



```
In [5]: import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

```
In [7]: df = pd.read_csv("blinkit_data.csv")
```

```
In [9]: df.head()
```

```
Out[9]:
```

	Item Fat Content	Item Identifier	Item Type	Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	
0	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Small
1	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Small
2	Regular	FDR28	Frozen Foods	2010	OUT046	Tier 1	Small	Small
3	Regular	FDL50	Canned	2000	OUT013	Tier 3	High	Small
4	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Small

```
In [11]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Item Fat Content                      8523 non-null   object
1   Item Identifier                      8523 non-null   object
2   Item Type                            8523 non-null   object
3   Outlet Establishment Year             8523 non-null   int64
4   Outlet Identifier                    8523 non-null   object
5   Outlet Location Type                 8523 non-null   object
6   Outlet Size                          8523 non-null   object
7   Outlet Type                          8523 non-null   object
8   Item Visibility                      8523 non-null   float64
9   Item Weight                          7060 non-null   float64
10  Sales                               8523 non-null   float64
11  Rating                              8523 non-null   float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB
```

```
In [13]: df['Item Fat Content'] = df['Item Fat Content'].replace({'LF': 'Low Fat', 'low f
```

```
In [15]: df['Item Fat Content'].unique()
```

```
Out[15]: array(['Regular', 'Low Fat'], dtype=object)
```

KPI'S REQUIRMENTS

```
In [40]: # Total Sales
total_sales = df['Sales'].sum()

# Average Sales
average_sales = df['Sales'].mean()

# no of iteam Sold
no_of_item_sold = df['Sales'].count()

# Average ratings
avg_ratings = df['Rating'].mean()
# Display
print(f"Total sales:$ {total_sales:.0f}")
print(f"Average Sales: ${average_sales:.1f}")
print(f"Item Sold:$ {no_of_item_sold:,.0f}")
print(f"Average ratings:$ {avg_ratings:.1f}")
```

```
Total sales:$ 1201681
Average Sales: $141.0
Item Sold:$ 8,523
Average ratings:$ 4.0
```

Chart requirments

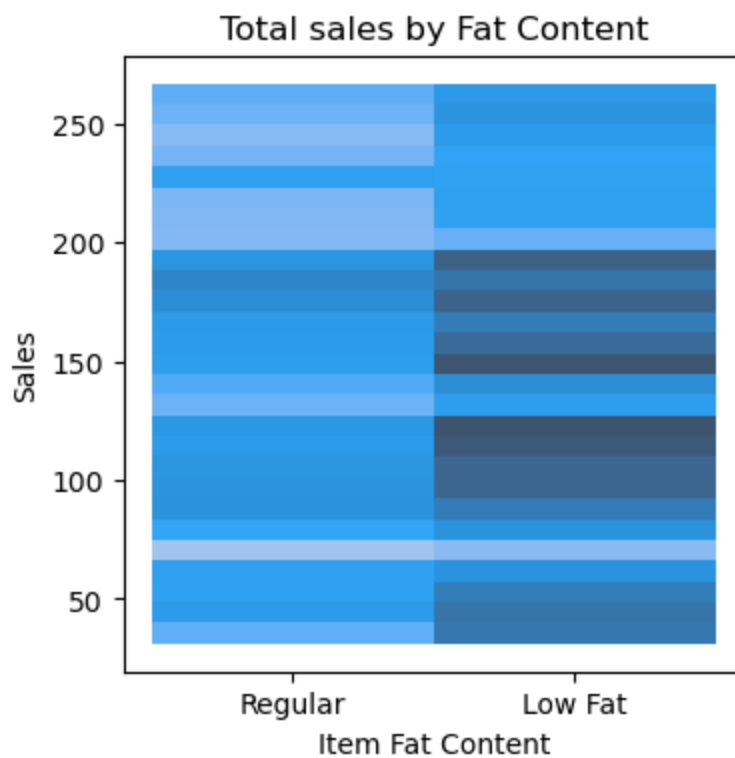
```
In [47]: # Total sales by Fat Content
Total_sales_by_Fat_Content = df.groupby('Item Fat Content')['Sales'].sum().reset_index()

Total_sales_by_Fat_Content
```

```
Out[47]:
```

	Item Fat Content	Sales
0	Low Fat	776319.6784
1	Regular	425361.8024

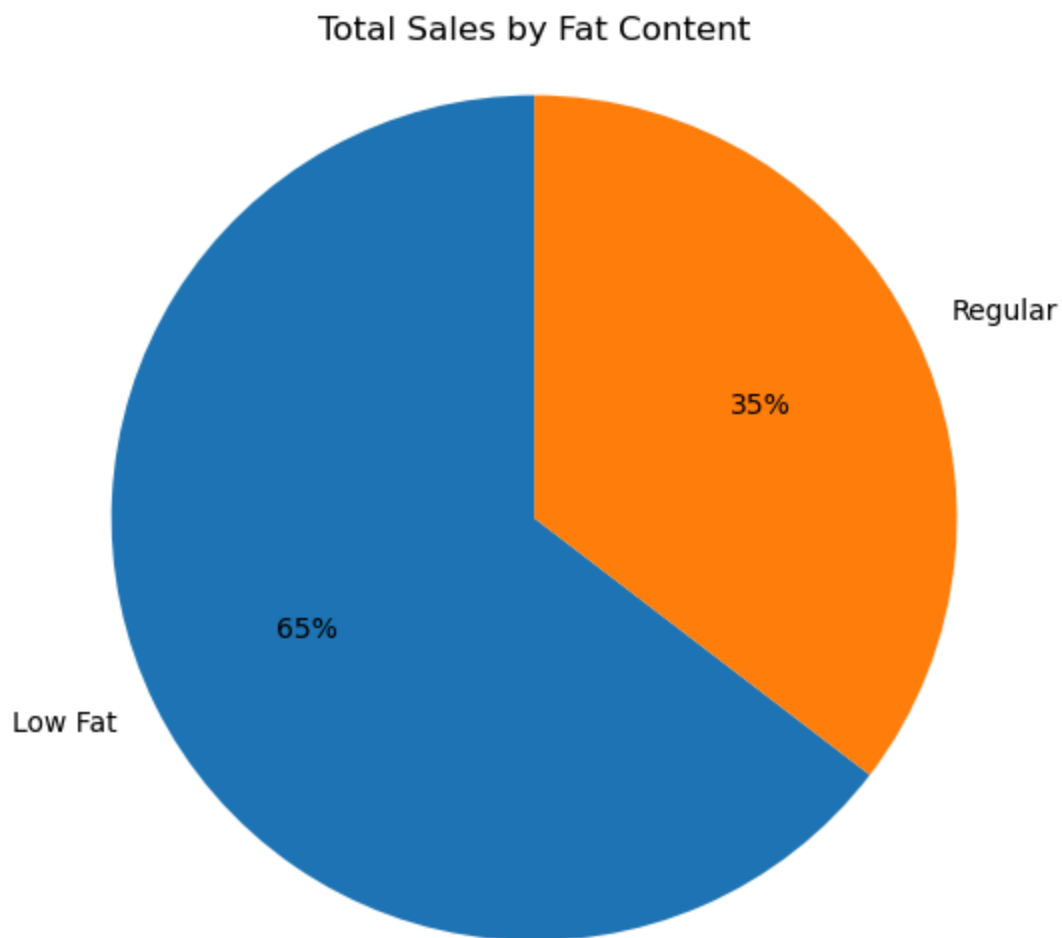
```
In [69]: plt.figure(figsize=(4,4))
ax = sns.histplot(data = df, x = 'Item Fat Content', y = 'Sales')
for container in ax.containers:
    ax.bar_label(container, fmt='%.0f')
plt.xlabel('Item Fat Content')
plt.ylabel('Sales')
plt.title("Total sales by Fat Content")
plt.show()
```



```
In [67]: # Grouping total sales by fat content
Total_sales_by_Fat_Content = df.groupby('Item Fat Content')['Sales'].sum()

# Making sure the values are numeric (optional safety step)
Total_sales_by_Fat_Content = Total_sales_by_Fat_Content.astype(float)

# Pie chart
plt.figure(figsize=(6, 6))
plt.pie(
    Total_sales_by_Fat_Content.values, # pass only numeric values here
    labels=Total_sales_by_Fat_Content.index, # labels as index
    autopct='%.0f%%',
    startangle=90
)
plt.title("Total Sales by Fat Content")
plt.axis('equal') # Makes the pie chart circular
plt.show()
```

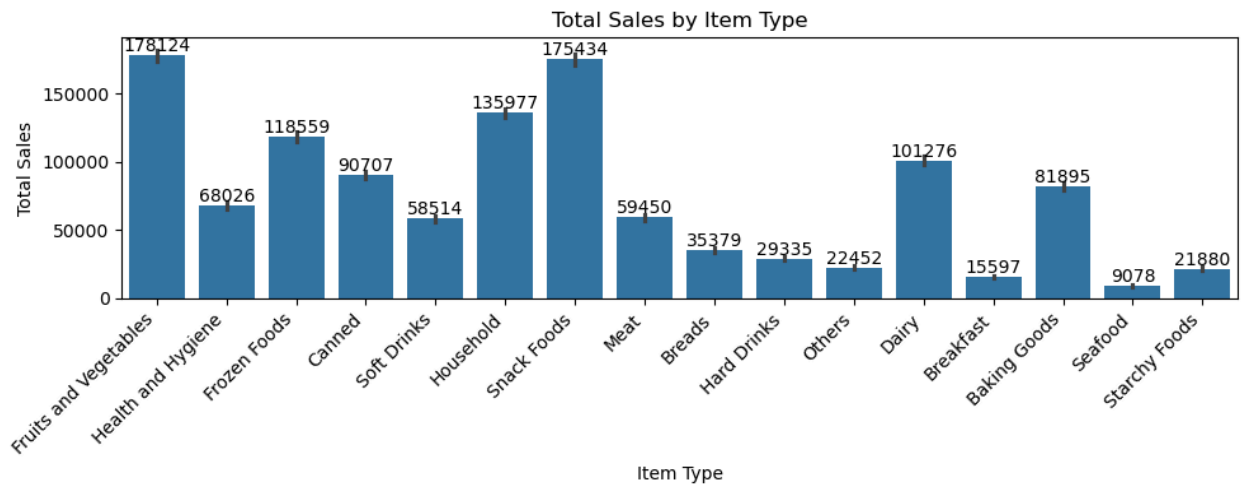


```
In [89]: # Total sales by Item Type
Total_sales_by_Item_type = df.groupby('Item Type')['Sales'].sum()
Total_sales_by_Item_type
# Making sure the values are numeric (optional safety step)
Total_sales_by_Item_type = Total_sales_by_Item_type.astype(float)
```

```
In [109]: plt.figure(figsize=(10, 4))
ax = sns.barplot(data=df, x='Item Type', y='Sales', estimator=sum)

# Add value labels using bar_label
for container in ax.containers:
    ax.bar_label(container, fmt='%.0f')

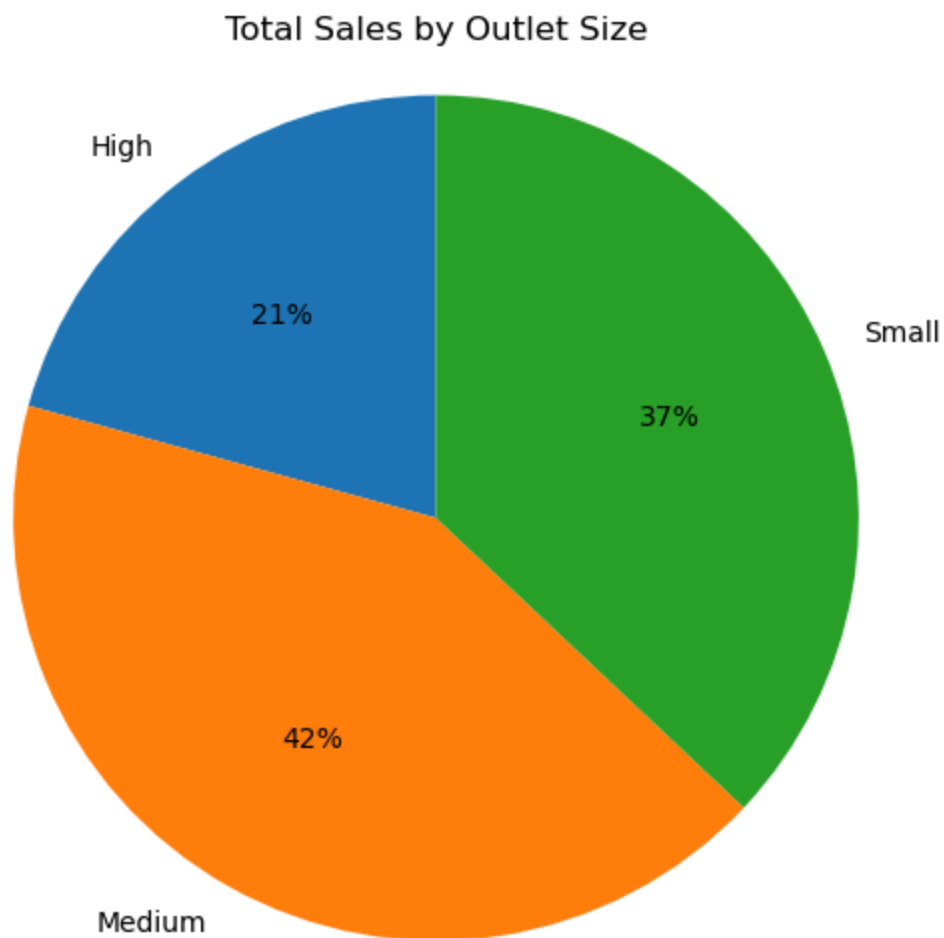
plt.xlabel('Item Type')
plt.ylabel('Total Sales')
plt.title("Total Sales by Item Type")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



```
In [129... # Total sales by Outlet size
Total_sales_by_Outlet_size = df.groupby('Outlet Size')['Sales'].sum().reset_index()
Total_sales_by_Outlet_size
# Making sure the values are numeric (optional safety step)
Total_sales_by_Outlet_size['Sales'] = Total_sales_by_Outlet_size['Sales'].astype(int)
Total_sales_by_Outlet_size
```

```
Out[129...      Outlet Size  Sales
0           High 248991
1        Medium 507895
2          Small 444794
```

```
In [137... plt.figure(figsize=(6, 6))
plt.pie(
    Total_sales_by_Outlet_size['Sales'], # only numeric values
    labels=Total_sales_by_Outlet_size['Outlet Size'], # actual category labels
    autopct='%0.0f%%',
    startangle=90
)
plt.title("Total Sales by Outlet Size")
plt.axis('equal') # Make the pie circular
plt.show()
```



```
In [143]: # Fat Content by Outlet for total sales
fat_content_by_outlet = df.groupby(['Outlet Location Type', 'Item Fat Content'])
fat_content_by_outlet
```

```
Out[143]:
```

