Coffee Shop Sales Analysis *

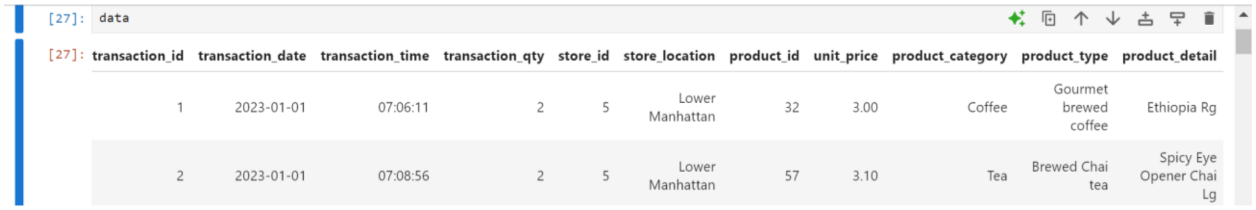
Introduction

Import Pandas and dataset: First, we import Pandas library into Python (Jupyter Notebook) and our dataset, using the code:

```
import pandas as pd

data = pd.read_excel('Coffee Shop Sales.xlsx')
```

Once both Pandas and the dataset are imported into Jupyter, we obtain a visualization of the dataset, which consists of 14,9116 rows and 11 columns:

```
data
```



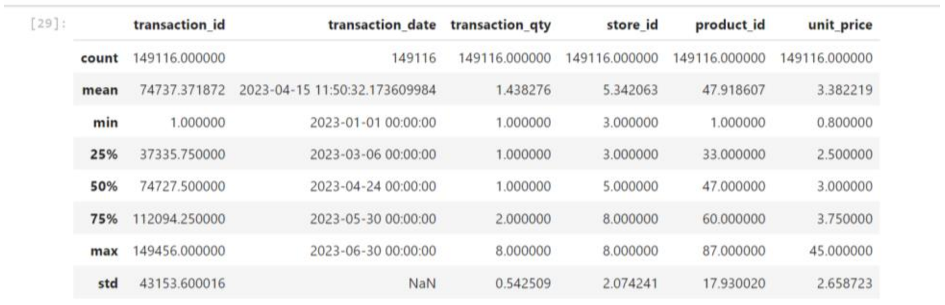
```
[27]: data
```

[27]:

transaction_id	transaction_date	transaction_time	transaction_qty	store_id	store_location	product_id	unit_price	product_category	product_type	product_detail
1	2023-01-01	07:06:11	2	5	Lower Manhattan	32	3.00	Coffee	Gourmet brewed coffee	Ethiopia Rg
2	2023-01-01	07:08:56	2	5	Lower Manhattan	57	3.10	Tea	Brewed Chai tea	Spicy Eye Opener Chai Lg

Our next step is to describe the dataset, with the following code:

```
data.describe()
```

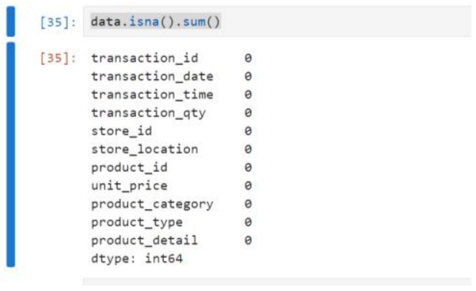


[29]:

	transaction_id	transaction_date	transaction_qty	store_id	product_id	unit_price
count	149116.000000	149116	149116.000000	149116.000000	149116.000000	149116.000000
mean	74737.371872	2023-04-15 11:50:32.173609984	1.438276	5.342063	47.918607	3.382219
min	1.000000	2023-01-01 00:00:00	1.000000	3.000000	1.000000	0.800000
25%	37335.750000	2023-03-06 00:00:00	1.000000	3.000000	33.000000	2.500000
50%	74727.500000	2023-04-24 00:00:00	1.000000	5.000000	47.000000	3.000000
75%	112094.250000	2023-05-30 00:00:00	2.000000	8.000000	60.000000	3.750000
max	149456.000000	2023-06-30 00:00:00	8.000000	8.000000	87.000000	45.000000
std	43153.600016	NaN	0.542509	2.074241	17.930020	2.658723

Next, we check for null values in our dataset, with the code:

data.isna().sum()



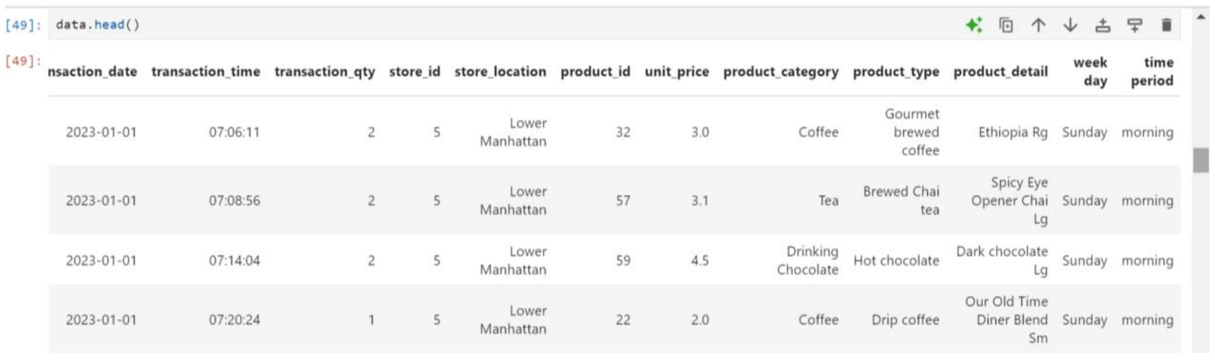
```
[35]: data.isna().sum()

[35]: transaction_id      0
      transaction_date    0
      transaction_time    0
      transaction_qty     0
      store_id            0
      store_location      0
      product_id          0
      unit_price          0
      product_category    0
      product_type        0
      product_detail      0
      dtype: int64
```

As we can see, there are no null values in our data.

