Food Order Sales Analysis: Revenue Trends by City, Influencer Impact, and Product Category

Objective

- Load the Zepto sales dataset using pandas
- Inspect the first few rows to understand the structure

```
import pandas as pd
```

```
food_orders = pd.read_csv("/Users/sourabh/Desktop/python/zepto_sales_dataset.csv")
```

```
food_orders.head(10)
```

	Product Name	Category	City	Original Price	Current Price	Discount	Orders	Total Revenue	Influencer Active
0	Britannia Cake	Snacks	Delhi	148	163	5	283	44714	No
1	Britannia Cake	Snacks	Pune	81	86	10	284	21584	Yes
2	Fortune Oil 1L	Grocery	Hyderabad	138	143	10	69	9177	No
3	Pepsi 500ml	Beverages	Delhi	127	127	10	83	9711	No
4	Aashirvaad Atta	Grocery	Chennai	34	49	10	169	6591	Yes
5	Amul Milk 500ml	Dairy	Delhi	149	159	0	246	39114	No
6	Britannia Cake	Snacks	Bangalore	82	87	0	254	22098	Yes
7	Amul Milk 500ml	Dairy	Bangalore	46	51	5	179	8234	No
8	Aashirvaad Atta	Grocery	Mumbai	137	137	10	268	34036	No
9	Maggi Noodles	Instant Food	Hyderabad	196	201	0	59	11859	Yes

Insights

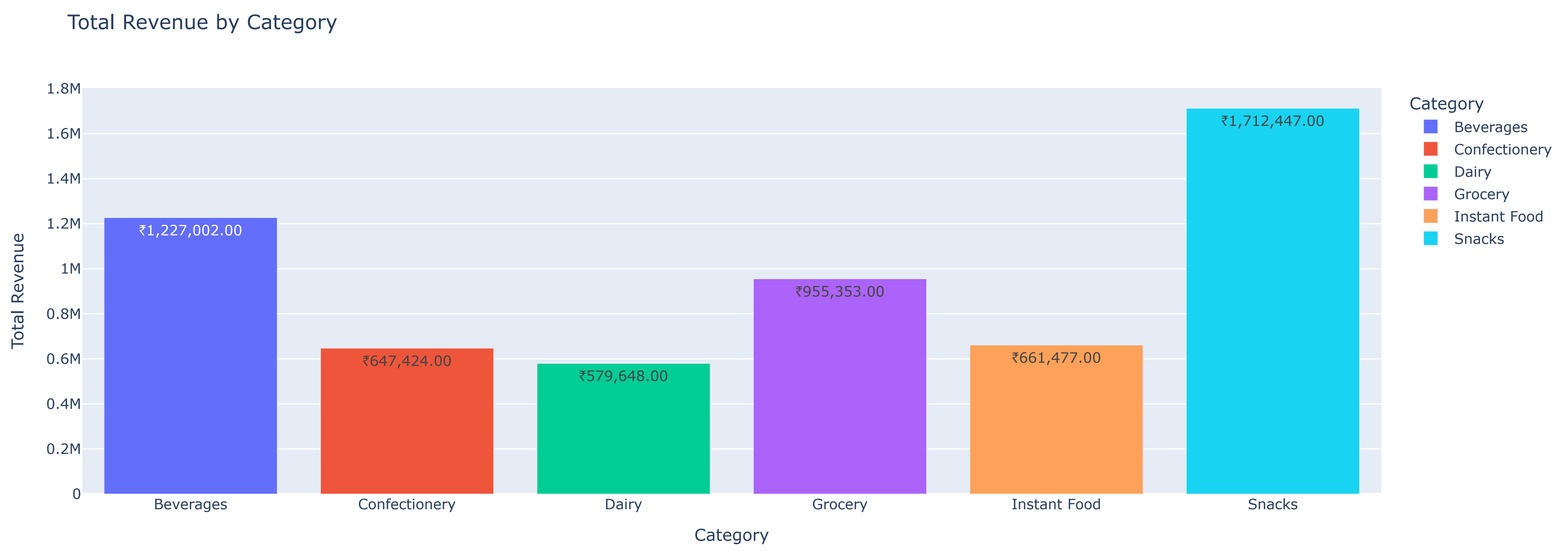
- Data looks clean and ready for analysis

Objective:

- Identify top revenue-generating categories

```
revenue_df = food_orders.groupby("Category")["Total Revenue"].sum().reset_index(name="Total Revenue Sum")
revenue_df = revenue_df.sort_values(by="Total Revenue Sum", ascending=False)
revenue_df["Total Revenue Sum"] = revenue_df["Total Revenue Sum"].apply(lambda x: f"₹{x:,.2f}")
import plotly.express as px
plot_df = food_orders.groupby("Category")["Total Revenue"].sum().reset_index()
plot_df["Formatted Revenue"] = plot_df["Total Revenue"].apply(lambda x: f"₹{x:,.2f}")

fig = px.bar(
    plot_df,
    x="Category",
    y="Total Revenue",
    text="Formatted Revenue",
    title="Total Revenue by Category",
    color="Category"
)
fig.show()
```



Insights

- Snacks and Beverages are the top revenue-generating categories.
- These categories can be prioritized in future marketing or stock decisions.

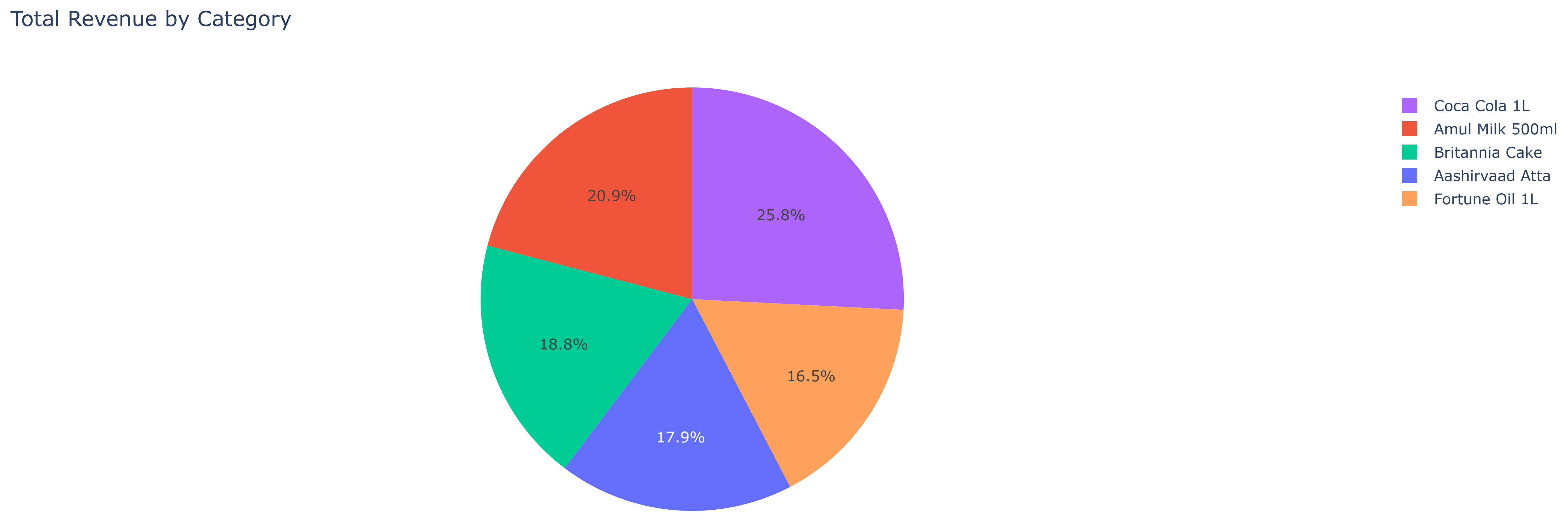
Objective

