PYTHON REINFORCEMENT PROJECT

RESTAURANT SALES DATA

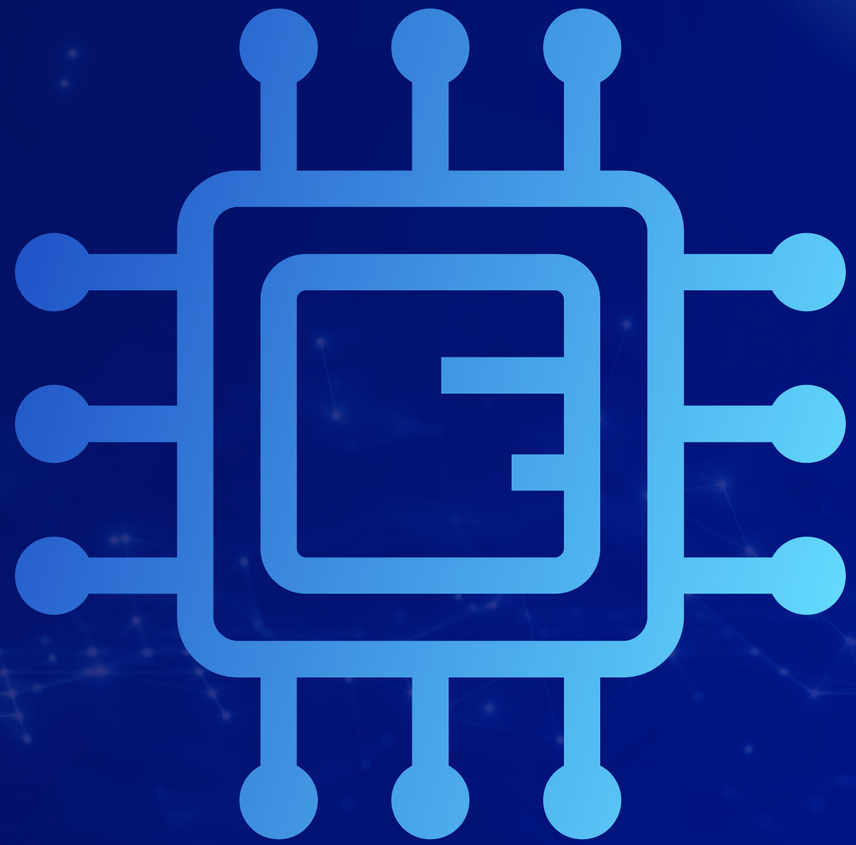
HARRISON RALPH I

DA&DS-JULY-2025

COHORT-A

AGENDA

- Introduction
- Data understanding
- Data cleaning
- Visual aids
- Handling missing values
- Outliers
- Data visualization
- Conclusion & Recommendations



INTRODUCTION

The restaurant sales dataset contains information about orders, sales, and customer transactions in a restaurant over a certain period. It is typically used to analyze business performance, customer behavior, and sales trends.

The dataset usually includes the following key features:

- Order ID: Unique identifier for each order.
- Date/Time: The timestamp when the order was placed.
- Category: Type of food item (e.g., Main Dishes, Beverages, Desserts).
- Item Name: Name of the food or drink ordered.
- Price: Price of a single item.
- Quantity: Number of items ordered.
- Order Total: Total amount for the order (Price × Quantity).
- Payment Method: Mode of payment (e.g., Cash, Card, UPI).

DATA UNDERSTANDING

The dataset contains detailed records of restaurant transactions, providing insights into sales performance and customer behavior. Understanding the data involves examining its structure, types of features, and their relevance.

