

# Business problem

What is the distribution of the quantity of pizzas ordered and What are the mean, median, and standard deviation of the total\_price of orders and Provide a summary of unit\_price for different pizza sizes and categories. What are the most popular pizza categories and Which pizza ingredients are most commonly ordered and Perform a t-test to compare the mean total\_price of orders for two different pizza categories (e.g., "Veg" vs. "Non-Veg") and Use a chi-square test to examine the relationship between pizza\_size and pizza\_category.

## importing Libraries

```
In [144]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import statistics
import scipy.stats as stats
import warnings
warnings.filterwarnings('ignore')
from scipy.stats import chi2_contingency
```

## uploading dataset

```
In [3]: df = pd.read_csv('pizza_sales.csv')
```

```
Out [4]:
```

	pizza_id	order_id	pizza_name_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	3	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	4	ital_supr_l	1	2015-01-01	11:57:40	20.75	20.75	L	Supreme	Calabrese Salsami, Capocollo, Tomatoes, Red Oni...	The Italian Supreme Pizza
4	5	5	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
...	...	...	...	...	...	...	...	...	...	...	...	...
48615	48616	21348	ckn_alfredo_m	1	2015-12-31	21:23:30	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	21:23:30	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	21:23:30	12.00	12.00	S	Classic	Tomatoes, Anchovies, Green Olives, Red Onions...	The Neapolitana 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

48620 rows x 12 columns

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

	pizza_id	order_id	pizza_name_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	3	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	4	ital_supr_l	1	2015-01-01	11:57:40	20.75	20.75	L	Supreme	Calabrese Salsami, Capocollo, Tomatoes, Red Oni...	The Italian Supreme Pizza
4	5	5	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

```
In [6]: df.tail()
```

	pizza_id	order_id	pizza_name_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	21:23:30	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	21:23:30	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	21:23:30	12.00	12.00	S	Classic	Tomatoes, Anchovies, Green Olives, Red Onions...	The Neapolitana 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

```
In [7]: df.info()
```

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

## statistical numbers

```
In [47]: df.describe()
```

```
Out [47]:
```

	pizza_id	order_id	quantity	order_date	unit_price	total_price
count	48620.000000	48620.000000	48620.000000	48620.000000	48620.000000	48620.000000
mean	24310.500000	10701.479761	1.019622	2015-06-29 11:03:43.611602560	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	2015-01-01 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	16035.529361	6180.119770	0.143077	NaN	3.621769	4.437398

```
In [9]: df['order_date'] = pd.DataFrame(df['order_date'])
```

```
In [10]: df['order_date'] = pd.to_datetime(df['order_date'])
```

```
In [12]: df.columns
```

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

```
In [24]: df['order_time'] = pd.DataFrame(df['order_time'])
```

```
In [26]: df['order_time'] = pd.to_datetime(df['order_time'], format='%m:%M:%S').dt.time
```

```
In [28]: df.dtypes
```

```
Out [28]:
pizza_id      int64
order_id      int64
pizza_name_id object
quantity      int64
order_date    datetime64[ns]
order_time    object
unit_pr_price float64
total_price   float64
pizza_size    object
pizza_category object
pizza_ingredients object
pizza_name    object
dtype: object
```

```
In [37]: df.isnull().sum()
```

```
Out [37]:
pizza_id      0
order_id      0
pizza_name_id 0
quantity      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 [31]: df.columns
```

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

```
In [33]: Numerical_data = df[['pizza_id', 'order_id', 'quantity', 'unit_price', 'total_price']]
```

```
In [40]: Numerical_data.describe()
```

```
Out [40]:
```

	pizza_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	16035.529361	6180.119770	0.143077	3.621769	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 [34]: Categorical_data = df[['pizza_name_id', 'pizza_size', 'pizza_category', 'pizza_ingredients', 'pizza_name']]
```

