Pizza Restaurant Sales Report

Data Analysis with Python

By

ROHIT KUMAR MAURYA

SHWETA VISHWAKARMA

AND TEAM

Accenture Training DA Phase 2 Batch2 (Group-8)

Problem Statement

1. What days and times do we tend to be busiest?

2. How many pizzas are we making during peak periods?

3. What are our best and worst-selling pizzas?

4. What's our average order value?

About Dataset

Contents:

This pizza sales dataset makes up 12 relevant features:

- **pizza_id:** The unique identifier for each pizza in the dataset.
- **order_id:** The unique identifier for each pizza order.
- **pizza_name_id**: The identifier for each specific pizza name.
- **quantity:** The number of pizzas ordered in each transaction.
- **order_date:** The date when the pizza order was placed.

- **order_time:** The time at which the pizza order was placed.
- **unit_price:** The price of a single unit of pizza.
- **total_price:** The total price of the pizza order, calculated as the unit price multiplied by the quantity.
- **pizza_size:** The size or dimensions of the pizza. (S,M,L,XL,XXL)
- **pizza_category:** The category or classification of the pizza, indicating its type or style. (Classic, Veggie, Supreme, Chicken)
- **pizza_ingredients:** The list of ingredients used in the pizza preparation.
- **pizza_name:** The name or label assigned to each specific pizza.

Exploratory Data Analysis and Data Cleaning

Top -5 data of dataset:

```
#view first five rows/records
Pizza_df.head()
```

	order_details_id	order_id	pizza_id	quantity	order_date	order_time	unit_price	total_price	pizza_size	pizza_category	pizza_ingredients	pizza_name
0	1	1	hawaiian_m	1	2015-01-01	11:38:36	13.25	13.25	M	Classic	Sliced Ham, Pineapple, Mozzarella Cheese	The Hawaiian Pizza
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,...	The Classic Deluxe Pizza
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...	The Five Cheese Pizza
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...	The Italian Supreme Pizza
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...	The Mexicana Pizza

Bottom-5 data of dataset:

```
#view last five rows/records
Pizza_df.tail()
```

	order_details_id	order_id	pizza_id	quantity	order_date	order_time	unit_price	total_price	pizza_size	pizza_category	pizza_ingredients	pizza_name
48615	48616	21348	ckn_alfredo_m	1	2015-12-31	21:23:10	16.75	16.75	M	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms, A...	The Chicken Alfredo Pizza
48616	48617	21348	four_cheese_l	1	2015-12-31	21:23:10	17.95	17.95	L	Veggie	Ricotta Cheese, Gorgonzola Piccante Cheese, Mo...	The Four Cheese Pizza
48617	48618	21348	napolitana_s	1	2015-12-31	21:23:10	12.00	12.00	S	Classic	Tomatoes, Anchovies, Green Olives, Red Onions,...	The Napolitana Pizza
48618	48619	21349	mexicana_l	1	2015-12-31	22:09:54	20.25	20.25	L	Veggie	Tomatoes, Red Peppers, Jalapeno Peppers, Red O...	The Mexicana Pizza
48619	48620	21350	bbq_ckn_s	1	2015-12-31	23:02:05	12.75	12.75	S	Chicken	Barbecued Chicken, Red Peppers, Green Peppers,...	The Barbecue Chicken Pizza

Total no of rows and columns in the dataset:

```
#total no. of rows & columns
print("No of rows and columns in the dataset:")
Pizza_df.shape
```

```
No of rows and columns in the dataset:
(48620, 18)
```

Information about the dataset with its datatype:

```
# to find column details with nulls
Pizza_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48620 entries, 0 to 48619
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype         
---  ------             --------------  -----         
 0   order_details_id   48620 non-null  int64         
 1   order_id           48620 non-null  int64         
 2   pizza_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
```

Summary of data from dataset:

```
# TO summarize the dataframe on
Pizza_df.describe()
```

	order_details_id	order_id	quantity	order_date	unit_price	total_price
count	48620.000000	48620.000000	48620.000000	48620	48620.000000	48620.000000
mean	24310.500000	10701.479761	1.019622	2015-06-29 11:03:43.611682560	16.494132	16.821474
min	1.000000	1.000000	1.000000	2015-01-01 00:00:00	9.750000	9.750000
25%	12155.750000	5337.000000	1.000000	2015-03-31 00:00:00	12.750000	12.750000
50%	24310.500000	10682.500000	1.000000	2015-06-28 00:00:00	16.500000	16.500000
75%	36465.250000	16100.000000	1.000000	2015-09-28 00:00:00	20.250000	20.500000
max	48620.000000	21350.000000	4.000000	2015-12-31 00:00:00	35.950000	83.000000
std	14035.529381	6180.119770	0.143077	NaN	3.621789	4.437398

```
Pizza_df.describe(include='object')
```

	pizza_id	order_time	pizza_size	pizza_category	pizza_ingredients	pizza_name
count	48620	48620	48620	48620	48620	48620
unique	91	16382	5	4	32	32
top	big_meat_s	12:32:00	L	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppers,...	The Classic Deluxe Pizza
freq	1811	26	18526	14579	2416	2416

Checking null value in dataset:

```
Pizza_df.isna().sum()
```

```
order_details_id     0
order_id             0
pizza_id             0
quantity             0
order_date           0
order_time           0
unit_price           0
total_price          0
pizza_size           0
pizza_category       0
pizza_ingredients    0
pizza_name           0
dtype: int64
```

Data Analysis and Visualization

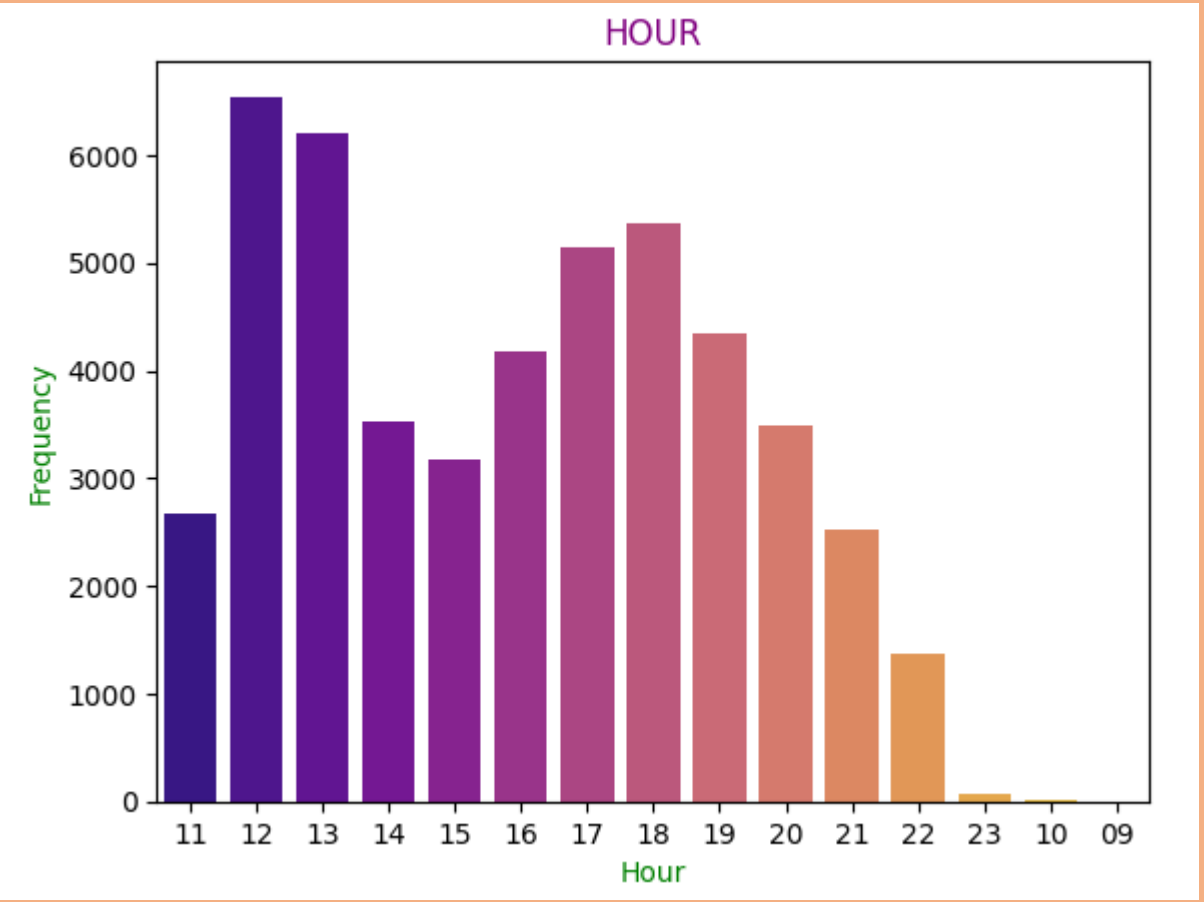
Extracting Hour form ‘order time’ column in the dataset:

```
Pizza_df['order_time']=Pizza_df['order_time'].astype('string')
Pizza_df[['Hour','Minute','Second']] = Pizza_df['order_time'].str.split(":", expand=True)
Pizza_df["Hour"].value_counts()
```

```
Hour
12    6543
13    6203
18    5359
17    5143
19    4350
16    4185
14    3521
20    3487
15    3170
11    2672
21    2528
22    1370
23      68
10      17
09       4
Name: count, dtype: Int64
```

- **Plotting hour wise:**

```
sns.countplot(data=Pizza_df, x="Hour", palette="plasma")
plt.xlabel("Hour",fontsize=10, color='green')
plt.ylabel("Frequency",fontsize=10,color="green")
plt.title("HOUR", color='Purple')
plt.show()
```



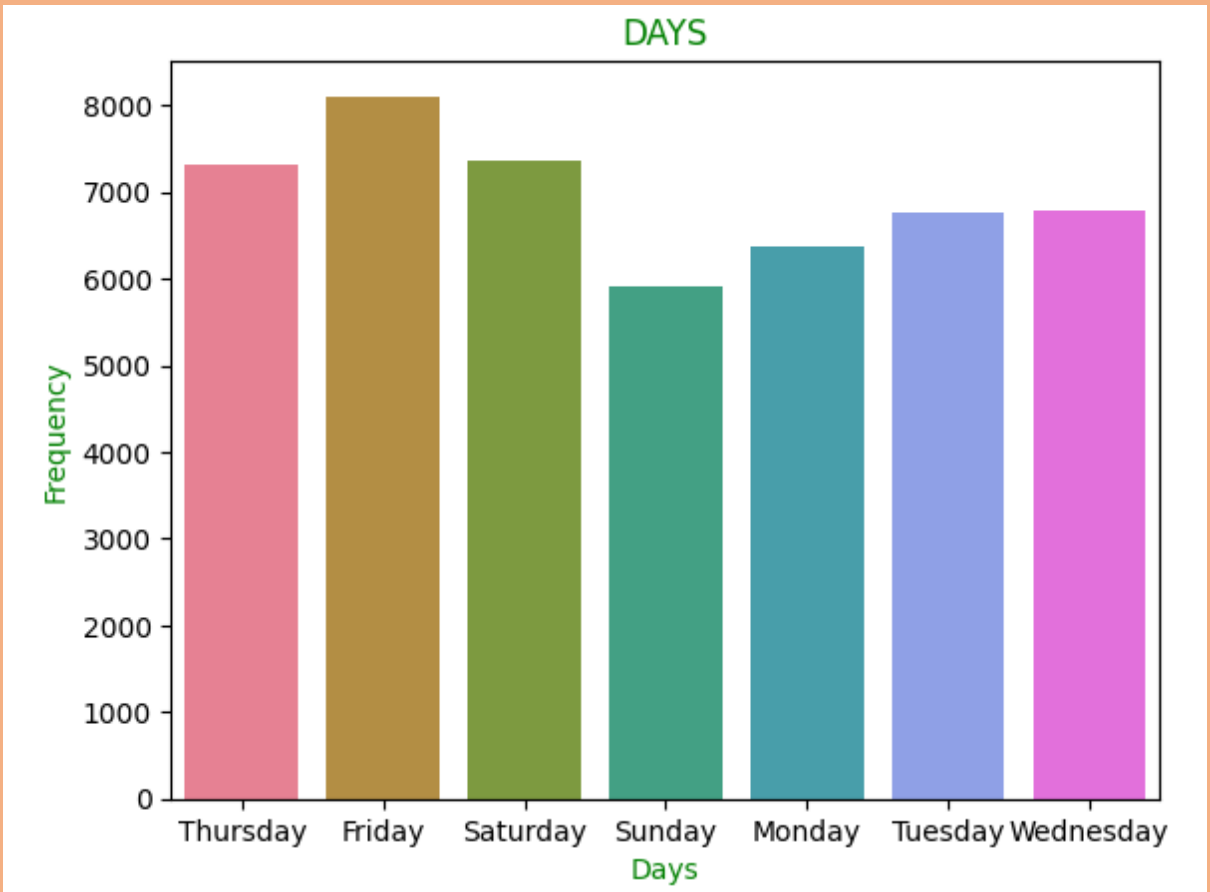
Extracting Days from order_date column in the dataset:

```
Pizza_df['order_dates'] = Pizza_df['order_date'].dt.day_name()
Pizza_df['order_dates'].value_counts()
```

```
order_dates
Friday       8106
Saturday     7355
Thursday     7323
Wednesday    6797
Tuesday      6753
Monday       6369
Sunday       5917
Name: count, dtype: int64
```

- **Plotting Day wise:**

```
sns.countplot(data = Pizza_df, x = 'order_dates', palette ="husl")
plt.xlabel('Days', fontsize=10, color="green")
plt.ylabel('Frequency', fontsize=10, color="green")
plt.title("DAYS", color="green")
plt.show()
```



- **Extracting the month from the 'order_month' column in the dataset:**

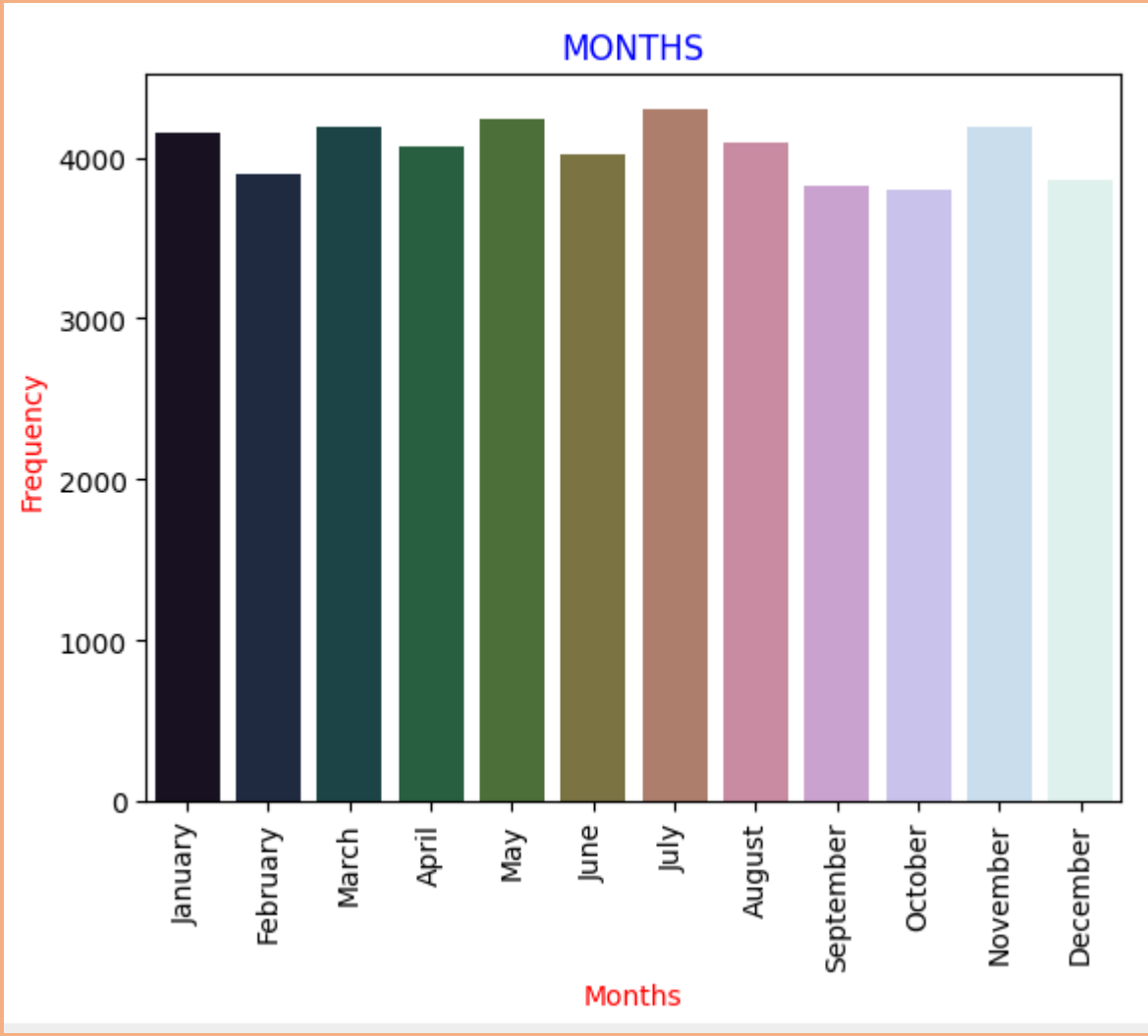
```
Pizza_df['order_month'] = pd.DatetimeIndex (Pizza_df['order_date']).month
Pizza_df.loc[(Pizza_df['order_month']==1), 'order_month'] = 'January'
Pizza_df.loc[(Pizza_df['order_month']==2), 'order_month'] = 'February'
Pizza_df.loc[(Pizza_df['order_month']==3), 'order_month'] = 'March'
Pizza_df.loc[(Pizza_df['order_month']==4), 'order_month'] = 'April'
Pizza_df.loc[(Pizza_df['order_month']==5), 'order_month'] = 'May'
Pizza_df.loc[(Pizza_df['order_month']==6), 'order_month'] = 'June'
Pizza_df.loc[(Pizza_df['order_month']==7), 'order_month'] = 'July'
Pizza_df.loc[(Pizza_df['order_month']==8), 'order_month'] = 'August'
Pizza_df.loc[(Pizza_df['order_month']==9), 'order_month'] = 'September'
Pizza_df.loc[(Pizza_df['order_month']==10), 'order_month'] = 'October'
Pizza_df.loc[(Pizza_df['order_month']==11), 'order_month'] = 'November'
Pizza_df.loc[(Pizza_df['order_month']==12), 'order_month'] = 'December'

Pizza_df['order_month'].value_counts()
```

```
order_month
July          4301
May           4239
March         4186
November      4185
January       4156
August        4094
April         4067
June          4025
February      3892
December      3859
September     3819
October       3797
Name: count, dtype: int64
```

- **Plotting Month Wise:**

```
sns.countplot(data=Pizza_df, x='order_month', palette='cubehelix')
plt.xticks(rotation=90)
plt.xlabel("Months", fontsize=10, color="red")
plt.ylabel("Frequency", fontsize=10, color="red")
plt.title("MONTHS", color='blue')
plt.show()
```



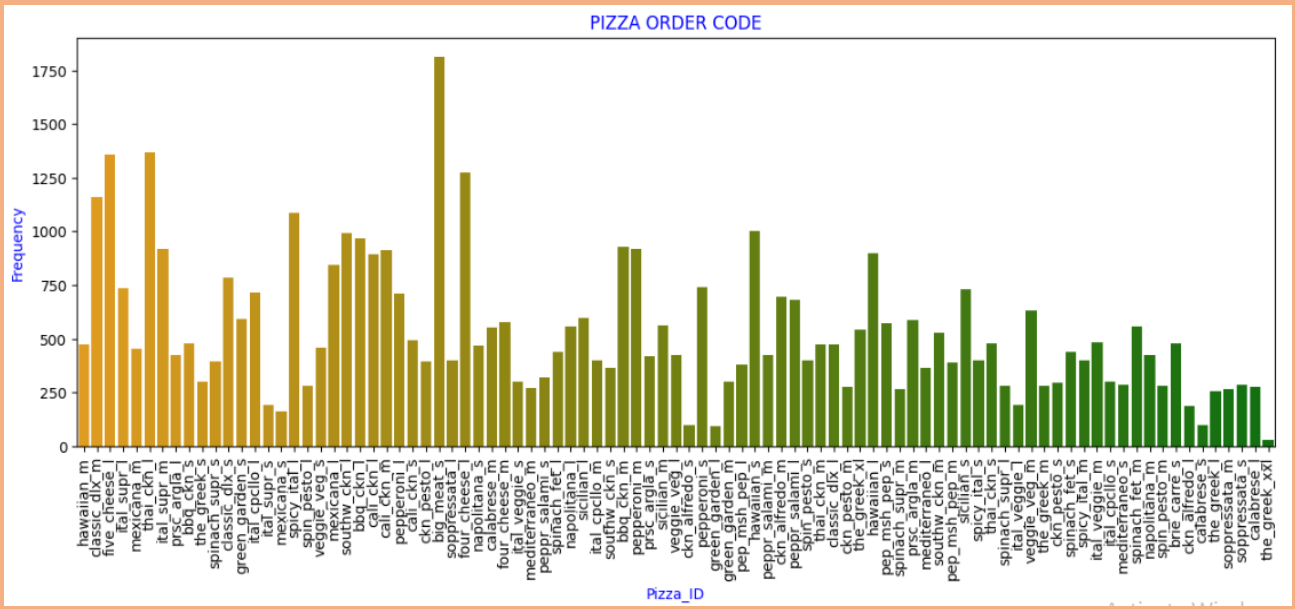
Counting the pizza_id value:

```
Pizza_df.pizza_id.value_counts()
```

```
pizza_id
big_meat_s        1811
thai_ckn_l        1365
five_cheese_l     1359
four_cheese_l     1273
classic_dlx_m     1159
                  ...
mexicana_s         160
calabrese_s         99
ckn_alfredo_s       96
green_garden_l      94
the_greek_xxl       28
Name: count, Length: 91, dtype: int64
```

- **Pizza_ID wise Plotting:**

```
fig, ax = plt.subplots(figsize=(15,5))
sns.countplot(data=Pizza_df,x='pizza_id',palette='blend:orange,green')
plt.xticks(rotation=90)
plt.xlabel('Pizza_ID',fontsize=10,color='blue')
plt.ylabel('Frequency',fontsize=10,color='blue')
plt.title('PIZZA ORDER CODE', color='blue')
plt.show()
```



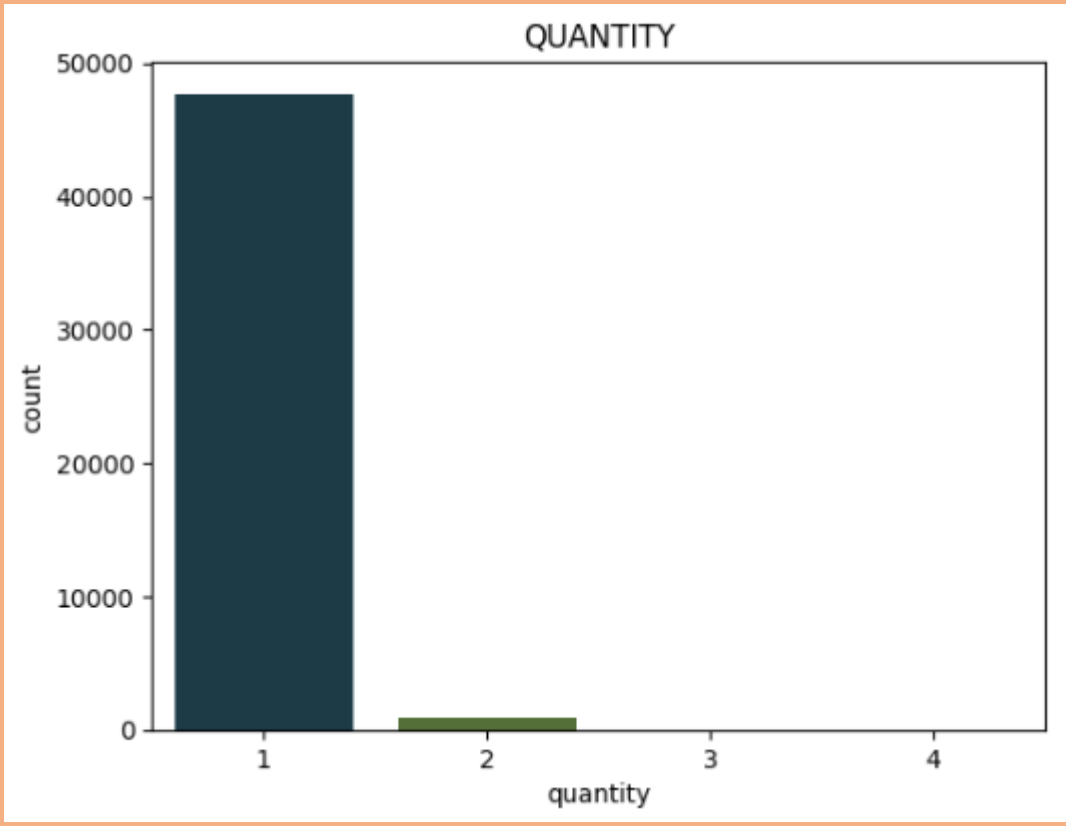
Counting the quantity value:

```
Pizza_df.quantity.value_counts()
```

```
quantity
1    47693
2      903
3       21
4        3
Name: count, dtype: int64
```

- **Quantity wise Plotting:**

```
sns.countplot(data=Pizza_df,x='quantity',palette='cubehelix')
plt.xlabel('quantity', fontsize=10, color='black')
plt.title('QUANTITY',color='black')
plt.show()
```



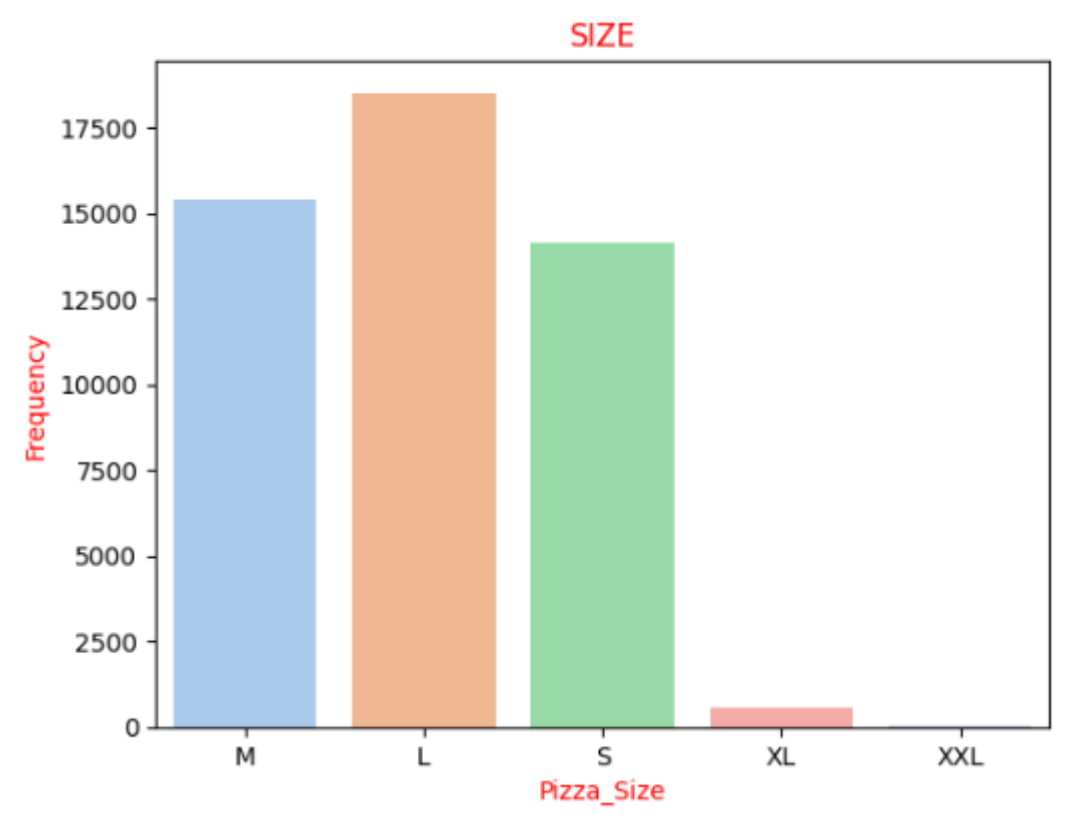
- **Counting the pizza_size value:**

```
Pizza_df.pizza_size.value_counts()
```

```
pizza_size
L      18526
M      15385
S      14137
XL       544
XXL       28
Name: count, dtype: int64
```

- **Pizza Size wise Plotting:**

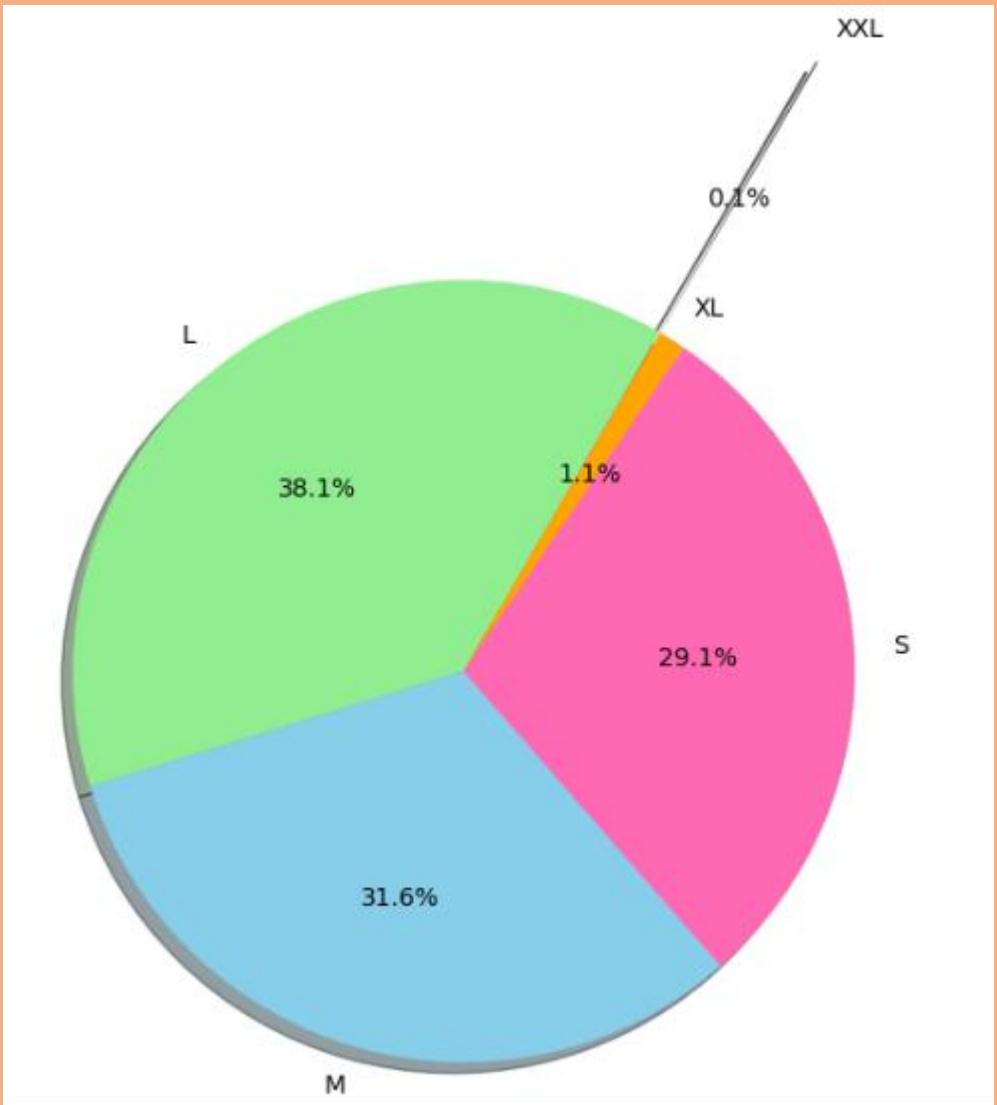
```
sns.countplot(data=Pizza_df, x='pizza_size', palette='pastel')
plt.xlabel('Pizza_Size',fontsize=10,color='red')
plt.ylabel('Frequency',fontsize=10,color='red')
plt.title('SIZE', color='red')
plt.show()
```



- **Pizza Size wise Pie Plotting:**

```
labels = Pizza_df['pizza_size'].value_counts().index
sizes = Pizza_df['pizza_size'].value_counts()
plt.figure(figsize= (8,10))

colors = ['lightgreen','skyblue','hotpink','orange','black']
plt.pie(sizes, labels=labels, rotatelabels=False, autopct='%1.1f%%', colors=colors, shadow=True,startangle=60,explode=(0,0,0,0,0.8))
plt.show()
```



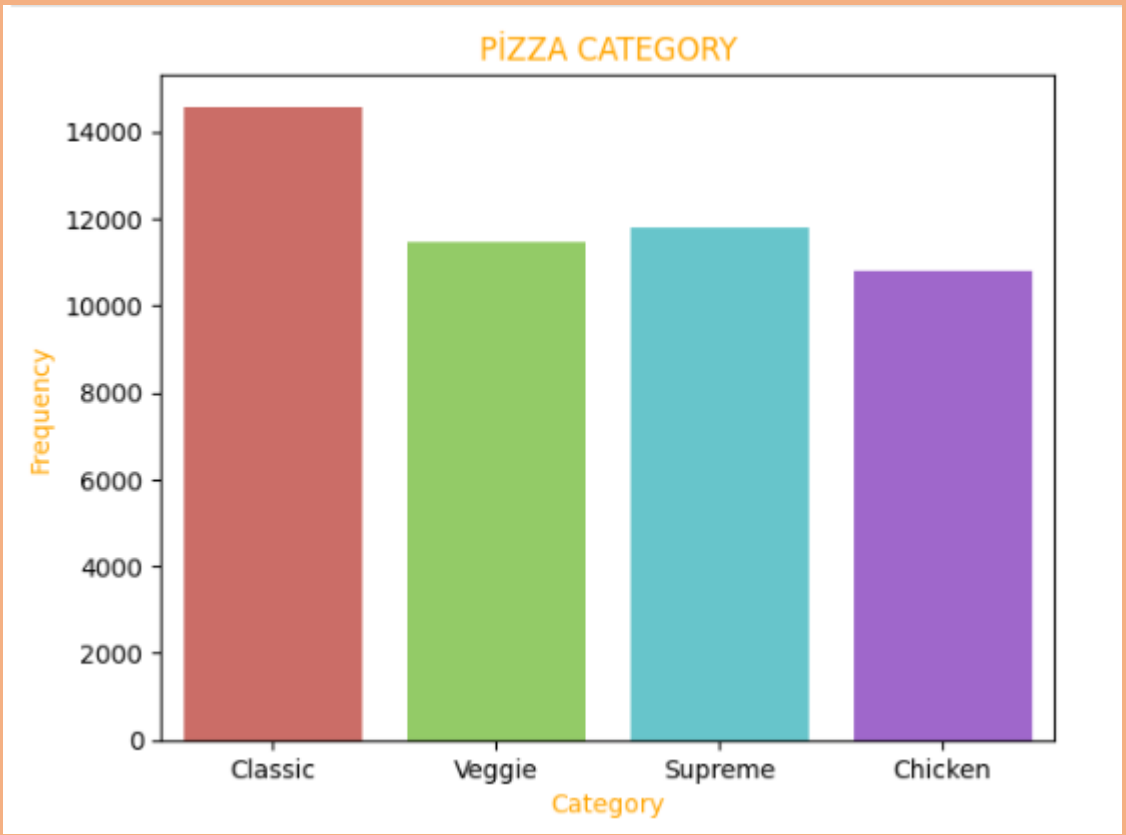
Counting the pizza_size value:

```
Pizza_df.pizza_category.value_counts()
```

```
pizza_category
Classic     14579
Supreme     11777
Veggie      11449
Chicken     10815
Name: count, dtype: int64
```

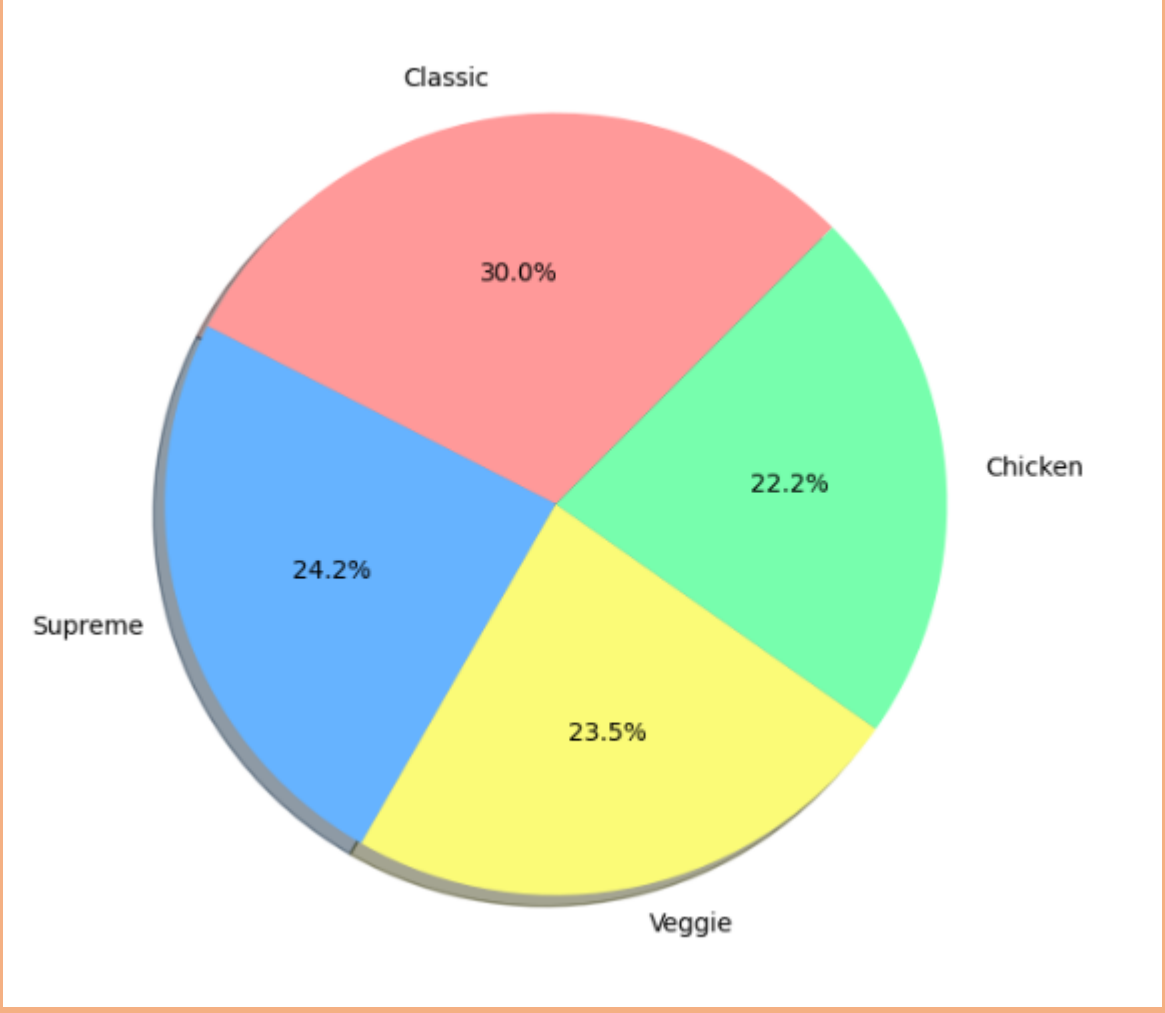
- **Pizza Category wise Plotting:**

```
sns.countplot(data=Pizza_df,x="pizza_category",palette="hls")
plt.xlabel("Category",fontsize=10,color="orange")
plt.ylabel("Frequency",fontsize=10,color="orange")
plt.title("PİZZA CATEGORY",color="orange")
plt.show()
```



- **Pizza Category wise Pie Plotting:**

```
labels = Pizza_df["pizza_category"].value_counts().index
sizes = Pizza_df["pizza_category"].value_counts()
colors = ['#ff9999','#66b3ff','#990f99','#ffcc89',"pink","yellow"]
plt.pie(sizes, labels=labels, rotatelabels=False, autopct='%1.1f%%',colors=colors,shadow=True, startangle=45)
plt.show()
```



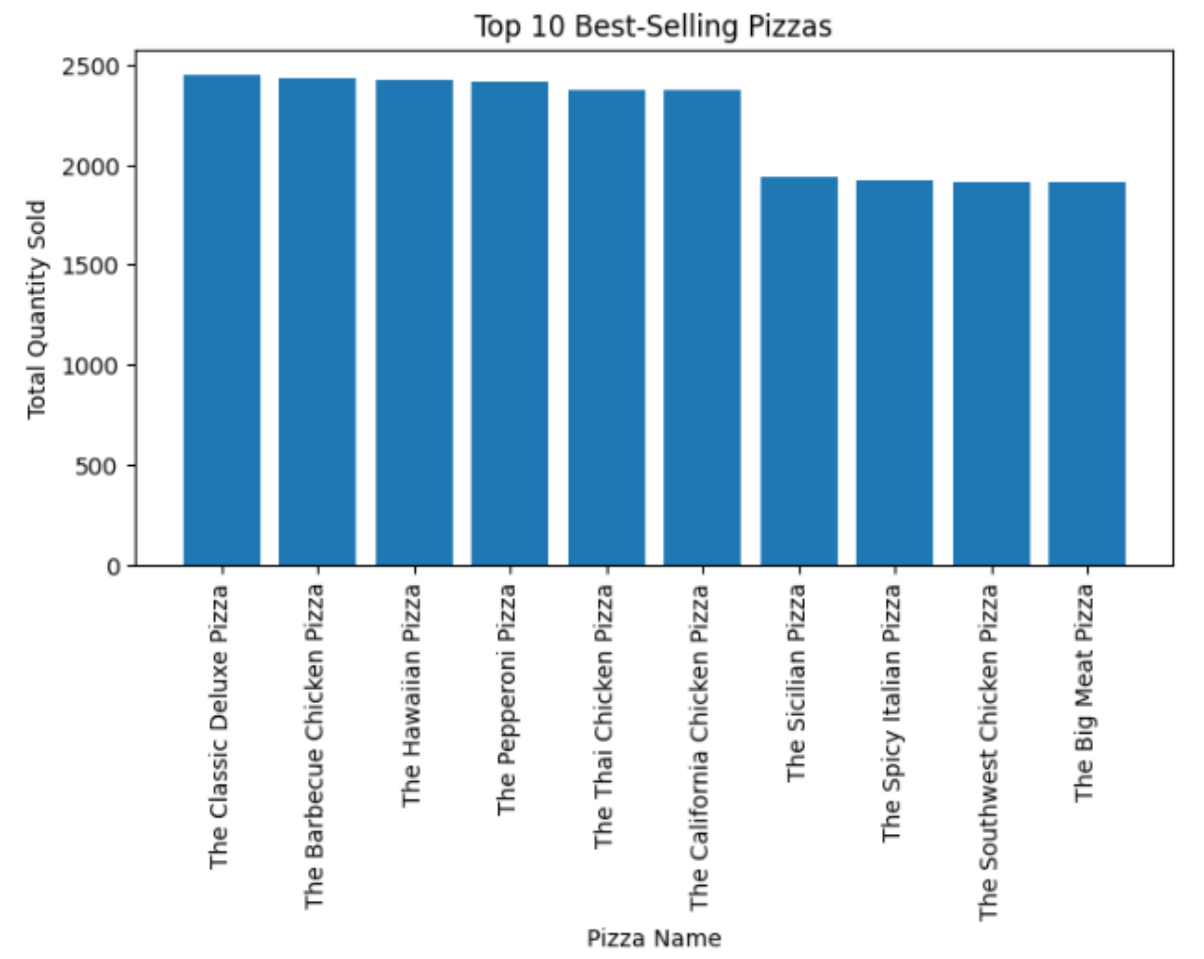
- **Top-10 best-selling pizzas:**

```
# Group by 'pizza_name' and calculate the total quantity sold
pizza_sales = Pizza_df.groupby('pizza_name')['quantity'].sum().sort_values(ascending=False)

top_n = 10  # Number of top-selling and worst-selling pizzas to display
plt.figure(figsize=(8, 4))

# Plot the top N best-selling pizzas
plt.bar(pizza_sales.head(top_n).index, pizza_sales.head(top_n))
plt.xlabel('Pizza Name')
plt.ylabel('Total Quantity Sold')
plt.title(f'Top {top_n} Best-Selling Pizzas')
plt.xticks(rotation=90)

plt.show()
```



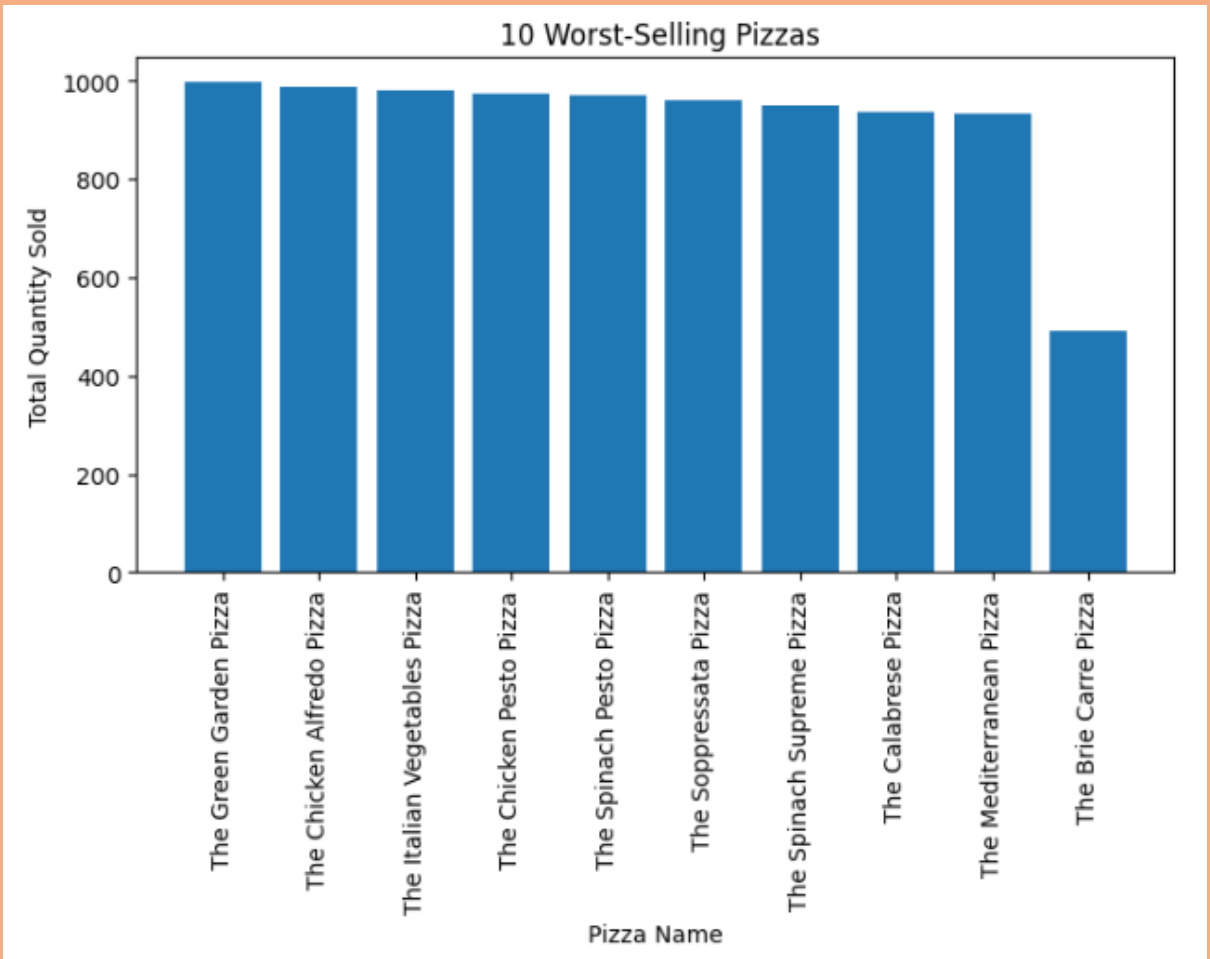
- **10 Worst-Selling Pizzas:**

```
# Group by 'pizza_name' and calculate the total quantity sold
pizza_sales = Pizza_df.groupby('pizza_name')['quantity'].sum().sort_values(ascending=False)

# Plotting
top_n = 10  # Number of top-selling and worst-selling pizzas to display
plt.figure(figsize=(12, 6))

# Plot the top N best-selling pizzas
plt.bar(pizza_sales.tail(top_n).index, pizza_sales.tail(top_n))
plt.xlabel('Pizza Name')
plt.ylabel('Total Quantity Sold')
plt.title(f' {top_n} Worst-Selling Pizzas')
plt.xticks(rotation=90)

plt.show()
```

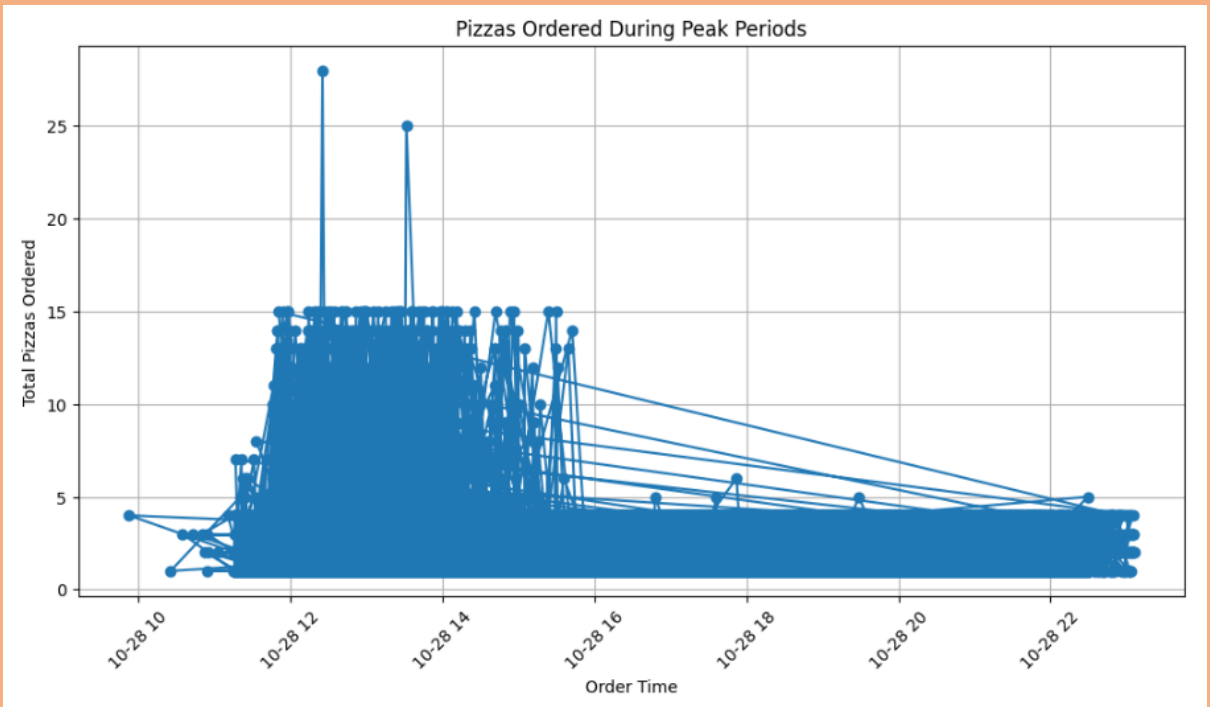


- **Pizzas Ordered During Peak Periods:**

```
# Convert 'order_time' to datetime
Pizza_df['order_time'] = pd.to_datetime(Pizza_df['order_time'])

# Group by date and time, and sum the quantity of pizzas
peak_period_data = Pizza_df.groupby(['order_date', 'order_time'])['quantity'].sum().reset_index()

# Plotting
plt.figure(figsize=(12, 6))
plt.plot(peak_period_data['order_time'], peak_period_data['quantity'], marker='o', linestyle='-')
plt.xlabel('Order Time')
plt.ylabel('Total Pizzas Ordered')
plt.title('Pizzas Ordered During Peak Periods')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



Solution of Problem Statement

1. According to the plotting we can see that the busiest day is Friday and the busiest time is 12 PM.

2. Approx 29 pizzas are made during peak periods.

3. Best selling pizza is 'The Classic Deluxe Pizza' and the worst selling pizza is 'The Brie Carre Pizza'.

4. The average order value is 16.821473673385437

Conclusion:

- Despite the pizza shop opening at 9, there are hardly any orders placed at that time.
- The Big Meat Pizza is a top choice among customers.
- The majority of customers opt for a single pizza.
- Large-sized pizzas are the most frequently ordered, while XXL-sized pizzas are the least popular.