Coffee Sales Visualization Project

About Dataset

Overview

This dataset contains detailed records of coffee sales from a vending machine.

The vending machine is the work of a dataset author who is committed to providing an open dataset to the community.

It is intended for analysis of purchasing patterns, sales trends, and customer preferences related to coffee products.

Data Collection Period

The dataset spans from March 2024 to Present time, capturing daily transaction data. And new information continues to be added.

Tasks

Time Series Exploratory Data Analysis

Next day/week/month sales

Specific customer purchases

```
# import specific modules which are required
import numpy as np,pandas as pd,seaborn as sns,matplotlib.pyplot as plt, warnings,os,zipfile
warnings.filterwarnings('ignore')
```

```
def unzip_file(zip_file_path, extract_to_folder):
    # Ensure the extraction folder exists
    os.makedirs(extract_to_folder, exist_ok=True)

    # Open the zip file and extract all contents
    with zipfile.ZipFile(zip_file_path, 'r') as zip_ref:
        zip_ref.extractall(extract_to_folder)
        print(f"Extracted all files to {extract_to_folder}")

def load_data():
    # Define file paths
    csv_file_path = 'content/index.csv'
    zip_file_path = 'coffee-sales.zip'

    # Check if the CSV file exists
    if os.path.exists(csv_file_path):
        return pd.read_csv(csv_file_path)

    # Otherwise, handle zip file download and extraction
    if not os.path.exists(zip_file_path):
        !kaggle datasets download -d ihelon/coffee-sales
    unzip_file(zip_file_path, 'content/')

    return pd.read_csv(csv_file_path)

# Load data
coffee_df = load_data()
print(coffee_df)
```

```
            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
...          ...                      ...       ...                  ...
1459  2024-09-05  2024-09-05 20:30:14.964      card  ANON-0000-0000-0587
1460  2024-09-05  2024-09-05 20:54:24.429      card  ANON-0000-0000-0588
1461  2024-09-05  2024-09-05 20:55:31.429      card  ANON-0000-0000-0588
1462  2024-09-05  2024-09-05 21:26:28.836      card  ANON-0000-0000-0040
1463  2024-09-05  2024-09-05 21:27:29.969      card  ANON-0000-0000-0040

      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
...     ...                  ...
1459  32.82           Cappuccino
1460  23.02            Americano
1461  32.82           Cappuccino
1462  27.92  Americano with Milk
1463  27.92  Americano with Milk

[1464 rows x 6 columns]
```

```
coffee_df.head()
```

	date	datetime	cash_type	card	money	coffee_name
0	2024-03-01	2024-03-01 10:15:50.520	card	ANON-0000-0000-0001	38.7	Latte
1	2024-03-01	2024-03-01 12:19:22.539	card	ANON-0000-0000-0002	38.7	Hot Chocolate
2	2024-03-01	2024-03-01 12:20:18.089	card	ANON-0000-0000-0002	38.7	Hot Chocolate
3	2024-03-01	2024-03-01 13:46:33.006	card	ANON-0000-0000-0003	28.9	Americano
4	2024-03-01	2024-03-01 13:48:14.626	card	ANON-0000-0000-0004	38.7	Latte

Next steps: Generate code with coffee_df | View recommended plots | New interactive sheet

```
coffee_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1464 entries, 0 to 1463
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   date         1464 non-null   object 
 1   datetime     1464 non-null   object 
 2   cash_type    1464 non-null   object 
 3   card         1375 non-null   object 
 4   money        1464 non-null   float64
 5   coffee_name  1464 non-null   object 
dtypes: float64(1), object(5)
memory usage: 68.8+ KB
```

Exploratory Data Analysis

- We clear the null values
- Data Extraction
- Unnessasary Columns removal etc

```
# In the above information we can see that the card column have some missing values
# Now we have to remove those columns
coffee_df.isna().sum() #Check for null values
coffee_df['card'].dropna(inplace=True) #Dropping Null values and appling to the same dataframe
```

```
#as we already have the datetime column we are now dropping the date column
coffee_df.drop('date',axis=1,inplace=True) #Dropping date column and appling to the same dataframe
coffee_df.sample() # To make sure that the changes are applied
```

	datetime	cash_type	card	money	coffee_name
1383	2024-08-26 10:37:44.464	card	ANON-0000-0000-0547	32.82	Latte

```
coffee_df['datetime']=pd.to_datetime(coffee_df['datetime']) # Convertion of object to date type
coffee_df['day']=coffee_df['datetime'].dt.day #Get date
coffee_df['month']=coffee_df['datetime'].dt.month #Get month
coffee_df['year']=coffee_df['datetime'].dt.year #Get year
coffee_df['hour']=coffee_df['datetime'].dt.hour #Get hour
coffee_df['minute']=coffee_df['datetime'].dt.minute #Get minutes
```

```
coffee_df['second']=coffee_df['datetime'].dt.second #Get seconds
#Get the weekday
coffee_df['weekday']=coffee_df['datetime'].dt.weekday.map({0:'Monday',1:'Tuesday',2:'Wednesday',3:'Thursday
#Dropping of 'datetime' column as we extracted all the info and also 'cash_type' columns as all the values
coffee_df.drop(['datetime','cash_type'],axis=1,inplace=True)
coffee_df['card']=coffee_df['card'].str.extract(r'(\d+)$')
#Check wheather the changes are applied or not
coffee_df
```

	card	money	coffee_name	day	month	year	hour	minute	second	weekday
0	0001	38.70	Latte	1	3	2024	10	15	50	Friday
1	0002	38.70	Hot Chocolate	1	3	2024	12	19	22	Friday
2	0002	38.70	Hot Chocolate	1	3	2024	12	20	18	Friday
3	0003	28.90	Americano	1	3	2024	13	46	33	Friday
4	0004	38.70	Latte	1	3	2024	13	48	14	Friday
...	...	...	...	...	...	...	...	...	...	...
1459	0587	32.82	Cappuccino	5	9	2024	20	30	14	Thursday
1460	0588	23.02	Americano	5	9	2024	20	54	24	Thursday
1461	0588	32.82	Cappuccino	5	9	2024	20	55	31	Thursday
1462	0040	27.92	Americano with Milk	5	9	2024	21	26	28	Thursday
1463	0040	27.92	Americano with Milk	5	9	2024	21	27	29	Thursday

1464 rows × 10 columns

Next steps: Generate code with coffee_df | View recommended plots | New interactive sheet

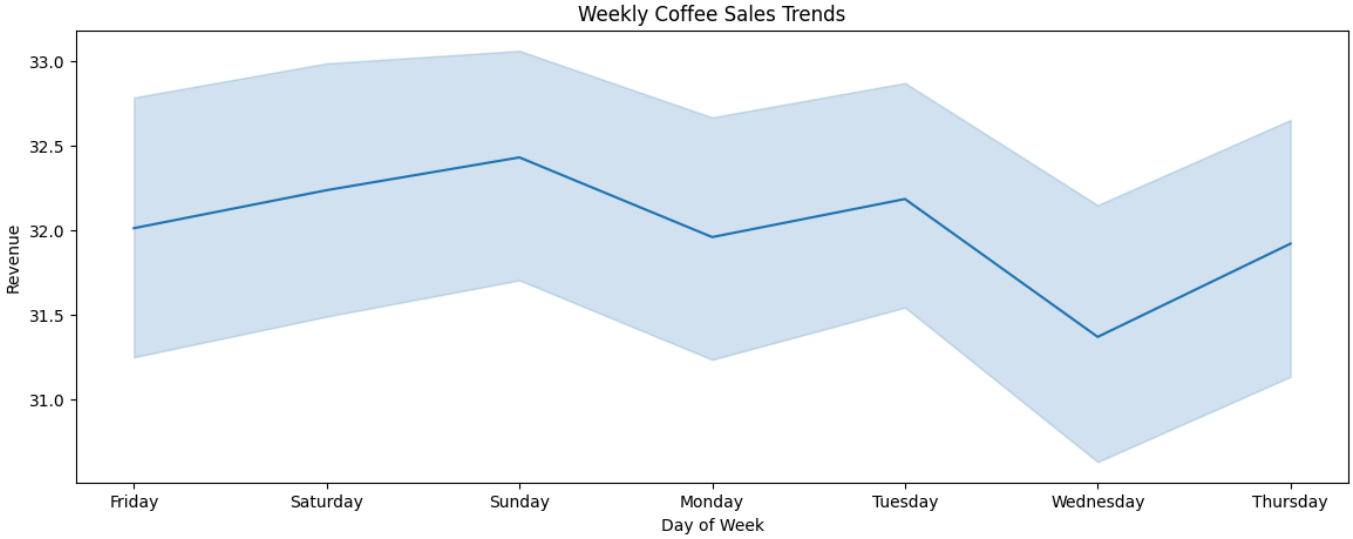
Time Series Exploratory Data Analysis

Weekly Coffee Sales Trends

- Create a graph that depicts the sales trends based on the day of the week

---

```
plt.figure(figsize=(14,5))
sns.lineplot(data=coffee_df,y='money',x='weekday')
plt.ylabel('Revenue')
plt.xlabel('Day of Week')
plt.xticks(coffee_df['weekday'].unique())
plt.title('Weekly Coffee Sales Trends')
plt.show()
```



Conclusion: from the above graph we can conclude that

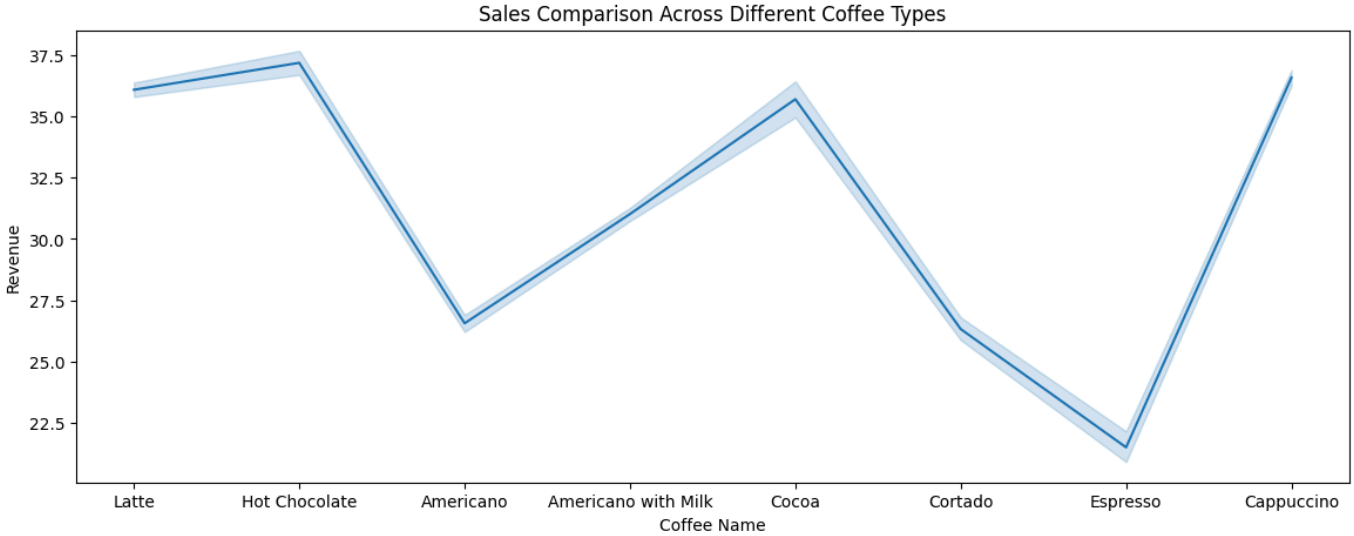
- Least revenue is generated on 'Wednesday'
- Highest revenue is generated by 'Sunday'

Sales Comparison Across Different Coffee Types

- Create a graph that depicts the sales trends based on the Different Coffee types

---

```
plt.figure(figsize=(14,5))
sns.lineplot(data=coffee_df,y='money',x='coffee_name')
plt.ylabel('Revenue')
plt.xlabel('Coffee Name')
plt.xticks(coffee_df['coffee_name'].unique())
plt.title('Sales Comparison Across Different Coffee Types')
plt.show()
```



Conclusion: from the above graph we can conclude that

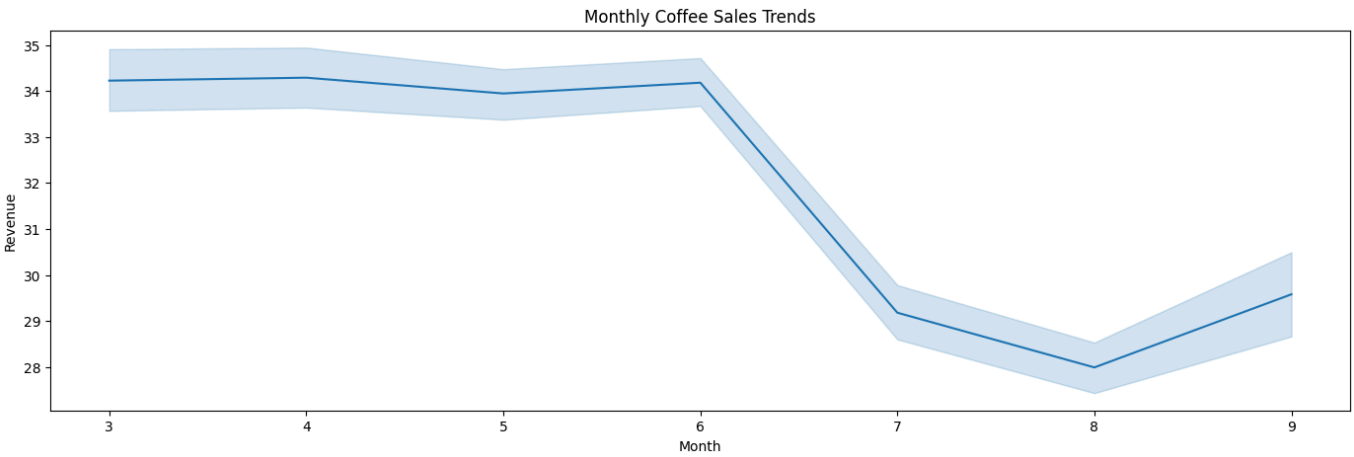
- Least revenue is generated by 'Express'
- Highest revenue is generated by 'Hot Chocolate'

Monthly Coffee Sales Trends

- Create a graph that depicts the sales trends based on the month

---

```
plt.figure(figsize=(17,5))
sns.lineplot(data=coffee_df,y='money',x='month')
plt.ylabel('Revenue')
plt.xlabel('Month')
plt.xticks(coffee_df['month'].unique())
plt.title('Monthly Coffee Sales Trends')
plt.show()
```



Conclusion: from the above graph we can conclude that

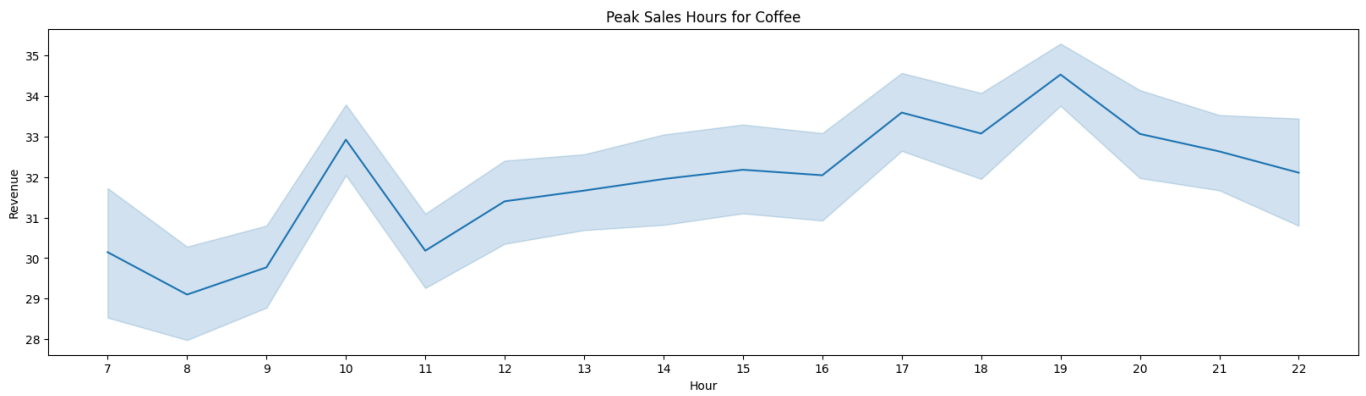
- Least revenue is generated in 'August'
- Highest revenue is generated in 'June'

Peak Sales Hours for Coffee

- Create a graph that depicts the sales trends based on the hour

---

```
plt.figure(figsize=(20,5))
sns.lineplot(data=coffee_df,y='money',x='hour')
plt.ylabel('Revenue')
plt.xlabel('Hour')
plt.title('Peak Sales Hours for Coffee')
plt.xticks(coffee_df['hour'].unique())
plt.show()
```



Conclusion: from the above graph we can conclude that

- Least revenue is generated at '19:00' or '7:00 PM'
- Highest revenue is generated at '8:00' or '8:00 AM'

Next day/week/month sales

```
#daily sales report
daily_sales=coffee_df.pivot_table(index='day',columns='coffee_name',values='money',aggfunc='sum').fillna(0)
daily_sales['total']=daily_sales.sum(axis=1)
daily_sales
```

coffee_name day	Americano	Americano with Milk	Cappuccino	Cocoa	Cortado	Espresso	Hot Chocolate	Latte	total
1	126.86	652.96	322.16	115.42	73.96	23.02	154.80	284.12	1753.30
2	245.40	504.74	185.98	32.82	51.92	0.00	0.00	367.04	1387.90
3	57.80	657.76	288.04	169.98	185.76	41.14	77.40	545.84	2023.72
4	50.94	398.24	278.24	0.00	112.66	24.00	75.44	401.52	1341.04
5	268.54	421.26	317.26	219.78	160.66	70.04	154.80	499.98	2112.32
6	198.48	288.74	337.52	32.82	107.76	0.00	32.82	424.22	1422.36
7	125.88	246.10	392.38	141.08	140.58	65.14	143.04	326.40	1580.60
8	190.54	252.98	295.88	0.00	125.88	43.12	0.00	219.46	1127.86
9	325.36	305.66	339.80	0.00	107.76	47.02	227.62	407.08	1760.30
10	76.04	187.76	109.24	98.46	74.94	73.00	115.44	443.46	1178.34
11	139.72	309.58	291.30	70.54	104.94	18.12	70.54	504.20	1508.94
12	137.64	337.50	103.36	0.00	80.82	18.12	316.94	327.06	1321.44
13	201.52	370.32	181.74	32.82	73.96	143.04	156.42	316.94	1476.76
14	370.30	256.90	332.62	0.00	272.58	66.12	115.12	468.46	1882.10
15	301.34	310.56	173.90	37.72	78.86	130.30	104.34	213.90	1350.92
16	80.82	521.42	77.70	77.40	115.60	59.26	259.14	184.98	1376.32
17	168.50	272.06	409.36	0.00	182.70	24.00	0.00	181.08	1237.70
18	264.50	370.32	436.28	0.00	131.76	18.12	110.22	246.40	1577.60
19	134.70	410.00	303.72	142.06	57.92	47.02	155.12	393.36	1643.90
20	307.70	460.68	190.56	115.42	131.76	18.12	38.70	784.10	2047.04
21	165.78	188.30	376.84	71.82	126.86	46.04	0.00	364.76	1340.40
22	210.62	348.28	300.42	0.00	78.86	24.00	37.72	212.90	1212.80
23	207.18	410.00	529.06	104.34	102.86	23.02	71.82	458.02	1906.30
24	124.90	346.32	287.06	0.00	152.82	72.48	37.72	219.78	1241.08
25	261.76	293.64	207.70	32.82	69.06	18.12	0.00	294.90	1178.00
26	312.50	293.90	357.60	76.42	239.52	41.14	189.58	648.58	2159.24
27	140.58	338.48	283.14	37.72	85.72	36.24	38.70	251.62	1212.20
28	163.60	215.04	222.70	37.72	113.64	41.14	0.00	222.40	1016.24
29	147.92	187.12	447.08	70.54	111.68	0.00	178.80	299.12	1442.26
30	73.96	687.24	218.80	0.00	69.06	105.30	75.44	473.70	1703.50
31	76.04	354.64	146.96	103.36	189.06	18.12	37.72	419.62	1345.52

Next steps: Generate code with daily_sales View recommended plots New interactive sheet

```
#daily sales report
daily_sales=coffee_df.pivot_table(index='month',columns='coffee_name',values='money',aggfunc='sum').fillna(
daily_sales['total']=daily_sales.sum(axis=1)

daily_sales
```

coffee_name	Americano	Americano with Milk	Cappuccino	Cocoa	Cortado	Espresso	Hot Chocolate	Latte	total
month									
3	1044.80	1154.00	780.50	232.20	869.20	241.00	854.00	1874.50	7050.20
4	1001.94	1407.74	1659.44	232.82	548.48	171.00	506.02	1193.12	6720.56
5	1348.80	1908.28	2078.44	340.76	474.64	185.14	529.36	2198.00	9063.42
6	390.88	2268.12	1735.12	189.88	530.48	230.20	528.08	1886.00	7758.76
7	858.12	1863.80	1079.64	300.28	322.28	273.28	361.02	1857.52	6915.94
8	851.74	2010.24	1115.88	361.02	920.80	253.68	196.92	1903.56	7613.84
9	161.14	586.32	295.38	164.10	46.04	0.00	0.00	492.30	1745.28

Next steps: Generate code with daily_sales View recommended plots New interactive sheet

```
#daily sales report
daily_sales=coffee_df.pivot_table(index='weekday',columns='coffee_name',values='money',aggfunc='sum').filln
daily_sales['total']=daily_sales.sum(axis=1)

daily_sales
```

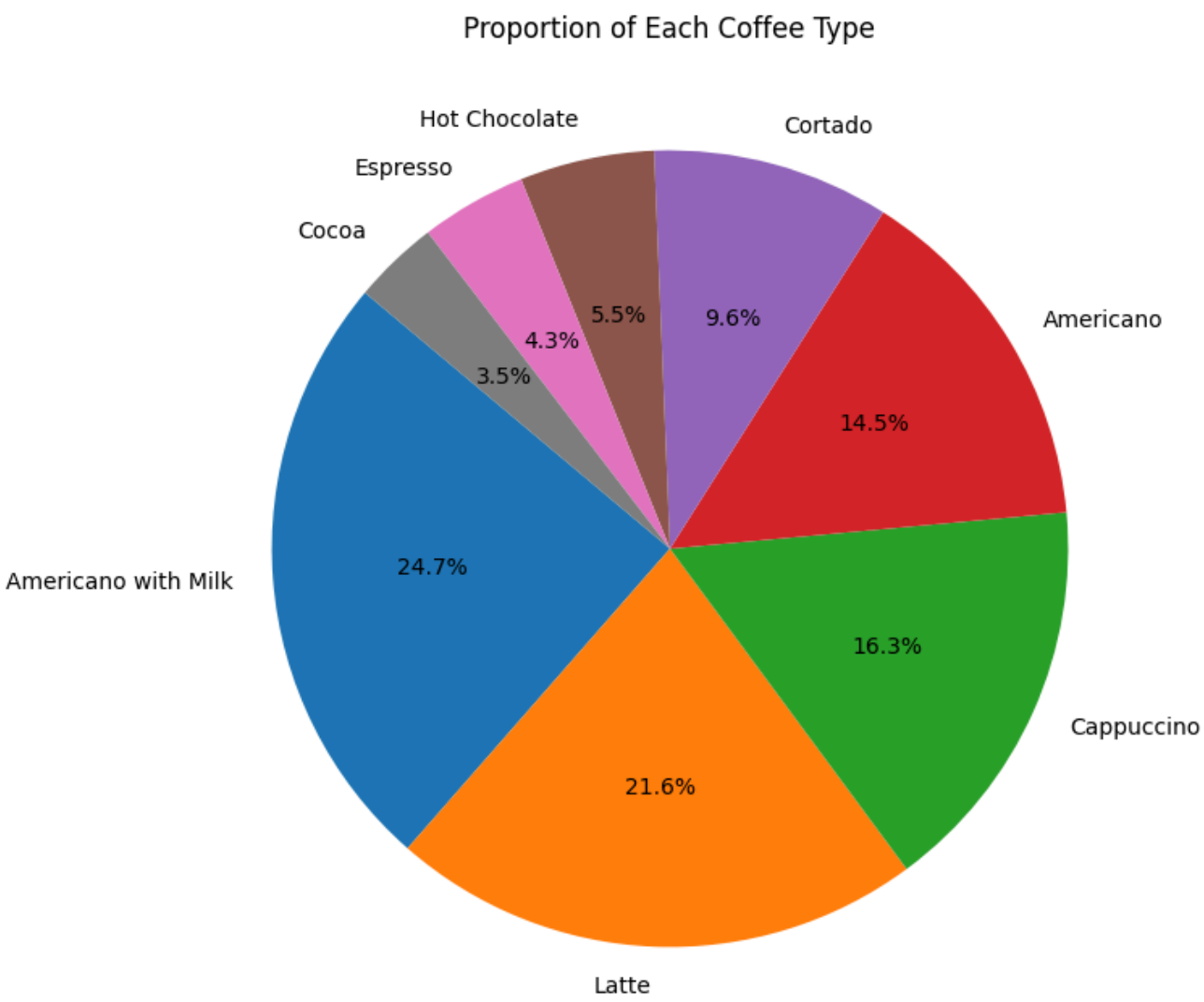
coffee_name	Americano	Americano with Milk	Cappuccino	Cocoa	Cortado	Espresso	Hot Chocolate	Latte	total
weekday									
Friday	906.28	1322.00	864.60	296.20	528.02	217.46	542.28	1629.36	6306.20
Monday	1175.10	1465.96	1329.46	239.54	380.22	115.14	212.60	1473.64	6391.66
Saturday	517.94	1982.62	1293.38	328.32	527.52	155.74	273.82	1400.16	6479.50
Sunday	588.02	1437.08	1460.58	137.16	512.10	200.32	521.22	1564.76	6421.24
Thursday	861.70	1538.10	1406.04	32.82	522.14	253.70	514.66	1956.94	7086.10
Tuesday	775.60	2073.62	1174.66	536.70	634.92	77.38	604.82	1749.92	7627.62
Wednesday	832.78	1379.12	1215.68	250.32	607.00	334.56	306.00	1630.22	6555.68

Next steps: Generate code with daily_sales View recommended plots New interactive sheet

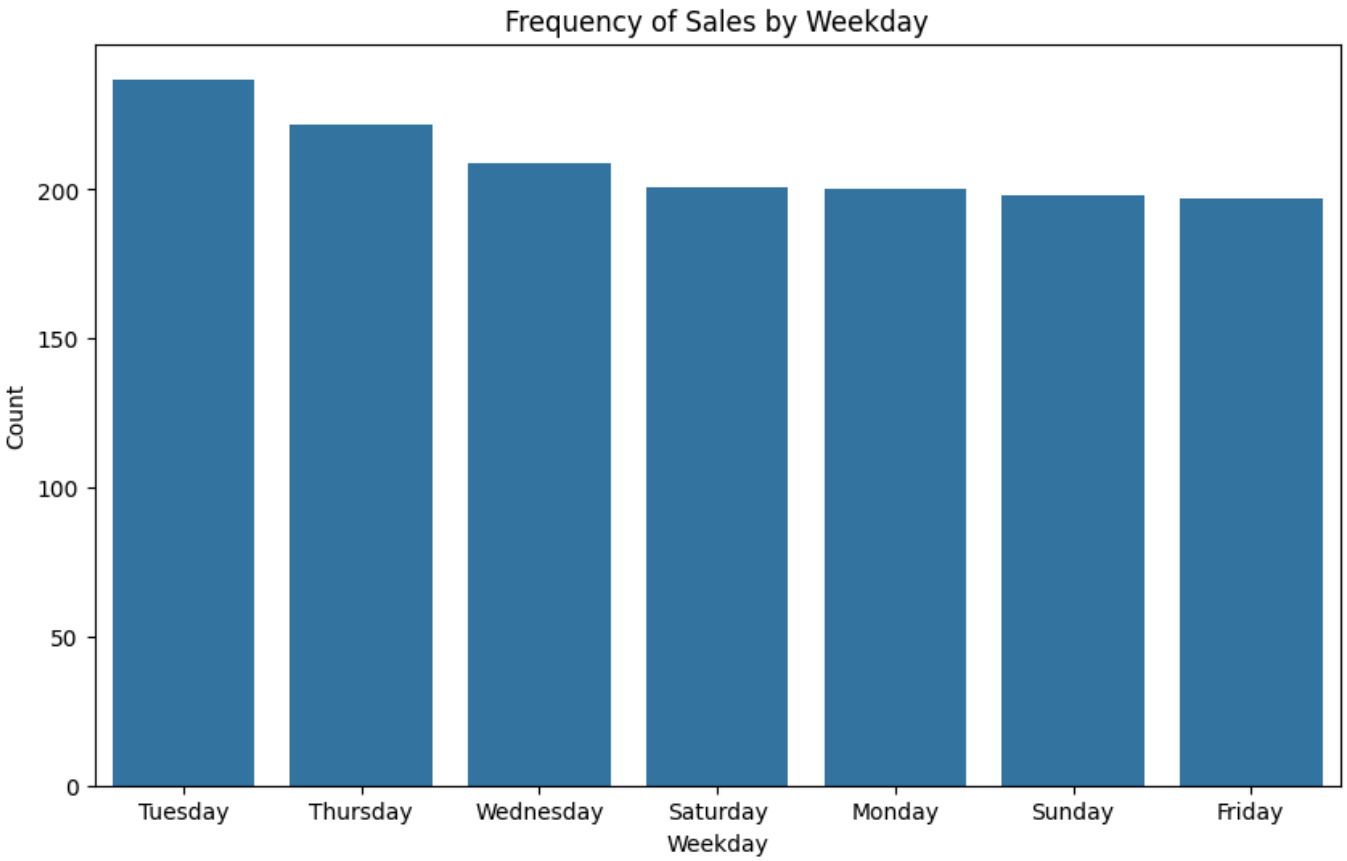
Specific customer purchases

```
# Pie Chart for coffee_name
```

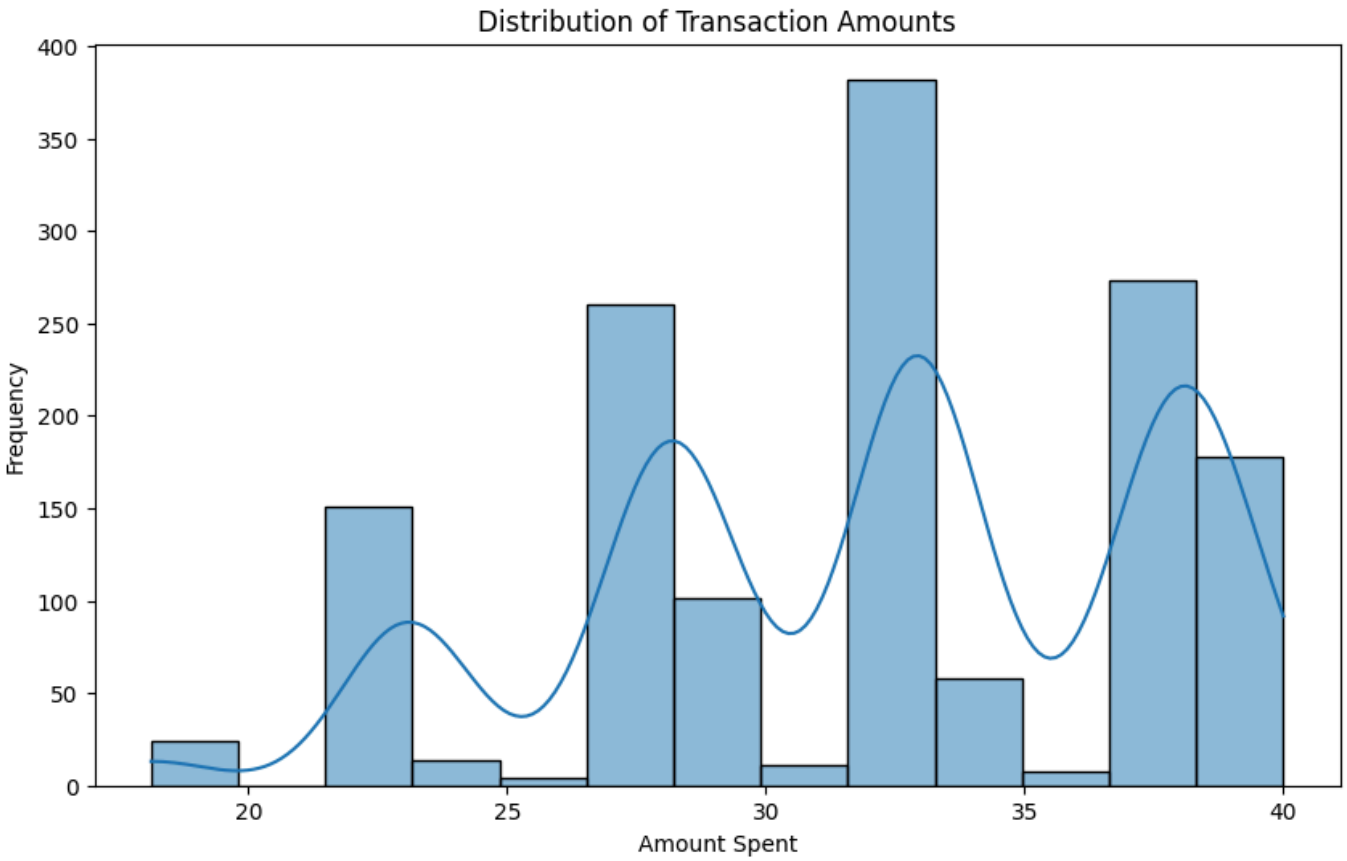
```
plt.figure(figsize=(8, 8))
coffee_counts = coffee_df['coffee_name'].value_counts()
plt.pie(coffee_counts, labels=coffee_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Proportion of Each Coffee Type')
plt.show()
```



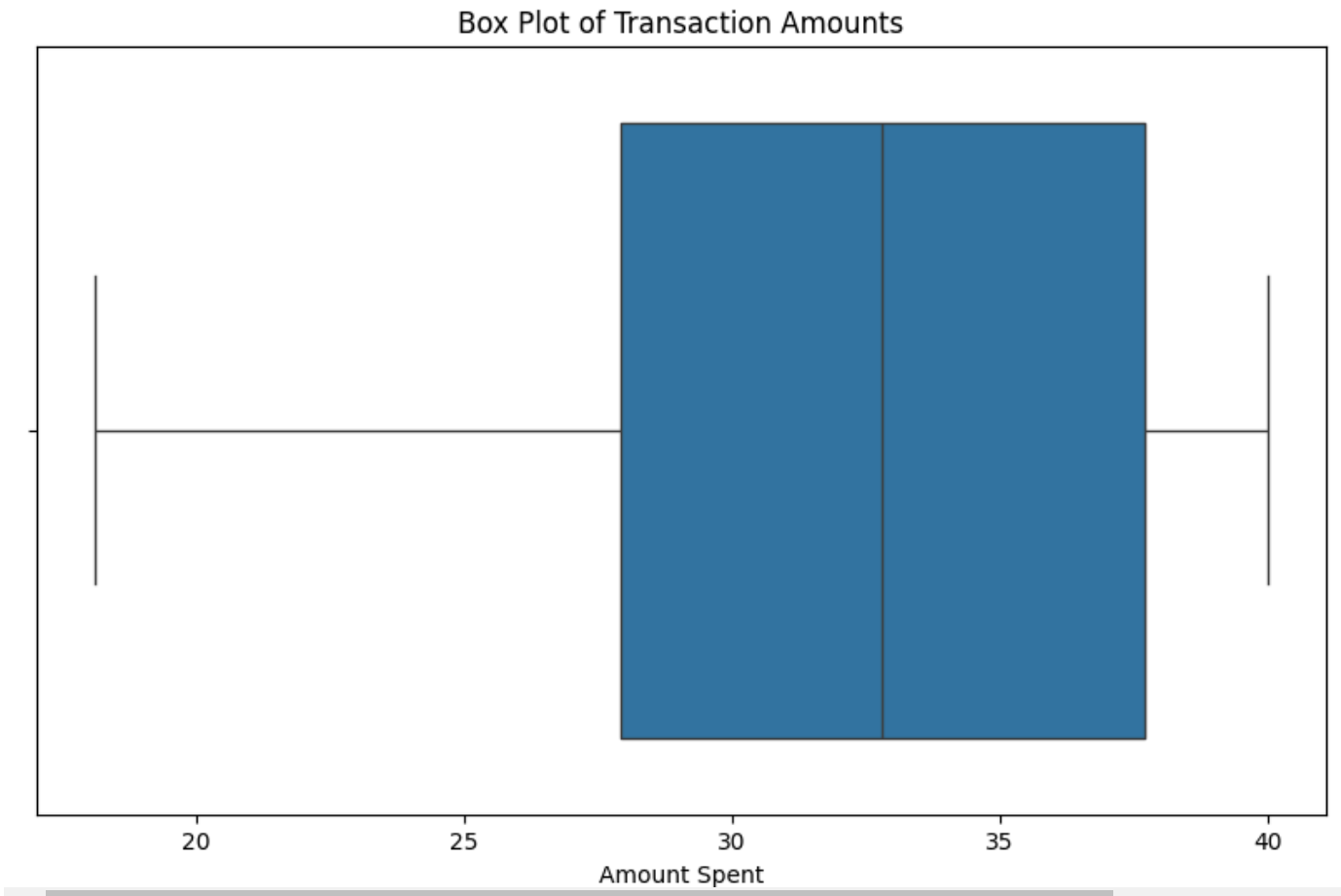

```
# Bar Chart for weekday
plt.figure(figsize=(10, 6))
sns.countplot(data=coffee_df, x='weekday', order=coffee_df['weekday'].value_counts().index)
plt.title('Frequency of Sales by Weekday')
plt.xlabel('Weekday')
plt.ylabel('Count')
plt.show()
```



