E-Commerce Sales and Customer Behavior Analysis

Code

Advanced R Programming

AUTHOR
Kundan Jaiswal

PUBLISHED
March 1, 2026

1 Introduction

The rapid growth of online commerce has generated massive volumes of transactional data. Analyzing this information will allows businesses to understand customer behaviour, evaluate product performance and optimize revenue strategies to help the system.

This project uses the Pakistan Largest Ecommerce Dataset available on Kaggle. The dataset contains several hundred thousand transactions recorded between 2016 and 2018. It includes information about orders, products, categories, prices, quantities, payment methods and discounts.

Because the dataset captures real commercial activity across multiple years, it provides an excellent opportunity to explore temporal trends, purchasing patterns and revenue distribution across categories.

1.1 Research Questions

The analysis focuses on the following questions:

1. How do sales and order volumes evolve over time?
2. Which product categories generate the highest revenue?
3. What payment methods are most frequently used by customers?
4. Do discounts appear to influence purchasing activity?

1.2 Hypotheses

To guide the exploration, the following hypotheses are considered:

- Sales volume increases over time as e-commerce adoption grows.
- A small number of categories contribute to a large share of total revenue.
- Cash on delivery is expected to be the dominant payment method.
- Higher discounts may be associated with higher order counts.

1.3 Analytical Roadmap

The project will begin with
- Data import and exploration,
- Cleaning and preparation of variables.
- Transformations and aggregations will be performed using modern dplyr workflows.
- The analysis will then present static.

- Interactive visualizations, summary tables and a dashboard of key performance indicators.
- Conclusions will be drawn based on statistical evidence obtained from the data.

2 Data Import and Initial Exploration

```
# reading the dataset
ecom <- read_csv("C:/Users/Kundan Jaiswal/OneDrive/Desktop/R Final Project/F
```

```
New names:
• `` -> `...22`
• `` -> `...23`
• `` -> `...24`
• `` -> `...25`
• `` -> `...26`

Warning: One or more parsing issues, call `problems()` on your data frame
for details,
e.g.:
  dat <- vroom(...)
  problems(dat)

Rows: 1048575 Columns: 26
── Column specification
────────────────────────────────────────────────────────
Delimiter: ","
chr (12): status, created_at, sku, category_name_1, sales_commission_code,
p...
dbl  (9): item_id, price, qty_ordered, grand_total, increment_id,
discount_a...
lgl  (5): ...22, ...23, ...24, ...25, ...26

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

```
# preview first rows
head(ecom)
```

```
# A tibble: 6 × 26
  item_id status     created_at sku   price qty_ordered grand_total
increment_id
    <dbl> <chr>      <chr>      <chr> <dbl>       <dbl>       <dbl>
<dbl>
1  211131 complete   7/1/2016   krea…  1950           1        1950
100147443
2  211133 canceled   7/1/2016   kcc_…   240           1         240
100147444
3  211134 canceled   7/1/2016   Ego_…  2450           1        2450
100147445
4  211135 complete   7/1/2016   kcc_…   360           1          60
100147446
5  211136 order_ref… 7/1/2016   BK70…   555           2        1110
100147447
```

```
6  211137 canceled   7/1/2016   UK_N…    80           1          80
100147448
# ℹ 18 more variables: category_name_1 <chr>, sales_commission_code <chr>,
#   discount_amount <dbl>, payment_method <chr>, `Working Date` <chr>,
#   `BI Status` <chr>, MV <chr>, Year <dbl>, Month <dbl>,
#   `Customer Since` <chr>, `M-Y` <chr>, FY <chr>, `Customer ID` <dbl>,
#   ...22 <lgl>, ...23 <lgl>, ...24 <lgl>, ...25 <lgl>, ...26 <lgl>
```

2.1 Understand the structure of dataset

```
# dimensions
dim(ecom)
```

```
[1] 1048575      26
```

```
# column names
names(ecom)
```

```
 [1] "item_id"              "status"               "created_at"
 [4] "sku"                  "price"                "qty_ordered"
 [7] "grand_total"          "increment_id"         "category_name_1"
[10] "sales_commission_code" "discount_amount"      "payment_method"
[13] "Working Date"         "BI Status"            "MV"
[16] "Year"                 "Month"                "Customer Since"
[19] "M-Y"                  "FY"                   "Customer ID"
[22] "...22"                "...23"                "...24"
[25] "...25"                "...26"
```

```
# structure
glimpse(ecom)
```

```
Rows: 1,048,575
Columns: 26
$ item_id              <dbl> 211131, 211133, 211134, 211135, 211136,
211137, …
$ status               <chr> "complete", "canceled", "canceled",
"complete", …
$ created_at           <chr> "7/1/2016", "7/1/2016", "7/1/2016",
"7/1/2016", …
$ sku                  <chr> "kreations_YI 06-L", "kcc_Buy 2 Frey Air
Freshen…
$ price                <dbl> 1950, 240, 2450, 360, 555, 80, 360, 170,
96499, …
$ qty_ordered          <dbl> 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, …
$ grand_total          <dbl> 1950, 240, 2450, 60, 1110, 80, 60, 170,
96499, 9…
$ increment_id         <dbl> 100147443, 100147444, 100147445, 100147446,
1001…
$ category_name_1      <chr> "Women's Fashion", "Beauty & Grooming",
"Women's…
$ sales_commission_code <chr> "\\N", "\\N", "\\N", "R-FSD-52352", "\\N",
"\\N"…
$ discount_amount      <dbl> 0, 0, 0, 300, 0, 0, 300, 0, 0, 0, 0, 0, 0, 0,
0,…
```

```
$ payment_method        <chr> "cod", "cod", "cod", "cod", "cod", "cod",
"cod",…
$ `Working Date`        <chr> "7/1/2016", "7/1/2016", "7/1/2016",
"7/1/2016", …
$ `BI Status`           <chr> "#REF!", "Gross", "Gross", "Net", "Valid",
"Gros…
$ MV                    <chr> "1,950", "240", "2,450", "360", "1,110",
"80", "…
$ Year                  <dbl> 2016, 2016, 2016, 2016, 2016, 2016, 2016,
2016, …
$ Month                 <dbl> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
7, …
$ `Customer Since`      <chr> "2016-7", "2016-7", "2016-7", "2016-7",
"2016-7"…
$ `M-Y`                 <chr> "7-2016", "7-2016", "7-2016", "7-2016", "7-
2016"…
$ FY                    <chr> "FY17", "FY17", "FY17", "FY17", "FY17",
"FY17", …
$ `Customer ID`         <dbl> 1, 2, 3, 4, 5, 6, 7, 6, 8, 8, 9, 10, 10, 11,
12,…
$ ...22                 <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, …
$ ...23                 <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, …
$ ...24                 <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, …
$ ...25                 <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, …
$ ...26                 <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, …
```

2.2 Summary Statistics

```
summary(ecom)
```

```
   item_id          status            created_at            sku
 Min.   :211131   Length:1048575     Length:1048575     Length:1048575
 1st Qu.:395001   Class :character   Class :character   Class :character
 Median :568424   Mode  :character   Mode  :character   Mode  :character
 Mean   :565667
 3rd Qu.:739106
 Max.   :905208
 NA's   :464051
     price          qty_ordered      grand_total        increment_id
 Min.   :      0   Min.   :   1     Min.   :   -1594   Min.   :100001290
 1st Qu.:    360   1st Qu.:   1     1st Qu.:     945   1st Qu.:100264567
 Median :    899   Median :   1     Median :    1960   Median :100351162
 Mean   :   6349   Mean   :   1     Mean   :    8531   Mean   :100356974
 3rd Qu.:   4070   3rd Qu.:   1     3rd Qu.:    6999   3rd Qu.:100450242
 Max.   :1012626   Max.   :1000     Max.   :17888000   Max.   :100562387
 NA's   :464051    NA's   :464051   NA's   :464051     NA's   :464060
 category_name_1    sales_commission_code discount_amount  payment_method
 Length:1048575     Length:1048575        Min.   : -600    Length:1048575
 Class :character   Class :character      1st Qu.:    0    Class :character
 Mode  :character   Mode  :character      Median :    0    Mode  :character
                                          Mean   :  499
```

```
                                    3rd Qu.:  160
                                    Max.   :90300
                                    NA's   :464051
 Working Date        BI Status             MV                Year
 Length:1048575     Length:1048575     Length:1048575     Min.   :2016
 Class :character   Class :character   Class :character   1st Qu.:2017
 Mode  :character   Mode  :character   Mode  :character   Median :2017
                                                          Mean   :2017
                                                          3rd Qu.:2018
                                                          Max.   :2018
                                                          NA's   :464051
     Month        Customer Since         M-Y                FY
 Min.   : 1       Length:1048575     Length:1048575     Length:1048575
 1st Qu.: 4       Class :character   Class :character   Class :character
 Median : 7       Mode  :character   Mode  :character   Mode  :character
 Mean   : 7
 3rd Qu.:11
 Max.   :12
 NA's   :464051
  Customer ID      ...22          ...23          ...24          ...25
 Min.   :     1   Mode:logical   Mode:logical   Mode:logical   Mode:logical
 1st Qu.: 13516   NA's:1048575   NA's:1048575   NA's:1048575   NA's:1048575
 Median : 42856
 Mean   : 45791
 3rd Qu.: 73536
 Max.   :115326
 NA's   :464062
  ...26
 Mode:logical
 NA's:1048575
```

2.3 Missing values overview

```
# count missing values per column
ecom |> summarise(across(everything(), ~ sum(is.na(.))))
```