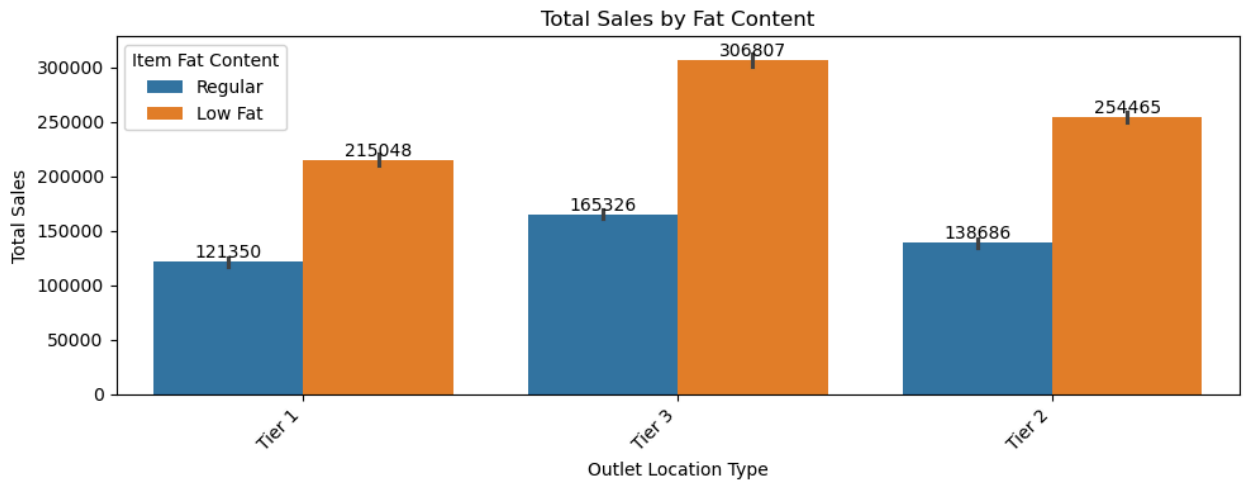
Item Fat Content	Low Fat	Regular
Outlet Location Type		
Tier 1	215047.9126	121349.8994
Tier 2	254464.7734	138685.8682
Tier 3	306806.9924	165326.0348

```
In [151]: plt.figure(figsize=(10, 4))
ax = sns.barplot(data=df, x='Outlet Location Type', y='Sales', estimator=sum,

# Add value labels using bar_label
for container in ax.containers:
    ax.bar_label(container, fmt='%.0f')

plt.xlabel('Outlet Location Type')
```

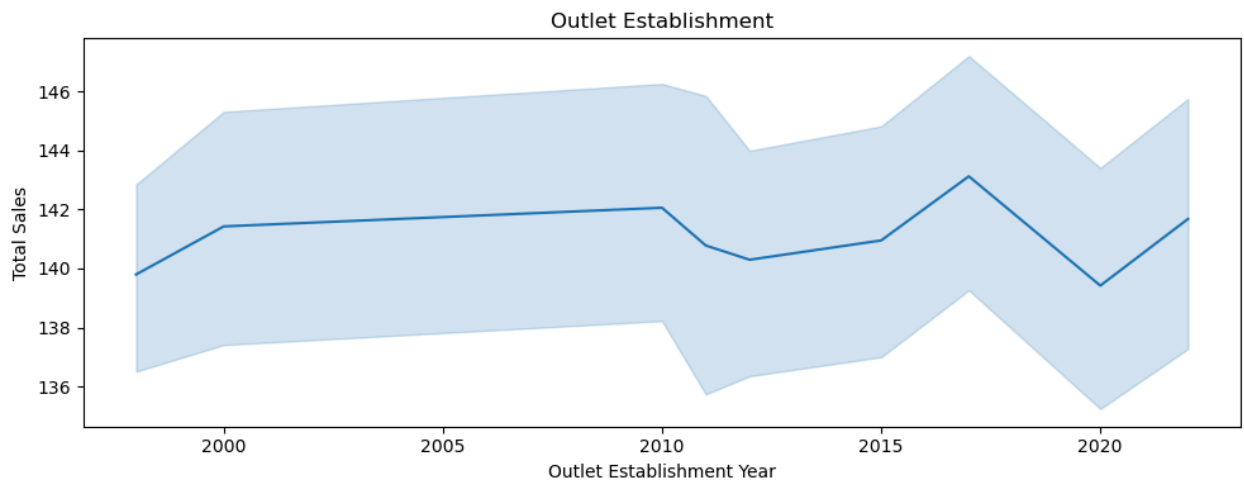
```
plt.ylabel('Total Sales')
plt.title("Total Sales by Fat Content")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



```
In [161]: # Total sales by Outlet Establishment Year
Outlet_Establishment_Year = df.groupby('Outlet Establishment Year')['Sales'].sum()
Outlet_Establishment_Year
```

```
Out[161]: Outlet Establishment Year
1998      204522.2570
2000      131809.0156
2010      132113.3698
2011       78131.5646
2012      130476.8598
2015      130942.7782
2017      133103.9070
2020      129103.9564
2022      131477.7724
Name: Sales, dtype: float64
```

```
In [167]: plt.figure(figsize=(10, 4))
ax = sns.lineplot(data=df, x='Outlet Establishment Year', y='Sales')
for container in ax.containers:
    ax.bar_label(container, fmt='%.0f')
plt.xlabel('Outlet Establishment Year')
plt.ylabel('Total Sales')
plt.title("Outlet Establishment")
plt.tight_layout()
plt.show()
```

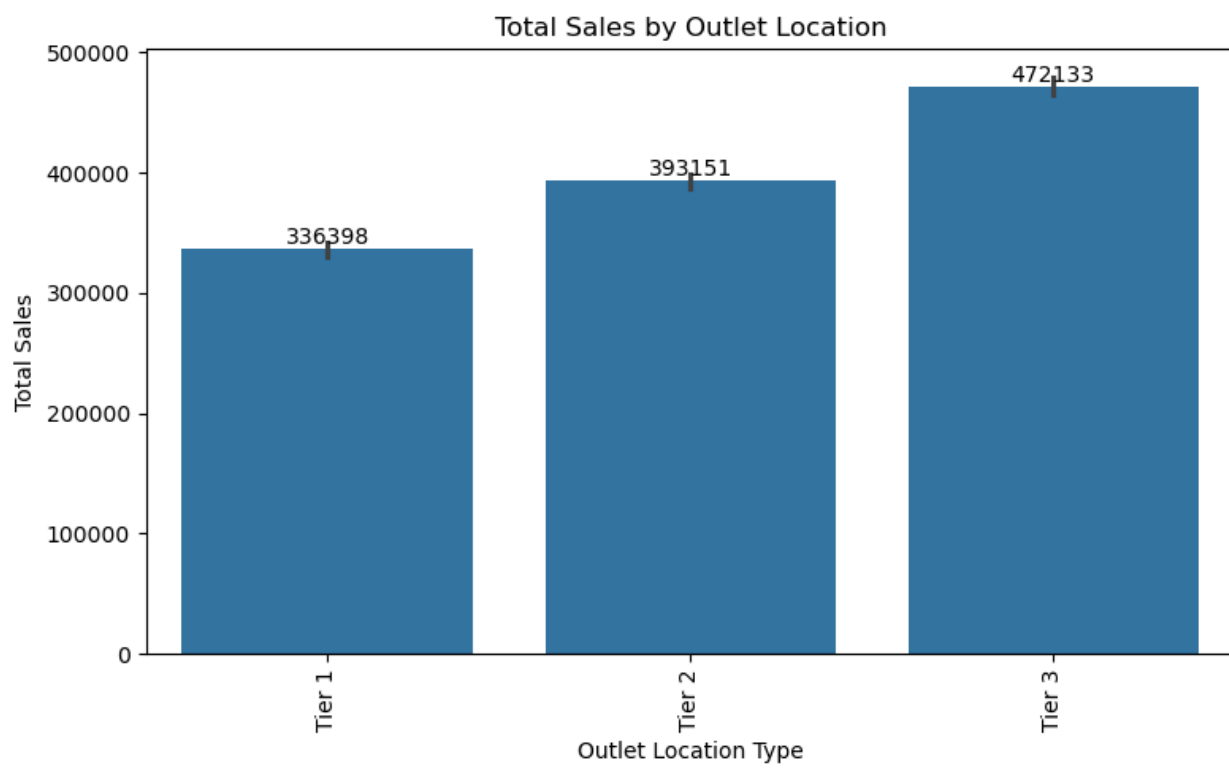


```
In [175... # Outlet Location
Outlet_Location = df.groupby('Outlet Location Type')['Sales'].sum().sort_index
Outlet_Location
```

```
Out[175... Outlet Location Type
Tier 1      336397.8120
Tier 2      393150.6416
Tier 3      472133.0272
Name: Sales, dtype: float64
```

```
In [195... plt.figure(figsize=(8, 5))
ax = sns.barplot(data=df, x='Outlet Location Type', y='Sales', estimator=sum, c
)
for container in ax.containers:
    ax.bar_label(container, fmt='%.0f')

plt.xlabel('Outlet Location Type')
plt.ylabel('Total Sales')
plt.title("Total Sales by Outlet Location")
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

In []: