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

In [2]:
```
import os
os.getcwd()
```

Out[2]: '/home/csl-4'

In [13]:
```
#df = pd.read_csv("Desktop/pokemondataset.csv")
```

In [14]:
```
# Display first 5 rows
df.head()
```

Out[14]:

	id	Name	Height(m)	Weight{kg}	HP	Attack	Defense	Sp.Atk	Sp.Def	Spe
0	1	Bulbasaur	0.7	6.9	45	49	49	65	65	
1	2	Ivysaur	1.0	13.0	60	62	63	80	80	
2	3	Venusaur	2.0	100.0	80	82	83	100	100	
3	4	Charmander	0.6	8.5	39	52	43	60	50	
4	5	Charmeleon	1.1	19.0	58	64	58	80	65	

5 rows × 24 columns

In [15]:
```
# Dataset information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 24 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   id                   1025 non-null   int64
 1   Name                 1025 non-null   object
 2   Height(m)            1025 non-null   float64
 3   Weight{kg}           1025 non-null   float64
 4   HP                   1025 non-null   int64
 5   Attack               1025 non-null   int64
 6   Defense              1025 non-null   int64
 7   Sp.Atk               1025 non-null   int64
 8   Sp.Def               1025 non-null   int64
 9   Speed                1025 non-null   int64
 10  Type_1               1025 non-null   object
 11  Type_2               526 non-null    object
 12  Is_Legendary         1025 non-null   bool
 13  Is_Mythical          1025 non-null   bool
 14  Egg_Group_1          1025 non-null   object
 15  Egg_Group_2          279 non-null    object
 16  Generation           1025 non-null   object
 17  Capture_Rate         1025 non-null   int64
 18  Base_Happiness       1025 non-null   int64
 19  Is_Baby              1025 non-null   bool
 20  Egg_Cycles           1025 non-null   int64
 21  Past_Type            24 non-null     object
 22  Is_Pseudo_Legendary  1025 non-null   bool
 23  Total_Stats          1025 non-null   int64
dtypes: bool(4), float64(2), int64(11), object(7)
memory usage: 164.3+ KB
```

In [16]:
```
# Shape of dataset
df.shape
```

Out[16]:
```
(1025, 24)
```

In [17]:
```
df.tail()
```

Out[17]:

	id	Name	Height(m)	Weight{kg}	HP	Attack	Defense	Sp.Atk	Sp.Def
1020	1021	Raging-bolt	5.2	480.0	125	73	91	137	89
1021	1022	Iron-boulder	1.5	162.5	90	120	80	68	108
1022	1023	Iron-crown	1.6	156.0	90	72	100	122	108
1023	1024	Terapagos	0.2	6.5	90	65	85	65	85
1024	1025	Pecharunt	0.3	0.3	88	88	160	88	88

5 rows × 24 columns

In [ ]:
```
#dirty cafe sales dataset
```

In [5]:
```
df = pd.read_csv("Desktop/dirty_cafe_sales.csv")
```

In [7]:
```
df.head()
```

Out[7]:

	Transaction ID	Item	Quantity	Price Per Unit	Total Spent	Payment Method	Location	Transaction Date
0	TXN_1961373	Coffee	2	2	4	Credit Card	Takeaway	2023-09-08
1	TXN_4977031	Cake	4	3	12	Cash	In-store	2023-05-16
2	TXN_4271903	Cookie	4	1	ERROR	Credit Card	In-store	2023-07-19
3	TXN_7034554	Salad	2	5	10	UNKNOWN	UNKNOWN	2023-04-27
4	TXN_3160411	Coffee	2	2	4	Digital Wallet	In-store	2023-06-11

In [9]:
```
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 [11]:
```
df.tail()
```

Out[11]:

	Transaction ID	Item	Quantity	Price Per Unit	Total Spent	Payment Method	Location	Transact D
9995	TXN_7672686	Coffee	2	2	4	NaN	UNKNOWN	2023-08
9996	TXN_9659401	NaN	3	NaN	3	Digital Wallet	NaN	2023-06
9997	TXN_5255387	Coffee	4	2	8	Digital Wallet	NaN	2023-03
9998	TXN_7695629	Cookie	3	NaN	3	Digital Wallet	NaN	2023-12
9999	TXN_6170729	Sandwich	3	4	12	Cash	In-store	2023-11

In [13]:
```
df.isnull().sum()
```

Out[13]:
```
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 [15]:
```
df.describe()
```

Out[15]:

	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_1961373	Juice	5	3	6	Digital Wallet	Takeaway	UNKNOWN
freq	1	1171	2013	2429	979	2291	3022	159

In [17]:
```
df.info()
df.shape
```

```
<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
```

Out[17]:
```
(10000, 8)
```

In [19]:
```
df.dtypes
```

Out[19]:
```
Transaction ID      object
Item                object
Quantity            object
Price Per Unit      object
Total Spent         object
Payment Method      object
Location            object
Transaction Date    object
dtype: object
```

In [21]:
```
numeric_cols = df.select_dtypes(include=np.number).columns
df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].mean())
```

In [23]:
```
# Numeric columns → fill with mean
numeric_cols = df.select_dtypes(include=np.number).columns
for col in numeric_cols:
    df[col] = df[col].fillna(df[col].mean())

# Categorical columns → fill with mode
categorical_cols = df.select_dtypes(include='object').columns
for col in categorical_cols:
    df[col] = df[col].fillna(df[col].mode()[0])
```