- Identify which individual products bring in the highest total revenue.

```
revenue_df = food_orders.groupby("Product Name")["Total Revenue"].sum().reset_index(name="Total Revenue Sum")
revenue_df = revenue_df.sort_values(by="Total Revenue Sum", ascending=False).head(5)
revenue_df["Total Revenue Sum"] = revenue_df["Total Revenue Sum"].apply(lambda x: f"₹{x:,.2f}")
import plotly.express as px
plot_df = food_orders.groupby("Product Name")["Total Revenue"].sum().reset_index().head(5)
plot_df["Formatted Revenue"] = plot_df["Total Revenue"].apply(lambda x: f"₹{x:,.2f}")

fig = px.pie(
    plot_df,
```

```
    names = "Product Name",
    values = "Total Revenue",
    title="Total Revenue by Category",
    color="Product Name"
)
fig.show()
```



Insights

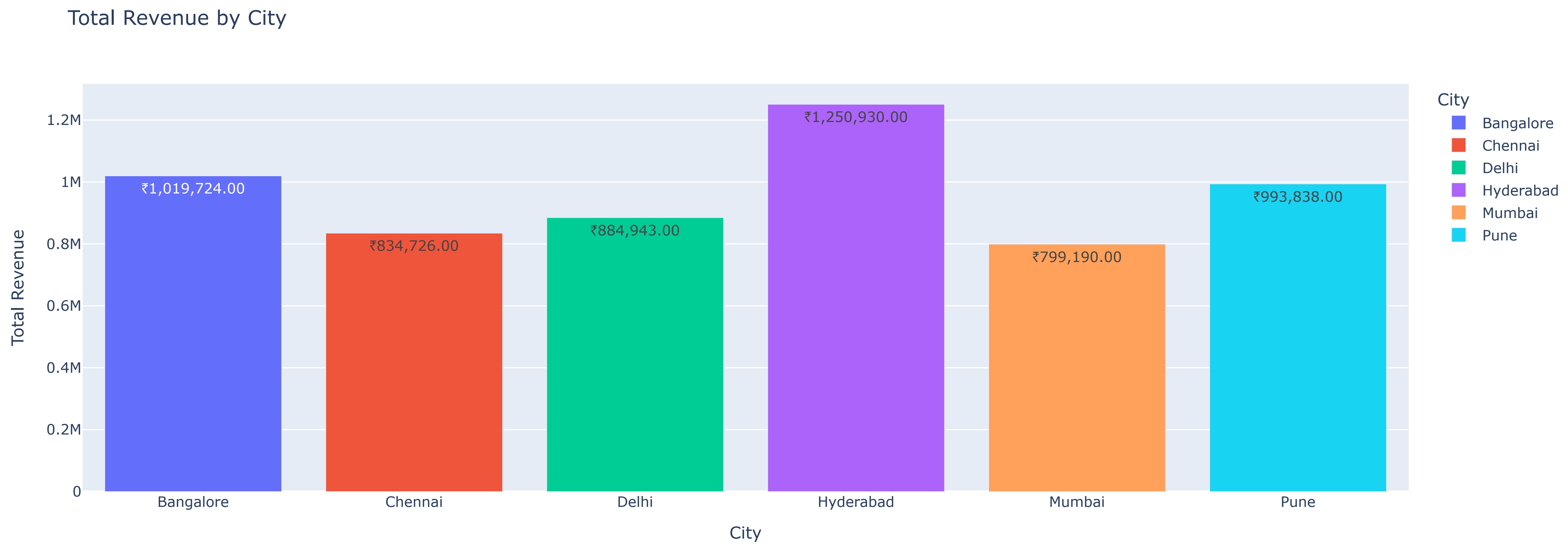
- Coca Cola 1L leads in revenue contribution.
- Beverages and dairy items dominate the top 5 list.
- Optimize product placements or promotions.