<u>Purpose of Data Understanding:</u>

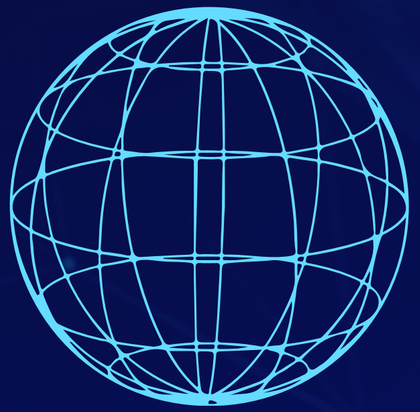
- Identify key metrics for analysis, such as total sales, average order value, and peak hours.
- Determine relationships between features (e.g., which categories generate the most revenue).
- Prepare the dataset for cleaning, visualization, and further analysis.

```
[8]: import pandas as pd
     import numpy as np
     df = pd.read_csv('restaurant_sales_data.csv')
     df.head()
```

[8]:

	Order ID	Customer ID	Category	Item	Price	Quantity	Order Total	Order Date	Payment Method
0	ORD_705844	CUST_092	Side Dishes	Side Salad	3.0	1.0	3.0	2023-12-21	Credit Card
1	ORD_338528	CUST_021	Side Dishes	Mashed Potatoes	4.0	3.0	12.0	2023-05-19	Digital Wallet
2	ORD_443849	CUST_029	Main Dishes	Grilled Chicken	15.0	4.0	60.0	2023-09-27	Credit Card
3	ORD_630508	CUST_075	Drinks	NaN	NaN	2.0	5.0	2022-08-09	Credit Card
4	ORD_648269	CUST_031	Main Dishes	Pasta Alfredo	12.0	4.0	48.0	2022-05-15	Cash

```
[9]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17534 entries, 0 to 17533
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Order ID        17534 non-null  object
 1   Customer ID     17534 non-null  object
 2   Category        17534 non-null  object
 3   Item            15776 non-null  object
 4   Price           16658 non-null  float64
 5   Quantity        17104 non-null  float64
 6   Order Total     17104 non-null  float64
 7   Order Date      17534 non-null  object
 8   Payment Method  16452 non-null  object
dtypes: float64(3), object(6)
memory usage: 1.2+ MB
```



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

[11]:

	Price	Quantity	Order Total
count	16658.000000	17104.000000	17104.000000
mean	6.586325	3.014149	19.914494
std	4.834652	1.414598	18.732549
min	1.000000	1.000000	1.000000
25%	3.000000	2.000000	7.500000
50%	5.000000	3.000000	15.000000
75%	7.000000	4.000000	25.000000
max	20.000000	5.000000	100.000000

```
[12]: df.head(10)
```

[12]:

	Order ID	Customer ID	Category	Item	Price	Quantity	Order Total	Order Date	Payment Method
0	ORD_705844	CUST_092	Side Dishes	Side Salad	3.0	1.0	3.0	2023-12-21	Credit Card
1	ORD_338528	CUST_021	Side Dishes	Mashed Potatoes	4.0	3.0	12.0	2023-05-19	Digital Wallet
2	ORD_443849	CUST_029	Main Dishes	Grilled Chicken	15.0	4.0	60.0	2023-09-27	Credit Card
3	ORD_630508	CUST_075	Drinks	NaN	NaN	2.0	5.0	2022-08-09	Credit Card
4	ORD_648269	CUST_031	Main Dishes	Pasta Alfredo	12.0	4.0	48.0	2022-05-15	Cash
5	ORD_381680	CUST_031	Main Dishes	Salmon	18.0	5.0	90.0	2022-07-20	Digital Wallet
6	ORD_270994	CUST_071	Side Dishes	Garlic Bread	4.0	5.0	20.0	2022-08-19	Credit Card
7	ORD_146656	CUST_077	Main Dishes	NaN	15.0	3.0	45.0	2023-02-15	Cash
8	ORD_428611	CUST_083	Desserts	NaN	6.0	2.0	12.0	2023-12-16	Cash
9	ORD_743636	CUST_085	Main Dishes	Vegetarian Platter	14.0	5.0	70.0	2022-08-07	NaN

DATA CLEANING

- Fixing Rows & Columns
- Handling Missing Values
- Data type conversion
- Checking outliers

FIX ROWS AND COLUMNS

This step ensures the dataset has a clean structure before moving into deeper analysis.

- Find duplicates (df.duplicated())

```
[14]: #duplicates
      df.duplicated()

[14]: 0        False
      1        False
      2        False
      3        False
      4        False
               ...
      17529    False
      17530    False
      17531    False
      17532    False
      17533    False
      Length: 17534, dtype: bool
```

HANDLING MISSING VALUES

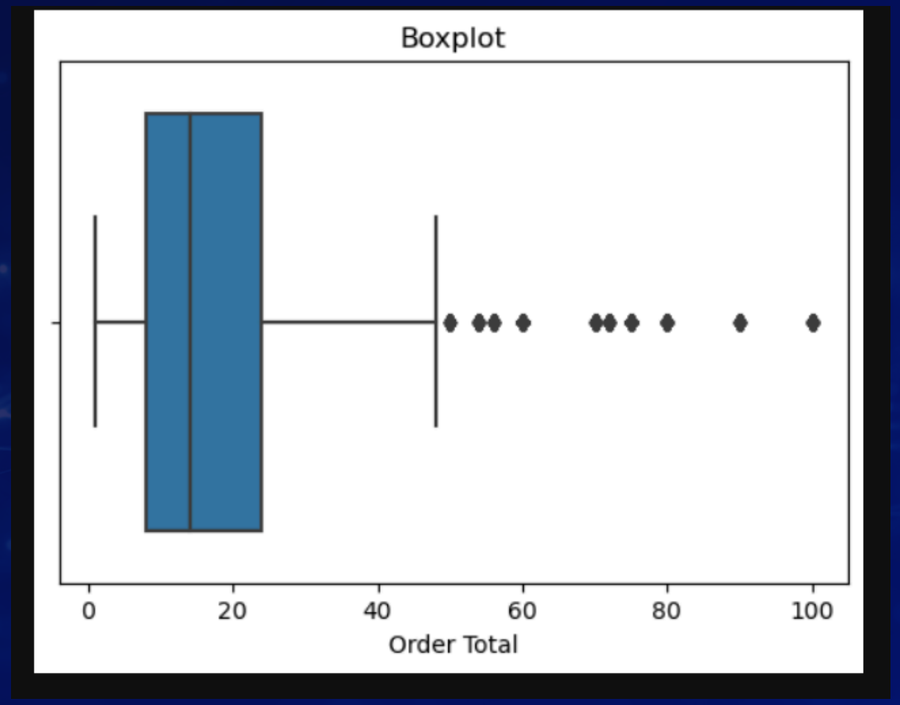
- Check for missing or null values in all columns.
- Impute missing numerical values using mean, median, or mode.
- Impute missing categorical values using mode or a placeholder like “Unknown”.

