☕ Coffee Sales Data Analysis & Prediction Project Summary

Internship Project – Unified Mentor (Data Analyst Role)

Objective: To analyze and interpret coffee sales data (March–July 2024) using data analytics and predictive modeling techniques. The goal was to identify best-selling products, study payment trends, understand time-based customer behavior, generate actionable business insights, and explore a simple predictive model

Tools & Technologies Used:

- Python (Pandas, NumPy, Seaborn, Matplotlib) & Jupyter Notebook
- Data Visualization & AI-powered step-by-step guidance for EDA

Key Steps & Work Done:

1. Data Collection & Cleaning:
   - Loaded and inspected the dataset.
   - Removed duplicates and ensured consistency between payment fields.

2. Exploratory Data Analysis (EDA):
   - Identified top coffee products by sales count and revenue.
   - Studied sales trends across months, weekdays vs weekends, and hours of the day.

3. Advanced Insights:
   - Determined the most popular coffee at each hour of the day.
   - Compared weekday vs weekend revenue contributions.
   - Evaluated customer behavior patterns for strategic recommendations.

3. Predictive Modeling:
   - Built a **Linear Regression model** to predict sales amount
   - Evaluated model performance using R^2 and Mean Squared Error (MSE).

Insights:

- Americano with Milk, Latte, and Cappuccino were the top-selling coffees.
- The **Linear Regression model** provided a baseline predictive capability for sales analysis
- Sales peaked in morning hours with distinct coffee preferences by time slot.
- Weekend revenue outperformed weekdays, showing stronger customer activity.

Conclusion:

This project highlighted customer behavior and sales patterns, helping guide decisions on inventory planning, promotional offers, and peak-hour staffing. In addition, the inclusion of a **predictive modeling step** demonstrated my ability to go beyond EDA and apply **basic machine learning** techniques. This project showcased the integration of **Python-based data analysis, visualization, AI assistance, and predictive modeling** to deliver actionable business insights.

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

file = r"C:\Users\ANKITA\Downloads\Coffee sales_ (Data Analyst)
(  ML _ FA _ DA projects ).pdf"
df = pd.read_csv(file)
df.head()
```

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

     coffee_name
0          Latte
1  Hot Chocolate
2  Hot Chocolate
3      Americano
4          Latte
```

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

```
date             0
datetime         0
cash_type        0
card            89
money            0
coffee_name      0
dtype: int64
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1133 entries, 0 to 1132
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   date         1133 non-null   object
 1   datetime     1133 non-null   object
 2   cash_type    1133 non-null   object
 3   card         1044 non-null   object
 4   money        1133 non-null   float64
```

```
 5   coffee_name  1133 non-null    object
dtypes: float64(1), object(5)
memory usage: 53.2+ KB
```

```
df.columns
```

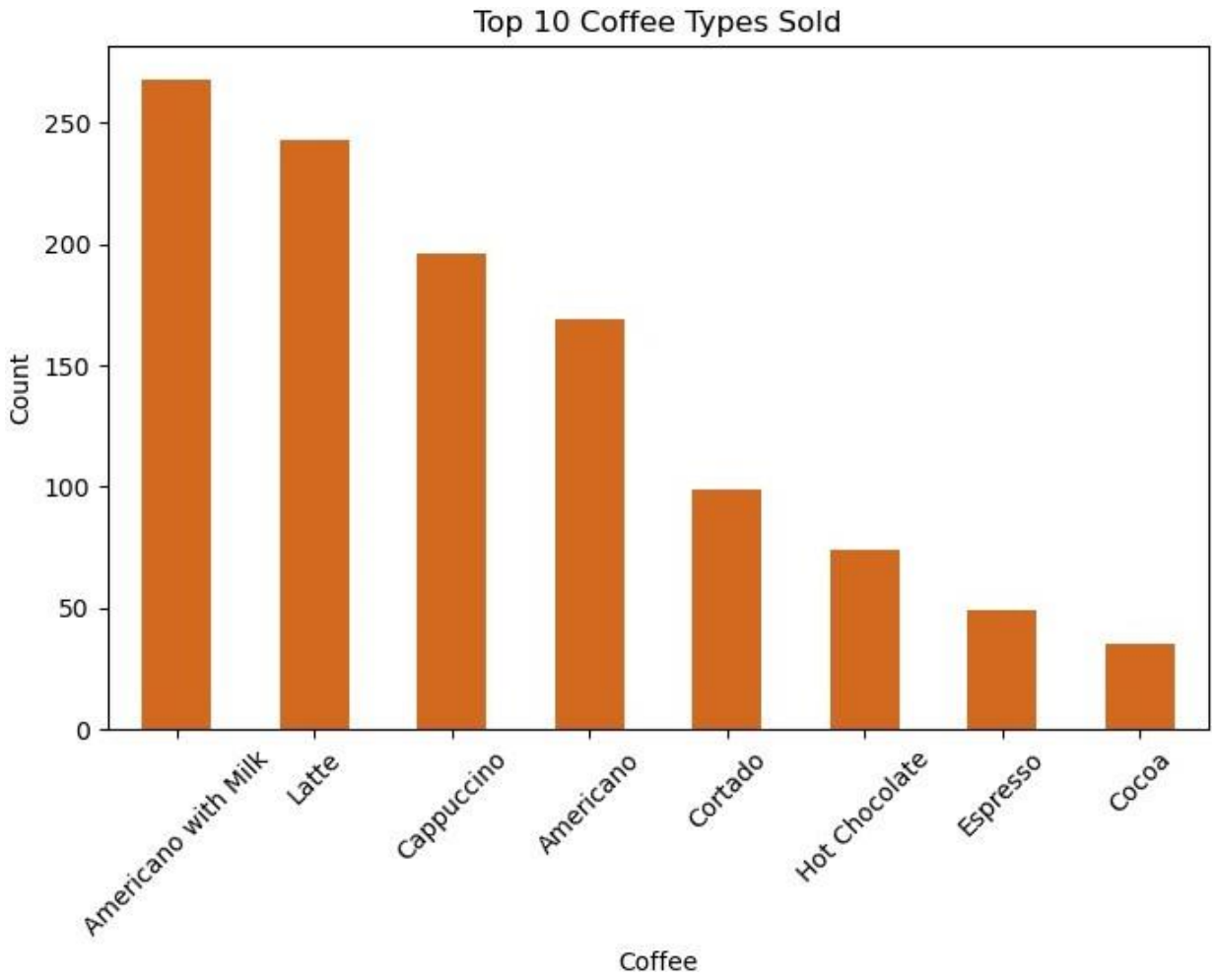
```
Index(['date', 'datetime', 'cash_type', 'card', 'money',
'coffee_name'], dtype='object')
```

```
df.describe()
```

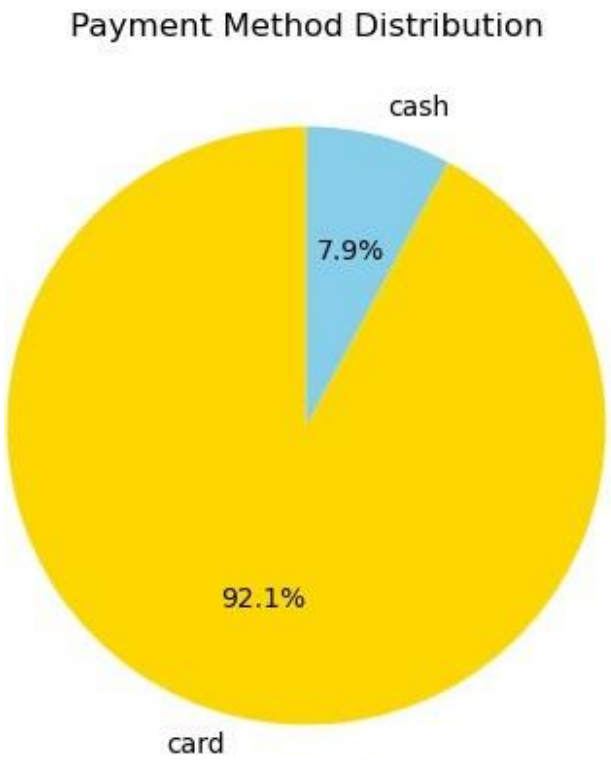
```
              money
count   1133.000000
mean      33.105808
std        5.035366
min       18.120000
25%       28.900000
50%       32.820000
75%       37.720000
max       40.000000
```

```
df['date'] = pd.to_datetime(df['date'], errors='coerce')
df['datetime'] = pd.to_datetime(df['datetime'], errors='coerce')
df['month'] = df['date'].dt.to_period('M')
df['day_name'] = df['date'].dt.day_name()
df['hour'] = df['datetime'].dt.hour
```

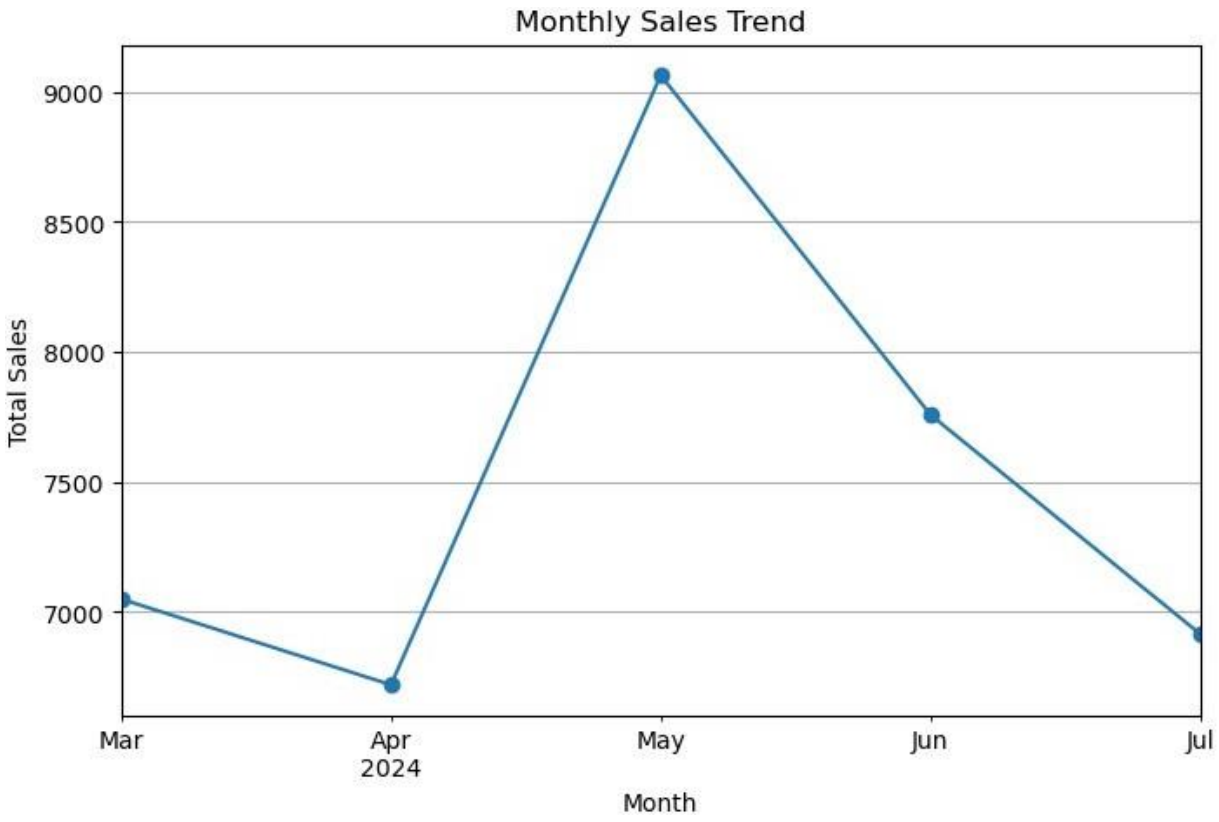
```
plt.figure(figsize=(8,5))
df['coffee_name'].value_counts().head(10).plot(kind='bar',
color='chocolate')
plt.title("Top 10 Coffee Types Sold")
plt.xlabel("Coffee")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.show()
```




```
plt.figure(figsize=(5,5))
df['cash_type'].value_counts().plot(kind='pie', autopct='%1.1f%%',
startangle=90, colors=['gold','skyblue'])
plt.title("Payment Method Distribution")
plt.ylabel("")
plt.show()
```

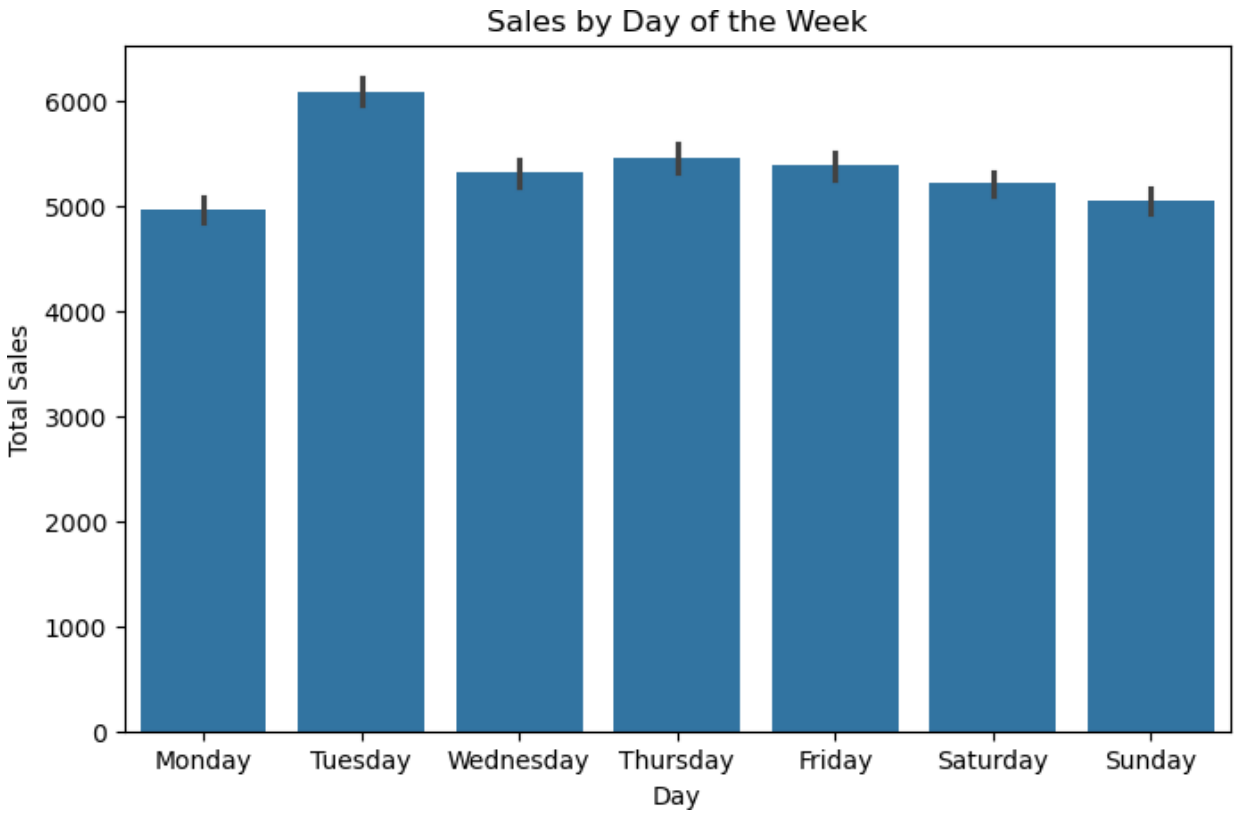



```
monthly_sales = df.groupby('month')['money'].sum()
plt.figure(figsize=(8,5))
monthly_sales.plot(marker='o')
plt.title("Monthly Sales Trend")
plt.xlabel("Month")
plt.ylabel("Total Sales")
plt.grid(True)
plt.show()
```

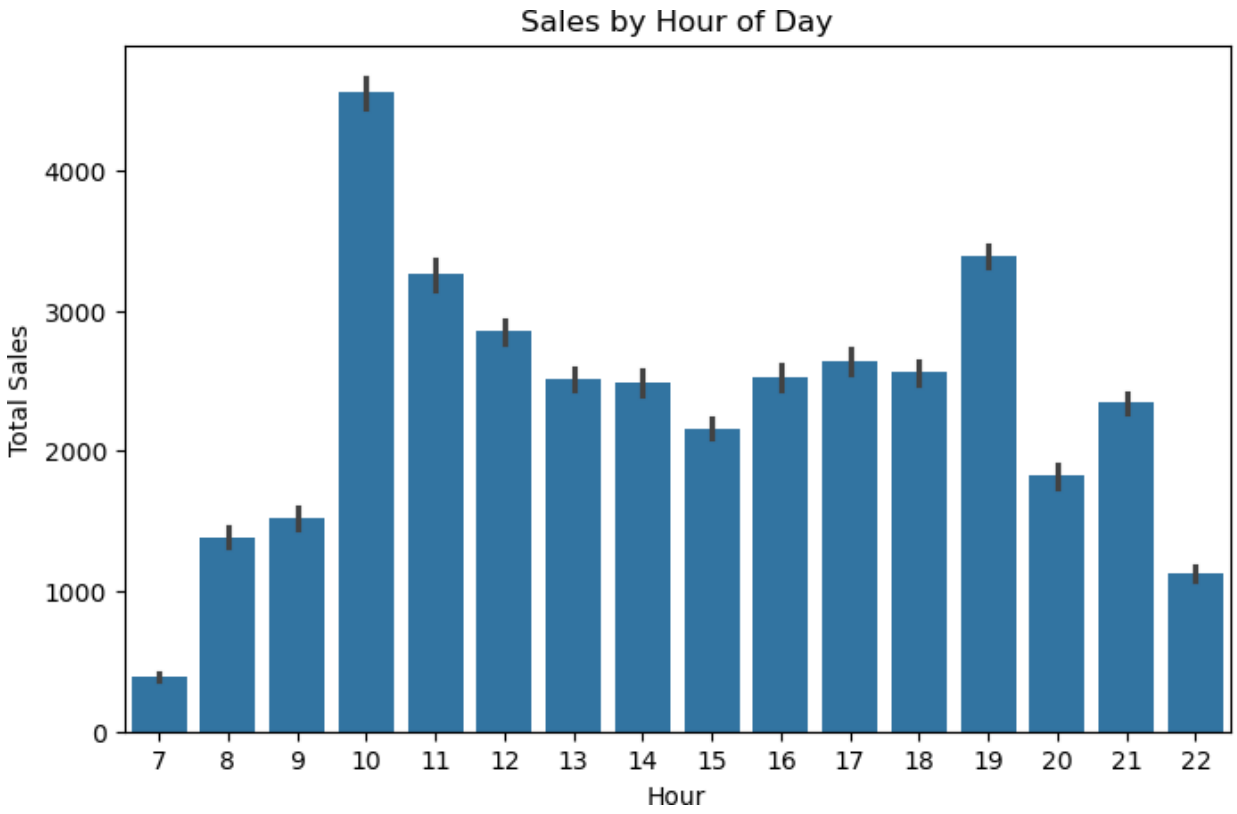



```
plt.figure(figsize=(8,5))
sns.barplot(x='day_name', y='money', data=df, estimator=sum,
order=['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','
Sunday'])

plt.title("Sales by Day of the Week")
plt.xlabel("Day")
plt.ylabel("Total Sales")
plt.show()
```

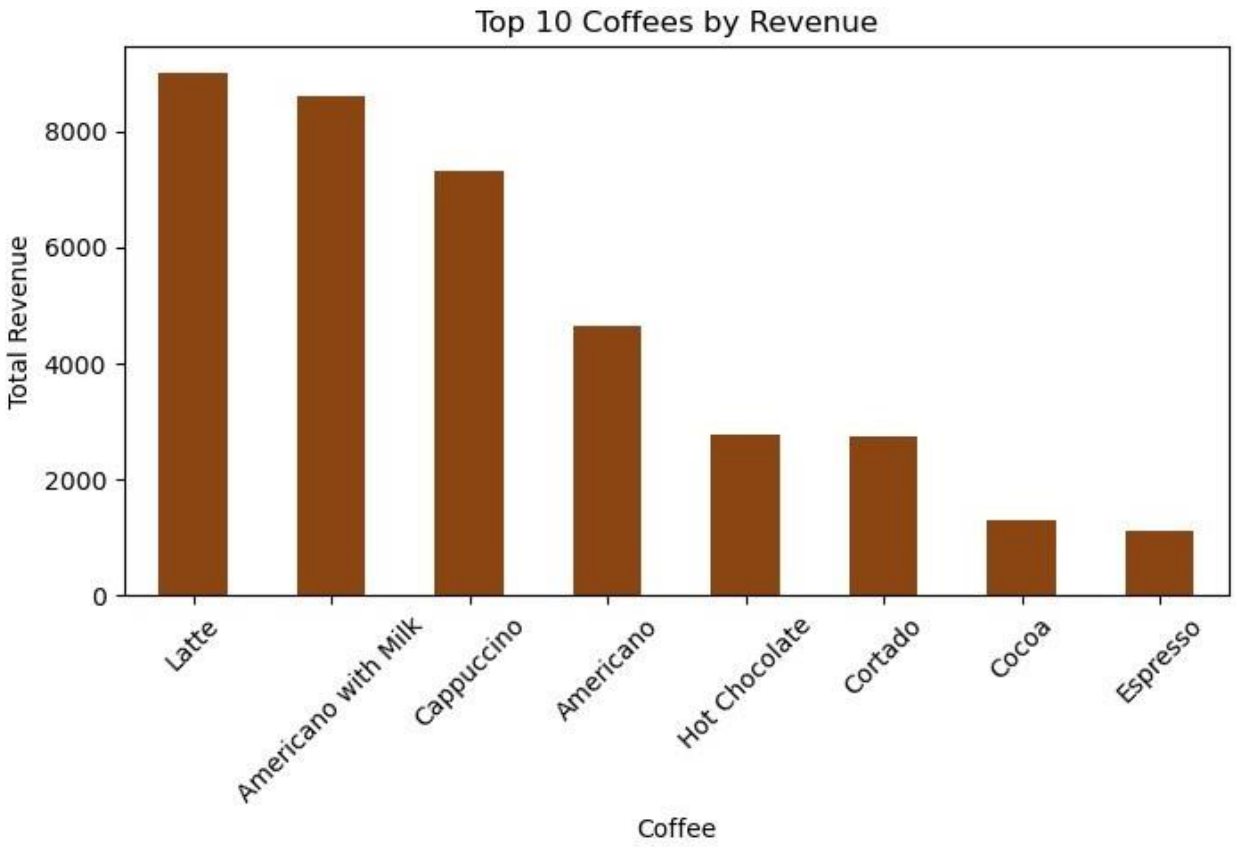



```
plt.figure(figsize=(8,5))
sns.barplot(x='hour', y='money', data=df, estimator=sum)
plt.title("Sales by Hour of Day")
plt.xlabel("Hour")
plt.ylabel("Total Sales")
plt.show()
```



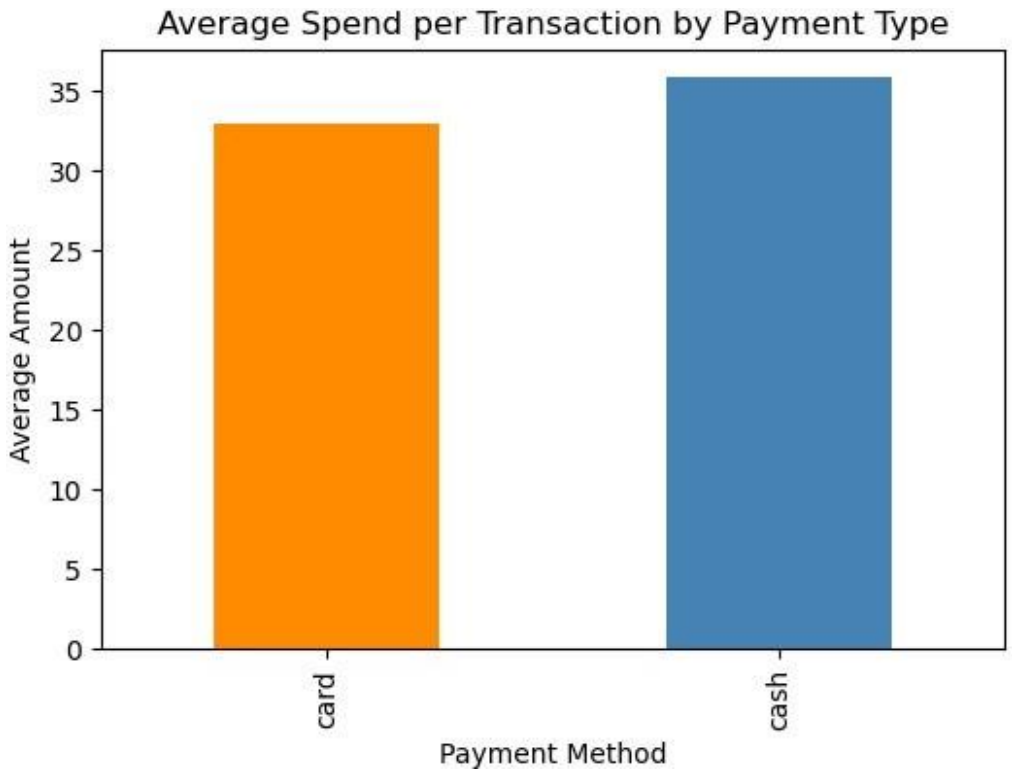

```
coffee_revenue = df.groupby('coffee_name')
['money'].sum().sort_values(ascending=False)

plt.figure(figsize=(8,4))
coffee_revenue.head(10).plot(kind='bar', color='saddlebrown')
plt.title("Top 10 Coffees by Revenue")
plt.xlabel("Coffee")
plt.ylabel("Total Revenue")
plt.xticks(rotation=45)
plt.show()
```



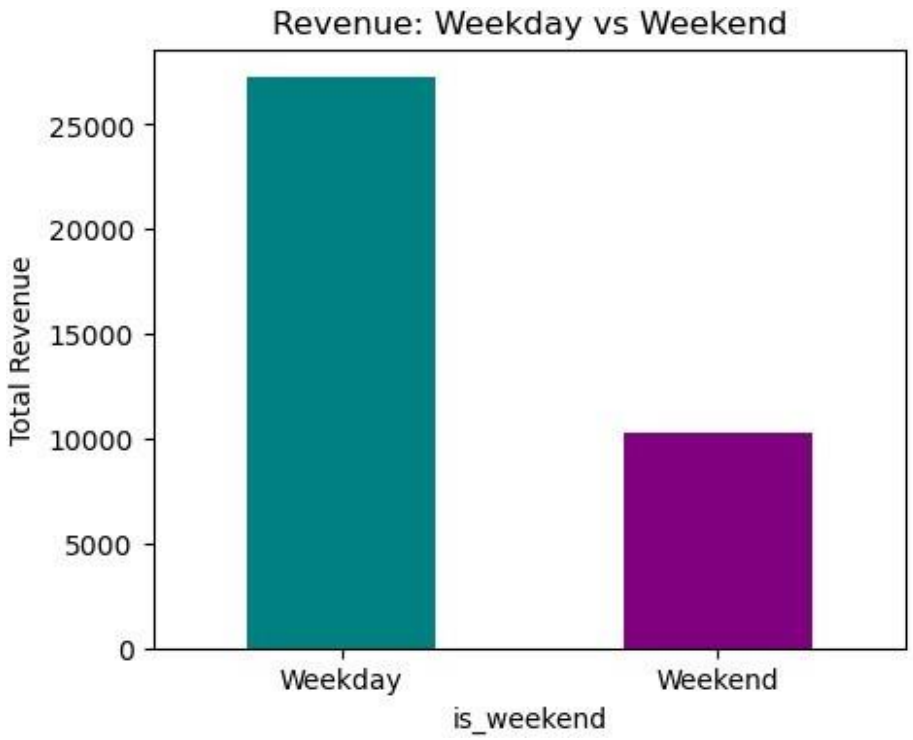

```
avg_spend = df.groupby('cash_type')['money'].mean()

plt.figure(figsize=(6,4))
avg_spend.plot(kind='bar', color=['darkorange','steelblue'])
plt.title("Average Spend per Transaction by Payment Type")
plt.ylabel("Average Amount")
plt.xlabel("Payment Method")
plt.show()
```




```
df['is_weekend'] = df['day_name'].isin(['Saturday','Sunday'])
weekend_sales = df.groupby('is_weekend')['money'].sum()

plt.figure(figsize=(5,4))
weekend_sales.plot(kind='bar', color=['teal','purple'])
plt.title("Revenue: Weekday vs Weekend")
plt.xticks([0,1], ['Weekday','Weekend'], rotation=0)
plt.ylabel("Total Revenue")
plt.show()
print(weekend_sales)
```

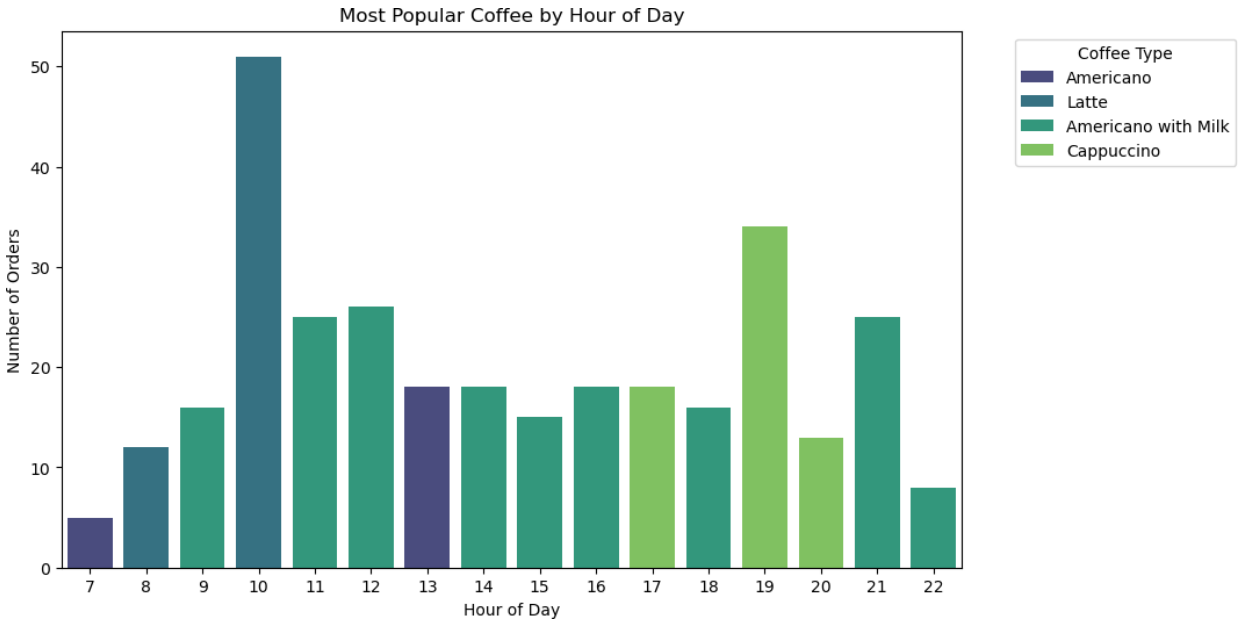



```
is_weekend
False    27242.42
True     10266.46
Name: money, dtype: float64
```

```
top_coffee_hour =
df.groupby(['hour','coffee_name']).size().reset_index(name='count')
top_by_hour = top_coffee_hour.loc[top_coffee_hour.groupby('hour')
['count'].idxmax()]

plt.figure(figsize=(10,6))
sns.barplot(x='hour', y='count', hue='coffee_name', data=top_by_hour,
dodge=False, palette='viridis')

plt.title("Most Popular Coffee by Hour of Day")
plt.xlabel("Hour of Day")
plt.ylabel("Number of Orders")
plt.legend(title="Coffee Type", bbox_to_anchor=(1.05, 1), loc='upper
left')
plt.show()
```




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

df_ml = df.copy()
le = LabelEncoder()

df_ml['coffee_name'] = le.fit_transform(df_ml['coffee_name'])
df_ml['day_name'] = le.fit_transform(df_ml['day_name'])
df_ml['cash_type'] = le.fit_transform(df_ml['cash_type'])  # cash=0,
card=1

X = df_ml[['coffee_name', 'day_name', 'hour', 'cash_type']]
y = df_ml['money']

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)

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R² Score:", r2)
```

```
Mean Squared Error: 15.464080189587678
R² Score: 0.16933348348835076
```