Objective

- Determine which cities generate the highest total revenue from orders.

```
revenue_df = food_orders.groupby("City")["Total Revenue"].sum().reset_index(name="Total Revenue Sum")
revenue_df = revenue_df.sort_values(by="Total Revenue Sum", ascending=False)
revenue_df["Total Revenue Sum"] = revenue_df["Total Revenue Sum"].apply(lambda x: f"₹{x:,.2f}")
import plotly.express as px
plot_df = food_orders.groupby("City")["Total Revenue"].sum().reset_index()
plot_df["Formatted Revenue"] = plot_df["Total Revenue"].apply(lambda x: f"₹{x:,.2f}")

fig = px.bar(
    plot_df,
    x="City",
    y="Total Revenue",
    text="Formatted Revenue",
    title="Total Revenue by City",
    color="City"
)
fig.show()
```



Insights

- Hyderabad is the top-performing city with over ₹1.25M in revenue.
- Bangalore and Pune also contribute significantly.
- This could guide regional marketing, inventory planning, or expansion strategies.

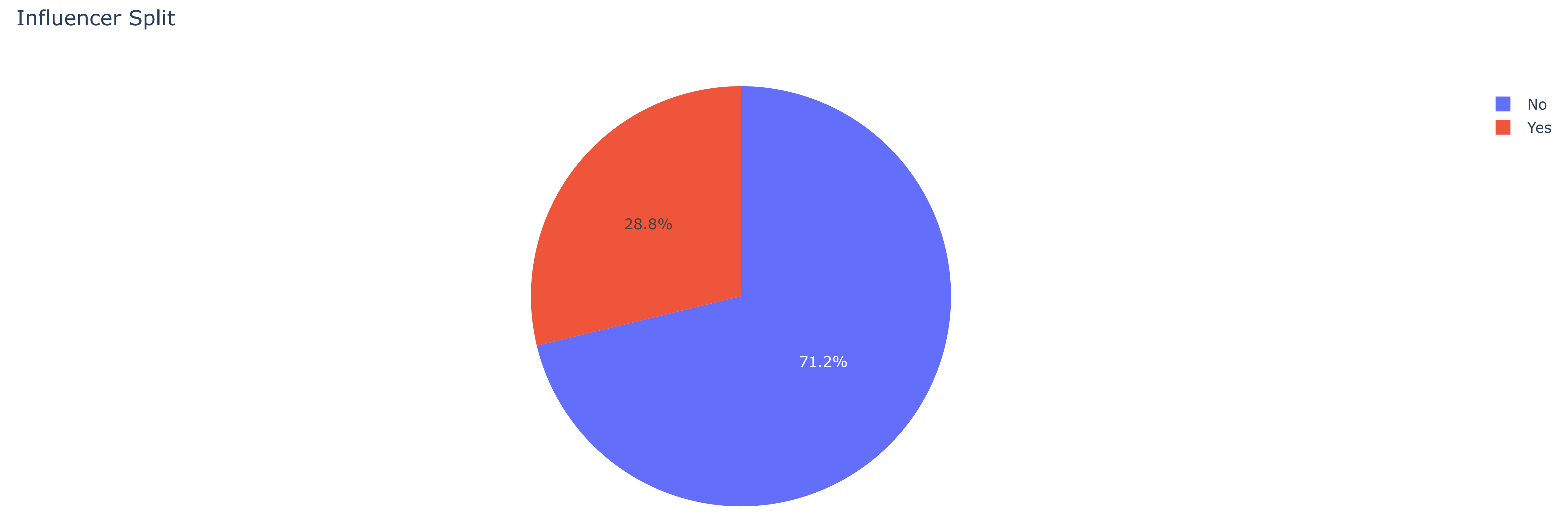
Objective

- Understand the revenue distribution between products promoted by influencers vs those not promoted.

```
revenue_df = food_orders.groupby("Influencer Active")["Total Revenue"].sum().reset_index(name="Total Revenue Sum")
revenue_df = revenue_df.sort_values(by="Total Revenue Sum", ascending=False)
revenue_df["Total Revenue Sum"] = revenue_df["Total Revenue Sum"].apply(lambda x: f"₹{x:,.2f}")
import plotly.express as px
plot_df = food_orders.groupby("Influencer Active")["Total Revenue"].sum().reset_index()
plot_df["Formatted Revenue"] = plot_df["Total Revenue"].apply(lambda x: f"₹{x:,.2f}")

fig = px.pie(
```

```
    plot_df,
    names = "Influencer Active",
    values = "Total Revenue",
    title="Influencer Split",
    color="Influencer Active"
)
fig.show()
```



Insights

- 71.2% of total revenue comes from non-influencer products.
- 28.8% is driven by influencer-promoted products.
- While influencers don't dominate revenue share, they still play a significant role and might have a stronger impact in specific categories.