Next, we add two (2) new columns, named ‘week day’ and ‘time period’, with the following codes:

data['week day'] = data['transaction_date'].dt.strftime('%A')

data['time period'] = data['transaction_time'].apply(lambda x : 'morning' if 6 <= x.hour < 12 else('afternoon' if 12 <= x.hour < 18 else 'evening'))



```
[49]: data.head()
```

[49]:

nsaction_date	transaction_time	transaction_qty	store_id	store_location	product_id	unit_price	product_category	product_type	product_detail	week day	time period
2023-01-01	07:06:11	2	5	Lower Manhattan	32	3.0	Coffee	Gourmet brewed coffee	Ethiopia Rg	Sunday	morning
2023-01-01	07:08:56	2	5	Lower Manhattan	57	3.1	Tea	Brewed Chai tea	Spicy Eye Opener Chai Lg	Sunday	morning
2023-01-01	07:14:04	2	5	Lower Manhattan	59	4.5	Drinking Chocolate	Hot chocolate	Dark chocolate Lg	Sunday	morning
2023-01-01	07:20:24	1	5	Lower Manhattan	22	2.0	Coffee	Drip coffee	Our Old Time Diner Blend Sm	Sunday	morning

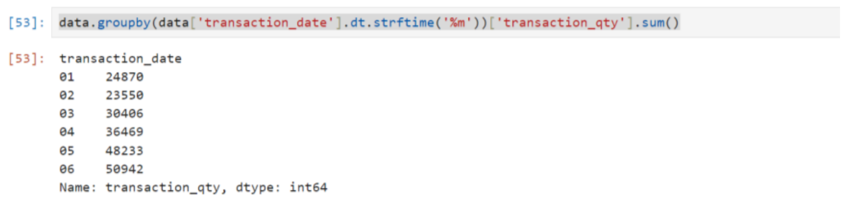
We will use these two new columns in our analysis.

Objectives/Hypotheses

1. Which month has the highest sales?

   We use the following code in order to obtain the month with the highest sales:

   data.groupby(data['transaction_date'].dt.strftime('%m'))['transaction_qty'].sum()



```
[53]: data.groupby(data['transaction_date'].dt.strftime('%m'))['transaction_qty'].sum()

[53]: transaction_date
      01    24870
      02    23550
      03    30406
      04    36469
      05    48233
      06    50942
      Name: transaction_qty, dtype: int64
```

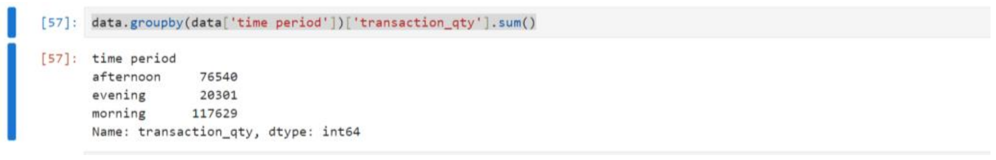
   As we can see in the results, the sixth month (and the last one included in the dataset, June) was the one with the highest sales, reaching 50,942 transactions.

2. Which time of the day are sales the most active?

   To answer this question, we use the following code:

   data.groupby(data['time period'])['transaction_qty'].sum()

   And we see that the most active time of the day is the 'morning', with 117,629 transactions.



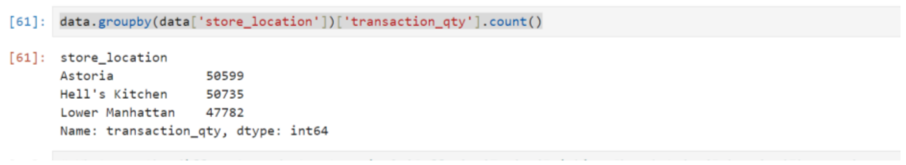
```
[57]: data.groupby(data['time period'])['transaction_qty'].sum()

[57]: time period
      afternoon     76540
      evening       20301
      morning      117629
      Name: transaction_qty, dtype: int64
```

3. Which store location has the highest number of sales?

   We use the next code in order to see which store has the highest number of sales:

   data.groupby(data['store_location'])['transaction_qty'].count()

   And the result is that the 'Hell's Kitchen' store has the highest number of sales, reaching a total of 50,735 transactions in the time analyzed.