```
In [35]: Numerical_data
```

```
Out [35]:
```

	pizza_id	order_id	quantity	unit_price	total_price
0	1	1	1	13.25	13.25
1	2	2	1	16.00	16.00
2	3	3	2	18.50	18.50
3	4	4	2	18.50	20.75
4	5	5	2	16.00	16.00
...	...	...	...	...	...
48615	48616	21348	1	16.75	16.75
48616	48617	21348	1	17.95	17.95
48617	48618	21348	1	12.00	12.00
48618	48619	21349	1	20.25	20.25
48619	48620	21350	1	12.75	12.75

48620 rows x 5 columns

```
In [36]: Categorical_data
```

```
Out [36]:
```

	pizza_name_id	pizza_size	pizza_category	pizza_ingredients	pizza_name
0	hawaiian_m	M	Classic	Sliced Ham, Pineapple, Mozzarella Cheese	The Hawaiian Pizza
1	classic_dlx_m	M	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppers...	The Classic Deluxe Pizza
2	five_cheese_l	L	Veggie	Mozzarella Cheese, Provolone Cheese, Smoked Go...	The Five Cheese Pizza
3	ital_supr_l	L	Supreme	Calabrese Salsami, Capocollo, Tomatoes, Red O...	The Italian Supreme Pizza
4	mexicana_m	M	Veggie	Tomatoes, Red Peppers, Jalapeno Peppers, Red O...	The Mexicana Pizza
...	...	...	...	...	...
48615	ckn_alfredo_m	M	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms, A...	The Chicken Alfredo Pizza
48616	four_cheese_l	L	Veggie	Ricotta Cheese, Gorgonzola Piccante Cheese, Mo...	The Four Cheese Pizza
48617	napolitana_s	S	Classic	Tomatoes, Anchovies, Green Olives, Red Onions...	The Neapolitana Pizza
48618	mexicana_l	L	Veggie	Tomatoes, Red Peppers, Jalapeno Peppers, Red O...	The Mexicana Pizza
48619	bbq_ckn_s	S	Chicken	Barbecued Chicken, Red Peppers, Green Peppers...	The Barbecue Chicken Pizza

48620 rows x 5 columns

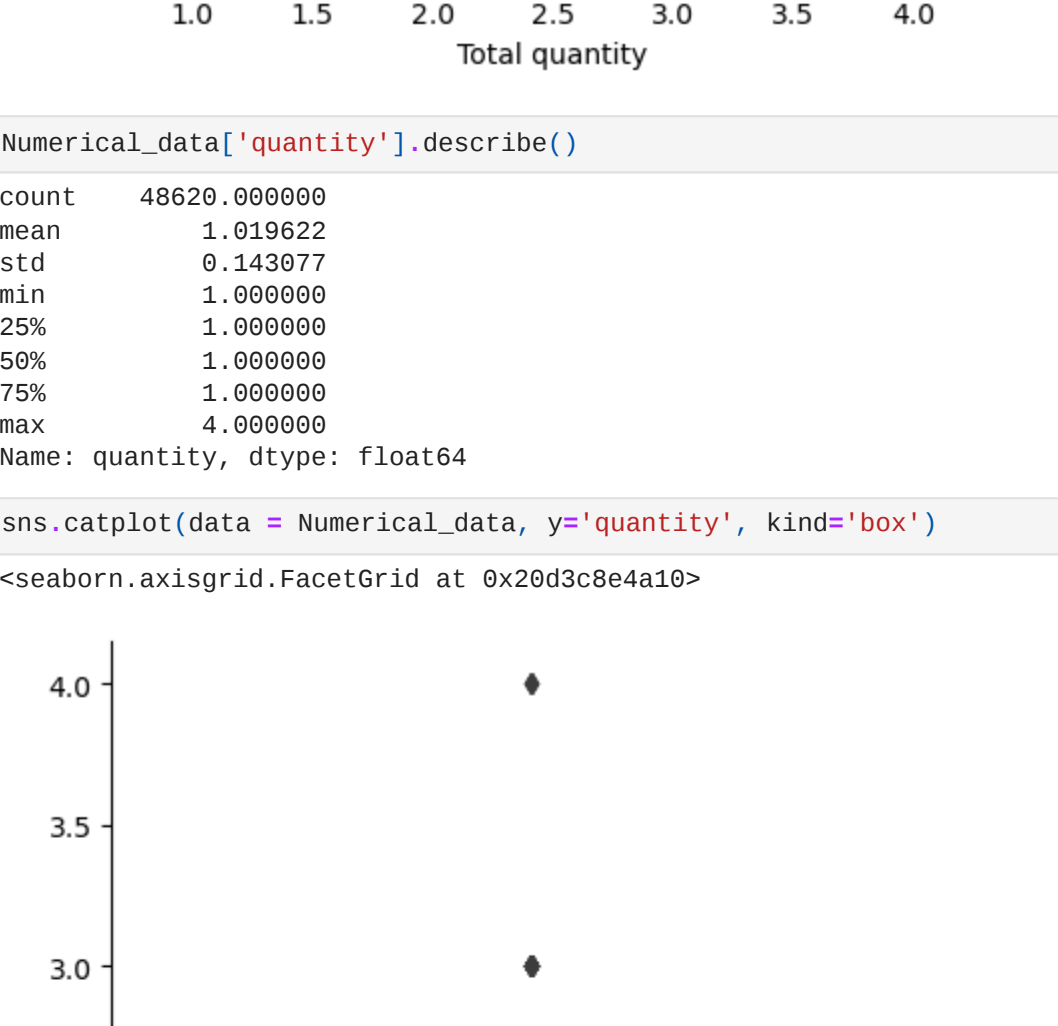
## Distribution of the quantity of pizzas ordered?

