In [2]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

In [3]:
```python
df = pd.read_csv(r'C:\Users\MALU\Downloads\zomato_dataset.csv')
```

In [4]:
```python
df
```

Out[4]:

	Restaurant Name	Rating	Cuisine	Average Price	Average Delivery Time	Safety Measure	Location
0	Campus Bakers	4.3	Bakery, Fast Food, Pizza, Sandwich, Burger	₹50 for one	36 min	Restaurant partner follows WHO protocol	Agra
1	Mama Chicken Mama Franky House	4	North Indian, Mughlai, Rolls, Burger, Momos	₹50 for one	22 min	Follows all Max Safety measures to ensure your...	Agra
2	GMB - Gopika Sweets & Restaurant	4.2	North Indian, South Indian, Chinese, Fast Food...	₹50 for one	27 min	Follows all Max Safety measures to ensure your...	Agra
3	Shree Bankey Bihari Misthan Bhandar	4.2	Mithai, Street Food, South Indian, Chinese, Ic...	₹50 for one	28 min	Restaurant partner follows WHO protocol	Agra
4	Burger King	4.2	Burger, Fast Food, Beverages	₹50 for one	26 min	Follows all Max Safety measures to ensure your...	Agra
...	...	...	...	...	...	...	...
44886	Bowl 99	-	North Indian	₹350 for one	Opens tomorrow at 11am	Follows all Max Safety measures to ensure your...	Warangal
44887	Vegan Mart	-	Juices, Beverages	₹350 for one	Opens tomorrow at 11am	Follows all Max Safety measures to ensure your...	Warangal
44888	Biryani Plaza	-	Biryani	₹350 for one	Opens tomorrow at 11am	Follows all Max Safety measures to ensure your...	Warangal
44889	Biryani Plaza	-	Biryani	₹350 for one	Opens tomorrow at 11am	Follows all Max Safety measures to ensure your...	Warangal
44890	Biryani Plaza	-	Biryani	₹350 for one	Opens tomorrow at 11am	Follows all Max Safety measures to ensure your...	Warangal

44891 rows × 7 columns

In [5]:
```python
print(df.isnull().sum())
```

```
Restaurant Name          0
Rating                   0
Cuisine                 19
Average Price            0
Average Delivery Time    0
Safety Measure           0
Location                 0
dtype: int64
```

In [6]:
```python
# Data statistics
df.describe()
```

Out[6]:

	Restaurant Name	Rating	Cuisine	Average Price	Average Delivery Time	Safety Measure	Location
count	44891	44891	44872	44891	44891	44891	44891
unique	35688	30	17580	28	398	2	98
top	Baskin Robbins - Ice Cream Desserts	4.0	North Indian	₹100 for one	30 min	Follows all Max Safety measures to ensure your...	Kolkata
freq	70	5575	2403	13019	1715	25726	1296

In [7]:
```python
# extracting numeric part
df['Average Price'] = df['Average Price'].str.extract('(\d+)').fillna(0).astype(int)
```

In [8]:
```python
df
```

Out[8]:

	Restaurant Name	Rating	Cuisine	Average Price	Average Delivery Time	Safety Measure	Location
0	Campus Bakers	4.3	Bakery, Fast Food, Pizza, Sandwich, Burger	50	36 min	Restaurant partner follows WHO protocol	Agra
1	Mama Chicken Mama Franky House	4	North Indian, Mughlai, Rolls, Burger, Momos	50	22 min	Follows all Max Safety measures to ensure your...	Agra
2	GMB - Gopika Sweets & Restaurant	4.2	North Indian, South Indian, Chinese, Fast Food...	50	27 min	Follows all Max Safety measures to ensure your...	Agra
3	Shree Bankey Bihari Misthan Bhandar	4.2	Mithai, Street Food, South Indian, Chinese, Ic...	50	28 min	Restaurant partner follows WHO protocol	Agra
4	Burger King	4.2	Burger, Fast Food, Beverages	50	26 min	Follows all Max Safety measures to ensure your...	Agra
...	...	...	...	...	...	...	...
44886	Bowl 99	-	North Indian	350	Opens tomorrow at 11am	Follows all Max Safety measures to ensure your...	Warangal
44887	Vegan Mart	-	Juices, Beverages	350	Opens tomorrow at 11am	Follows all Max Safety measures to ensure your...	Warangal
44888	Biryani Plaza	-	Biryani	350	Opens tomorrow at 11am	Follows all Max Safety measures to ensure your...	Warangal
44889	Biryani Plaza	-	Biryani	350	Opens tomorrow at 11am	Follows all Max Safety measures to ensure your...	Warangal
44890	Biryani Plaza	-	Biryani	350	Opens tomorrow at 11am	Follows all Max Safety measures to ensure your...	Warangal

44891 rows × 7 columns

In [9]:
```python
# Restaurant with rating rating '_' is replaced with o
df['Rating'] = df['Rating'].replace('-','0')
```

In [10]:
```python
df
```

Out[10]:

	Restaurant Name	Rating	Cuisine	Average Price	Average Delivery Time	Safety Measure	Location
0	Campus Bakers	4.3	Bakery, Fast Food, Pizza, Sandwich, Burger	50	36 min	Restaurant partner follows WHO protocol	Agra
1	Mama Chicken Mama Franky House	4	North Indian, Mughlai, Rolls, Burger, Momos	50	22 min	Follows all Max Safety measures to ensure your...	Agra
2	GMB - Gopika Sweets & Restaurant	4.2	North Indian, South Indian, Chinese, Fast Food...	50	27 min	Follows all Max Safety measures to ensure your...	Agra
3	Shree Bankey Bihari Misthan Bhandar	4.2	Mithai, Street Food, South Indian, Chinese, Ic...	50	28 min	Restaurant partner follows WHO protocol	Agra
4	Burger King	4.2	Burger, Fast Food, Beverages	50	26 min	Follows all Max Safety measures to ensure your...	Agra
...	...	...	...	...	...	...	...
44886	Bowl 99	0	North Indian	350	Opens tomorrow at 11am	Follows all Max Safety measures to ensure your...	Warangal
44887	Vegan Mart	0	Juices, Beverages	350	Opens tomorrow at 11am	Follows all Max Safety measures to ensure your...	Warangal
44888	Biryani Plaza	0	Biryani	350	Opens tomorrow at 11am	Follows all Max Safety measures to ensure your...	Warangal
44889	Biryani Plaza	0	Biryani	350	Opens tomorrow at 11am	Follows all Max Safety measures to ensure your...	Warangal
44890	Biryani Plaza	0	Biryani	350	Opens tomorrow at 11am	Follows all Max Safety measures to ensure your...	Warangal

