

# Coffee Data Visualization

July 23, 2024

Data Visualization Coffee Sales

```
[130]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

This dataset contains detailed records of coffee sales from a vending machine. it is intended for analysis of purchasing patterns, sales trends, and customer preferences related to coffee products.

```
[152]: df=pd.read_csv("/home/lenovo/Downloads/new/coffee.csv")
df
```

```
[152]:
```

	date	datetime	cash_type	card	\
0	2024-03-01	2024-03-01 10:15:50.520	card	ANON-0000-0000-0001	
1	2024-03-01	2024-03-01 12:19:22.539	card	ANON-0000-0000-0002	
2	2024-03-01	2024-03-01 12:20:18.089	card	ANON-0000-0000-0002	
3	2024-03-01	2024-03-01 13:46:33.006	card	ANON-0000-0000-0003	
4	2024-03-01	2024-03-01 13:48:14.626	card	ANON-0000-0000-0004	
..	...	...	...	...	
971	2024-07-14	2024-07-14 22:31:29.976	card	ANON-0000-0000-0376	
972	2024-07-15	2024-07-15 07:33:05.557	card	ANON-0000-0000-0377	
973	2024-07-16	2024-07-16 12:23:37.467	card	ANON-0000-0000-0378	
974	2024-07-16	2024-07-16 19:29:25.485	card	ANON-0000-0000-0367	
975	2024-07-17	2024-07-17 13:04:59.744	card	ANON-0000-0000-0379	

	money	coffee_name
0	38.70	Latte
1	38.70	Hot Chocolate
2	38.70	Hot Chocolate
3	28.90	Americano
4	38.70	Latte
..	...	...
971	32.82	Latte
972	32.82	Cappuccino
973	27.92	Americano with Milk

```

974 32.82      Hot Chocolate
975 27.92  Americano with Milk

```

[976 rows x 6 columns]

```
[153]: df=df.rename(columns={'date':'Date','datetime' : 'Date Time' })
df
```

```
[153]:
```

	Date	Date Time	cash_type	card \
0	2024-03-01	2024-03-01 10:15:50.520	card	ANON-0000-0000-0001
1	2024-03-01	2024-03-01 12:19:22.539	card	ANON-0000-0000-0002
2	2024-03-01	2024-03-01 12:20:18.089	card	ANON-0000-0000-0002
3	2024-03-01	2024-03-01 13:46:33.006	card	ANON-0000-0000-0003
4	2024-03-01	2024-03-01 13:48:14.626	card	ANON-0000-0000-0004
..	...	...	...	...
971	2024-07-14	2024-07-14 22:31:29.976	card	ANON-0000-0000-0376
972	2024-07-15	2024-07-15 07:33:05.557	card	ANON-0000-0000-0377
973	2024-07-16	2024-07-16 12:23:37.467	card	ANON-0000-0000-0378
974	2024-07-16	2024-07-16 19:29:25.485	card	ANON-0000-0000-0367
975	2024-07-17	2024-07-17 13:04:59.744	card	ANON-0000-0000-0379

	money	coffee_name
0	38.70	Latte
1	38.70	Hot Chocolate
2	38.70	Hot Chocolate
3	28.90	Americano
4	38.70	Latte
..	...	...
971	32.82	Latte
972	32.82	Cappuccino
973	27.92	Americano with Milk
974	32.82	Hot Chocolate
975	27.92	Americano with Milk

[976 rows x 6 columns]

```
[154]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 976 entries, 0 to 975
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Date             976 non-null   object
1   Date Time        976 non-null   object
2   cash_type        976 non-null   object
3   card             887 non-null   object

```

```

4    money          976 non-null    float64
5    coffee_name    976 non-null    object
dtypes: float64(1), object(5)
memory usage: 45.9+ KB

```

```
[155]: df.describe()
```

```

[155]:          money
count  976.000000
mean    33.879754
std      4.640887
min     23.020000
25%     28.900000
50%     33.800000
75%     37.720000
max     40.000000

```

```

[156]: df.isnull().sum()

value_counts = df['cash_type'].value_counts()
value_counts

```

```

[156]: cash_type
card    887
cash     89
Name: count, dtype: int64

```

```

[162]: new=df.drop(['card'],axis=1)
new

```

```

[162]:
   Date      Date Time  cash_type  money  coffee_name
0  2024-03-01  2024-03-01  10:15:50.520    card  38.70      Latte
1  2024-03-01  2024-03-01  12:19:22.539    card  38.70  Hot Chocolate
2  2024-03-01  2024-03-01  12:20:18.089    card  38.70  Hot Chocolate
3  2024-03-01  2024-03-01  13:46:33.006    card  28.90    Americano
4  2024-03-01  2024-03-01  13:48:14.626    card  38.70      Latte
..      ...
971  2024-07-14  2024-07-14  22:31:29.976    card  32.82      Latte
972  2024-07-15  2024-07-15  07:33:05.557    card  32.82    Cappuccino
973  2024-07-16  2024-07-16  12:23:37.467    card  27.92  Americano with Milk
974  2024-07-16  2024-07-16  19:29:25.485    card  32.82    Hot Chocolate
975  2024-07-17  2024-07-17  13:04:59.744    card  27.92  Americano with Milk

[976 rows x 5 columns]

```

```

[163]: df['coffee_name'].unique()
print(new.columns)

```

```
Index(['Date', 'Date Time', 'cash_type', 'money', 'coffee_name'],
      dtype='object')
```

```
[164]: new['Date'] = pd.to_datetime(new['Date'])
new.set_index('Date',inplace=True)
daily=new.resample('D')['money'].sum()
daily_df=daily.to_frame('daily').reset_index()
```

```
[165]: new['Date Time'] = pd.to_datetime(new['Date Time'])
new['Month'] = new['Date Time'].dt.month
new['Year'] = new['Date Time'].dt.year
```

```
[166]: daily_df
```

```
[166]:
```

	Date	daily
0	2024-03-01	396.30
1	2024-03-02	228.10
2	2024-03-03	349.10
3	2024-03-04	135.20
4	2024-03-05	338.50
..	...	...
134	2024-07-13	65.64
135	2024-07-14	172.42
136	2024-07-15	32.82
137	2024-07-16	60.74
138	2024-07-17	27.92

[139 rows x 2 columns]

```
[167]: daily_df.describe()
```

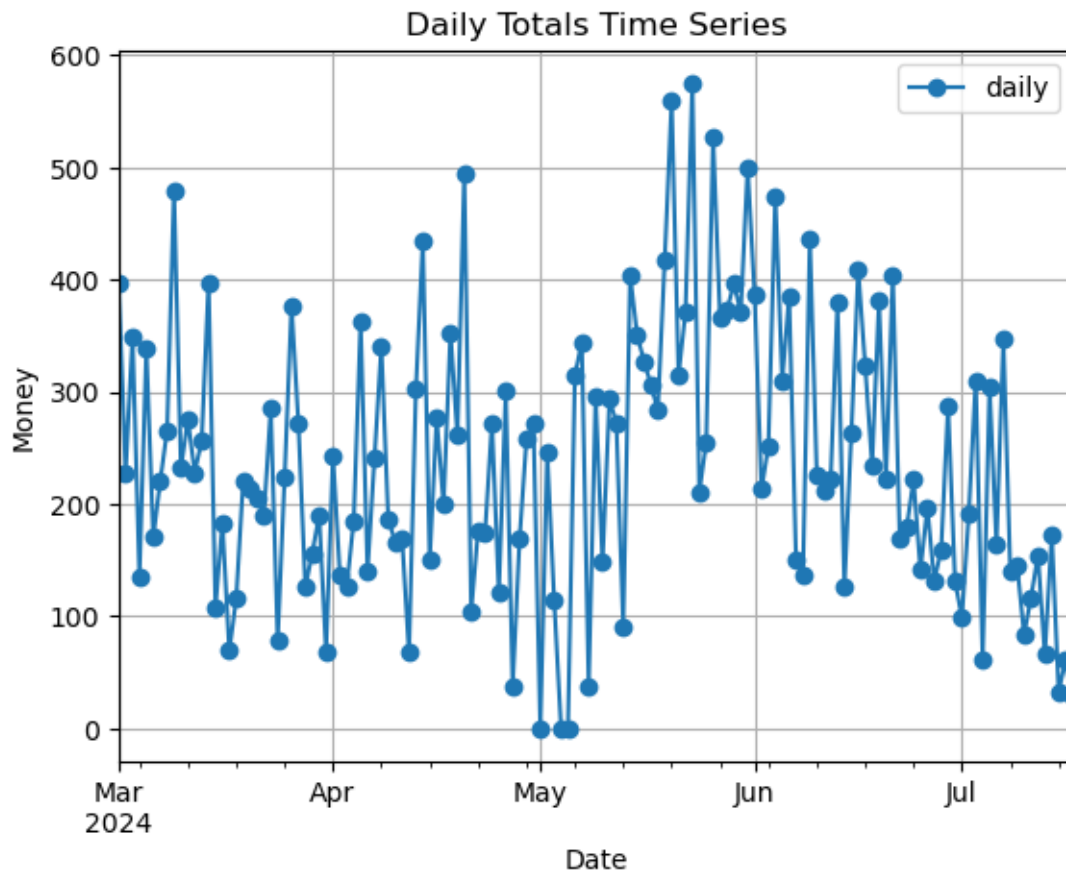
```
[167]:
```

	Date	daily
count	139	139.000000
mean	2024-05-09 00:00:00	237.889496
min	2024-03-01 00:00:00	0.000000
25%	2024-04-04 12:00:00	146.470000
50%	2024-05-09 00:00:00	223.600000
75%	2024-06-12 12:00:00	319.100000
max	2024-07-17 00:00:00	575.400000
std	NaN	123.579342

Graph About Daily Coffee Sales

```
[168]: daily_df.plot(x='Date',y='daily',kind='line',marker='o')
plt.title("Daily Totals Time Series")
plt.xlabel('Date')
plt.ylabel('Money')
plt.grid(True)
```

```
plt.show()
```



Pie chart about Caffee type sales

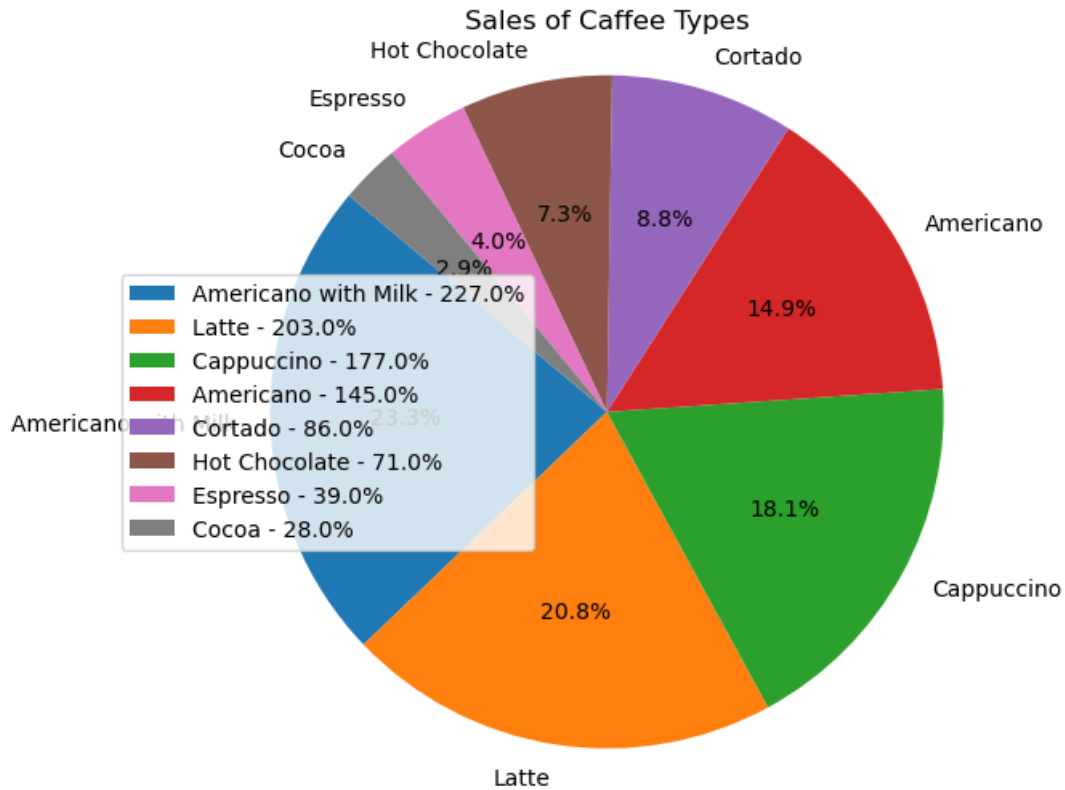
```
[169]: category_counts = new['coffee_name'].value_counts()

labels = category_counts.index
sizes = category_counts.values
plt.figure(figsize=(8, 6))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)

plt.legend(loc='best', labels=['{} - {:.1f}%'.format(label, size) for label,
↪size in zip(labels, sizes)])

plt.title('Sales of Caffee Types')
plt.axis('equal')

plt.show()
```



Pie Chart About Cash type that used to buy coffee

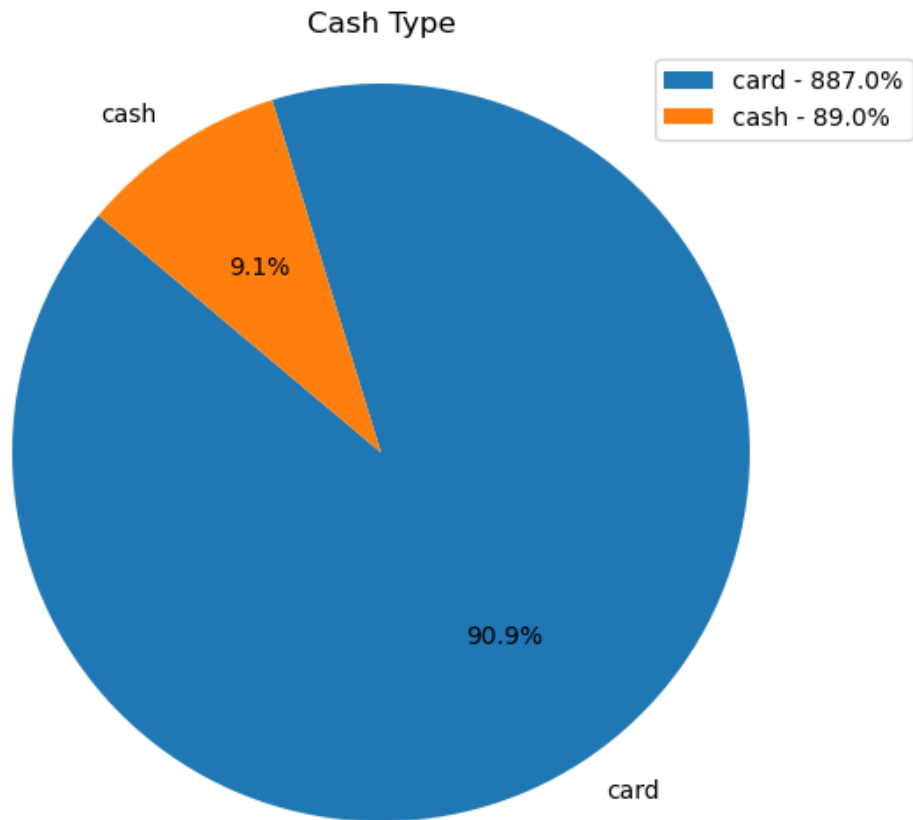
```
[170]: category_counts = new['cash_type'].value_counts()

labels = category_counts.index
sizes = category_counts.values
plt.figure(figsize=(8, 6))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)

plt.legend(loc='best', labels=['{} - {:.1f}%'.format(label, size) for label,
↪size in zip(labels, sizes)])

plt.title('Cash Type')
plt.axis('equal')

plt.show()
```



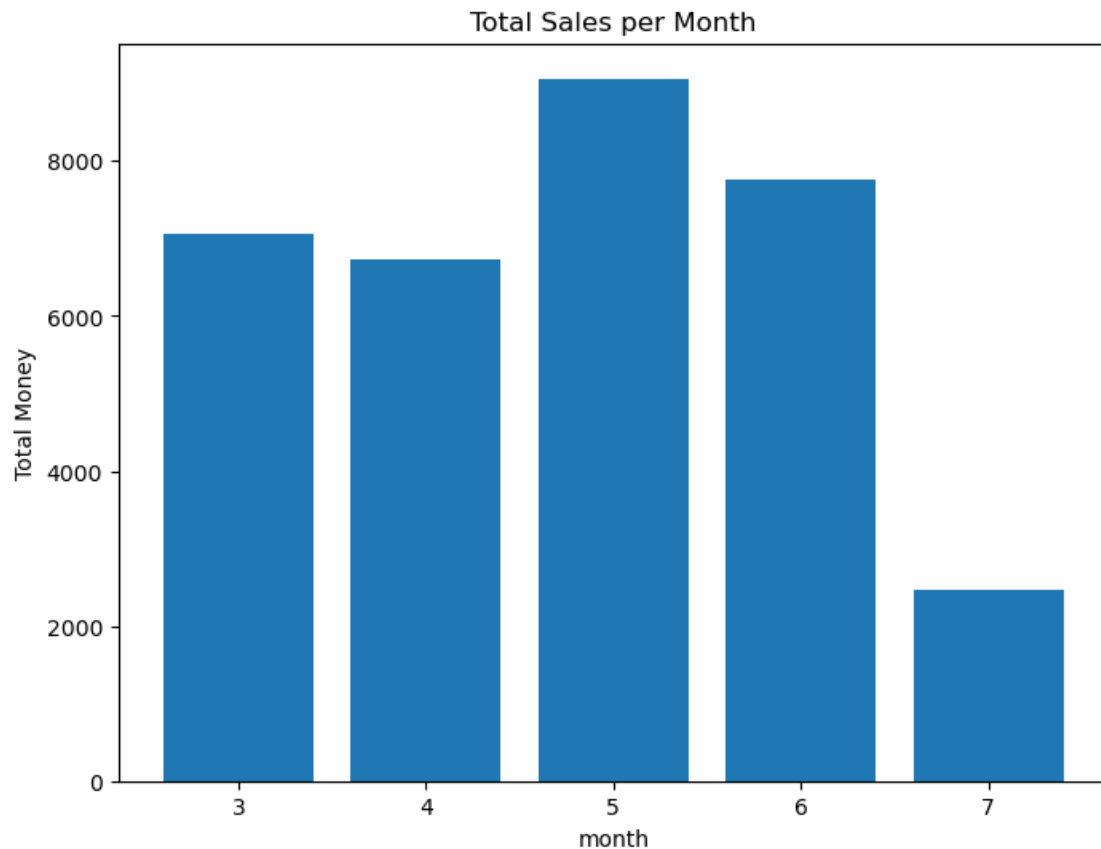
Graph About Sales per Month

```
[171]: summary = new.groupby('Month')['money'].sum().reset_index()
print(summary)
```

	Month	money
0	3	7050.20
1	4	6720.56
2	5	9063.42
3	6	7758.76
4	7	2473.70

```
[194]: plt.figure(figsize=(8, 6))
plt.bar(summary['Month'], summary['money'])
plt.xlabel('month')
plt.ylabel('Total Money')
plt.title('Total Sales per Month')

plt.show()
```



Graph About Sales per Coffee

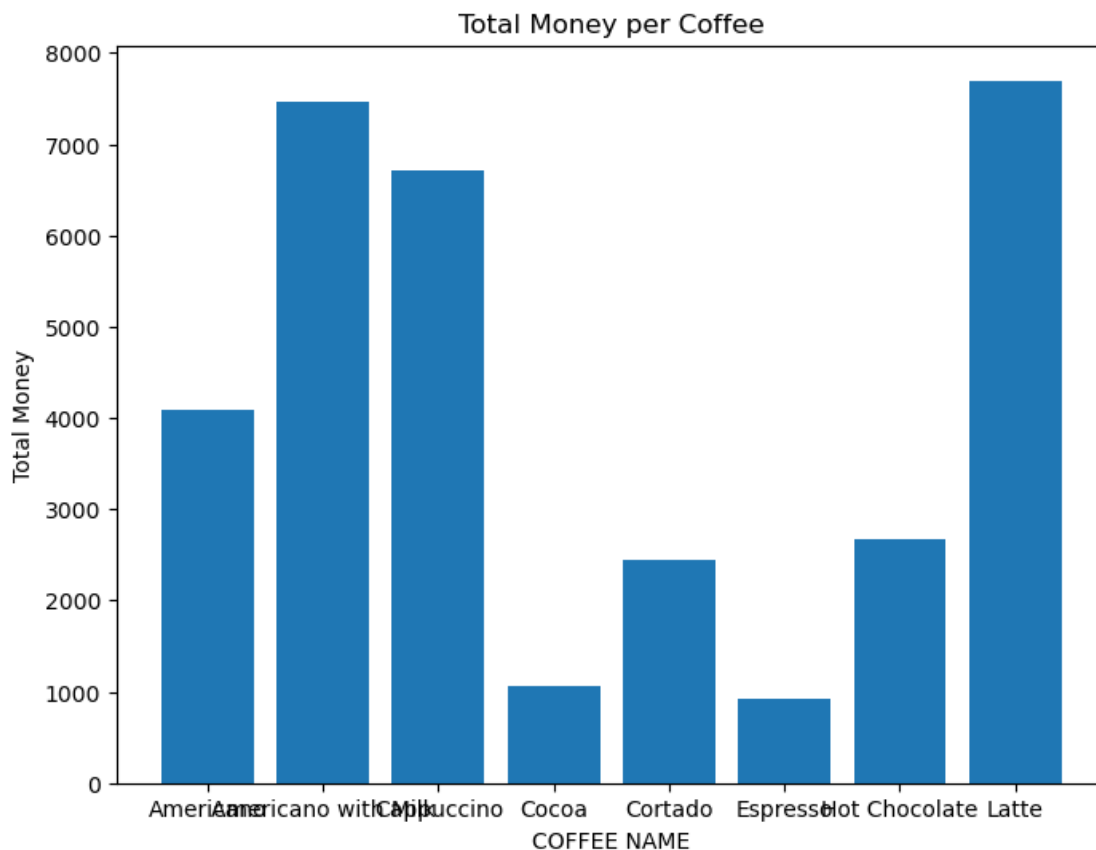
```
[173]: gh=new.groupby('coffee_name')['money'].sum().reset_index()
print(gh)
```

	coffee_name	money
0	Americano	4092.06
1	Americano with Milk	7457.22
2	Cappuccino	6709.56
3	Cocoa	1066.20
4	Cortado	2445.82
5	Espresso	919.42
6	Hot Chocolate	2680.02
7	Latte	7696.34

```
[174]: plt.figure(figsize=(8, 6))
plt.bar(gh['coffee_name'], gh['money'])
plt.xlabel('COFFEE NAME')
plt.ylabel('Total Money')
plt.title('Total Sales per Coffee')
```



```
plt.show()
```

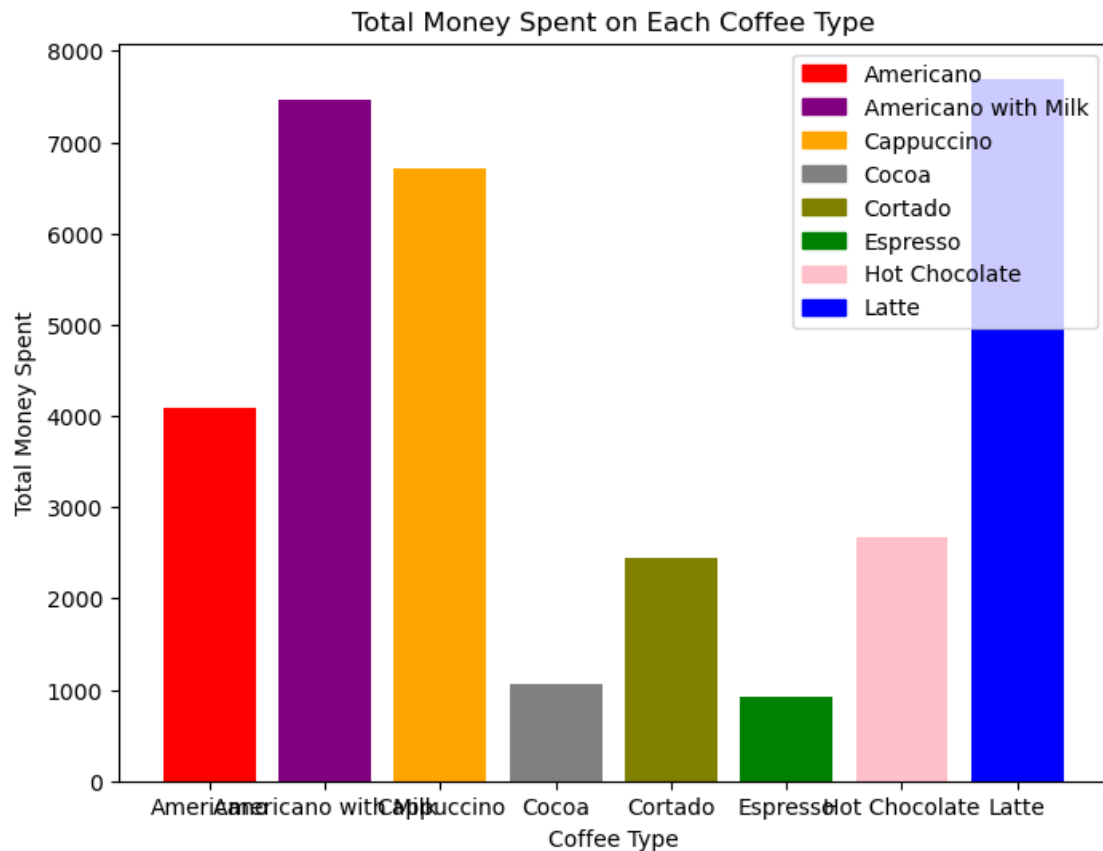


```
[175]: colors = {'Latte': 'blue', 'Espresso': 'green', 'Cappuccino': 'orange', 'Hot_
    ↳Chocolate': 'pink', 'Americano': 'red', 'Americano with Milk': 'purple', 'Cocoa':
    ↳'gray', 'Cortado': 'olive'}

plt.figure(figsize=(8, 6))
bars = plt.bar(gh['coffee_name'], gh['money'], color=[colors[type] for type in_
    ↳gh['coffee_name']])
plt.xlabel('Coffee Type')
plt.ylabel('Total Money Spent')
plt.title('Total Money Spent on Each Coffee Type')

legend_colors = [plt.Rectangle((0,0),1,1, color=colors[type]) for type in_
    ↳gh['coffee_name'].unique()]
plt.legend(legend_colors, gh['coffee_name'].unique())
```

```
[175]: <matplotlib.legend.Legend at 0x7a3fbb1b6fd0>
```



Graph about Sales per week

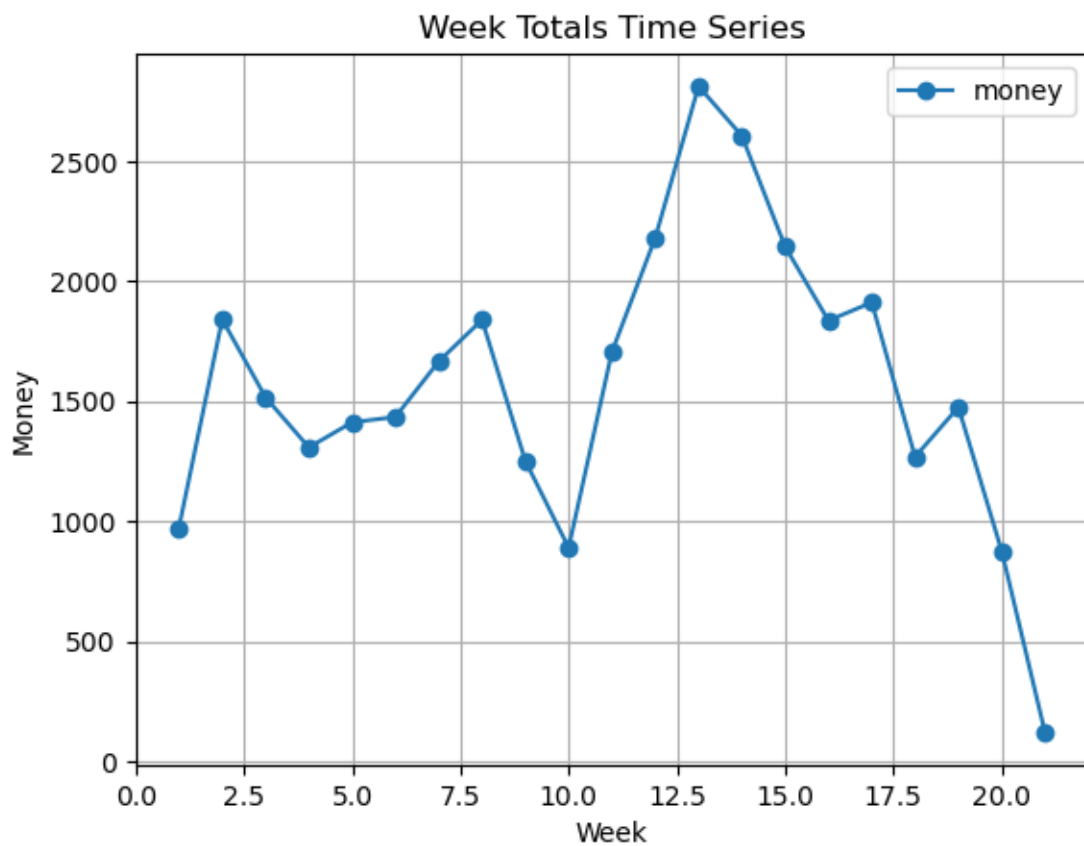
```
[196]: week1=new['money'].resample('W').sum().reset_index()
weeks=pd.DataFrame(week1)
weeks
weeks['Week']=range(1,22)
weeks
```

```
[196]:
```

	Date	money	Week
0	2024-03-03	973.50	1
1	2024-03-10	1840.50	2
2	2024-03-17	1516.30	3
3	2024-03-24	1307.80	4
4	2024-03-31	1412.10	5
5	2024-04-07	1434.50	6
6	2024-04-14	1666.00	7
7	2024-04-21	1838.84	8
8	2024-04-28	1251.20	9
9	2024-05-05	890.18	10
10	2024-05-12	1705.80	11

11	2024-05-19	2180.26	12
12	2024-05-26	2811.80	13
13	2024-06-02	2605.00	14
14	2024-06-09	2143.52	15
15	2024-06-16	1835.98	16
16	2024-06-23	1911.42	17
17	2024-06-30	1268.24	18
18	2024-07-07	1475.42	19
19	2024-07-14	876.80	20
20	2024-07-21	121.48	21

```
[191]: weeks.plot(x='Week',y='money',kind='line',marker='o')
plt.title("Week Totals Time Series")
plt.xlabel('Week')
plt.ylabel('Money')
plt.grid(True)
plt.show()
```



```
[199]: plt.figure(figsize=(8, 6))
plt.bar(weeks['Week'], weeks['money'])
```

```
plt.xlabel('Week')
plt.ylabel('Total Money')
plt.title('Total Sales per Week')

plt.show()
```