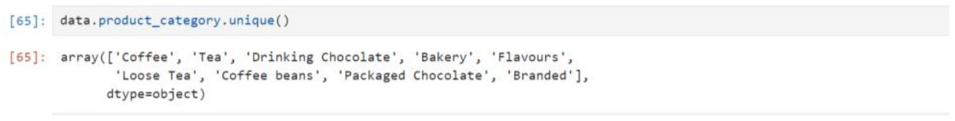


```
[61]: data.groupby(data['store_location'])['transaction_qty'].count()

[61]: store_location
      Astoria            50599
      Hell's Kitchen     50735
      Lower Manhattan    47782
      Name: transaction_qty, dtype: int64
```

4. What are the different product categories?

   In order to obtain an array of the product categories sold in the coffee store, we write the following code:

   data.product_category.unique()



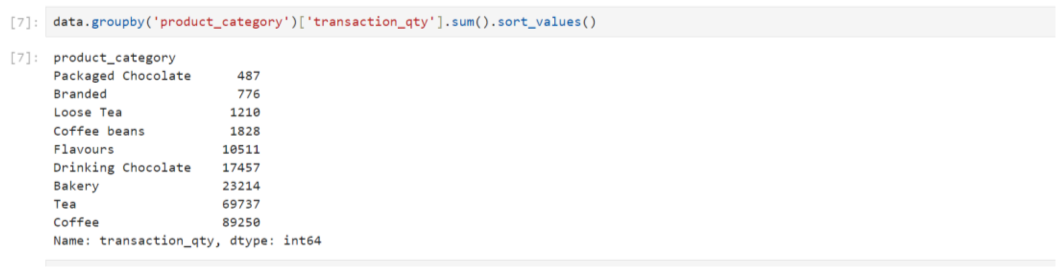
```
[65]: data.product_category.unique()

[65]: array(['Coffee', 'Tea', 'Drinking Chocolate', 'Bakery', 'Flavours',
             'Loose Tea', 'Coffee beans', 'Packaged Chocolate', 'Branded'],
            dtype=object)
```

   And we obtain that the product categories are: 'Coffee', 'Tea', 'Drinking Chocolate', 'Bakery', 'Flavours', 'Loose Tea', 'Coffee beans', 'Packaged Chocolate' and 'Branded'.

5. Which product category has the highest number of sales?

   We wish to see which product category is the most sold in the store, and for that we use the code:

   data.groupby('product_category')['transaction_qty'].sum().sort_values()



```
[7]: data.groupby('product_category')['transaction_qty'].sum().sort_values()

[7]: product_category
     Packaged Chocolate      487
     Branded                 776
     Loose Tea              1210
     Coffee beans           1828
     Flavours              10511
     Drinking Chocolate    17457
     Bakery                23214
     Tea                   69737
     Coffee                89250
     Name: transaction_qty, dtype: int64
```

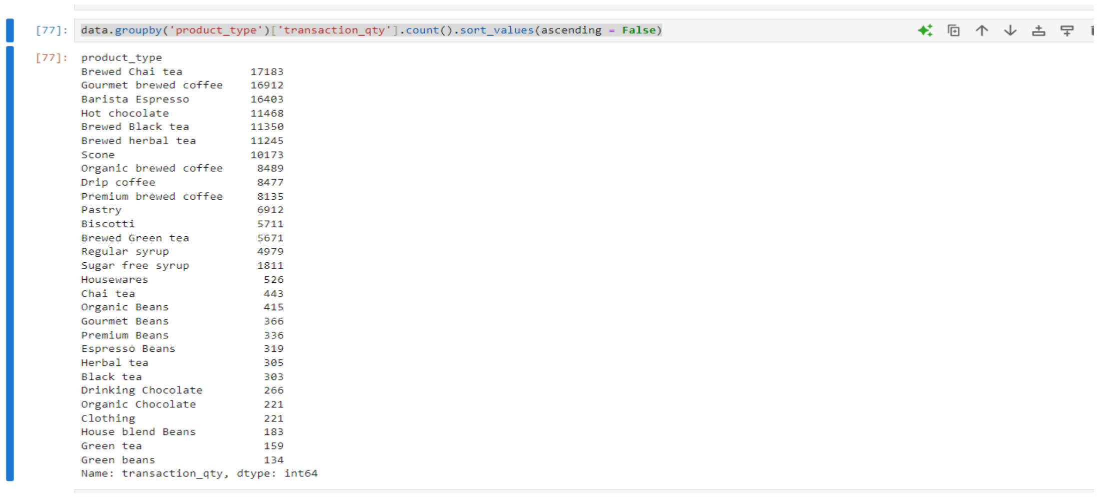
   As we can observe, the category with the highest sales is ‘Coffee’, reaching a total of 89,250 transactions.

6. Which product type has the highest sales?

   We want to see which product type is the most sold, and for that we use the next code:

   data.groupby('product_type')['transaction_qty'].count().sort_values(ascending = False)

   The result we obtain is that the type with the highest sales is the 'Brewed Chai Tea', with 17,183 transactions. We can see the result in the screenshot shown below:

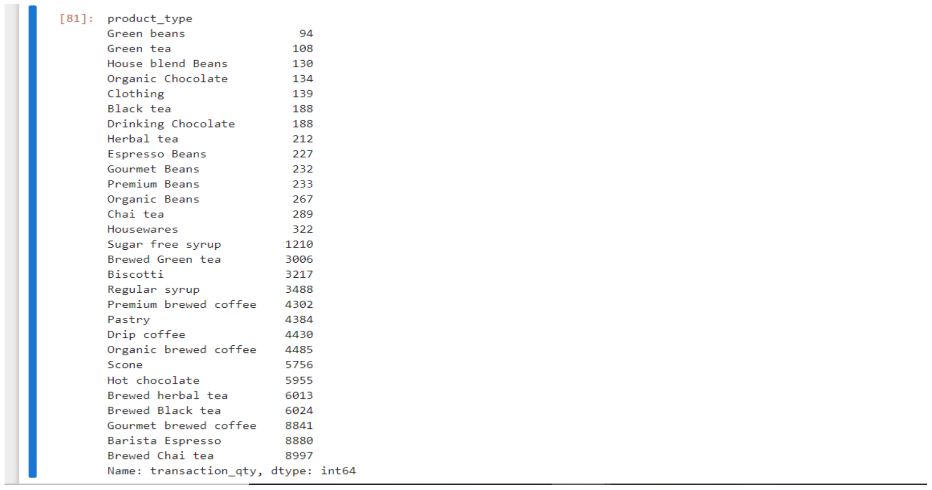


```
[77]: data.groupby('product_type')['transaction_qty'].count().sort_values(ascending = False)

[77]: product_type
      Brewed Chai tea          17183
      Gourmet brewed coffee    16912
      Barista Espresso         16403
      Hot chocolate            11468
      Brewed Black tea         11350
      Brewed herbal tea        11245
      Scone                    10173
      Organic brewed coffee     8489
      Drip coffee               8477
      Premium brewed coffee     8135
      Pastry                    6912
      Biscotti                  5711
      Brewed Green tea          5671
      Regular syrup             4979
      Sugar free syrup          1811
      Housewares                 526
      Chai tea                   443
      Organic Beans              415
      Gourmet Beans              366
      Premium Beans              336
      Espresso Beans             319
      Herbal tea                 305
      Black tea                  303
      Drinking Chocolate         266
      Organic Chocolate          221
      Clothing                   221
      House blend Beans          183
      Green tea                  159
      Green beans                134
      Name: transaction_qty, dtype: int64
```

7. Which product has the most sales in the morning?

   Next, we use the following code in order to see which product is the most sold during the morning time:

   data.loc[data['time period'] == 'morning'].groupby('product_type')['transaction_qty'].count().sort_values()



```
[81]: product_type
      Green beans                94
      Green tea                 108
      House blend Beans         130
      Organic Chocolate         134
      Clothing                  139
      Black tea                 188
      Drinking Chocolate        188
      Herbal tea                212
      Espresso Beans            227
      Gourmet Beans             232
      Premium Beans             233
      Organic Beans             267
      Chai tea                  289
      Housewares                322
      Sugar free syrup         1210
      Brewed Green tea         3006
      Biscotti                 3217
      Regular syrup            3488
      Premium brewed coffee    4302
      Pastry                   4384
      Drip coffee              4430
      Organic brewed coffee    4485
      Scone                    5756
      Hot chocolate            5955
      Brewed herbal tea        6013
      Brewed Black tea         6024
      Gourmet brewed coffee    8841
      Barista Espresso         8880
      Brewed Chai tea          8997
      Name: transaction_qty, dtype: int64
```

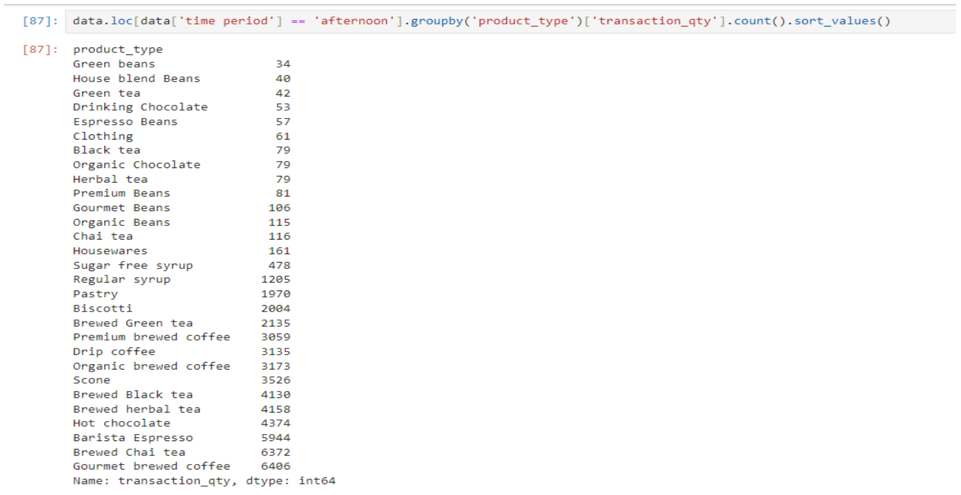
As we see, the most sold product type during the morning is the ‘Brewed Chai Tea’, with 8,997 transactions.

8. Which product is the most active during the afternoon?

   In order to know which product is the most active during the afternoon we use the code:

   data.loc[data['time period'] == 'afternoon'].groupby('product_type')['transaction_qty'].count().sort_values()

   As we can observe, the 'Gourmet brewed coffee' is the most active during the afternoon, with 6,406 transactions.

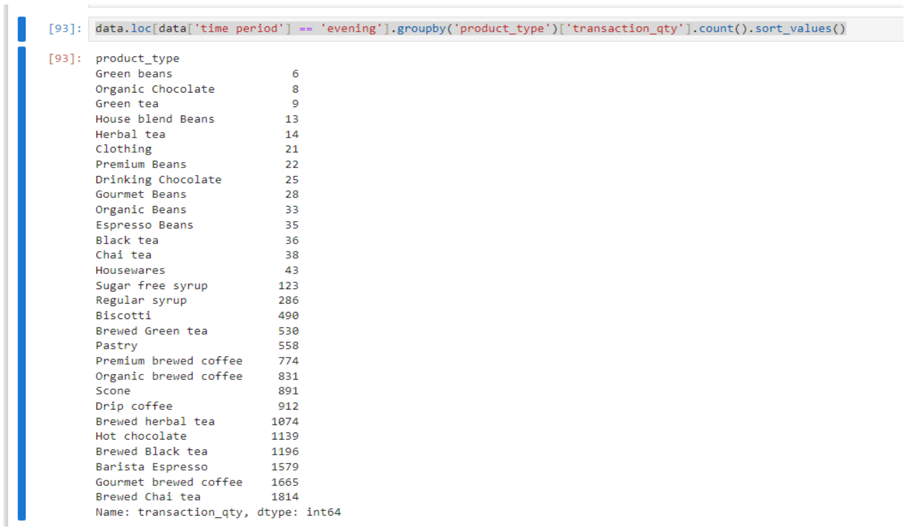


```
[87]: data.loc[data['time period'] == 'afternoon'].groupby('product_type')['transaction_qty'].count().sort_values()

[87]: product_type
      Green beans                34
      House blend Beans          40
      Green tea                  42
      Drinking Chocolate         53
      Espresso Beans             57
      Clothing                   61
      Black tea                  79
      Organic Chocolate          79
      Herbal tea                 79
      Premium Beans              81
      Gourmet Beans             106
      Organic Beans             115
      Chai tea                  116
      Housewares                161
      Sugar free syrup          478
      Regular syrup            1205
      Pastry                   1970
      Biscotti                 2004
      Brewed Green tea         2135
      Premium brewed coffee    3059
      Drip coffee              3135
      Organic brewed coffee    3173
      Scone                    3526
      Brewed Black tea         4130
      Brewed herbal tea        4158
      Hot chocolate            4374
      Barista Espresso         5944
      Brewed Chai tea          6372
      Gourmet brewed coffee    6406
      Name: transaction_qty, dtype: int64
```

9. Which product is the most active in the evening?

The product most active in the evening is the ‘Brewed Chai tea’, reaching a total of 1,814 transactions. In order to obtain this result, we use the following code:

data.loc[data['time period'] == 'evening'].groupby('product_type')['transaction_qty'].count().sort_values()

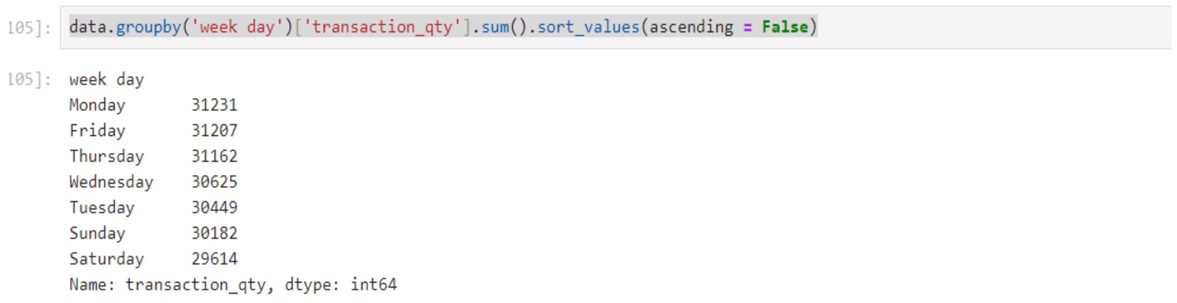


```
[93]: data.loc[data['time period'] == 'evening'].groupby('product_type')['transaction_qty'].count().sort_values()

[93]: product_type
      Green beans                 6
      Organic Chocolate           8
      Green tea                   9
      House blend Beans          13
      Herbal tea                 14
      Clothing                   21
      Premium Beans              22
      Drinking Chocolate         25
      Gourmet Beans              28
      Organic Beans              33
      Espresso Beans             35
      Black tea                  36
      Chai tea                   38
      Housewares                 43
      Sugar free syrup          123
      Regular syrup             286
      Biscotti                  490
      Brewed Green tea          530
      Pastry                    558
      Premium brewed coffee     774
      Organic brewed coffee     831
      Scone                     891
      Drip coffee               912
      Brewed herbal tea        1074
      Hot chocolate            1139
      Brewed Black tea         1196
      Barista Espresso         1579
      Gourmet brewed coffee    1665
      Brewed Chai tea          1814
      Name: transaction_qty, dtype: int64
```

10. Which day of the week has the highest number of sales?

Monday is the day of the week with the highest number of sales, reaching 31,231 transactions. We write the code below in order to obtain the result described:

data.groupby('week day')['transaction_qty'].sum().sort_values(ascending = False)



```
105]: data.groupby('week day')['transaction_qty'].sum().sort_values(ascending = False)

105]: week day
      Monday       31231
      Friday       31207
      Thursday     31162
      Wednesday    30625
      Tuesday      30449
      Sunday       30182
      Saturday     29614
      Name: transaction_qty, dtype: int64
```

Charts

In this section, we show charts describing the results we obtained previously.

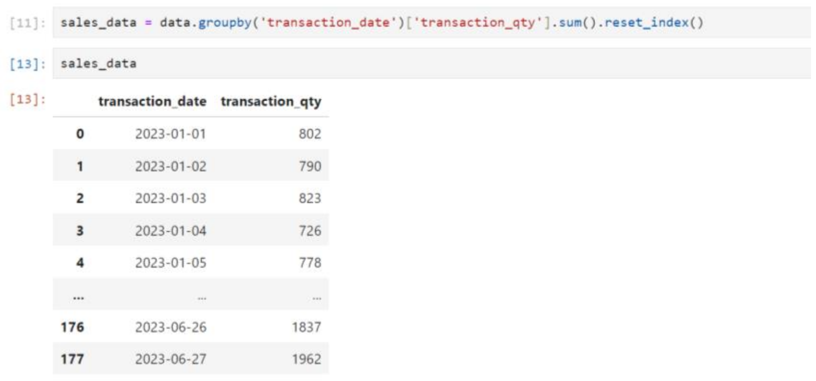
Before we plot our chart, we import matplotlib and seaborn libraries into Jupyter, and make a few modifications to the data, in order to get readable and easy-to-understand information.

First, we modify ‘transaction date’ into ‘date time’ with the following code:

```
data['transaction_date'] = pd.to_datetime(data['transaction_date'])
```

Next, we group by date and sum sales, by writing the code:

```
sales_data = data.groupby('transaction_date')['transaction_qty'].sum().reset_index()
```



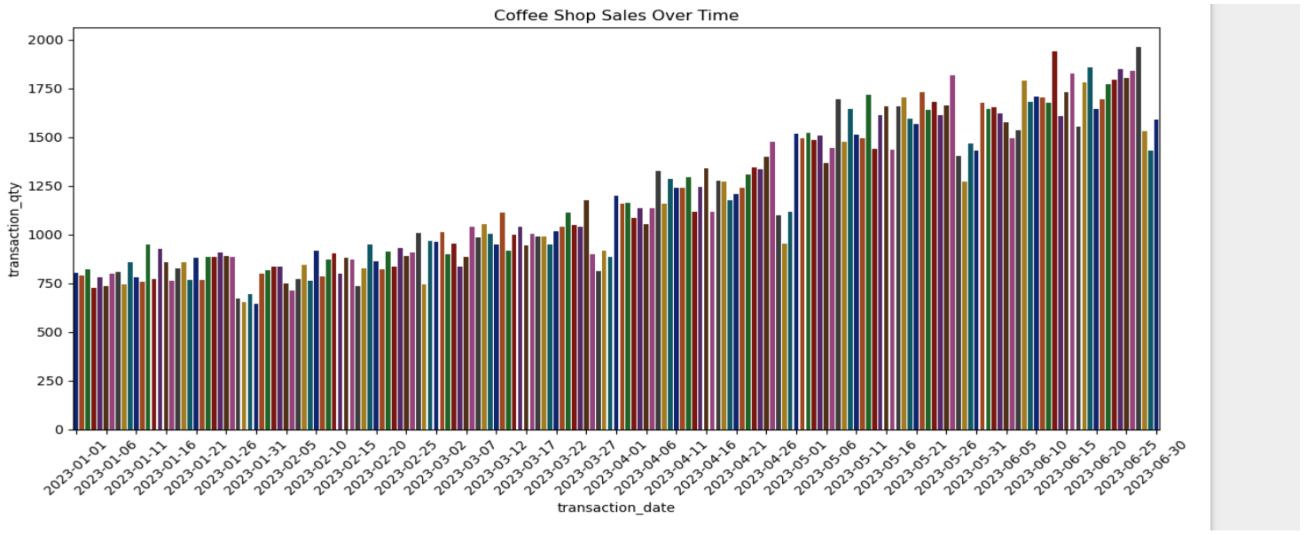
```
[11]: sales_data = data.groupby('transaction_date')['transaction_qty'].sum().reset_index()

[13]: sales_data

[13]:
```

	transaction_date	transaction_qty
0	2023-01-01	802
1	2023-01-02	790
2	2023-01-03	823
3	2023-01-04	726
4	2023-01-05	778
...	...	...
176	2023-06-26	1837
177	2023-06-27	1962

<u>Bar chart</u>:

We create a chart by plotting 'transaction_date' in the x axis, and 'transaction_qty' in the y axis, to show what we call "Coffee Shop Sales Over Time", representing all the sales that took place in the first 6 months of the year 2023. Next, we plot our bat chart showing the sales over the time analyzed. For this task, we use the following code:

```
plt.figure(figsize=(12, 6))

sns.barplot(x='transaction_date', y='transaction_qty', data=sales_data, palette='dark')

plt.title('Coffee Shop Sales Over Time')

plt.xlabel('transaction_date')

plt.ylabel('transaction_qty')

plt.xticks(ticks=sales_data.index[::5],
labels=sales_data['transaction_date'].dt.strftime('%Y-%m-%d')[::5], rotation=45)

plt.tight_layout()

plt.show()
```



We proceed to build another bar chart showing us the sales through each day of the week, and we will be able to observe which day is the most active in sales. For this, we use the column created in the beginning of our analysis (‘week day’), we group sales by day of the week, and we plot the results. We write the following codes:

- First, we group by 'days of the week' and sum the transaction quantities:

```
sales_per_weekday = data.groupby('week day')['transaction_qty'].sum().reset_index()
```

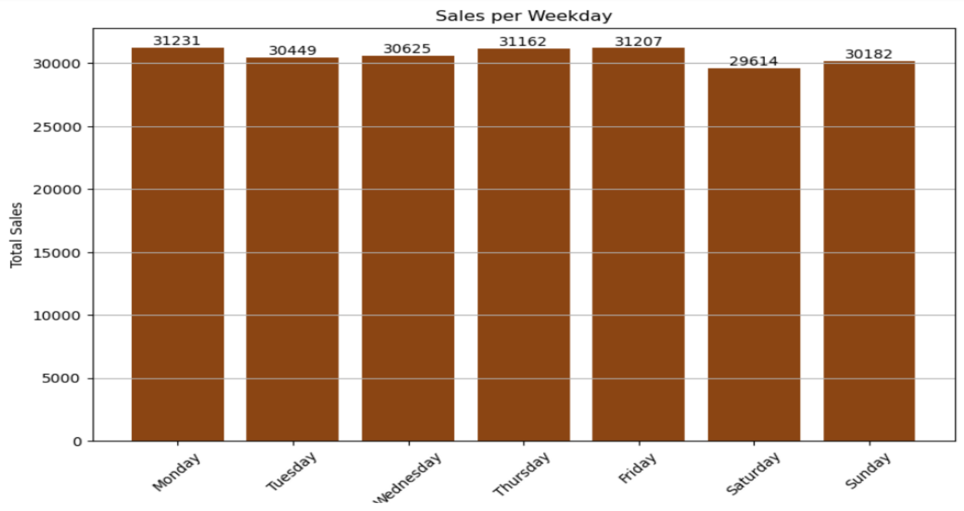
- Next, we reorder 'week day' for proper visualization:

```
days_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']

sales_per_weekday['week day'] = pd.Categorical(sales_per_weekday['week day'], categories=days_order, ordered=True)

sales_per_weekday = sales_per_weekday.sort_values('week day')
```

- Then, we plot 'Total Sales by Day of the week':

```
plt.figure(figsize=(10, 6))

bars = plt.bar(sales_per_weekday['week day'], sales_per_weekday['transaction_qty'], color = 'saddlebrown')

 for bar in bars:

   yval = bar.get_height()

   plt.text(bar.get_x() + bar.get_width()/2, yval, int(yval), ha='center', va='bottom')

 plt.title('Sales per Weekday')

plt.xlabel('Weekday')

plt.ylabel('Total Sales')

plt.xticks(rotation=45)

plt.grid(axis='y')
plt.show()
```

And we obtain the next bar chart, showing that the day with the most sales is Monday (31,231 transactions).



Next, and last, we want to show which product type and which time of the day are the most active in sales. We follow the next steps:

- First, we aggregate transaction quantities by time period:

```
sales_by_time = data.groupby('time period')['transaction_qty'].sum()
```

- Next, we count transactions by product type for each time period:

```
morning_counts = data.loc[data['time period'] == 'morning'].groupby('product_type')['transaction_qty'].count()

afternoon_counts = data.loc[data['time period'] == 'afternoon'].groupby('product_type')['transaction_qty'].count()

evening_counts = data.loc[data['time period'] == 'evening'].groupby('product_type')['transaction_qty'].count()
```

- We combine counts into a DataFrame for easy plotting:

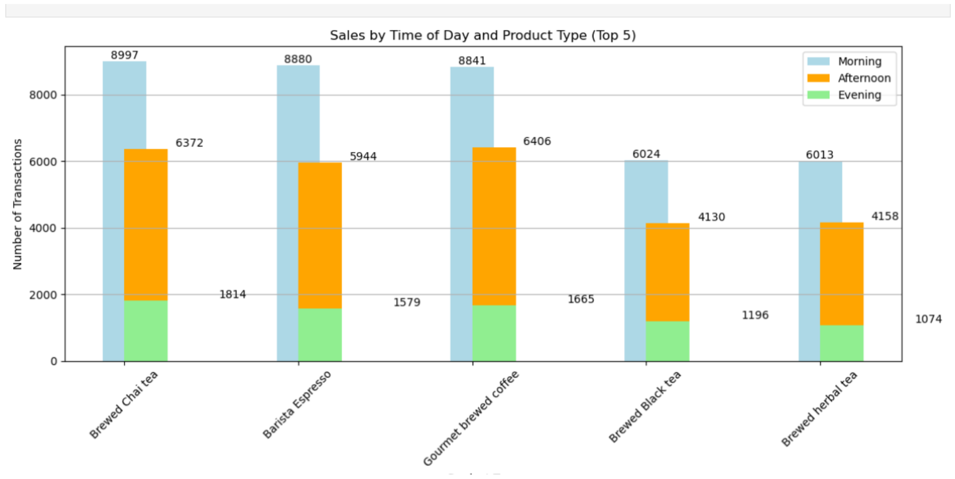
```
counts_df = pd.DataFrame({
    'Morning': morning_counts,
    'Afternoon': afternoon_counts,
    'Evening': evening_counts
}).fillna(0)
```

- #We want to show the top 5 product type sold buy time period, prioritizing ‘Morning’ since we already know that is the most active time of the day in sales:

```
top_n = 5
counts_df = counts_df.nlargest(top_n, 'Morning')
```

- Plotting our data:

```
plt.figure(figsize=(12, 6))
bars_morning = plt.bar(counts_df.index, counts_df['Morning'], label='Morning', color='lightblue', width=0.25, align='center')
bars_afternoon = plt.bar(counts_df.index, counts_df['Afternoon'], label='Afternoon', color='orange', width=0.25, align='edge' )
bars_evening = plt.bar(counts_df.index, counts_df['Evening'], label='Evening', color='lightgreen', width=0.25, align='edge')


for bar in bars_morning:
    yval=bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, int(yval), ha='center', va='bottom')


for bar in bars_afternoon:
    yval= bar.get_height()

    plt.text(bar.get_x() + bar.get_width()/2 + 0.25, yval, int(yval), ha='center', va='bottom')

for bar in bars_evening:

    yval = bar.get_height()

    plt.text(bar.get_x() + bar.get_width()/2 + 0.5, yval, int(yval), ha='center', va='bottom')


plt.title('Sales by Time of Day and Product Type (Top 5)')

plt.xlabel('Product Type')

plt.ylabel('Number of Transactions')

plt.xticks(rotation=45)

plt.legend()

plt.grid(axis='y')

plt.tight_layout()

plt.show()
```



As we can see, the ‘Morning’ time and the ‘Brewed Chai Tea’ type are the most active in sales, with 117,629 transactions and 17,183 transactions, respectively.

<u>Pie Chart:</u>

Now, we want to show ‘Sales distribution by Product Category’ and observe which category is the most sold. For this, we will utilize a pie chart, and write the following codes:

- First, we group and sum transaction quantities by product category:

```
category_sales = data.groupby('product_category')['transaction_qty'].sum().sort_values()
```

- Next, we define which categories to show:

```
top_categories = category_sales.nlargest(5)

other_sales = category_sales.sum() - top_categories.sum()
```

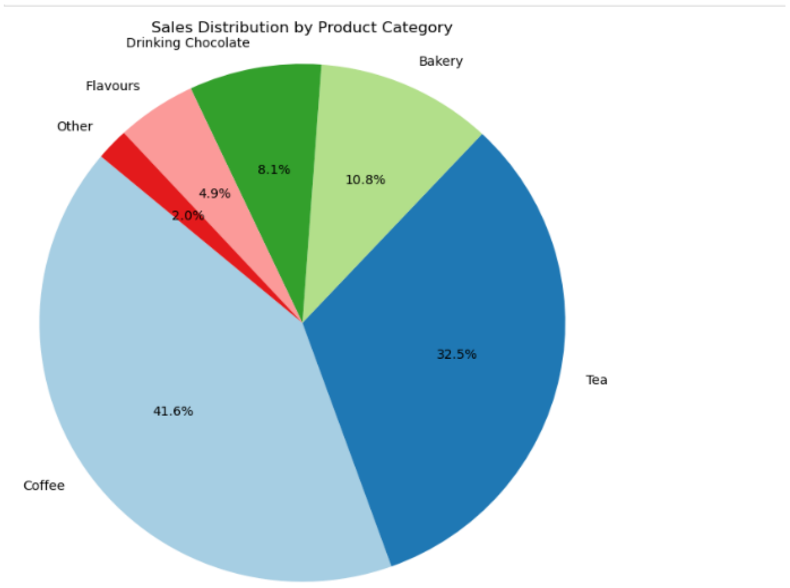
- Then, we create a new series including only top categories and "Other":

```
labels = top_categories.index.tolist() + ['Other']

sizes = top_categories.tolist() + [other_sales]
```

- Plotting our pie chart:

```
plt.figure(figsize=(8, 8))

plt.pie(sizes, labels=labels, autopct = '%1.1f%%',

    startangle=140, colors=plt.cm.Paired.colors)

plt.title('Sales Distribution by Product Category')

plt.axis('equal')

plt.show()
```

And we get:



Showing us, that ‘Coffee’ is the most sold product category, representing 41.6% of total sales.

Conclusions

Our analysis of coffee sales gave us a number of insights about how this business performs in sales.

We will summarize the most important results of our analysis, in order to obtain clear and understandable information that can be used to make well informed decisions.

- The sixth **month** (and the last one included in the dataset, *June*) was the one **with the highest sales**, reaching 50,942 transactions. -
- The most active **time of the day** is the *'morning'*, with 117,629 transactions. -
- The *'Hell's Kitchen*' **store** has the highest number of sales, reaching a total of 50,735 transactions. -
- The **product category** with the highest sales is '*Coffee*', reaching a total of 89,250 transactions. -
- The **product type** with the highest sales is the '*Brewed Chai Tea*', with 17,183 transactions. -
- The most sold **product type** during the **morning** is the '*Brewed Chai Tea*', with 8,997 transactions. -
- The '*Gourmet brewed coffee*' is the most active **product type** sold during the **afternoon**, with 6,406 transactions. -
- The **product type** most active in the **evening** is the '*Brewed Chai tea*', reaching a total of 1,814 transactions. -
- *Monday* is the **day of the week** with the **highest number of sales**, reaching 31,231 transactions. -

*I would like to thank MERN Stack Dev https://www.youtube.com/@mern_stack_dev/featured for the dataset and the project idea.