```
In [95]: plt.figure(figsize=(8,6))
sns.distplot(Numerical_data['quantity'], bins=25)
plt.title('Distribution of the quantity of pizzas ordered')
plt.xlabel('Total quantity')
plt.ylabel('Number of orders')
```

```
Out [95]:
```

Text(0.5, 1.0, 'Distribution of the quantity of pizzas ordered')

<Figure size 800x600 with 8 Axes>



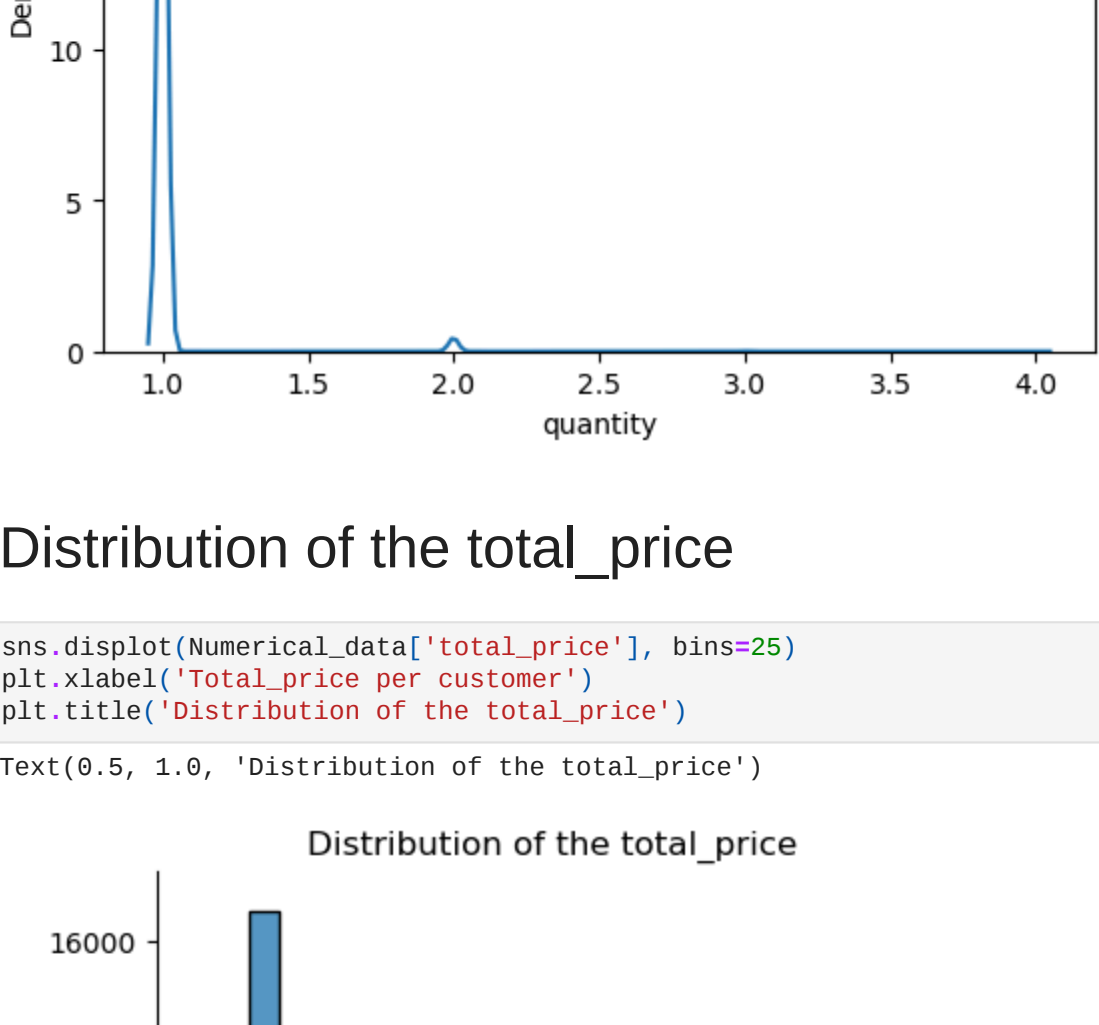
```
In [49]: Numerical_data['quantity'].describe()
```

```
Out [49]:
```

	count	mean	std	min	25%	50%	75%	max
quantity	48620.000000	1.019622	0.143077	1.000000	1.000000	1.000000	1.000000	4.000000

Name: quantity, dtype: float64

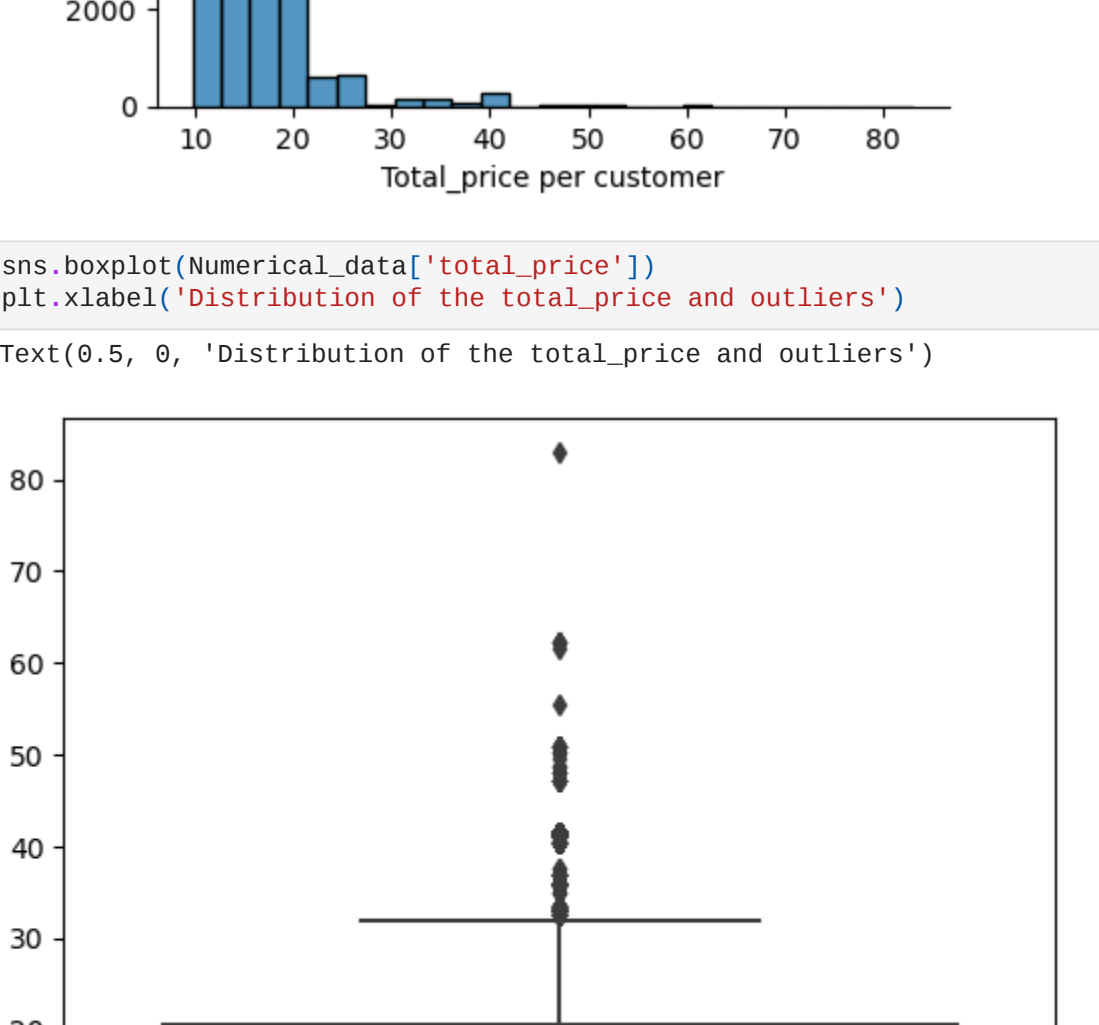
```
In [51]: sns.catplot(data = Numerical_data, y='quantity', kind='box')
<seaborn.axisgrid.FacetGrid at 0x20d3d8e4a10>
```



```
In [52]: sns.kdeplot(Numerical_data['quantity'])
```

```
Out [52]:
```

<Axes: xlabel='quantity', ylabel='density'>



## Distribution of the total\_price

```
In [96]: sns.distplot(Numerical_data['total_price'], bins=25)
plt.xlabel('Distribution of the total price')
plt.ylabel('Number of orders')
```

