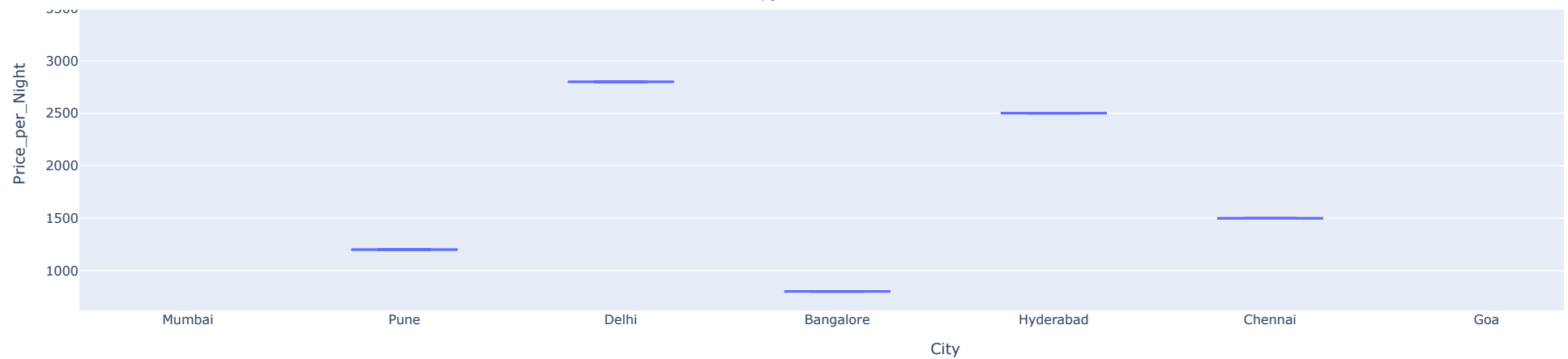
⚡ Generated Code:

```
import plotly.express as px

fig = px.box(df, x='City', y='Price_per_Night', title='Price per Night Distribution per City')
fig.show()
```

Price per Night Distribution per City





✖ Error:  
Image export using the "kaleido" engine requires the kaleido package,  
which can be installed using pip:  
\$ pip install -U kaleido

💡 Enter your visualization query (or type 'exit' to quit): Rating distribution per Room Type

⚡ Generated Code:

```
import plotly.express as px
```

```
fig = px.box(df, x='Room_Type', y='Rating', title='Rating Distribution per Room Type')  
fig.show()
```

Rating Distribution per Room Type