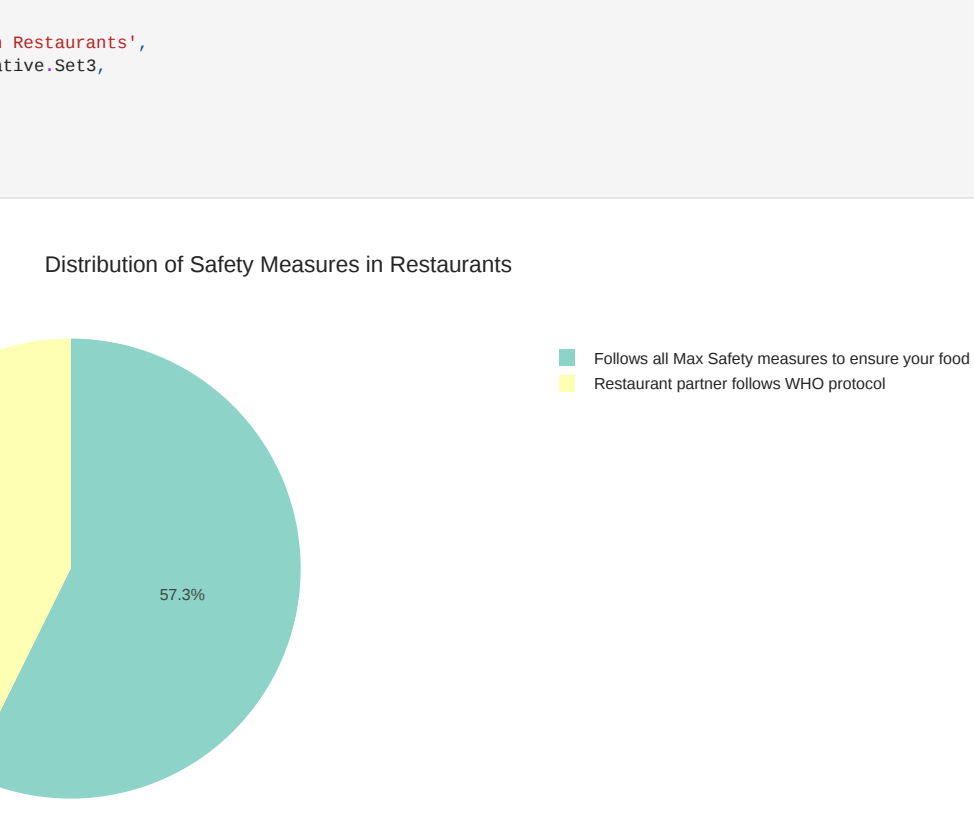
44891 rows × 7 columns

In [23]:
```python
# Exploratory Data Analysis
```

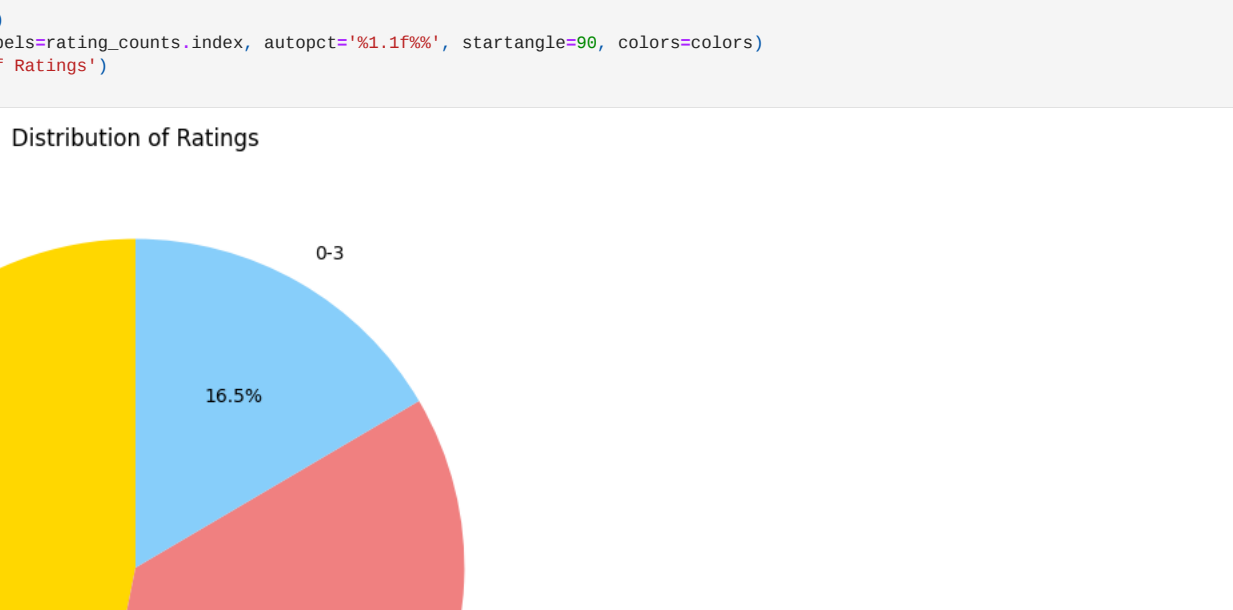
In [12]:
```python
df['Delivery status'] = df['Average Delivery Time'].apply(lambda x: 'open' if 'min' in str(x) else 'closed')
delivery_counts = df['Delivery status'].value_counts().reset_index()
```

In [14]:
```python
delivery_counts.columns = ['Delivery status','Count']
```

In [17]:
```python
fig = px.pie(delivery_counts,
            names = 'Delivery status',
            values = 'Count',
            title = 'Distribution of open and closed Restaurants',
            color_discrete_sequence=px.colors.qualitative.Pastel1,
            template = 'seaborn')
fig.update_traces(textposition ="outside")
fig.show()
```

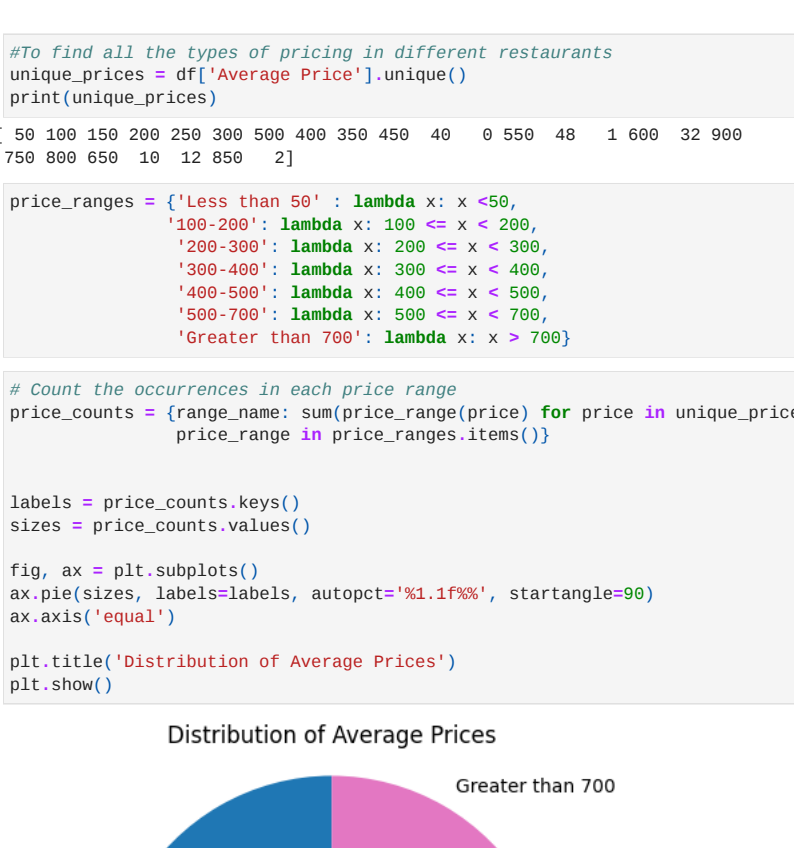


In [18]:
```python
safety_counts = df['Safety Measure'].value_counts().reset_index()
safety_counts.columns = ['Safety Measure', 'Count']

fig = px.pie(
    safety_counts,
    names='Safety Measure',
    values='Count',
    title='Distribution of Safety Measures in Restaurants',
    color_discrete_sequence=px.colors.qualitative.Set3,
    template='seaborn',
)


fig.show()
```



In [19]:
```python
import matplotlib.pyplot as plt
```

In [30]:
```python
df['Rating'] = pd.to_numeric(df['Rating'], errors='coerce')
df['Rating Category'] = df['Rating'].apply(lambda x: '0-3' if pd.isna(x) or x < 3 else ('3-4' if x < 4 else '4+'))
rating_counts = df['Rating Category'].value_counts()
colors = ['gold', 'lightcoral', 'lightskyblue']

plt.figure(figsize=(8, 8))
plt.pie(rating_counts, labels=rating_counts.index, autopct='%1.1f%%', startangle=90, colors=colors)
plt.title('Distribution of Ratings')
plt.show()
```

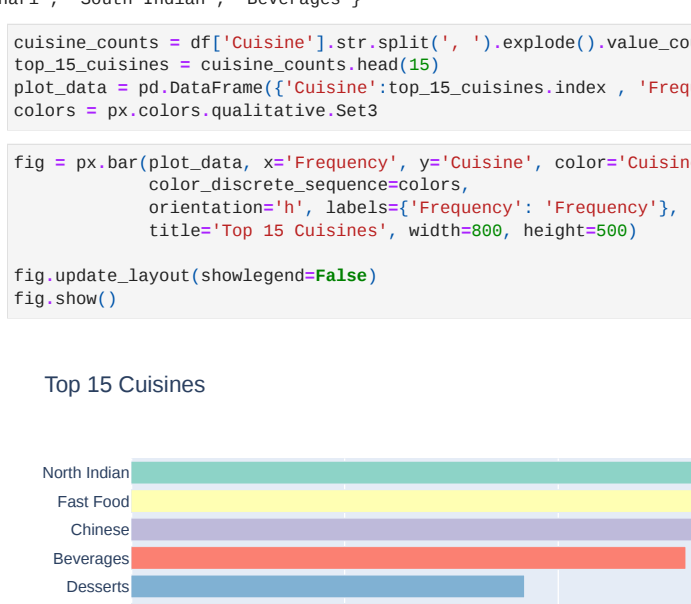


In [31]:
```python
#To find all the types of pricing in different restaurants
unique_prices = df['Average Price'].unique()
print(unique_prices)
```

```
[ 50 100 150 200 250 300 500 400 350 450  40   0 550  48   1 600  32 900
 750 800 650  10  12 850   2]
```

In [32]:
```python
price_ranges = {'Less than 50' : lambda x: x <50,
               '100-200': lambda x: 100 <= x < 200,
                '200-300': lambda x: 200 <= x < 300,
                '300-400': lambda x: 300 <= x < 400,
                '400-500': lambda x: 400 <= x < 500,
                '500-700': lambda x: 500 <= x < 700,
                'Greater than 700': lambda x: x > 700}
```

In [34]:
```python
# Count the occurrences in each price range
price_counts = {range_name: sum(price_range(price) for price in unique_prices) for range_name,
                price_range in price_ranges.items()}


labels = price_counts.keys()
sizes = price_counts.values()

fig, ax = plt.subplots()
ax.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)
ax.axis('equal')

plt.title('Distribution of Average Prices')
plt.show()
```



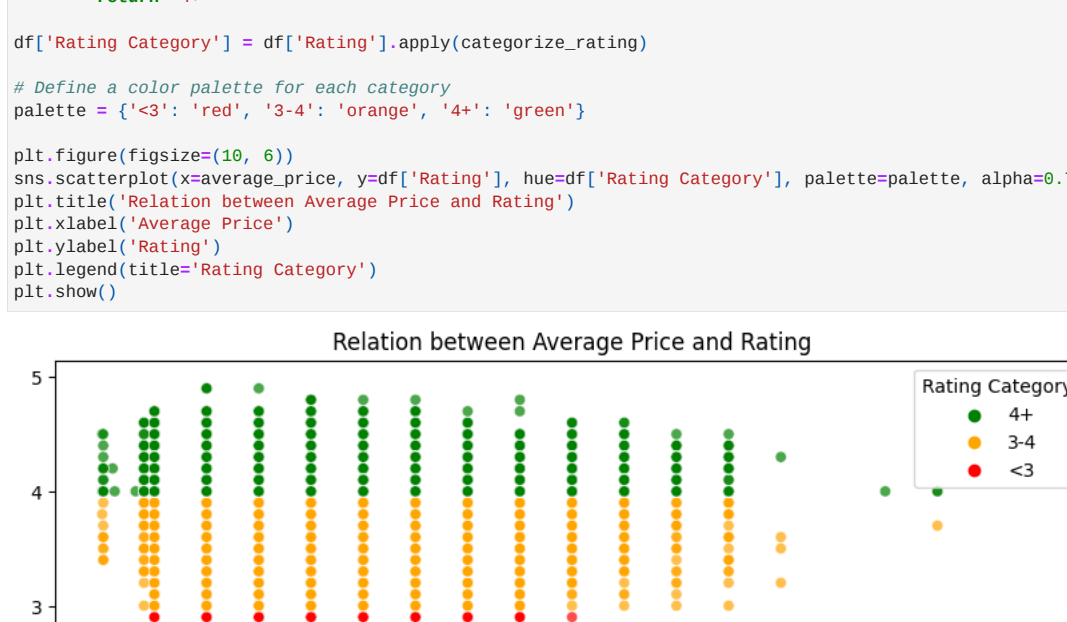
In [36]:
```python
# To find all possible cuisine in the dataset
cuisine_column = df['Cuisine']
all_cuisines = [cuisine.split(', ') for cuisine in cuisine_column if pd.notna(cuisine)]
unique_cuisines = set([c for sublist in all_cuisines for c in sublist])
print('All possible cuisines', unique_cuisines)
```

```
All possible cuisines {'', 'Singaporean', 'Bengali', 'Pizza', 'Street Food', 'Mishti', 'Rolls', 'Pancake', 'African', 'Burger', 'BBQ', 'Fast Food', 'Indonesian', 'Shawarma', 'Cake', 'Andhra', 'Sandwich', 'Ice Cre
am', 'Iranian', 'Kathiyawadi', 'Lebanese', 'Waffle', 'Paan', 'Turkish', 'Manipuri', 'Biryani', 'Salad', 'Saoji', 'Egyptian', 'Spanish', 'Juices', 'Tex-Mex', 'Gujarati', 'Middle Eastern', 'Frozen Yogurt', 'Portugu
ese', 'German', 'Steak', 'European', 'Hyderabadi', 'Malaysian', 'Healthy Food', 'Mongolian', 'Sichuan', 'Lucknowi', 'Bubble Tea', 'Raw Meats', 'Continental', 'Odia', 'Konkan', 'Afghani', 'Kerala', 'Maharashtria
n', 'Bohri', 'Asian', 'Bar Food', 'Chinese', 'Garhwali', 'Tea', 'Poké', 'Tibetan', 'Vietnamese', 'Shake', 'Afghan', 'American', 'Goan', 'Kebab', 'Mangalorean', 'Himachali', 'Momos', 'Kashmiri', 'North Eastern',
'Cuisine Varies', 'Sushi', 'Desserts,', 'Roast Chicken', 'Bangladeshi', 'Rajasthani', 'Arabian', 'Bakery,', 'Thai', 'Parsi', 'Wraps', 'Cafe', 'Pasta', 'Mughlai', 'Beverages,', 'Mandi', 'Mithai', 'Hot dogs', 'Nort
h Indian', 'Mediterranean', 'Burmese', 'Chettinad', 'French', 'Bakery', 'Italian', 'Modern Indian', 'Russian', 'Naga', 'Indian', 'Sri Lankan', 'Sindhi', 'Japanese', 'Desserts', 'Finger Food', 'Panini', 'Malwani',
'Assamese', 'Greek', 'Brazilian', 'Grilled Chicken', 'Tamil', 'Nepalese', 'Drinks Only', 'Oriental', 'Cantonese', 'Belgian', 'Charcoal Chicken', 'Seafood', 'Mexican', 'Coffee', 'Korean', 'Moroccan', 'Awadhi', 'Bi
hari', 'South Indian', 'Beverages'}
```

In [42]:
```python
cuisine_counts = df['Cuisine'].str.split(', ').explode().value_counts()
top_15_cuisines = cuisine_counts.head(15)
plot_data = pd.DataFrame({'Cuisine':top_15_cuisines.index , 'Frequency':top_15_cuisines.values})
colors = px.colors.qualitative.Set3
```

In [43]:
```python
fig = px.bar(plot_data, x='Frequency', y='Cuisine', color='Cuisine',
            color_discrete_sequence=colors,
            orientation='h', labels={'Frequency': 'Frequency'},
            title='Top 15 Cuisines', width=800, height=500)

fig.update_layout(showlegend=False)
fig.show()
```

In [44]:
```python
from wordcloud import WordCloud
import matplotlib.pyplot as plt

cuisine_text = ', '.join(df['Cuisine'].dropna())
wordcloud = WordCloud( width=800,height=400,background_color='white').generate(cuisine_text)
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```



In [45]:
```python
import matplotlib.pyplot as plt
import seaborn as sns

average_price = df['Average Price']
df['Rating'] = pd.to_numeric(df['Rating'], errors='coerce')

# Define rating categories
def categorize_rating(rating):
    if rating < 3:
        return '<3'
    elif 3 <= rating < 4:
        return '3-4'
    else:
        return '4+'

df['Rating Category'] = df['Rating'].apply(categorize_rating)

# Define a color palette for each category
palette = {'<3': 'red', '3-4': 'orange', '4+': 'green'}

plt.figure(figsize=(10, 6))
sns.scatterplot(x=average_price, y=df['Rating'], hue=df['Rating Category'], palette=palette, alpha=0.7)
plt.title('Relation between Average Price and Rating')
plt.xlabel('Average Price')
plt.ylabel('Rating')
plt.legend(title='Rating Category')
plt.show()
```


In [ ]: