

Restaurant Sales Analysis

- Data

```
menu_data.head()
```

	menu_item_id	item_name	category	price
0	101	Hamburger	American	12.95
1	102	Cheeseburger	American	13.95
2	103	Hot Dog	American	9.00
3	104	Veggie Burger	American	10.50
4	105	Mac & Cheese	American	7.00

```
order_data.head()
```

	order_details_id	order_id	order_date	order_time	item_id
0	1	1	1/1/23	11:38:36 AM	109.0
1	2	2	1/1/23	11:57:40 AM	108.0
2	3	2	1/1/23	11:57:40 AM	124.0
3	4	2	1/1/23	11:57:40 AM	117.0
4	5	2	1/1/23	11:57:40 AM	129.0

- Data Info

```
: menu_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 4 columns):
#   Column          Non-Null Count  Dtype
---  -
0   menu_item_id    32 non-null    int64
1   item_name       32 non-null    object
2   category        32 non-null    object
3   price           32 non-null    float64
dtypes: float64(1), int64(1), object(2)
memory usage: 1.1+ KB
```

```
order_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 12234 entries, 0 to 12233  
Data columns (total 5 columns):  
#   Column             Non-Null Count  Dtype  
---  -  
0   order_details_id    12234 non-null  int64  
1   order_id            12234 non-null  int64  
2   order_date          12234 non-null  object  
3   order_time          12234 non-null  object  
4   item_id             12097 non-null  float64  
dtypes: float64(1), int64(2), object(2)  
memory usage: 478.0+ KB
```

- **Check Null**

```
#Check Null  
menu_data.isnull().sum()
```

```
menu_item_id    0  
item_name       0  
category        0  
price          0  
dtype: int64
```

```
#Check Null  
order_data.isnull().sum()
```

```
order_details_id    0  
order_id            0  
order_date          0  
order_time          0  
item_id             137  
dtype: int64
```

- Here we can see in table order_data column item_id contains 137 null values.

- **Check Data Duplication**

```
#Check Duplication
menu_data.duplicated().sum()
```

```
0
```

```
#Check Duplication
order_data.duplicated().sum()
```

```
0
```

- **Handle Null Values**

- Drop Null values present in order_data table.

```
order_data["item_id"].unique()
```

```
array([109., 108., 124., 117., 129., 106., 119., 101., 114., 123., 126.,
       110., 122., 130., 132., 105., 102., 113., 104., 107., 121., 125.,
       111., 116., 127., 128., 118., 131., 120., 103., 112., nan, 115.])
```

```
order_data = order_data.dropna()
```

```
order_data["item_id"].unique()
```

```
array([109., 108., 124., 117., 129., 106., 119., 101., 114., 123., 126.,
       110., 122., 130., 132., 105., 102., 113., 104., 107., 121., 125.,
       111., 116., 127., 128., 118., 131., 120., 103., 112., 115.])
```

```
#Check Null
order_data.isnull().sum()
```

```
order_details_id    0
order_id            0
order_date          0
order_time          0
item_id             0
dtype: int64
```

```
order_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 12097 entries, 0 to 12233
Data columns (total 5 columns):
#   Column             Non-Null Count  Dtype
---  ---
0   order_details_id    12097 non-null  int64
1   order_id            12097 non-null  int64
2   order_date          12097 non-null  object
3   order_time          12097 non-null  object
4   item_id             12097 non-null  float64
dtypes: float64(1), int64(2), object(2)
memory usage: 567.0+ KB
```

- **Data Transformation**

- Change datatype of item_id float to integer.
- Change datatype of order_date object to date.

```
#convert item_id datatype float into int
order_data["item_id"] = np.ceil(order_data["item_id"]).astype(int)
```

```
order_data["item_id"].dtype
```

```
dtype('int32')
```

```
#convert order_date datatype object into date
order_data["order_date"] = pd.to_datetime(order_data["order_date"], format="mixed")
```

```
order_data["order_date"].info()
```

```
<class 'pandas.core.series.Series'>
Index: 12097 entries, 0 to 12233
Series name: order_date
Non-Null Count  Dtype
-----
12097 non-null  datetime64[ns]
dtypes: datetime64[ns](1)
memory usage: 189.0 KB
```

```
order_data.head()
```