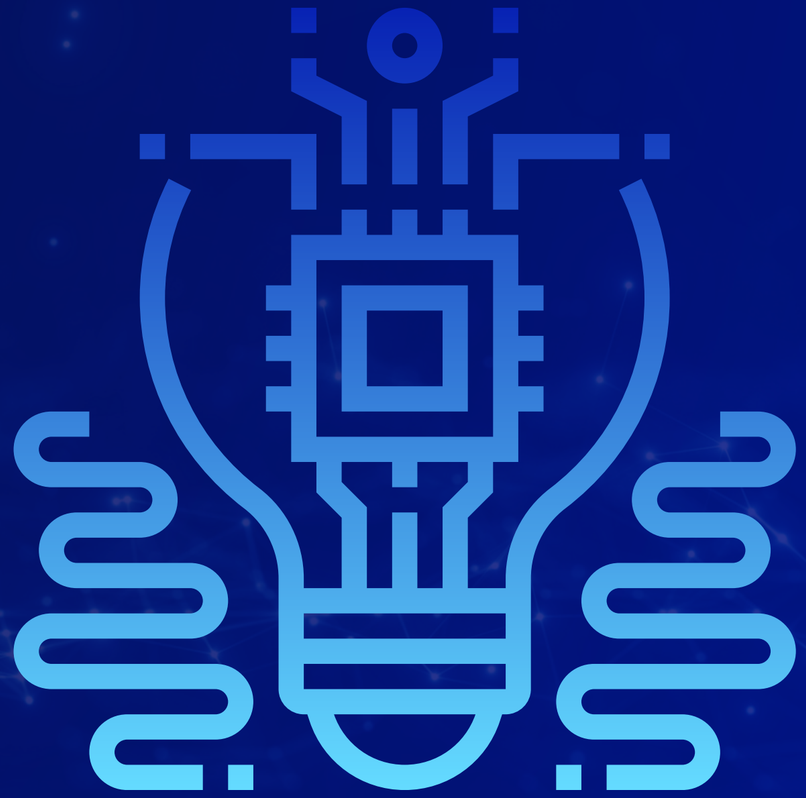
```
[21]: df['Price'] = df['Price'].fillna(df['Price'].mode()[0])
      df['Price']

[21]: 0         3.0
      1         4.0
      2        15.0
      3         5.0
      4        12.0
               ...
      17529     5.0
      17530     5.0
      17531     5.0
      17532     4.0
      17533     7.0
      Name: Price, Length: 17534, dtype: float64
```

Outlier Detection and Treatment

Outliers in the Order Total column were visually detected using a boxplot, where individual points beyond the normal range indicate values much higher than the majority of the data.





BASIC UNDERSTANDING

- The df.info() command provides a summary of the DataFrame, showing the total number of entries, column names, data types, and count of non-null values for each column.
- This summary helps in identifying column data types, presence of missing values, and the overall structure of the dataset before further analysis.

```
[9]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17534 entries, 0 to 17533
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Order ID        17534 non-null  object
 1   Customer ID     17534 non-null  object
 2   Category        17534 non-null  object
 3   Item            15776 non-null  object
 4   Price           16658 non-null  float64
 5   Quantity        17104 non-null  float64
 6   Order Total     17104 non-null  float64
 7   Order Date      17534 non-null  object
 8   Payment Method  16452 non-null  object
dtypes: float64(3), object(6)
memory usage: 1.2+ MB
```

STATISTIC ANALYSIS

Statistical Analysis is the process of collecting, organizing, analyzing, interpreting, and presenting data to uncover patterns, relationships, and trends. It helps in making informed decisions, testing hypotheses, and predicting future outcomes. By applying descriptive and inferential statistical methods, analysts can summarize data effectively, identify anomalies, and draw meaningful conclusions to support research or business objectives.

Four types of statistic analysis:

- T-test
- Z-test
- Chi-square test
- ANOVA test

ONE SAMPLE T-TEST

A one-sample t-test was conducted to determine whether the mean Order Total differs from the population mean of 3.

[28]:
```
# ONE TEST
from scipy import stats


sample_data = df['Order Total']

# Known population mean
population_mean = 3

# Perform the one-sample t-test
t_statistic, p_value = stats.ttest_1samp(sample_data, population_mean)

# Output the results
print(f"T-statistic: {t_statistic}")
print(f"P-value: {p_value}")

# Determine the result
alpha = 0.05
if p_value < alpha:
    print("Reject the null hypothesis: The sample mean is significantly different from the population mean.")
else:
    print("Fail to reject the null hypothesis: The sample mean is not significantly different from the population mean.")
```

```
T-statistic: 119.40807382716588
P-value: 0.0
Reject the null hypothesis: The sample mean is significantly different from the population mean.
```

TWO SAMPLE Z-TEST

A two-sample t-test compared Main Dishes and Side Dishes. The result showed a significant difference between their mean order totals.

[30]:
```
# Z TEST
import scipy.stats as stats

from statsmodels.stats.weightstats import ztest

Main_Dishes = df[df['Category'] == 'Main Dishes']['Price']
Side_Dishes= df[df['Category'] == 'Side Dishes']['Price']

z_stat, p_val = ztest(Main_Dishes, Side_Dishes)

print("Z-statistic:", z_stat)
print("p-value:", p_val)

# Define significance level (alpha)
alpha = 0.05

# Make a decision based on the p-value
if p_val < alpha:
    print("Reject the null hypothesis: Means are significantly different.")
else:
    print("Fail to reject the null hypothesis: Means are not significantly different.")
```

```
Z-statistic: 169.2755650490628
p-value: 0.0
Reject the null hypothesis: Means are significantly different.
```

DATA VISUALIZATION

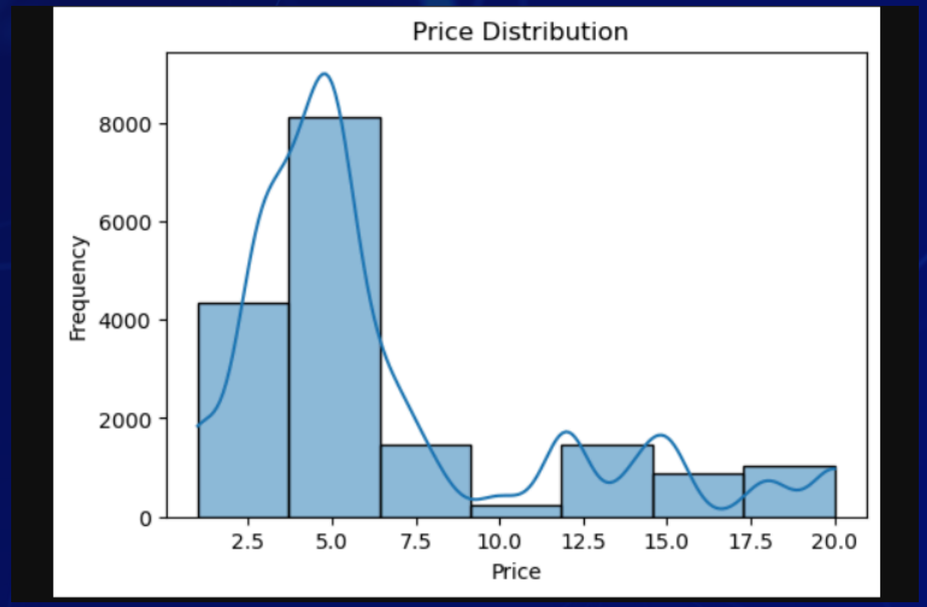
Data visualization is a key part of Exploratory Data Analysis (EDA) that helps turn raw restaurant data into actionable insights using graphs and charts.

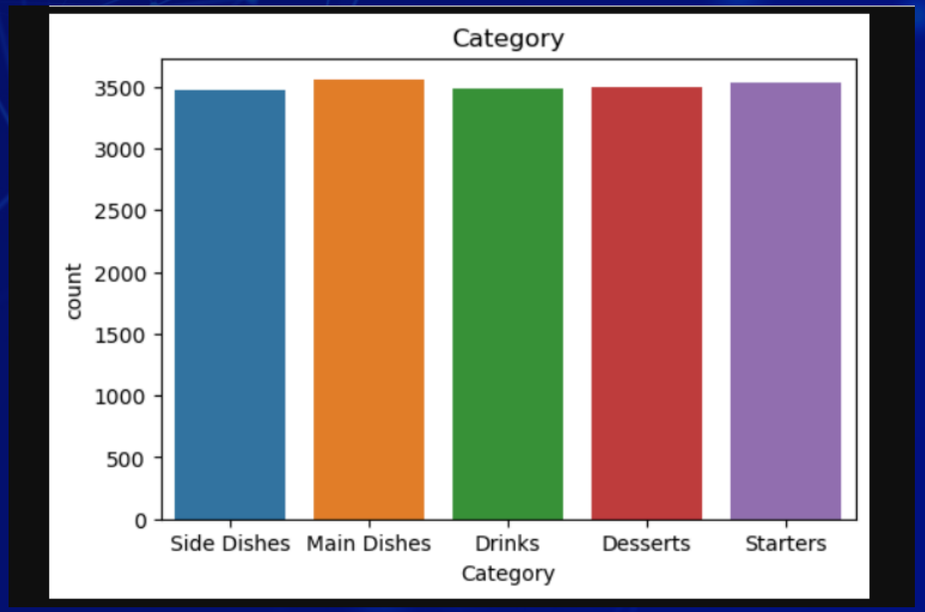
In this project, visualizations are used to study restaurant sales, order patterns, and customer preferences.

- Univariate analysis (histograms, count plots) shows the distribution of single features such as Order Total, Quantity, and Payment Method.
- Bivariate analysis (boxplots, violin plots) highlights relationships, e.g., how Time of Day or Payment Method affects total sales, or how Quantity impacts Order Total.
- Multivariate analysis (heatmaps) reveals correlations among multiple features like Price, Quantity, and Order Total.

UNIVARIENT ANALYSIS

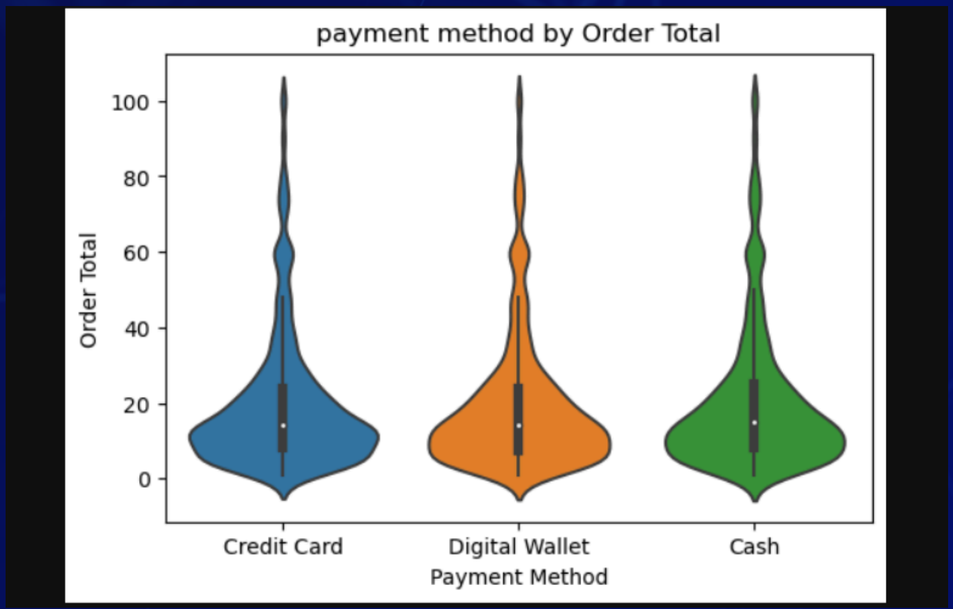
- The count plot shows the number of records in each product category, including Side Dishes, Main Dishes, Drinks, Desserts, and Starters.
- Category counts are relatively balanced, indicating a well-distributed dataset across different menu types.
- A balanced distribution helps avoid bias and ensures diverse sales insights across categories.
- The price histogram reveals that most items are priced between 2 and 7 units, with fewer items at higher price ranges.
- The concentration of lower-priced items suggests an accessible menu, guiding pricing strategies and promotional decisions.





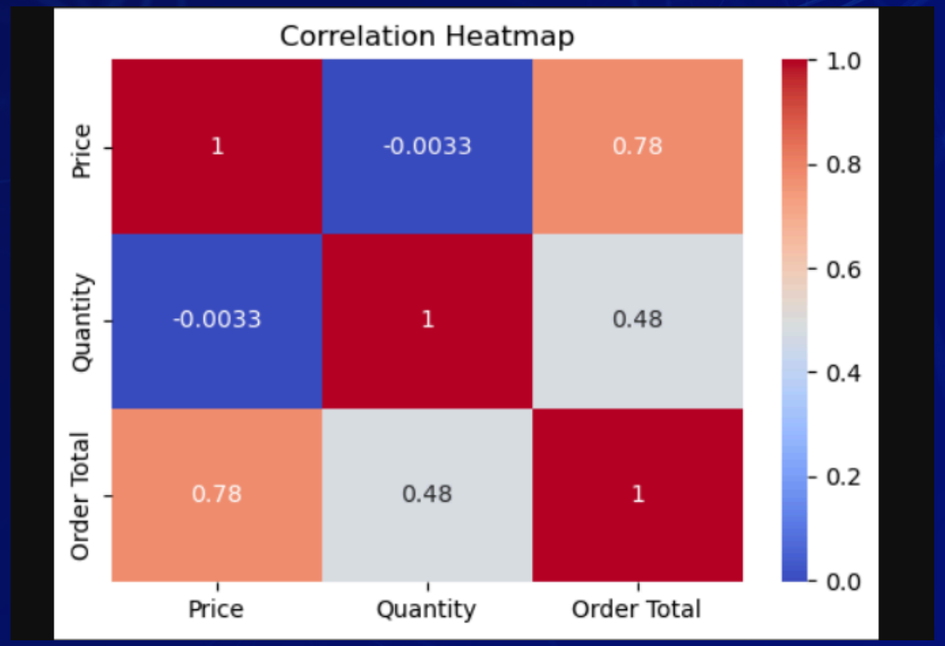
BIVARIENT ANALYSIS

- The violin plot shows the distribution of order totals across payment methods: Credit Card, Digital Wallet, and Cash.
- All payment methods have similar distributions with most orders in lower total ranges and some high-value outliers.
- This indicates customers use different payment options consistently regardless of order size.
- The scatter plot confirms that all payment methods process transactions across a wide range of order values.
- Insights highlight that payment flexibility is important, with no single method dominating large or small purchases.



MULTIVARIENT ANALYSIS

- The heatmap visualizes the strength and direction of correlations between Price, Quantity, and Order Total.
- Order Total has a strong positive correlation with Price (r = 0.78), indicating higher-priced items increase order value.
- There is a moderate positive correlation between Quantity and Order Total (r = 0.48), meaning buying more items also raises total sales.
- Price and Quantity show almost no correlation (r = -0.0033), suggesting item price and quantity purchased are independent.
- This analysis highlights that both price and quantity influence revenue, but they do so largely independently in the dataset



OVERALL INSIGHTS FROM ANALYSIS

- **High Revenue Items:** Main Dishes generate the most revenue per order.
- **Top Selling Category:** Beverages and Side Dishes have the highest number of orders.
- **Order Quantity:** Most orders contain 1–3 items.
- **Price vs Quantity:** Higher-priced items are ordered less frequently.
- **Order Total:** Total order values mostly fall in the lower range; few large orders exist.
- **Payment Method:** Cash is the most commonly used payment method.
- **Peak Times:** Orders are highest during lunch and dinner hours.
- **Correlation:** Price and Order Total show a strong positive correlation.
- **Outliers**: Some orders have unusually high quantities or totals, indicating bulk orders.
- **Customer Preference:** Most customers order combinations of Main Dishes with Side Dishes or Beverages.

CONCLUSION

- The analysis of the Restaurant Sales Dataset provides meaningful insights into the restaurant’s sales performance and customer behavior. From the data, it is clear that certain categories, such as Main Dishes, contribute the most to revenue, while others like Desserts and Beverages have moderate sales.

- Peak ordering times are observed during lunch and dinner hours, which indicates when staff and inventory should be optimized. The preferred payment methods show that most customers pay via cards, followed by cash and digital payments.