

Coffe sales

July 19, 2024

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

```
[3]: df = pd.read_excel('Coffee Shop Sales.xlsx')
```

```
[4]: df
```

```
[4]:      transaction_id transaction_date transaction_time transaction_qty \
0                1      2023-01-01      07:06:11                2
1                2      2023-01-01      07:08:56                2
2                3      2023-01-01      07:14:04                2
3                4      2023-01-01      07:20:24                1
4                5      2023-01-01      07:22:41                2
...          ...          ...          ...          ...
149111          149452      2023-06-30      20:18:41                2
149112          149453      2023-06-30      20:25:10                2
149113          149454      2023-06-30      20:31:34                1
149114          149455      2023-06-30      20:57:19                1
149115          149456      2023-06-30      20:57:19                2

      store_id  store_location  product_id  unit_price  product_category \
0            5  Lower Manhattan        32         3.00             Coffee
1            5  Lower Manhattan        57         3.10              Tea
2            5  Lower Manhattan        59         4.50  Drinking Chocolate
3            5  Lower Manhattan        22         2.00             Coffee
4            5  Lower Manhattan        57         3.10              Tea
...          ...          ...          ...          ...
149111          8  Hell's Kitchen        44         2.50              Tea
149112          8  Hell's Kitchen        49         3.00              Tea
149113          8  Hell's Kitchen        45         3.00              Tea
149114          8  Hell's Kitchen        40         3.75             Coffee
149115          8  Hell's Kitchen        64         0.80      Flavours

      product_type      product_detail
0  Gourmet brewed coffee      Ethiopia Rg
1    Brewed Chai tea  Spicy Eye Opener Chai Lg
2    Hot chocolate      Dark chocolate Lg
```

```

3           Drip coffee  Our Old Time Diner Blend Sm
4       Brewed Chai tea      Spicy Eye Opener Chai Lg
...
149111    Brewed herbal tea                Peppermint Rg
149112    Brewed Black tea      English Breakfast Lg
149113    Brewed herbal tea                Peppermint Lg
149114    Barista Espresso                Cappuccino
149115    Regular syrup                Hazelnut syrup

[149116 rows x 11 columns]

```

```
[6]: missing_values = df.isnull().sum()
      print(missing_values)
```

```

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

```

```
[7]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 149116 entries, 0 to 149115
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   transaction_id         149116 non-null  int64
1   transaction_date       149116 non-null  datetime64[ns]
2   transaction_time       149116 non-null  object
3   transaction_qty        149116 non-null  int64
4   store_id               149116 non-null  int64
5   store_location         149116 non-null  object
6   product_id             149116 non-null  int64
7   unit_price             149116 non-null  float64
8   product_category       149116 non-null  object
9   product_type           149116 non-null  object
10  product_detail         149116 non-null  object
dtypes: datetime64[ns](1), float64(1), int64(4), object(5)
memory usage: 12.5+ MB

```

```
[9]: df['transaction_time'] = df['transaction_time'].astype(str)
df['transaction_time'] = pd.to_timedelta(df['transaction_time'])
```

```
[10]: df['sales'] = df['transaction_qty'] * df['unit_price']
```

```
[11]: df['datetime'] = df['transaction_date'] + df['transaction_time']
df.head()
```

```
[11]:
```

	transaction_id	transaction_date	transaction_time	transaction_qty	\
0	1	2023-01-01	0 days 07:06:11	2	
1	2	2023-01-01	0 days 07:08:56	2	
2	3	2023-01-01	0 days 07:14:04	2	
3	4	2023-01-01	0 days 07:20:24	1	
4	5	2023-01-01	0 days 07:22:41	2	

	store_id	store_location	product_id	unit_price	product_category	\
0	5	Lower Manhattan	32	3.0	Coffee	
1	5	Lower Manhattan	57	3.1	Tea	
2	5	Lower Manhattan	59	4.5	Drinking Chocolate	
3	5	Lower Manhattan	22	2.0	Coffee	
4	5	Lower Manhattan	57	3.1	Tea	

	product_type	product_detail	sales	\
0	Gourmet brewed coffee	Ethiopia Rg	6.0	
1	Brewed Chai tea	Spicy Eye Opener Chai Lg	6.2	
2	Hot chocolate	Dark chocolate Lg	9.0	
3	Drip coffee	Our Old Time Diner Blend Sm	2.0	
4	Brewed Chai tea	Spicy Eye Opener Chai Lg	6.2	

	datetime
0	2023-01-01 07:06:11
1	2023-01-01 07:08:56
2	2023-01-01 07:14:04
3	2023-01-01 07:20:24
4	2023-01-01 07:22:41

```
[12]: df.describe()
```

```
[12]:
```

	transaction_id	transaction_date	\
count	149116.000000	149116	
mean	74737.371872	2023-04-15 11:50:32.173609984	
min	1.000000	2023-01-01 00:00:00	
25%	37335.750000	2023-03-06 00:00:00	
50%	74727.500000	2023-04-24 00:00:00	
75%	112094.250000	2023-05-30 00:00:00	
max	149456.000000	2023-06-30 00:00:00	
std	43153.600016	NaN	

		transaction_time	transaction_qty	store_id \
count		149116	149116.000000	149116.000000
mean	0 days	12:14:15.815794415	1.438276	5.342063
min		0 days 06:00:00	1.000000	3.000000
25%	0 days	09:05:10.500000	1.000000	3.000000
50%		0 days 11:15:28	1.000000	5.000000
75%		0 days 15:25:57	2.000000	8.000000
max		0 days 20:59:32	8.000000	8.000000
std	0 days	03:45:57.901686173	0.542509	2.074241

	product_id	unit_price	sales \
count	149116.000000	149116.000000	149116.000000
mean	47.918607	3.382219	4.686367
min	1.000000	0.800000	0.800000
25%	33.000000	2.500000	3.000000
50%	47.000000	3.000000	3.750000
75%	60.000000	3.750000	6.000000
max	87.000000	45.000000	360.000000
std	17.930020	2.658723	4.227099

	datetime
count	149116
mean	2023-04-16 00:04:47.989404160
min	2023-01-01 07:06:11
25%	2023-03-06 13:37:08
50%	2023-04-24 08:24:32.500000
75%	2023-05-30 07:16:13
max	2023-06-30 20:57:19
std	NaN

```
[13]: df_location = df.groupby('store_location').agg({
        'sales': 'sum',
        'transaction_id': 'count'
    })
df_location
```

```
[13]:
```

	sales	transaction_id
store_location		
Astoria	232243.91	50599
Hell's Kitchen	236511.17	50735
Lower Manhattan	230057.25	47782

```
[14]: daily_sales_by_location = df.groupby(['transaction_date',
        ↪ 'store_location'])['sales'].sum().unstack()
daily_sales_by_location
```

```
[14]: store_location    Astoria    Hell's Kitchen    Lower Manhattan
transaction_date
2023-01-01           868.40           851.45           788.35
2023-01-02           925.50           828.80           649.05
2023-01-03           902.75           906.25           756.00
2023-01-04           808.25           781.65           630.20
2023-01-05           903.05           714.90           800.90
...
2023-06-26           1975.10          1746.10          2154.70
2023-06-27           1861.55          1676.70          2437.40
2023-06-28           1758.10          1445.85          1524.95
2023-06-29           1852.75          1298.55          1299.45
2023-06-30           1807.65          1904.93          1768.74
```