	order_details_id	order_id	order_date	order_time	item_id
0	1	1	2023-01-01	11:38:36 AM	109
1	2	2	2023-01-01	11:57:40 AM	108
2	3	2	2023-01-01	11:57:40 AM	124
3	4	2	2023-01-01	11:57:40 AM	117
4	5	2	2023-01-01	11:57:40 AM	129

```
#Extract Month from date
order_data["Order Month"] = order_data["order_date"].dt.strftime('%b')
```

```
order_data.head()
```

	order_details_id	order_id	order_date	order_time	item_id	Order Month
0	1	1	2023-01-01	11:38:36 AM	109	Jan
1	2	2	2023-01-01	11:57:40 AM	108	Jan
2	3	2	2023-01-01	11:57:40 AM	124	Jan
3	4	2	2023-01-01	11:57:40 AM	117	Jan
4	5	2	2023-01-01	11:57:40 AM	129	Jan

- Merge two dataframes.

```
#Merge two Dataframes
data = pd.merge(menu_data, order_data, left_on='menu_item_id', right_on='item_id', how='inner')
data.head()
```

	menu_item_id	item_name	category	price	order_details_id	order_id	order_date	order_time	item_id	Order Month
0	101	Hamburger	American	12.95	11	6	2023-01-01	12:29:36 PM	101	Jan
1	101	Hamburger	American	12.95	26	11	2023-01-01	1:02:59 PM	101	Jan
2	101	Hamburger	American	12.95	43	17	2023-01-01	1:53:00 PM	101	Jan
3	101	Hamburger	American	12.95	63	24	2023-01-01	2:23:01 PM	101	Jan
4	101	Hamburger	American	12.95	71	27	2023-01-01	3:11:17 PM	101	Jan

- **Data Analysis and Data Visualization**

- **Different categories in Menu**

```
: #Category in Menu
category_list = pd.DataFrame(menu_data["category"].unique(), columns = ["category"])
category_list
```

	category
0	American
1	Asian
2	Mexican
3	Italian

- **Items Present in Menu**

```
##item list in menu
item_list = pd.DataFrame(menu_data["item_name"].unique(), columns = ["Items"])
item_list
```

	Items
0	Hamburger
1	Cheeseburger
2	Hot Dog
3	Veggie Burger
4	Mac & Cheese
5	French Fries
6	Orange Chicken
7	Tofu Pad Thai
8	Korean Beef Bowl
9	Pork Ramen
10	California Roll
11	Salmon Roll
12	Edamame

13	Potstickers
14	Chicken Tacos
15	Steak Tacos
16	Chicken Burrito
17	Steak Burrito
18	Chicken Torta
19	Steak Torta
20	Cheese Quesadillas
21	Chips & Salsa
22	Chips & Guacamole
23	Spaghetti
24	Spaghetti & Meatballs
25	Fettuccine Alfredo
26	Meat Lasagna
27	Cheese Lasagna
28	Mushroom Ravioli
29	Shrimp Scampi
30	Chicken Parmesan
31	Eggplant Parmesan

- Most Expensive and Most Cheap Item from Menu

```
#Most Expensive Item in menu
most_exp_item = menu_data.sort_values(['price'],ascending=[False])
most_exp_item.head(1)
```

	menu_item_id	item_name	category	price
29	130	Shrimp Scampi	Italian	19.95

```
#Most Cheap Item
most_cheap_item = menu_data.sort_values(['price'],ascending=[True])
most_cheap_item.head(1)
```

	menu_item_id	item_name	category	price
12	113	Edamame	Asian	5.0

- Total Orders and Total Sales

```
print('Total Orders : ',order_data["order_id"].nunique())
```

Total Orders : 5343

```
print('Total Sales : ',data["price"].sum().round(2))
```