Objective

- Analyze how order value varies across product categories.
- Identify categories with high variability, outliers, and consistent performance.

```
food_orders["Revenue per Order"] = food_orders["Total Revenue"] / food_orders["Orders"]
```

```
fig = px.box(
    food_orders,
    x="Category",
    y="Revenue per Order",
    title="Revenue per Order Distribution by Category",
    points="all",
    color="Category"
)
fig.update_layout(yaxis_title="Revenue per Order (₹)", xaxis_title="Category", showlegend=False)
fig.show()
```

Insights

- Beverages, Dairy, and Confectionery have higher median revenue per order.
- Grocery orders have more variability and some lower revenue orders.
- All categories show outliers, suggesting unique or bulk purchase behaviors.
- Snacks and Instant Food have more tightly packed distributions, indicating consistent pricing or order sizes.

```
food_orders["Discount %"] = (food_orders["Discount"] / food_orders["Current Price"]) * 100
```

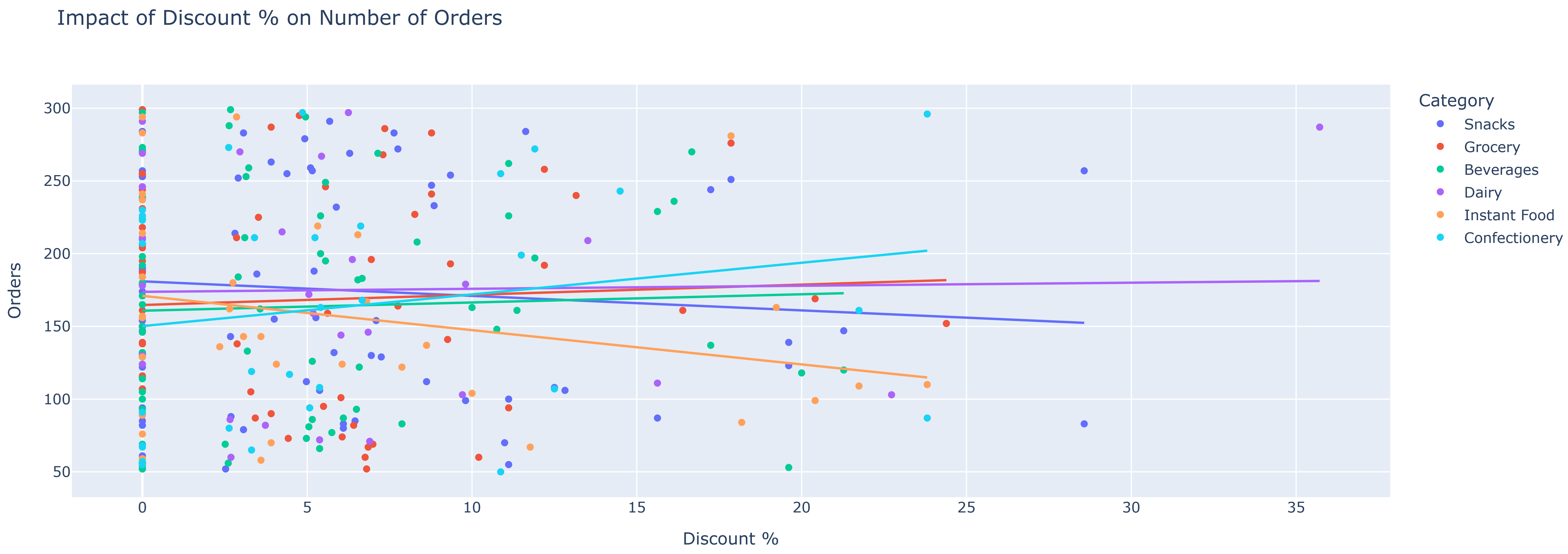
Objective

- To analyze how discount percentages affect order numbers across different product categories.

```
import plotly.express as px

fig = px.scatter(
    food_orders,
```

```
    x="Discount %",
    y="Orders",
    color="Category",
    title="Impact of Discount % on Number of Orders",
    trendline="ols"
)
fig.show()
```



Insights

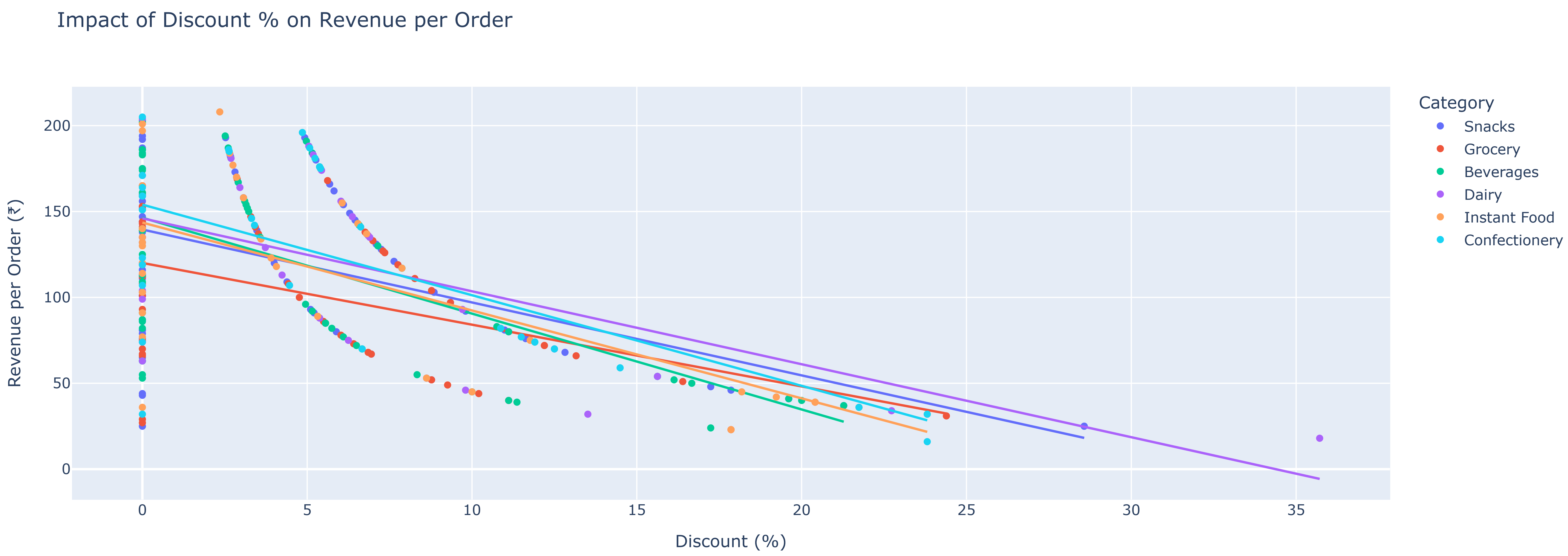
- The impact of discounts on order volume varies by category.
- Snacks and Confectionery see a positive correlation
- while others like Instant Food, Dairy see negative or flat trends, suggesting a need for targeted discount strategies.

Objective

- To assess how discount percentages impact revenue per order, segmented by product category.

```
fig.update_layout(xaxis_title="Discount (%)", yaxis_title="Number of Orders")

fig = px.scatter(
    food_orders,
    x="Discount %",
    y="Revenue per Order",
    color="Category",
    title="Impact of Discount % on Revenue per Order",
    trendline="ols"
)
fig.update_layout(xaxis_title="Discount (%)", yaxis_title="Revenue per Order (₹)")
fig.show()
```



Insights

- Increased discounts consistently decrease revenue per order across all categories.
- This highlights a trade-off between potentially higher order volume (for some categories) and lower per-order revenue.
- Necessitating a balanced discount approach.