In [25]:
```
df.isnull().sum()
```

Out[25]:
```
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 [27]:
```
df.index
```

Out[27]:
```
RangeIndex(start=0, stop=10000, step=1)
```

In [29]:
```
df.columns
```

Out[29]:
```
Index(['Transaction ID', 'Item', 'Quantity', 'Price Per Unit', 'Total Sp
ent',
       'Payment Method', 'Location', 'Transaction Date'],
      dtype='object')
```

In [31]:
```
df.values
```

Out[31]:
```
array([['TXN_1961373', 'Coffee', '2', ..., 'Credit Card', 'Takeaway',
        '2023-09-08'],
       ['TXN_4977031', 'Cake', '4', ..., 'Cash', 'In-store',
        '2023-05-16'],
       ['TXN_4271903', 'Cookie', '4', ..., 'Credit Card', 'In-store',
        '2023-07-19'],
       ...,
       ['TXN_5255387', 'Coffee', '4', ..., 'Digital Wallet', 'Takeaway',
        '2023-03-02'],
       ['TXN_7695629', 'Cookie', '3', ..., 'Digital Wallet', 'Takeaway',
        '2023-12-02'],
       ['TXN_6170729', 'Sandwich', '3', ..., 'Cash', 'In-store',
        '2023-11-07']], dtype=object)
```

In [33]:
```
df.value_counts()
```

Out[33]:
```
Transaction ID  Item      Quantity  Price Per Unit  Total Spent  Payment
Method  Location  Transaction Date
TXN_1000555     Tea       1         1.5             1.5          Credit
Card     In-store  2023-10-19          1
TXN_6953244     Smoothie  5         4               20           Cash
Takeaway  2023-09-13          1
TXN_6933222     Tea       1         1.5             1.5          Cash
Takeaway  2023-01-22          1
TXN_6935106     Tea       5         1.5             7.5          Digital
Wallet  In-store  UNKNOWN            1
TXN_6935600     Coffee    3         2               6            UNKNOWN
Takeaway  2023-01-02          1

..
TXN_3951329     Sandwich  ERROR     4               4            Credit
Card     Takeaway  2023-12-28          1
TXN_3951460     Juice     4         4               16           Digital
Wallet  Takeaway  2023-04-21          1
TXN_3951552     Salad     2         UNKNOWN         10           Cash
In-store  2023-03-25          1
TXN_3952218     Salad     5         5               UNKNOWN      Digital
Wallet  Takeaway  2023-04-07          1
TXN_9999124     Juice     2         3               6            Digital
Wallet  Takeaway  UNKNOWN            1
Name: count, Length: 10000, dtype: int64
```

In [35]:
```
df.ndim
```

Out[35]:
```
2
```

In [37]:
```
df.empty
```

Out[37]:
```
False
```

In [39]:
```
print(df["Item"])
```

```
0         Coffee
1           Cake
2         Cookie
3          Salad
4         Coffee
           ...
9995      Coffee
9996       Juice
9997      Coffee
9998      Cookie
9999    Sandwich
Name: Item, Length: 10000, dtype: object
```

In [41]:
```
df.loc[5,'Transaction ID']
```

Out[41]:
```
'TXN_2602893'
```

In [43]:
```
df.loc[5]
```

Out[43]:
```
Transaction ID      TXN_2602893
Item                   Smoothie
Quantity                      5
Price Per Unit                4
Total Spent                  20
Payment Method      Credit Card
Location               Takeaway
Transaction Date     2023-03-31
Name: 5, dtype: object
```

In [45]:
```
print(df.loc[2:5])
```

```
  Transaction ID      Item Quantity Price Per Unit Total Spent  \
2    TXN_4271903    Cookie        4              1       ERROR
3    TXN_7034554     Salad        2              5          10
4    TXN_3160411    Coffee        2              2           4
5    TXN_2602893  Smoothie        5              4          20

   Payment Method  Location Transaction Date
2     Credit Card  In-store       2023-07-19
3         UNKNOWN   UNKNOWN       2023-04-27
4  Digital Wallet  In-store       2023-06-11
5     Credit Card  Takeaway       2023-03-31
```

In [47]:
```
print(df[df.Item=='Cookie'])
```

```
     Transaction ID    Item Quantity Price Per Unit Total Spent  \
2       TXN_4271903  Cookie        4              1       ERROR
13      TXN_9437049  Cookie        5              1           5
26      TXN_5183041  Cookie        5              1           5
44      TXN_1491578  Cookie        2              1           2
55      TXN_5522862  Cookie    ERROR              1           2
...             ...     ...      ...            ...         ...
9945    TXN_8153550  Cookie        2              1           2
9956    TXN_1958525  Cookie        3              1           3
9976    TXN_3528020  Cookie        1              1           1
9982    TXN_8567525  Cookie        2              1           2
9998    TXN_7695629  Cookie        3              3           3

      Payment Method  Location Transaction Date
2        Credit Card  In-store       2023-07-19
13    Digital Wallet  Takeaway       2023-06-01
26       Credit Card  In-store       2023-04-20
44    Digital Wallet  Takeaway       2023-02-23
55       Credit Card  Takeaway       2023-03-19
...              ...       ...              ...
9945            Cash  Takeaway       2023-10-12
9956  Digital Wallet  Takeaway       2023-10-08
9976  Digital Wallet  Takeaway       2023-08-26
9982  Digital Wallet  Takeaway       2023-12-30
9998  Digital Wallet  Takeaway       2023-12-02

[1092 rows x 8 columns]
```

In [49]:
```
print(df[(df.Item=='Cookie') ])
```

```
     Transaction ID    Item Quantity Price Per Unit Total Spent  \
2       TXN_4271903  Cookie        4              1       ERROR
13      TXN_9437049  Cookie        5              1           5
26      TXN_5183041  Cookie        5              1           5
44      TXN_1491578  Cookie        2              1           2
55      TXN_5522862  Cookie    ERROR              1           2
...             ...     ...      ...            ...         ...
9945    TXN_8153550  Cookie        2              1           2
9956    TXN_1958525  Cookie        3              1           3
9976    TXN_3528020  Cookie        1              1           1
9982    TXN_8567525  Cookie        2              1           2
9998    TXN_7695629  Cookie        3              3           3

      Payment Method  Location Transaction Date
2        Credit Card  In-store       2023-07-19
13   Digital Wallet  Takeaway       2023-06-01
26       Credit Card  In-store       2023-04-20
44   Digital Wallet  Takeaway       2023-02-23
55       Credit Card  Takeaway       2023-03-19
...              ...       ...              ...
9945            Cash  Takeaway       2023-10-12
9956  Digital Wallet  Takeaway       2023-10-08
9976  Digital Wallet  Takeaway       2023-08-26
9982  Digital Wallet  Takeaway       2023-12-30
9998  Digital Wallet  Takeaway       2023-12-02

[1092 rows x 8 columns]
```

In [51]:
```
print(df)
```

```
     Transaction ID      Item Quantity Price Per Unit Total Spent  \
0       TXN_1961373    Coffee        2              2           4
1       TXN_4977031      Cake        4              3          12
2       TXN_4271903    Cookie        4              1       ERROR
3       TXN_7034554     Salad        2              5          10
4       TXN_3160411    Coffee        2              2           4
...             ...       ...      ...            ...         ...
9995    TXN_7672686    Coffee        2              2           4
9996    TXN_9659401     Juice        3              3           3
9997    TXN_5255387    Coffee        4              2           8
9998    TXN_7695629    Cookie        3              3           3
9999    TXN_6170729  Sandwich        3              4          12

      Payment Method  Location Transaction Date
0        Credit Card  Takeaway       2023-09-08
1               Cash  In-store       2023-05-16
2        Credit Card  In-store       2023-07-19
3            UNKNOWN   UNKNOWN       2023-04-27
4     Digital Wallet  In-store       2023-06-11
...              ...       ...              ...
9995  Digital Wallet   UNKNOWN       2023-08-30
9996  Digital Wallet  Takeaway       2023-06-02
9997  Digital Wallet  Takeaway       2023-03-02
9998  Digital Wallet  Takeaway       2023-12-02
9999            Cash  In-store       2023-11-07

[10000 rows x 8 columns]
```

In [53]:
```
#pd.set_option('display.max_rows', None)
#pd.set_option('display.max_columns', None)
```

```
display(df)
```

	Transaction ID	Item	Quantity	Price Per Unit	Total Spent	Payment Method	Location	Transac I
0	TXN_1961373	Coffee	2	2	4	Credit Card	Takeaway	2023-0
1	TXN_4977031	Cake	4	3	12	Cash	In-store	2023-0
2	TXN_4271903	Cookie	4	1	ERROR	Credit Card	In-store	2023-0
3	TXN_7034554	Salad	2	5	10	UNKNOWN	UNKNOWN	2023-0
4	TXN_3160411	Coffee	2	2	4	Digital Wallet	In-store	2023-0
...	...	...	...	...	...	...	...	
9995	TXN_7672686	Coffee	2	2	4	Digital Wallet	UNKNOWN	2023-0
9996	TXN_9659401	Juice	3	3	3	Digital Wallet	Takeaway	2023-0
9997	TXN_5255387	Coffee	4	2	8	Digital Wallet	Takeaway	2023-0
9998	TXN_7695629	Cookie	3	3	3	Digital Wallet	Takeaway	2023-1
9999	TXN_6170729	Sandwich	3	4	12	Cash	In-store	2023-1

10000 rows × 8 columns

In [55]:
```
#df.isnull()
```

In [57]:
```
#df.isna()
```

In [59]:
```
print(df.isnull().values.any())
```

```
False
```

In [61]:
```
print(df.isna().values.any())
```

```
False
```

In [63]:
```
for col in numeric_cols:
    plt.boxplot(df[col])
    plt.title(f"Outliers in {col}")
    plt.show()
```

In [65]:
```
for col in numeric_cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    df = df[(df[col] >= Q1 - 1.5*IQR) & (df[col] <= Q3 + 1.5*IQR)]
```

In [67]:
```
print(df.isnull().values.any())
```

```
False
```

In [69]:
```
print(df.isna().values.any())
```

```
False
```

In [71]:
```
print(df.duplicated().values.any())
```

```
False
```

In [146…
```
#data_missing = pd.read_csv("https://storage.googleapis.com/dqlab-dataset
#print(data_missing.isnull().values.any())

import pandas as pd

# Load the CSV file (change path if needed)
data_missing = pd.read_csv("Desktop/dirty_cafe_sales.csv")

# Check if any missing values exist
print(data_missing.isnull().values.any())

# Count missing values column-wise
print(data_missing.isnull().sum())
```

```
True
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 [148…
```
data_missing.dropna()
```

Out[148…

	Transaction ID	Item	Quantity	Price Per Unit	Total Spent	Payment Method	Location	Tra
0	TXN_1961373	Coffee	2	2	4	Credit Card	Takeaway	20
1	TXN_4977031	Cake	4	3	12	Cash	In-store	20
2	TXN_4271903	Cookie	4	1	ERROR	Credit Card	In-store	20
3	TXN_7034554	Salad	2	5	10	UNKNOWN	UNKNOWN	20
4	TXN_3160411	Coffee	2	2	4	Digital Wallet	In-store	20
…	…	…	…	…	…	…	…	
9984	TXN_3142496	Smoothie	UNKNOWN	4	4	Cash	Takeaway	20
9986	TXN_2858441	Sandwich	2	4	8	Credit Card	In-store	20
9991	TXN_3897619	Sandwich	3	4	12	Cash	Takeaway	20
9992	TXN_2739140	Smoothie	4	4	16	UNKNOWN	In-store	20
9999	TXN_6170729	Sandwich	3	4	12	Cash	In-store	20

4550 rows × 8 columns

In [166…
```
data_missing.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 [168…
```
data_missing.mean(numeric_only=True)
```

Out[168…
```
Series([], dtype: float64)
```

In [170…
```
print(data_missing.describe())
```

```
       Transaction ID   Item Quantity Price Per Unit Total Spent  \
count           10000   9667     9862           9821        9827   
unique          10000     10        7              8          19   
top       TXN_1961373  Juice        5              3           6   
freq                1   1171     2013           2429         979   

        Payment Method  Location Transaction Date  
count             7421      6735             9841  
unique               5         4              367  
top     Digital Wallet  Takeaway          UNKNOWN  
freq              2291      3022              159  
```

In [172…
```
print(data_missing.describe(include="all"))
```

```
       Transaction ID   Item Quantity Price Per Unit Total Spent  \
count           10000   9667     9862           9821        9827   
unique          10000     10        7              8          19   
top       TXN_1961373  Juice        5              3           6   
freq                1   1171     2013           2429         979   

        Payment Method  Location Transaction Date  
count             7421      6735             9841  
unique               5         4              367  
top     Digital Wallet  Takeaway          UNKNOWN  
freq              2291      3022              159  
```

In [174…
```
data_missing.columns
```

Out[174…
```
Index(['Transaction ID', 'Item', 'Quantity', 'Price Per Unit', 'Total Sp
ent',
       'Payment Method', 'Location', 'Transaction Date'],
      dtype='object')
```

In [178…
```
data_missing['Quantity'] = pd.to_numeric(data_missing['Quantity'], errors
data_missing['Price Per Unit'] = pd.to_numeric(data_missing['Price Per Un
data_missing['Total Spent'] = pd.to_numeric(data_missing['Total Spent'],
```

In [180…
```
data_missing.describe(include=[np.number])
```

Out[180…

	Quantity	Price Per Unit	Total Spent
count	9521.000000	9467.000000	9498.000000
mean	3.028463	2.949984	8.924352
std	1.419007	1.278450	6.009919
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 [182…
```
print(data_missing.describe())
```

```
          Quantity  Price Per Unit  Total Spent
count  9521.000000     9467.000000  9498.000000
mean      3.028463        2.949984     8.924352
std       1.419007        1.278450     6.009919
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 [184…
```
print(data_missing.describe(include="all"))
```

```
       Transaction ID   Item     Quantity  Price Per Unit  Total Spent  \
count           10000   9667  9521.000000     9467.000000  9498.000000
unique          10000     10          NaN             NaN          NaN
top       TXN_1961373  Juice          NaN             NaN          NaN
freq                1   1171          NaN             NaN          NaN
mean              NaN    NaN     3.028463        2.949984     8.924352
std               NaN    NaN     1.419007        1.278450     6.009919
min               NaN    NaN     1.000000        1.000000     1.000000
25%               NaN    NaN     2.000000        2.000000     4.000000
50%               NaN    NaN     3.000000        3.000000     8.000000
75%               NaN    NaN     4.000000        4.000000    12.000000
max               NaN    NaN     5.000000        5.000000    25.000000

       Payment Method  Location Transaction Date
count            7421      6735             9841
unique              5         4              367
top     Digital Wallet  Takeaway          UNKNOWN
freq             2291      3022              159
mean              NaN       NaN              NaN
std               NaN       NaN              NaN
min               NaN       NaN              NaN
25%               NaN       NaN              NaN
50%               NaN       NaN              NaN
75%               NaN       NaN              NaN
max               NaN       NaN              NaN
```

In [186…
```
data_missing.describe(include=[np.number])
```

Out[186…

	Quantity	Price Per Unit	Total Spent
count	9521.000000	9467.000000	9498.000000
mean	3.028463	2.949984	8.924352
std	1.419007	1.278450	6.009919
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 [188…
```
print(data_missing.describe(exclude=["0"]))
```

```
          Quantity  Price Per Unit  Total Spent
count  9521.000000     9467.000000  9498.000000
mean      3.028463        2.949984     8.924352
std       1.419007        1.278450     6.009919
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 [190…
```
data_missing.isnull()
```

Out[190…

	Transaction ID	Item	Quantity	Price Per Unit	Total Spent	Payment Method	Location	Transaction Date
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	True	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
…	…	…	…	…	…	…	…	…
9995	False	False	False	False	False	True	False	False
9996	False	True	False	True	False	False	True	False
9997	False	False	False	False	False	False	True	False
9998	False	False	False	True	False	False	True	False
9999	False	False	False	False	False	False	False	False

10000 rows × 8 columns

In [192…
```
data_missing.isna()
```

Out[192…

	id	Name	Height(m)	Weight{kg}	HP	Attack	Defense	Sp.Atk	Sp.Def	S
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	
…	…	…	…	…	…	…	…	…	…	
1020	False	False	False	False	False	False	False	False	False	
1021	False	False	False	False	False	False	False	False	False	
1022	False	False	False	False	False	False	False	False	False	
1023	False	False	False	False	False	False	False	False	False	
1024	False	False	False	False	False	False	False	False	False	

1025 rows × 24 columns

In [196…

```
print(data_missing.isnull().values.any())
print(data_missing.isna().values.any())
print(data_missing.duplicated().values.any())
```

```
True
True
False
```

In [198…

```
data_missing=data_missing.dropna()
data_missing.dropna()
```

Out[198…

	Transaction ID	Item	Quantity	Price Per Unit	Total Spent	Payment Method	Location	Tran
0	TXN_1961373	Coffee	2.0	2.0	4.0	Credit Card	Takeaway	202
1	TXN_4977031	Cake	4.0	3.0	12.0	Cash	In-store	202
3	TXN_7034554	Salad	2.0	5.0	10.0	UNKNOWN	UNKNOWN	202
4	TXN_3160411	Coffee	2.0	2.0	4.0	Digital Wallet	In-store	202
6	TXN_4433211	UNKNOWN	3.0	3.0	9.0	ERROR	Takeaway	202
…	…	…	…	…	…	…	…	
9979	TXN_9933628	Smoothie	5.0	4.0	20.0	Cash	In-store	202
9986	TXN_2858441	Sandwich	2.0	4.0	8.0	Credit Card	In-store	202
9991	TXN_3897619	Sandwich	3.0	4.0	12.0	Cash	Takeaway	202
9992	TXN_2739140	Smoothie	4.0	4.0	16.0	UNKNOWN	In-store	202
9999	TXN_6170729	Sandwich	3.0	4.0	12.0	Cash	In-store	202

4096 rows × 8 columns

In [200…
```
print(data_missing.isnull().values.any())
```

```
False
```

In [202…
```
print(data_missing.isna().values.any())
```

```
False
```

In [206…
```
data_missing.dtypes
```

Out[206…
```
Transaction ID       object
Item                 object
Quantity            float64
Price Per Unit      float64
Total Spent         float64
Payment Method       object
Location             object
Transaction Date     object
dtype: object
```

In [210…
```
#df[['Quantity', 'Price Per Unit', 'Total Spent']].mean()
data_missing[['Quantity', 'Price Per Unit', 'Total Spent']].mean()
```

Out[210…
```
Quantity          3.013916
Price Per Unit    2.954712
Total Spent       8.906738
dtype: float64
```

In [212…
```
data_missing[['Quantity', 'Price Per Unit', 'Total Spent']].median()
```

Out[212…
```
Quantity          3.0
Price Per Unit    3.0
Total Spent       8.0
dtype: float64
```

In [224…
```
data_filling = data_missing.fillna(data_missing.mean(numeric_only=True))
```

In [226…
```
print(data_filling.head(10))
```