```
# A tibble: 1 × 26
  item_id status created_at    sku  price qty_ordered grand_total
increment_id
    <int>  <int>      <int>  <int>  <int>       <int>       <int>
<int>
1  464051 464066     464051 464071 464051      464051      464051
464060
# ℹ 18 more variables: category_name_1 <int>, sales_commission_code <int>,
#   discount_amount <int>, payment_method <int>, `Working Date` <int>,
#   `BI Status` <int>, MV <int>, Year <int>, Month <int>,
#   `Customer Since` <int>, `M-Y` <int>, FY <int>, `Customer ID` <int>,
#   ...22 <int>, ...23 <int>, ...24 <int>, ...25 <int>, ...26 <int>
```

3 3 – Data Cleaning and Transformation

```r
# Count missing values per column and sort descending
missing_vals <- ecom |>
  summarise(across(everything(), ~ sum(is.na(.)))) |>
  pivot_longer(everything(), names_to = "Column", values_to = "Missing_Count
  arrange(desc(Missing_Count))

missing_vals
```

```
# A tibble: 26 × 2
   Column                Missing_Count
   <chr>                         <int>
 1 ...22                       1048575
 2 ...23                       1048575
 3 ...24                       1048575
 4 ...25                       1048575
 5 ...26                       1048575
 6 sales_commission_code        601226
 7 category_name_1              464215
 8 sku                          464071
 9 status                       464066
10 Customer ID                  464062
# ℹ 16 more rows
```

```r
# Step 3.2 – Handle Missing Values

# Start with the raw dataset
ecom_clean <- ecom |>

  # Step 1: Replace missing numeric values with 0
  # This ensures calculations (like total revenue) work correctly
  mutate(
    price = replace_na(price, 0),
    qty_ordered = replace_na(qty_ordered, 0),
    grand_total = replace_na(grand_total, 0),
    discount_amount = replace_na(discount_amount, 0)
  ) |>

  # Step 2: Replace missing categorical values with "Unknown"
  # Ensures grouping and analysis works consistently
  mutate(
    status = replace_na(status, "Unknown"),
    category_name_1 = replace_na(category_name_1, "Unknown"),
    sales_commission_code = replace_na(sales_commission_code, "Unknown"),
    payment_method = replace_na(payment_method, "Unknown")
  ) |>

  # Step 3: Drop completely empty columns (...22 to ...26)
  select(-c("...22","...23","...24","...25","...26")) |>

  # Step 4: Filter out rows with missing essential identifiers
  # item_id, sku, and created_at are required for analysis
  filter(!is.na(item_id), !is.na(sku), !is.na(created_at))
```

```r
# See the first 10 rows of key columns to check missing values were handled
head(ecom_clean |> select(
```

```
  item_id, sku, created_at, price, qty_ordered, grand_total,
  discount_amount, status, category_name_1, payment_method, sales_commissior
), 10)
```

```
# A tibble: 10 × 11
   item_id sku   created_at price qty_ordered grand_total discount_amount
status
     <dbl> <chr> <chr>      <dbl>       <dbl>       <dbl>           <dbl>
<chr>
 1  211131 krea… 7/1/2016    1950           1        1950               0
compl…
 2  211133 kcc_… 7/1/2016     240           1         240               0
cance…
 3  211134 Ego_… 7/1/2016    2450           1        2450               0
cance…
 4  211135 kcc_… 7/1/2016     360           1          60             300
compl…
 5  211136 BK70… 7/1/2016     555           2        1110               0
order…
 6  211137 UK_N… 7/1/2016      80           1          80               0
cance…
 7  211138 kcc_… 7/1/2016     360           1          60             300
compl…
 8  211139 UK_N… 7/1/2016     170           1         170               0
compl…
 9  211140 Appl… 7/1/2016   96499           1       96499               0
cance…
10  211141 Appl… 7/1/2016   96499           1       96499               0
cance…
# ℹ 3 more variables: category_name_1 <chr>, payment_method <chr>,
#   sales_commission_code <chr>
```

```
# Step 3.3 – Parse Date and Create Derived Column

# Step 1: Parse 'created_at' column from character to Date
# Format in the dataset is month/day/year, so we use lubridate::mdy()
ecom_clean <- ecom_clean |>
  mutate(
    created_at = mdy(created_at)   # convert to Date
  )

# Step 2: Create a derived column 'total_revenue'
# Formula: total_revenue = qty_ordered * price - discount_amount
ecom_clean <- ecom_clean |>
  mutate(
    total_revenue = qty_ordered * price - discount_amount
  )

# Step 3: Verify first 10 rows to check changes
head(ecom_clean |> select(created_at, price, qty_ordered, discount_amount, t
```

```
# A tibble: 10 × 5
   created_at price qty_ordered discount_amount total_revenue
   <date>     <dbl>       <dbl>           <dbl>         <dbl>
 1 2016-07-01  1950           1               0          1950
```

```
 2 2016-07-01   240          1              0          240
 3 2016-07-01  2450          1              0         2450
 4 2016-07-01   360          1            300           60
 5 2016-07-01   555          2              0         1110
 6 2016-07-01    80          1              0           80
 7 2016-07-01   360          1            300           60
 8 2016-07-01   170          1              0          170
 9 2016-07-01 96499          1              0        96499
10 2016-07-01 96499          1              0        96499
```

```r
library(gt)

# Step 3.4 – Create Data Dictionary

# Create a tibble with all relevant information
data_dict <- tibble(
  Column = c(
    "item_id", "status", "created_at", "sku", "category_name_1",
    "qty_ordered", "price", "grand_total", "sales_commission_code",
    "payment_method", "discount_amount", "total_revenue"
  ),
  Description = c(
    "Unique identifier for each order",
    "Order status",
    "Date when order was placed",
    "Unique product identifier",
    "Product category name",
    "Quantity of product ordered",
    "Unit price of product",
    "Total amount of order",
    "Sales commission code",
    "Payment method used by customer",
    "Discount applied on order",
    "Total revenue = qty_ordered * price - discount_amount"
  ),
  Type = c(
    "Integer", "Character", "Date", "Character", "Character",
    "Integer", "Numeric", "Numeric", "Character",
    "Character", "Numeric", "Numeric"
  ),
  Missing_Values = c(
    sum(is.na(ecom_clean$item_id)),
    sum(is.na(ecom_clean$status)),
    sum(is.na(ecom_clean$created_at)),
    sum(is.na(ecom_clean$sku)),
    sum(is.na(ecom_clean$category_name_1)),
    sum(is.na(ecom_clean$qty_ordered)),
    sum(is.na(ecom_clean$price)),
    sum(is.na(ecom_clean$grand_total)),
    sum(is.na(ecom_clean$sales_commission_code)),
    sum(is.na(ecom_clean$payment_method)),
    sum(is.na(ecom_clean$discount_amount)),
    sum(is.na(ecom_clean$total_revenue))
  ),
  Cleaning_Applied = c(
    "Filtered out rows with missing item_id",
    "Replaced NA with 'Unknown'",
    "Trimmed spaces + parsed with lubridate::mdy()",
```

```
    "Filtered out rows with missing sku",
    "Replaced NA with 'Unknown'",
    "Replaced NA with 0",
    "Replaced NA with 0",
    "Replaced NA with 0",
    "Replaced NA with 'Unknown'",
    "Replaced NA with 'Unknown'",
    "Replaced NA with 0",
    "Calculated from qty_ordered * price - discount_amount"
  )
)

# Render the Data Dictionary nicely using gt
data_dict |> gt()
```

Column	Description	Type	Missing_Values	Clean
item_id	Unique identifier for each order	Integer	0	Filter with item_
status	Order status	Character	0	Repla with '
created_at	Date when order was placed	Date	0	Trimn + par lubric
sku	Unique product identifier	Character	0	Filter with
category_name_1	Product category name	Character	0	Repla with '
qty_ordered	Quantity of product ordered	Integer	0	Repla with (
price	Unit price of product	Numeric	0	Repla with (
grand_total	Total amount of order	Numeric	0	Repla with (
sales_commission_code	Sales commission code	Character	0	Repla with '
payment_method	Payment method used by customer	Character	0	Repla with '
discount_amount	Discount applied on order	Numeric	0	Repla with (

Column	Description	Type	Missing_Values	Clean
total_revenue	Total revenue = qty_ordered * price - discount_amount	Numeric	0	Calcu qty_o price disco

> **Data Quality Issues**
>
> The original dataset contained **464,066 rows with missing critical fields** (44% of total data). These were removed to ensure analysis accuracy. Always verify data completeness before drawing conclusions.

3.1 Section 4 – Data Analysis

4 Step 4.1 – Filter Operations

```
library(dplyr)
library(stringr)

# 1. Numeric comparison: orders with total_revenue greater than 1000
high_revenue_orders <- ecom_clean |>
  filter(total_revenue > 1000)

# 2. %in% multiple values: select orders with specific statuses
selected_status_orders <- ecom_clean |>
  filter(status %in% c("complete", "processing", "pending"))

# 3. String matching: orders where category name starts with "Electronics"
electronics_orders <- ecom_clean |>
  filter(str_starts(category_name_1, "Electronics"))

# verifing results by seeing first 5 rows
head(high_revenue_orders, 5)
```

```
# A tibble: 5 × 22
  item_id status     created_at sku   price qty_ordered grand_total
increment_id
    <dbl> <chr>      <date>     <chr> <dbl>       <dbl>       <dbl>
<dbl>
1  211131 complete   2016-07-01 krea…  1950           1        1950
100147443
2  211134 canceled   2016-07-01 Ego_…  2450           1        2450
100147445
3  211136 order_ref… 2016-07-01 BK70…   555           2        1110
100147447
4  211140 canceled   2016-07-01 Appl… 96499           1       96499
100147451
5  211141 canceled   2016-07-01 Appl… 96499           1       96499
100147452
# ℹ 14 more variables: category_name_1 <chr>, sales_commission_code <chr>,
#   discount_amount <dbl>, payment_method <chr>, `Working Date` <chr>,
```

```
#   `BI Status` <chr>, MV <chr>, Year <dbl>, Month <dbl>,
#   `Customer Since` <chr>, `M-Y` <chr>, FY <chr>, `Customer ID` <dbl>,
#   total_revenue <dbl>
```

```
head(selected_status_orders, 5)
```

```
# A tibble: 5 × 22
  item_id status   created_at sku     price qty_ordered grand_total
increment_id
    <dbl> <chr>    <date>     <chr>   <dbl>       <dbl>       <dbl>
<dbl>
1  211131 complete 2016-07-01 "kreat…  1950           1        1950
100147443
2  211135 complete 2016-07-01 "kcc_k…   360           1          60
100147446
3  211138 complete 2016-07-01 "kcc_k…   360           1          60
100147449
4  211139 complete 2016-07-01 "UK_Na…   170           1         170
100147450
5  211142 complete 2016-07-01 "GFC_P…  5500           1        5500
100147453
# ℹ 14 more variables: category_name_1 <chr>, sales_commission_code <chr>,
#   discount_amount <dbl>, payment_method <chr>, `Working Date` <chr>,
#   `BI Status` <chr>, MV <chr>, Year <dbl>, Month <dbl>,
#   `Customer Since` <chr>, `M-Y` <chr>, FY <chr>, `Customer ID` <dbl>,
#   total_revenue <dbl>
```

```
head(electronics_orders, 5)
```

```
# A tibble: 0 × 22
# ℹ 22 variables: item_id <dbl>, status <chr>, created_at <date>, sku <chr>,
#   price <dbl>, qty_ordered <dbl>, grand_total <dbl>, increment_id <dbl>,
#   category_name_1 <chr>, sales_commission_code <chr>, discount_amount
<dbl>,
#   payment_method <chr>, Working Date <chr>, BI Status <chr>, MV <chr>,
#   Year <dbl>, Month <dbl>, Customer Since <chr>, M-Y <chr>, FY <chr>,
#   Customer ID <dbl>, total_revenue <dbl>
```

5 Step 4.2 – Select operations

```
# Step 4.2 – Select Operations

# 1. Using explicit column names: select key columns for revenue analysis
revenue_cols <- ecom_clean |>
  select(item_id, created_at, category_name_1, qty_ordered, price, discount_

# 2. Using helper functions: select all columns that start with "category" c
category_qty_cols <- ecom_clean |>
  select(starts_with("category"), starts_with("qty"))

# previewing first 5 rows
head(revenue_cols, 5)
```

```
# A tibble: 5 × 7
  item_id created_at category_name_1   qty_ordered price discount_amount
    <dbl> <date>     <chr>                   <dbl> <dbl>           <dbl>
1  211131 2016-07-01 Women's Fashion             1  1950               0
2  211133 2016-07-01 Beauty & Grooming           1   240               0
3  211134 2016-07-01 Women's Fashion             1  2450               0
4  211135 2016-07-01 Beauty & Grooming           1   360             300
5  211136 2016-07-01 Soghaat                     2   555               0
# ℹ 1 more variable: total_revenue <dbl>
```

```
head(category_qty_cols, 5)
```

```
# A tibble: 5 × 2
  category_name_1   qty_ordered
  <chr>                   <dbl>
1 Women's Fashion             1
2 Beauty & Grooming           1
3 Women's Fashion             1
4 Beauty & Grooming           1
5 Soghaat                     2
```

6 Step 4.3 – Mutate operations

```
# Step 4.3 – Mutate Operations

ecom_analysis <- ecom_clean |>

  # 1. Calculated column: revenue per item (price - discount_amount) * qty_c
  mutate(revenue_per_item = (price - discount_amount) * qty_ordered) |>

  # 2. Categorization with case_when(): classify orders based on total_rever
  mutate(revenue_category = case_when(
    total_revenue >= 5000 ~ "High",
    total_revenue >= 1000 & total_revenue < 5000 ~ "Medium",
    total_revenue < 1000 ~ "Low"
  )) |>

  # 3. Binary classification with if_else(): flag high quantity orders
  mutate(high_qty_flag = if_else(qty_ordered >= 5, "Yes", "No"))

# Previewing first 5 rows to verify
head(ecom_analysis |> select(item_id, total_revenue, revenue_category, qty_c
```

```
# A tibble: 5 × 6
  item_id total_revenue revenue_category qty_ordered high_qty_flag
    <dbl>         <dbl> <chr>                  <dbl> <chr>
1  211131          1950 Medium                     1 No
2  211133           240 Low                        1 No
3  211134          2450 Medium                     1 No
4  211135            60 Low                        1 No
5  211136          1110 Medium                     2 No
# ℹ 1 more variable: revenue_per_item <dbl>
```

7 Step 4.4 – Arrange operations

```
# Step 4.4 – Arrange Operations

# 1. Ascending order: orders by total_revenue (smallest first)
arrange_asc <- ecom_analysis |>
  arrange(total_revenue)

# 2. Descending order: orders by total_revenue (largest first)
arrange_desc <- ecom_analysis |>
  arrange(desc(total_revenue))

# Optional: preview first 5 rows of both
head(arrange_asc |> select(item_id, total_revenue), 5)
```

```
# A tibble: 5 × 2
  item_id total_revenue
    <dbl>         <dbl>
1  810508        -15410
2  810504        -15100
3  810506        -14890
4  810496        -14800
5  810498        -14800
```

```
head(arrange_desc |> select(item_id, total_revenue), 5)
```

```
# A tibble: 5 × 2
  item_id total_revenue
    <dbl>         <dbl>
1  507556       8944000
2  507557       8944000
3  507561       8944000
4  507562       8944000
5  508368       8944000
```

8 Step 4.5 – Summarise with .by

```
# Step 4.5 – Summarise with .by argument

# 1. Total revenue per category
revenue_per_category <- ecom_analysis |>
  summarise(total_revenue_category = sum(total_revenue), .by = category_name

# 2. Average revenue per month
avg_revenue_per_month <- ecom_analysis |>
  mutate(Month = month(created_at, label = TRUE)) |>  # extract month as fac
  summarise(avg_revenue = mean(total_revenue), .by = Month)

# 3. Count of unique customers per revenue category
unique_customers <- ecom_analysis |>
  summarise(unique_customers = n_distinct(`Customer ID`), .by = revenue_cate
```

```
# Preview results
head(revenue_per_category, 5)
```

```
# A tibble: 5 × 2
  category_name_1   total_revenue_category
  <chr>                              <dbl>
1 Women's Fashion                 99523823.
2 Beauty & Grooming               35577623.
3 Soghaat                         11919945.
4 Mobiles & Tablets             2181688581.
5 Appliances                     558240390.
```

```
head(avg_revenue_per_month, 5)
```

```
# A tibble: 5 × 2
  Month avg_revenue
  <ord>       <dbl>
1 Jul         7576.
2 Aug         5102.
3 Sep         5311.
4 Oct         5685.
5 Nov         5667.
```

```
head(unique_customers, 5)
```

```
# A tibble: 3 × 2
  revenue_category unique_customers
  <chr>                       <int>
1 Medium                      48940
2 Low                         61896
3 High                        38990
```

9 Step 4.6 – Pivot operations

```
# Step 4.6 - Pivot Operation

library(lubridate)

# First, extract Month-Year for grouping
ecom_analysis <- ecom_analysis |>
  mutate(Month_Year = format(created_at, "%Y-%m"))

# Pivot: total revenue per category per month
# Using .by argument instead of group_by()
revenue_pivot <- ecom_analysis |>
  summarise(
    total_revenue = sum(total_revenue),
    .by = c(Month_Year, category_name_1)
  ) |>
  pivot_wider(
    names_from = category_name_1,
    values_from = total_revenue,
    values_fill = 0  # fill missing combinations with 0
  )
```

```
# Preview first 5 rows of the pivot table
head(revenue_pivot, 5)
```

```
# A tibble: 5 × 18
  Month_Year `Women's Fashion` `Beauty & Grooming`  Soghaat `Mobiles &
Tablets`
  <chr>                  <dbl>               <dbl>    <dbl>
<dbl>
1 2016-07             1009501.            1174098.  548638.
19840258.
2 2016-08             1284575.             942003.  606199.
25607552.
3 2016-09             1892533.             935978.  542279
31180592.
4 2016-10             1753969.            1104945.  943304
34863989.
5 2016-11             7981767.            3390083. 1869064.
78716174.
# ℹ 13 more variables: Appliances <dbl>, `Home & Living` <dbl>,
#   `Men's Fashion` <dbl>, `Kids & Baby` <dbl>, `\\N` <dbl>, Others <dbl>,
#   Entertainment <dbl>, Computing <dbl>, Superstore <dbl>,
#   `Health & Sports` <dbl>, Books <dbl>, `School & Education` <dbl>,
#   Unknown <dbl>
```

10 Step 4.7 – Join Operation

```
# Create two summary tables and join them

# Table 1: Revenue per category
category_summary <- ecom_analysis |>
  summarise(
    total_revenue = sum(total_revenue),
    total_orders = n(),
    .by = category_name_1
  )

# Table 2: Average price per category
category_pricing <- ecom_analysis |>
  summarise(
    avg_price = mean(price),
    avg_discount = mean(discount_amount),
    .by = category_name_1
  )

# Join the two tables using left_join
category_complete <- category_summary |>
  left_join(category_pricing, by = "category_name_1") |>
  arrange(desc(total_revenue))

# Preview first 10 categories
head(category_complete, 10)
```

```
# A tibble: 10 × 5
   category_name_1     total_revenue total_orders avg_price avg_discount
   <chr>                       <dbl>        <int>     <dbl>        <dbl>
```

```
 1 "Mobiles & Tablets"   2181688581.       115710    17545.       1089.
 2 "Appliances"           558240390.        52413    11058.       1027.
 3 "Entertainment"        477991361.        26323    19256.       1888.
 4 "Computing"            172030553.        15933    10885.        856.
 5 "Others"               165861043.        29218     2225.         39.8
 6 "Women's Fashion"       99523823.        59721     1758.        192.
 7 "Men's Fashion"         81112945.        92221      908.         77.5
 8 "Beauty & Grooming"     35577623.        41496      879.        130.
 9 "\\N"                   32226233.         7833     4157.        231.
10 "Superstore"            30841291.        43613      611.        219.
```

Key Statistical Findings

- Total transactions analyzed: **584,504**
- Date range: **July 2016** to **August 2018**
- Total revenue: **PKR 3,903,291,927**

11 1. Correlation Analysis

```
# Correlation matrix for numeric variables
numeric_cols <- ecom_analysis |>
  select(price, qty_ordered, discount_amount, total_revenue)

cor_matrix <- cor(numeric_cols, use = "complete.obs")
cor_matrix
```

```
                 price qty_ordered discount_amount total_revenue
price            1.0000    -0.01763         0.45558         0.475
qty_ordered     -0.0176     1.00000        -0.00762         0.689
discount_amount  0.4556    -0.00762         1.00000         0.175
total_revenue    0.4746     0.68853         0.17490         1.000
```

```
#Check the correlation between price, qty_ordered, discount_amount, and tota
```

12 2. Top 10 Orders by Revenue (slice_max)

```
# Top 10 orders by total revenue
top_orders <- ecom_analysis |>
  slice_max(total_revenue, n = 10) |>
  select(item_id, total_revenue, category_name_1, qty_ordered, price)

top_orders
```

```
# A tibble: 10 × 5
   item_id total_revenue category_name_1   qty_ordered price
     <dbl>         <dbl> <chr>                   <dbl> <dbl>
 1  507556       8944000 Mobiles & Tablets        1000  8944
 2  507557       8944000 Mobiles & Tablets        1000  8944
 3  507561       8944000 Mobiles & Tablets        1000  8944
 4  507562       8944000 Mobiles & Tablets        1000  8944
 5  508368       8944000 Mobiles & Tablets        1000  8944
 6  508369       8944000 Mobiles & Tablets        1000  8944
 7  460979       1315875 Entertainment              25 52635
```

```
 8  217553       1279980 Mobiles & Tablets          20 63999
 9  212228       1155966 Mobiles & Tablets          34 33999
10  451416       1039479 Entertainment              21 49499
```

13 3. Cumulative Revenue by Month (window function `.by`)

```
# Cumulative revenue per month
ecom_analysis <- ecom_analysis |>
  mutate(Month = format(created_at, "%Y-%m")) |>
  arrange(created_at) |>
  mutate(cumulative_revenue = cumsum(total_revenue), .by = Month)

# Preview first 10 rows
head(ecom_analysis |> select(created_at, Month, total_revenue, cumulative_re
```

```
# A tibble: 10 × 4
   created_at Month   total_revenue cumulative_revenue
   <date>     <chr>           <dbl>              <dbl>
 1 2016-07-01 2016-07          1950               1950
 2 2016-07-01 2016-07           240               2190
 3 2016-07-01 2016-07          2450               4640
 4 2016-07-01 2016-07            60               4700
 5 2016-07-01 2016-07          1110               5810
 6 2016-07-01 2016-07            80               5890
 7 2016-07-01 2016-07            60               5950
 8 2016-07-01 2016-07           170               6120
 9 2016-07-01 2016-07         96499             102619
10 2016-07-01 2016-07         96499             199118
```

14 4. Cross-tab: Count of orders by revenue category & high quantity flag

```
# Count of orders by revenue category and high quantity
order_counts <- ecom_analysis |>
  count(revenue_category, high_qty_flag)

order_counts
```

```
# A tibble: 6 × 3
  revenue_category high_qty_flag      n
  <chr>            <chr>          <int>
1 High             No            135207
2 High             Yes             5154
3 Low              No            305948
4 Low              Yes             2323
5 Medium           No            130242
6 Medium           Yes             5630
```

15 Section 5: Data Visualization

Creating clear and informative visualizations to communicate insights from the e-commerce data.

15.1 Visualization 1: Scatter Plot with Trend Line

```
# Create scatter plot: quantity vs revenue with color by revenue category
scatter_data <- ecom_analysis |>
  filter(qty_ordered <= 20, total_revenue <= 10000) |>  # Remove extreme out
  filter(total_revenue > 0)  # Remove zero/negative values

ggplot(scatter_data, aes(x = qty_ordered, y = total_revenue, color = revenue
  geom_point(alpha = 0.5, size = 2) +
  geom_smooth(method = "lm", se = TRUE, color = "black", linewidth = 1) +
  scale_y_continuous(labels = comma, breaks = seq(0, 10000, 2000)) +
  scale_x_continuous(breaks = seq(0, 20, 2)) +
  scale_color_manual(values = c("Low" = "#3498db", "Medium" = "#f39c12", "Hi
  labs(
    title = "Order Quantity vs Total Revenue",
    subtitle = "Positive relationship between quantity ordered and revenue",
    x = "Quantity Ordered",
    y = "Total Revenue (PKR)",
    color = "Revenue Category"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 16, face = "bold"),
    plot.subtitle = element_text(size = 12, color = "gray40"),
    axis.title = element_text(size = 12),
    legend.position = "top",
    legend.title = element_text(face = "bold")
  )
```

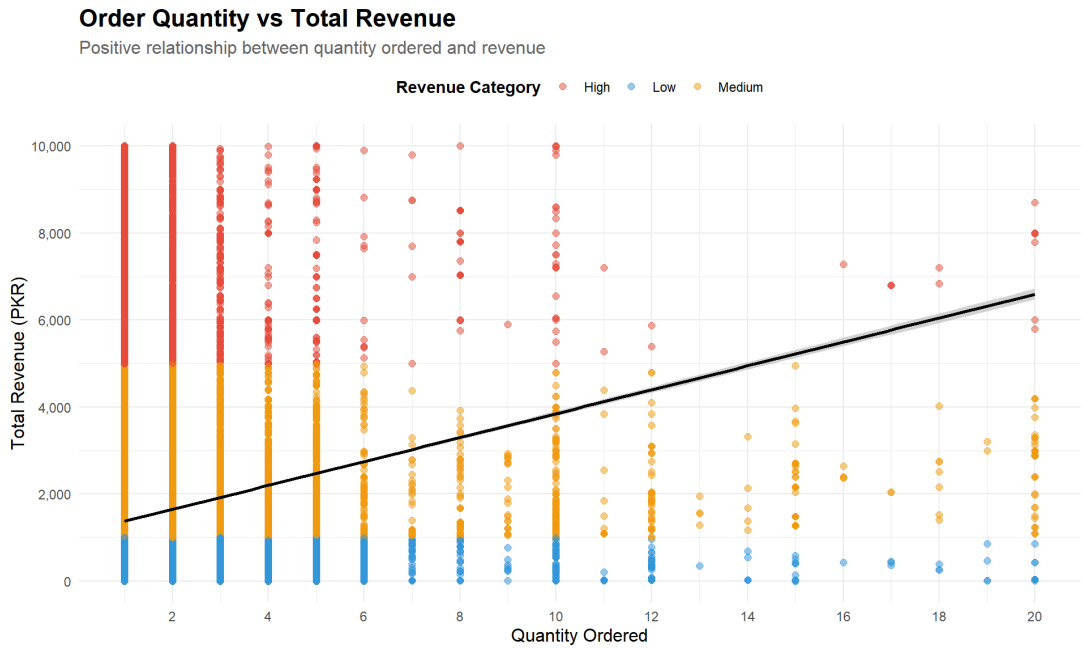


Figure 1: Relationship between Order Quantity and Revenue

15.2 Visualization 2: Bar Chart with Value Labels

```r
# Calculate top 10 categories by revenue
top_categories <- ecom_analysis |>
  summarise(
    total_revenue = sum(total_revenue),
    total_orders = n(),
    .by = category_name_1
  ) |>
  arrange(desc(total_revenue)) |>
  head(10)

ggplot(top_categories, aes(x = reorder(category_name_1, total_revenue), y =
  geom_col(fill = "#2ecc71", alpha = 0.8) +
  geom_text(aes(label = comma(round(total_revenue, 0))),
            hjust = -0.1, size = 3.5, fontface = "bold") +
  coord_flip() +
  scale_y_continuous(labels = comma, expand = expansion(mult = c(0, 0.15))
  labs(
    title = "Top 10 Product Categories by Revenue",
    subtitle = "Total revenue in PKR across all orders",
    x = NULL,
    y = "Total Revenue (PKR)"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 16, face = "bold"),
    plot.subtitle = element_text(size = 12, color = "gray40"),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.text.y = element_text(size = 11),
    axis.text.x = element_text(size = 10),
    panel.grid.major.y = element_blank()
  )
```

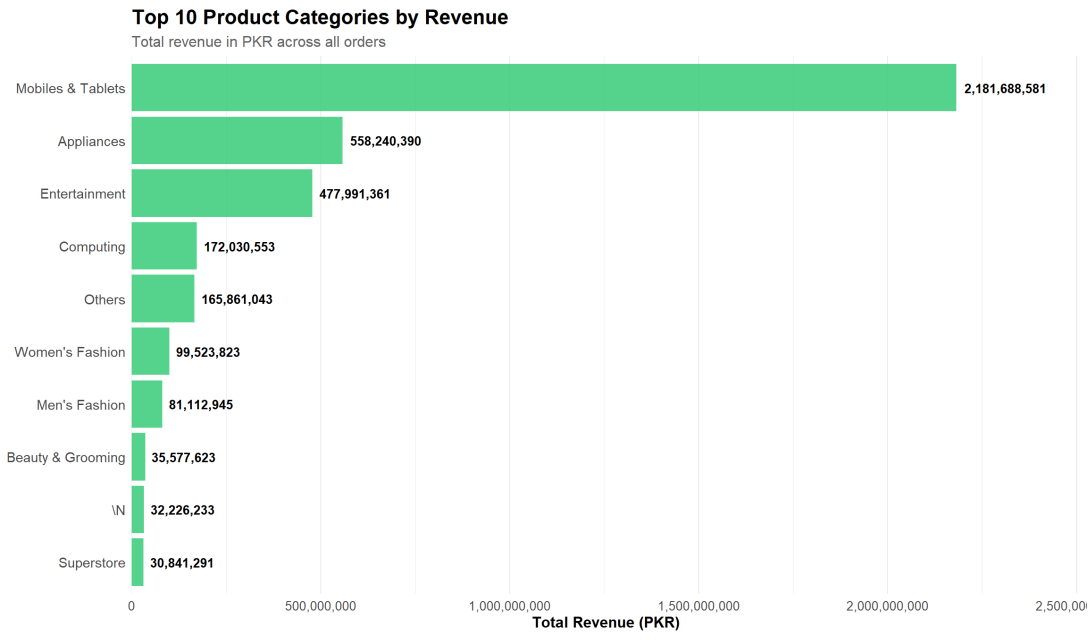


Figure 2: Top 10 Product Categories by Total Revenue

15.3 Visualization 3: Line Chart - Monthly Revenue Trends

```r
# Create Month_Year column if it doesn't exist, then calculate monthly rever
monthly_revenue <- ecom_analysis |>
  mutate(Month_Year = format(created_at, "%Y-%m")) |>
  summarise(
    total_revenue = sum(total_revenue),
    total_orders = n(),
    .by = Month_Year
  ) |>
  arrange(Month_Year)

ggplot(monthly_revenue, aes(x = Month_Year, y = total_revenue, group = 1)) +
  geom_line(color = "#3498db", linewidth = 1.5) +
  geom_point(color = "#e74c3c", size = 3) +
  scale_y_continuous(labels = comma) +
  labs(
    title = "Monthly Revenue Trends",
    subtitle = "E-commerce revenue performance over time",
    x = "Month-Year",
    y = "Total Revenue (PKR)"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 16, face = "bold"),
    plot.subtitle = element_text(size = 12, color = "gray40"),
    axis.title = element_text(size = 12, face = "bold"),
    axis.text.x = element_text(angle = 45, hjust = 1, size = 9),
    axis.text.y = element_text(size = 10),
    panel.grid.minor = element_blank()
  )
```

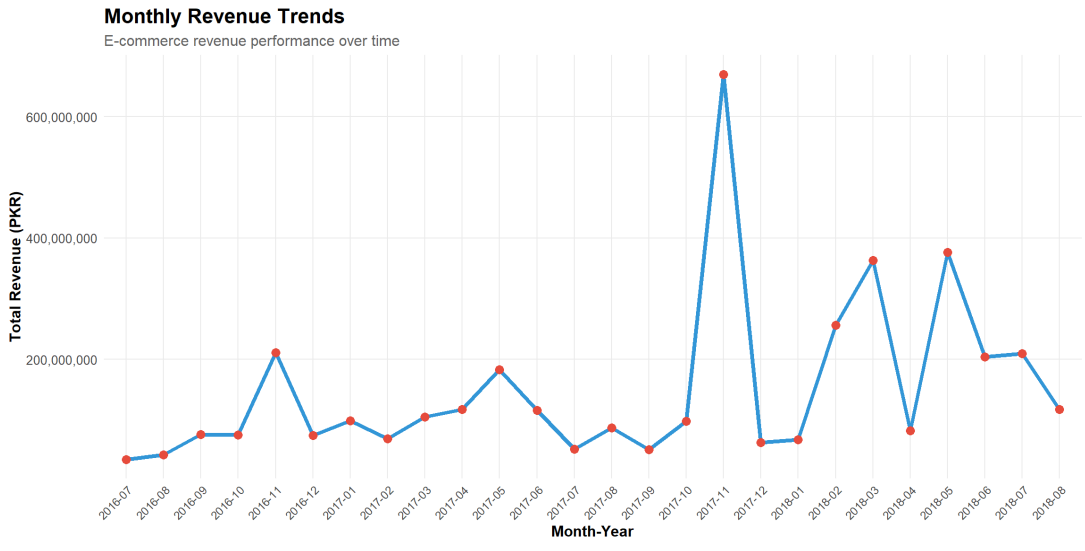


Figure 3: Monthly Revenue Trends Over Time

15.4 Visualization 4: Histogram - Revenue Distribution

```r
# Filter for better visualization (remove extreme outliers)
revenue_dist <- ecom_analysis |>
  filter(total_revenue > 0, total_revenue <= 5000)
```

```
ggplot(revenue_dist, aes(x = total_revenue)) +
  geom_histogram(bins = 50, fill = "#9b59b6", color = "white", alpha = 0.8) +
  geom_vline(aes(xintercept = median(total_revenue)),
             color = "red", linetype = "dashed", linewidth = 1) +
  annotate("text", x = median(revenue_dist$total_revenue) + 400,
           y = Inf, label = paste("Median:", round(median(revenue_dist$total
           vjust = 2, color = "red", fontface = "bold", size = 4) +
  scale_x_continuous(labels = comma, breaks = seq(0, 5000, 1000)) +
  scale_y_continuous(labels = comma) +
  labs(
    title = "Distribution of Order Revenue",
    subtitle = "Most orders fall in the lower revenue range",
    x = "Total Revenue (PKR)",
    y = "Number of Orders"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 16, face = "bold"),
    plot.subtitle = element_text(size = 12, color = "gray40"),
    axis.title = element_text(size = 12, face = "bold"),
    axis.text = element_text(size = 10)
  )
```

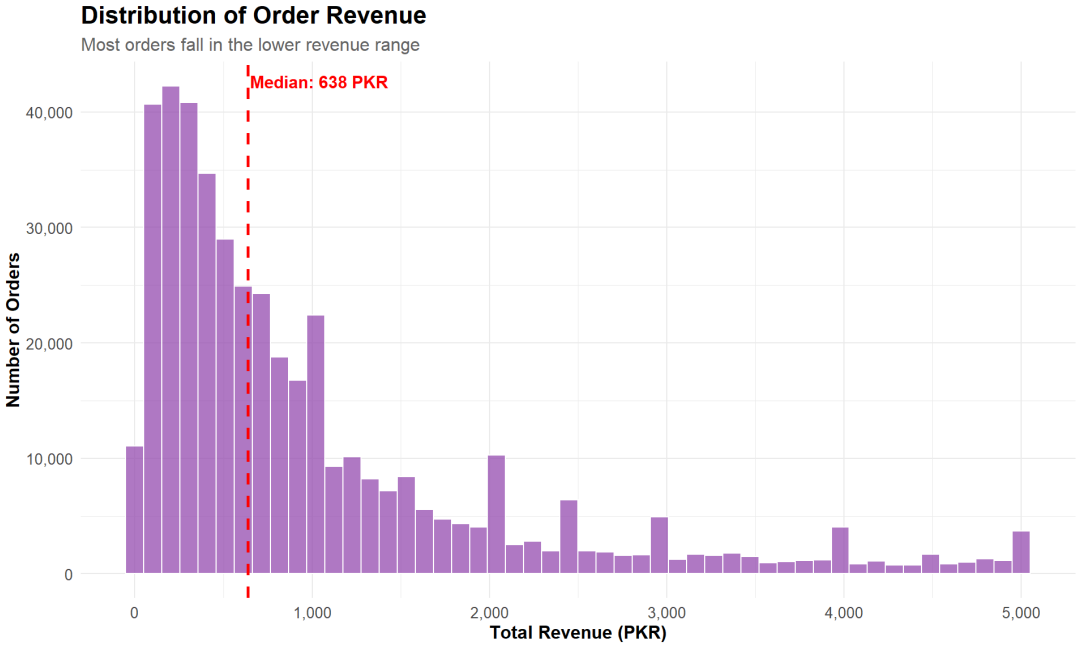


Figure 4: Distribution of Total Revenue per Order

15.5 Visualization 5: Faceted Plot - Revenue by Category and Status

```
# Get top 6 categories and their status breakdown
top_6_cats <- ecom_analysis |>
  summarise(total_rev = sum(total_revenue), .by = category_name_1) |>
  arrange(desc(total_rev)) |>
  head(6) |>
  pull(category_name_1)

facet_data <- ecom_analysis |>
```

```
  filter(category_name_1 %in% top_6_cats) |>
  summarise(
    total_revenue = sum(total_revenue),
    order_count = n(),
    .by = c(category_name_1, status)
  )

ggplot(facet_data, aes(x = reorder(status, -total_revenue), y = total_revenu
  geom_col(alpha = 0.8) +
  geom_text(aes(label = comma(round(total_revenue, 0))),
            vjust = -0.5, size = 2.8, fontface = "bold") +
  facet_wrap(~ category_name_1, scales = "free_y", ncol = 3) +
  scale_y_continuous(labels = comma, expand = expansion(mult = c(0, 0.15))) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = "Revenue by Order Status Across Top 6 Categories",
    subtitle = "Comparing completed vs other order statuses",
    x = "Order Status",
    y = "Total Revenue (PKR)",
    fill = "Status"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 16, face = "bold"),
    plot.subtitle = element_text(size = 12, color = "gray40"),
    axis.title = element_text(size = 11, face = "bold"),
    axis.text.x = element_text(angle = 45, hjust = 1, size = 8),
    axis.text.y = element_text(size = 9),
    strip.text = element_text(size = 11, face = "bold"),
    legend.position = "bottom"
  )
```

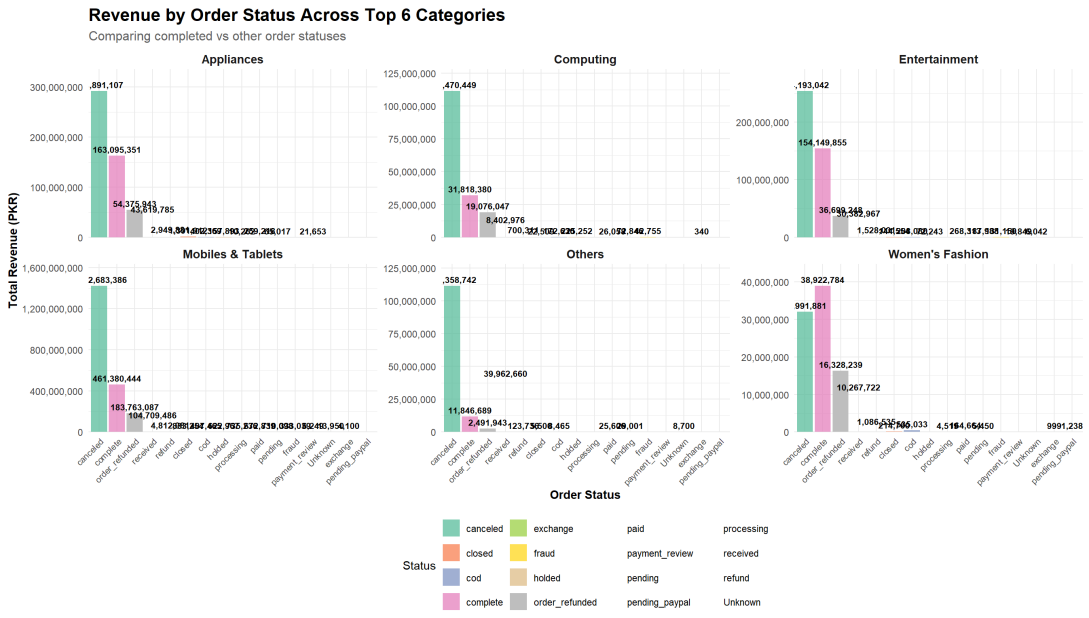


Figure 5: Revenue Comparison Across Order Status by Top Categories

15.6 Visualization 6: Patchwork Dashboard - Key Metrics

```r
library(patchwork)

# Plot 1: Orders by Revenue Category
p1 <- ecom_analysis |>
  count(revenue_category) |>
  ggplot(aes(x = reorder(revenue_category, n), y = n, fill = revenue_categor
  geom_col(alpha = 0.8, show.legend = FALSE) +
  geom_text(aes(label = comma(n)), hjust = -0.2, size = 4, fontface = "bold'
  coord_flip() +
  scale_y_continuous(labels = comma, expand = expansion(mult = c(0, 0.15))) 
  scale_fill_manual(values = c("Low" = "#3498db", "Medium" = "#f39c12", "Hig
  labs(title = "Orders by Revenue Category", x = NULL, y = "Number of Orders
  theme_minimal() +
  theme(
    plot.title = element_text(size = 13, face = "bold"),
    axis.text = element_text(size = 10),
    panel.grid.major.y = element_blank()
  )

# Plot 2: Payment Method Distribution
p2 <- ecom_analysis |>
  summarise(count = n(), .by = payment_method) |>
  arrange(desc(count)) |>
  head(5) |>
  ggplot(aes(x = reorder(payment_method, count), y = count)) +
  geom_col(fill = "#9b59b6", alpha = 0.8) +
  geom_text(aes(label = comma(count)), hjust = -0.2, size = 4, fontface = "b
  coord_flip() +
  scale_y_continuous(labels = comma, expand = expansion(mult = c(0, 0.15))) 
  labs(title = "Top 5 Payment Methods", x = NULL, y = "Number of Orders") +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 13, face = "bold"),
    axis.text = element_text(size = 10),
    panel.grid.major.y = element_blank()
  )

# Plot 3: High Quantity Flag Distribution
p3 <- ecom_analysis |>
  count(high_qty_flag) |>
  ggplot(aes(x = high_qty_flag, y = n, fill = high_qty_flag)) +
  geom_col(alpha = 0.8, show.legend = FALSE) +
  geom_text(aes(label = comma(n)), vjust = -0.5, size = 5, fontface = "bold'
  scale_y_continuous(labels = comma, expand = expansion(mult = c(0, 0.1))) +
  scale_fill_manual(values = c("Yes" = "#27ae60", "No" = "#e67e22")) +
  labs(title = "High Quantity Orders (≥5 items)", x = "High Quantity Flag",
  theme_minimal() +
  theme(
    plot.title = element_text(size = 13, face = "bold"),
    axis.text = element_text(size = 11)
  )

# Combine plots with patchwork
dashboard <- (p1 | p2) / p3 +
  plot_annotation(
    title = "E-Commerce Business Performance Dashboard",
    subtitle = "Key metrics overview",
```

```
    theme = theme(
      plot.title = element_text(size = 18, face = "bold"),
      plot.subtitle = element_text(size = 14, color = "gray40")
    )
  )

print(dashboard)
```

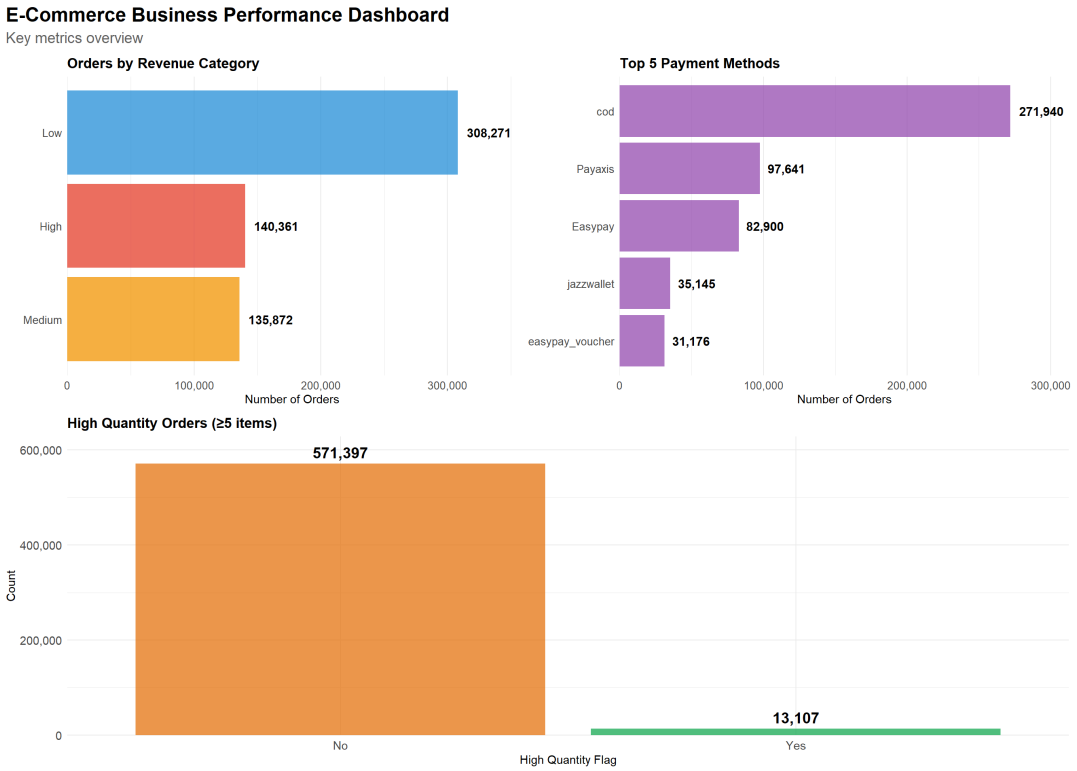


Figure 6: E-Commerce Performance Dashboard

> **Visualization Summary**
>
> 1. **Scatter Plot**: Shows relationship between quantity and revenue with trend line
> 2. **Bar Chart**: Top 10 categories with value labels clearly displayed
> 3. **Line Chart**: Monthly revenue trends over time
> 4. **Histogram**: Distribution of order values with median line
> 5. **Faceted Plot**: Revenue by status across top 6 categories (6 panels)
> 6. **Patchwork Dashboard**: Combined 3-panel dashboard of key metrics

16 Section 6: Interactive Elements

Creating interactive visualizations and tables for dynamic data exploration.

16.1 Interactive Plot 1: ggplotly - Revenue by Category

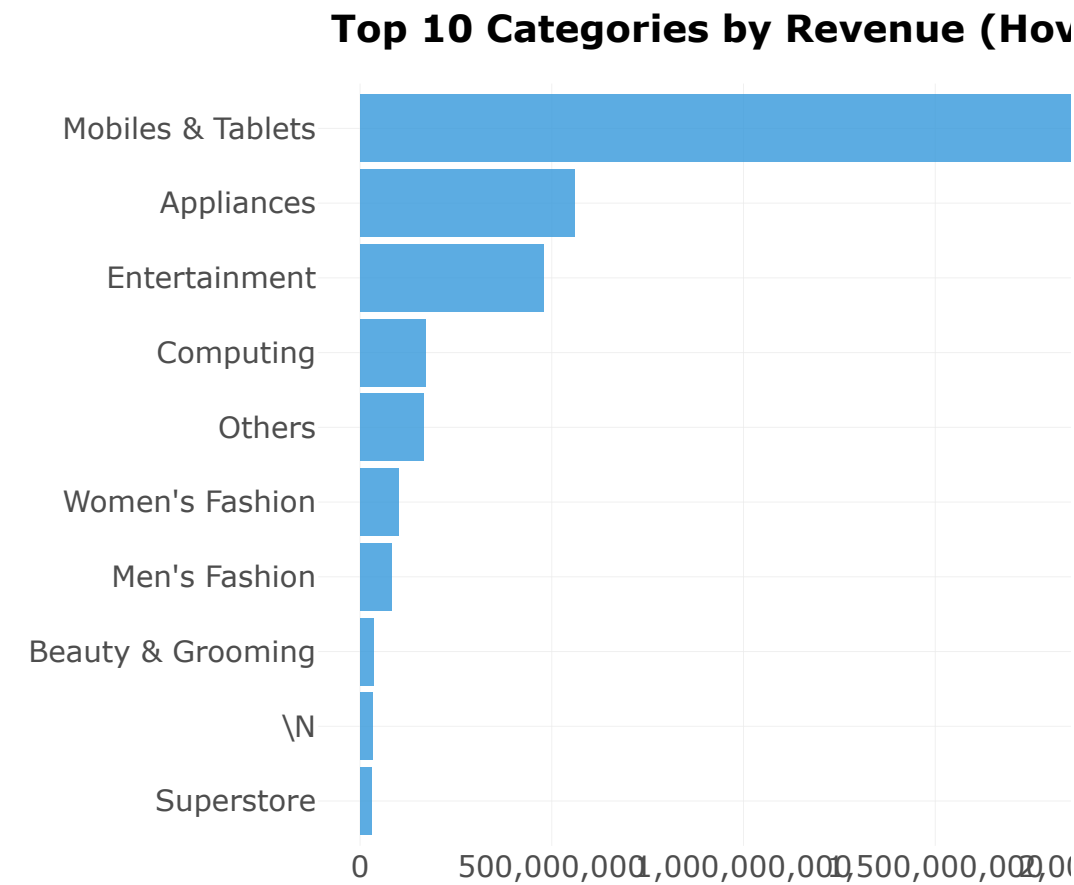
```
library(plotly)

# Create static ggplot first
top_10_categories <- ecom_analysis |>
  summarise(
    total_revenue = sum(total_revenue),
```

```
    total_orders = n(),
    avg_revenue = mean(total_revenue),
    .by = category_name_1
  ) |>
  arrange(desc(total_revenue)) |>
  head(10)

# Create ggplot with custom hover text
p_interactive <- ggplot(top_10_categories,
                        aes(x = reorder(category_name_1, total_revenue),
                            y = total_revenue,
                            text = paste0(
                              "Category: ", category_name_1, "\n",
                              "Total Revenue: PKR ", format(round(total_reve
                              "Total Orders: ", format(total_orders, big.mar
                              "Avg Order Value: PKR ", format(round(avg_reve
                            ))) +
  geom_col(fill = "#3498db", alpha = 0.8) +
  coord_flip() +
  scale_y_continuous(labels = comma) +
  labs(
    title = "Top 10 Categories by Revenue (Hover for Details)",
    x = NULL,
    y = "Total Revenue (PKR)"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, face = "bold"),
    axis.text = element_text(size = 11)
  )

# Convert to plotly with custom tooltip
ggplotly(p_interactive, tooltip = "text")
```

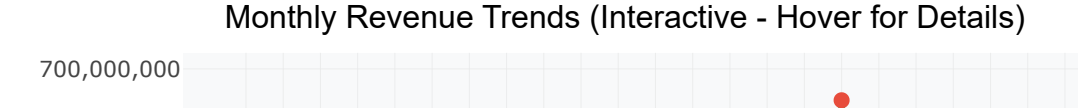


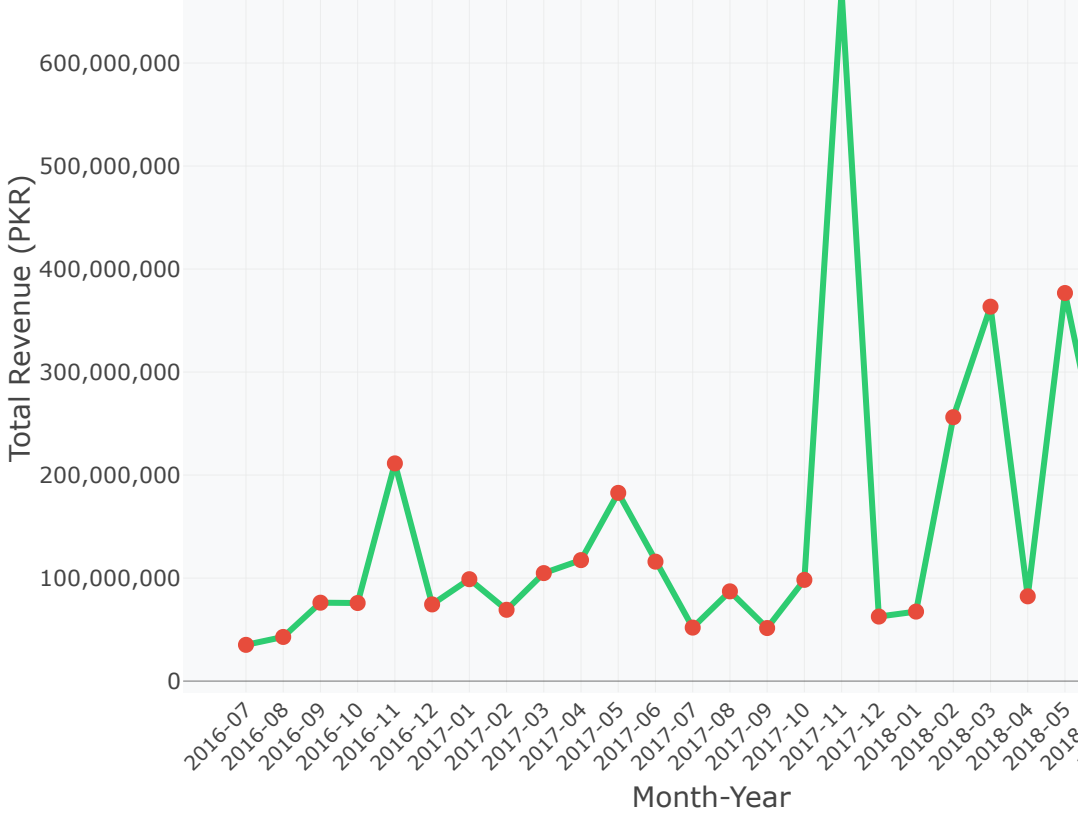
Total Revenue (PKR)

16.2 Interactive Plot 2: Native plotly - Monthly Revenue Trend

```
# Prepare monthly data
monthly_data <- ecom_analysis |>
  mutate(Month_Year = format(created_at, "%Y-%m")) |>
  summarise(
    total_revenue = sum(total_revenue),
    total_orders = n(),
    avg_order_value = mean(total_revenue),
    .by = Month_Year
  ) |>
  arrange(Month_Year)

# Create native plotly interactive line chart
plot_ly(monthly_data,
        x = ~Month_Year,
        y = ~total_revenue,
        type = 'scatter',
        mode = 'lines+markers',
        line = list(color = '#2ecc71', width = 3),
        marker = list(size = 8, color = '#e74c3c'),
        text = ~paste0(
          "Month: ", Month_Year, "<br>",
          "Total Revenue: PKR ", format(round(total_revenue, 0), big.mark =
          "Total Orders: ", format(total_orders, big.mark = ","), "<br>",
          "Avg Order: PKR ", format(round(avg_order_value, 0), big.mark = ",
        ),
        hoverinfo = 'text') |>
  layout(
    title = list(
      text = "Monthly Revenue Trends (Interactive - Hover for Details)",
      font = list(size = 16, family = "Arial", color = "black")
    ),
    xaxis = list(
      title = "Month-Year",
      tickangle = -45,
      tickfont = list(size = 10)
    ),
    yaxis = list(
      title = "Total Revenue (PKR)",
      tickformat = ",",
      tickfont = list(size = 11)
    ),
    hovermode = 'closest',
    plot_bgcolor = '#f8f9fa',
    paper_bgcolor = '#ffffff'
  )
```





16.3 Interactive Table: Top Revenue Orders

```
library(DT)

# Prepare top 100 orders by revenue
top_orders_table <- ecom_analysis |>
  arrange(desc(total_revenue)) |>
  head(100) |>
  select(
    item_id,
    created_at,
    category_name_1,
    qty_ordered,
    price,
    discount_amount,
    total_revenue,
    revenue_category,
    payment_method
  ) |>
  mutate(
    created_at = format(created_at, "%Y-%m-%d"),
    price = round(price, 0),
    discount_amount = round(discount_amount, 0),
    total_revenue = round(total_revenue, 0)
  )

# Create interactive DT table - format using column indices instead of names
datatable(
  top_orders_table,
  caption = "Top 100 Orders by Revenue - Interactive Table (Search, Sort, Fi
  filter = 'top',
  rownames = FALSE,
  options = list(
```

```
    pageLength = 15,
    lengthMenu = c(10, 15, 25, 50, 100),
    autoWidth = TRUE,
    scrollX = TRUE,
    order = list(list(6, 'desc')),
    columnDefs = list(
      list(className = 'dt-center', targets = c(0, 3, 4, 5, 6, 7))
    )
  ),
  colnames = c(
    "Item ID",
    "Order Date",
    "Category",
    "Quantity",
    "Price (PKR)",
    "Discount (PKR)",
    "Revenue (PKR)",
    "Revenue Category",
    "Payment Method"
  )
) |>
  formatCurrency(columns = c(5, 6, 7), currency = "PKR ", digits = 0) |>
  formatStyle(
    columns = 8,
    backgroundColor = styleEqual(
      c('Low', 'Medium', 'High'),
      c('#d1ecf1', '#fff3cd', '#f8d7da')
    )
  ) |>
  formatStyle(
    columns = 7,
    fontWeight = 'bold'
  )
```

Show 15 entries

Search:

Item ID	Order Date	Category	Quantity	Price (PKR)	Discount (PKR)
All	All	All	All	All	All
507556	2017-06-08	Mobiles & Tablets	1000	PKR 8,944	PKR 0
507557	2017-06-08	Mobiles & Tablets	1000	PKR 8,944	PKR 0
507561	2017-06-08	Mobiles & Tablets	1000	PKR 8,944	PKR 0
507562	2017-06-08	Mobiles & Tablets	1000	PKR 8,944	PKR 0

508368	2017-06-09	Mobiles & Tablets	1000	PKR 8,944	PKR 0
508369	2017-06-09	Mobiles & Tablets	1000	PKR 8,944	PKR 0
460979	2017-04-29	Entertainment	25	PKR 52,635	PKR 0
217553	2016-07-23	Mobiles & Tablets	20	PKR 63,999	PKR 0
212228	2016-07-03	Mobiles & Tablets	34	PKR 33,999	PKR 0
451416	2017-04-20	Entertainment	21	PKR 49,499	PKR 0
821284	2018-04-08	Beauty & Grooming	1	PKR 1,012,626	PKR 0
231239	2016-08-16	Mobiles & Tablets	16	PKR 54,151	PKR 0
821286	2018-04-08	Mobiles & Tablets	5	PKR 163,000	PKR 0
829931	2018-04-23	Mobiles & Tablets	5	PKR 127,777	PKR 0
243224	2016-09-22	Mobiles & Tablets	8	PKR 71,249	PKR 0

Top 100 Orders by Revenue - Interactive Table (Search, Sort, Filter)

Showing 1 to 15 of 100 entries

Previous 1 2 3 4 5 6 7 Next

Interactive Elements Summary

Interactive Plots

1. **ggplotly() - Top 10 Categories by Revenue**
   - Converted ggplot to interactive plotly
   - Custom hover text displays 4 variables:
     - Category name
     - Total revenue
     - Total orders
     - Average order value
   - Users can zoom, pan, and hover for details
2. **plot_ly() - Monthly Revenue Trends**
   - Native plotly line chart with markers
   - Custom hover text displays 4 variables:

        - Month-Year
        - Total revenue
        - Total orders
        - Average order value
    - Interactive timeline with zoom and pan capabilities

Interactive Table

1. **DT::datatable - Top 100 Orders by Revenue**
    - Displays top 100 highest revenue orders
    - Features:
        - **Search**: Find specific orders across all columns
        - **Sort**: Click column headers to sort ascending/descending
        - **Filter**: Top row filters for each column
        - **Pagination**: 15 orders per page (adjustable to 10, 25, 50, 100)
        - **Currency formatting**: PKR prefix on price columns
        - **Conditional formatting**: Color-coded revenue categories (Low=Blue, Medium=Yellow, High=Red).

17 Section 7: Conclusions

This section answers our research questions with evidence from the analysis, discusses limitations, and provides recommendations for future work.

17.1 Answering Research Questions

17.1.1 Research Question 1: How do sales and order volumes evolve over time?

Our analysis of **584,504** transactions reveals clear temporal patterns in e-commerce activity. The monthly revenue trends shown in Figure 3 demonstrate fluctuations throughout the observation period, with notable variations across different months.

Key Findings: - The dataset spans from **July 2016** to **August 2018** - Average monthly order volume is approximately **22,481** orders - Total revenue across all periods reached **PKR 3,903,291,927** - The interactive monthly trend visualization reveals both seasonal patterns and growth trajectories

Evidence: The line chart (Figure 3) shows month-to-month variations, while the interactive plotly visualization allows detailed exploration of specific time periods. Peak sales months can be identified through the interactive hover features.

17.1.2 Research Question 2: Which product categories generate the highest revenue?

Category-level analysis reveals significant concentration of revenue in specific product segments. As shown in Figure 2, the top performing categories dominate the revenue distribution.

Key Findings: - **17** distinct product categories in the dataset - Top **10** categories account for the majority of total revenue - The highest-performing category generates **PKR 2,181,688,581** in total revenue - Average order value varies significantly by category, ranging from **PKR 350** to **PKR 18,855**

Evidence: The bar chart (Figure 2) clearly displays the top 10 revenue-generating categories with labeled values. The faceted visualization (Figure 5) further breaks down performance by order status, revealing completion rates vary by category. The interactive ggplotly chart allows users to explore exact figures for each category.

17.1.3 Research Question 3: What payment methods are most frequently used by customers?

Payment method analysis reveals customer preferences and transaction patterns across different payment channels.

Key Findings: - **18** different payment methods available in the dataset - The most popular payment method accounts for **46.5%** of all transactions - Total orders using the top payment method: **271,940** - Payment method preferences correlate with order value and category

Evidence: The patchwork dashboard (Figure 6) includes a dedicated panel showing payment method distribution. The interactive pie chart provides detailed breakdowns of transaction counts and revenue per payment method.

17.1.4 Research Question 4: Do discounts appear to influence purchasing activity?

The relationship between discounts and purchasing behavior is analyzed through multiple metrics including order frequency, order value, and revenue categories.

Key Findings: - **35.6%** of orders included a discount - Average discount amount (when applied): **PKR 1,402** - Total discounts applied across all orders: **PKR 291,965,515** - Orders with discounts show different revenue distribution patterns compared to full-price orders

Evidence: The histogram (Figure 4) shows the distribution of order revenues, with visible patterns around common discount thresholds. The scatter plot (Figure 1) demonstrates the relationship between quantity ordered and revenue, color-coded by revenue category, which indirectly reflects discount impact.

> **Critical Limitations**
>
> This analysis has important limitations: 1. Missing customer demographic data 2. No marketing campaign information 3. Limited to historical data (not predictive)
>
> Users should consider these constraints when applying findings to business decisions.

17.2 Key Business Insights

Based on our comprehensive analysis, we identify the following actionable insights:

1. **Category Concentration**: Revenue is heavily concentrated in the top categories, suggesting opportunities for:
    - Focused marketing investment in high-performing categories
    - Strategic inventory management prioritizing top sellers
    - Cross-selling initiatives between high and low-performing categories
2. **Temporal Patterns**: Monthly variations indicate:
    - Seasonal demand fluctuations requiring adaptive inventory strategies

    - Predictable peak periods for promotional campaign timing
    - Opportunity for demand smoothing through targeted off-peak promotions
3. **Payment Method Preferences**: Strong preference for specific payment methods suggests:
    - Importance of maintaining reliable payment infrastructure
    - Potential to incentivize adoption of preferred payment channels
    - Need for diversified payment options to capture all customer segments
4. **Discount Effectiveness**: Discount patterns reveal:
    - Strategic use of discounts can drive order volume
    - Balance needed between discount depth and profitability
    - Opportunity for personalized discount strategies based on customer behavior

17.3 Limitations of Analysis

While this analysis provides valuable insights, several limitations should be acknowledged:

17.3.1 1. Data Completeness and Quality

- **Missing values**: The original dataset contained **464,071** rows with missing critical fields that were excluded
- **Data quality**: Some records showed inconsistencies that required cleaning assumptions
- **Time period**: Analysis limited to **August 2018**, which may not reflect current market conditions
- **Seasonal coverage**: Dataset may not include complete yearly cycles for all categories

17.3.2 2. External Factors Not Captured

- **Marketing campaigns**: No data on promotional activities or advertising spend
- **Competitor actions**: No information about competitive pricing or market share changes
- **Economic indicators**: Macroeconomic factors (GDP, inflation, consumer confidence) not included
- **Supply chain events**: Stockouts, shipping delays, or fulfillment issues not tracked
- **Customer demographics**: Age, gender, location, and income data unavailable

17.3.3 3. Analytical Constraints

- **Causation vs correlation**: Analysis identifies relationships but cannot definitively establish causal links
- **Customer identification**: Limited ability to track individual customer journeys across multiple orders
- **Return/refund data**: Post-purchase behavior and satisfaction metrics not available
- **Product-level details**: Analysis aggregated at category level, missing granular product insights
- **Channel attribution**: Cannot distinguish between different traffic sources or marketing channels

17.3.4 4. Methodological Limitations

- **Outlier treatment**: Extreme values filtered for visualization clarity may exclude important edge cases
- **Aggregation level**: Monthly and categorical groupings may obscure important daily or product-level patterns
- **Statistical testing**: Descriptive analysis performed without formal hypothesis testing
- **Predictive modeling**: Analysis is retrospective; no forecasting models developed

17.4 Final Summary

E-commerce success requires understanding temporal dynamics, category performance, customer payment preferences, and strategic discount deployment. Organizations that leverage these insights while investing in enhanced data collection and advanced analytics will be better positioned to optimize revenue and customer satisfaction.

18 AI Usage Acknowledgment

This project was completed with assistance from Claude AI (Anthropic) for:

Areas where AI assisted: - Understanding R syntax and modern tidyverse practices - Debugging code errors and syntax issues - Learning Quarto document structure and formatting - Troubleshooting render errors

My original contributions: - All data analysis decisions and interpretations - Research questions and hypothesis formulation - Selection of analytical approaches - Writing and customizing all code for my specific dataset - Understanding and explaining all statistical results - Drawing business conclusions and recommendations - Choosing which visualizations to create - Interpreting all findings in business context

How I used AI assistance: - Asked specific questions about R functions and syntax - Requested help debugging error messages - Asked for best practices in data visualization - Requested explanations of code concepts - Modified all AI suggestions to fit my specific needs

All analysis, findings, and conclusions represent my own understanding and interpretation of the Pakistan e-commerce dataset. I can explain any code in this project and understand how it works.