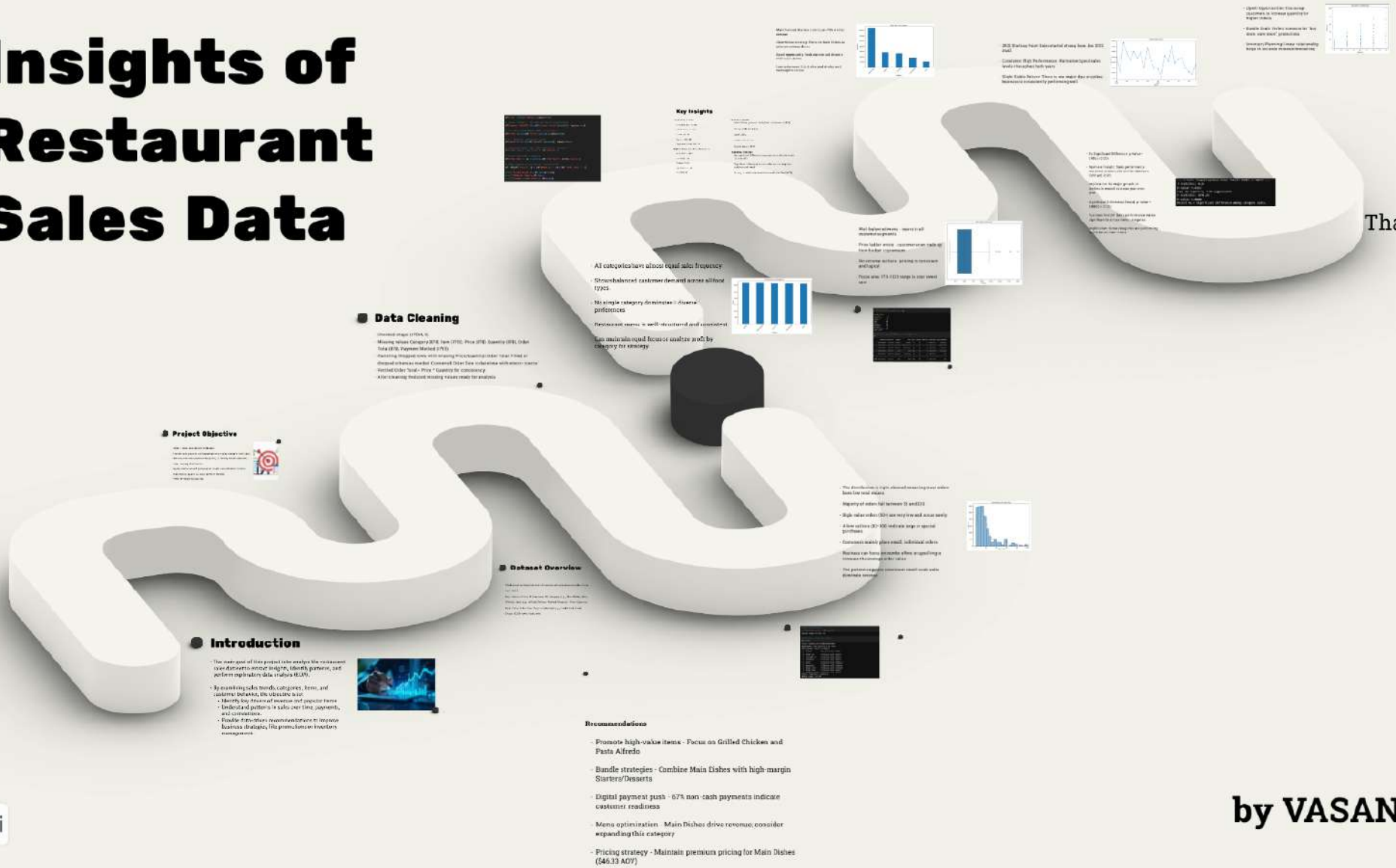
Insights of Restaurant Sales Data



Thank You !!

by VASANTH M

Introduction

- The main goal of this project is to analyze the restaurant sales dataset to extract insights, identify patterns, and perform exploratory data analysis (EDA).

