```
In [2]: import pandas as pd
import numpy as np
import seaborn as sea
import matplotlib.pyplot as mtn
import psycopg2
```

```
In [4]: db = psycopg2.connect(
    host='localhost',
    user='postgres',
    password='SQUAL#QUE',
    database='Pizzahut'
)

cur = db.cursor()
```

1) Retrieve the total number of orders placed.

```
In [7]: query = """ select count(order_id) as Total_orders from orders """
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ["Total Orders"])
df
```

Out[7]:

	Total Orders
0	21350

2) Calculate the total revenue generated from pizza sales.

```
In [9]: query = """ select round(sum(p.price*o.quantity),0) as Total_Revenue from pizzas as p
join orders_details as o
ON p.pizza_id = o.pizza_id"""
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ["Total Revenue"])
df
```

Out[9]:

	Total Revenue
0	817860

3) Identify the highest-priced pizza.

```
In [12]: query = """ SELECT pz.name, round(sum(p.price),0) as Price FROM pizza_forms as pz
join pizzas as p
ON pz.pizza_type_id = p.pizza_type_id
group by 1
order by 2 desc
limit 1"""
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ["Pizza name", "Price"])
df
```

Out[12]:

	Pizza name	Price
0	The Greek Pizza	110

4) Identify the most common pizza size ordered.

```
In [14]: query = """ select p.size, count(o.order_id)as Total_orders from orders as o
join orders_details as og
ON o.order_id = og.order_id
join pizzas as p
ON og.pizza_id = p.pizza_id
group by 1
order by 2 desc
limit 1"""
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ["Size", "Total_orders"])
df
```

Out[14]:

	Size	Total_orders
0	L	18526

L size has the highest numbers of orders

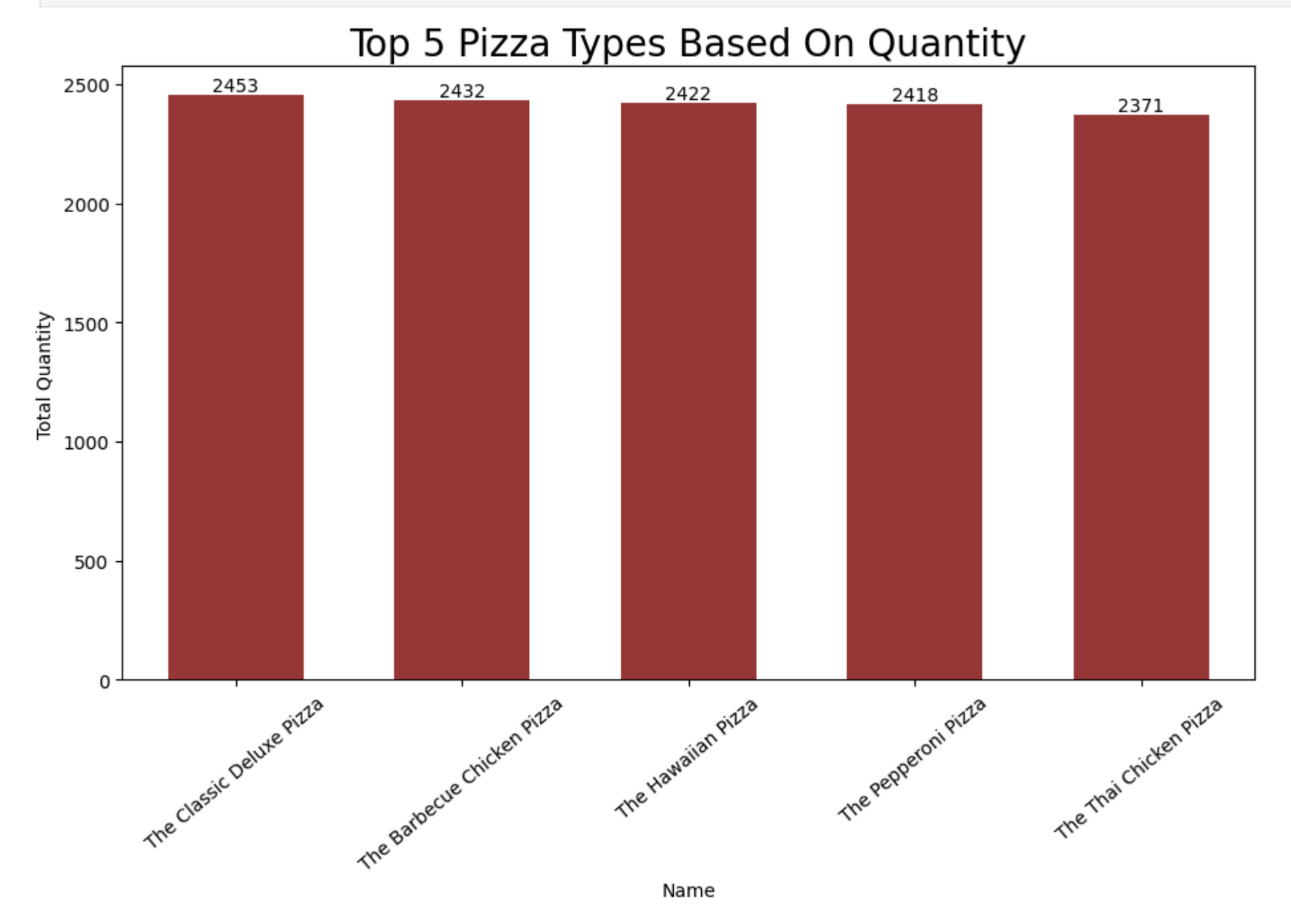
5) List the top 5 most ordered pizza types along with their quantities with the help of Bar Graph.

```
In [16]: query = """ select pz.name, sum(o.quantity) as Total_quantity from pizza_forms as pz
join pizzas as p
ON pz.pizza_type_id = p.pizza_type_id
join orders_details as o
ON p.pizza_id = o.pizza_id
group by 1
order by 2 desc
limit 5"""
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ["Name", "Total Quantity"])
df
```

Out[16]:

	Name	Total 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 [22]: mtn.figure(figsize = (11,6))
ax = sea.barplot(x = "Name", y = "Total Quantity", data = df, width = 0.6, color = 'Brown')
ax.bar_label(ax.containers[0])
mtn.title("Top 5 Pizza Types Based On Quantity", fontsize = 20)
mtn.xticks(rotation = 40)
mtn.show()
```



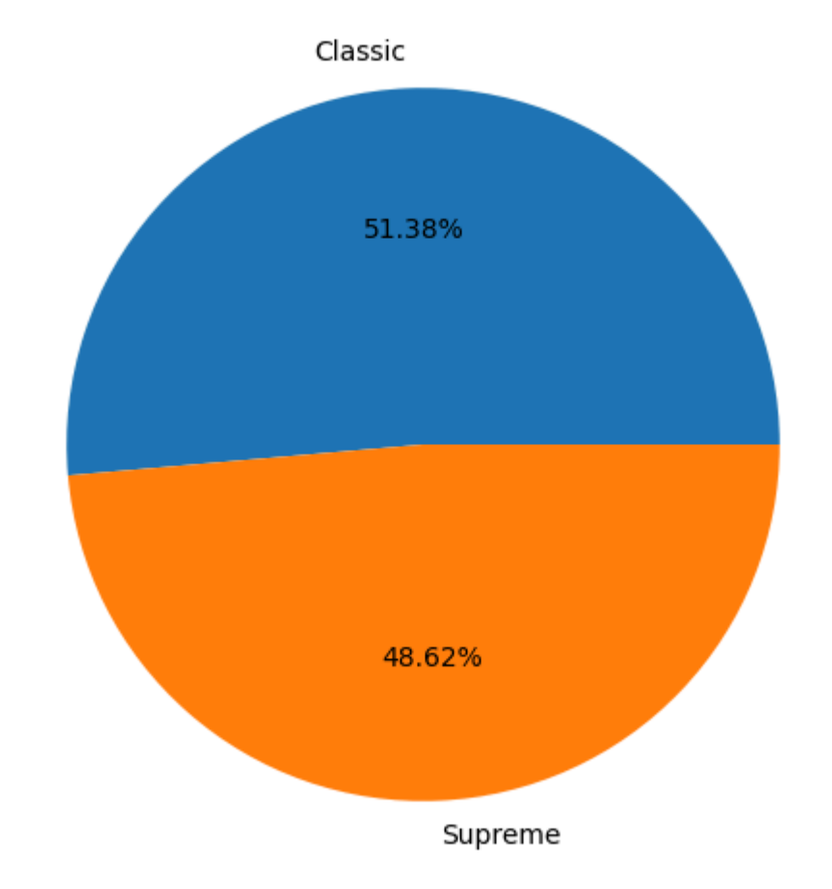
6. Describe the Total percentage Contribution in Revenue of Top 2 Pizza Category .

```
In [25]: query = """ select pz.category, round(sum(p.price*o.quantity),0) as Revenue from pizza_forms as pz
join pizzas as p
ON pz.pizza_type_id = p.pizza_type_id
join orders_details as o
ON p.pizza_id = o.pizza_id
group by 1
order by 2 desc
limit 2"""
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ['Pizza Category', "Revenue"])
df
```

Out[25]:

	Pizza Category	Revenue
0	Classic	220053
1	Supreme	208197

```
In [27]: mtn.figure(figsize = (11,6))
mtn.pie(df["Revenue"], autopct = "%1.2f%%", labels = df["Pizza Category"])
mtn.title("Percentage Contribution Of Top 2 Pizza Category")
mtn.show()
```



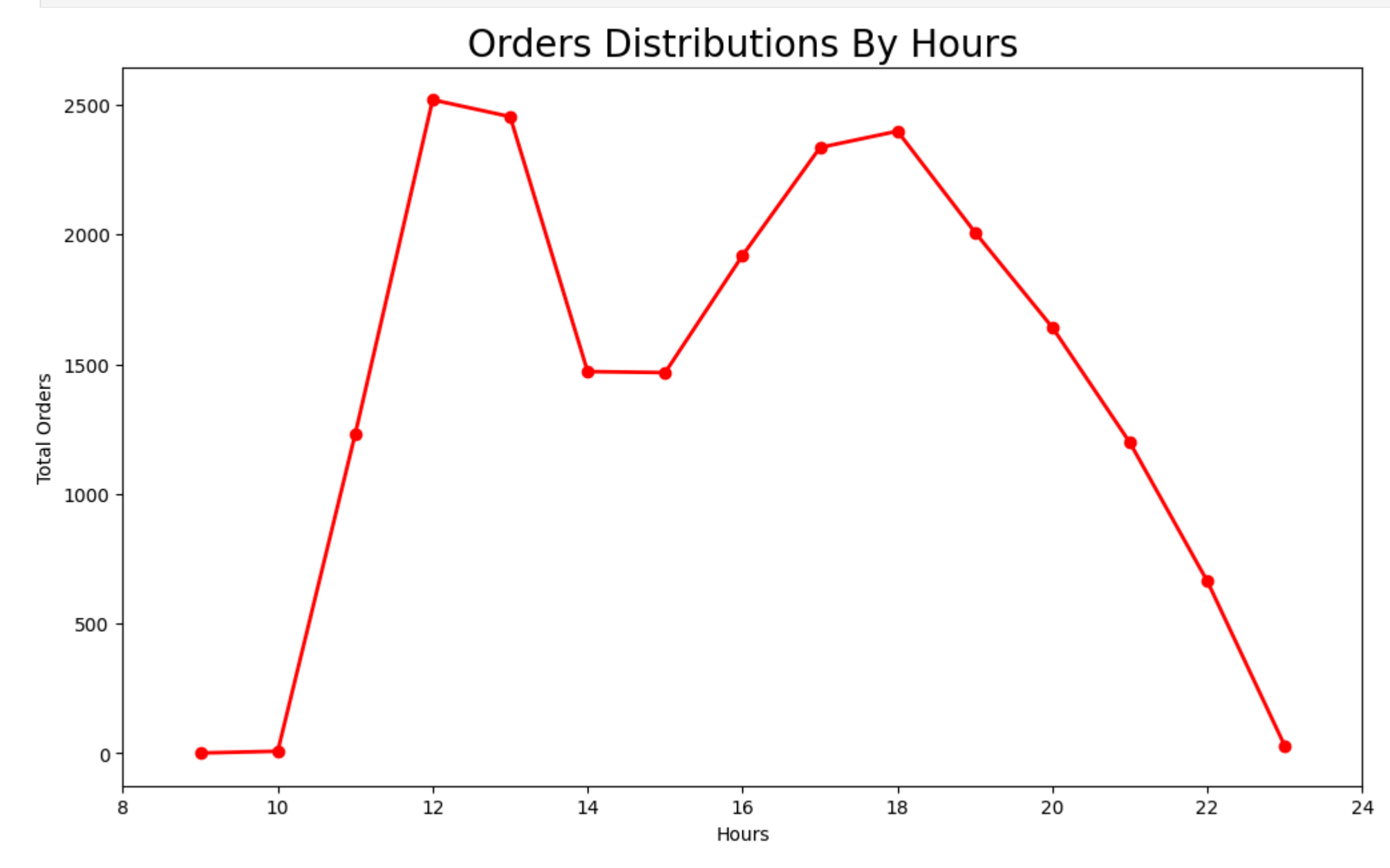
7. Determine the distribution of orders by hour of the day.

```
In [30]: query = """ select extract(hour from time) as hours, count(order_id) as Total_orders from orders
group by 1
order by 1"""
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ["Hours", "Total Orders"])
df
```

Out[30]:

	Hours	Total Orders
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 [45]: mtn.figure(figsize = (12,7))
mtn.plot(df["Hours"], df["Total Orders"], color = 'r', linewidth = 2, marker = 'o')
mtn.title("Orders Distributions By Hours", fontsize = 20)
mtn.xlabel("Hours")
mtn.ylabel("Total Orders")
mtn.xlim(8,24)
mtn.show()
```



8. Join relevant tables to find the category-wise distribution of pizzas.

```
In [47]: query = """ Select category, count(*) AS Total_contribution from pizza_forms
group by 1"""
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ["Categories", "Contribution"])
df
```

Out[47]:

	Categories	Contribution
0	Supreme	9
1	Classic	8
2	Veggie	9
3	Chicken	6

9. Group the orders by date and calculate the average number of pizzas ordered per day.

```
In [53]: query = """ select round(avg(Total_orders),0) as avg_pizza_per_day from
(select date, count(order_id) as Total_orders from orders
group by 1)as o"""
cur.execute(query)
data = cur.fetchall()
"Avg Pizza per day is", data[0][0]
```

```
Out[53]: ('Avg Pizza per day is', Decimal('60'))
```

