In [1]:
```
import pandas as pd
import sqlite3
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
#Need to create a connection to sqlite3
#Read sqlite tables with Pandas as a dataframe
con = sqlite3.connect("/Users/josezavala/Mock_Restaurants_Sales.db")
df = pd.read_sql_query("SELECT * FROM BWL_2018_Sales_Data", con)
```

In [3]:
```
print(df)
```

```
    Year  Period Name  Transactions  Price  Consumer_spend
0   2018       1  BWL          8934    6.8         60751.2
1   2018       2  BWL          7998    6.8         54386.4
2   2018       3  BWL          7568    6.8         51462.4
3   2018       4  BWL          6879    6.8         46777.2
4   2018       5  BWL          7821    6.8         53182.8
5   2018       6  BWL          8324    6.8         56603.2
6   2018       7  BWL          8478    6.8         57650.4
7   2018       8  BWL          8814    6.8         59935.2
8   2018       9  BWL          8631    6.8         58690.8
9   2018      10  BWL          7998    6.8         54386.4
10  2018      11  BWL          8213    6.8         55848.4
11  2018      12  BWL          9032    6.8         61417.6
12  2018      13  BWL          9213    6.8         62648.4
```

In [4]:
```
df_2018_consumer_spend = df['Consumer_spend'].mean()
print(df_2018_consumer_spend)
```

```
56441.56923076923
```

In [5]:
```
pd.read_sql_query('SELECT * FROM Items', con)
```

Out[5]:

	Abbreviation	Name	Price
0	BWL	Bowl	6.80
1	BWL_PR	Bowl_Premium	8.05
2	PL	Plate	8.30
3	PL_PR	Plate_Premium	9.55

In [6]:
```
pd.read_sql_query('SELECT * FROM BWL_2018_Sales_Data UNION SELECT * FROM BW
```

Out[6]:

	Year	Period	Name	Transactions	Price	Consumer_spend
0	2018	1	BWL	8934	6.8	60751.2
1	2018	2	BWL	7998	6.8	54386.4
2	2018	3	BWL	7568	6.8	51462.4
3	2018	4	BWL	6879	6.8	46777.2
4	2018	5	BWL	7821	6.8	53182.8
5	2018	6	BWL	8324	6.8	56603.2
6	2018	7	BWL	8478	6.8	57650.4
7	2018	8	BWL	8814	6.8	59935.2
8	2018	9	BWL	8631	6.8	58690.8
9	2018	10	BWL	7998	6.8	54386.4
10	2018	11	BWL	8213	6.8	55848.4
11	2018	12	BWL	9032	6.8	61417.6
12	2018	13	BWL	9213	6.8	62648.4
13	2019	1	BWL	9724	6.8	66123.2
14	2019	2	BWL	8132	6.8	55297.6
15	2019	3	BWL	7981	6.8	54270.8
16	2019	4	BWL	6894	6.8	46879.2
17	2019	5	BWL	8345	6.8	56746.0
18	2019	6	BWL	8678	6.8	59010.4
19	2019	7	BWL	8245	6.8	56066.0
20	2019	8	BWL	9199	6.8	62553.2
21	2019	9	BWL	8832	6.8	60057.6
22	2019	10	BWL	8375	6.8	56950.0
23	2019	11	BWL	8627	6.8	58663.6
24	2019	12	BWL	9450	6.8	64260.0
25	2019	13	BWL	9678	6.8	65810.4

In [3]:
```
MR_Sales = pd.read_sql_query('SELECT * FROM BWL_2018_Sales_Data UNION SELEC
print(MR_Sales)
```

```
     Year  Period   Name  Transactions  Price  Consumer_spend
0    2018       1    BWL          8934   6.80        60751.20
1    2018       2    BWL          7998   6.80        54386.40
2    2018       3    BWL          7568   6.80        51462.40
3    2018       4    BWL          6879   6.80        46777.20
4    2018       5    BWL          7821   6.80        53182.80
..    ...     ...    ...           ...    ...             ...
99   2019       9  PL_PR          9674   9.55        92386.70
100  2019      10  PL_PR          9879   9.55        94344.45
101  2019      11  PL_PR          9904   9.55        94583.20
102  2019      12  PL_PR          9964   9.55        95156.20
103  2019      13  PL_PR         10245   9.55        97839.75

[104 rows x 6 columns]
```

In [4]:
```
#I just finished the Pandas course from Kaggle and I will be doing the exer
```

In [8]:
```
#Select the 'Consumer_spend' column from 'MR_Sales' and assign to variable
revenue = MR_Sales['Consumer_spend']
print(revenue)
type(revenue)
```

```
0       60751.20
1       54386.40
2       51462.40
3       46777.20
4       53182.80
          ...
99      92386.70
100     94344.45
101     94583.20
102     95156.20
103     97839.75
Name: Consumer_spend, Length: 104, dtype: float64
```

Out[8]:
```
pandas.core.series.Series
```

In [10]:
```
#Select the first value from the 'Consumer_spend' column of `MR_Sales`, ass
first_revenue = MR_Sales['Consumer_spend'][0]
print(first_revenue)
```

```
60751.2
```

In [13]:
```
#Select the first row of data (the first record) from 'MR_Sales', assigning
first_row = MR_Sales.iloc[0]
print(first_row)
```

```
Year                 2018
Period                  1
Name                  BWL
Transactions         8934
Price                 6.8
Consumer_spend    60751.2
Name: 0, dtype: object
```

In [17]:
```
#Select the first 10 values from the 'Consumer_spend' column in 'MR_Sales'
MR_Sales.Consumer_spend.iloc[:10]
```

Out[17]:
```
0    60751.2
1    54386.4
2    51462.4
3    46777.2
4    53182.8
5    56603.2
6    57650.4
7    59935.2
8    58690.8
9    54386.4
Name: Consumer_spend, dtype: float64
```

In [19]:
```
#Select the records with index labels 1, 2, 3, 5, and 8
#iloc is only numeric index and columns
MR_Sales.iloc[[1, 2, 3, 5, 8], :]
```

Out[19]:

	Year	Period	Name	Transactions	Price	Consumer_spend
1	2018	2	BWL	7998	6.8	54386.4
2	2018	3	BWL	7568	6.8	51462.4
3	2018	4	BWL	6879	6.8	46777.2
5	2018	6	BWL	8324	6.8	56603.2
8	2018	9	BWL	8631	6.8	58690.8

In [26]:
```
#Select the Year, Name, Transactions, and Consumer_spend columns of the reco
#loc is for numeric and string
MR_Sales.loc[[0,1,10,100], ['Year', 'Name', 'Transactions', 'Consumer_spend'
```

Out[26]:

	Year	Name	Transactions	Consumer_spend
0	2018	BWL	8934	60751.20
1	2018	BWL	7998	54386.40
10	2018	BWL	8213	55848.40
100	2019	PL_PR	9879	94344.45

In [30]:
```
#Select the Name and Transactions columns of the first 100 records
MR_Sales.loc[:99, ['Name', 'Transactions']]
```

Out[30]:

	Name	Transactions
0	BWL	8934
1	BWL	7998
2	BWL	7568
3	BWL	6879
4	BWL	7821
...	...	...
95	PL_PR	8890
96	PL_PR	8735
97	PL_PR	9342
98	PL_PR	9579
99	PL_PR	9674

100 rows × 2 columns

In [61]:
```
#Select sales for 'Bowls'
MR_Sales.loc[MR_Sales.Name == 'BWL']
```

Out[61]:

	Year	Period	Name	Transactions	Price	Consumer_spend
0	2018	1	BWL	8934	6.8	60751.2
1	2018	2	BWL	7998	6.8	54386.4
2	2018	3	BWL	7568	6.8	51462.4
3	2018	4	BWL	6879	6.8	46777.2
4	2018	5	BWL	7821	6.8	53182.8
5	2018	6	BWL	8324	6.8	56603.2
6	2018	7	BWL	8478	6.8	57650.4
7	2018	8	BWL	8814	6.8	59935.2
8	2018	9	BWL	8631	6.8	58690.8
9	2018	10	BWL	7998	6.8	54386.4
10	2018	11	BWL	8213	6.8	55848.4
11	2018	12	BWL	9032	6.8	61417.6
12	2018	13	BWL	9213	6.8	62648.4
13	2019	1	BWL	9724	6.8	66123.2
14	2019	2	BWL	8132	6.8	55297.6
15	2019	3	BWL	7981	6.8	54270.8
16	2019	4	BWL	6894	6.8	46879.2
17	2019	5	BWL	8345	6.8	56746.0
18	2019	6	BWL	8678	6.8	59010.4
19	2019	7	BWL	8245	6.8	56066.0
20	2019	8	BWL	9199	6.8	62553.2
21	2019	9	BWL	8832	6.8	60057.6
22	2019	10	BWL	8375	6.8	56950.0
23	2019	11	BWL	8627	6.8	58663.6
24	2019	12	BWL	9450	6.8	64260.0
25	2019	13	BWL	9678	6.8	65810.4

In [62]:
```
#Select sales for 'Bwl' and 'BWL_PR' from the Year 2018 with Transaction ov
MR_Sales.loc[MR_Sales.Name.isin(['BWL', 'BWL_PR']) & (MR_Sales.Year == 2018
```

Out[62]:

	Year	Period	Name	Transactions	Price	Consumer_spend
0	2018	1	BWL	8934	6.80	60751.20
1	2018	2	BWL	7998	6.80	54386.40
2	2018	3	BWL	7568	6.80	51462.40
4	2018	5	BWL	7821	6.80	53182.80
5	2018	6	BWL	8324	6.80	56603.20
6	2018	7	BWL	8478	6.80	57650.40
7	2018	8	BWL	8814	6.80	59935.20
8	2018	9	BWL	8631	6.80	58690.80
9	2018	10	BWL	7998	6.80	54386.40
10	2018	11	BWL	8213	6.80	55848.40
11	2018	12	BWL	9032	6.80	61417.60
12	2018	13	BWL	9213	6.80	62648.40
31	2018	6	BWL_PR	7103	8.05	57179.15
32	2018	7	BWL_PR	7293	8.05	58708.65
33	2018	8	BWL_PR	7673	8.05	61767.65
34	2018	9	BWL_PR	7132	8.05	57412.60
36	2018	11	BWL_PR	7434	8.05	59842.70
37	2018	12	BWL_PR	7521	8.05	60544.05
38	2018	13	BWL_PR	7741	8.05	62315.05

In [60]:
```
#Select sales from PL_PR with Transactions above 10,000
MR_Sales.loc[MR_Sales.Name.isin(['PL_PR']) & (MR_Sales.Transactions >= 1000
```

Out[60]:

	Year	Period	Name	Transactions	Price	Consumer_spend
90	2018	13	PL_PR	10032	9.55	95805.60
91	2019	1	PL_PR	10126	9.55	96703.30
103	2019	13	PL_PR	10245	9.55	97839.75

In [70]:
```
#Compare sales for 'BWL' from 2018 to 2019 with transactions over 8,000
MR_Sales.loc[MR_Sales.Name.isin(['BWL']) & (MR_Sales.Transactions >= 8000)]
```

Out[70]:

	Year	Period	Name	Transactions	Price	Consumer_spend
0	2018	1	BWL	8934	6.8	60751.2
5	2018	6	BWL	8324	6.8	56603.2
6	2018	7	BWL	8478	6.8	57650.4
7	2018	8	BWL	8814	6.8	59935.2
8	2018	9	BWL	8631	6.8	58690.8
10	2018	11	BWL	8213	6.8	55848.4
11	2018	12	BWL	9032	6.8	61417.6
12	2018	13	BWL	9213	6.8	62648.4
13	2019	1	BWL	9724	6.8	66123.2
14	2019	2	BWL	8132	6.8	55297.6
17	2019	5	BWL	8345	6.8	56746.0
18	2019	6	BWL	8678	6.8	59010.4
19	2019	7	BWL	8245	6.8	56066.0
20	2019	8	BWL	9199	6.8	62553.2
21	2019	9	BWL	8832	6.8	60057.6
22	2019	10	BWL	8375	6.8	56950.0
23	2019	11	BWL	8627	6.8	58663.6
24	2019	12	BWL	9450	6.8	64260.0
25	2019	13	BWL	9678	6.8	65810.4

In [4]:
```
#What is the median of the Consumer_spend column
MR_Sales.Consumer_spend.median()
```

Out[4]:
```
72703.65
```

In [5]:
```
#What items are represented in the dataset?
MR_Sales.Name.unique()
```

Out[5]:
```
array(['BWL', 'BWL_PR', 'PL', 'PL_PR'], dtype=object)
```

In [7]:
```
#How often does each item appear in the dataset?
MR_Sales.Name.value_counts()
```

Out[7]:
```
PL_PR     26
BWL       26
BWL_PR    26
PL        26
Name: Name, dtype: int64
```

In [10]:
```
#Create variable centered_price containing a version of the Consumer_spend
centered_consumer_spend = MR_Sales.Consumer_spend - MR_Sales.Consumer_spend
print(centered_consumer_spend)
```

```
0     -15499.432692
1     -21864.232692
2     -24788.232692
3     -29473.432692
4     -23067.832692
           ...
99     16136.067308
100    18093.817308
101    18332.567308
102    18905.567308
103    21589.117308
Name: Consumer_spend, Length: 104, dtype: float64
```

In [17]:
```
#What is the most expensive/inexpensive item
expensive_item = MR_Sales.loc[MR_Sales.Price.idxmax(), 'Name']
inexpensive_item = MR_Sales.loc[MR_Sales.Price.idxmin(), 'Name']
print(expensive_item)
print(inexpensive_item)
```

```
PL_PR
BWL
```

In [ ]: