In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.linear_model import LinearRegression
```

In [2]:

```
df = pd.read_csv("orders.csv")
df.head()
```

Out[2]:

	date	product_category	price	orders
0	2024-01-01	Electronics	1200	45
1	2024-01-02	Electronics	1150	38
2	2024-01-03	Clothing	800	60
3	2024-01-04	Clothing	750	72
4	2024-01-05	Grocery	300	120

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 4 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   date              7 non-null      object
 1   product_category  7 non-null      object
 2   price             7 non-null      int64 
 3   orders            7 non-null      int64 
dtypes: int64(2), object(2)
memory usage: 356.0+ bytes
```

In [4]:

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

Out[4]:

```
date                0
product_category    0
price               0
orders              0
dtype: int64
```

In [5]:

```
df['date'] = pd.to_datetime(df['date'])
df['day'] = df['date'].dt.day
df['month'] = df['date'].dt.month
df['day_of_week'] = df['date'].dt.dayofweek
```

In [6]:

```
daily_orders = df.groupby('date').size().reset_index(name='order_count')
daily_orders.head()
```

Out[6]:

	date	order_count
0	2024-01-01	1
1	2024-01-02	1
2	2024-01-03	1
3	2024-01-04	1
4	2024-01-05	1

In [7]:

```
X = daily_orders[['date']]
y = daily_orders['order_count']
```

In [8]:

```
X['date'] = X['date'].map(pd.Timestamp.toordinal)
```

```
C:\Users\C.Madhubalan\AppData\Local\Temp\ipykernel_9620\1555319093.py:1: SettingW
ithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['date'] = X['date'].map(pd.Timestamp.toordinal)
```

In [9]:

```
X = daily_orders[['date']].copy()
```

In [10]:

```
X['date'] = X['date'].map(pd.Timestamp.toordinal)
```

In [11]:

```
from sklearn.model_selection import train_test_split

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

In [12]:

```
from sklearn.linear_model import LinearRegression

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

Out[12]:

```
▾ LinearRegression  i ?

▶ Parameters
```

In [13]:

```
y_pred = model.predict(X_test)
```

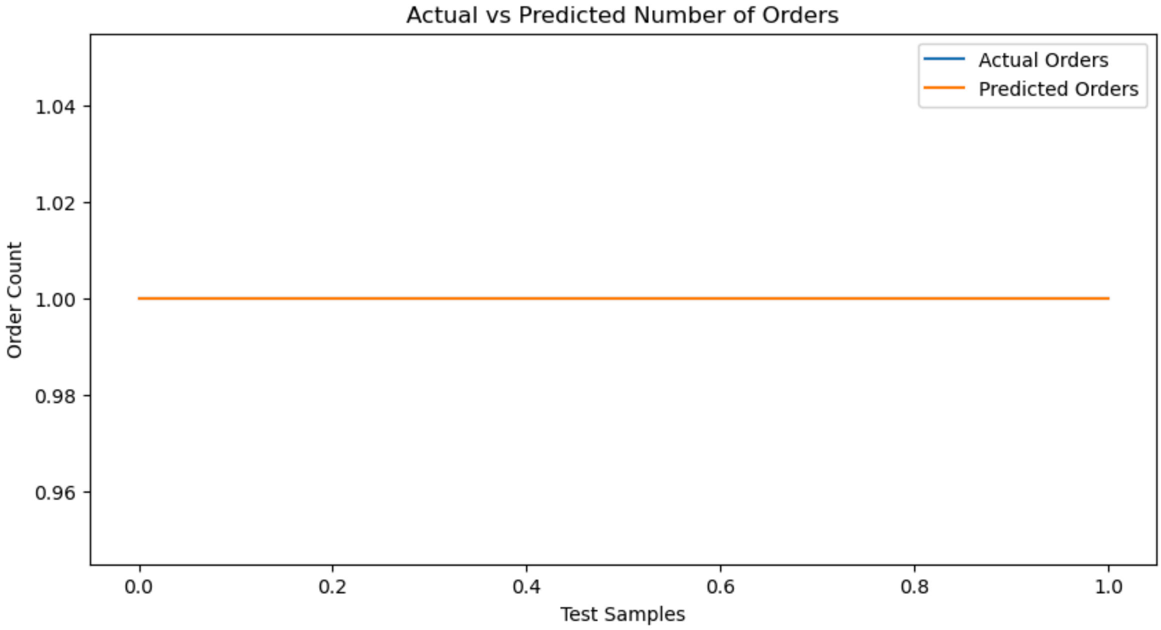
In [14]:

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)

print("MAE :", mae)
print("RMSE:", rmse)
print("R²  :", r2)
```

```
MAE : 0.0
RMSE: 0.0
R²  : 1.0
```

In [15]:

```
import matplotlib.pyplot as plt

plt.figure(figsize=(10,5))
plt.plot(y_test.values, label='Actual Orders')
plt.plot(y_pred, label='Predicted Orders')
plt.legend()
plt.title("Actual vs Predicted Number of Orders")
plt.xlabel("Test Samples")
plt.ylabel("Order Count")
plt.show()
```



In [ ]: