

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: data = {
    "Date": pd.date_range(start="2025-01-01", periods=30),
    "Category": [
        "Groceries", "Transport", "Utilities", "Rent", "Entertainment"
    ] * 6,
    "Amount": [
        500, 120, 300, 8000, 400,
        450, 100, 280, 8000, 350,
        520, 130, 310, 8000, 500,
        480, 110, 290, 8000, 450,
        510, 125, 305, 8000, 420,
        470, 115, 295, 8000, 390
    ]
}

df = pd.DataFrame(data)
df.head()
```

Out[2]:

	Date	Category	Amount
0	2025-01-01	Groceries	500
1	2025-01-02	Transport	120
2	2025-01-03	Utilities	300
3	2025-01-04	Rent	8000
4	2025-01-05	Entertainment	400

In [3]: `df.info()`
`df.describe()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Date        30 non-null    datetime64[ns]
1   Category    30 non-null    object
2   Amount      30 non-null    int64
dtypes: datetime64[ns](1), int64(1), object(1)
memory usage: 852.0+ bytes
```

Out[3]:

	Date	Amount
count	30	30.000000
mean	2025-01-15 12:00:00	1864.000000
min	2025-01-01 00:00:00	100.000000
25%	2025-01-08 06:00:00	291.250000
50%	2025-01-15 12:00:00	410.000000
75%	2025-01-22 18:00:00	507.500000
max	2025-01-30 00:00:00	8000.000000
std	NaN	3123.182616

In [4]: `# Convert Date column to datetime (important)`
`df["Date"] = pd.to_datetime(df["Date"])`

`# Check for missing values`
`df.isnull().sum()`

Out[4]:

	0
Date	0
Category	0
Amount	0

dtype: int64

```
In [5]: daily_expense = df.groupby("Date")["Amount"].sum()

print("Average Daily Expense:", daily_expense.mean())
```

Average Daily Expense: 1864.0

```
In [6]: category_expense = df.groupby("Category")["Amount"].sum()
category_expense
```

Out[6]:

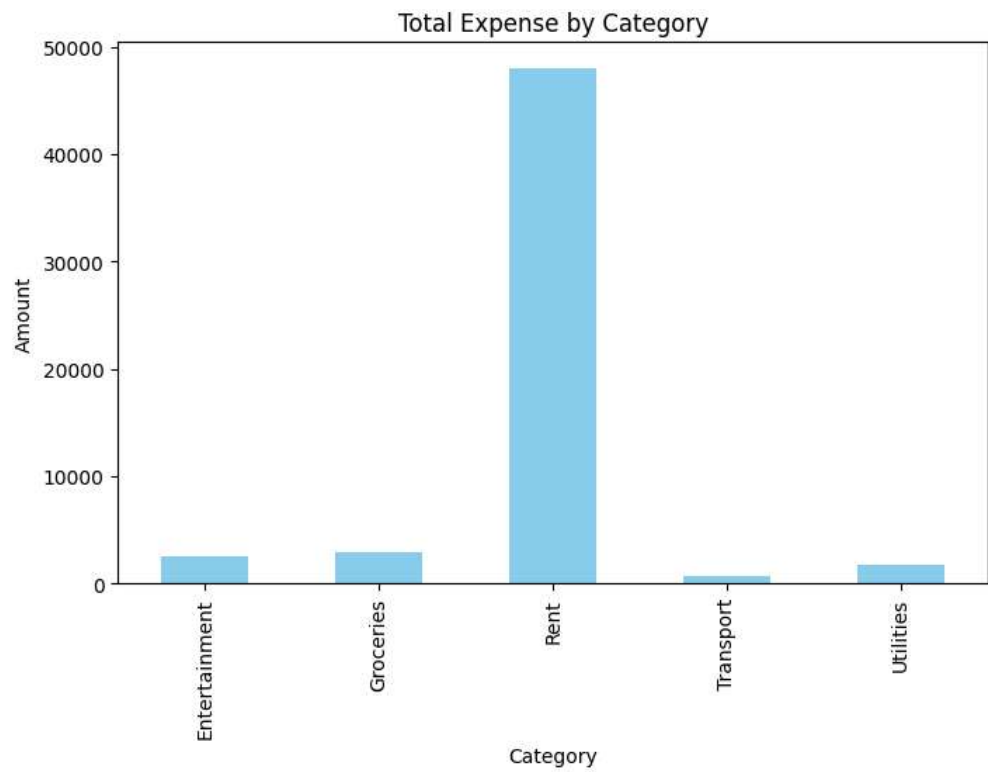
	Amount
Category	
Entertainment	2510
Groceries	2930
Rent	48000
Transport	700
Utilities	1780

