In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import squarify
import warnings
warnings.simplefilter(action="ignore",category =FutureWarning)
import sqlalchemy as sal
%matplotlib inline
```

In [2]:
```
pizza_sales = pd.read_csv("pizza sales.csv")
```

In [3]:
```
pizza_sales.head(3)
```

Out[3]:

	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	1/1/2015	11:38:36	13.25	13.25	M	Classic	Sliced Ham, Pineapple, Mozzarella Cheese	The Hawaiian Pizza
1	2	2	classic_dlx_m	1	1/1/2015	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	1/1/2015	11:57:40	18.50	18.50	L	Veggie	Mozzarella Cheese, Provolone Cheese, Smoked Go...	The Five Cheese Pizza

In [4]:
```
pizza_sales.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  object 
 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: float64(2), int64(3), object(7)
memory usage: 4.5+ MB
```

In [5]:
```
pizza_sales.describe()
```

Out[5]:

	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

In [6]:
```
pizza_sales.isna().sum()
```

Out[6]:
```
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
```

In [7]:
```
pizza_sales.isnull().sum()
```

Out[7]:
```
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 is clean

In [8]:
```
pizza_sales.shape
```

Out[8]:
```
(48620, 12)
```

Question:-

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?

In [9]:
```
pizza_sales.head()
```

Out[9]:

	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	1/1/2015	11:38:36	13.25	13.25	M	Classic	Sliced Ham, Pineapple, Mozzarella Cheese	The Hawaiian Pizza
1	2	2	classic_dlx_m	1	1/1/2015	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	1/1/2015	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	1/1/2015	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	1/1/2015	11:57:40	16.00	16.00	M	Veggie	Tomatoes, Red Peppers, Jalapeno Peppers, Red O...	The Mexicana Pizza

In [10]:
```
pizza_sales.pizza_category.value_counts()
```

Out[10]:
```
pizza_category
Classic     14579
Supreme     11777
Veggie      11449
Chicken     10815
Name: count, dtype: int64
```

In [11]:
```
pizza_sales.pizza_size.value_counts()
```

Out[11]:
```
pizza_size
L      18526
M      15385
S      14137
XL       544
XXL       28
Name: count, dtype: int64
```

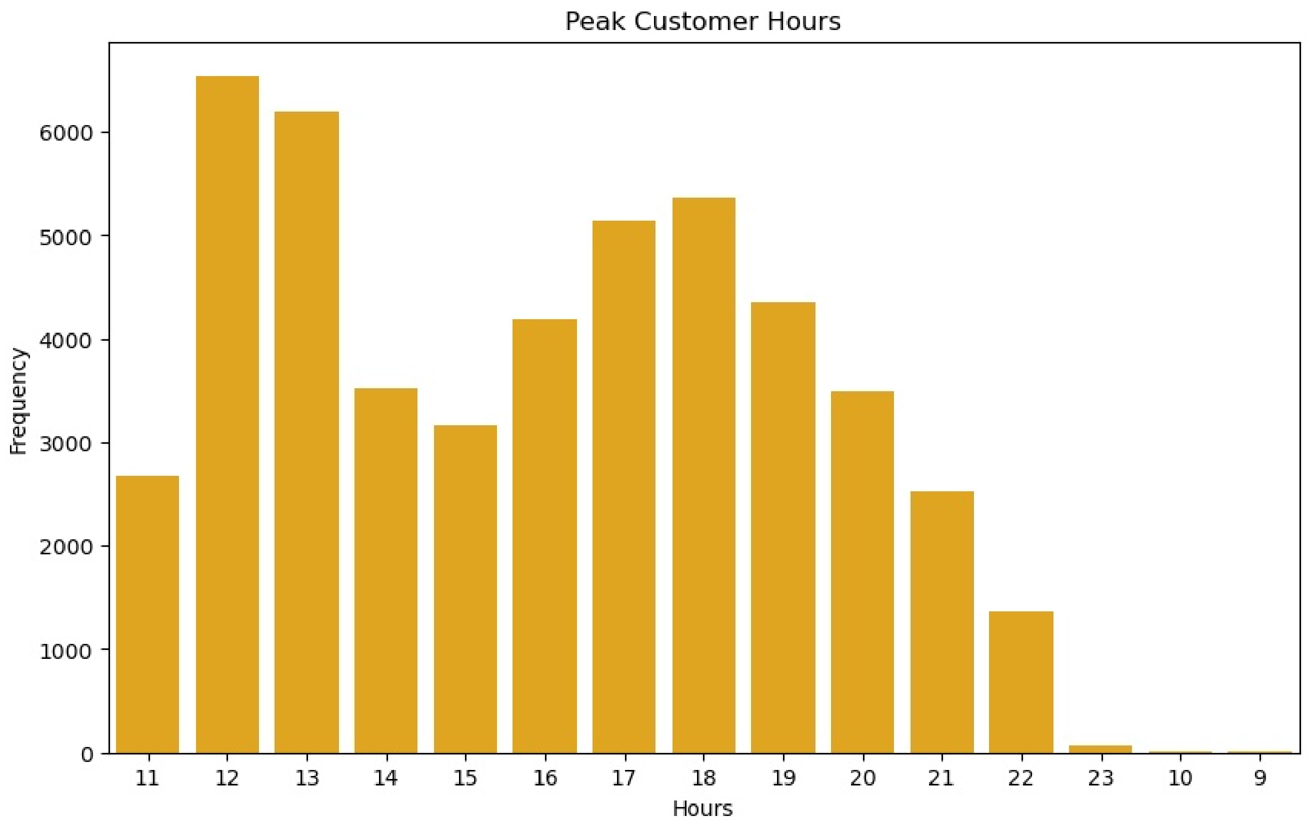
In [12]:
```
pizza_sales['order_date'] = pd.to_datetime(pizza_sales["order_date"])
```

In [13]:
```
pizza_sales["order_time"] = pizza_sales["order_time"].astype("string")
pizza_sales[['hour','minute','second']] = pizza_sales["order_time"].str.split(":",expand=True)
```

In [14]:
```
pizza_sales['hour'].value_counts()
```

```
Out[14]: 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
         9        4
         Name: count, dtype: Int64
```

```
In [15]: plt.figure(figsize=(10,6))
         sns.countplot(pizza_sales,x="hour",color="#FFB200")
         plt.xlabel("Hours")
         plt.ylabel("Frequency")
         plt.title("Peak Customer Hours")
         plt.show()
```

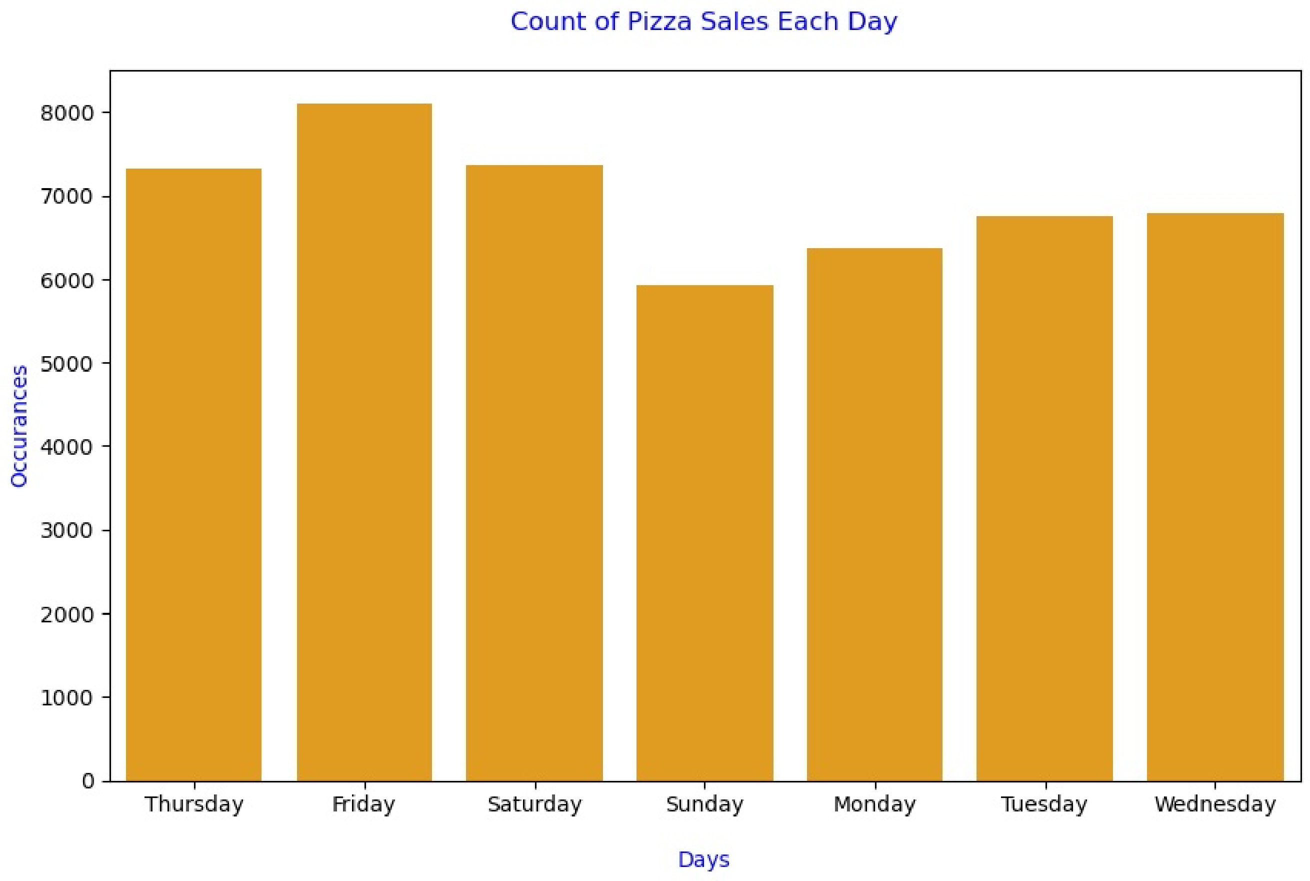


```
In [16]: pizza_sales["day_of_week"]= pizza_sales["order_date"].dt.day_name()
         pizza_sales["year"]=pizza_sales["order_date"].dt.year
```

```
In [17]: pizza_sales.day_of_week.value_counts()
```

```
Out[17]: day_of_week
         Friday       8106
         Saturday     7355
         Thursday     7323
         Wednesday    6797
         Tuesday      6753
         Monday       6369
         Sunday       5917
         Name: count, dtype: int64
```

```
In [18]: plt.figure(figsize=(10,6))
         sns.countplot(data=pizza_sales,x="day_of_week",color="orange")
         plt.xlabel("\nDays",color="blue")
         plt.ylabel("Occurances",color="blue")
         plt.title("Count of Pizza Sales Each Day\n",color="blue")
         plt.show()
```



```
In [19]: pizza_sales.year.value_counts()
```

Out[19]:
```
year
2015    48620
Name: count, dtype: int64
```

This data is of only a single year i.e. 2015

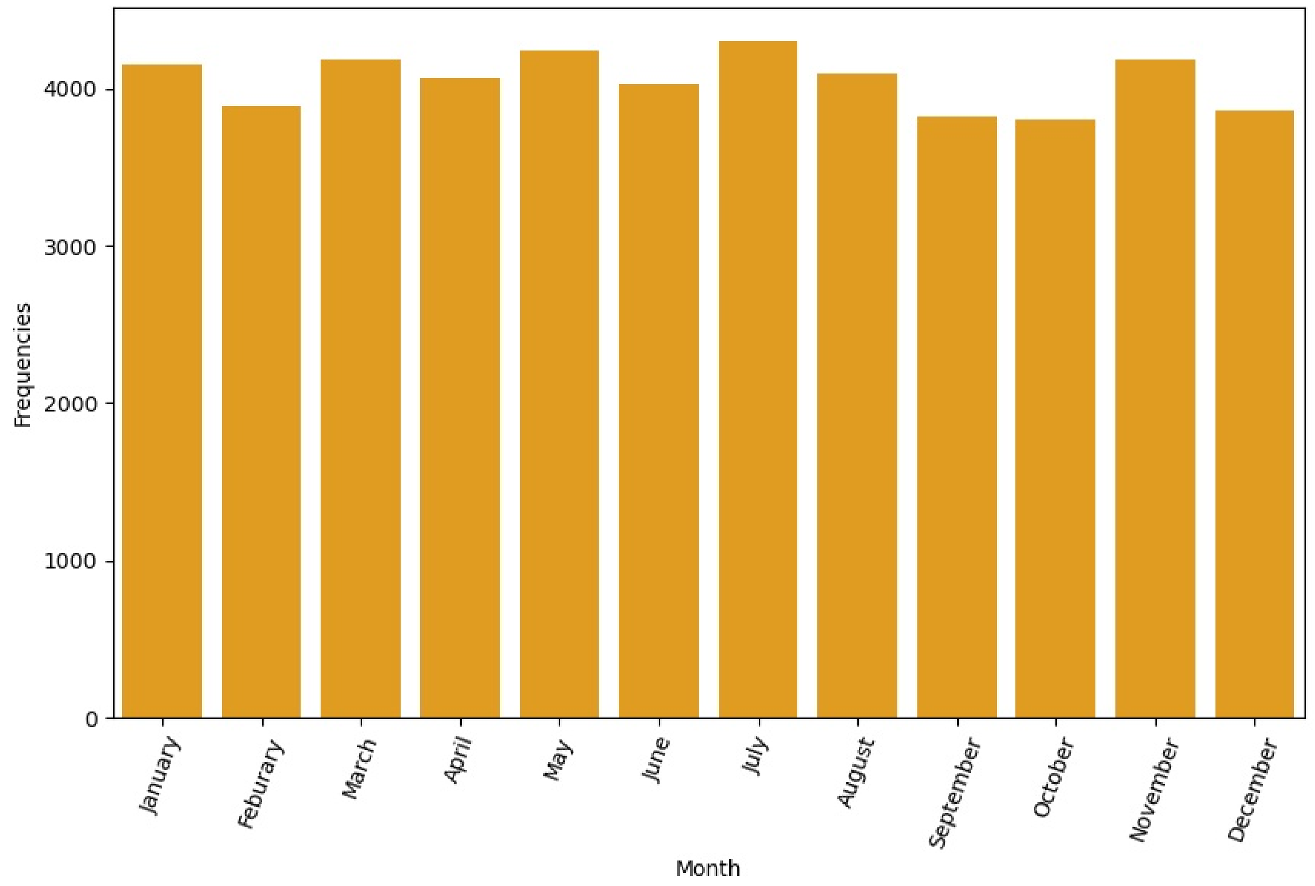
In [20]:
```
pizza_sales["month"] = pizza_sales['order_date'].dt.month
```

In [21]:
```
def rename_month(month):
    month_mapping = {
        1:"January",2:"Feburary",3:"March",4:"April",5:"May",6:"June",7:"July",8:"August",9:"September",10:"October",11:"November",12:"December"
    }
    return month_mapping.get(month,"Void")
pizza_sales["month"] = pizza_sales['month'].apply(rename_month)
```

In [22]:
```
pizza_sales.month.value_counts()
```

Out[22]:
```
month
July         4301
May          4239
March        4186
November     4185
January      4156
August       4094
April        4067
June         4025
Feburary     3892
December     3859
September    3819
October      3797
Name: count, dtype: int64
```

