```
!pip install pandas numpy matplotlib seaborn scikit-learn
```

```
Requirement already satisfied: pandas in c:\users\lenovo\anaconda3\
lib\site-packages (2.2.2)
Requirement already satisfied: numpy in c:\users\lenovo\anaconda3\lib\
site-packages (1.26.4)
Requirement already satisfied: matplotlib in c:\users\lenovo\
anaconda3\lib\site-packages (3.9.2)
Requirement already satisfied: seaborn in c:\users\lenovo\anaconda3\
lib\site-packages (0.13.2)
Requirement already satisfied: scikit-learn in c:\users\lenovo\
anaconda3\lib\site-packages (1.5.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\
lenovo\anaconda3\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\lenovo\
anaconda3\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\lenovo\
anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\lenovo\
anaconda3\lib\site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\lenovo\
anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\lenovo\
anaconda3\lib\site-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\lenovo\
anaconda3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\lenovo\
anaconda3\lib\site-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in c:\users\lenovo\anaconda3\
lib\site-packages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\lenovo\
anaconda3\lib\site-packages (from matplotlib) (3.1.2)
Requirement already satisfied: scipy>=1.6.0 in c:\users\lenovo\
anaconda3\lib\site-packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\lenovo\
anaconda3\lib\site-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\
lenovo\anaconda3\lib\site-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: six>=1.5 in c:\users\lenovo\anaconda3\
lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
```

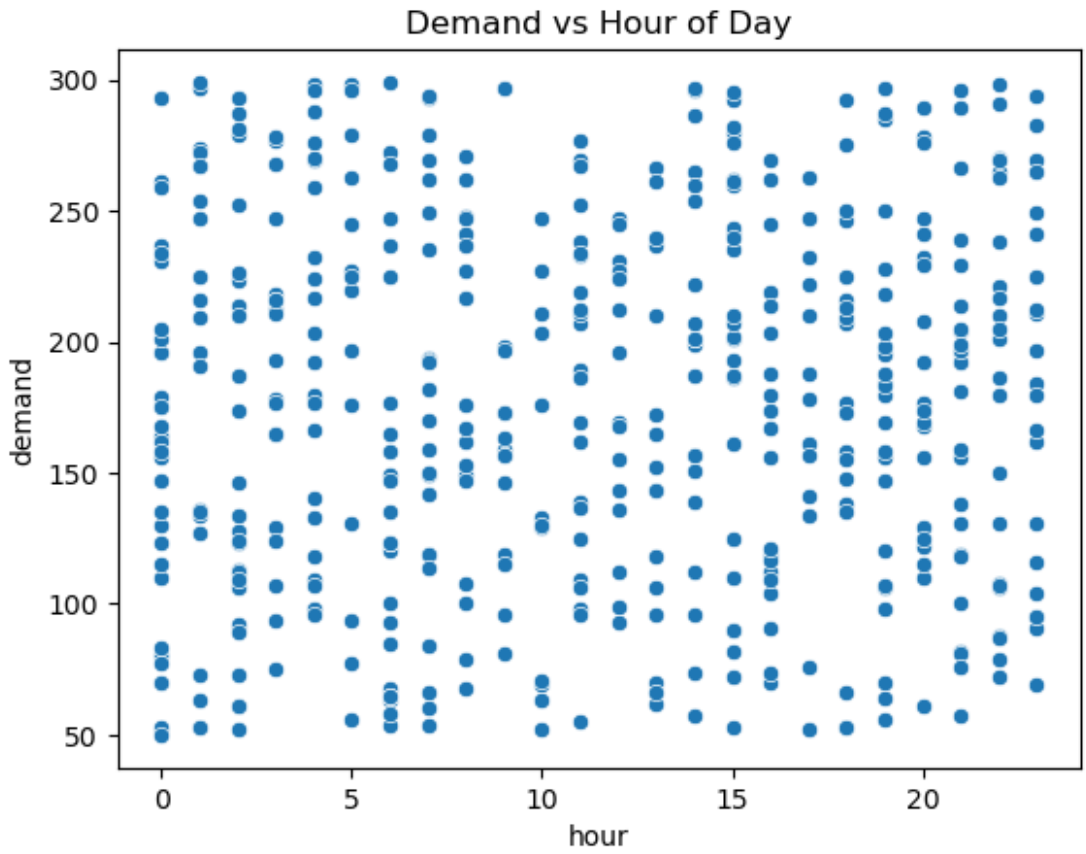
```
# Generate fake Uber demand data
np.random.seed(42)
num_samples = 500

data = pd.DataFrame({
    'hour': np.random.randint(0, 24, size=num_samples),               #
Hour of the day
    'day_of_week': np.random.randint(0, 7, size=num_samples),         #
0 = Monday, 6 = Sunday
    'location_id': np.random.randint(1, 6, size=num_samples),         #
5 different city areas
    'temperature': np.random.normal(30, 5, size=num_samples),         #
Random temperature
    'rain': np.random.choice([0, 1], size=num_samples),               #
1 = Rainy, 0 = Not rainy
    'demand': np.random.randint(50, 300, size=num_samples)            #
Number of ride requests
})

data.head()
```

	hour	day_of_week	location_id	temperature	rain	demand
0	6	6	2	17.353313	0	237
1	19	1	2	36.490889	1	106
2	14	0	3	37.220865	0	222
3	10	3	3	20.477440	1	69
4	7	2	1	30.877105	1	66

```
sns.scatterplot(x='hour', y='demand', data=data)
plt.title('Demand vs Hour of Day')
plt.show()
```




```
X = data[['hour', 'day_of_week', 'location_id', 'temperature',
'rain']]
y = data['demand']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)
```

```
LinearRegression()
```

```
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse:.2f}")
```

```
Mean Squared Error: 4687.50
```

```
# Predict for 5 pm on Friday at Location 3, 32°C, no rain
sample = pd.DataFrame({
    'hour': [17],
    'day_of_week': [4],   # 0=Monday, 4=Friday
    'location_id': [3],
    'temperature': [32],
```

```
    'rain': [0]
})

predicted_demand = model.predict(sample)
print(f"🚖 Predicted Uber Demand: {int(predicted_demand[0])} rides")
```

```
🚖 Predicted Uber Demand: 185 rides
```