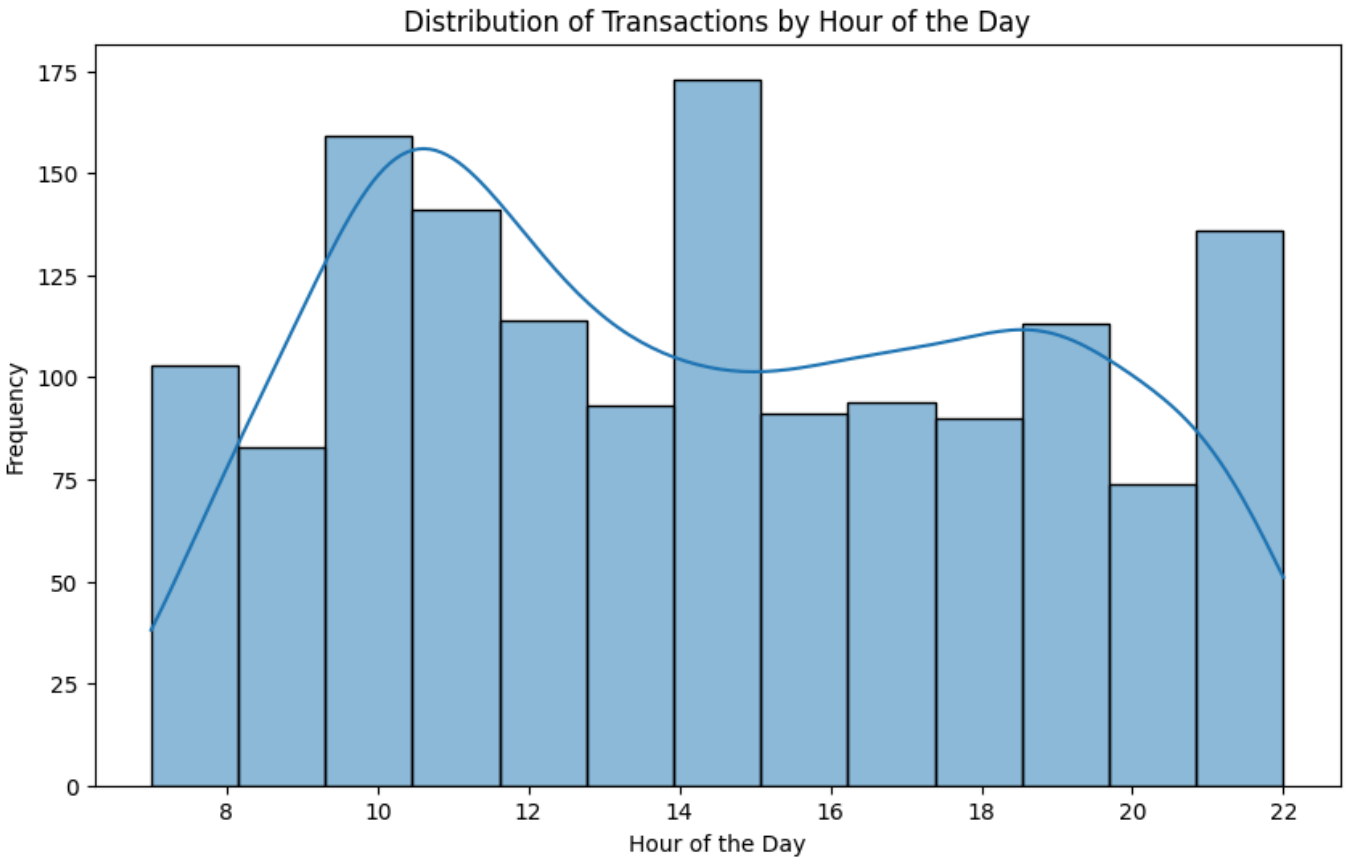

```
# Histogram for money
plt.figure(figsize=(10, 6))
sns.histplot(coffee_df['money'], kde=True)
plt.title('Distribution of Transaction Amounts')
plt.xlabel('Amount Spent')
plt.ylabel('Frequency')
plt.show()
```



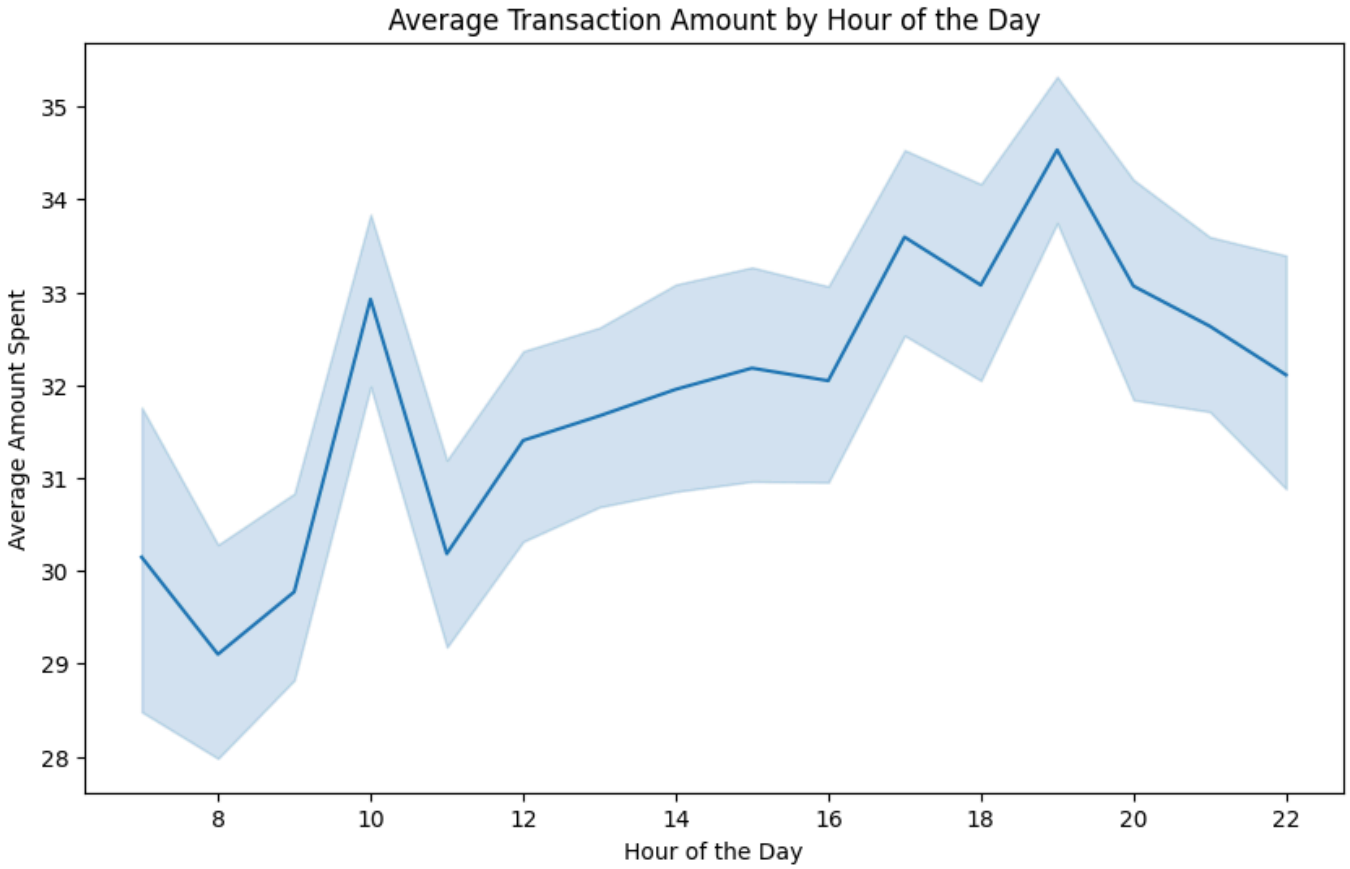

```
# Box Plot for money
plt.figure(figsize=(10, 6))
sns.boxplot(x=coffee_df['money'])
plt.title('Box Plot of Transaction Amounts')
plt.xlabel('Amount Spent')
plt.show()
```



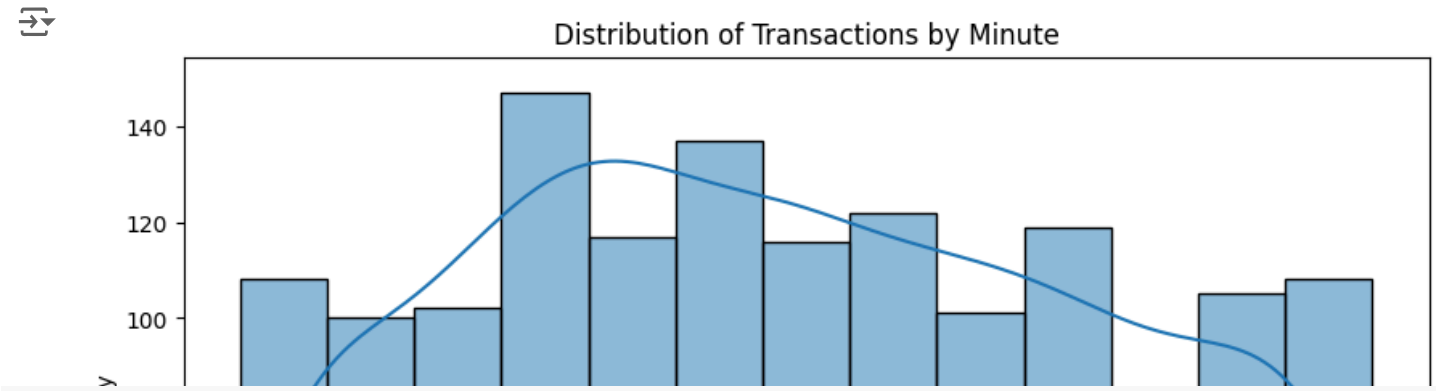

```
# Histogram for hour
plt.figure(figsize=(10, 6))
sns.histplot(coffee_df['hour'], kde=True)
plt.title('Distribution of Transactions by Hour of the Day')
plt.xlabel('Hour of the Day')
plt.ylabel('Frequency')
plt.show()
```




```
# Line Plot for hour (if there is a trend)
plt.figure(figsize=(10, 6))
sns.lineplot(data=coffee_df, x='hour', y='money', estimator='mean')
plt.title('Average Transaction Amount by Hour of the Day')
plt.xlabel('Hour of the Day')
plt.ylabel('Average Amount Spent')
plt.show()
```




```
# Histogram for minute
plt.figure(figsize=(10, 6))
sns.histplot(coffee_df['minute'],  kde=True)
plt.title('Distribution of Transactions by Minute')
plt.xlabel('Minute')
plt.ylabel('Frequency')
plt.show()
```




```
# Histogram for second
plt.figure(figsize=(10, 6))
sns.histplot(coffee_df['second'], kde=True)
plt.title('Distribution of Transactions by Second')
plt.xlabel('Second')
plt.ylabel('Frequency')
plt.show()
```