dtype: int64

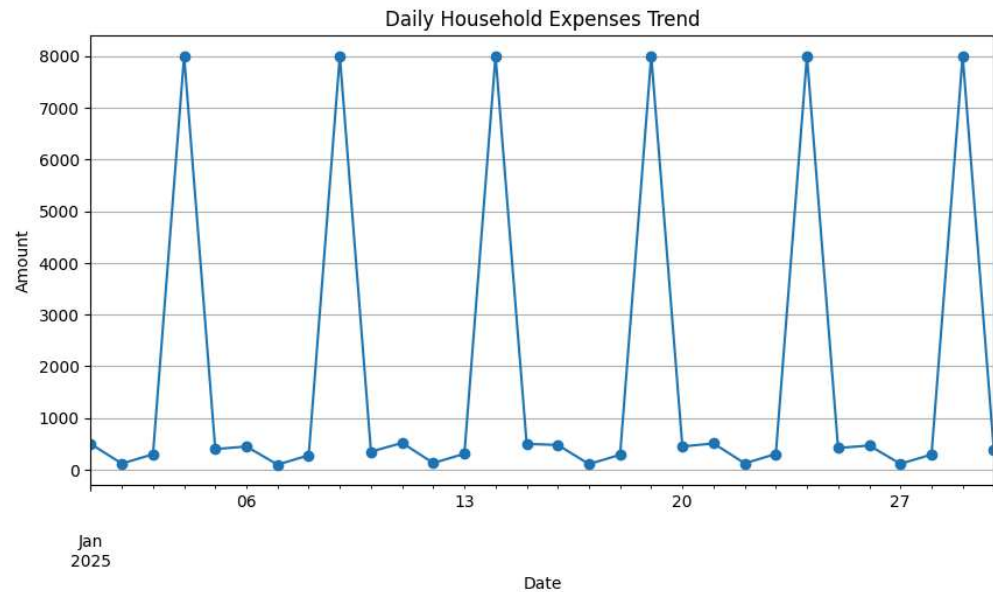
```
In [7]: highest_category = category_expense.idxmax()
print("Highest Spending Category:", highest_category)
```

Highest Spending Category: Rent

```
In [8]: plt.figure(figsize=(8,5))
category_expense.plot(kind="bar", color="skyblue")
plt.title("Total Expense by Category")
plt.ylabel("Amount")
plt.xlabel("Category")
plt.show()
```



```
In [9]: plt.figure(figsize=(10,5))
daily_expense.plot(marker="o")
plt.title("Daily Household Expenses Trend")
plt.ylabel("Amount")
plt.xlabel("Date")
plt.grid(True)
plt.show()
```



```
In [10]: df["Month"] = df["Date"].dt.month

monthly_expense = df.groupby("Month")["Amount"].sum()
monthly_expense
```

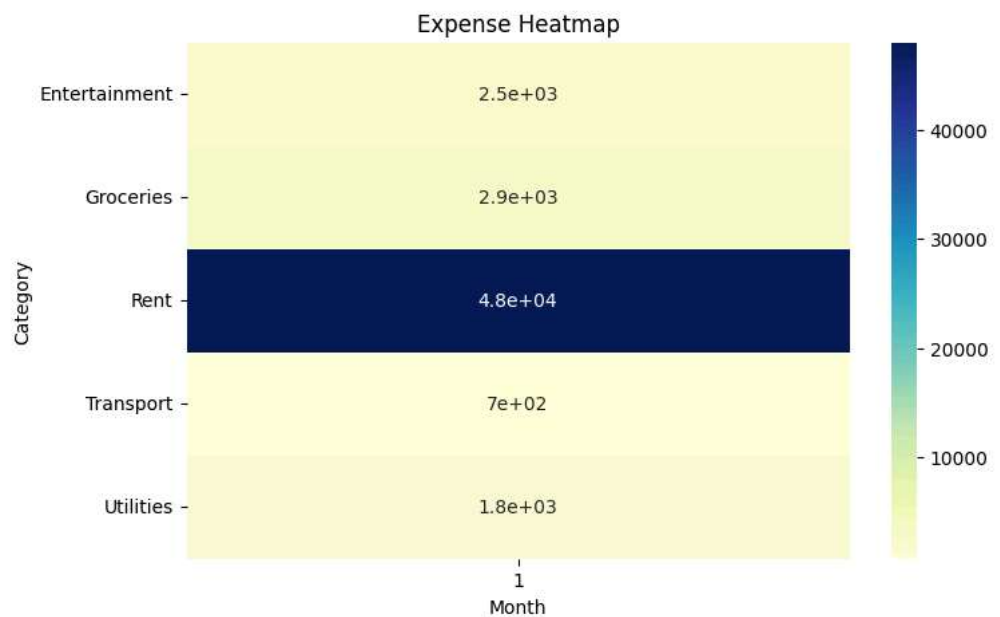
Out[10]:

	Amount
Month	
1	55920

dtype: int64

```
In [11]: pivot_table = pd.pivot_table(
    df,
    values="Amount",
    index="Category",
    columns="Month",
    aggfunc="sum"
)

plt.figure(figsize=(8,5))
sns.heatmap(pivot_table, annot=True, cmap="YlGnBu")
plt.title("Expense Heatmap")
plt.show()
```



Exported with [runcell](#) — convert notebooks to HTML or PDF anytime at [runcell.dev](#).