Total Sales : 159217.9

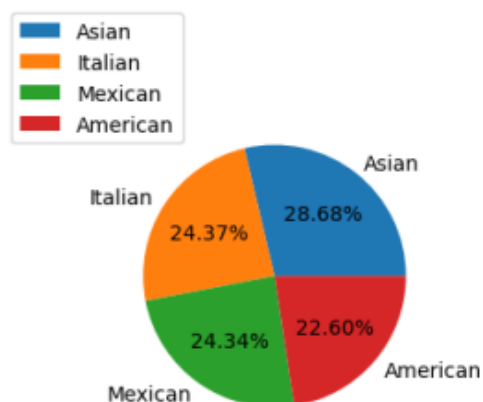
- Most Ordered Items by Category

```
#Most Item ordered by which Category
#Create Dataframe
category_data = data[["order_details_id", "category"]].copy()
#Rename columns
category_data = category_data.rename(columns = {'order_details_id' : 'order', 'category': 'category'})
#Filter and Sorting
category_data = category_data.groupby('category').count().sort_values('order',ascending=[False])
category_data = category_data.head()
#Reset Index
category_data = category_data.reset_index()
category_data
```

	category	order
0	Asian	3470
1	Italian	2948
2	Mexican	2945
3	American	2734

```
: plt.pie(category_data["order"], labels = category_data["category"], autopct = "%0.2f%%", radius = 0.60)
plt.legend(loc=2)
plt.title("Most Item ordered by which Category")
plt.show()
```

Most Item ordered by which Category

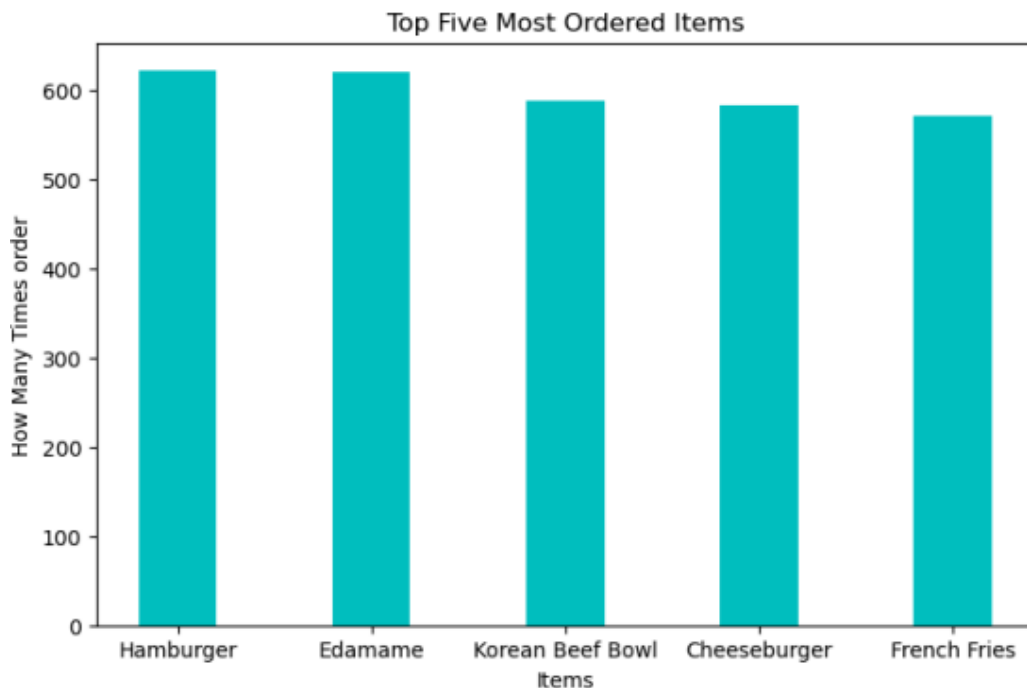


- Top Five Most Ordered Items From Menu

```
#Top five most ordered item
#Create Dataframe
most_order_data = data[["order_details_id","item_name"]].copy()
#Rename columns names
most_order_data = most_order_data.rename(columns ={'order_details_id' : 'How Many Times order','item_name': 'items'})
#Group by and sorting
most_order_data = most_order_data.groupby('items').count().sort_values('How Many Times order',ascending=[False])
most_order_data = most_order_data.head()
#Reset Index
most_order_data = most_order_data.reset_index()
most_order_data
```

	items	How Many Times order
0	Hamburger	622
1	Edamame	620
2	Korean Beef Bowl	588
3	Cheeseburger	583
4	French Fries	571

```
plt.figure(figsize=(8,5))
plt.bar(most_order_data["items"],most_order_data["How Many Times order"],width = 0.4,color = "c")
plt.xlabel("Items")
plt.ylabel("How Many Times order")
plt.title("Top Five Most Ordered Items")
plt.show()
```

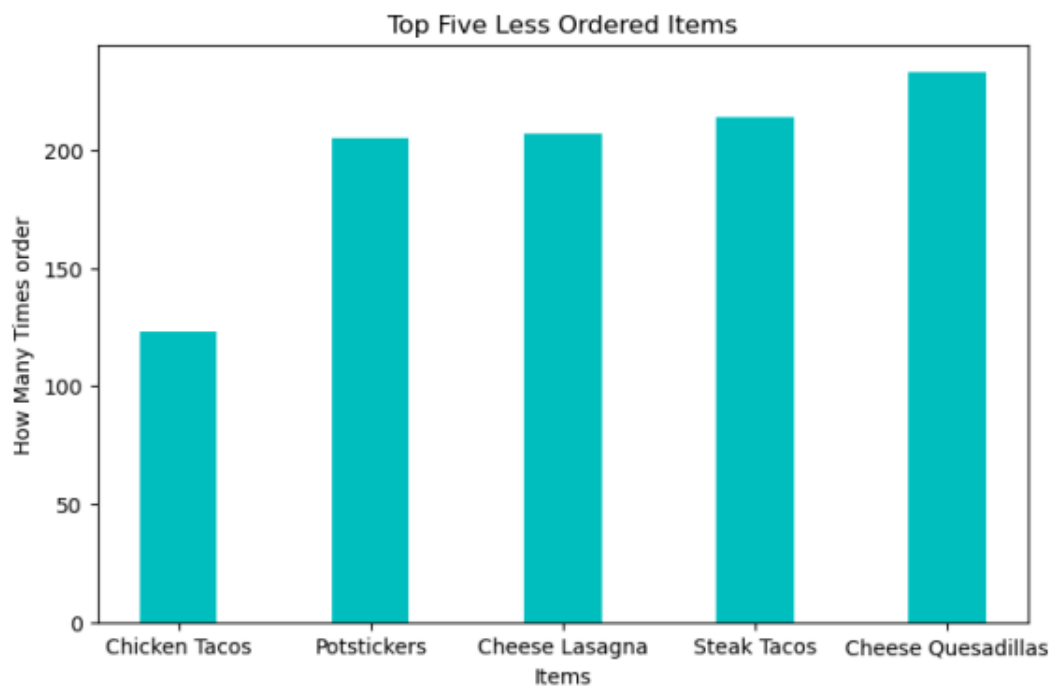


- Top Five Less Ordered Items From Menu

```
#Top five less ordered item
#Create Dataframe
less_order_data = data[["order_details_id", "item_name"]].copy()
#Rename columns names
less_order_data = less_order_data.rename(columns = {'order_details_id' : 'How Many Times order', 'item_name': 'items'})
#Group by and sorting
less_order_data = less_order_data.groupby('items').count().sort_values('How Many Times order', ascending=[True])
less_order_data = less_order_data.head()
#Reset Index
less_order_data = less_order_data.reset_index()
less_order_data
```

	items	How Many Times order
0	Chicken Tacos	123
1	Potstickers	205
2	Cheese Lasagna	207
3	Steak Tacos	214
4	Cheese Quesadillas	233

```
: plt.figure(figsize=(8,5))
plt.bar(less_order_data["items"],less_order_data["How Many Times order"],width = 0.4,color = "c")
plt.xlabel("Items")
plt.ylabel("How Many Times order")
plt.title("Top Five Less Ordered Items")
plt.show()
```

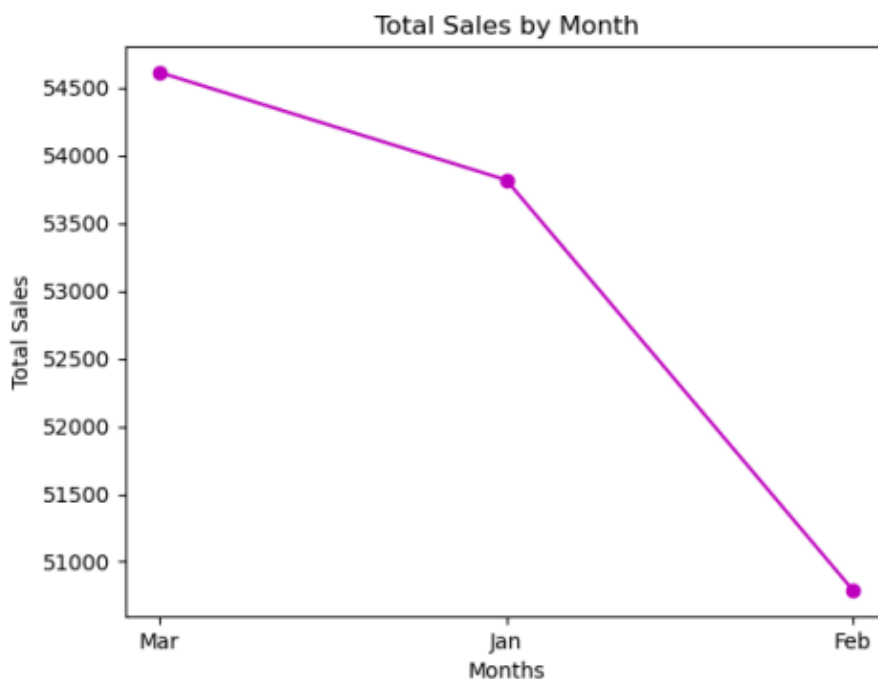


- Total Sales by Months

```
#Sales by Month
#Create Dataframe
sales_month_data = data[["Order Month", "price"]].copy()
#Rename Columns
sales_month_data = sales_month_data.rename(columns = {'price' : 'Total Sales'})
#Grouping and Sorting
sales_month_data = sales_month_data.groupby('Order Month').sum().sort_values('Total Sales',ascending=[False])
#Reset Index
sales_month_data = sales_month_data.reset_index()
sales_month_data
```

	Order Month	Total Sales
0	Mar	54610.60
1	Jan	53816.95
2	Feb	50790.35

```
plt.plot(sales_month_data["Order Month"],sales_month_data["Total Sales"],color = "m",marker = "o")
plt.xlabel("Months")
plt.ylabel("Total Sales")
plt.title("Total Sales by Month")
plt.show()
```

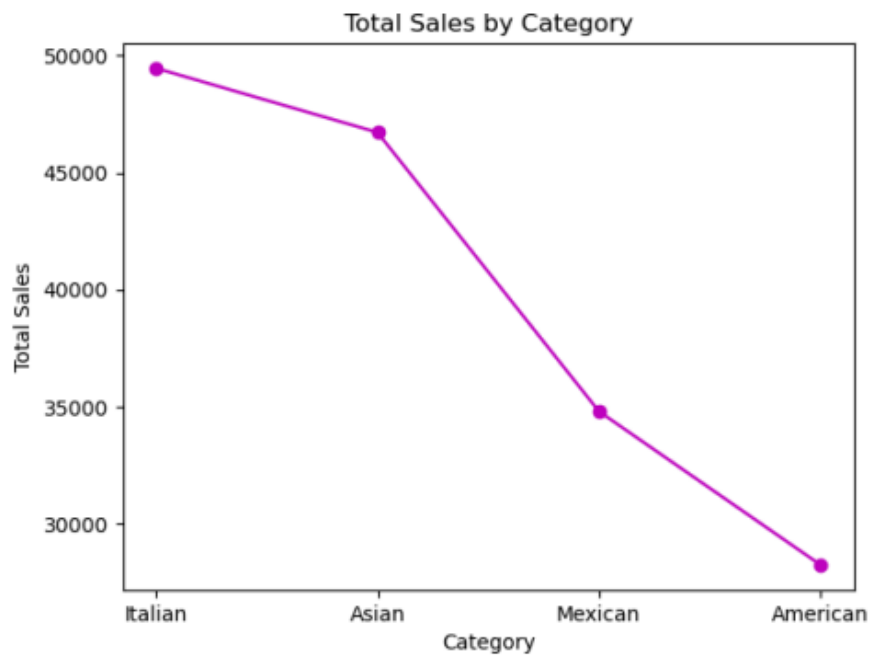


- Total Sales by Category

```
#Total Sales by Category
#Create DataFrame
Category_sales_data = data[["category", "price"]].copy()
#Rename columns
Category_sales_data = Category_sales_data.rename(columns = {'price' : 'Total Sales'})
#Grouping and Sorting
Category_sales_data = Category_sales_data.groupby('category').sum().sort_values('Total Sales', ascending=[False])
#Reset Index
Category_sales_data = Category_sales_data.reset_index()
Category_sales_data
```

	category	Total Sales
0	Italian	49462.70
1	Asian	46720.65
2	Mexican	34796.80
3	American	28237.75

```
plt.plot(Category_sales_data["category"], Category_sales_data["Total Sales"], color = "m", marker = "o")
plt.xlabel("Category")
plt.ylabel("Total Sales")
plt.title("Total Sales by Category")
plt.show()
```



- Total Sales by Items From Menu

```
#Total Sales by Items
#Create Dataframe
items_sales_data = data[["item_name","price"]].copy()
#Rename Columns
items_sales_data = items_sales_data.rename(columns ={'price' : 'Total Sales'})
#Grouping and Sorting
items_sales_data = items_sales_data.groupby('item_name').sum().sort_values('Total Sales',ascending=False)
#Reset Index
items_sales_data = items_sales_data.reset_index()
items_sales_data
```

	item_name	Total Sales
0	Korean Beef Bowl	10554.60
1	Spaghetti & Meatballs	8436.50
2	Tofu Pad Thai	8149.00
3	Cheeseburger	8132.85
4	Hamburger	8054.90
5	Orange Chicken	7524.00
6	Eggplant Parmesan	7119.00
7	Steak Torta	6821.55
8	Chicken Parmesan	6533.80
9	Pork Ramen	6462.00
10	Chicken Burrito	5892.25
11	Mushroom Ravioli	5564.50
12	Spaghetti	5321.50

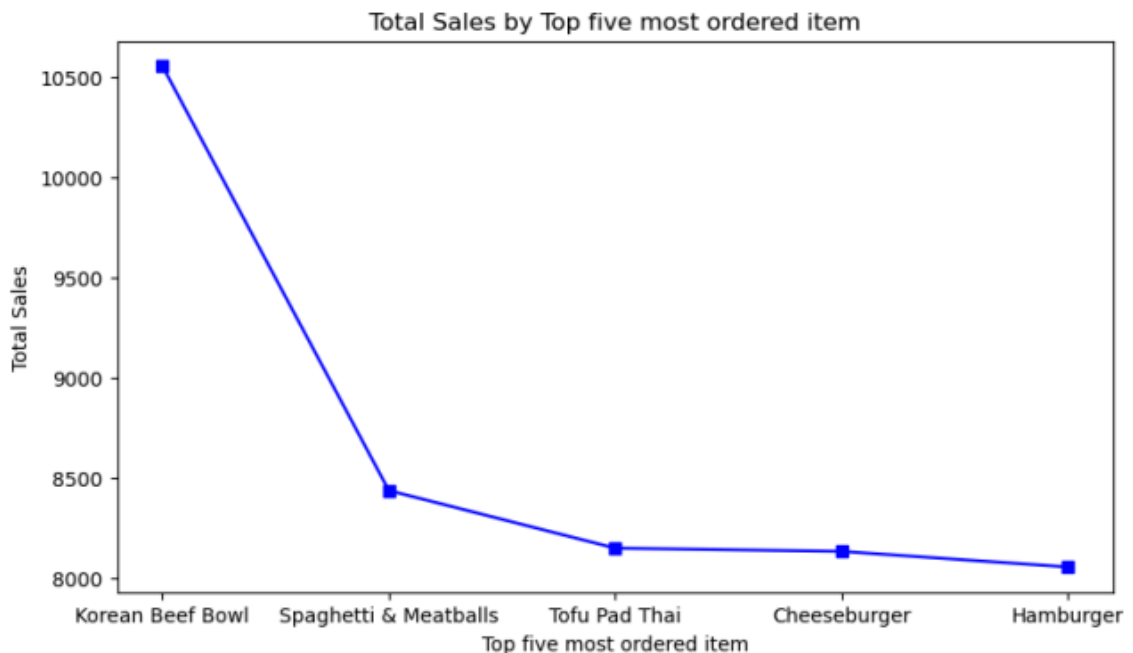
13	Steak Burrito	5292.30
14	Meat Lasagna	4900.35
15	Salmon Roll	4843.80
16	Shrimp Scampi	4768.05
17	Chicken Torta	4529.05
18	California Roll	4242.25
19	French Fries	3997.00
20	Fettuccine Alfredo	3610.50
21	Mac & Cheese	3241.00
22	Chips & Salsa	3227.00
23	Cheese Lasagna	3208.50
24	Edamame	3100.00
25	Steak Tacos	2985.30
26	Veggie Burger	2499.00
27	Cheese Quesadillas	2446.50
28	Hot Dog	2313.00
29	Chips & Guacamole	2133.00
30	Potstickers	1845.00
31	Chicken Tacos	1469.85

- Total Sales by Top Five Most Ordered Items From Menu

```
#Total Sales by Top five most ordered item
#Create Dataframe
sales_most_order_data = data[["price", "item_name"]].copy()
#Rename columns names
sales_most_order_data = sales_most_order_data.rename(columns = {'price' : 'Total Sales', 'item_name': 'items'})
#Group by and sorting
sales_most_order_data = sales_most_order_data.groupby('items').sum().sort_values('Total Sales', ascending=False)
sales_most_order_data = sales_most_order_data.head()
#Reset Index
sales_most_order_data = sales_most_order_data.reset_index()
sales_most_order_data
```

	items	Total Sales
0	Korean Beef Bowl	10554.60
1	Spaghetti & Meatballs	8436.50
2	Tofu Pad Thai	8149.00
3	Cheeseburger	8132.85
4	Hamburger	8054.90

```
plt.figure(figsize=(9,5))
plt.plot(sales_most_order_data["items"], sales_most_order_data["Total Sales"], color = "b", marker = "s")
plt.xlabel("Top five most ordered item")
plt.ylabel("Total Sales")
plt.title("Total Sales by Top five most ordered item")
plt.show()
```

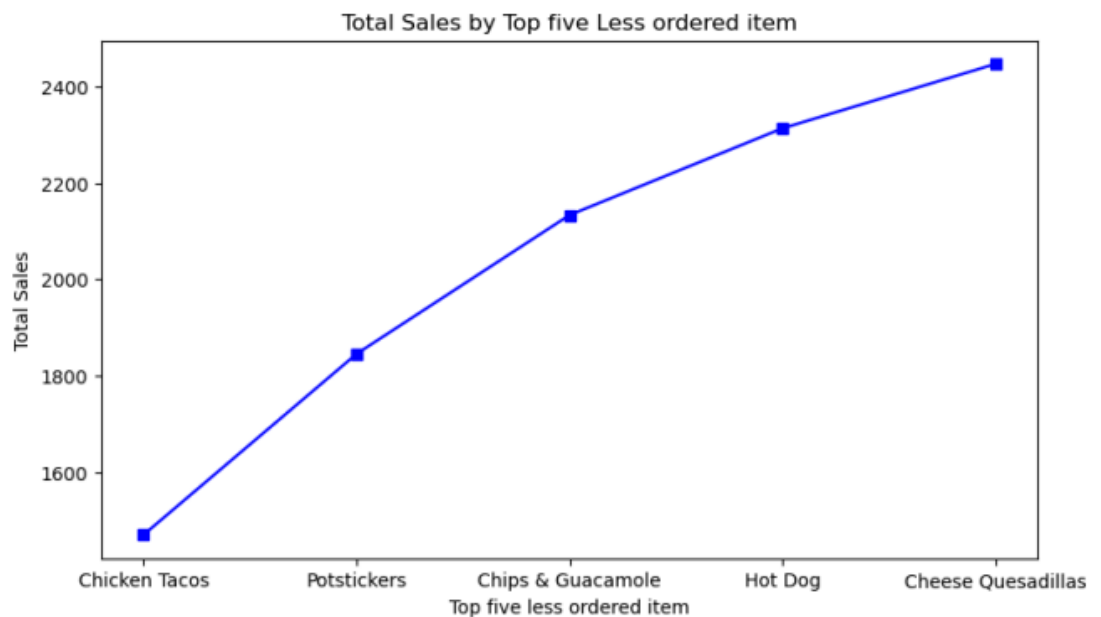


- Total Sales by Top Five Less Ordered Items from Menu

```
#Total Sales by Top five less ordered item
#Create Dataframe
sales_less_order_data = data[["price", "item_name"]].copy()
#Rename columns names
sales_less_order_data = sales_less_order_data.rename(columns={'price' : 'Total Sales', 'item_name': 'items'})
#Group by and sorting
sales_less_order_data = sales_less_order_data.groupby('items').sum().sort_values('Total Sales',ascending=[True])
sales_less_order_data = sales_less_order_data.head()
#Reset Index
sales_less_order_data = sales_less_order_data.reset_index()
sales_less_order_data
```

	items	Total Sales
0	Chicken Tacos	1469.85
1	Potstickers	1845.00
2	Chips & Guacamole	2133.00
3	Hot Dog	2313.00
4	Cheese Quesadillas	2446.50

```
plt.figure(figsize=(9,5))
plt.plot(sales_less_order_data["items"],sales_less_order_data["Total Sales"],color = "b",marker = "s")
plt.xlabel("Top five less ordered item")
plt.ylabel("Total Sales")
plt.title("Total Sales by Top five Less ordered item")
plt.show()
```



Key Findings & Conclusion

In this project, we have analyzed a dataset containing detailed restaurant menu data and order records. The objective was to gain insights into customer preferences, menu item popularity, and sales trends, with the aim of optimizing restaurant operations and enhancing the dining experience.

Key Findings

1. Restaurant Menu Categories:

- The dataset was categorized into four main culinary styles: American, Asian, Mexican, and Italian. Each category reflects a distinct type of cuisine, contributing to the restaurant's diverse menu offerings.

2. Most Popular Category:

- Analysis of customer orders revealed that the **Asian** category is the most favored among customers. This suggests a strong preference for Asian cuisine, which may be attributed to its unique flavors and variety of dishes.

3. Top Five Most Ordered Items:

- The following menu items emerged as the most popular among customers:
 - **Hamburger**
 - **Edamame**
 - **Korean Beef Bowl**
 - **Cheeseburger**
 - **French Fries**
- These items consistently top the order list, indicating their high appeal and frequent demand.

4. **Bottom Five Least Ordered Items:**

- Conversely, the least ordered items were:
 - **Chicken Tacos**
 - **Potstickers**
 - **Cheese Lasagna**
 - **Steak Tacos**
 - **Cheese Quesadillas**
- These dishes are less favored by customers, which may suggest a need for revising their placement on the menu or promotional strategies to boost their appeal.

5. **Sales Trends by Month in Quarter 1:**

- In the first quarter, sales data indicated that **March** was the highest-grossing month, followed by **January** and **February**. This trend highlights March as a peak period for sales, which could be due to seasonal factors or promotional events.

6. **Sales by Cuisine Category in Quarter 1:**

- Sales distribution across different culinary categories for the first quarter showed the following ranking:
 - **Italian**
 - **Asian**
 - **Mexican**
 - **American**
- Italian cuisine led the sales, followed by Asian, Mexican, and American. This reflects a notable preference for Italian dishes during this period, with Asian cuisine also showing significant popularity.

Implications and Recommendations

- **Menu Optimization:**

- Given the popularity of Asian cuisine and top items like the Hamburger and Korean Beef Bowl, there may be an opportunity to expand or highlight these offerings to enhance customer satisfaction and increase sales.
- For the least ordered items, consider evaluating the recipes, presentation, or marketing strategies to determine why these dishes are less popular and whether adjustments could improve their appeal.

- **Sales Strategy:**

- The sales peak in March suggests that targeted marketing or special promotions during this month could be beneficial. Additionally, understanding the reasons behind this peak could help replicate its success in other months.
- The preference for Italian and Asian cuisines in Quarter 1 can guide menu planning and inventory management to align with customer preferences, potentially boosting overall sales.

Overall, this analysis provides valuable insights into customer preferences and sales patterns, offering actionable recommendations to optimize menu offerings and improve restaurant performance. By leveraging these insights, the restaurant can make informed decisions to enhance customer satisfaction and drive sales growth.