In [94]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [95]:
```
pd.set_option('display.max_columns', 200)
pd.set_option('display.width', 120)
%matplotlib inline
```

In [96]:
```
df=pd.read_csv("dirty_cafe_sales.csv")
df
```

Out[96]:

	Transaction ID	Item	Quantity	Price Per Unit	Total Spent	Payment Method	Location	Transaction Date
0	TXN_1961373	Coffee	2	2.0	4.0	Credit Card	Takeaway	2023-09-08
1	TXN_4977031	Cake	4	3.0	12.0	Cash	In-store	2023-05-16
2	TXN_4271903	Cookie	4	1.0	ERROR	Credit Card	In-store	2023-07-19
3	TXN_7034554	Salad	2	5.0	10.0	UNKNOWN	UNKNOWN	2023-04-27
4	TXN_3160411	Coffee	2	2.0	4.0	Digital Wallet	In-store	2023-06-11
...	...	...	...	...	...	...	...	...
9995	TXN_7672686	Coffee	2	2.0	4.0	NaN	UNKNOWN	2023-08-30
9996	TXN_9659401	NaN	3	NaN	3.0	Digital Wallet	NaN	2023-06-02
9997	TXN_5255387	Coffee	4	2.0	8.0	Digital Wallet	NaN	2023-03-02
9998	TXN_7695629	Cookie	3	NaN	3.0	Digital Wallet	NaN	2023-12-02
9999	TXN_6170729	Sandwich	3	4.0	12.0	Cash	In-store	2023-11-07

10000 rows × 8 columns

In [97]:
```
print(df.shape)
```

```
(10000, 8)
```

In [98]:
```
print(df.columns)
print(df.columns.to_list)
```

```
Index(['Transaction ID', 'Item', 'Quantity', 'Price Per Unit', 'Total Spent', 'Payme
nt Method', 'Location',
       'Transaction Date'],
      dtype='object')
<bound method IndexOpsMixin.tolist of Index(['Transaction ID', 'Item', 'Quantity',
'Price Per Unit', 'Total Spent', 'Payment Method', 'Location',
       'Transaction Date'],
      dtype='object')>
```

In [99]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Transaction ID    10000 non-null  object
 1   Item              9667 non-null   object
 2   Quantity          9862 non-null   object
 3   Price Per Unit    9821 non-null   object
 4   Total Spent       9827 non-null   object
 5   Payment Method    7421 non-null   object
 6   Location          6735 non-null   object
 7   Transaction Date  9841 non-null   object
dtypes: object(8)
memory usage: 625.1+ KB
```

In [100…
```
print(df.value_counts())
```

```
Transaction ID  Item      Quantity  Price Per Unit  Total Spent  Payment Method  Loc
ation  Transaction Date
TXN_9999124     Juice     2         3.0             6.0          Digital Wallet  Tak
eaway  UNKNOWN             1
TXN_1000555     Tea       1         1.5             1.5          Credit Card     In-
store  2023-10-19          1
TXN_1001832     Salad     2         5.0             10.0         Cash            Tak
eaway  UNKNOWN             1
TXN_1002457     Cookie    5         1.0             5.0          Digital Wallet  Tak
eaway  2023-09-29          1
TXN_1004184     Smoothie  1         4.0             4.0          Credit Card     In-
store  2023-05-18          1

..
TXN_1010950     Cookie    ERROR     1.0             1.0          Digital Wallet  Tak
eaway  2023-01-07          1
TXN_1009421     Cookie    4         1.0             4.0          Cash            Tak
eaway  2023-02-21          1
TXN_1007347     Salad     4         5.0             20.0         Digital Wallet  In-
store  2023-08-28          1
TXN_1006942     Salad     1         5.0             5.0          Credit Card     In-
store  2023-11-30          1
TXN_1005377     Cake      5         UNKNOWN         15.0         Digital Wallet  Tak
eaway  2023-06-03          1
Name: count, Length: 4550, dtype: int64
```

In [101…
```
df.describe(include='all')
```

Out[101…

	Transaction ID	Item	Quantity	Price Per Unit	Total Spent	Payment Method	Location	Transaction Date
count	10000	9667	9862	9821	9827	7421	6735	9841
unique	10000	10	7	8	19	5	4	367
top	TXN_9226047	Juice	5	3.0	6.0	Digital Wallet	Takeaway	UNKNOWN
freq	1	1171	2013	2429	979	2291	3022	159

In [102…
```
Mising_Valuee=df.isnull().sum()
print(Mising_Valuee)
```

```
Transaction ID         0
Item                 333
Quantity             138
Price Per Unit       179
Total Spent          173
Payment Method      2579
Location            3265
Transaction Date     159
dtype: int64
```

In [103…
```
Mising_Present=df.isna().mean()*100

print(Mising_Present)
```

```
Transaction ID       0.00
Item                 3.33
Quantity             1.38
Price Per Unit       1.79
Total Spent          1.73
Payment Method      25.79
Location            32.65
Transaction Date     1.59
dtype: float64
```

In [104…
```
df = df.dropna(subset=['Location', 'Payment Method'])
```

In [105…
```
df = df[df['Location'].notna()]
```

In [106…
```
print(Mising_Present)
```

```
Transaction ID       0.00
Item                 3.33
Quantity             1.38
Price Per Unit       1.79
Total Spent          1.73
Payment Method      25.79
Location            32.65
Transaction Date     1.59
dtype: float64
```

In [107…
```
df['Item'].fillna('Unknown',inplace=True)
for col in['Quantity','Price Per Unit','Total Spent']:
   df[col].fillna(df[col].median,inplace=True)
```

