

Swiggy Sales Analysis

Import Libraries

```
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 Data

```
In [10]: df = pd.read_excel("D:/Python Data Science Projects/swiggy_data.xlsx")
```

```
In [11]: df.head()
```

	State	City	Order Date	Restaurant Name	Location	Category	Dish Name	Price (INR)	Rating	Rating Count
0	Karnataka	Bengaluru	2025-06-29	Anand Sweets & Savouries	Rajarajeshwari Nagar	Snack	Butter Murukku-200gm	133.9	4.0	0
1	Karnataka	Bengaluru	2025-04-03	Srinidhi Sagar Deluxe	Kengeri	Recommended	Badam Milk	52.0	4.5	25
2	Karnataka	Bengaluru	2025-01-15	Srinidhi Sagar Deluxe	Kengeri	Recommended	Chow Chow Bath	117.0	4.7	48
3	Karnataka	Bengaluru	2025-04-17	Srinidhi Sagar Deluxe	Kengeri	Recommended	Kesar Bath	65.0	4.6	65
4	Karnataka	Bengaluru	2025-03-13	Srinidhi Sagar Deluxe	Kengeri	Recommended	Mix Raita	150.0	4.0	0

```
In [14]: df.tail()
```

```
Out[14]:
```

	State	City	Order Date	Restaurant Name	Location	Category	Dish Name	Price (INR)	Rating	Rating Count
197425	Sikkim	Gangtok	2025-01-25	Mama's Kitchen	Gangtok	Momos	Soya cheese chili momo ...	112.0	4.4	0
197426	Sikkim	Gangtok	2025-07-02	Mama's Kitchen	Gangtok	Momos	Kurkure momo fried ...	140.0	4.4	0
197427	Sikkim	Gangtok	2025-03-25	Mama's Kitchen	Gangtok	Momos	Chilli cheese momo	126.0	4.4	0
197428	Sikkim	Gangtok	2025-03-26	Mama's Kitchen	Gangtok	Momos	Veg Momos (8 Pcs)	85.0	4.4	0
197429	Sikkim	Gangtok	2025-03-27	Mama's Kitchen	Gangtok	Momos	Soya Momo	100.0	4.4	0

Metadata

```
In [15]: print("The number of rows:", df.shape[0])
```

The number of rows 197430

```
In [16]: print("The number of fields:", df.shape[1])
```

The number of fields 10

```
In [18]: df.info
```

```
Out[18]: <class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 197430 entries, 0 to 197430
```

```
Data columns (total 10 columns):
```

```
  #� State          : object
```

```
  #� City           : object
```

```
  #� Order Date    : datetime64[ns]
```

```
  #� Restaurant Name: object
```

```
  #� Location       : object
```

```
  #� Category        : object
```

```
  #� Dish Name       : object
```

```
  #� Price (INR)     : float64
```

```
  #� Rating          : float64
```

```
  #� Rating Count    : int64
```

```
[197430 rows x 10 columns]
```

Data Types

```
In [19]: df.dtypes
```

```
Out[19]: State          : object
```

```
  #� City           : object
```

```
  #� Order Date    : datetime64[ns]
```

```
  #� Restaurant Name: object
```

```
  #� Location       : object
```

```
  #� Category        : object
```

```
  #� Dish Name       : object
```

```
  #� Price (INR)     : float64
```

```
  #� Rating          : float64
```

```
  #� Rating Count    : int64
```

```
In [20]: df.describe()
```

```
Out[20]:
```

```
   #� Order Date      : Order Date
```

```
   #� Price (INR)     : Price (INR)
```

```
   #� Rating          : Rating
```

```
   #� Rating Count    : Rating Count
```

```
   #� count           : count
```

```
   #� mean            : mean
```

```
   #� min             : min
```

```
   #� 25%             : 25%
```

```
   #� 50%             : 50%
```

```
   #� 75%             : 75%
```

```
   #� max             : max
```

```
   #� std              : std
```

KPI's

Total Sales

```
In [21]: total_sales = df["Price (INR)"].sum()
```

print("Total Sales (INR):", round(total_sales,2))

Total Sales (INR): 53012505.77

Average Rating

```
In [23]: average_rating = df["Rating"].mean()
```

print("Average Rating:", round(average_rating,2))

Average Rating: 4.34

Average Order Value

```
In [26]: avg_order_value = df["Price (INR)"].mean()
```

print("Avg Order Value (INR):", round(avg_order_value,2))

Avg Order Value (INR): 268.51

Ratings Count

```
In [27]: ratings_count = df["Rating Count"].sum()
```

print("Ratings Count (INR):", round(ratings_count,2))

Ratings Count (INR): 5591574

Total Orders

```
In [28]: total_orders = len(df)
```

print("Total Orders:", round(total_orders,2))

Total Orders: 197430

CHARTS DESIGN

```
In [30]: # Convert Order Date to datetime
```

df["Order Date"] = pd.to_datetime(df["Order Date"])

```
# Create Year-Month column
```

df["YearMonth"] = df["Order Date"].dt.to_period("M").astype(str)

```
# Calculate monthly revenue
```

monthly_revenue = df.groupby("YearMonth")["Price (INR)"].sum().reset_index()

```
# Plot monthly revenue trend
```

plt.figure(figsize=(10,5))

plt.plot(monthly_revenue["YearMonth"], monthly_revenue["Price (INR)"])

plt.xlabel("Month")

plt.ylabel("Revenue (INR)")

plt.title("Monthly Revenue Trend")

plt.tight_layout()

plt.show()

Revenue Contribution: Veg vs Non-Veg

fig = px.pie(df,

values="Price (INR)",

name="Food Category",

hole=0.3,

title="Revenue Contribution: Veg vs Non-Veg",

color_discrete_sequence=["blue", "red"]

fig.update_traces(textinfo="percent+label",

pull=[0.05, 0])

fig.update_layout(height=500,

margin=dict(t=60, b=40, l=40, r=40))

fig.show()

Revenue Contribution: Veg vs Non-Veg

Non-Veg 37.1%

Veg 62.9%