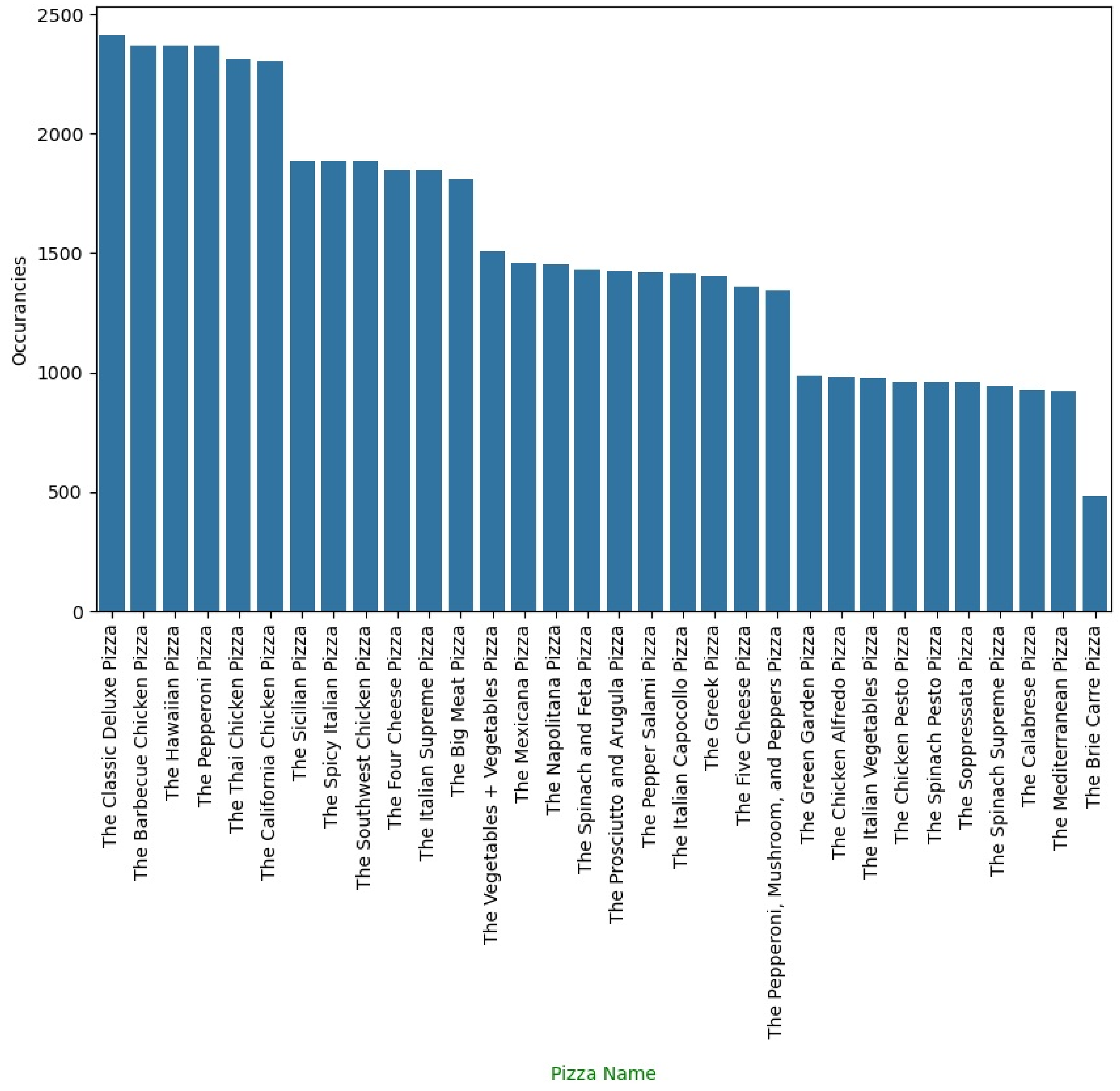
In [23]:
```
plt.figure(figsize=(10,6))
sns.countplot(data=pizza_sales,x="month",color="orange")
plt.xticks(rotation=70)
plt.xlabel("Month")
plt.ylabel("Frequencies")
plt.show()
```



In [24]:
```
pizza_order=pizza_sales.pizza_name.value_counts().index
pizza_sales.pizza_name.value_counts()
```

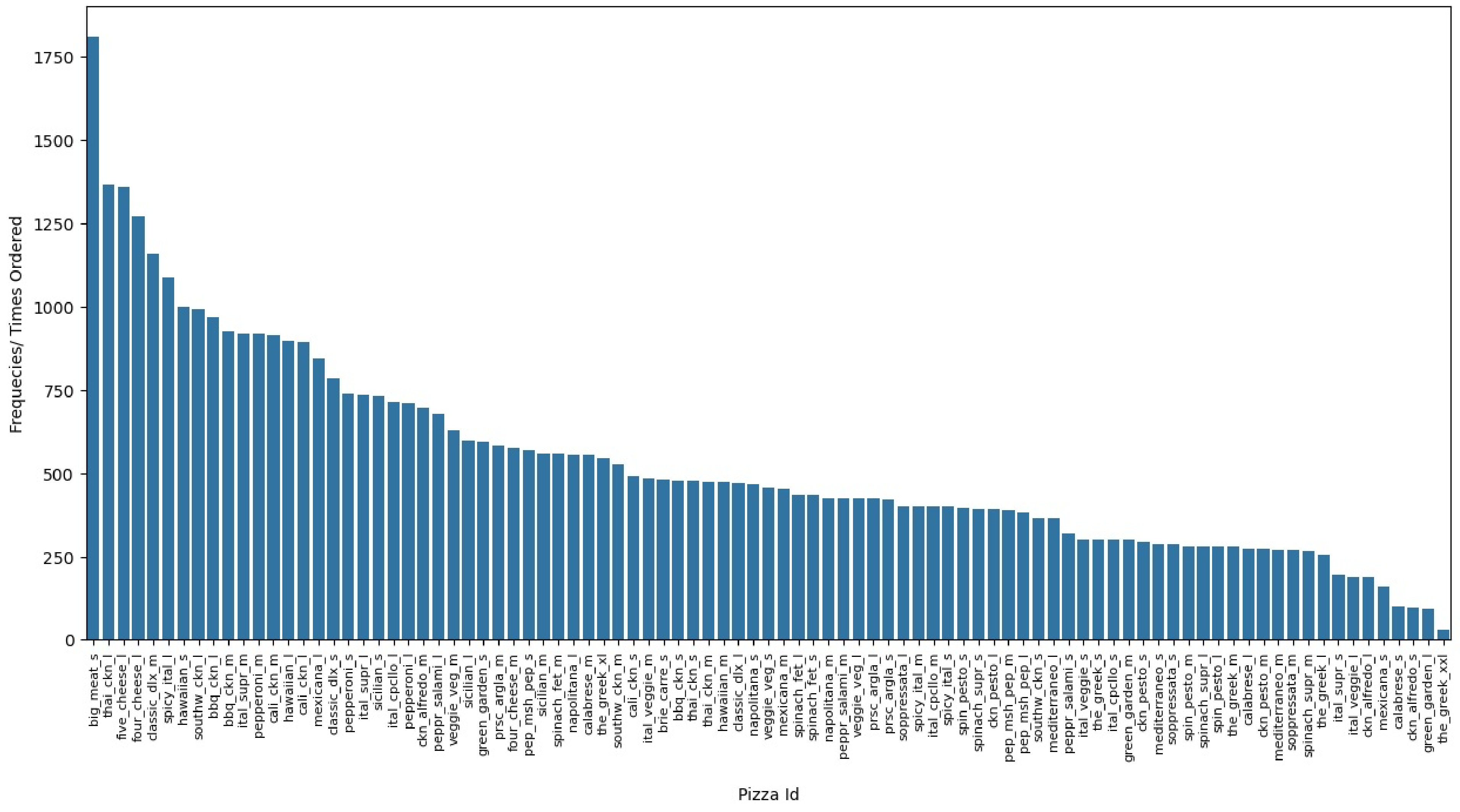
```
Out[24]: pizza_name
         The Classic Deluxe Pizza                      2416
         The Barbecue Chicken Pizza                    2372
         The Hawaiian Pizza                            2370
         The Pepperoni Pizza                           2369
         The Thai Chicken Pizza                        2315
         The California Chicken Pizza                  2302
         The Sicilian Pizza                            1887
         The Spicy Italian Pizza                       1887
         The Southwest Chicken Pizza                   1885
         The Four Cheese Pizza                         1850
         The Italian Supreme Pizza                     1849
         The Big Meat Pizza                            1811
         The Vegetables + Vegetables Pizza             1510
         The Mexicana Pizza                            1456
         The Napolitana Pizza                          1451
         The Spinach and Feta Pizza                    1432
         The Prosciutto and Arugula Pizza              1428
         The Pepper Salami Pizza                       1422
         The Italian Capocollo Pizza                   1414
         The Greek Pizza                               1406
         The Five Cheese Pizza                         1359
         The Pepperoni, Mushroom, and Peppers Pizza    1342
         The Green Garden Pizza                         987
         The Chicken Alfredo Pizza                      980
         The Italian Vegetables Pizza                   975
         The Chicken Pesto Pizza                        961
         The Spinach Pesto Pizza                        957
         The Soppressata Pizza                          957
         The Spinach Supreme Pizza                      940
         The Calabrese Pizza                            927
         The Mediterranean Pizza                        923
         The Brie Carre Pizza                           480
         Name: count, dtype: int64
```

```
In [25]: plt.figure(figsize=(10,6))
         sns.countplot(data=pizza_sales,x="pizza_name",order=pizza_order)
         plt.xticks(rotation =90)
         plt.xlabel("\nPizza Name",color="green",)
         plt.ylabel("Occurancies")
         plt.show()
```

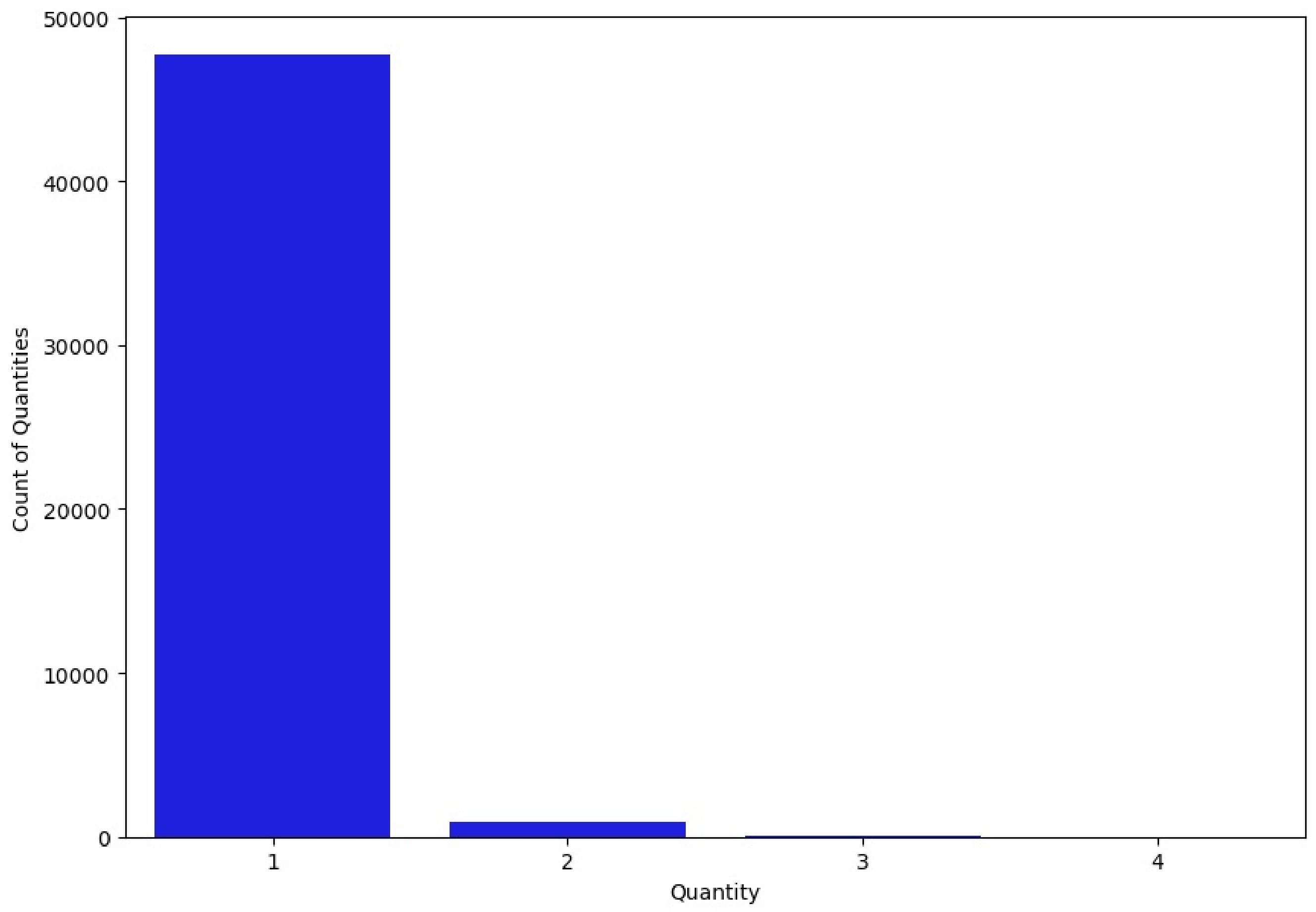


In [26]:
```
pizza_ids = pizza_sales.pizza_id.value_counts().index
plt.figure(figsize=(15,7))
sns.countplot(data=pizza_sales,x="pizza_id",order=pizza_ids)
plt.xticks(rotation=90,fontsize=8)
plt.xlabel("\nPizza Id")
plt.ylabel("Frequecies/ Times Ordered")
plt.show()
```

```
#pizza_id
```



In [27]:
```
pizza_quantity=pizza_sales.quantity.value_counts().index
plt.figure(figsize=(10,7))
sns.countplot(data=pizza_sales,x="quantity",order=pizza_quantity,color="blue")
plt.xlabel("Quantity")
plt.ylabel("Count of Quantities")
plt.show()
```

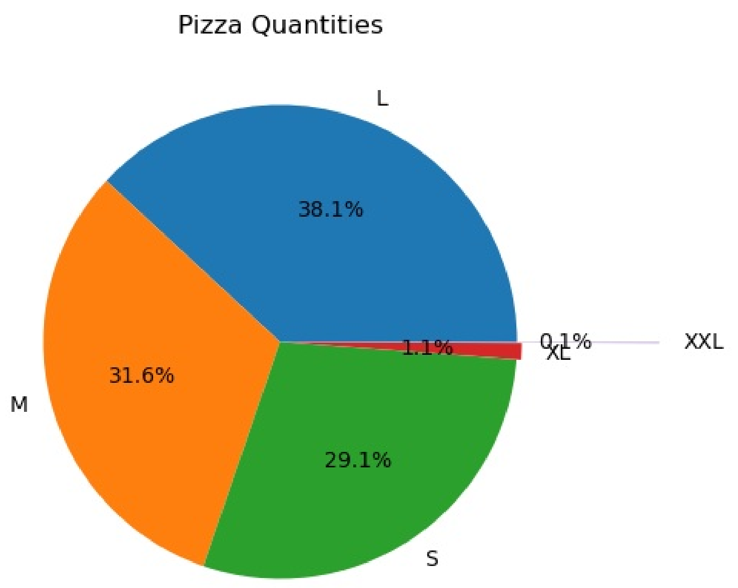


In [28]:
```
pizza_sales.quantity.value_counts()
```

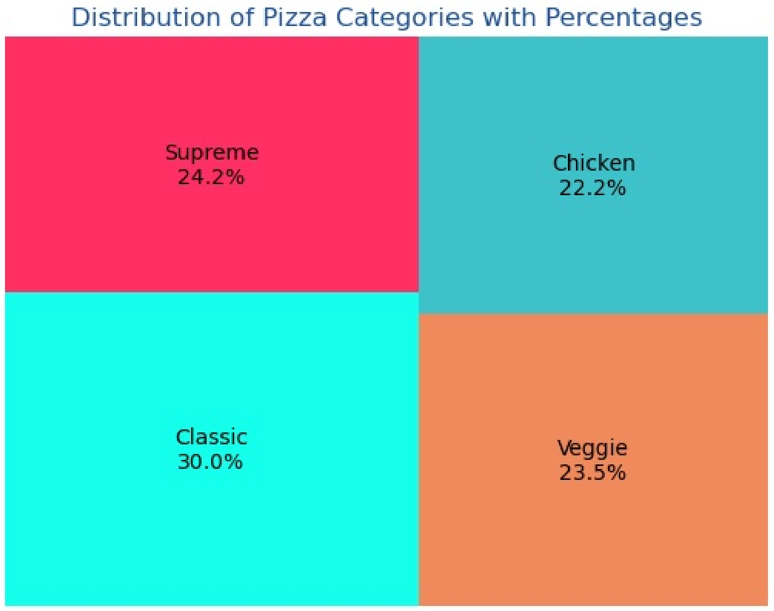
Out[28]:
```
quantity
1    47693
2      903
3       21
4        3
Name: count, dtype: int64
```

In [29]:
```
labels = pizza_sales["pizza_size"].value_counts().index
values = pizza_sales["pizza_size"].value_counts()
plt.figure(figsize=(6,5))

plt.pie(values,labels=labels,autopct="%1.1f%%",explode=(0,0,0,0.02,0.6))
plt.title("Pizza Quantities")
plt.show()
#pizza_category
```



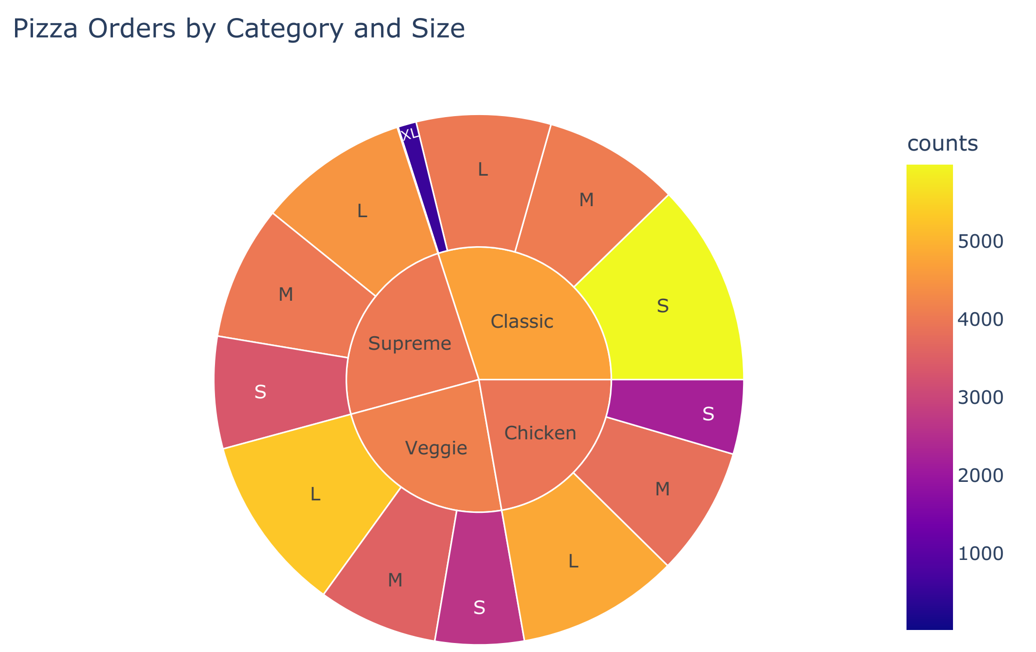
In [30]:
```
pzactgry = pizza_sales["pizza_category"].value_counts()
pzactgry_label = pizza_sales['pizza_category'].value_counts().index
total = pzactgry.sum()
labels_ = [f"{label}\n{value/total:.1%}" for label, value in zip(pzactgry_label, pzactgry)]
color = ['#14FFEC','#FF2E63',"#F08A5D","#3FC1C9"]
squarify.plot(sizes=pzactgry,label=labels_,color=color)
plt.axis("off")
plt.title("Distribution of Pizza Categories with Percentages",color="#205295")
plt.show()
```



In [31]:
```
plt.figure(figsize=(10,6))
pizza_counts = pizza_sales.groupby(['pizza_category', 'pizza_size']).size().reset_index(name='counts')

fig = px.sunburst(
    pizza_counts,
    path=['pizza_category', 'pizza_size'],  # Define the hierarchy
    values='counts',                        # Use the count of occurrences as the size of each segment
    color='counts',                         # Color by counts
    title="Pizza Orders by Category and Size"
)


fig.show()
```



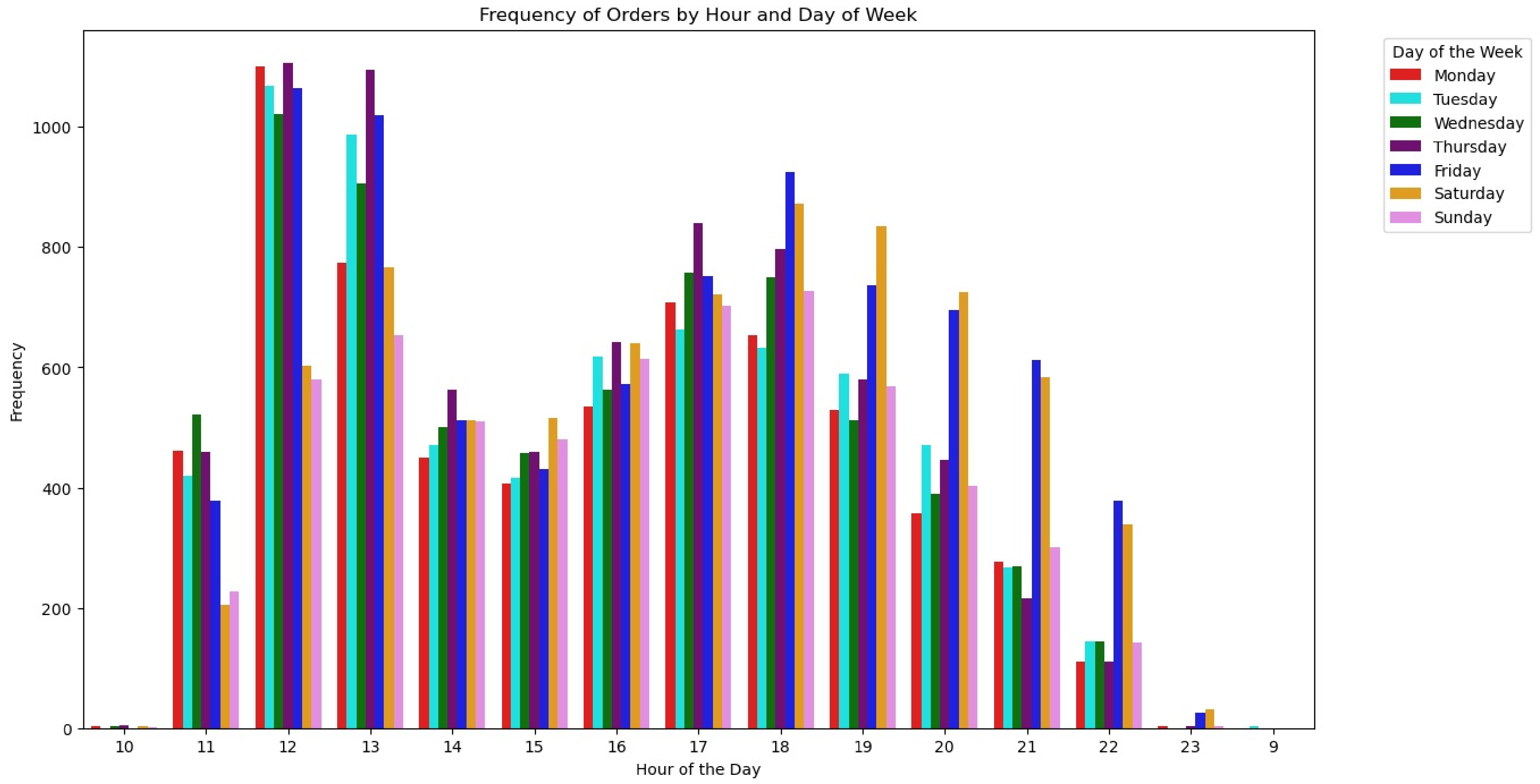
```
<Figure size 1000x600 with 0 Axes>
```

In [32]:
```
bar_data = pizza_sales.groupby(['hour', 'day_of_week']).size().reset_index(name='frequency')
day_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']


bar_data['day_of_week'] = pd.Categorical(bar_data['day_of_week'], categories=day_order, ordered=True)

palette = ["red", "cyan", "green", "purple", "blue", "orange", "violet"]

plt.figure(figsize=(14, 8))
sns.barplot(data=bar_data, x='hour', y='frequency', hue='day_of_week', palette=palette)
plt.title('Frequency of Orders by Hour and Day of Week')
plt.xlabel('Hour of the Day')
plt.ylabel('Frequency')
plt.legend(title='Day of the Week', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```


Frequency of Orders by Hour and Day of Week
Day of the Week
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Frequency
0
200
400
600
800
1000
Hour of the Day
10
11
12
13
14
15
16
17
18
19
20
21
22
23
9

```
In [ ]:
```