```
C:\Users\s\AppData\Local\Temp\ipykernel_1708\2301191473.py:1: FutureWarning: A value
is trying to be set on a copy of a DataFrame or Series through chained assignment us
ing an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because
the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method
({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform
the operation inplace on the original object.


  df['Item'].fillna('Unknown',inplace=True)
```

In [108…
```
df['Transaction Date']=df['Transaction Date'].ffill()
```

In [109…
```
df.isnull().sum()
```

Out[109…
```
Transaction ID      0
Item                0
Quantity            0
Price Per Unit      0
Total Spent         0
Payment Method      0
Location            0
Transaction Date    0
dtype: int64
```

In [110…
```
print(Mising_Present)
```

```
Transaction ID       0.00
Item                 3.33
Quantity             1.38
Price Per Unit       1.79
Total Spent          1.73
Payment Method      25.79
Location            32.65
Transaction Date     1.59
dtype: float64
```

In [111…
```
Mising_Present2=df.isna().mean()*100

print(Mising_Present2)
```

```
Transaction ID      0.0
Item                0.0
Quantity            0.0
Price Per Unit      0.0
Total Spent         0.0
Payment Method      0.0
Location            0.0
Transaction Date    0.0
dtype: float64
```

In [112…
```
print(df.columns)
```

```
Index(['Transaction ID', 'Item', 'Quantity', 'Price Per Unit', 'Total Spent', 'Payme
nt Method', 'Location',
       'Transaction Date'],
      dtype='object')
```

In [113…
```
df['Transaction Date']=pd.to_datetime(df['Transaction Date'],errors='coerce')

df['year'] = df['Transaction Date'].dt.year
df['month'] = df['Transaction Date'].dt.month
df['hour'] = df['Transaction Date'].dt.hour
```

In [114…
```
df['Quantity'] = pd.to_numeric(df['Quantity'], errors='coerce')
df['Price Per Unit']=pd.to_numeric(df['Price Per Unit'],errors='coerce')
df['Total Spent']=pd.to_numeric(df['Total Spent'],errors='coerce')
```

In [115…
```
df.dtypes
```

Out[115…
```
Transaction ID              object
Item                        object
Quantity                   float64
Price Per Unit             float64
Total Spent                float64
Payment Method              object
Location                    object
Transaction Date    datetime64[ns]
year                       float64
month                      float64
hour                       float64
dtype: object
```

In [116…
```
df['Payment Method']=df['Payment Method'].replace(['ERROR', 'UNKNOWN'], np.nan)
```

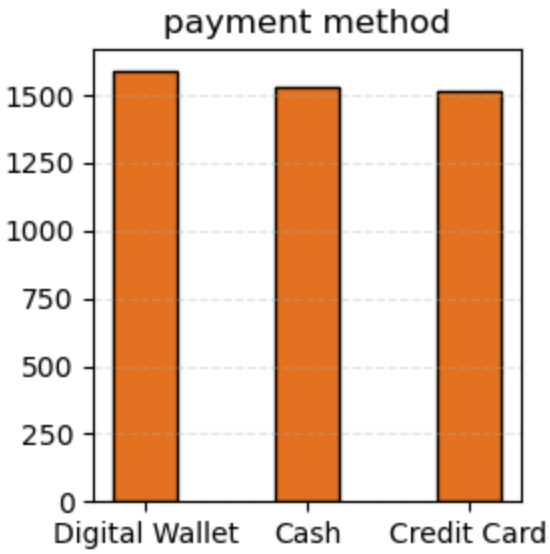
In [117…
```
df['Item']=df['Item'].replace(['ERROR', 'UNKNOWN','Unknown'], np.nan)
```

In [118…
```
# clean for location
df['Location'] = df['Location'].replace(['ERROR', 'UNKNOWN'], np.nan)
```

In [119…
```
paymentMethod=df['Payment Method'].value_counts().sort_values(ascending=False)
print(paymentMethod)
plt.figure(figsize=(3,3))


plt.bar(paymentMethod.index,
```

```
        paymentMethod.values,
        width=0.4,
        color="#E37120",
        edgecolor='black',
        linewidth=1)

plt.title('payment method')
plt.grid(axis='y', linestyle='--', alpha=0.35)
plt.tight_layout()
plt.show()
```

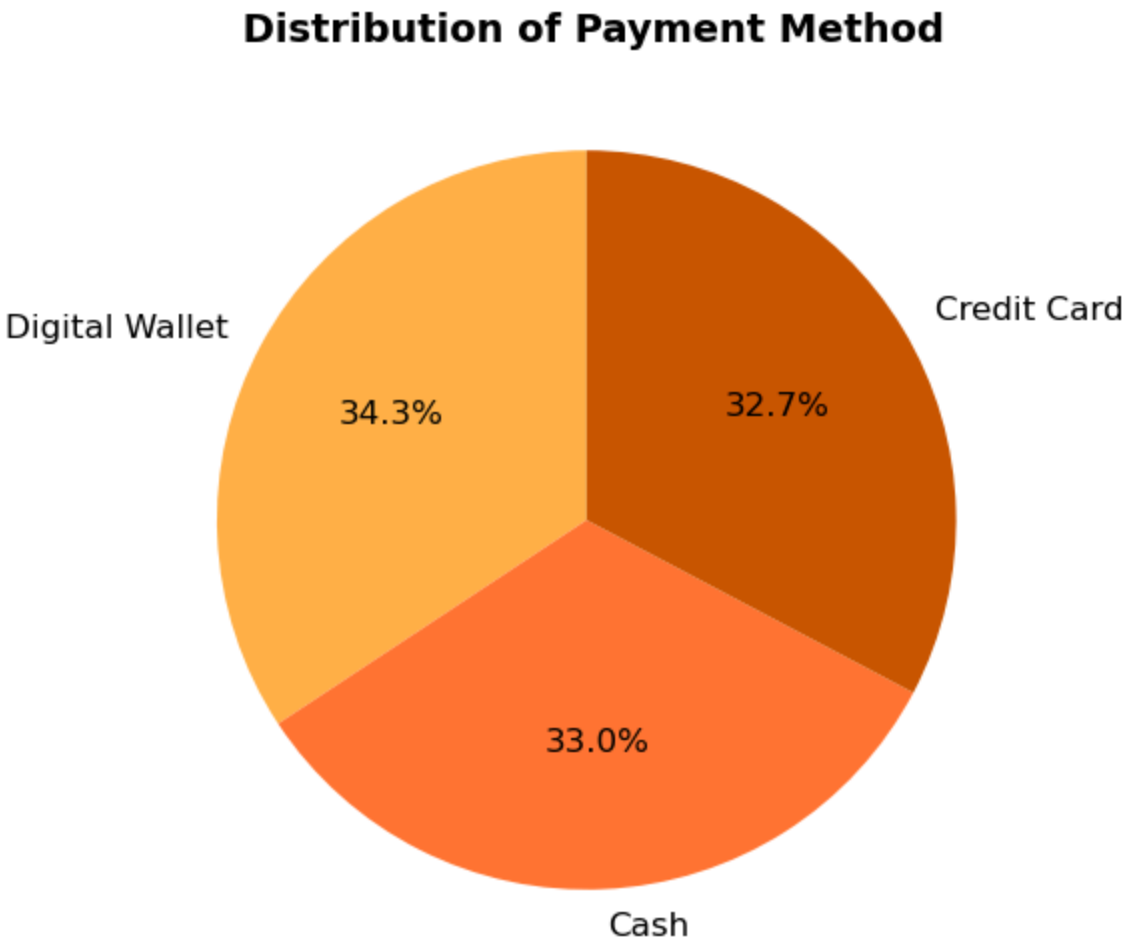
```
Payment Method
Digital Wallet    1588
Cash              1527
Credit Card       1516
Name: count, dtype: int64
```



In [120…

```
import matplotlib.pyplot as plt


paymentMethod = df['Payment Method'].value_counts().sort_values(ascending=False)
print(paymentMethod)


autumn_colors = ['#FFB347', '#FF7733', '#CC5500', '#8B4513', '#FFD580']


plt.figure(figsize=(6,6))
plt.pie(
    paymentMethod,
    labels=paymentMethod.index,
    colors=autumn_colors[:len(paymentMethod)],
    autopct='%1.1f%%',
    startangle=90,
    textprops={'fontsize': 12, 'color': 'black'}
)
plt.title(' Distribution of Payment Method', fontsize=14, fontweight='bold')
plt.show()
```

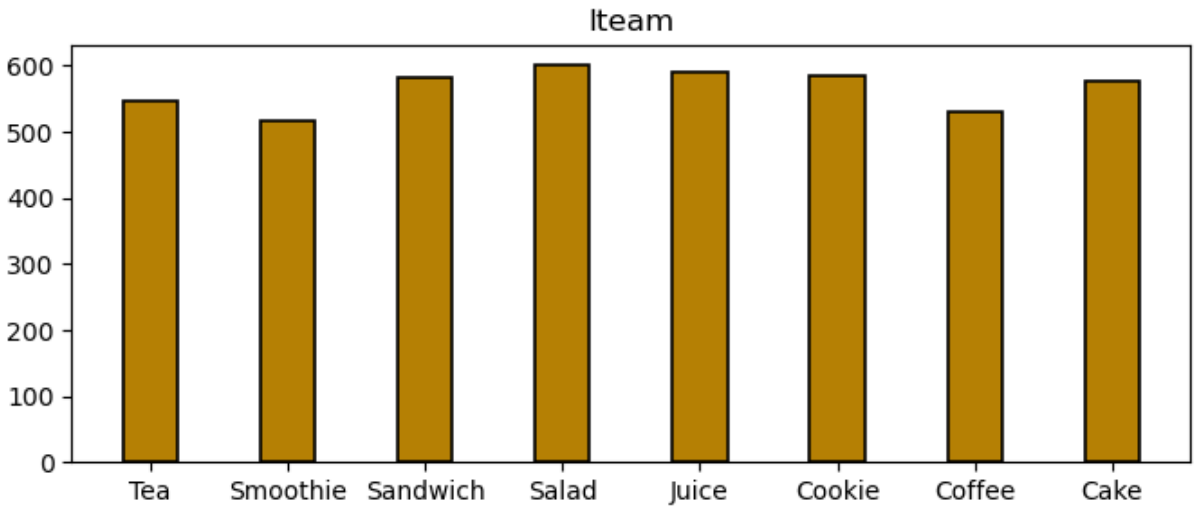
```
Payment Method
Digital Wallet    1588
Cash              1527
Credit Card       1516
Name: count, dtype: int64
```



In [121…

```
Iteam=df['Item'].value_counts().sort_index(ascending=False)
print(Iteam)
plt.figure(figsize=(8,3))
plt.bar(Iteam.index,Iteam.values,width=0.4,color="#B58004",
        edgecolor='black',
        linewidth=1.2)
plt.title('Iteam')

plt.show()
```

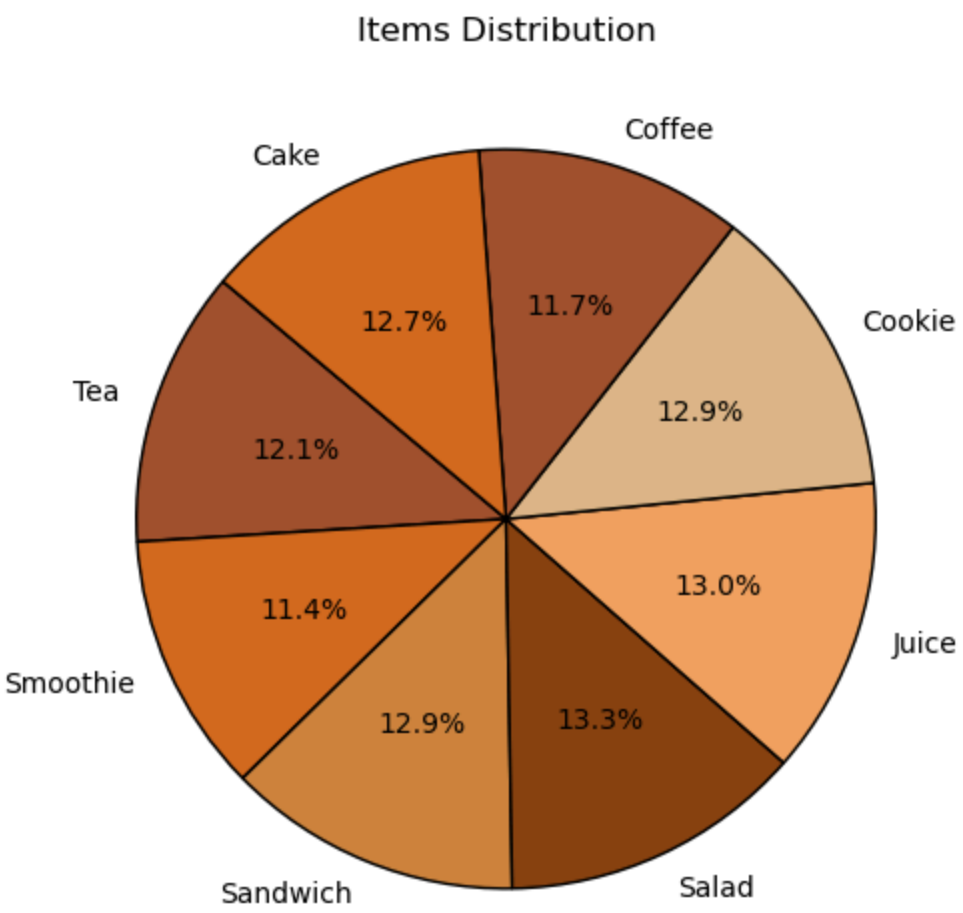
```
Item
Tea         547
Smoothie    515
Sandwich    583
Salad       600
Juice       589
Cookie      584
Coffee      529
Cake        575
Name: count, dtype: int64
```



In [122…

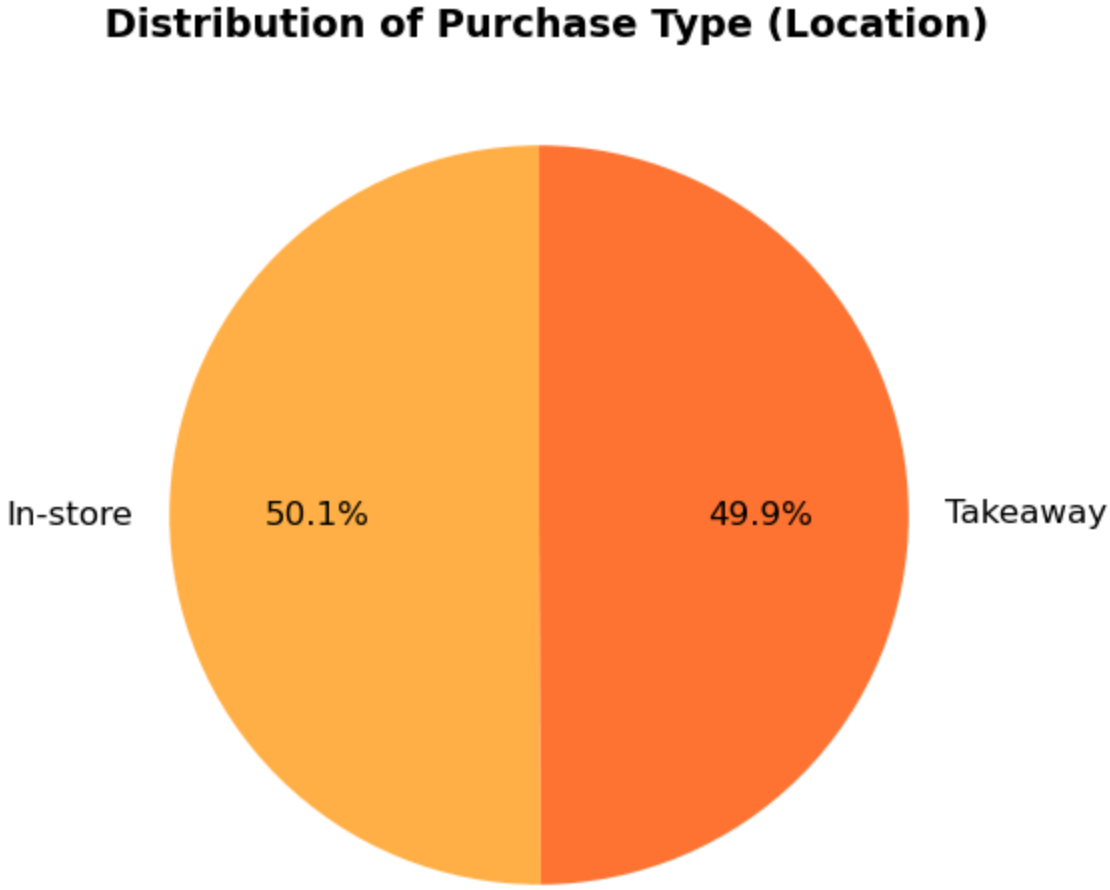
```python
import matplotlib.pyplot as plt

# Count and sort items
Iteam = df['Item'].value_counts().sort_index(ascending=False)
print(Iteam)

# Define autumn coffee shop colors
autumn_colors = ['#A0522D', '#D2691E', '#CD853F', '#8B4513', '#F4A460', '#DEB887']

# Create pie chart
plt.figure(figsize=(6,6))
plt.pie(Iteam.values, labels=Iteam.index, autopct='%1.1f%%', startangle=140, colors
plt.title('Items Distribution')
plt.show()
```

```
Item
Tea         547
Smoothie    515
Sandwich    583
Salad       600
Juice       589
Cookie      584
Coffee      529
Cake        575
Name: count, dtype: int64
```



In [123…

```
import matplotlib.pyplot as plt


purchasetype = df['Location'].value_counts(ascending=False)

autumn_colors = ['#FFB347', '#FF7733', '#CC5500', '#8B4513', '#FFD580']


plt.figure(figsize=(6,6))
plt.pie(
    purchasetype,
    labels=purchasetype.index,
    colors=autumn_colors[:len(purchasetype)],
    autopct='%1.1f%%',
    startangle=90,
    textprops={'fontsize': 12, 'color': 'black'}
)
plt.title(' Distribution of Purchase Type (Location)', fontsize=14, fontweight='bol
plt.show()
```



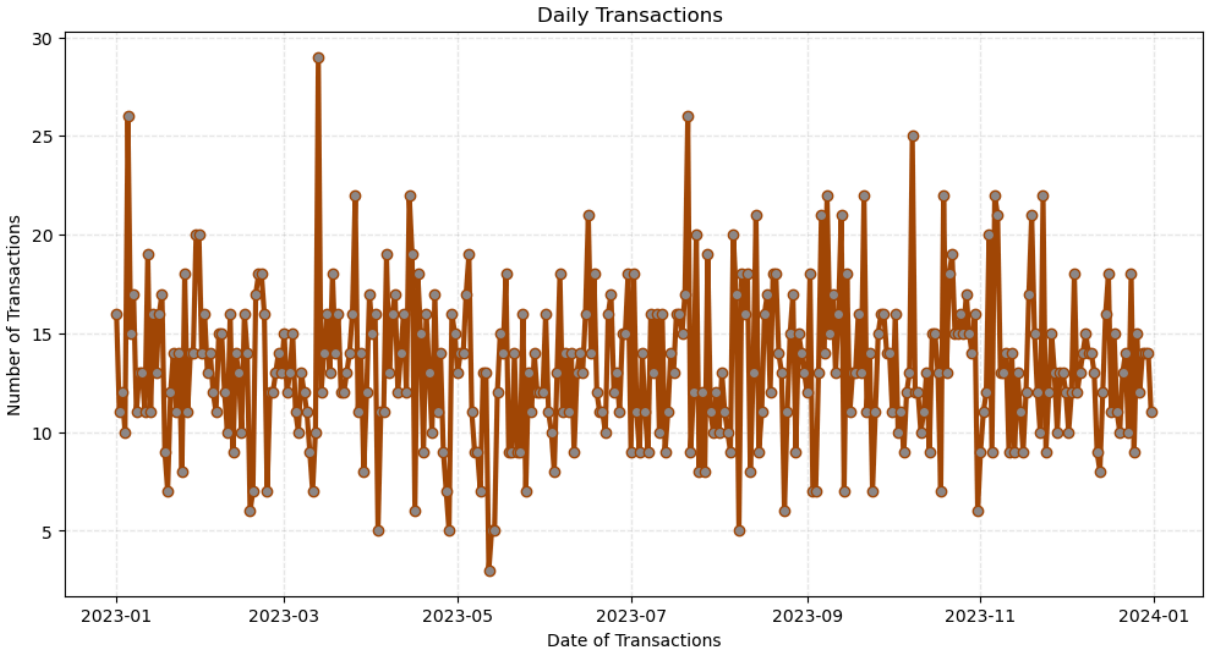
In [124…
```
print("min sales data",df['Transaction Date'].min())
print('max sales data',df['Transaction Date'].max())
```

```
min sales data 2023-01-01 00:00:00
max sales data 2023-12-31 00:00:00
```

In [125…
```
transactions_per_day = df.groupby('Transaction Date').size()
print(transactions_per_day.head())
```

```
Transaction Date
2023-01-01    16
2023-01-02    11
2023-01-03    12
2023-01-04    10
2023-01-05    26
dtype: int64
```

In [126…
```
plt.figure(figsize=(12,6))
plt.plot(transactions_per_day, color="#A44908", linewidth=3, marker='o', markerface
plt.xlabel("Date of Transactions")
plt.ylabel("Number of Transactions")
plt.title("Daily Transactions ")
plt.grid(True, linestyle='--', alpha=0.3)
plt.show()
```



In [127…
```
Sales_Year_Top=df.groupby('year')['Total Spent'].mean()
print(Sales_Year_Top)
```

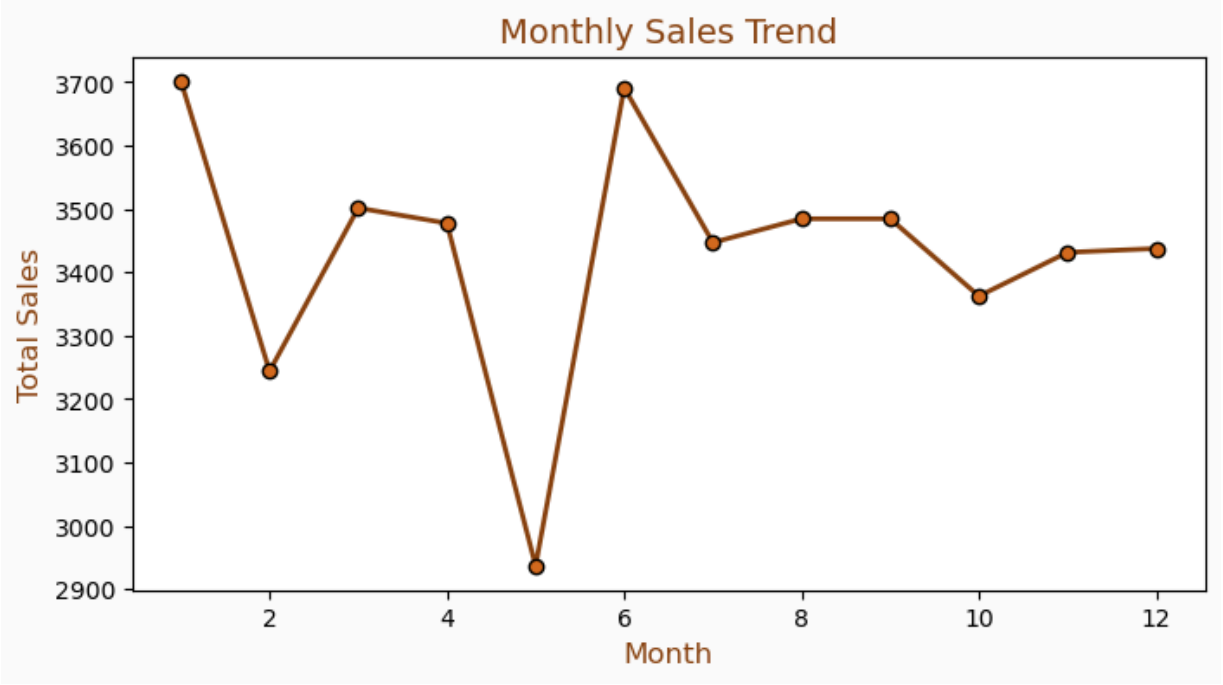
```
year
2023.0    8.910143
Name: Total Spent, dtype: float64
```

In [128…
```
month_spent = df.groupby('month')['Total Spent'].sum().sort_index()
print(month_spent)
```

```
month
1.0     3702.0
2.0     3244.0
3.0     3501.5
4.0     3477.5
5.0     2937.0
6.0     3691.0
7.0     3447.0
8.0     3484.5
9.0     3484.5
10.0    3362.5
11.0    3431.5
12.0    3437.5
Name: Total Spent, dtype: float64
```

In [129…
```
monthly_sales = df.groupby('month')['Total Spent'].sum().reset_index()

# Autumn coffee shop theme colors
line_color = '#8B4513'   # Dark coffee brown
marker_face = '#D2691E'  # Pumpkin spice orange
bg_color = "#FCFCFC"     # Creamy background
grid_color = "#ADADAD"   # Light caramel

plt.figure(figsize=(8,4), facecolor=bg_color)
plt.plot(monthly_sales['month'], monthly_sales['Total Spent'],
         marker='o', linestyle='-', linewidth=2,
```

```
        color=line_color, markerfacecolor=marker_face, markeredgecolor='black')

# Titles and labels
plt.title("Monthly Sales Trend", fontsize=14, color=line_color)
plt.xlabel("Month", fontsize=12, color=line_color)
plt.ylabel("Total Sales", fontsize=12, color=line_color)


plt.show()
```



In [130…
```
df[['Quantity', 'Price Per Unit', 'Total Spent']].describe()
```

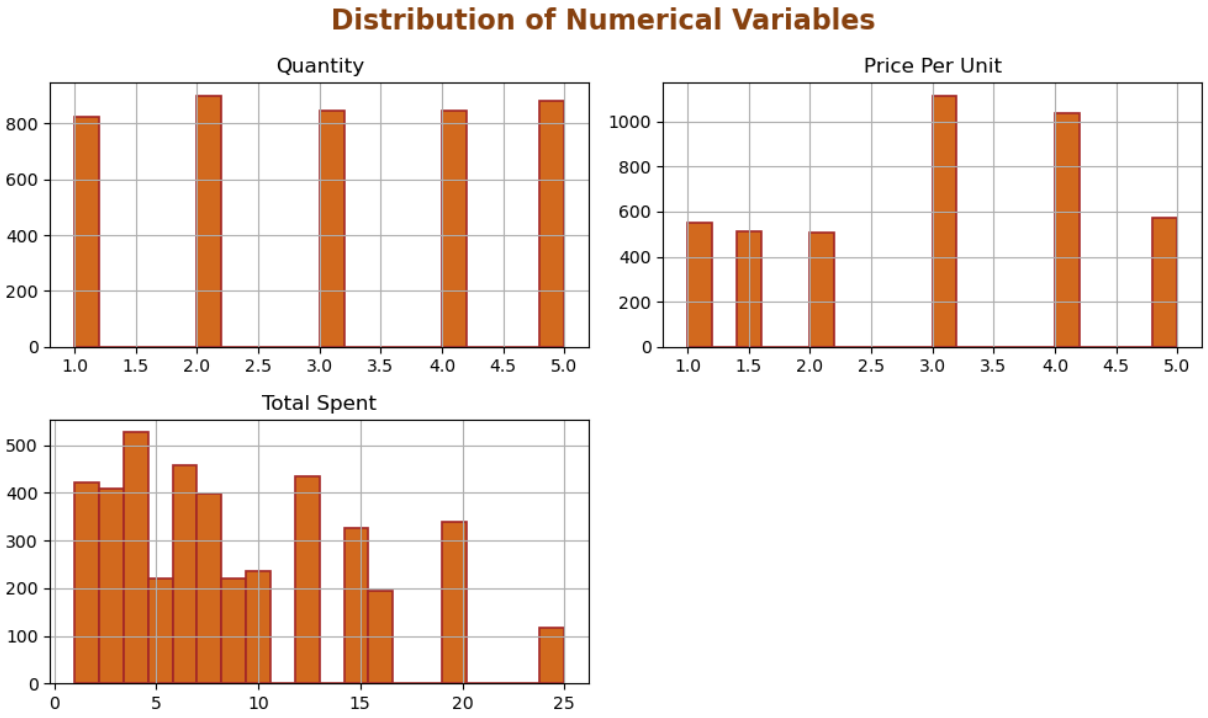
Out[130…

	Quantity	Price Per Unit	Total Spent
count	4772.000000	4740.000000	4762.000000
mean	3.024518	2.953270	8.914847
std	1.413927	1.294794	6.020300
min	1.000000	1.000000	1.000000
25%	2.000000	2.000000	4.000000
50%	3.000000	3.000000	8.000000
75%	4.000000	4.000000	12.000000
max	5.000000	5.000000	25.000000

In [131…
```
df = df.dropna(subset=['Quantity', 'Price Per Unit', 'Total Spent'])
df[['Quantity', 'Price Per Unit', 'Total Spent']].isna().sum()
```

Out[131…
```
Quantity          0
Price Per Unit    0
Total Spent       0
dtype: int64
```

In [132…
```
import matplotlib.pyplot as plt

df[['Quantity', 'Price Per Unit', 'Total Spent']].hist(
    figsize=(10, 6),
    bins=20,
    color='#D2691E',
    edgecolor='brown',
    linewidth=1.2
)

plt.suptitle('Distribution of Numerical Variables' ,
             fontsize=16, fontweight='bold', color='#8B4513')
plt.tight_layout()
plt.show()
```

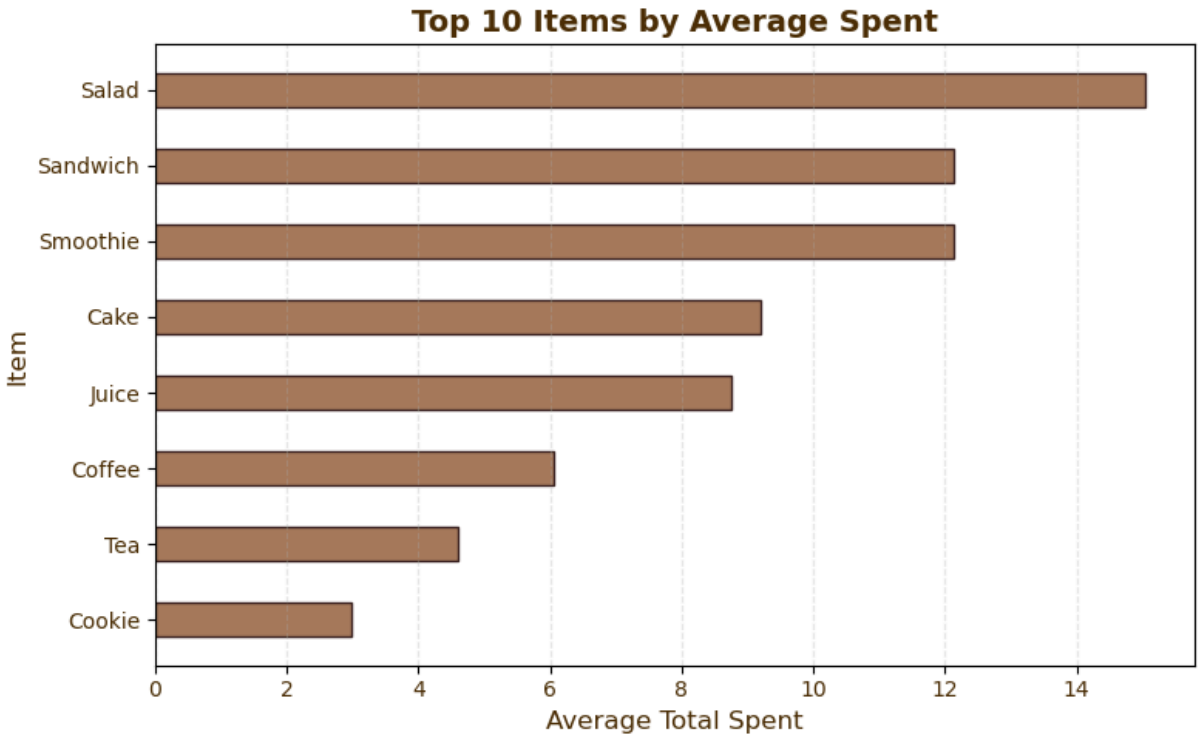


EDA PART2

In [133…
```
Average_price_per_item = (
    df.groupby('Item')['Total Spent']
    .mean()
    .sort_values(ascending=False)
    .head(10)
)

print("TOP 10 ITEMS BY AVERAGE SPENT")
print(Average_price_per_item)
```

```
import matplotlib.pyplot as plt

# Top 10 average spent per item
Average_price_per_item = (
    df.groupby('Item')['Total Spent']
    .mean()
    .sort_values(ascending=False)
    .head(10)
)


plt.figure(figsize=(8,5))
plt.barh(Average_price_per_item.index[::-1],
         Average_price_per_item.values[::-1],
         color='#A67B5B',  # single coffee color
         edgecolor='#3E2723', linewidth=1,
         height=0.45)

plt.title("Top 10 Items by Average Spent", fontsize=14, fontweight='bold', color='#
plt.xlabel("Average Total Spent", fontsize=12, color='#4B2E05')
plt.ylabel("Item", fontsize=12, color='#4B2E05')
plt.xticks(color='#4B2E05')
plt.yticks(color='#4B2E05')
plt.grid(axis='x', linestyle='--', alpha=0.3)

plt.tight_layout()
plt.show()
```

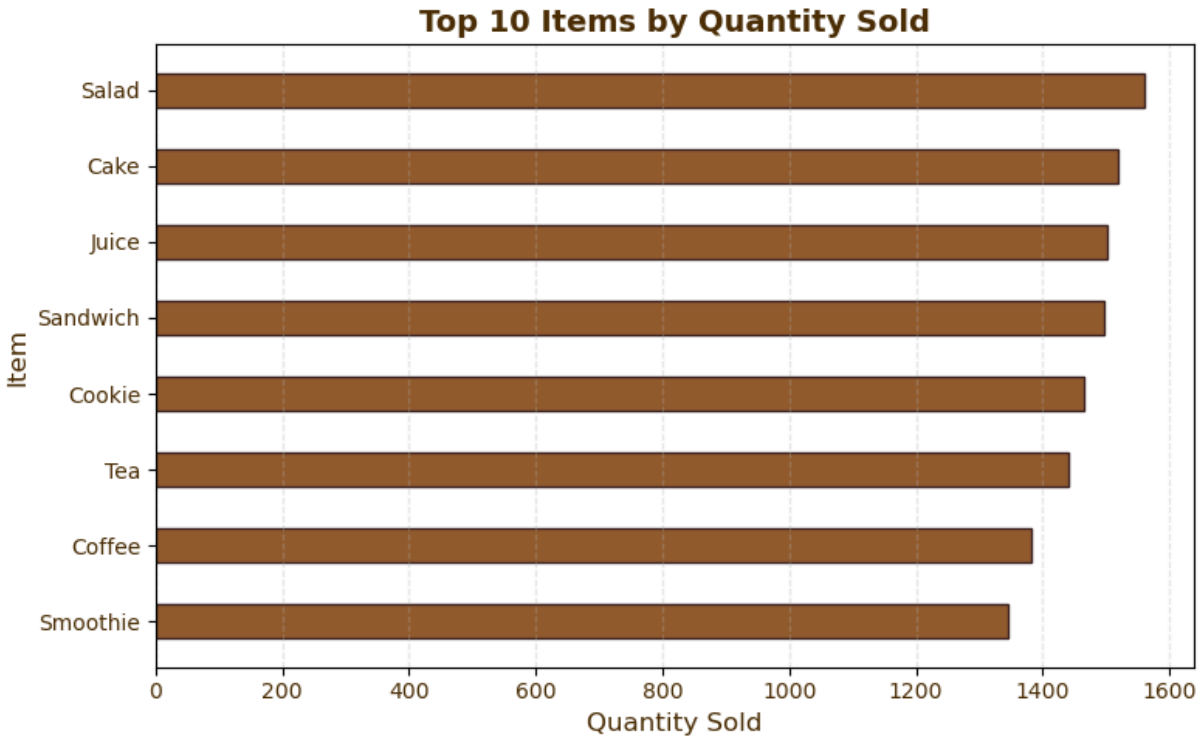
```
TOP 10 ITEMS BY AVERAGE SPENT
Item
Salad       15.038536
Sandwich    12.137931
Smoothie    12.126126
Cake         9.187500
Juice        8.760700
Coffee       6.048140
Tea          4.602128
Cookie       2.979675
Name: Total Spent, dtype: float64
```



In [134…
```
total_quantity = df.groupby('Item')['Quantity'].sum().sort_values(ascending=False).
print(total_quantity)
```

```
Item
Salad       1561.0
Cake        1519.0
Juice       1501.0
Sandwich    1496.0
Cookie      1466.0
Tea         1442.0
Coffee      1382.0
Smoothie    1346.0
Name: Quantity, dtype: float64
```

In [135…
```
import matplotlib.pyplot as plt

plt.figure(figsize=(8,5))
plt.barh(total_quantity.index[::-1],
         total_quantity.values[::-1],
         color="#935E30",  # coffee single color
         height=0.45,
         edgecolor='#3E2723', linewidth=1)

plt.title("Top 10 Items by Quantity Sold", fontsize=14, fontweight='bold', color='#
plt.xlabel("Quantity Sold", fontsize=12, color='#4B2E05')
plt.ylabel("Item", fontsize=12, color='#4B2E05')
plt.xticks(color='#4B2E05')
plt.yticks(color='#4B2E05')
plt.grid(axis='x', linestyle='--', alpha=0.3)

plt.tight_layout()
plt.show()
```


Top 10 Items by Quantity Sold
Salad
Cake
Juice
Sandwich
Cookie
Tea
Coffee
Smoothie
Item
0
200
400
600
800
1000
1200
1400
1600
Quantity Sold