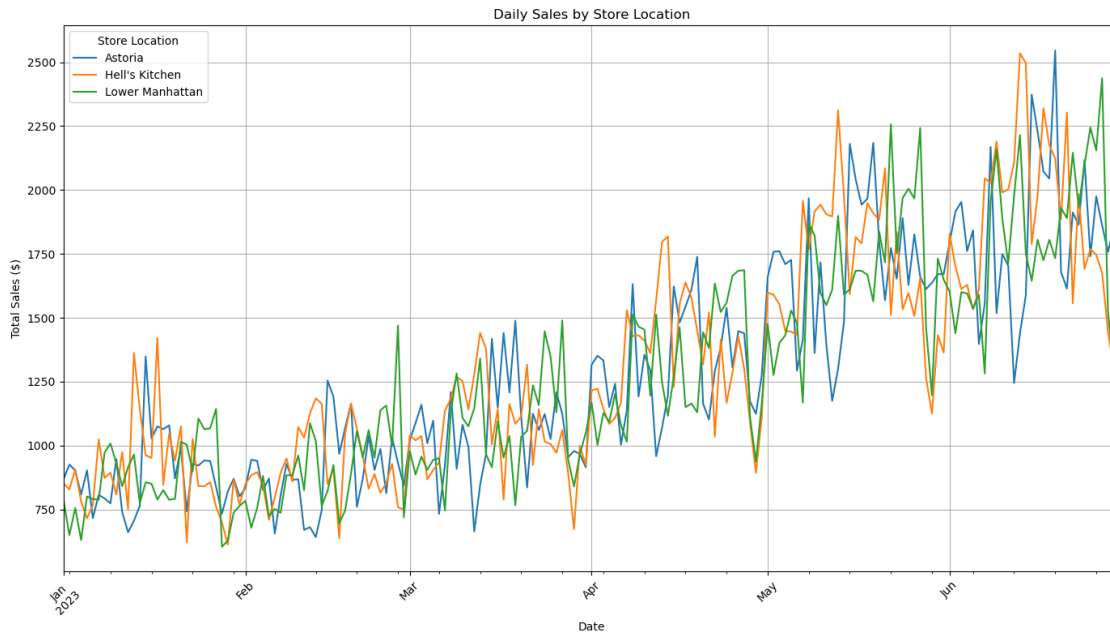
[181 rows x 3 columns]

```
[15]: test = df.groupby(['transaction_date', 'store_location'])['sales'].sum()
test
```

```
[15]: transaction_date    store_location
2023-01-01             Astoria           868.40
                Hell's Kitchen           851.45
                Lower Manhattan           788.35
2023-01-02             Astoria           925.50
                Hell's Kitchen           828.80
...
2023-06-29             Hell's Kitchen       1298.55
                Lower Manhattan       1299.45
2023-06-30             Astoria           1807.65
                Hell's Kitchen       1904.93
                Lower Manhattan       1768.74
```

Name: sales, Length: 543, dtype: float64

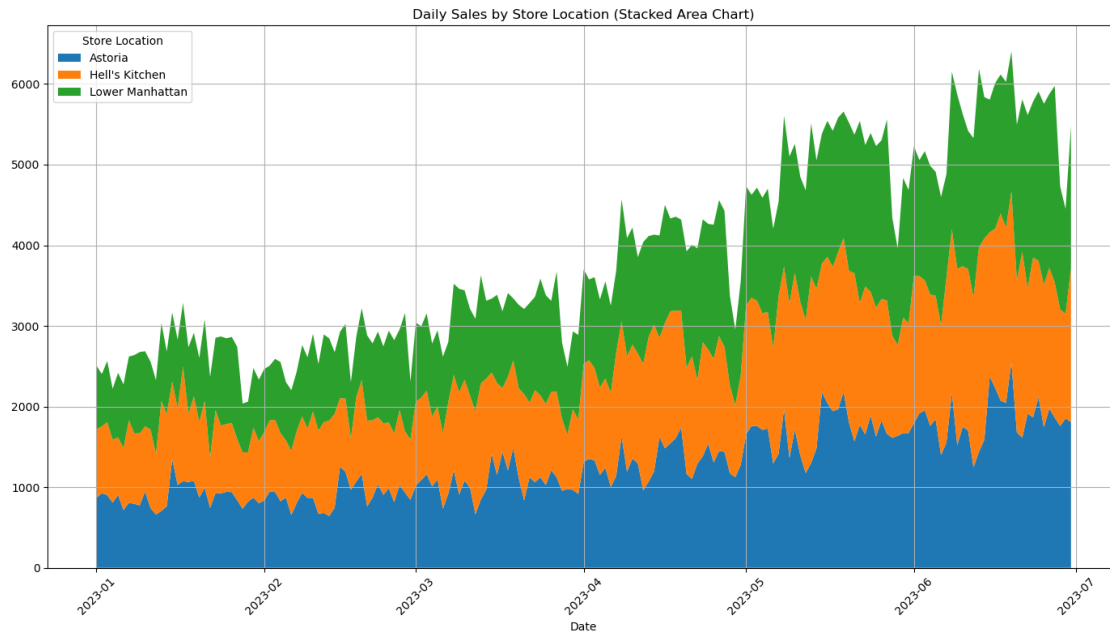
```
[16]: daily_sales_by_location.plot(figsize=(14,8), title='Daily Sales by Store_
      ↪Location')
plt.xlabel('Date')
plt.ylabel('Total Sales ($)')
plt.legend(title='Store Location')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
#show the plot
plt.show()
```



```
[18]: plt.figure(figsize=(14,8))
plt.stackplot(daily_sales_by_location.index, daily_sales_by_location.T, labels_
    ↪= daily_sales_by_location.columns)
plt.title('Daily Sales by Store Location (Stacked Area Chart)')
plt.xlabel('Date')
plt.legend(title='Store Location', loc='upper left')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
#show the plot

plt.show
```

```
[18]: <function matplotlib.pyplot.show(close=None, block=None)>
```



```
[24]: daily_sales = df.groupby('transaction_date')['sales'].sum().reset_index()

df['day_of_week'] = df['datetime'].dt.day_name()

weekly_sales = df.groupby('day_of_week')['sales'].sum().reindex(
    ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']).reset_index()

df['month'] = df['datetime'].dt.to_period('M')
monthly_sales = df.groupby('month')['sales'].sum().reset_index()
daily_sales, weekly_sales, monthly_sales
```

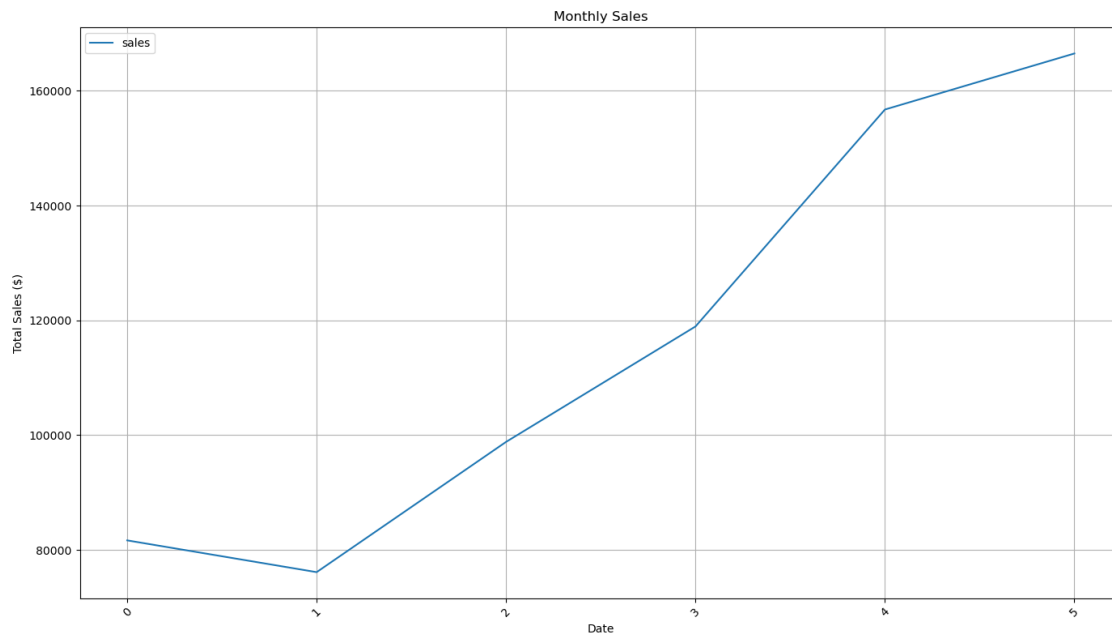
```
[24]: (   transaction_date    sales
0      2023-01-01  2508.20
1      2023-01-02  2403.35
2      2023-01-03  2565.00
3      2023-01-04  2220.10
4      2023-01-05  2418.85
..      ...
176    2023-06-26  5875.90
177    2023-06-27  5975.65
178    2023-06-28  4728.90
179    2023-06-29  4450.75
180    2023-06-30  5481.32
```

```
[181 rows x 2 columns],
```

	day_of_week	sales
0	Monday	101677.28
1	Tuesday	99455.94
2	Wednesday	100313.54
3	Thursday	100767.78
4	Friday	101373.00
5	Saturday	96894.48
6	Sunday	98330.31,

	month	sales
0	2023-01	81677.74
1	2023-02	76145.19
2	2023-03	98834.68
3	2023-04	118941.08
4	2023-05	156727.76
5	2023-06	166485.88)

```
[26]: monthly_sales.plot(figsize=(14,8), title='Monthly Sales')
plt.xlabel('Date')
plt.ylabel('Total Sales ($)')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
#show the plot
plt.show()
```




```
[27]: weekly_sales = df.groupby(['day_of_week', 'store_location'])['sales'].sum().
      ↪unstack()
      days_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
      ↪'Saturday', 'Sunday']
      weekly_sales = weekly_sales.reindex(days_order)
```

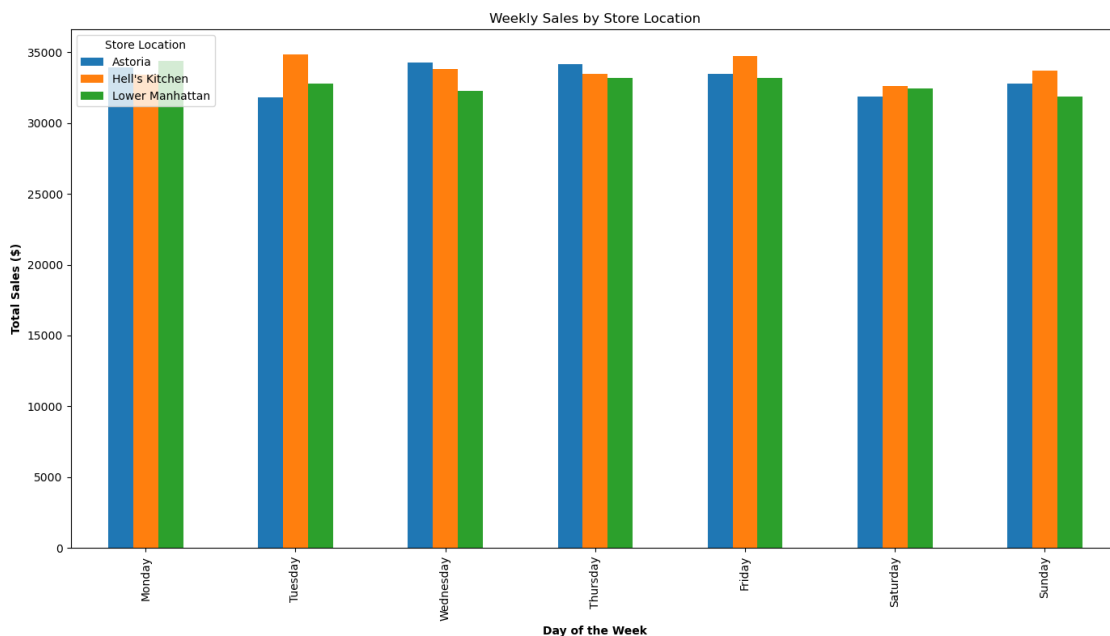
```
[28]: weekly_sales
```

```
[28]: store_location  Astoria  Hell's Kitchen  Lower Manhattan
day_of_week
Monday           33928.29        33389.51        34359.48
Tuesday          31816.76        34846.47        32792.71
Wednesday        34244.63        33779.09        32289.82
Thursday          34140.37        33468.06        33159.35
Friday            33472.75        34743.18        33157.07
Saturday          31845.93        32599.23        32449.32
Sunday            32795.18        33685.63        31849.50
```

```
[29]: weekly_sales[["Astoria","Hell's Kitchen","Lower Manhattan"]].
      ↪plot(kind='bar',figsize=(14,8))

plt.xlabel('Day of the Week', fontweight='bold')
plt.ylabel('Total Sales ($)', fontweight='bold')
plt.title('Weekly Sales by Store Location')
plt.legend(title='Store Location', loc='upper left')

#show the plot
plt.tight_layout()
plt.show()
```



```
[30]: df_category = df.groupby('product_category').agg({
        'sales': 'sum',
        'transaction_id': 'count'
    }).sort_values('sales', ascending = False)

df_category
```

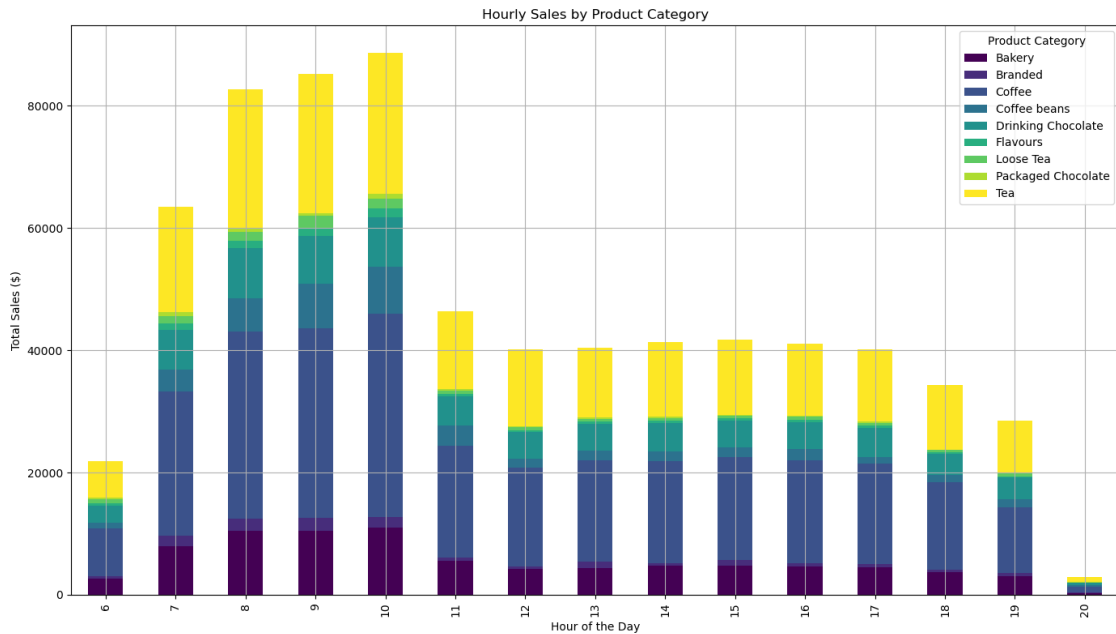
```
[30]:
```

	sales	transaction_id
product_category		
Coffee	269952.45	58416
Tea	196405.95	45449
Bakery	82315.64	22796
Drinking Chocolate	72416.00	11468
Coffee beans	40085.25	1753
Branded	13607.00	747
Loose Tea	11213.60	1210
Flavours	8408.80	6790
Packaged Chocolate	4407.64	487

```
[32]: df['hour'] = df['datetime'].dt.hour
hourly_sales_by_product = df.groupby(['hour', 'product_category'])['sales'].
    ↪sum().unstack()

hourly_sales_by_product.plot(kind='bar', stacked=True, figsize=(14,8),
    ↪colormap='viridis')
plt.title('Hourly Sales by Product Category')
plt.xlabel('Hour of the Day')
plt.ylabel('Total Sales ($)')
plt.legend(title='Product Category', loc='upper right')
plt.grid(True)
plt.tight_layout()

plt.show()
```



```
[33]: df_category['avg_sales'] = df_category['sales']/df_category['transaction_id']
df_category
```

```
[33]:
```

	sales	transaction_id	avg_sales
product_category			
Coffee	269952.45	58416	4.621207
Tea	196405.95	45449	4.321458
Bakery	82315.64	22796	3.610969
Drinking Chocolate	72416.00	11468	6.314615
Coffee beans	40085.25	1753	22.866657
Branded	13607.00	747	18.215529
Loose Tea	11213.60	1210	9.267438
Flavours	8408.80	6790	1.238409
Packaged Chocolate	4407.64	487	9.050595

```
[36]: sales_by_product = df.groupby('product_type').agg({
    'sales': 'sum',
    'unit_price': 'mean'
}).sort_values('sales', ascending = False)

sales_by_product
```

```
[36]:
```

	sales	unit_price
product_type		
Barista Espresso	91406.20	3.655200
Brewed Chai tea	77081.95	2.936993

Hot chocolate	72416.00	4.148827
Gourmet brewed coffee	70034.60	2.696641
Brewed Black tea	47932.00	2.744670
Brewed herbal tea	47539.50	2.744731
Premium brewed coffee	38781.15	3.122452
Organic brewed coffee	37746.50	2.898021
Scone	36866.12	3.534638
Drip coffee	31984.00	2.484959
Pastry	25655.99	3.685979
Brewed Green tea	23852.50	2.742462
Biscotti	19793.53	3.419503
Premium Beans	14583.50	34.028274
Organic Beans	8509.50	20.287952
Housewares	7444.00	13.448669
Gourmet Beans	6798.00	18.573770
Clothing	6163.00	27.886878
Regular syrup	6084.80	0.800000
Espresso Beans	5560.25	17.430251
Chai tea	4301.25	9.709368
House blend Beans	3294.00	18.000000
Herbal tea	2729.75	8.950000
Drinking Chocolate	2728.04	10.255789
Black tea	2711.85	8.950000
Sugar free syrup	2324.00	0.800000
Organic Chocolate	1679.60	7.600000
Green tea	1470.75	9.250000
Green beans	1340.00	10.000000

```
[37]: product_sales_df = df.groupby(['product_category', 'product_type'])['sales'].
      ↪sum().reset_index()
      product_sales_df
```

```
[37]:
```

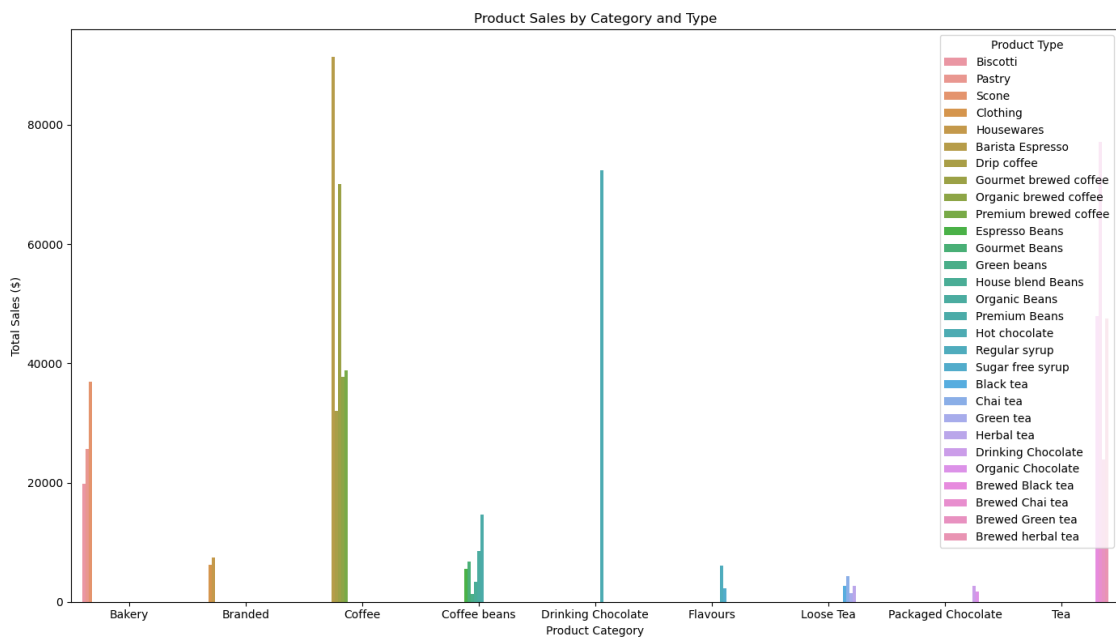
	product_category	product_type	sales
0	Bakery	Biscotti	19793.53
1	Bakery	Pastry	25655.99
2	Bakery	Scone	36866.12
3	Branded	Clothing	6163.00
4	Branded	Housewares	7444.00
5	Coffee	Barista Espresso	91406.20
6	Coffee	Drip coffee	31984.00
7	Coffee	Gourmet brewed coffee	70034.60
8	Coffee	Organic brewed coffee	37746.50
9	Coffee	Premium brewed coffee	38781.15
10	Coffee beans	Espresso Beans	5560.25
11	Coffee beans	Gourmet Beans	6798.00
12	Coffee beans	Green beans	1340.00
13	Coffee beans	House blend Beans	3294.00

14	Coffee beans	Organic Beans	8509.50
15	Coffee beans	Premium Beans	14583.50
16	Drinking Chocolate	Hot chocolate	72416.00
17	Flavours	Regular syrup	6084.80
18	Flavours	Sugar free syrup	2324.00
19	Loose Tea	Black tea	2711.85
20	Loose Tea	Chai tea	4301.25
21	Loose Tea	Green tea	1470.75
22	Loose Tea	Herbal tea	2729.75
23	Packaged Chocolate	Drinking Chocolate	2728.04
24	Packaged Chocolate	Organic Chocolate	1679.60
25	Tea	Brewed Black tea	47932.00
26	Tea	Brewed Chai tea	77081.95
27	Tea	Brewed Green tea	23852.50
28	Tea	Brewed herbal tea	47539.50

```
[39]: plt.figure(figsize=(14,8))
sns.barplot(data=product_sales_df, x='product_category', y='sales',
            hue='product_type')

plt.title('Product Sales by Category and Type')
plt.xlabel('Product Category')
plt.ylabel('Total Sales ($)')
plt.legend(title='Product Type', loc='upper right')

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



[]: