

# PROBLEM STATEMENT

---

We want to visualize the various aspects of our pizza sales to gain insights and understand key trends. We have identified the following requirements for creating charts:

- 1. Daily trend for total orders:** Create a bar chart that displays the daily trend of total orders over a specific time period. This chart will help identify any patterns in order volumes on a daily basis
- 2. Hourly trend for total orders:** Create a line chart that illustrates the hourly trend of total orders throughout the day. This chart will help identify peak hours of high order activity.
- 3. Percentage of sales by pizza category:** Create a pie chart that shows the distribution of sales across different pizza categories. This chart will help provide insights into the popularity of various pizza categories and their contributions to overall sales.
- 4. Percentage of sales by Pizza size:** create a pie chart that represents the percentage of sales attributed to different pizza sizes. This will help understand customer preferences for pizza sales and their impact on sales
- 5. Total sales sold by pizza category:** create a funnel chart that represents the total number of pizzas sold for each pizza category. This chart will help us compare the sales performance of different pizza categories
- 6. Top 5 Best sellers by Revenue, total quantity and total orders:** create a bar chart that highlights the top 5 best selling pizzas based on revenue, total quantity and total orders. It will help identify the most popular pizza options
- 7. Bottom 5 Worst sellers by Revenue, total quantity and total orders:** Create a bar chart to help highlight the worst selling pizzas based on the revenue, total quantity and total orders. This chart will help identify underperforming or less popular pizza options

## Import Libraries

In [2]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

## Read Dataset:

```
In [3]: pizza = pd.read_excel(r"C:\Users\Admin\Desktop\PythonFiles\data_science\Pizza sales\pizza_sales_excel_file.xlsx")
```

```
In [4]: pizza.head(10)
```

	pizza_id	order_id	pizza_name_id	quantity	order_date	order_time	unit_price	total_price	pizza_size	pizza_category	pizza_ingredients	pi
0	1	1	hawaiian_m	1	2015-01-01	11:38:36	13.25	13.25	M	Classic	Sliced Ham, Pineapple, Mozzarella Cheese	ai
1	2	2	classic_dlx_m	1	2015-01-01	11:57:40	16.00	16.00	M	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppers,...	ai
2	3	2	five_cheese_l	1	2015-01-01	11:57:40	18.50	18.50	L	Veggie	Mozzarella Cheese, Provolone Cheese, Smoked Go...	ai
3	4	2	ital_supr_l	1	2015-01-01	11:57:40	20.75	20.75	L	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Oni...	ai
4	5	2	mexicana_m	1	2015-01-01	11:57:40	16.00	16.00	M	Veggie	Tomatoes, Red Peppers, Jalapeno Peppers, Red O...	ai
5	6	2	thai_ckn_l	1	2015-01-01	11:57:40	20.75	20.75	L	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, Tha...	ai
6	7	3	ital_supr_m	1	2015-01-01	12:12:28	16.50	16.50	M	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Oni...	ai
7	8	3	prsc_argla_l	1	2015-01-01	12:12:28	20.75	20.75	L	Supreme	Prosciutto di San Daniele, Arugula, Mozzarella...	ai
8	9	4	ital_supr_m	1	2015-01-01	12:16:31	16.50	16.50	M	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Oni...	ai

pizza_id	order_id	pizza_name_id	quantity	order_date	order_time	unit_price	total_price	pizza_size	pizza_category	pizza_ingredients	pi
9	10	5	ital_supr_m	1	2015-01-01	12:21:30	16.50	16.50	M	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red

Question 1: Find the total revenue ????????????

```
In [5]: # Find the sum of the total_price field
print("The total revenue is: $" , pizza["total_price"].sum())
```

The total revenue is: \$ 817860.05

Question 2: Find the Average order value ????????????

```
In [6]: #Derived field from dividing the total order sales by the total number of Orders
average_order_value = pizza["total_price"].sum() / pizza.groupby("order_id").sum().shape[0]
print("The average order value is: $" , average_order_value)

# pizza["order_id"].nunique()
```

The average order value is: \$ 38.30726229508197

Question 3: Find the Number of Total pizzas sold ????????????

```
In [7]: print ("The Number of Pizzas sold is:" , pizza["quantity"].sum())
```

The Number of Pizzas sold is: 49574

Question 4: Find the total number of orders placed?????????????????????

```
In [8]: print("The Total number of Orders placed is:" , pizza["order_id"].nunique())
```

The Total number of Orders placed is: 21350

Question 5: Find the Average Number of Pizzas per order?????????????????????

```
In [9]: #Derived field gotten by dividing the total number of pizzas by total number of orders
print( "The Average number of pizzas per order is:", pizza["quantity"].sum() / pizza["order_id"].nunique())
```

The Average number of pizzas per order is: 2.321967213114754

# Charts/ Graphs

## 1. Daily weekly trend for Total Orders:

```
In [10]: pizza.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48620 entries, 0 to 48619
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   pizza_id         48620 non-null   int64  
 1   order_id         48620 non-null   int64  
 2   pizza_name_id    48620 non-null   object  
 3   quantity          48620 non-null   int64  
 4   order_date        48620 non-null   datetime64[ns]
 5   order_time        48620 non-null   object  
 6   unit_price        48620 non-null   float64 
 7   total_price       48620 non-null   float64 
 8   pizza_size        48620 non-null   object  
 9   pizza_category    48620 non-null   object  
 10  pizza_ingredients 48620 non-null   object  
 11  pizza_name        48620 non-null   object  
dtypes: datetime64[ns](1), float64(2), int64(3), object(6)
memory usage: 4.5+ MB
```

```
In [11]: # create a new field for the weekday
pizza["weekday"] = pizza["order_date"].dt.weekday
```

```
#Drop duplicates on "order_id" then group by "weekday"
daily_total_order = pizza.drop_duplicates("order_id").groupby("weekday")["order_id"].count().reset_index(name="count")
daily_total_order.sort_values("count", ascending=False)
```

Out[11]:

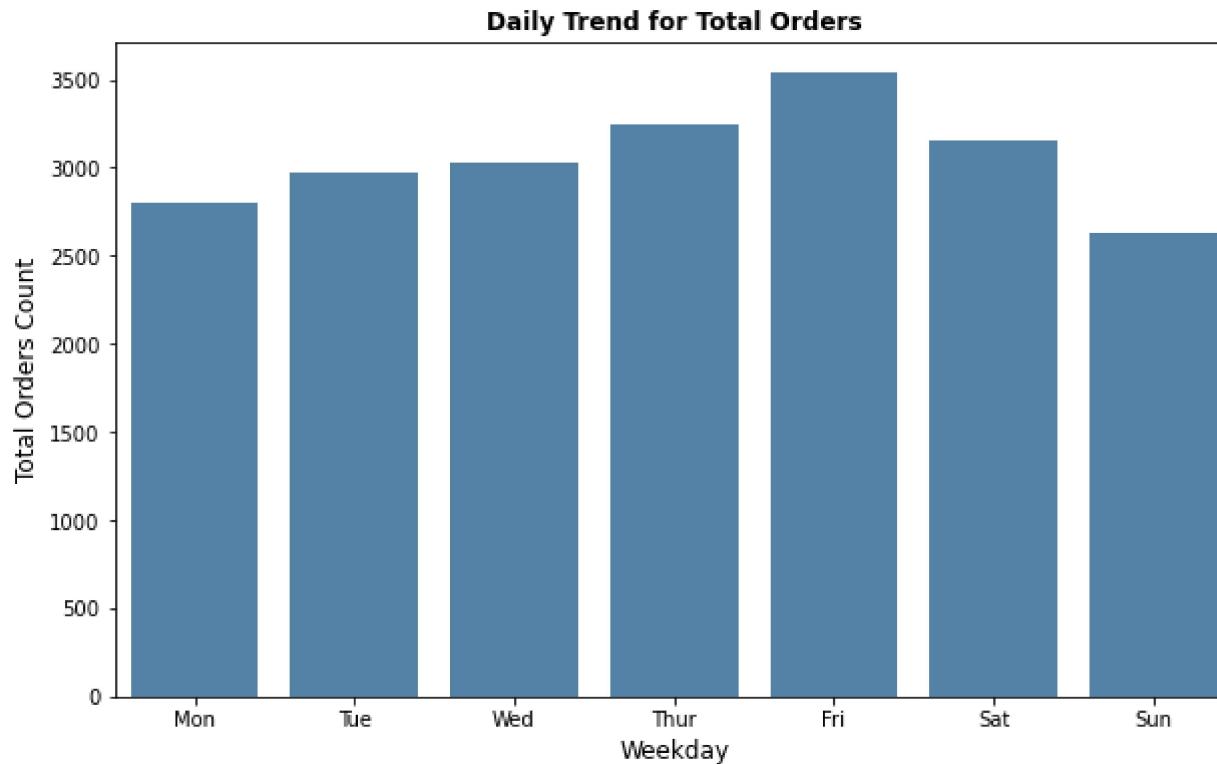
	weekday	count
4	4	3538
3	3	3239
5	5	3158
2	2	3024
1	1	2973
0	0	2794
6	6	2624

In [12]:

```
#Visaulization
x= [0,1,2,3,4,5,6]
days = ["Mon", "Tue", "Wed", "Thur", "Fri", "Sat", "Sun"]

plt.figure(figsize=(10,6))
A = sns.barplot(x = daily_total_order["weekday"], y= daily_total_order["count"], color="steelblue")
plt.title("Daily Trend for Total Orders",weight="bold",size=12)
plt.ylabel("Total Orders Count", size=12)
plt.xlabel("Weekday",size=12)
plt.xticks(x,days)

plt.show()
```



From the diagrams above, Friday has the most number of Orders!!!

## 2. Monthly trend for Total Orders:

```
In [13]: #Create month field n dataframe
pizza["month"] = pizza["order_date"].dt.month

monthly_total_order = pizza.drop_duplicates("order_id").groupby("month")["order_id"].count().reset_index(name="count")
monthly_total_order
```

Out[13]:

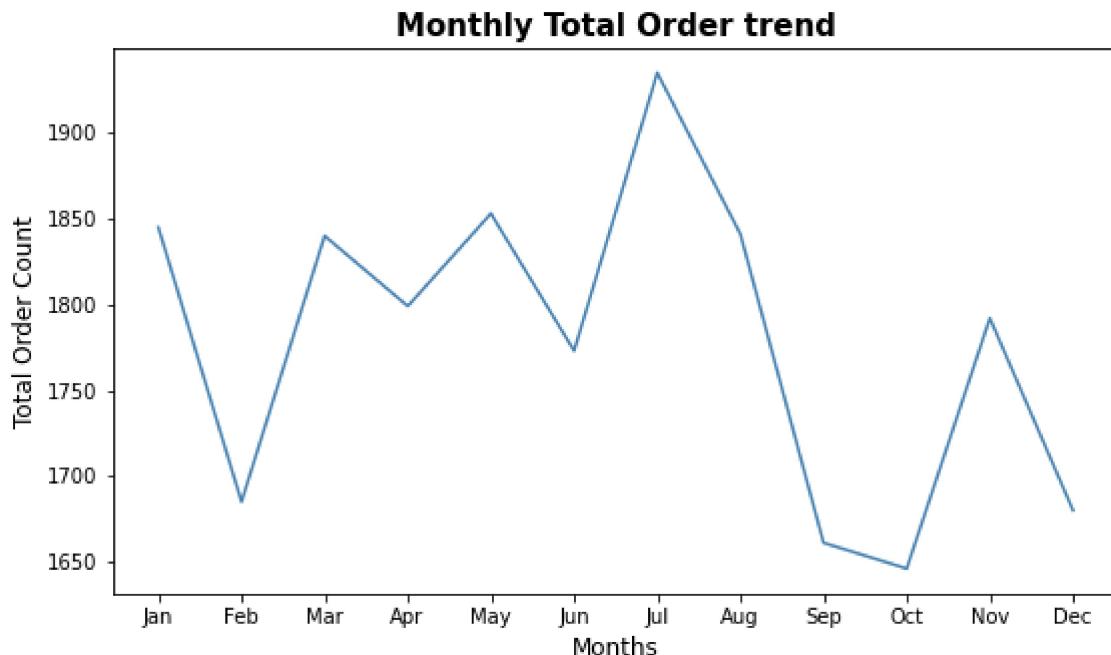
	month	count
0	1	1845
1	2	1685
2	3	1840
3	4	1799
4	5	1853
5	6	1773
6	7	1935
7	8	1841
8	9	1661
9	10	1646
10	11	1792
11	12	1680

In [14]: *##### Visualization(line chart)*

```
x = [1,2,3,4,5,6,7,8,9,10,11,12]
months=["Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"]

plt.figure(figsize=(9,5))
B = sns.lineplot(x = monthly_total_order["month"],y = monthly_total_order["count"], color="steelblue")
plt.title("Monthly Total Order trend",weight="bold",size=15)
plt.xticks(x, months)
plt.ylabel("Total Order Count",size=12)
plt.xlabel("Months",size=12)

plt.show()
```



From the chart above, we can see that July had the most Number of Total orders while October had the least number of Total orders

#### \* Hourly Total Order Trend

```
In [15]: #Augment the dataframe with hour column
pizza["order_time"] = pizza["order_time"].astype("str")
pizza["hour"] = pizza["order_time"].str[0:2]
pizza.head(2)

#change hour fromat from str to integer
pizza["hour"] = pizza["hour"].astype("int64")

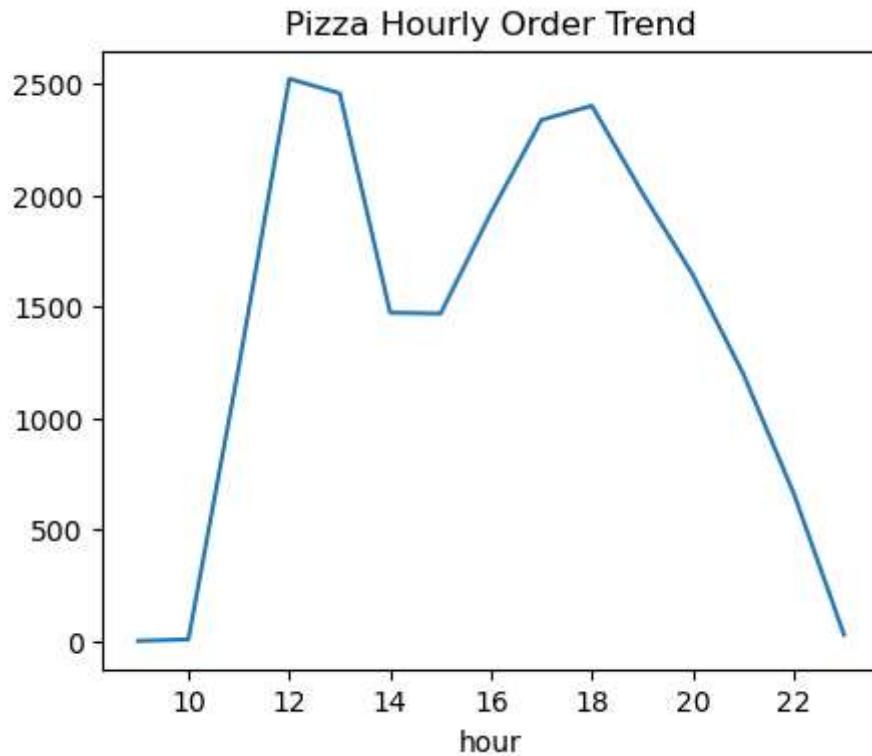
#group the data by hour
piz_hour_order = pizza.drop_duplicates("order_id").groupby("hour")["order_id"].count().reset_index(name="count").sort_v
piz_hour_order
```

Out[15]:

	hour	count
0	9	1
1	10	8
2	11	1231
3	12	2520
4	13	2455
5	14	1472
6	15	1468
7	16	1920
8	17	2336
9	18	2399
10	19	2009
11	20	1642
12	21	1198
13	22	663
14	23	28

In [16]:

```
#visualizations
plt.style.use("default")
piz_hour_order.plot(x="hour", title="Pizza Hourly Order Trend", figsize=(5,4))
plt.gca().get_legend().remove()
```



From the lineplot above, we see that the most Pizzas order occur at noon (12pm) and then by 1pm. It then peaks at 6pm towards the evening time

### 3. Percentage of Sales per Pizza Category:

```
In [17]: #Get Percentage of sales per Pizza category
prop = pizza.groupby("pizza_category")["total_price"].sum().sort_values(ascending=False).reset_index()
prop[%] = (100*prop["total_price"])/prop["total_price"].sum().round(2)
prop
```

```
Out[17]: pizza_category  total_price      %
```

	pizza_category	total_price	%
0	Classic	220053.10	26.91
1	Supreme	208197.00	25.46
2	Chicken	195919.50	23.96
3	Veggie	193690.45	23.68

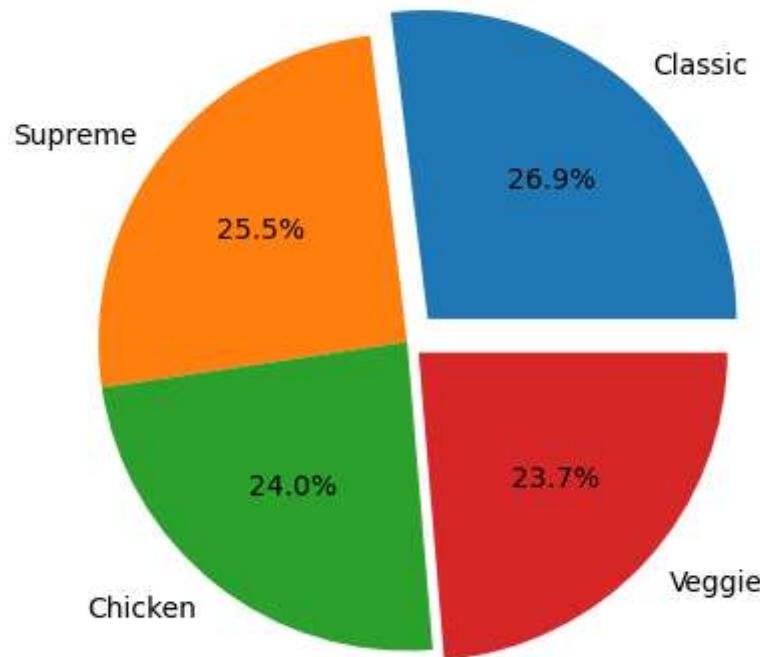
```
In [18]: #Visualization(Pie-chart)
```

```
explode= [0.1, 0, 0, 0.05]
labels = prop["pizza_category"]

plt.figure(figsize=(6,5))
plt.pie(prop["%"], autopct="%1.1f%%", labels=labels, explode = explode)
plt.title("Percentage of sales per Pizza category")

plt.show()
```

Percentage of sales per Pizza category



From the diagram above:

1. The **Classic** pizza category has the most sales by **26.9%**
2. followed by the **supreme** category with **25.5%**
3. **Veggie** pizza category with **24%**
4. And lastly, **chicken** Pizza with **23.7%**

#### 4. Percentage of Sales by Pizza size:

```
In [19]: #Get the percentage of sales by size  
prop1 = pizza.groupby("pizza_size")["total_price"].sum().reset_index()
```

```
prop1["%"] = (100 * prop1["total_price"]/prop1["total_price"].sum()).round(2)
prop1
```

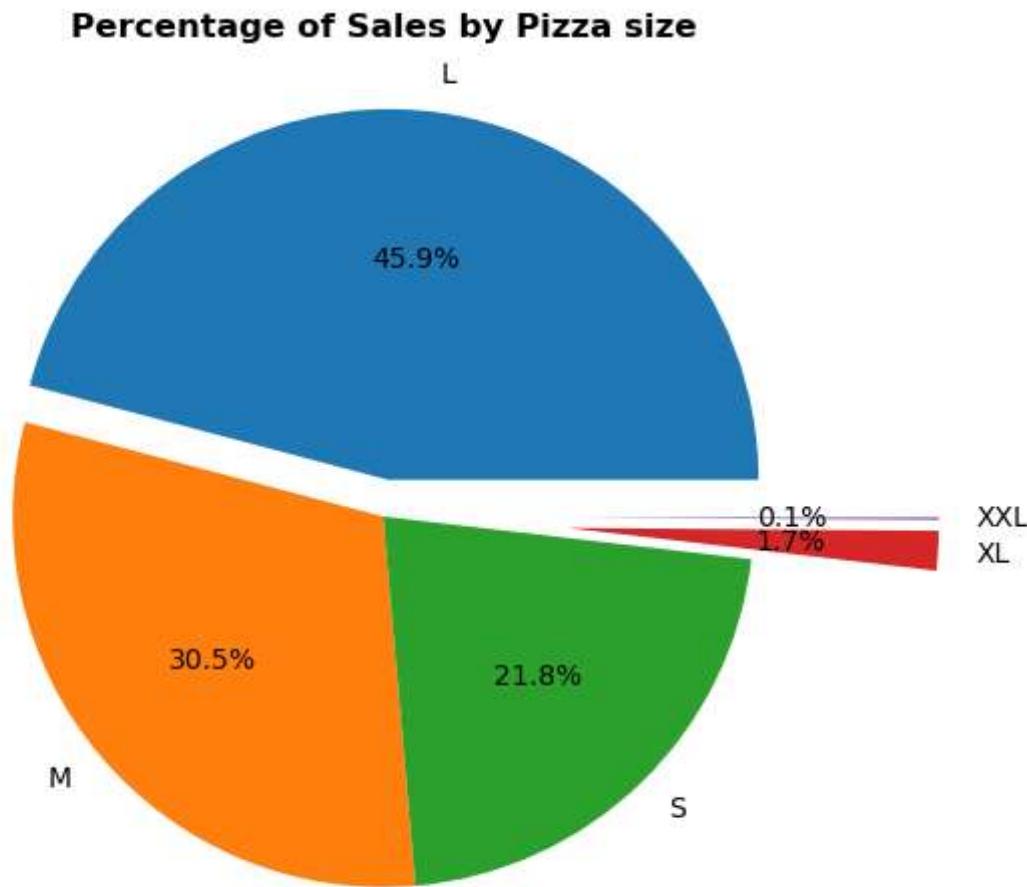
Out[19]:

	pizza_size	total_price	%
0	L	375318.70	45.89
1	M	249382.25	30.49
2	S	178076.50	21.77
3	XL	14076.00	1.72
4	XXL	1006.60	0.12

In [20]:

```
#Visualiixation(pie-chart)
labels = ["L","M","S","XL","XXL"]
explode = [0.1,0,0,0.5,0.5]
plt.figure(figsize=(8,6))
plt.pie(prop1["%"], labels=labels, explode=explode, autopct="%1.1f%%")
plt.title("Percentage of Sales by Pizza size", weight="bold")

plt.show()
```



From the diagram above:

1. The **Large(L)** pizza size has the most sales by **45.9%**
2. followed by the **Medium(M)** pizza size with **30.5%**
3. Then, **Small(S)** pizza size with **21.8%**
4. Then, **Extra-Large(XL)** pizza size with **1.7%**
5. And, Lastly **Extra-Extra-Large(XXL)** pizza size has the least sales with **0.1%**

## 5. Total Pizzas by category:

```
In [29]: #Get the total number of pizzas sold for each category
    piz = pizza.groupby("pizza_category")["quantity"].sum().reset_index(name="count").sort_values("count", ascending=False)
    piz
```

```
Out[29]:   pizza_category  count
1           Classic    14888
2          Supreme    11987
3         Veggie     11649
0        Chicken    11050
```

```
In [30]: #visualization[funnel chart]
    fig = px.funnel(piz, x = "count", y = "pizza_category")
    fig
```

## 6 .Top 5 best sellers by Revenue, Total Quantity and Total Orders:

```
In [23]: #Revenue  
piz_top_rev = pizza.groupby("pizza_name")["total_price"].sum().sort_values(ascending=False).head().reset_index()  
piz_top_rev
```

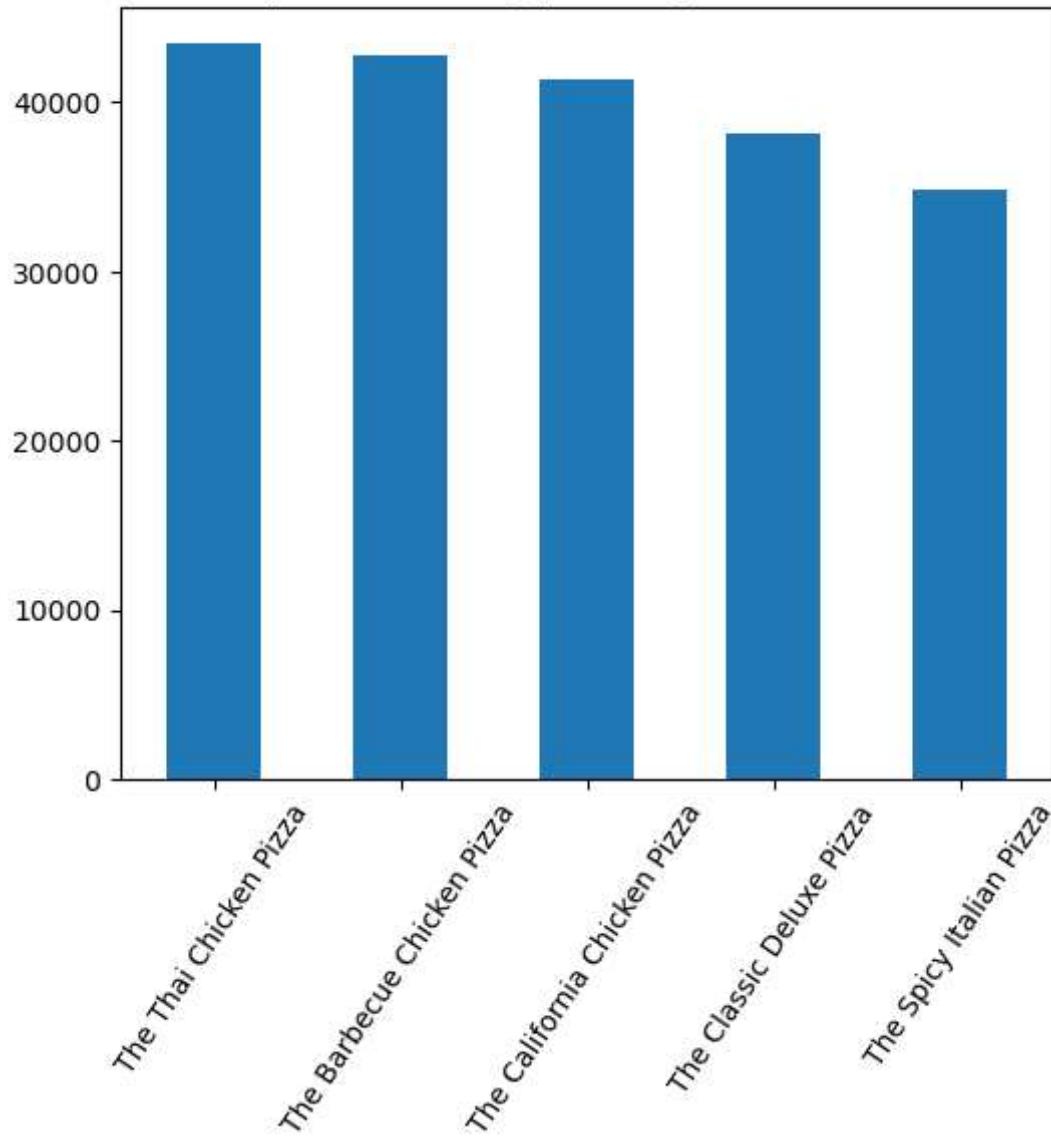
Out[23]:

	pizza_name	total_price
0	The Thai Chicken Pizza	43434.25
1	The Barbecue Chicken Pizza	42768.00
2	The California Chicken Pizza	41409.50
3	The Classic Deluxe Pizza	38180.50
4	The Spicy Italian Pizza	34831.25

In [24]:

```
#Revenue visualization(top 5)
piz_top_rev.plot(kind="bar",x="pizza_name",rot=56,figsize=(6,5),
                  title="Top 5 Best Selling pizzas by Total Revenue",xlabel="")
plt.gca().get_legend().remove()
```

Top 5 Best Selling pizzas by Total Revenue



```
In [25]: #Total Quantity
pizza_top_qty = pizza.groupby("pizza_name")["quantity"].sum().sort_values(ascending=False).head().reset_index()
pizza_top_qty
```

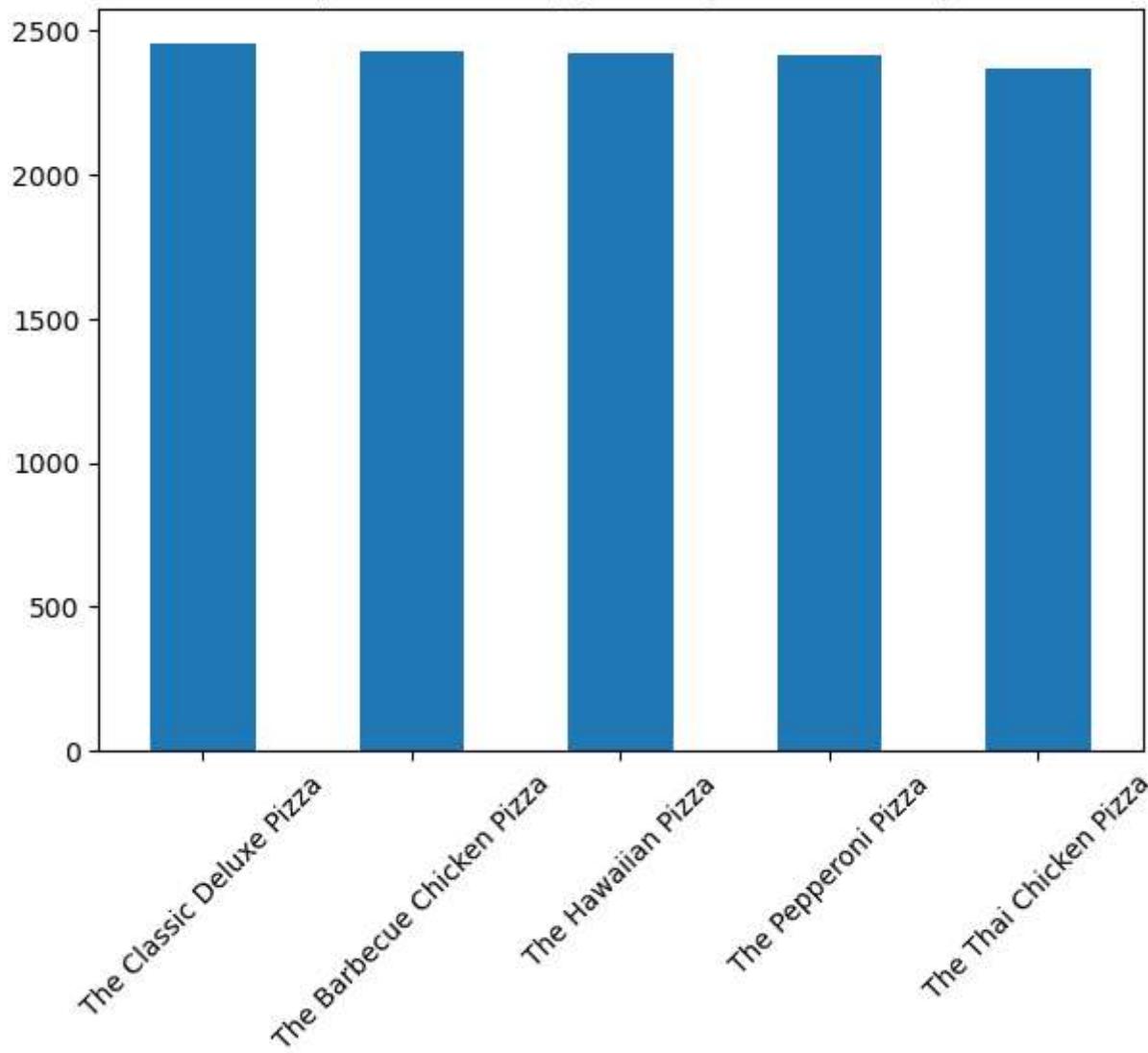
Out[25]:

	pizza_name	quantity
0	The Classic Deluxe Pizza	2453
1	The Barbecue Chicken Pizza	2432
2	The Hawaiian Pizza	2422
3	The Pepperoni Pizza	2418
4	The Thai Chicken Pizza	2371

In [26]:

```
#visualization(top 5 by total quantity)
piz_top_qty.plot(kind="bar",x = "pizza_name",rot=45,xlabel="",figsize=(7,5),
                  title="Top 5 best selling pizzas by Total Quantity")
plt.gca().get_legend().remove()
```

Top 5 best selling pizzas by Total Quantity



```
In [27]: #Total orders
pizza.drop_duplicates(["order_id", "pizza_name"]).groupby("pizza_name").count()["pizza_id"].sort_values
pizza_top_orders
```

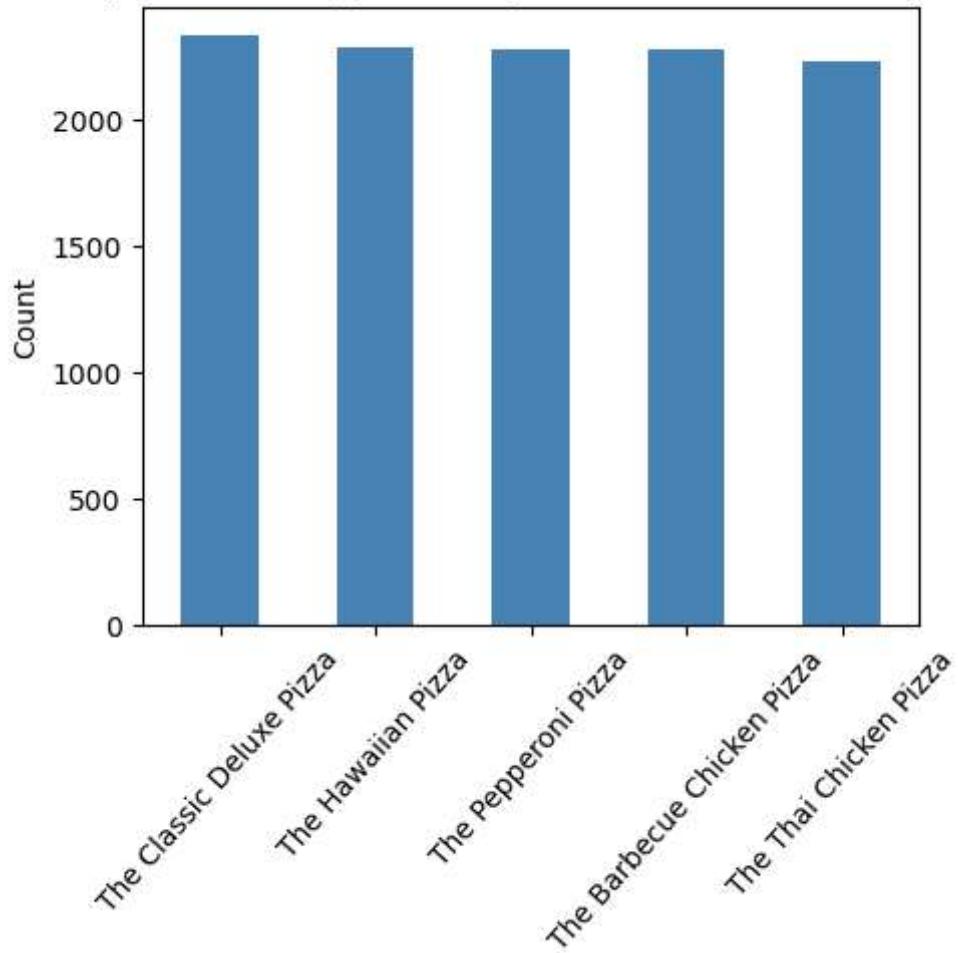
Out[27]:

	pizza_name	count
0	The Classic Deluxe Pizza	2329
1	The Hawaiian Pizza	2280
2	The Pepperoni Pizza	2278
3	The Barbecue Chicken Pizza	2273
4	The Thai Chicken Pizza	2225

In [40]: *#visualization*

```
piz_top_orders.plot(kind="bar",x="pizza_name", rot=45, figsize=(5,4), ylabel="Count",
                     title="Top 5 best selling Pizzas by Total Orders", color="steelblue", xlabel="")
plt.gca().get_legend().remove()
plt.show()
```

Top 5 best selling Pizzas by Total number of Pizzas Sold



Clearly from the above figure, `The Classic Deluxe Pizza` has the most number of Pizzas sold. Therefore it is the most selling pizza

From the bar Chart above, we can see that the least selling pizza is `The Brie Carre Pizza` wth just 480 pizzas sold in the year 2015

#### 7. Bottom 5 worst sellers by Revenue, Total Quantity and total orders:

```
In [31]: #Revenue  
piz_bottom_rev = pizza.groupby("pizza_name")["total_price"].sum().sort_values().head().reset_index()
```

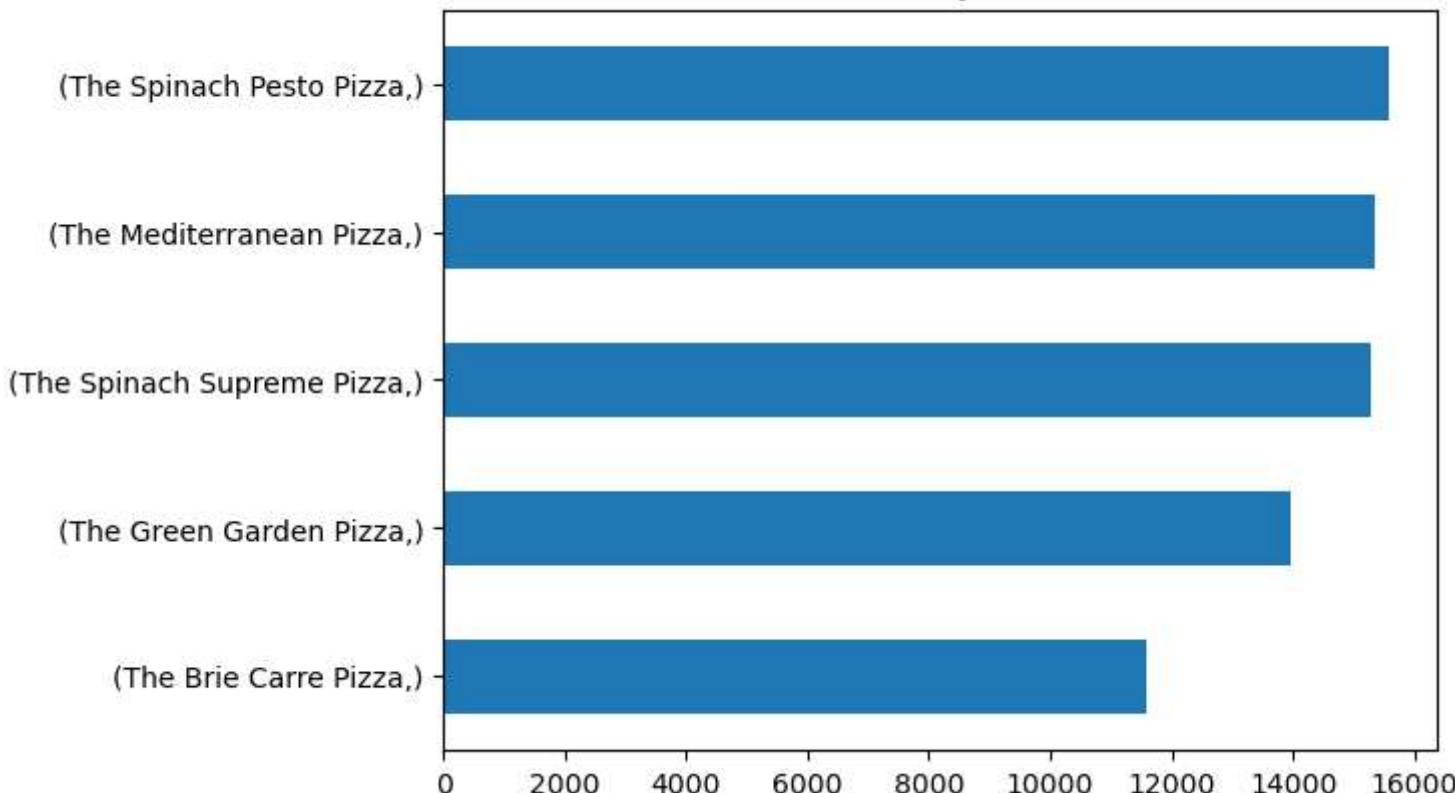
```
piz_bottom_rev
```

```
Out[31]:
```

	pizza_name	total_price
0	The Brie Carre Pizza	11588.50
1	The Green Garden Pizza	13955.75
2	The Spinach Supreme Pizza	15277.75
3	The Mediterranean Pizza	15360.50
4	The Spinach Pesto Pizza	15596.00

```
In [32]: #visualization
x= [piz_bottom_rev["pizza_name"]]
piz_bottom_rev.index = x
piz_bottom_rev.plot(kind="barh", title="Bottom 5 Pizzas by Total Revenue", xlabel="")
plt.gca().get_legend().remove()
```

Bottom 5 Pizzas by Total Revenue



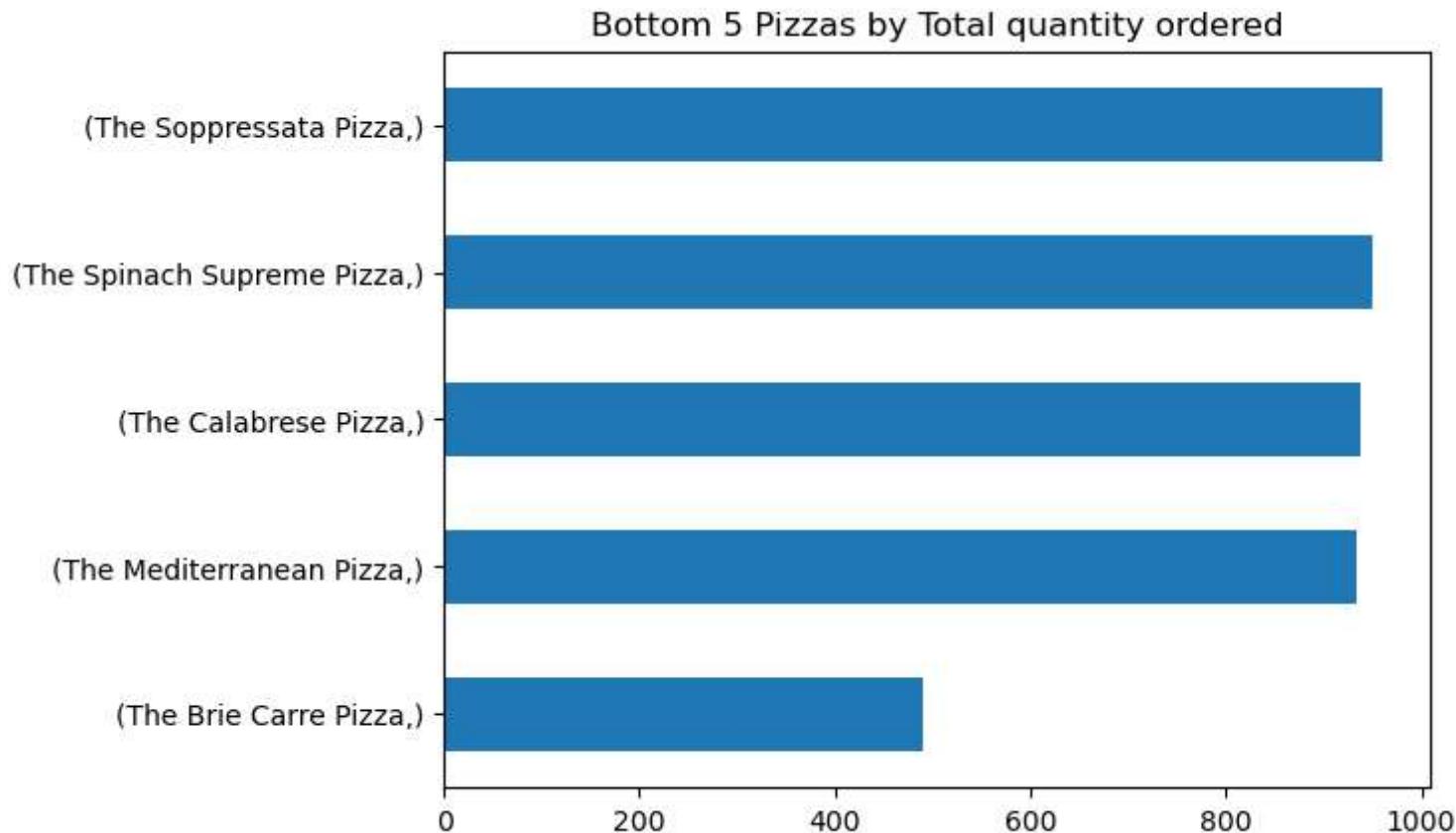
```
In [33]: #Total quantity
pizza_bottom_qty = pizza.groupby("pizza_name")["quantity"].sum().sort_values().head().reset_index()
pizza_bottom_qty
```

```
Out[33]:
```

	pizza_name	quantity
0	The Brie Carre Pizza	490
1	The Mediterranean Pizza	934
2	The Calabrese Pizza	937
3	The Spinach Supreme Pizza	950
4	The Soppressata Pizza	961

```
In [34]:
```

```
#visualization
x= [piz_bottom_qty["pizza_name"]]
piz_bottom_qty.index = x
piz_bottom_qty.plot(kind="barh", title="Bottom 5 Pizzas by Total quantity ordered", xlabel="")
plt.gca().get_legend().remove()
```



```
In [35]:
```

```
#Total orders
piz_bottom_orders = pizza.drop_duplicates(["order_id","pizza_name"]).groupby("pizza_name").count()["pizza_id"].sort_val
piz_bottom_orders
```

Out[35]:

	pizza_name	order_count
0	The Brie Carre Pizza	480
1	The Mediterranean Pizza	912
2	The Calabrese Pizza	918
3	The Spinach Supreme Pizza	918
4	The Chicken Pesto Pizza	938

In [36]:

```
#visualization
x= [piz_bottom_orders["pizza_name"]]
piz_bottom_orders.index = x
piz_bottom_orders.plot(kind="barh", title="Bottom 5 Pizzas by Total orders", xlabel="")
plt.gca().get_legend().remove()
```