```
   Transaction ID      Item  Quantity  Price Per Unit  Total Spent  \
0     TXN_1961373    Coffee       2.0             2.0          4.0
1     TXN_4977031      Cake       4.0             3.0         12.0
3     TXN_7034554     Salad       2.0             5.0         10.0
4     TXN_3160411    Coffee       2.0             2.0          4.0
6     TXN_4433211   UNKNOWN       3.0             3.0          9.0
7     TXN_6699534  Sandwich       4.0             4.0         16.0
10    TXN_2548360     Salad       5.0             5.0         25.0
11    TXN_3051279  Sandwich       2.0             4.0          8.0
12    TXN_7619095  Sandwich       2.0             4.0          8.0
15    TXN_2847255     Salad       3.0             5.0         15.0

    Payment Method  Location Transaction Date
0      Credit Card  Takeaway       2023-09-08
1             Cash  In-store       2023-05-16
3          UNKNOWN   UNKNOWN       2023-04-27
4   Digital Wallet  In-store       2023-06-11
6            ERROR  Takeaway       2023-10-06
7             Cash   UNKNOWN       2023-10-28
10            Cash  Takeaway       2023-11-07
11     Credit Card  Takeaway            ERROR
12            Cash  In-store       2023-05-03
15     Credit Card  In-store       2023-11-15
```

In [216…
```
data_filling = data_missing.fillna(data_missing.median(numeric_only=True)
```

In [218…
```
print(data_filling.head(10))
```

```
   Transaction ID      Item  Quantity  Price Per Unit  Total Spent  \
0     TXN_1961373    Coffee       2.0             2.0          4.0
1     TXN_4977031      Cake       4.0             3.0         12.0
3     TXN_7034554     Salad       2.0             5.0         10.0
4     TXN_3160411    Coffee       2.0             2.0          4.0
6     TXN_4433211   UNKNOWN       3.0             3.0          9.0
7     TXN_6699534  Sandwich       4.0             4.0         16.0
10    TXN_2548360     Salad       5.0             5.0         25.0
11    TXN_3051279  Sandwich       2.0             4.0          8.0
12    TXN_7619095  Sandwich       2.0             4.0          8.0
15    TXN_2847255     Salad       3.0             5.0         15.0

    Payment Method  Location Transaction Date
0      Credit Card  Takeaway       2023-09-08
1             Cash  In-store       2023-05-16
3          UNKNOWN   UNKNOWN       2023-04-27
4   Digital Wallet  In-store       2023-06-11
6            ERROR  Takeaway       2023-10-06
7             Cash   UNKNOWN       2023-10-28
10            Cash  Takeaway       2023-11-07
11     Credit Card  Takeaway            ERROR
12            Cash  In-store       2023-05-03
15     Credit Card  In-store       2023-11-15
```

In [244…
```
# Remove leading/trailing spaces from column names
data_missing.columns = data_missing.columns.str.strip()

# Convert Total Spent to float safely
data_missing.loc[:, 'Total Spent'] = data_missing['Total Spent'].astype(f

print(data_missing['Total Spent'])
```

```
0        4.0
1       12.0
3       10.0
4        4.0
6        9.0
        ...
9979    20.0
9986     8.0
9991    12.0
9992    16.0
9999    12.0
Name: Total Spent, Length: 4096, dtype: float64
```

In [246…
```
data_missing.dtypes
```

Out[246…
```
Transaction ID       object
Item                 object
Quantity            float64
Price Per Unit      float64
Total Spent         float64
Payment Method       object
Location             object
Transaction Date     object
dtype: object
```

In [250…
```
categorical_cols = data_missing.select_dtypes(include='object').columns.t
```

In [252…
```
df_encoded = pd.get_dummies(data_missing, columns=categorical_cols, drop_
```

In [254…
```
print("Data after encoding categorical variables:")
print(df_encoded.head())
```

```
Data after encoding categorical variables:
   Quantity  Price Per Unit  Total Spent  Transaction ID_TXN_1001832  \
0       2.0             2.0          4.0                       False
1       4.0             3.0         12.0                       False
3       2.0             5.0         10.0                       False
4       2.0             2.0          4.0                       False
6       3.0             3.0          9.0                       False

   Transaction ID_TXN_1002457  Transaction ID_TXN_1004184  \
0                       False                       False
1                       False                       False
3                       False                       False
4                       False                       False
6                       False                       False

   Transaction ID_TXN_1004563  Transaction ID_TXN_1005331  \
0                       False                       False
1                       False                       False
3                       False                       False
4                       False                       False
6                       False                       False

   Transaction ID_TXN_1006942  Transaction ID_TXN_1007347  ...  \
0                       False                       False  ...
1                       False                       False  ...
3                       False                       False  ...
4                       False                       False  ...
6                       False                       False  ...

   Transaction Date_2023-12-24  Transaction Date_2023-12-25  \
0                        False                        False
1                        False                        False
3                        False                        False
4                        False                        False
6                        False                        False

   Transaction Date_2023-12-26  Transaction Date_2023-12-27  \
0                        False                        False
1                        False                        False
3                        False                        False
4                        False                        False
6                        False                        False

   Transaction Date_2023-12-28  Transaction Date_2023-12-29  \
0                        False                        False
1                        False                        False
3                        False                        False
4                        False                        False
6                        False                        False

   Transaction Date_2023-12-30  Transaction Date_2023-12-31  \
0                        False                        False
1                        False                        False
3                        False                        False
4                        False                        False
6                        False                        False

   Transaction Date_ERROR  Transaction Date_UNKNOWN
0                   False                     False
1                   False                     False
```

```
3                    False                    False
4                    False                    False
6                    False                    False

[5 rows x 4480 columns]
```

In [256…
```
for col in numeric_cols:
    Q1 = df_encoded[col].quantile(0.25)
    Q3 = df_encoded[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    # Cap outliers
    df_encoded[col] = np.where(df_encoded[col] < lower_bound, lower_bound
    df_encoded[col] = np.where(df_encoded[col] > upper_bound, upper_bound
```

In [260…
```
# Select object type columns
categorical_cols = data_missing.select_dtypes(include='object').columns.t
print("Categorical columns:", categorical_cols)
```

```
Categorical columns: ['Transaction ID', 'Item', 'Payment Method', 'Locatio
n', 'Transaction Date']
```

In [264…
```
# Strip spaces and replace spaces with underscores
data_missing.columns = data_missing.columns.str.strip().str.replace(' ',
print(data_missing.columns.tolist())
```

```
['Transaction_ID', 'Item', 'Quantity', 'Price_Per_Unit', 'Total_Spent', 'P
ayment_Method', 'Location', 'Transaction_Date']
```

In [266…
```
cols_to_encode = ['Item', 'Payment_Method', 'Location']

df_encoded = pd.get_dummies(data_missing, columns=cols_to_encode, drop_fi

print("Dataset after One-Hot Encoding:")
print(df_encoded.head())
```

```
Dataset after One-Hot Encoding:
  Transaction_ID  Quantity  Price_Per_Unit  Total_Spent Transaction_Date
\
0    TXN_1961373       2.0             2.0          4.0       2023-09-08
1    TXN_4977031       4.0             3.0         12.0       2023-05-16
3    TXN_7034554       2.0             5.0         10.0       2023-04-27
4    TXN_3160411       2.0             2.0          4.0       2023-06-11
6    TXN_4433211       3.0             3.0          9.0       2023-10-06

   Item_Coffee  Item_Cookie  Item_ERROR  Item_Juice  Item_Salad  ...  \
0         True        False       False       False       False  ...
1        False        False       False       False       False  ...
3        False        False       False       False        True  ...
4         True        False       False       False       False  ...
6        False        False       False       False       False  ...

   Item_Smoothie  Item_Tea  Item_UNKNOWN  Payment_Method_Credit Card  \
0          False     False         False                        True
1          False     False         False                       False
3          False     False         False                       False
4          False     False         False                       False
6          False     False          True                       False

   Payment_Method_Digital Wallet  Payment_Method_ERROR  \
0                          False                 False
1                          False                 False
3                          False                 False
4                           True                 False
6                          False                  True

   Payment_Method_UNKNOWN  Location_In-store  Location_Takeaway  \
0                   False              False               True
1                   False               True              False
3                    True              False              False
4                   False               True              False
6                   False              False               True

   Location_UNKNOWN
0             False
1             False
3              True
4             False
6             False

[5 rows x 21 columns]
```

In [270…

```
from sklearn.preprocessing import LabelEncoder
df_label = data_missing.copy()
le = LabelEncoder()

for col in cols_to_encode:
    df_label[col] = le.fit_transform(df_label[col])

print("Dataset after Label Encoding:")
print(df_label.head())
```

```
Dataset after Label Encoding:
  Transaction_ID  Item  Quantity  Price_Per_Unit  Total_Spent  Payment_Met
hod  \
0    TXN_1961373     1       2.0             2.0          4.0
1
1    TXN_4977031     0       4.0             3.0         12.0
0
3    TXN_7034554     5       2.0             5.0         10.0
4
4    TXN_3160411     1       2.0             2.0          4.0
2
6    TXN_4433211     9       3.0             3.0          9.0
3

   Location Transaction_Date
0         2       2023-09-08
1         1       2023-05-16
3         3       2023-04-27
4         1       2023-06-11
6         2       2023-10-06
```

In [274…

```
print("Info of OHE dataset:")
print(df_encoded.info())

print("Info of Label Encoded dataset:")
print(df_label.info())
```

```
Info of OHE dataset:
<class 'pandas.core.frame.DataFrame'>
Index: 4096 entries, 0 to 9999
Data columns (total 21 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Transaction_ID                  4096 non-null   object
 1   Quantity                        4096 non-null   float64
 2   Price_Per_Unit                  4096 non-null   float64
 3   Total_Spent                     4096 non-null   float64
 4   Transaction_Date                4096 non-null   object
 5   Item_Coffee                     4096 non-null   bool
 6   Item_Cookie                     4096 non-null   bool
 7   Item_ERROR                      4096 non-null   bool
 8   Item_Juice                      4096 non-null   bool
 9   Item_Salad                      4096 non-null   bool
 10  Item_Sandwich                   4096 non-null   bool
 11  Item_Smoothie                   4096 non-null   bool
 12  Item_Tea                        4096 non-null   bool
 13  Item_UNKNOWN                    4096 non-null   bool
 14  Payment_Method_Credit Card      4096 non-null   bool
 15  Payment_Method_Digital Wallet   4096 non-null   bool
 16  Payment_Method_ERROR            4096 non-null   bool
 17  Payment_Method_UNKNOWN          4096 non-null   bool
 18  Location_In-store               4096 non-null   bool
 19  Location_Takeaway               4096 non-null   bool
 20  Location_UNKNOWN                4096 non-null   bool
dtypes: bool(16), float64(3), object(2)
memory usage: 256.0+ KB
None
Info of Label Encoded dataset:
<class 'pandas.core.frame.DataFrame'>
Index: 4096 entries, 0 to 9999
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Transaction_ID    4096 non-null   object
 1   Item              4096 non-null   int64
 2   Quantity          4096 non-null   float64
 3   Price_Per_Unit    4096 non-null   float64
 4   Total_Spent       4096 non-null   float64
 5   Payment_Method    4096 non-null   int64
 6   Location          4096 non-null   int64
 7   Transaction_Date  4096 non-null   object
dtypes: float64(3), int64(3), object(2)
memory usage: 288.0+ KB
None
```

In [282…

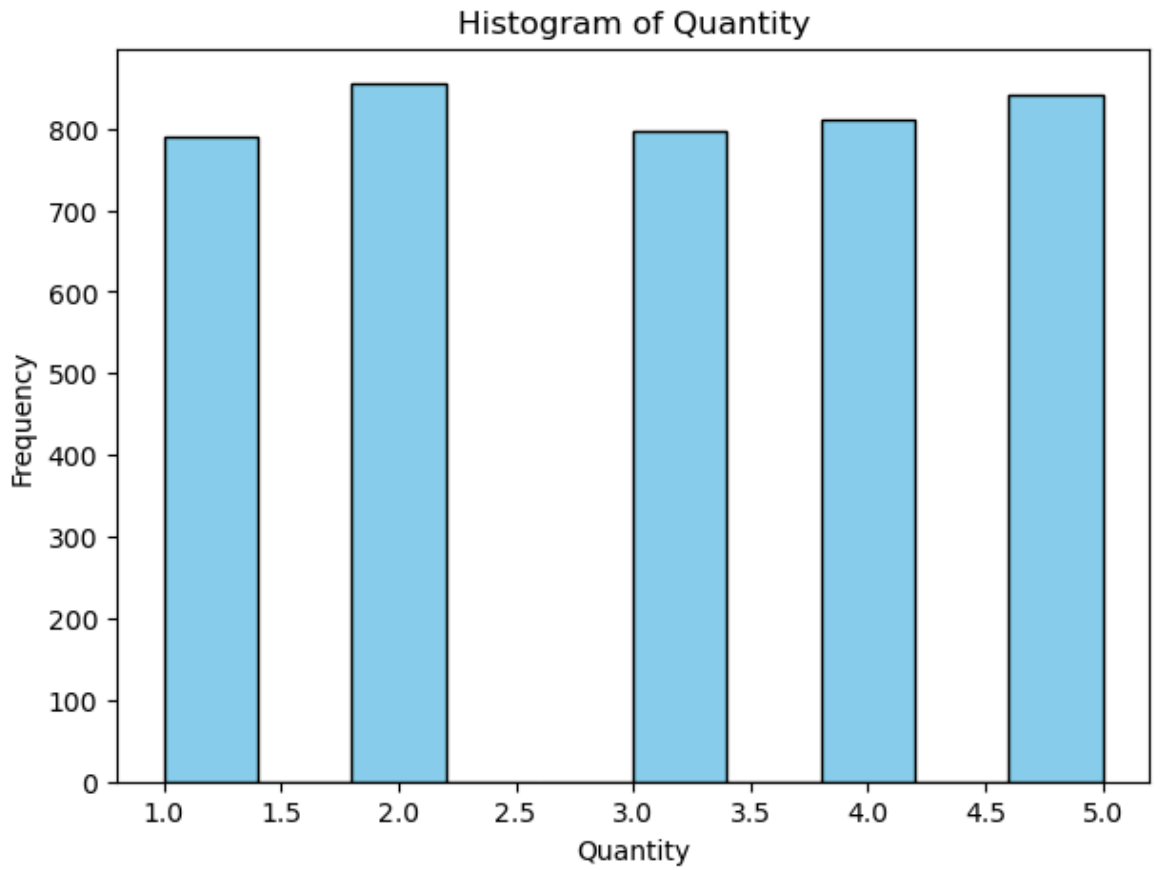
```
if 'Quantity' in data_missing.columns:
    data_missing['Quantity_Binned'] = pd.cut(data_missing['Quantity'], bi
    print("\nQuantity after binning:")
    print(data_missing[['Quantity', 'Quantity_Binned']].head())
```

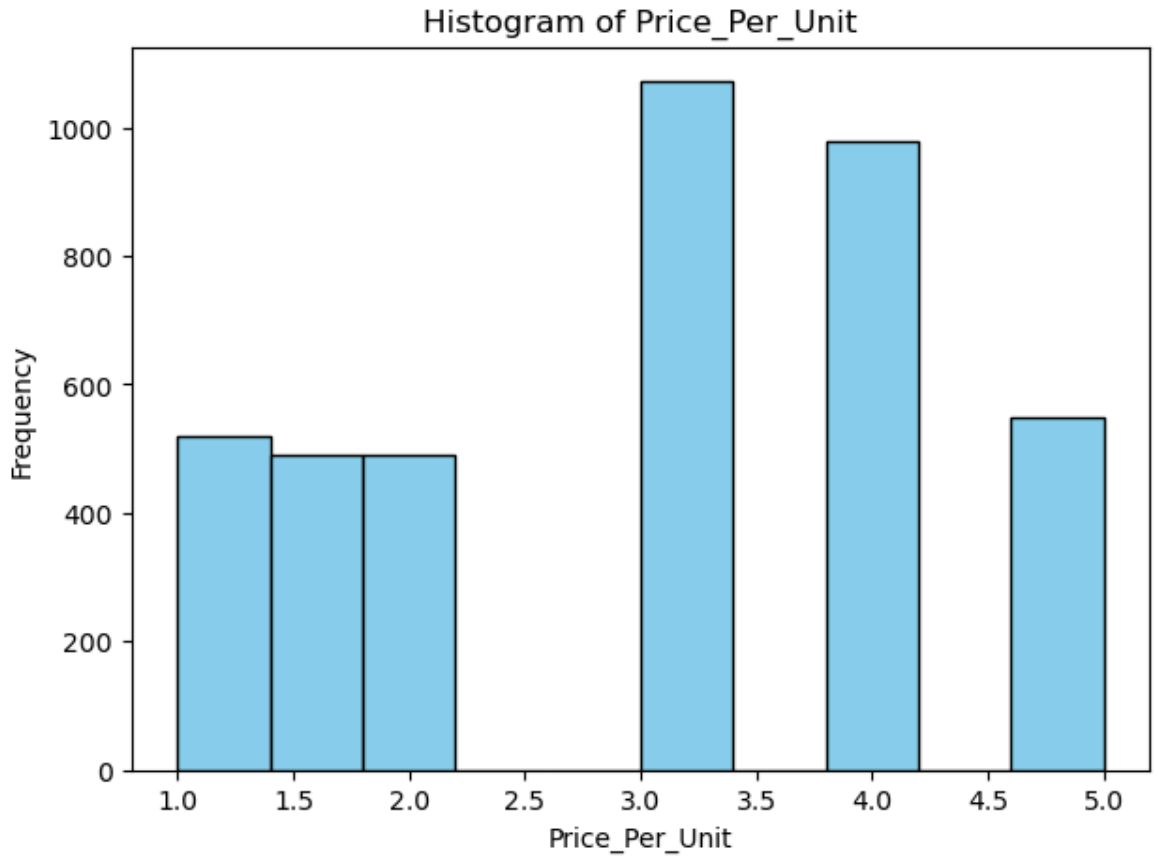
```
Quantity after binning:
   Quantity Quantity_Binned
0       2.0             Low
1       4.0             Low
3       2.0             Low
4       2.0             Low
6       3.0             Low
```

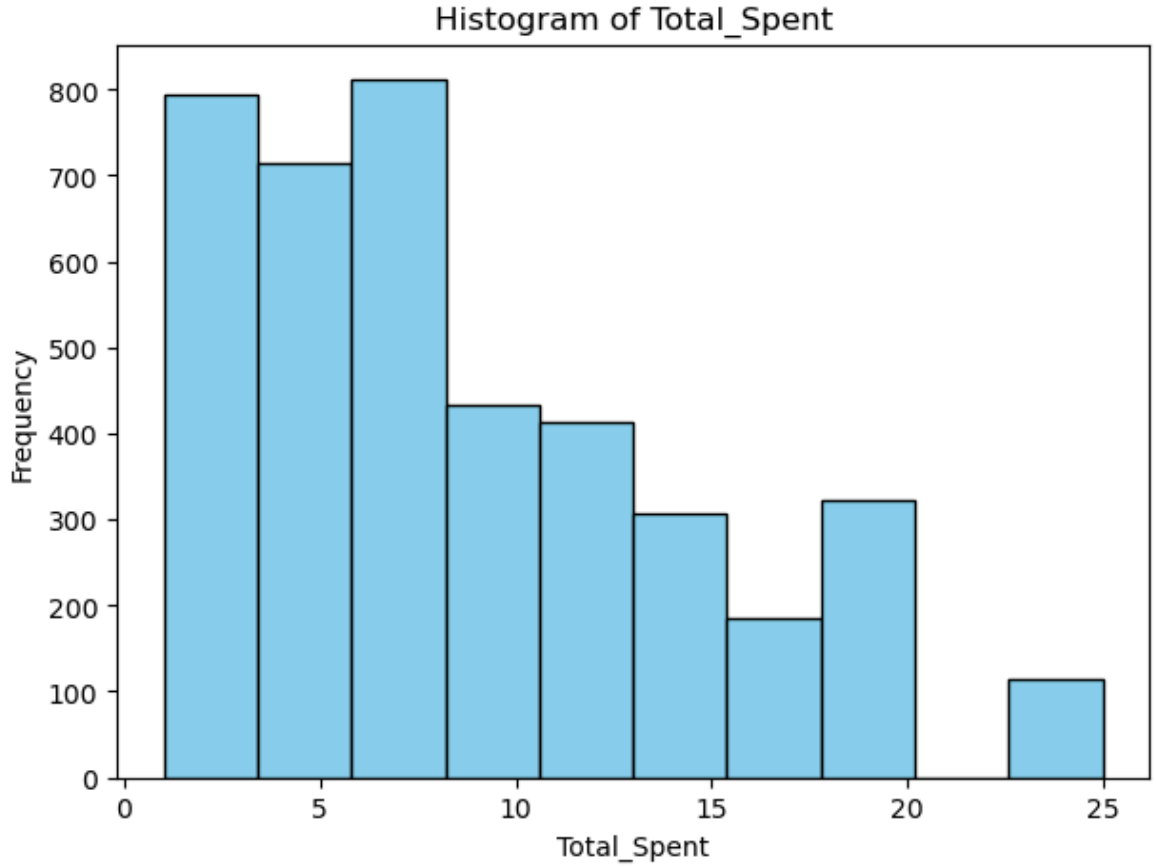
In [284…
```
if 'Price_Per_Unit' in data_missing.columns:
    data_missing['Price_Binned'] = pd.cut(data_missing['Price_Per_Unit'],
    print("\nPrice_Per_Unit after binning:")
    print(data_missing[['Price_Per_Unit', 'Price_Binned']].head())
```

```
Price_Per_Unit after binning:
   Price_Per_Unit Price_Binned
0             2.0        Cheap
1             3.0     Moderate
3             5.0    Expensive
4             2.0        Cheap
6             3.0     Moderate
```

In [286…
```
numeric_cols = data_missing.select_dtypes(include=np.number).columns.toli
for col in numeric_cols:
    plt.figure(figsize=(7, 5))
    plt.hist(data_missing[col].dropna(), bins=10, color='skyblue', edgeco
    plt.title(f'Histogram of {col}')
    plt.xlabel(col)
    plt.ylabel('Frequency')
    plt.show()
```







In [288…

```
if 'Payment_Method' in data_missing.columns:
    df_payment_ohe = pd.get_dummies(data_missing, columns=['Payment_Metho
    print("\nOne-Hot Encoding on Payment_Method:")
    print(df_payment_ohe.head())
```

```
One-Hot Encoding on Payment_Method:
  Transaction_ID     Item  Quantity  Price_Per_Unit  Total_Spent  Location
\
0    TXN_1961373   Coffee       2.0             2.0          4.0  Takeaway
1    TXN_4977031     Cake       4.0             3.0         12.0  In-store
3    TXN_7034554    Salad       2.0             5.0         10.0   UNKNOWN
4    TXN_3160411   Coffee       2.0             2.0          4.0  In-store
6    TXN_4433211  UNKNOWN       3.0             3.0          9.0  Takeaway

  Transaction_Date Quantity_Binned Price_Binned  Payment_Method_Credit Car
d  \
0       2023-09-08             Low        Cheap                       Tru
e
1       2023-05-16             Low     Moderate                      Fals
e
3       2023-04-27             Low    Expensive                      Fals
e
4       2023-06-11             Low        Cheap                      Fals
e
6       2023-10-06             Low     Moderate                      Fals
e

   Payment_Method_Digital Wallet  Payment_Method_ERROR  Payment_Method_UNK
NOWN
0                          False                 False                   F
alse
1                          False                 False                   F
alse
3                          False                 False
True
4                           True                 False                   F
alse
6                          False                  True                   F
alse
```

In [290…

```
if 'Location' in data_missing.columns:
    data_missing['Location_Label'] = le.fit_transform(data_missing['Locat
    print("\nLabel Encoding on Location:")
    print(data_missing[['Location', 'Location_Label']].head())
```

```
Label Encoding on Location:
   Location  Location_Label
0  Takeaway               2
1  In-store               1
3   UNKNOWN               3
4  In-store               1
6  Takeaway               2
```

In [ ]: