🍕🍽 Pizza Restaurant Sales with Python

Problem Statement In this pizza sales analysis using Python, we dive deep into your pizza sales data to uncover valuable information and trends. Through the power of data analysis and visualization, we aim to answer crucial questions, optimize your operations, and enhance your decision-making.

Import Library

In [1]:
```
import pandas as pd
```

In [2]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import seaborn as sns
```

```
C:\Users\Syed Arif\anaconda3\lib\site-packages\scipy\__init__.py:146: UserWar
ning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of Sc
iPy (detected version 1.25.1
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

Uploading Csv fle

In [3]:
```
df = pd.read_excel(r"C:\Users\Syed Arif\Desktop\Data Model - Pizza Sales.xlsx"
```

Data Preprocessing

.head()

head is used show to the By default = 5 rows in the dataset

In [4]:
```
df.head()
```

Out[4]:

	order_details_id	order_id	pizza_id	quantity	order_date	order_time	unit_price	total_pric
0	1	1	hawaiian_m	1	2015-01-01	11:38:36	13.25	13.2
1	2	2	classic_dlx_m	1	2015-01-01	11:57:40	16.00	16.0
2	3	2	five_cheese_l	1	2015-01-01	11:57:40	18.50	18.5
3	4	2	ital_supr_l	1	2015-01-01	11:57:40	20.75	20.7
4	5	2	mexicana_m	1	2015-01-01	11:57:40	16.00	16.0

.tail()

tail is used to show rows by Descending order

In [5]:
```
df.tail()
```

Out[5]:

	order_details_id	order_id	pizza_id	quantity	order_date	order_time	unit_price	tota
48615	48616	21348	ckn_alfredo_m	1	2015-12-31	21:23:10	16.75	
48616	48617	21348	four_cheese_l	1	2015-12-31	21:23:10	17.95	
48617	48618	21348	napolitana_s	1	2015-12-31	21:23:10	12.00	
48618	48619	21349	mexicana_l	1	2015-12-31	22:09:54	20.25	
48619	48620	21350	bbq_ckn_s	1	2015-12-31	23:02:05	12.75	

.shape

It show the total no of rows & Column in the dataset

In [6]:
```
df.shape
```

Out[6]:
```
(48620, 12)
```

.Columns

It show the no of each Column

In [7]:
```
df.columns
```

Out[7]:
```
Index(['order_details_id', 'order_id', 'pizza_id', 'quantity', 'order_date',
       'order_time', 'unit_price', 'total_price', 'pizza_size',
       'pizza_category', 'pizza_ingredients', 'pizza_name'],
      dtype='object')
```

.dtypes

This Attribute show the data type of each column

In [8]:
```
df.dtypes
```

Out[8]:
```
order_details_id              int64
order_id                      int64
pizza_id                     object
quantity                      int64
order_date           datetime64[ns]
order_time                   object
unit_price                  float64
total_price                 float64
pizza_size                   object
pizza_category               object
pizza_ingredients            object
pizza_name                   object
dtype: object
```

.unique()

In a column, It show the unique value of specific column.

In [9]:
```
df["pizza_category"].unique()
```

Out[9]:
```
array(['Classic', 'Veggie', 'Supreme', 'Chicken'], dtype=object)
```

.nuique()

It will show the total no of unque value from whole data frame

In [10]:
```
df.nunique()
```

Out[10]:
```
order_details_id     48620
order_id             21350
pizza_id                91
quantity                 4
order_date             358
order_time           16382
unit_price              25
total_price             56
pizza_size               5
pizza_category           4
pizza_ingredients       32
pizza_name              32
dtype: int64
```

.describe()

It show the Count, mean , median etc

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

Out[11]:

	order_details_id	order_id	quantity	unit_price	total_price
count	48620.000000	48620.000000	48620.000000	48620.000000	48620.000000
mean	24310.500000	10701.479761	1.019622	16.494132	16.821474
std	14035.529381	6180.119770	0.143077	3.621789	4.437398
min	1.000000	1.000000	1.000000	9.750000	9.750000
25%	12155.750000	5337.000000	1.000000	12.750000	12.750000
50%	24310.500000	10682.500000	1.000000	16.500000	16.500000
75%	36465.250000	16100.000000	1.000000	20.250000	20.500000
max	48620.000000	21350.000000	4.000000	35.950000	83.000000

.value_counts

It Shows all the unique values with their count

In [13]:
```
df["pizza_category"].value_counts()
```

Out[13]:
```
Classic    14579
Supreme    11777
Veggie     11449
Chicken    10815
Name: pizza_category, dtype: int64
```

.isnull()

It shows the how many null values

In [14]:
```
df.isnull()
```

Out[14]:

	order_details_id	order_id	pizza_id	quantity	order_date	order_time	unit_price	total_pric
0	False	False	False	False	False	False	False	Fals
1	False	False	False	False	False	False	False	Fals
2	False	False	False	False	False	False	False	Fals
3	False	False	False	False	False	False	False	Fals
4	False	False	False	False	False	False	False	Fals
...	...	...	...	...	...	...	...	
48615	False	False	False	False	False	False	False	Fals
48616	False	False	False	False	False	False	False	Fals
48617	False	False	False	False	False	False	False	Fals
48618	False	False	False	False	False	False	False	Fals
48619	False	False	False	False	False	False	False	Fals

48620 rows × 12 columns

In [17]:
```
# 1. For each order, what is their total price?
order_total_price = df.groupby('order_id')['unit_price'].sum().reset_index()
print(order_total_price)
```

```
       order_id  unit_price
0             1       13.25
1             2       92.00
2             3       37.25
3             4       16.50
4             5       16.50
...         ...         ...
21345     21346       62.25
21346     21347       66.50
21347     21348       46.70
21348     21349       20.25
21349     21350       12.75

[21350 rows x 2 columns]
```

In [21]:
```
# 2. Which size of pizza is ordered the most?
most_ordered_size = df['pizza_size'].mode()
most_ordered_size
```

Out[21]:
```
0    L
Name: pizza_size, dtype: object
```

In [23]:
```
# 3. Which category of pizza is ordered the most?
most_ordered_category = df['pizza_category'].mode()
most_ordered_category
```

Out[23]:
```
0    Classic
Name: pizza_category, dtype: object
```

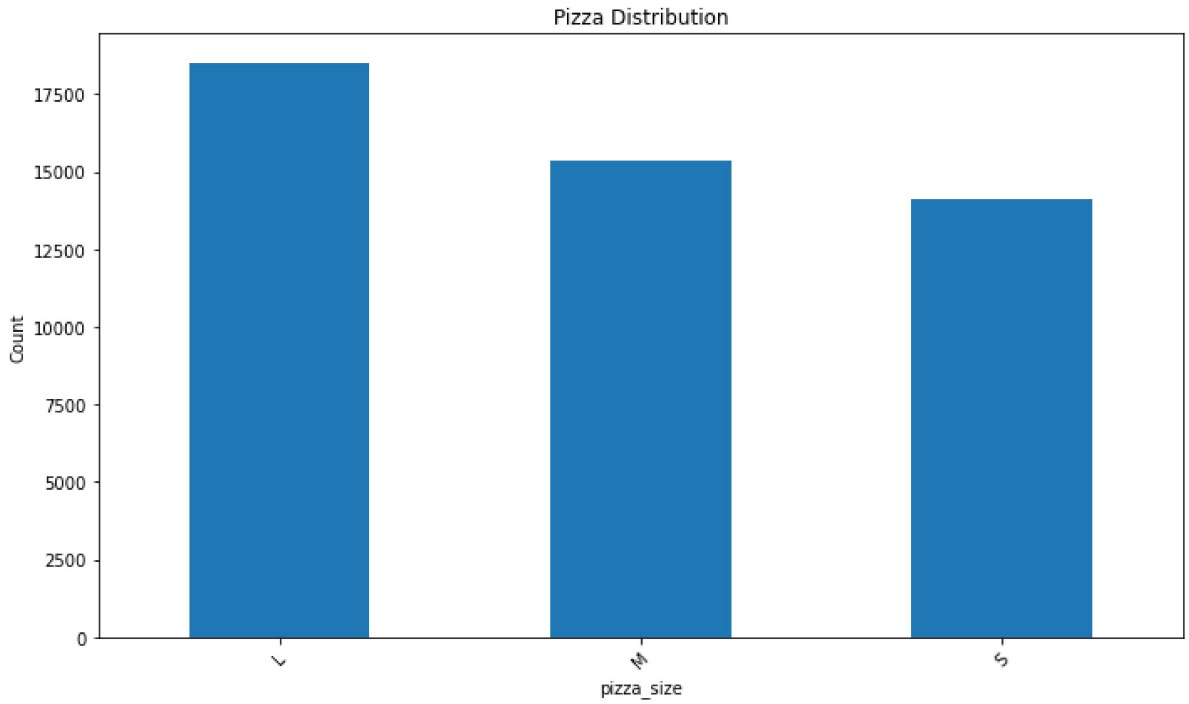
In [29]:
```
# Show the number of orders for each category of pizza
categories = df.pizza_category.value_counts()
categories
```

Out[29]:
```
Classic    14579
Supreme    11777
Veggie     11449
Chicken    10815
Name: pizza_category, dtype: int64
```

In [31]:
```
# 8. What is the total revenue up to the latest order date?
total_revenue = df['unit_price'].sum()
total_revenue
```

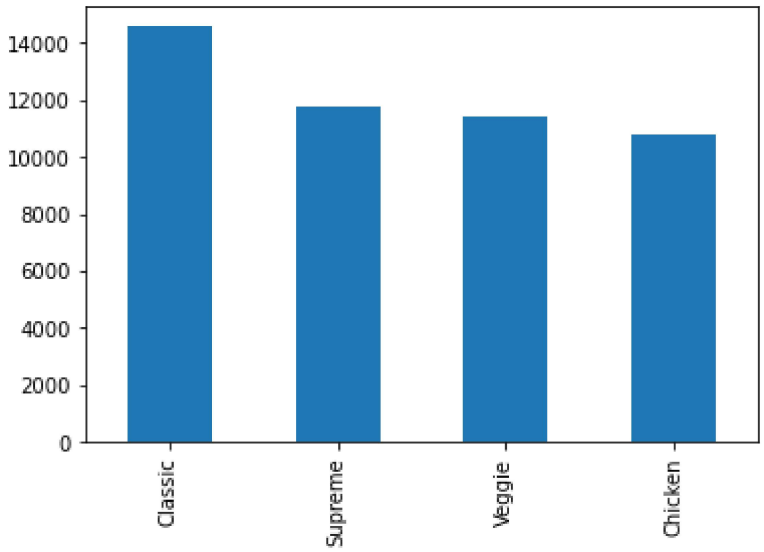
Out[31]:
```
801944.7000000001
```

In [37]:
```
# Count the occurrences of each currency
source_counts = df['pizza_size'].value_counts().head(3)

# Create a bar plot
plt.figure(figsize=(10, 6))
source_counts.plot(kind='bar')
plt.title('Pizza Distribution')
plt.xlabel('pizza_size')
plt.ylabel('Count')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.tight_layout()

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

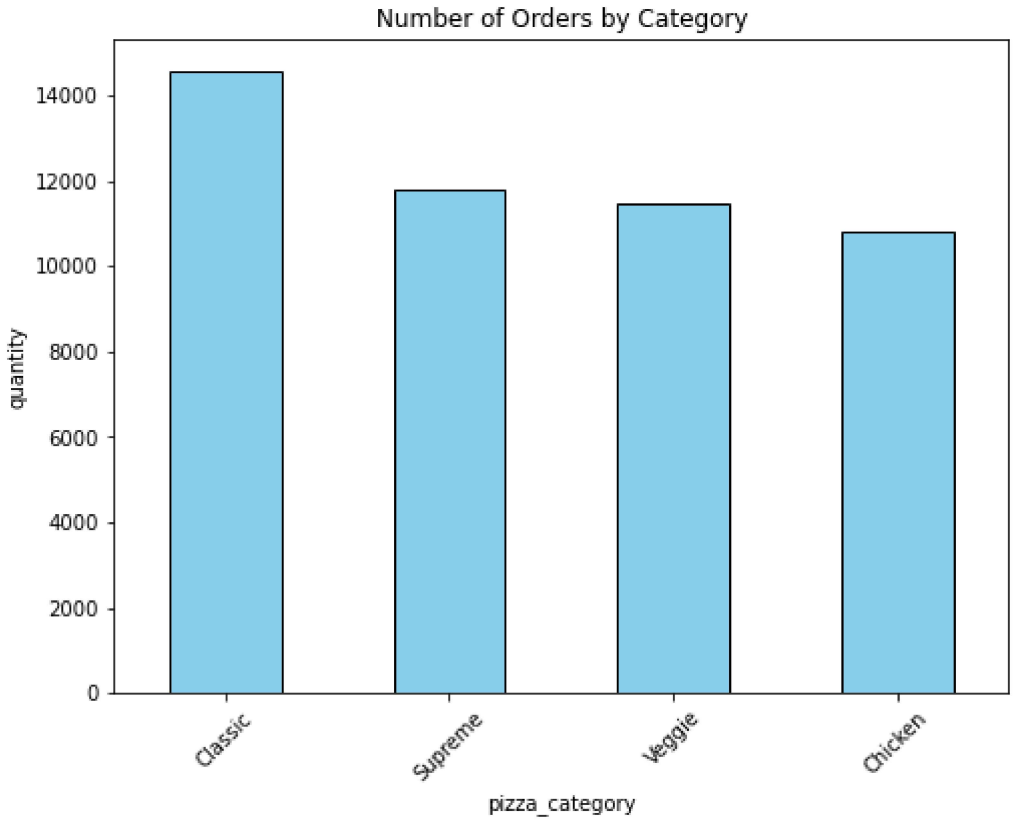


In [33]:
```
df.pizza_category.value_counts().plot(kind = 'bar')
```

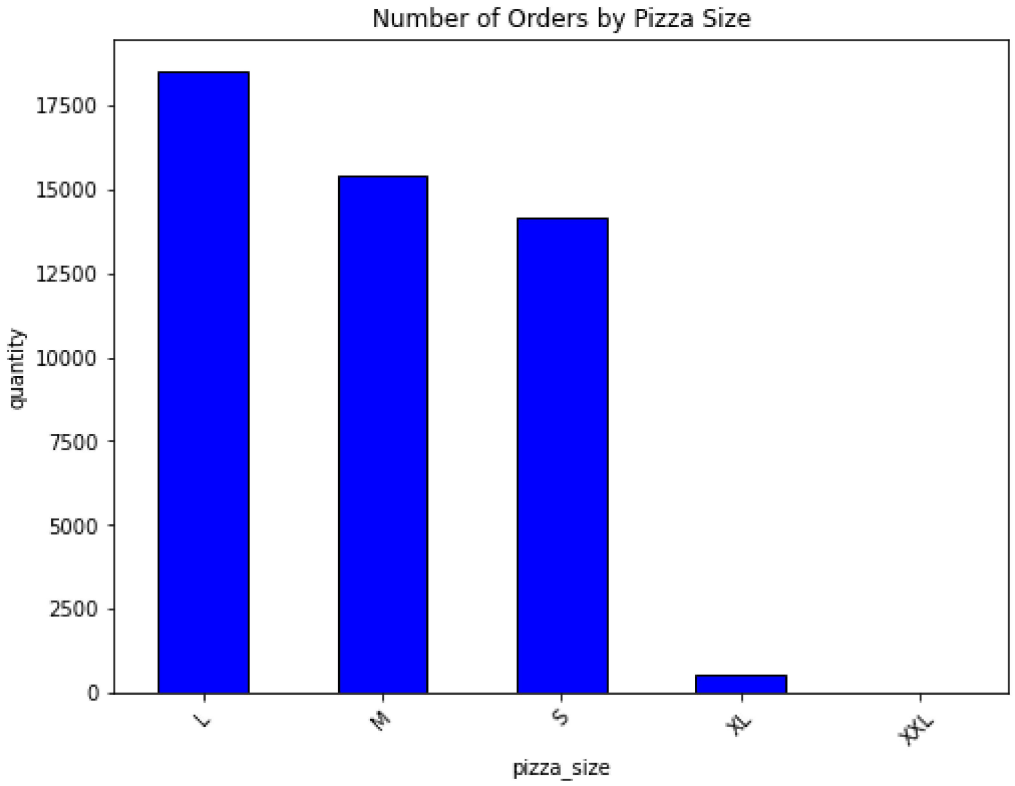
Out[33]: <AxesSubplot:>



In [48]:
```
order_count_by_category = df['pizza_category'].value_counts()

# Create a bar plot
plt.figure(figsize=(8, 6))
order_count_by_category.plot(kind='bar', color='skyblue', edgecolor='black')
plt.xlabel('pizza_category')
plt.ylabel('quantity')
plt.title('Number of Orders by Category')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.show()
```



In [52]:
```
order_count_by_category = df['pizza_size'].value_counts()

# Create a bar plot
plt.figure(figsize=(8, 6))
order_count_by_category.plot(kind='bar', color='blue', edgecolor='black')
plt.xlabel('pizza_size')
plt.ylabel('quantity')
plt.title('Number of Orders by Pizza Size')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.show()
```



In [ ]: