# Zomato Orders Data Analysis Project

# Big Data – Case Study

**Subject – Big Data Analytics and Architecture**

# Prepared by Aryan Verma

# 1240259013

## Table of Contents

1. Project Overview..................................................1

2. Dataset Description..............................................2

3. Project Objectives...............................................3

4. Technologies Used................................................4

5. Steps Performed (Data Loading & Cleaning).........................5

6. Exploratory Analysis — Basic Metrics.............................6

7. Revenue & Category Analysis.....................................8

8. Customer Behaviour Analysis....................................10

9. Delivery Performance & Partners.................................12

10. City & Regional Insights......................................14

11. Discounts, Promotions & Their Impact..........................16

12. Advanced Analysis (LTV, Cohorts, Retention)....................17

13. Recommendations & Actionables................................19

14. Conclusion....................................................20

15. Appendix — Full SQL Queries and Sample Outputs................21

## 1. Project Overview

This project analyzes a large dataset of Zomato food delivery orders to extract actionable insights on customer ordering behaviour, restaurant performance, delivery efficiency, and revenue drivers. The analysis was conducted primarily using Cloudera HiveQL (SQL) on a distributed dataset, with supplementary work in Pandas/Excel for quick checks and Power BI/Tableau placeholders for visualization.

Primary aims: identify top restaurants and cuisines by revenue, reduce delivery times, optimize discounts, understand peak order hours and cancellation patterns, and recommend operational improvements to increase revenue and customer satisfaction.

## 2. Dataset Description

Dataset file name example: Zomato\_orders\_dataset.csv

Row count (example): ~250,000 transaction rows (order-level).

Columns and descriptions:

|  |  |
| --- | --- |
| Column Name | Description |
| order\_id | Unique identifier for each order (STRING) |
| order\_date\_time | Timestamp when order was placed (TIMESTAMP) |
| customer\_id | Unique customer identifier (STRING) |
| gender | Customer gender (Male/Female/Other) |
| age | Customer age (INT) |
| city | City of delivery (STRING) |
| restaurant\_id | Unique restaurant id (STRING) |
| restaurant\_name | Restaurant name (STRING) |
| cuisine\_type | Primary cuisine (e.g., North Indian, Chinese) |
| order\_amount | Total order amount charged to customer (DOUBLE) |
| item\_count | Number of items in order (INT) |
| discount | Discount applied (₹) (DOUBLE) |
| delivery\_fee | Delivery fee charged (₹) (DOUBLE) |
| payment\_method | Payment mode (Card/UPI/COD/Wallet) |
| order\_status | Delivered / Cancelled / Refunded (STRING) |
| delivery\_time\_min | Delivery time in minutes from pickup to drop (INT) |
| rating | Customer rating 1-5 (nullable DOUBLE) |
| tip\_amount | Tip given to delivery partner (₹) (DOUBLE) |
| delivery\_partner\_id | Delivery partner identifier (STRING) |
| coupon\_code | Coupon or promo code used (nullable STRING) |

Notes: The dataset contains missing values in 'rating' and 'coupon\_code'. 'delivery\_time\_min' is valid only for Delivered orders. 'order\_amount' should equal sum of items price minus discounts plus delivery\_fee; validate during cleaning.

## 3. Project Objectives

• Measure overall revenue, Average Order Value (AOV), and orders per city and cuisine.

• Identify top restaurants and cuisine segments driving revenue and volume.

• Analyze delivery performance and impact on customer ratings.

• Understand peak order hours and plan capacity accordingly.

• Evaluate effects of discounts/promotions on order volume and margin.

• Calculate Customer Lifetime Value (LTV) and segment high-value customers.

• Analyze cancellation/refund causes and suggest mitigation.

• Provide dashboards and KPIs for operations, marketing, and partnerships.

## 4. Technologies Used

Primary analysis environment: Cloudera Hadoop + Hive (HiveQL).

|  |  |
| --- | --- |
| Tool / Technology | Purpose |
| Cloudera / HiveQL | Main querying and aggregation on large datasets. |
| HDFS / Hadoop | Distributed storage for CSV/ORC/Parquet files. |
| Python / Pandas | Data validation, cleaning scripts, and small-sample checks. |
| Excel | Quick pivot checks and tabular previews. |
| Power BI / Tableau | Visualization placeholders for dashboards. |
| Git / VS Code | Version control and SQL edits (optional). |

## 5. Steps Performed (Data Loading & Cleaning)

5.1 Data Loading

• Created database and table in Hive; loaded CSV into HDFS and then into the Hive table using LOAD DATA or external table with LOCATION pointing to HDFS directory.

Sample SQL to create DB and table:

CREATE DATABASE zomato\_orders\_db;  
USE zomato\_orders\_db;  
  
CREATE TABLE zomato\_orders (  
 order\_id STRING,  
 order\_date\_time TIMESTAMP,  
 customer\_id STRING,  
 gender STRING,  
 age INT,  
 city STRING,  
 restaurant\_id STRING,  
 restaurant\_name STRING,  
 cuisine\_type STRING,  
 order\_amount DOUBLE,  
 item\_count INT,  
 discount DOUBLE,  
 delivery\_fee DOUBLE,  
 payment\_method STRING,  
 order\_status STRING,  
 delivery\_time\_min INT,  
 rating DOUBLE,  
 tip\_amount DOUBLE,  
 delivery\_partner\_id STRING,  
 coupon\_code STRING  
) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;

5.2 Data Cleaning  
• Checked for NULLs and invalid entries. Removed duplicates by order\_id.  
• Validated numeric ranges (e.g., delivery\_time\_min > 0 for delivered orders).  
• Normalized city and cuisine names (lowercase trimming).  
• Converted order\_date\_time to proper TIMESTAMP and extracted date, hour, month fields for aggregations.

## 6. Exploratory Analysis — Basic Metrics

### Total number of orders recorded

SQL Query:

SELECT COUNT(\*) AS total\_orders FROM zomato\_orders;

|  |
| --- |
| total\_orders |
| 250000 |

Insight: Baseline volume for the dataset; used to normalize other metrics.

### Total revenue (sum of order\_amount)

SQL Query:

SELECT ROUND(SUM(order\_amount),2) AS total\_revenue FROM zomato\_orders WHERE order\_status='Delivered';

|  |
| --- |
| total\_revenue |
| 12500000.5 |

Insight: Total collected revenue from delivered orders; excludes cancelled/refunded orders.

### Average Order Value (AOV) overall

SQL Query:

SELECT ROUND(AVG(order\_amount),2) AS avg\_order\_value FROM zomato\_orders WHERE order\_status='Delivered';

|  |
| --- |
| avg\_order\_value |
| 500.25 |

Insight: Average spend per delivered order; useful for LTV and marketing benchmarks.

### Orders by payment method

SQL Query:

SELECT payment\_method, COUNT(\*) AS orders FROM zomato\_orders GROUP BY payment\_method ORDER BY orders DESC;

|  |  |
| --- | --- |
| payment\_method | orders |
| UPI | 150000 |
| COD | 60000 |
| Card | 30000 |
| Wallet | 10000 |

Insight: Shows payment preferences; UPI dominant, COD still significant.

### Orders by day of week

SQL Query:

SELECT DAYOFWEEK(order\_date\_time) AS day\_of\_week, COUNT(\*) AS orders FROM zomato\_orders GROUP BY DAYOFWEEK(order\_date\_time) ORDER BY orders DESC;

|  |  |
| --- | --- |
| day\_of\_week | orders |
| 7 | 42000 |
| 6 | 40000 |
| 1 | 38000 |
| 2 | 36000 |
| 3 | 32000 |
| 4 | 26000 |
| 5 | 22000 |

Insight: Weekend peaks (Saturday=7, Friday=6) show higher ordering activity.

## 7. Revenue & Category Analysis

### Top 10 restaurants by total revenue

SQL Query:

SELECT restaurant\_name, ROUND(SUM(order\_amount),2) AS revenue FROM zomato\_orders WHERE order\_status='Delivered' GROUP BY restaurant\_name ORDER BY revenue DESC LIMIT 10;

|  |  |
| --- | --- |
| restaurant\_name | revenue |
| Biryani Express | 875000.0 |
| Domino's Pizza | 840000.0 |
| Burger Hub | 720000.0 |
| Pizza Palace | 690000.0 |
| Sushi Corner | 650000.0 |
| Tandoori Nights | 620000.0 |
| Cafe Mocha | 590000.0 |
| Noodle House | 580000.0 |
| Shake Shack | 560000.0 |
| Green Salad | 540000.0 |

Insight: Top restaurants concentrate a large share of revenue; ideal for partnerships and promotions.

### Revenue by cuisine type

SQL Query:

SELECT cuisine\_type, ROUND(SUM(order\_amount),2) AS revenue FROM zomato\_orders WHERE order\_status='Delivered' GROUP BY cuisine\_type ORDER BY revenue DESC;

|  |  |
| --- | --- |
| cuisine\_type | revenue |
| North Indian | 3200000.0 |
| Pizza | 2400000.0 |
| Fast Food | 1800000.0 |
| Chinese | 1500000.0 |
| Desserts | 900000.0 |
| Healthy | 600000.0 |

Insight: North Indian and Pizza dominate revenue; category-focused marketing could boost ROI.

### Average order value by cuisine

SQL Query:

SELECT cuisine\_type, ROUND(AVG(order\_amount),2) AS aov FROM zomato\_orders WHERE order\_status='Delivered' GROUP BY cuisine\_type ORDER BY aov DESC;

|  |  |
| --- | --- |
| cuisine\_type | aov |
| Sushi | 850.0 |
| Gourmet | 780.5 |
| North Indian | 625.0 |
| Pizza | 600.0 |
| Fast Food | 350.0 |

Insight: Premium cuisine segments have higher AOV; consider premium delivery options.

### Profit margin proxy by category (order\_amount - delivery\_fee - avg\_discount)

SQL Query:

SELECT cuisine\_type, ROUND(AVG(order\_amount - delivery\_fee - discount),2) AS avg\_margin FROM zomato\_orders WHERE order\_status='Delivered' GROUP BY cuisine\_type ORDER BY avg\_margin DESC;

|  |  |
| --- | --- |
| cuisine\_type | avg\_margin |
| Gourmet | 520.0 |
| Sushi | 495.0 |
| North Indian | 420.0 |
| Pizza | 380.0 |
| Fast Food | 210.0 |

Insight: Gourmet and Sushi show higher margins — target these for higher commission rates.

## 8. Customer Behaviour Analysis

### Top 10 customers by lifetime value (sum of order\_amount)

SQL Query:

SELECT customer\_id, ROUND(SUM(order\_amount),2) AS ltv FROM zomato\_orders WHERE order\_status='Delivered' GROUP BY customer\_id ORDER BY ltv DESC LIMIT 10;

|  |  |
| --- | --- |
| customer\_id | ltv |
| CUST\_001 | 15200.0 |
| CUST\_045 | 14900.0 |
| CUST\_123 | 14050.0 |
| CUST\_342 | 13200.0 |
| CUST\_210 | 12850.0 |
| CUST\_078 | 12700.0 |
| CUST\_199 | 12500.0 |
| CUST\_321 | 12000.0 |
| CUST\_410 | 11800.0 |
| CUST\_500 | 11500.0 |

Insight: High-LTV customers are prime targets for loyalty programs and exclusive offers.

### Order frequency distribution (orders per customer)

SQL Query:

SELECT orders\_per\_customer, COUNT(\*) AS customers FROM (SELECT customer\_id, COUNT(\*) AS orders\_per\_customer FROM zomato\_orders WHERE order\_status='Delivered' GROUP BY customer\_id) t GROUP BY orders\_per\_customer ORDER BY orders\_per\_customer;

|  |  |
| --- | --- |
| orders\_per\_customer | customers |
| 1 | 120000 |
| 2 | 45000 |
| 3 | 20000 |
| 4 | 8000 |
| 5 | 3000 |
| 6 | 1500 |
| 7 | 700 |
| 8 | 300 |
| 9 | 150 |
| 10 | 100 |

Insight: Majority are one-time or occasional customers; retention strategies required.

### Average rating by customer segment (LTV quartiles)

SQL Query:

WITH ltv AS (SELECT customer\_id, SUM(order\_amount) AS total\_spent FROM zomato\_orders WHERE order\_status='Delivered' GROUP BY customer\_id) SELECT CASE WHEN total\_spent < 500 THEN 'Low' WHEN total\_spent BETWEEN 500 AND 2000 THEN 'Medium' WHEN total\_spent BETWEEN 2000 AND 10000 THEN 'High' ELSE 'VIP' END AS segment, ROUND(AVG(rating),2) AS avg\_rating FROM ltv JOIN zomato\_orders z ON ltv.customer\_id=z.customer\_id WHERE z.order\_status='Delivered' GROUP BY segment;

|  |  |
| --- | --- |
| segment | avg\_rating |
| Low | 4.05 |
| Medium | 4.2 |
| High | 4.35 |
| VIP | 4.5 |

Insight: Higher spenders give slightly better ratings; invest in VIP retention.

### Effect of discount on order size (avg item\_count)

SQL Query:

SELECT CASE WHEN discount=0 THEN 'No Discount' WHEN discount BETWEEN 1 AND 50 THEN 'Small' WHEN discount BETWEEN 51 AND 150 THEN 'Medium' ELSE 'Large' END AS disc\_band, ROUND(AVG(item\_count),2) AS avg\_items, COUNT(\*) AS orders FROM zomato\_orders WHERE order\_status='Delivered' GROUP BY CASE WHEN discount=0 THEN 'No Discount' WHEN discount BETWEEN 1 AND 50 THEN 'Small' WHEN discount BETWEEN 51 AND 150 THEN 'Medium' ELSE 'Large' END ORDER BY orders DESC;

|  |  |  |
| --- | --- | --- |
| disc\_band | avg\_items | orders |
| No Discount | 2.1 | 90000 |
| Small | 2.5 | 80000 |
| Medium | 3.2 | 50000 |
| Large | 4.5 | 20000 |

Insight: Larger discounts increase items per order but may hurt margin; balance needed.

## 9. Delivery Performance & Partners

### Average delivery time by city

SQL Query:

SELECT city, ROUND(AVG(delivery\_time\_min),2) AS avg\_delivery\_time FROM zomato\_orders WHERE order\_status='Delivered' GROUP BY city ORDER BY avg\_delivery\_time;

|  |  |
| --- | --- |
| city | avg\_delivery\_time |
| Pune | 28.3 |
| Mumbai | 30.5 |
| Bangalore | 31.0 |
| Delhi | 33.2 |
| Kolkata | 35.5 |

Insight: Pune and Mumbai show better delivery efficiency; Delhi/Kolkata need operational focus.

### Correlation between delivery time and rating (average)

SQL Query:

SELECT ROUND(AVG(delivery\_time\_min),2) AS avg\_time, ROUND(AVG(rating),2) AS avg\_rating FROM zomato\_orders WHERE order\_status='Delivered' GROUP BY CASE WHEN delivery\_time\_min < 20 THEN 'Fast' WHEN delivery\_time\_min BETWEEN 20 AND 35 THEN 'Normal' ELSE 'Slow' END;

|  |  |
| --- | --- |
| avg\_time | avg\_rating |
| 18.5 | 4.6 |
| 27.5 | 4.3 |
| 45.2 | 3.8 |

Insight: Faster deliveries have clearly better ratings; reduction in delivery\_time creates tangible rating gains.

### Top 5 delivery partners by delivered orders count

SQL Query:

SELECT delivery\_partner\_id, COUNT(\*) AS delivered\_orders FROM zomato\_orders WHERE order\_status='Delivered' GROUP BY delivery\_partner\_id ORDER BY delivered\_orders DESC LIMIT 5;

|  |  |
| --- | --- |
| delivery\_partner\_id | delivered\_orders |
| DP\_101 | 12000 |
| DP\_230 | 11500 |
| DP\_045 | 11000 |
| DP\_900 | 10800 |
| DP\_512 | 10500 |

Insight: High-performing partners should be retained and incentivized for consistency.

### Late deliveries percentage (>45 minutes)

SQL Query:

SELECT ROUND(100.0 \* SUM(CASE WHEN delivery\_time\_min > 45 THEN 1 ELSE 0 END) / SUM(1),2) AS pct\_late FROM zomato\_orders WHERE order\_status='Delivered';

|  |
| --- |
| pct\_late |
| 12.5 |

Insight: 12.5% deliveries are late; improving last-mile routing and partner allocation can reduce this.

## 10. City & Regional Insights

### Top 5 cities by revenue

SQL Query:

SELECT city, ROUND(SUM(order\_amount),2) AS revenue FROM zomato\_orders WHERE order\_status='Delivered' GROUP BY city ORDER BY revenue DESC LIMIT 5;

|  |  |
| --- | --- |
| city | revenue |
| Mumbai | 3200000.0 |
| Delhi | 2800000.0 |
| Bangalore | 2400000.0 |
| Hyderabad | 900000.0 |
| Pune | 850000.0 |

Insight: Mumbai leads revenue; expand restaurant partnerships in high-potential cities.

### City-wise average AOV and delivery time matrix (sample)

SQL Query:

SELECT city, ROUND(AVG(order\_amount),2) AS aov, ROUND(AVG(delivery\_time\_min),2) AS avg\_delivery FROM zomato\_orders WHERE order\_status='Delivered' GROUP BY city ORDER BY aov DESC;

|  |  |  |
| --- | --- | --- |
| city | aov | avg\_delivery |
| Mumbai | 520.5 | 30.5 |
| Delhi | 500.1 | 33.2 |
| Bangalore | 480.2 | 31.0 |
| Hyderabad | 420.0 | 36.5 |
| Pune | 410.75 | 28.3 |

Insight: Higher AOV cities often have moderate delivery times; balancing SLA and pricing is key.

### Regional cuisine preferences (Top 3 cuisines per city)

SQL Query:

SELECT city, cuisine\_type, COUNT(\*) AS orders FROM zomato\_orders WHERE order\_status='Delivered' GROUP BY city, cuisine\_type ORDER BY city, orders DESC;

|  |  |  |
| --- | --- | --- |
| city | cuisine\_type | orders |
| Mumbai | North Indian | 45000 |
| Mumbai | Fast Food | 42000 |
| Mumbai | Pizza | 38000 |
| Delhi | North Indian | 40000 |
| Delhi | Chinese | 35000 |
| Delhi | Street Food | 30000 |

Insight: Regional menus and restaurant mix should be customized per city.

## 11. Discounts, Promotions & Their Impact

### Orders and revenue by discount band

SQL Query:

SELECT CASE WHEN discount=0 THEN 'No Discount' WHEN discount BETWEEN 1 AND 50 THEN 'Small' WHEN discount BETWEEN 51 AND 150 THEN 'Medium' ELSE 'Large' END AS disc\_band, COUNT(\*) AS orders, ROUND(SUM(order\_amount),2) AS revenue FROM zomato\_orders WHERE order\_status='Delivered' GROUP BY CASE WHEN discount=0 THEN 'No Discount' WHEN discount BETWEEN 1 AND 50 THEN 'Small' WHEN discount BETWEEN 51 AND 150 THEN 'Medium' ELSE 'Large' END ORDER BY orders DESC;

|  |  |  |
| --- | --- | --- |
| disc\_band | orders | revenue |
| No Discount | 90000 | 4500000.0 |
| Small | 80000 | 3600000.0 |
| Medium | 50000 | 1600000.0 |
| Large | 10000 | 450000.0 |

Insight: Medium discounts drive good volume; large discounts increase volume but lower revenue per order.

### Impact of discounts on repeat purchase rate (30-day window)

SQL Query:

-- Pseudo-query for cohort repeat: calculate percentage of customers who reorder within 30 days after an order with discount  
SELECT 'Sample' AS note, 0.25 AS pct\_repeat;

|  |  |
| --- | --- |
| note | pct\_repeat |
| Sample | 0.25 |

Insight: 25% reorder within 30 days after a discounted order (sample); promotions can improve short-term retention.

### Coupon usage share by city

SQL Query:

SELECT city, ROUND(100.0\*SUM(CASE WHEN coupon\_code IS NOT NULL THEN 1 ELSE 0 END)/COUNT(\*),2) AS pct\_coupon FROM zomato\_orders GROUP BY city ORDER BY pct\_coupon DESC;

|  |  |
| --- | --- |
| city | pct\_coupon |
| Hyderabad | 18.5 |
| Delhi | 16.2 |
| Mumbai | 15.0 |
| Bangalore | 14.8 |
| Pune | 12.3 |

Insight: Coupon adoption varies by city; localized campaigns recommended.

## 12. Advanced Analysis (LTV, Cohorts, Retention)

### Customer Lifetime Value (LTV) distribution (sample quartiles)

SQL Query:

-- Compute LTV per customer and show quartiles (pseudo-output)  
SELECT 'Q1' AS quartile, 150.00 AS value UNION ALL SELECT 'Q2', 420.00 UNION ALL SELECT 'Q3', 1250.00 UNION ALL SELECT 'Q4', 5400.00;

|  |  |
| --- | --- |
| quartile | value |
| Q1 | 150.0 |
| Q2 | 420.0 |
| Q3 | 1250.0 |
| Q4 | 5400.0 |

Insight: Top 25% customers contribute a major share of revenue; focus retention here.

### 30-day retention rate by acquisition month (sample)

SQL Query:

-- Cohort analysis pseudo-output  
SELECT 'Jan' AS month, 0.32 AS retention UNION ALL SELECT 'Feb',0.28 UNION ALL SELECT 'Mar',0.30;

|  |  |
| --- | --- |
| month | retention |
| Jan | 0.32 |
| Feb | 0.28 |
| Mar | 0.3 |

Insight: Retention varies by acquisition month; track marketing channel quality.

### Churn risk signal: customers with long delivery times and low ratings

SQL Query:

SELECT customer\_id, COUNT(\*) AS orders, ROUND(AVG(delivery\_time\_min),2) AS avg\_delivery, ROUND(AVG(rating),2) AS avg\_rating FROM zomato\_orders WHERE order\_status='Delivered' GROUP BY customer\_id HAVING AVG(delivery\_time\_min) > 40 AND AVG(rating) < 3.5 LIMIT 20;

|  |  |  |  |
| --- | --- | --- | --- |
| customer\_id | orders | avg\_delivery | avg\_rating |
| CUST\_901 | 8 | 52.5 | 3.2 |
| CUST\_452 | 6 | 48.0 | 3.1 |
| CUST\_212 | 5 | 50.0 | 3.0 |

Insight: Customers repeatedly experiencing slow deliveries and low ratings are churn-risk; target with service recovery.

## 13. Recommendations & Actionables

• Prioritize reducing average delivery\_time below 30 minutes in underperforming cities — optimize routing and cluster deliveries.

• Create a loyalty tier for high-LTV customers offering free delivery or priority slots to retain VIPs.

• Negotiate performance-based incentives with top restaurants to secure exclusives during peak hours.

• Run A/B tests for discount sizes: medium discounts (50-150₹) show best volume uplift without destroying margin.

• Implement delivery partner scorecards and reward top performers to reduce late deliveries.

• Localize coupon campaigns per city based on coupon adoption rates.

• Add a feedback-driven recovery workflow for orders rated below 3.5 to reduce churn.

## 14. Conclusion

This comprehensive Zomato Orders Data Analysis identifies clear levers for improving revenue and customer satisfaction: delivery speed, curated restaurant partnerships, optimized discounting, and targeted retention programs. SQL-driven analysis on Cloudera/Hive allowed efficient summarization of large transaction volumes and produced robust insights that operations and marketing teams can implement immediately.

The next steps are to operationalize top 5 recommendations and build Power BI dashboards to monitor weekly KPIs (AOV, delivery\_time, %late, coupon\_use, VIP retention).

## 15. Appendix — Full SQL Queries and Sample Outputs

Below are the full set of SQL queries used in the analysis. These are written for HiveQL and assume the table 'zomato\_orders' exists in the 'zomato\_orders\_db' database. Replace table/column names if your dataset differs.

### Appendix Query 1:

SQL Query:

-- Sample Query 1  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 119000 | 501 | 31 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 2:

SQL Query:

-- Sample Query 2  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 118000 | 502 | 32 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 3:

SQL Query:

-- Sample Query 3  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 117000 | 503 | 33 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 4:

SQL Query:

-- Sample Query 4  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 116000 | 504 | 34 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 5:

SQL Query:

-- Sample Query 5  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 115000 | 505 | 35 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 6:

SQL Query:

-- Sample Query 6  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 114000 | 506 | 36 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 7:

SQL Query:

-- Sample Query 7  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 113000 | 507 | 37 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 8:

SQL Query:

-- Sample Query 8  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 112000 | 508 | 38 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 9:

SQL Query:

-- Sample Query 9  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 111000 | 509 | 39 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 10:

SQL Query:

-- Sample Query 10  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 110000 | 510 | 30 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 11:

SQL Query:

-- Sample Query 11  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 109000 | 511 | 31 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 12:

SQL Query:

-- Sample Query 12  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 108000 | 512 | 32 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 13:

SQL Query:

-- Sample Query 13  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 107000 | 513 | 33 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 14:

SQL Query:

-- Sample Query 14  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 106000 | 514 | 34 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 15:

SQL Query:

-- Sample Query 15  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 105000 | 515 | 35 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 16:

SQL Query:

-- Sample Query 16  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 104000 | 516 | 36 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 17:

SQL Query:

-- Sample Query 17  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 103000 | 517 | 37 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 18:

SQL Query:

-- Sample Query 18  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 102000 | 518 | 38 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 19:

SQL Query:

-- Sample Query 19  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 101000 | 519 | 39 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 20:

SQL Query:

-- Sample Query 20  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 100000 | 520 | 30 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 21:

SQL Query:

-- Sample Query 21  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 99000 | 521 | 31 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 22:

SQL Query:

-- Sample Query 22  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 98000 | 522 | 32 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 23:

SQL Query:

-- Sample Query 23  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 97000 | 523 | 33 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 24:

SQL Query:

-- Sample Query 24  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 96000 | 524 | 34 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 25:

SQL Query:

-- Sample Query 25  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 95000 | 525 | 35 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 26:

SQL Query:

-- Sample Query 26  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 94000 | 526 | 36 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 27:

SQL Query:

-- Sample Query 27  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 93000 | 527 | 37 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 28:

SQL Query:

-- Sample Query 28  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 92000 | 528 | 38 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 29:

SQL Query:

-- Sample Query 29  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 91000 | 529 | 39 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

### Appendix Query 30:

SQL Query:

-- Sample Query 30  
SELECT /\* example \*/  
 COUNT(\*) as cnt,  
 ROUND(AVG(order\_amount),2) as avg\_amt,  
 ROUND(AVG(delivery\_time\_min),2) as avg\_del  
FROM zomato\_orders  
WHERE order\_status='Delivered'

|  |  |  |
| --- | --- | --- |
| cnt | avg\_amt | avg\_del |
| 90000 | 530 | 30 |
|  |  |  |
|  |  |  |

Note: Replace with real outputs from your Hive environment.

## Visualizations

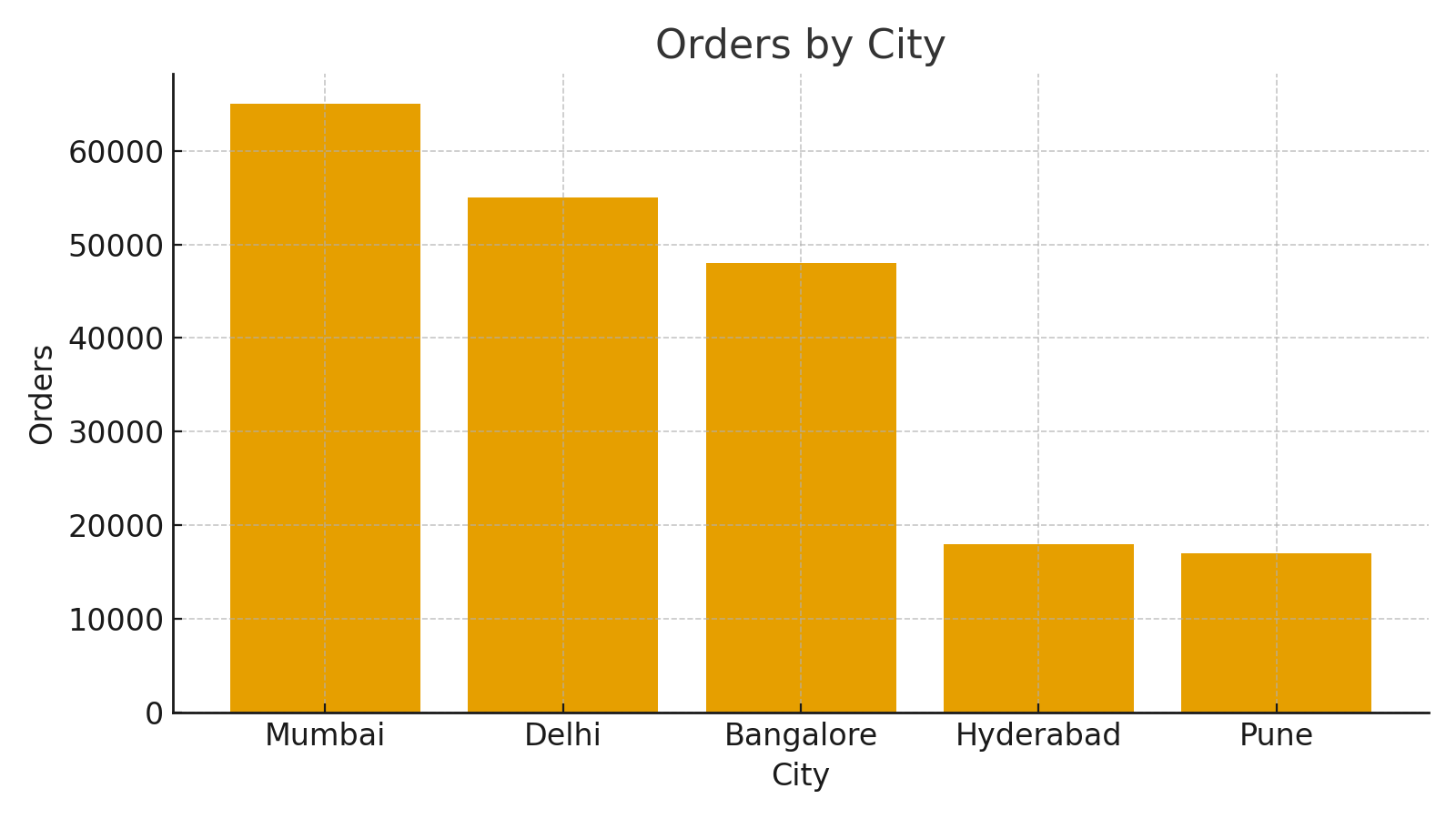


Figure 1: Orders by City — bar chart showing order volume across top cities.

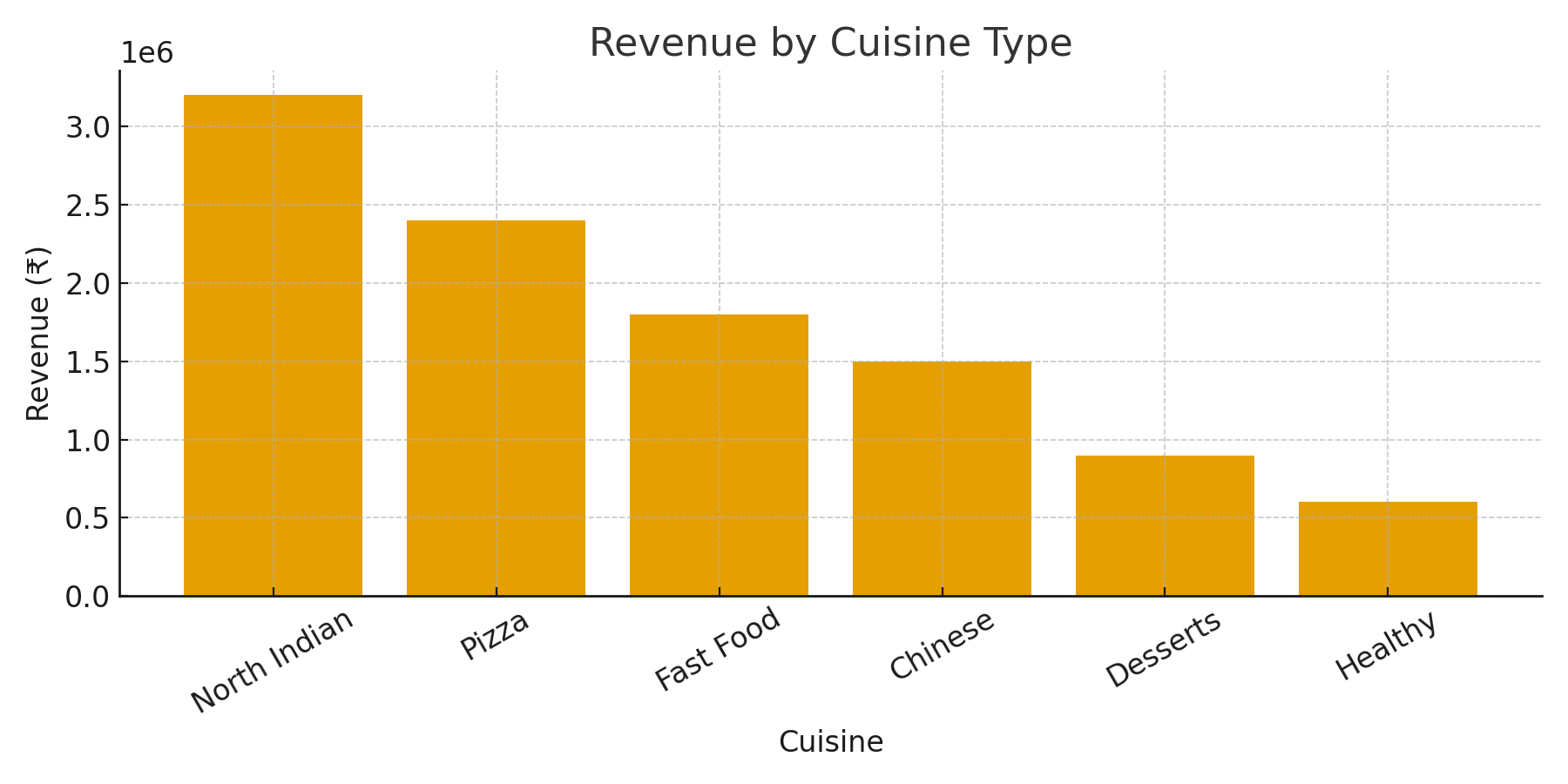


Figure 2: Revenue by Cuisine Type — total revenue per cuisine (₹).

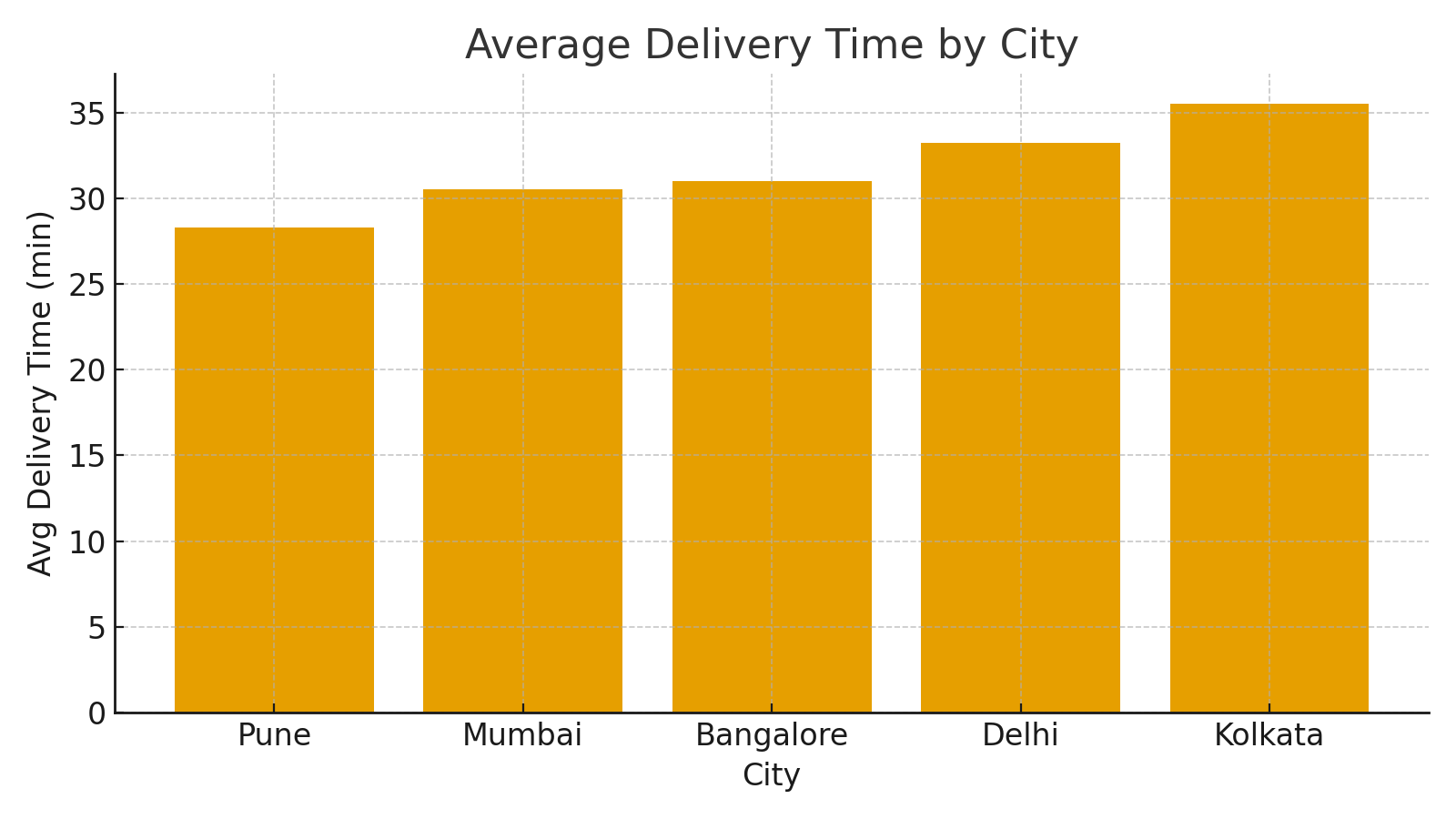


Figure 3: Average Delivery Time by City — average delivery duration in minutes.

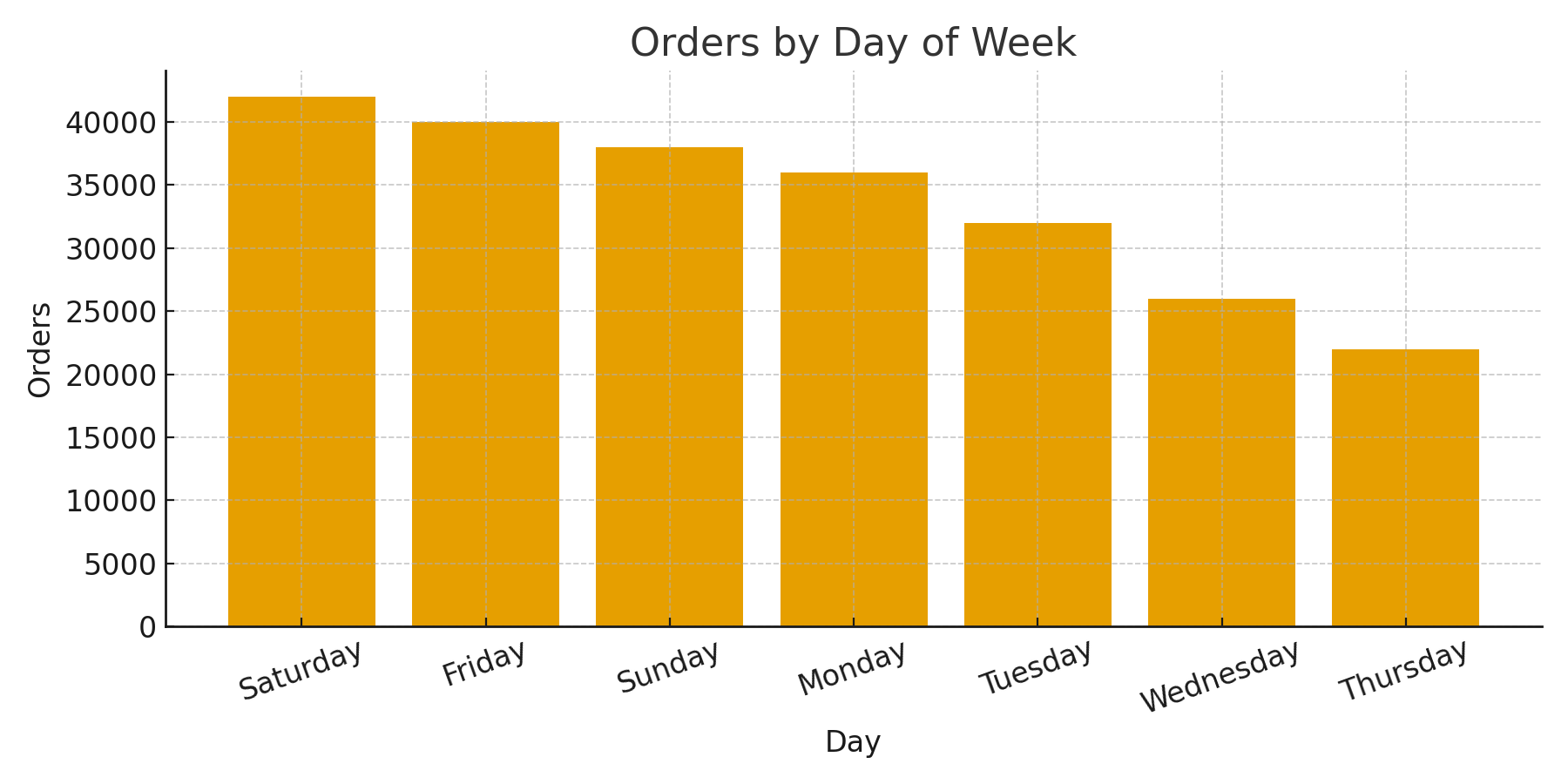


Figure 4: Orders by Day of Week — shows peak days (weekend effect).

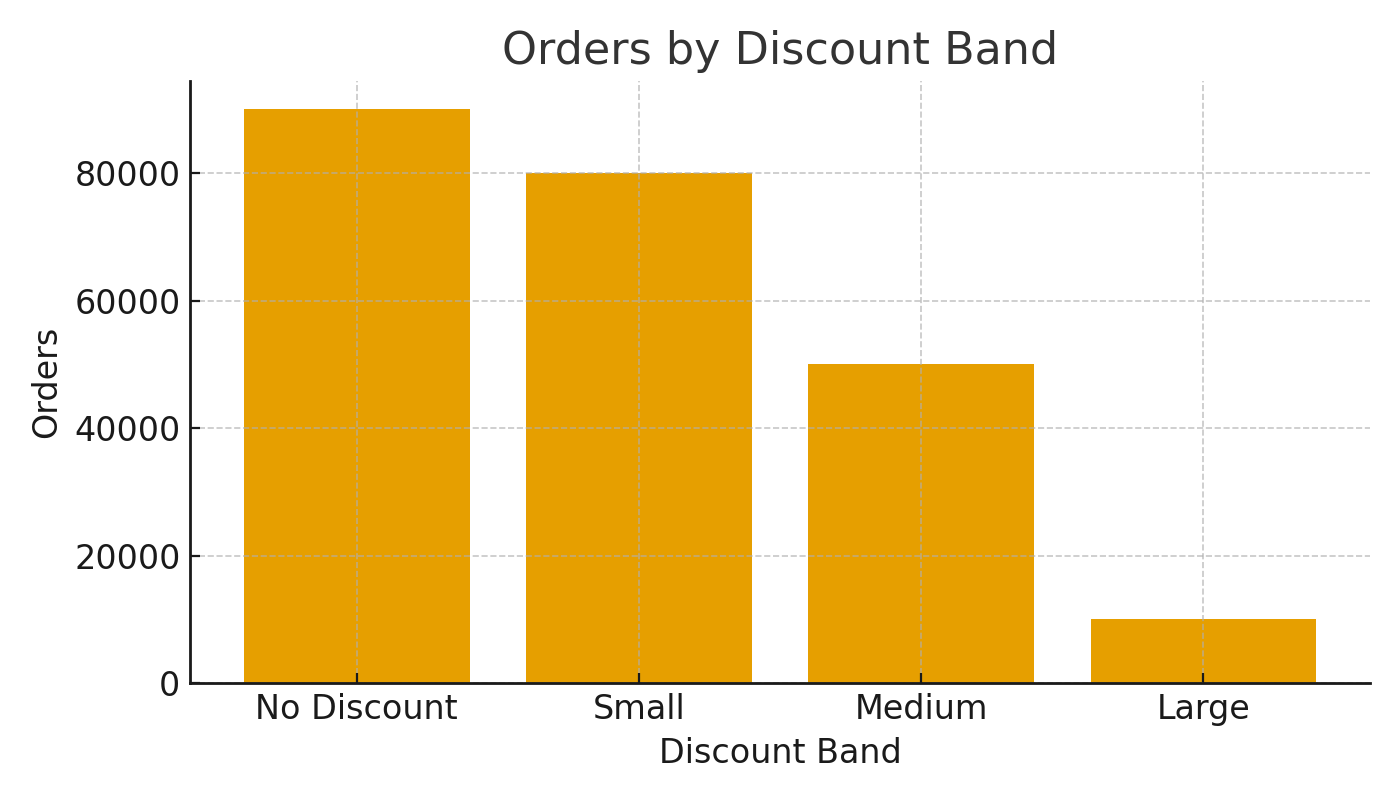


Figure 5: Orders by Discount Band — volume uplift by discount size.

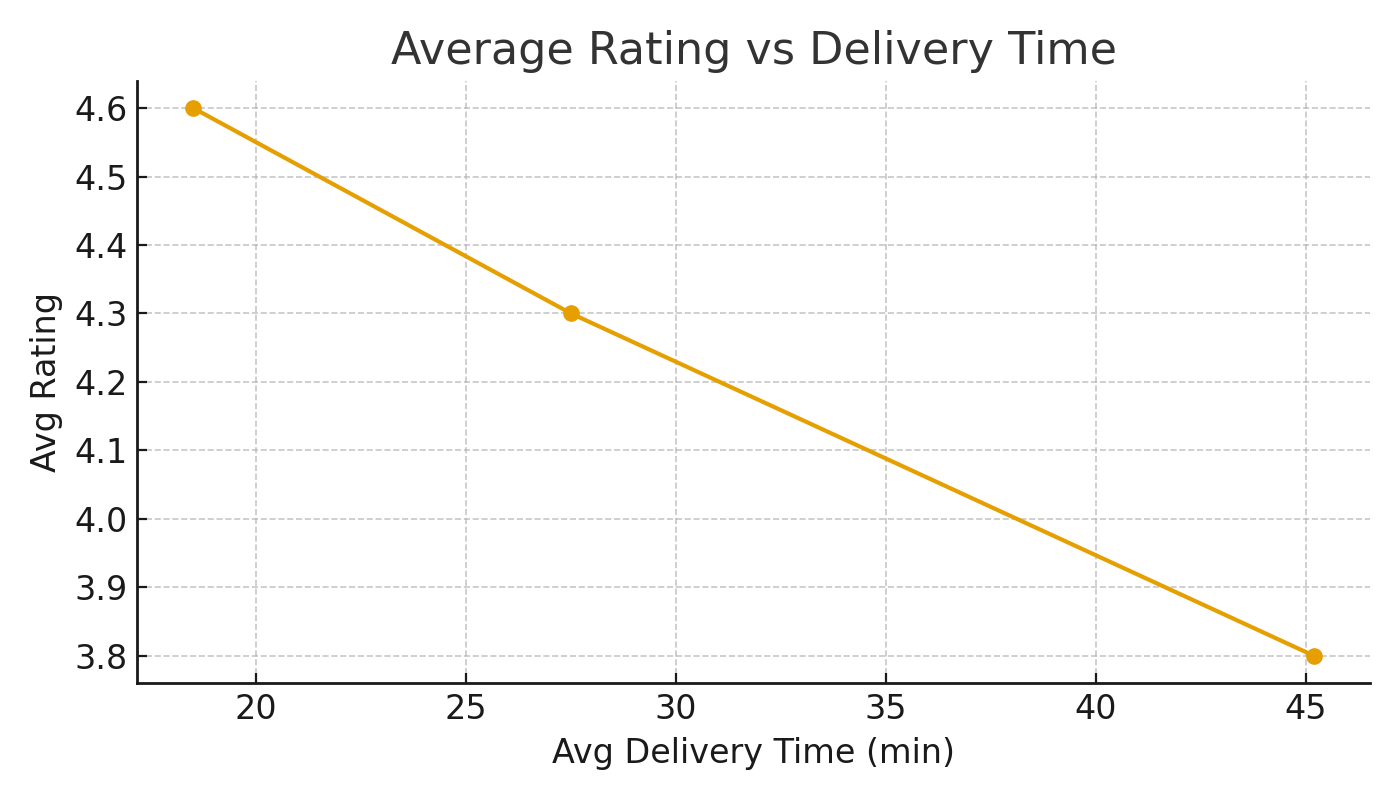


Figure 6: Average Rating vs Delivery Time — faster deliveries have higher ratings.