- By examining sales trends, categories, items, and customer behavior, the objective is to:
  - Identify key drivers of revenue and popular items.
  - Understand patterns in sales over time, payments, and correlations.
  - Provide data-driven recommendations to improve business strategies, like promotions or inventory management.

Project Objective

- Import, clean, and explore the dataset.
- Explore sales patterns using aggregations (e.g., by category, item, date).
- Use visualizations (Seaborn/Matplotlib) to identify trends, sales over time, category distribution.
- Apply correlation and grouping for insights on restaurant decision-making (e.g., quality vs. sales, payment trends).
- Perform hypothesis testing

Dataset Overview

- The dataset contains detailed information about restaurant orders from 2022-2023.
- Key columns: Order ID, Customer ID, Category (e.g., Main Dishes, Side Dishes), Item (e.g., Grilled Chicken, Mashed Potatoes), Price, Quantity, Order Total, Order Date, Payment Method (e.g., Credit Card, Cash).
- Shape: 17,534 rows, 9 columns.

Data Cleaning

- Checked shape: (17534, 9).
- Missing values: Category (878), Item (1753), Price (878), Quantity (878), Order Total (878), Payment Method (1753),
- Handling: Dropped rows with missing Price/Quantity/Order Total. Filled or dropped others as needed. Converted Order Date to datetime with errors='coerce'.
- Verified Order Total = Price * Quantity for consistency.
- After cleaning: Reduced missing values, ready for analysis.

```python
# Shape of the dataset
print(f"Dataset shape: {df.shape}")
```

```
Dataset shape: (17534, 9)
```

```python
# Data types and non-null counts
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
```

```python
# Check missing values
print("\nMissing values:\n", df.isnull().sum())
```

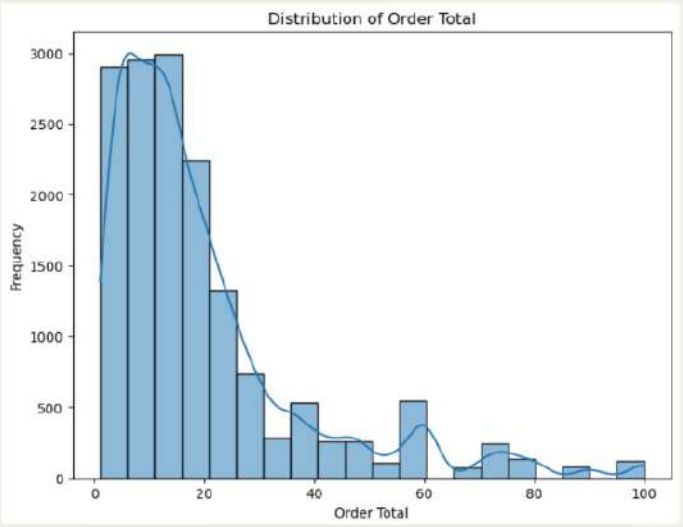
```
Missing values:
 Order ID            0
Customer ID          0
Category             0
Item              1758
Price              876
Quantity           430
Order Total        430
Order Date           0
Payment Method    1082
dtype: int64
```

```python
# Convert 'Order Date' to datetime
df['Order Date'] = pd.to_datetime(df['Order Date'], errors='coerce')
df
```