```
Out [96]:
```

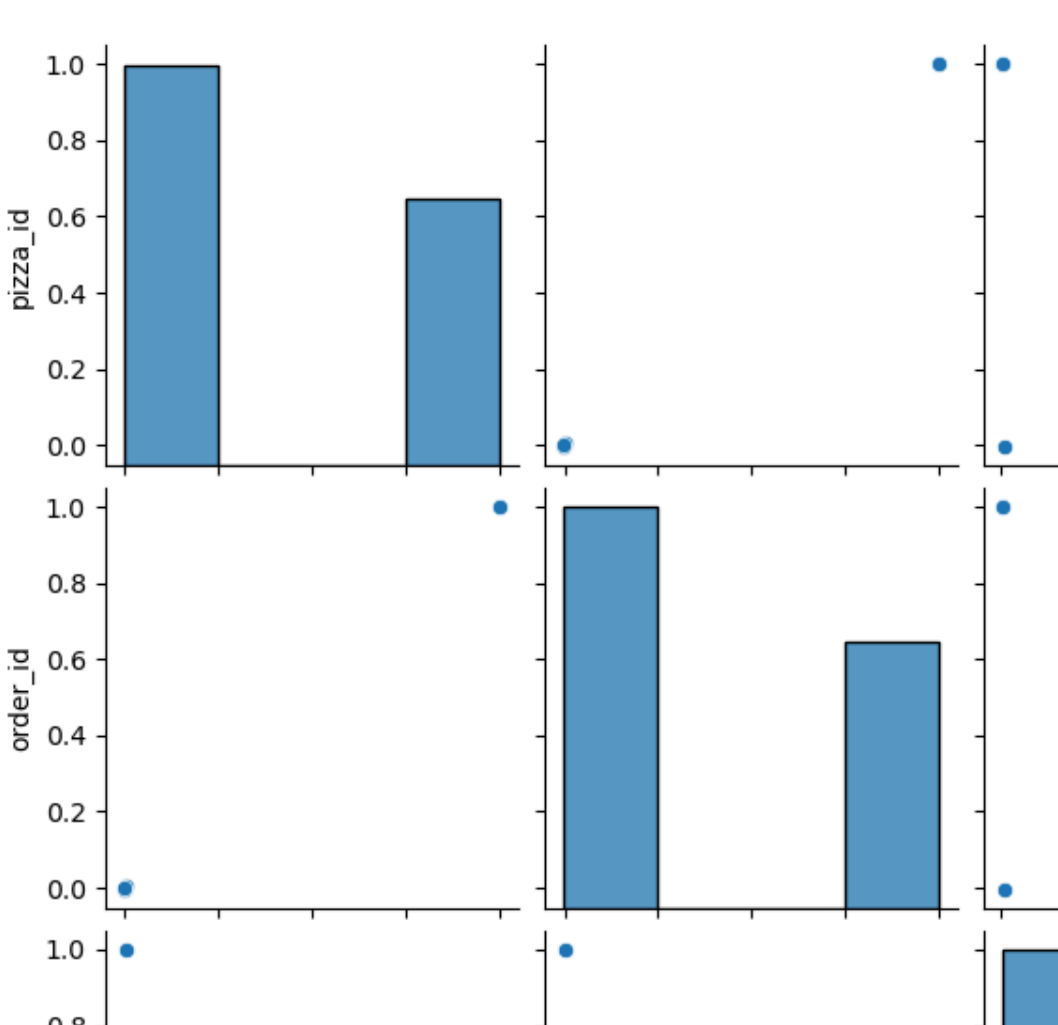
Text(0.5, 1.0, 'Distribution of the total price')



```
In [99]: sns.boxplot(Numerical_data['total_price'])
plt.xlabel('Distribution of the total price and outliers')
```

```
Out [99]:
```

Text(0.5, 0, 'Distribution of the total price and outliers')



```
In [100]: Numerical_data['total_price'].describe()
```

```
Out [100]:
```

	count	mean	std	min	25%	50%	75%	max
total_price	48620.000000	16.821474	4.437398	9.750000	12.750000	16.500000	20.500000	83.000000

Name: total\_price, dtype: float64

## correlation

```
In [104]: correlation = Numerical_data.corr()
```

```
In [105]: correlation
```

```
Out [105]:
```

	pizza_id	order_id	quantity	unit_price	total_price
pizza_id	1.000000	0.999990	0.003639	-0.002885	-0.002947
order_id	0.999990	1.000000	0.003621	-0.003323	-0.003089
quantity	0.003639	0.003621	1.000000	0.007142	0.541926
unit_price	-0.002885	-0.003323	0.007142	1.000000	0.836087
total_price	-0.002947	-0.003089	0.541926	0.836087	1.000000

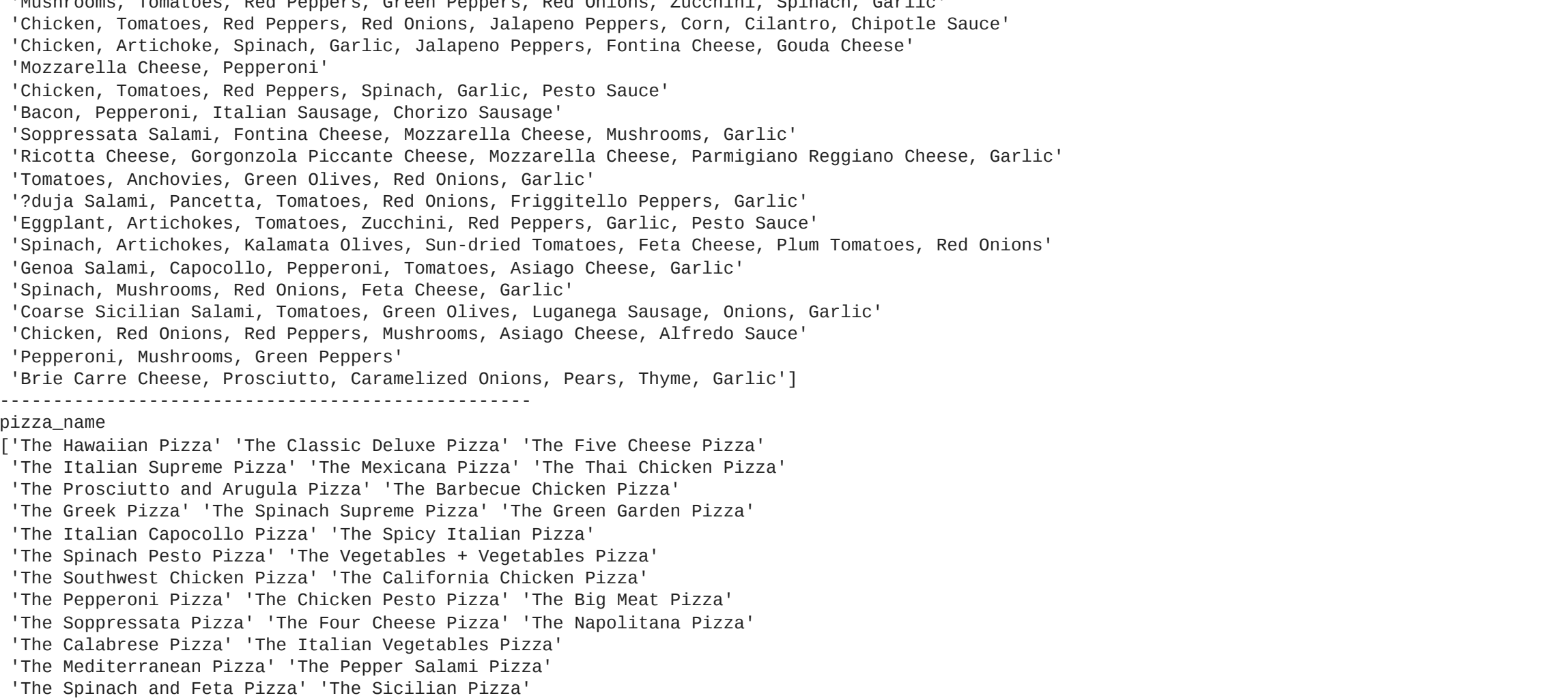
```
In [106]: sns.heatmap(correlation)
```

```
Out [106]:
```

<Axes: >



```
In [107]: sns.pairplot(correlation)
<seaborn.axisgrid.PairGrid at 0x20d46059b10>
```



## Categorical\_data Distribution

```
In [85]: Categorical_data.describe(include = 'object')
```

```
Out [85]:
```

	pizza_name_id	pizza_size	pizza_category	pizza_ingredients	pizza_name
count	48620	48620	48620	48620	48620
unique	91	5	4	32	32
top	big_meat_s	L	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppers...	The Classic Deluxe Piz
freq	1811	18526	14579	2416	2416

```
In [87]: for col in Categorical_data.describe(include = 'object').columns:
print(col)
print(Categorical_data[col].unique())
print("\n")

pizza_name_id
['The Hawaiian Pizza' 'The Classic Deluxe Pizza' 'The Five Cheese Pizza' 'The Italian Supreme Pizza' 'The Mexicana Pizza' 'The Neapolitana Pizza' 'The Neapolitana Supreme Pizza' 'The Pepp
```

```
In [89]: Categorical_data.columns
```

```
Out [89]:
```

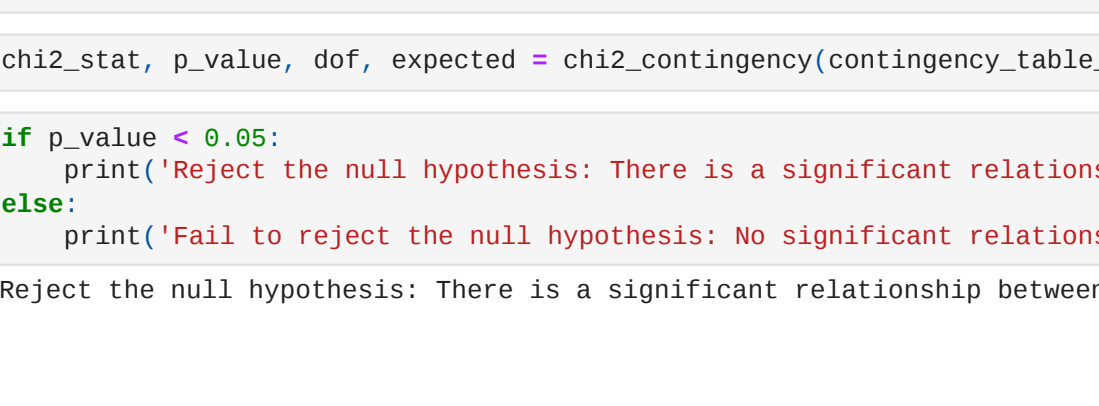
Index(['pizza\_name\_id', 'pizza\_size', 'pizza\_category', 'pizza\_ingredients', 'pizza\_name'], dtype='object')

```
In [112]: Numerical_data.columns
```

```
Out [112]:
```

Index(['pizza\_id', 'order\_id', 'quantity', 'unit\_price', 'total\_price'], dtype='object')

```
In [126]: sns.barplot(data = df, x = 'pizza_category', y = 'total_price')
<Axes: xlabel='pizza_category', ylabel='total_price'>
```



```
In [133]: sns.barplot(data = df, x = 'pizza_size', y = 'total_price')
<Axes: xlabel='pizza_size', ylabel='total_price'>
```



## T-Test: Comparing Mean Total Price for "Veg" vs. "Non-Veg" Pizza Categories

```
In [134]: veg_orders = df[df['pizza_category'] == 'Veggie']['total_price']
```

```
In [135]: nonveg_orders = df[df['pizza_category'] == 'Chicken']['total_price']
```

```
In [136]: ttest_pvalue = stats.ttest_ind(veg_orders, nonveg_orders)
```

```
In [137]: print(ttest_pvalue)
```

TtestResult(statistic=-1.862995471785688, pvalue=7.416431263993803e-105, df=22282.0)

```
In [139]: if pvalue < 0.05:
print('Reject the null hypothesis: There is a significant difference in mean total_price between Veg and Non-Veg pizza orders')
else:
print('Fail to reject the null hypothesis: No significant difference in mean total_price between Veg and Non-Veg pizza orders')
```

Reject the null hypothesis: There is a significant difference in mean total\_price between Veg and Non-Veg pizza orders

## Chi-Square Test: Relationship Between Pizza Size and Pizza Category

```
In [141]: contingency_table_size_category = pd.crosstab(df['pizza_size'], df['pizza_category'])
```

```
In [142]: chi2_stat, p_value, dof, expected = chi2.contingency(contingency_table_size_category)
```

```
In [143]: if p_value < 0.05:
print('Reject the null hypothesis: There is a significant relationship between pizza size and pizza category')
else:
print('Fail to reject the null hypothesis: No significant relationship between pizza size and pizza category')
```

Reject the null hypothesis: There is a significant relationship between pizza size and pizza category