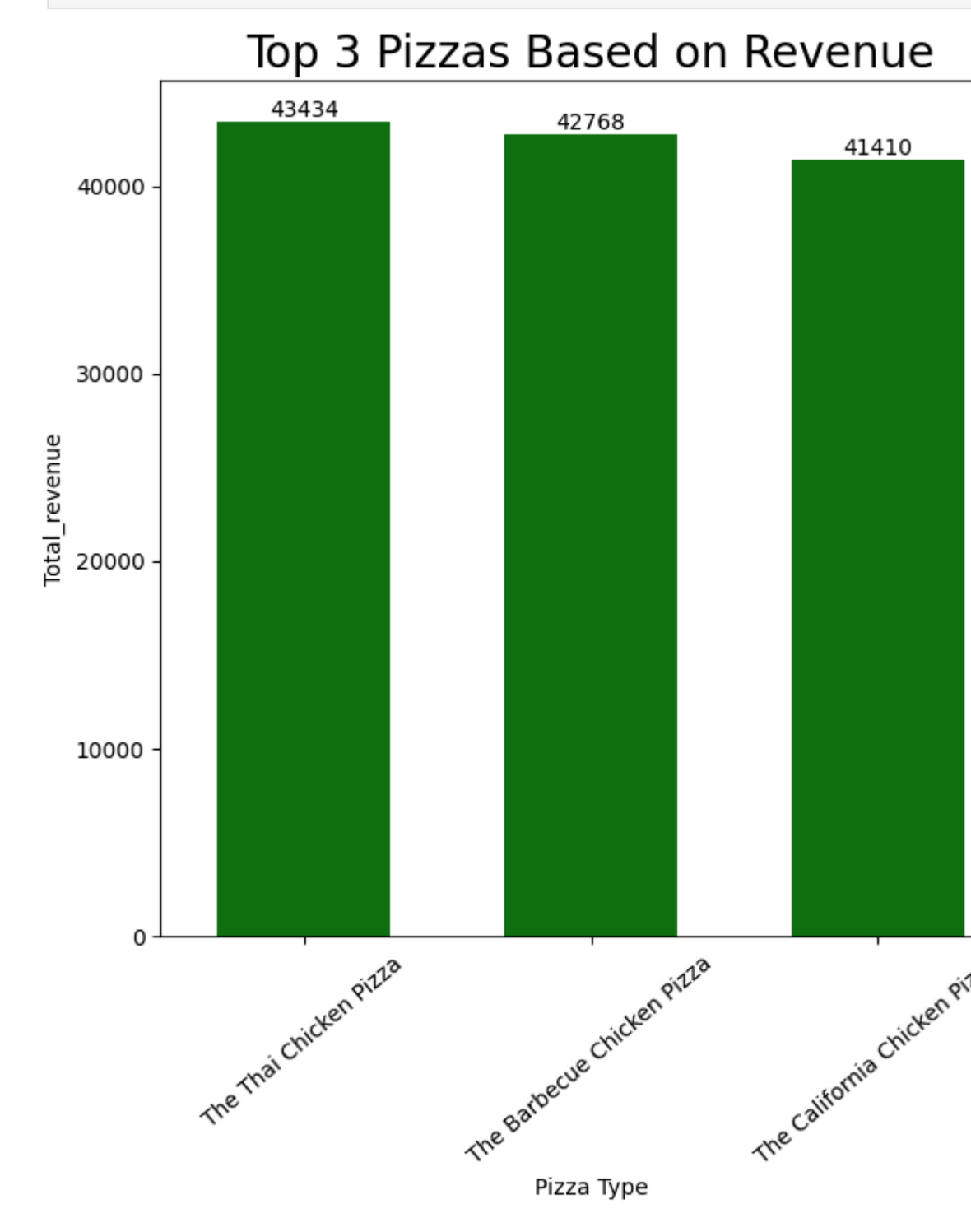
10. Determine the top 3 most ordered pizza types based on revenue.

```
In [55]: query = """ select pz.name, Round(sum(p.price*o.quantity),0) AS Total_revenue from pizza_forms as pz
join pizzas as p
ON pz.pizza_type_id = p.pizza_type_id
join orders_details as o
ON p.pizza_id = o.pizza_id
group by 1
order by 2 desc
limit 3"""
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ["Name", "Total_revenue"])
df
```

Out[55]:

	Name	Total_revenue
0	The Thai Chicken Pizza	43434
1	The Barbecue Chicken Pizza	42768
2	The California Chicken Pizza	41410

```
In [63]: mtn.figure(figsize = (7,7))
ax = sea.barplot(x = 'Name', y = "Total_revenue", data = df, color = 'Green', width = 0.6)
ax.bar_label(ax.containers[0])
mtn.xlabel("Pizza Type")
mtn.xticks(rotation = 40)
mtn.title("Top 3 Pizzas Based on Revenue", fontsize = 20)
mtn.show()
```



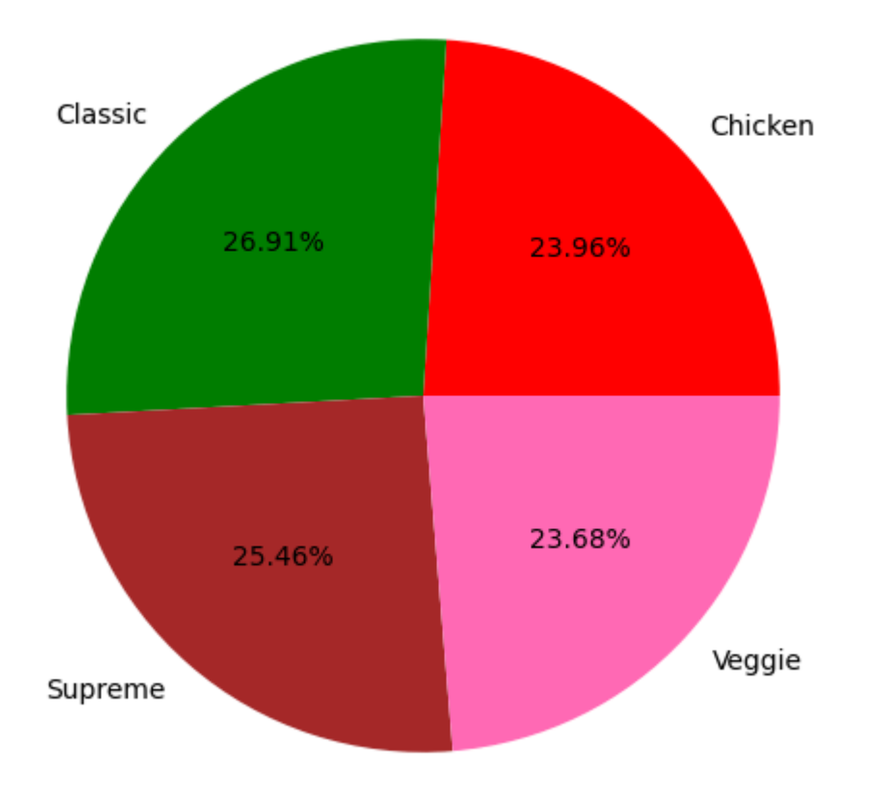
11. Calculate the percentage contribution of each pizza type to total revenue.

```
In [66]: query = """ select category, round((sum(Total_revenue)/(select sum(p.price*o.quantity) from pizzas as p
join orders_details as o
ON p.pizza_id = o.pizza_id))* 100,2) as percn_contribution from
(select pz.category, Round(sum(p.price*o.quantity),0) AS Total_revenue from pizza_forms as pz
join pizzas as p
ON pz.pizza_type_id = p.pizza_type_id
join orders_details as o
ON p.pizza_id = o.pizza_id
group by 1
order by 2 desc)as k
group by 1 """
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ["Name", "% contribution"])
df
```

Out[66]:

	Name	% contribution
0	Chicken	23.96
1	Classic	26.91
2	Supreme	25.46
3	Veggie	23.68

```
In [76]: mtn.figure(figsize = (10,6))
c = ['r','g','brown','hotpink']
mtn.pie(df["% contribution"], autopct = "%1.2f%%", colors = c , labels = df["Name"])
mtn.show()
```



12. Analyze the cumulative revenue generated over time.

```
In [77]: query = """ select *, sum(Total_revenue)over(order by date) as cumulative_revenue from
(select o.date, Round(sum(p.price*og.quantity),0) AS Total_revenue from pizza_forms as pz
join pizzas as p
ON pz.pizza_type_id = p.pizza_type_id
join orders_details as og
ON p.pizza_id = og.pizza_id
join orders as o
ON og.order_id = o.order_id
group by 1
order by 1)as k
"""
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ['Date', "Revenue", "Cumulative Revenue"])
df
```

Out[77]:

	Date	Revenue	Cumulative Revenue
0	2015-01-01	2714	2714
1	2015-01-02	2732	5446
2	2015-01-03	2662	8108
3	2015-01-04	1755	9863
4	2015-01-05	2066	11929
...	...	...	...
353	2015-12-27	1419	810626
354	2015-12-28	1637	812263
355	2015-12-29	1353	813616
356	2015-12-30	1338	814954
357	2015-12-31	2916	817870

358 rows × 3 columns

13.Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
In [81]: query = """ select name, category, revenue from
(select *, rank()over(partition by category order by revenue desc) as ranks from
(select pz.name, pz.category, sum(p.price*o.quantity) as Revenue from pizza_forms as pz
join pizzas as p
ON pz.pizza_type_id = p.pizza_type_id
join orders_details as o
ON p.pizza_id = o.pizza_id
group by 1,2)as k)as j
where ranks <= 3"""
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ["Name", "Category", "Revenue"])
df
```

Out[81]:

	Name	Category	Revenue
0	The Thai Chicken Pizza	Chicken	43434.25
1	The Barbecue Chicken Pizza	Chicken	42768.00
2	The California Chicken Pizza	Chicken	41409.50
3	The Classic Deluxe Pizza	Classic	38180.50
4	The Hawaiian Pizza	Classic	32273.25
5	The Pepperoni Pizza	Classic	30161.75
6	The Spicy Italian Pizza	Supreme	34831.25
7	The Italian Supreme Pizza	Supreme	33476.75
8	The Sicilian Pizza	Supreme	30940.50
9	The Four Cheese Pizza	Veggie	32265.70
10	The Mexicana Pizza	Veggie	26780.75

91	The Five Cheese Pizza	Veggie	26366.92