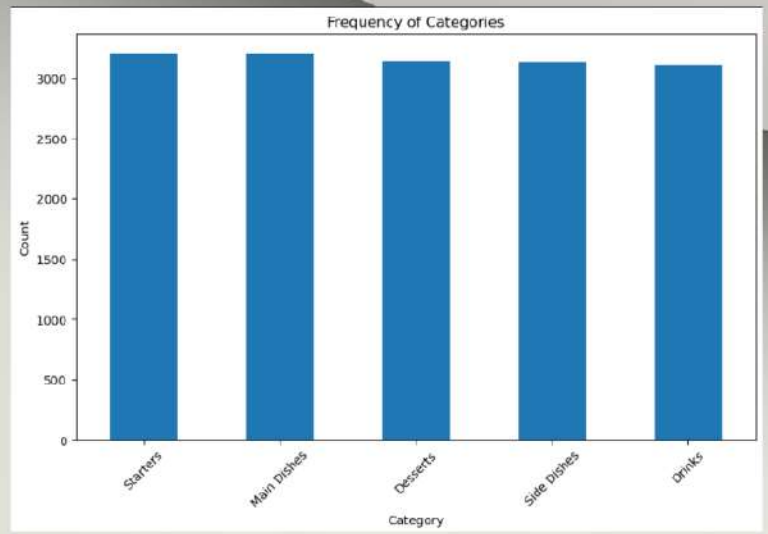
	Order ID	Customer ID	Category	Item	Price	Quantity	Order Total	Order Date	Payment Method
0	ORD_705844	CUST_092	Side Dishes	Side Salad	3.0	1.0	3.0	2023-12-21	Credit Card
1	ORD_338528	CUST_021	Side Dishes	Mashed Potatoes	4.0	3.0	12.0	2023-05-19	Digital Wallet
2	ORD_443849	CUST_029	Main Dishes	Grilled Chicken	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
...	...	...	...	...	...	...	...	...	...
17529	ORD_320102	CUST_021	Drinks	NaN	NaN	4.0	4.0	2023-12-23	Cash

```python
df['Item'].fillna('Unknown', inplace=True)

# 'Payment Method' → fill with mode (most frequent method)
df['Payment Method'].fillna(df['Payment Method'].mode()[0], inplace=True)

# Fill 'Price' with median (robust to outliers)
df['Price'].fillna(df['Price'].median(), inplace=True)

# Fill 'Quantity' with median as well
df['Quantity'].fillna(df['Quantity'].median(), inplace=True)

# Recalculate 'Order Total' where missing (or incorrect)
df['Order Total'] = df['Price'] * df['Quantity']

# Convert Order Date to datetime
df['Order Date'] = pd.to_datetime(df['Order Date'], errors='coerce')

# Remove negative or zero entries (invalid data)
df = df[(df['Price'] > 0) & (df['Quantity'] > 0) & (df['Order Total'] > 0)]

print("Missing Values:\n", df.isnull().sum())
print("\nDataset shape:", df.shape)
print("\nSample cleaned data:\n", df.head())
```

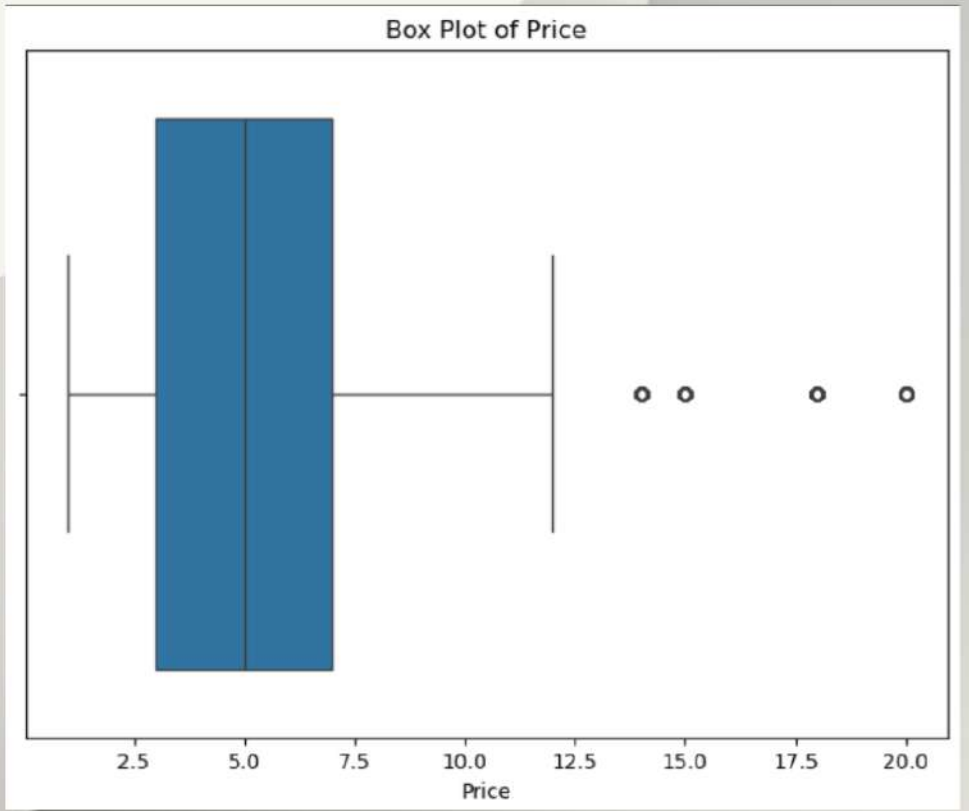
- The distribution is right-skewed, meaning most orders have low total values.
- Majority of orders fall between $5 and $20.
- High-value orders (50+) are very few and occur rarely.
- A few outliers (80–100) indicate large or special purchases.
- Customers mainly place small, individual orders.
- Business can focus on combo offers or upselling to increase the average order value.
- The pattern suggests consistent small-scale sales dominate revenue.



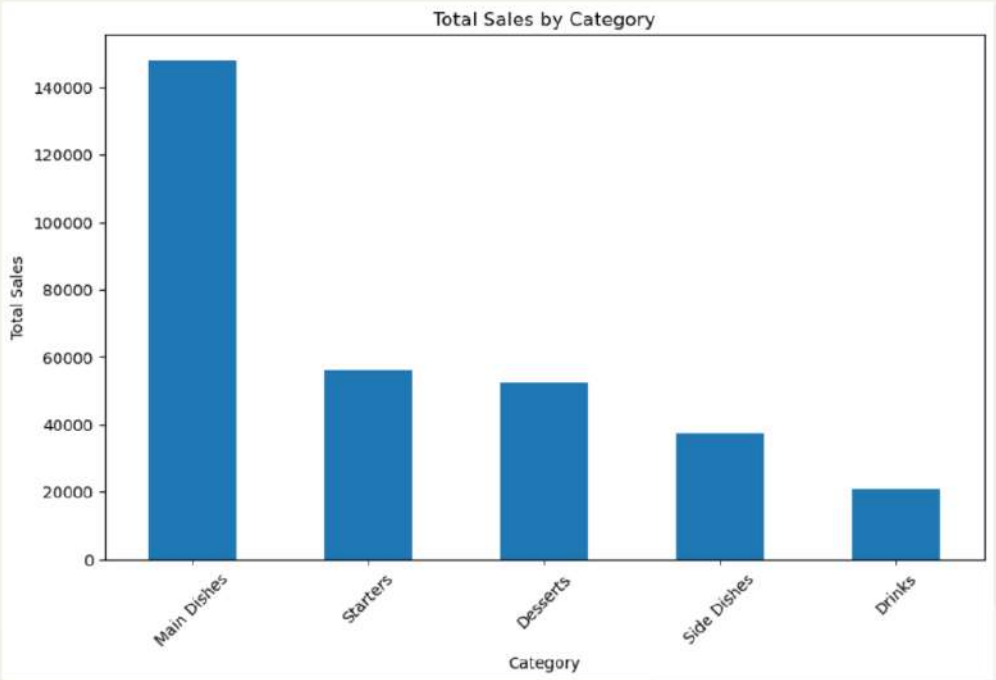
- All categories have almost equal sales frequency.
- Shows balanced customer demand across all food types.
- No single category dominates diverse preferences.
- Restaurant menu is well-structured and consistent.
- Can maintain equal focus or analyze profit by category for strategy.



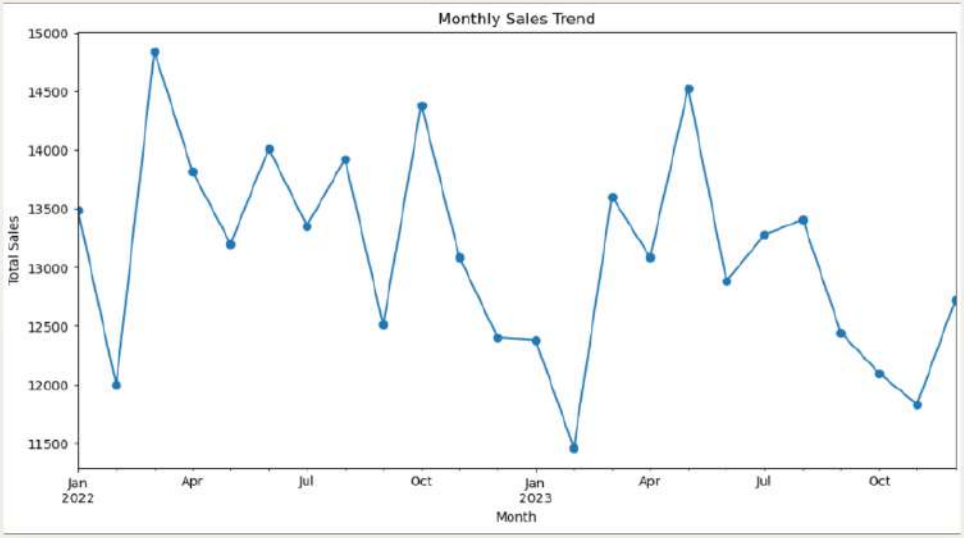
- Well-balanced menu - caters to all customer segments
- Price ladder exists - customers can trade up from budget to premium
- No extreme outliers - pricing is consistent and logical
- Focus area: 7.5-12.5 range is your sweet spot



- Main Dishes & Starters contribute ~70% of total revenue
- Clear winner strategy: Focus on Main Dishes as primary revenue driver
- Upsell opportunity: Push starters and desserts with main dishes
- Low performers: Side dishes and drinks need menu optimization



- 2022 Starting Point: Sales started strong from Jan 2022 itself
- Consistent High Performance: Maintained good sales levels throughout both years
- Slight Stable Pattern: There is one major dips or spikes - business is consistently performing well



- No Significant Difference: p-value = 0.416 (> 0.05)
- Business Insight: Sales performance remained statistically similar between 2022 and 2023
- Implication: No major growth or decline in overall revenue year-over-year
- Significant Differences Found: p-value = 0.0000 (< 0.05)
- Business Insight: Sales performance varies significantly across menu categories
- Implication: Some categories are performing much better than others

```
=== T-Test: Comparing Mean Order Totals (2022 vs 2023) ===
T-statistic: 0.81
P-value: 0.4163
Fail to reject H₀ → No significant
F-statistic: 4790.25
P-value: 0.0000
Reject H₀ → Significant difference among category sales.
```

Key Insights

Top 5 Items by Revenue:

- Grilled Chicken - $37,888
- Pasta Alfredo - $37,318
- Steak - $35,320
- Salmon - $32,608
- Vegetarian Platter - $13,798

Highest Average Order Value by Category:

- Main Dishes: $46.33
- Starters: $27.53
- Desserts: $16.76
- Side Dishes: $11.88
- Drinks: $6.66

Revenue Insights

- Main Dishes generate the highest total sales (~$140K)
- Payment Method Split:
- Cash: 35.6%
- Credit Card: 32.3%
- Digital Wallet: 32.1%

Statistical Findings

- No significant difference in average order value between 2022 vs 2023
- Significant differences exist in sales across categories (ANOVA confirmed)
- Strong correlation between Price and Order Total (0.79)

Recommendations

- Promote high-value items - Focus on Grilled Chicken and Pasta Alfredo
- Bundle strategies - Combine Main Dishes with high-margin Starters/Desserts
- Digital payment push - 67% non-cash payments indicate customer readiness
- Menu optimization - Main Dishes drive revenue; consider expanding this category
- Pricing strategy - Maintain premium pricing for Main Dishes ($46.33 AOV)

Thank You !!