Project: Amazone Sales Report Analysis; By Anumoy Modak. Date: 22/11/2024

1) Data Visualization Excercise : Data Frame

In [2]:
```
import pandas as pd

# Loading  the uploaded CSV file to inspect its contents

file_path = r'C:\Users\LENOVO\OneDrive\Desktop\Independent projects\archive\Amazon Sale Report.csv'
data = pd.read_csv(file_path)

# Display basic information and first few rows of the dataset

data_info = data.info()
data_head = data.head()

data_info, data_head
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_19048\1804306606.py:6: DtypeWarning: Columns (23) have mixed types. Specify dtype option on import or set low_memory=False.
  data = pd.read_csv(file_path)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 128975 entries, 0 to 128974
Data columns (total 24 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   index               128975 non-null  int64
 1   Order ID            128975 non-null  object
 2   Date                128975 non-null  object
 3   Status              128975 non-null  object
 4   Fulfilment          128975 non-null  object
 5   Sales Channel       128975 non-null  object
 6   ship-service-level  128975 non-null  object
 7   Style               128975 non-null  object
 8   SKU                 128975 non-null  object
 9   Category            128975 non-null  object
 10  Size                128975 non-null  object
 11  ASIN                128975 non-null  object
 12  Courier Status      122103 non-null  object
 13  Qty                 128975 non-null  int64
 14  currency            121180 non-null  object
 15  Amount              121180 non-null  float64
 16  ship-city           128942 non-null  object
 17  ship-state          128942 non-null  object
 18  ship-postal-code    128942 non-null  float64
 19  ship-country        128942 non-null  object
 20  promotion-ids       79822 non-null   object
 21  B2B                 128975 non-null  bool
 22  fulfilled-by        39277 non-null   object
 23  Unnamed: 22         79925 non-null   object
dtypes: bool(1), float64(2), int64(2), object(19)
memory usage: 22.8+ MB
```

Out[2]:
```
(None,
   index             Order ID      Date                       Status  \
0      0  405-8078784-5731545  04-30-22                    Cancelled
1      1  171-9198151-1101146  04-30-22  Shipped - Delivered to Buyer
2      2  404-0687676-7273146  04-30-22                      Shipped
3      3  403-9615377-8133951  04-30-22                    Cancelled
4      4  407-1069790-7240320  04-30-22                      Shipped

  Fulfilment Sales Channel  ship-service-level    Style              SKU  \
0   Merchant      Amazon.in           Standard   SET389   SET389-KR-NP-S
1   Merchant      Amazon.in           Standard  JNE3781  JNE3781-KR-XXXL
2     Amazon      Amazon.in          Expedited  JNE3371    JNE3371-KR-XL
3   Merchant      Amazon.in           Standard    J0341       J0341-DR-L
4     Amazon      Amazon.in          Expedited  JNE3671  JNE3671-TU-XXXL

        Category  ... currency  Amount    ship-city   ship-state  \
0            Set  ...      INR  647.62       MUMBAI  MAHARASHTRA
1          kurta  ...      INR  406.00    BENGALURU    KARNATAKA
2          kurta  ...      INR  329.00  NAVI MUMBAI  MAHARASHTRA
3  Western Dress  ...      INR  753.33   PUDUCHERRY   PUDUCHERRY
4            Top  ...      INR  574.00      CHENNAI   TAMIL NADU

  ship-postal-code  ship-country  \
0         400081.0            IN
1         560085.0            IN
2         410210.0            IN
3         605008.0            IN
4         600073.0            IN

                                        promotion-ids    B2B  fulfilled-by  \
0                                                 NaN  False     Easy Ship
1  Amazon PLCC Free-Financing Universal Merchant ...  False     Easy Ship
2       IN Core Free Shipping 2015/04/08 23-48-5-108   True           NaN
3                                                 NaN  False     Easy Ship
4                                                 NaN  False           NaN

  Unnamed: 22
0         NaN
1         NaN
2         NaN
3         NaN
4         NaN

[5 rows x 24 columns])
```

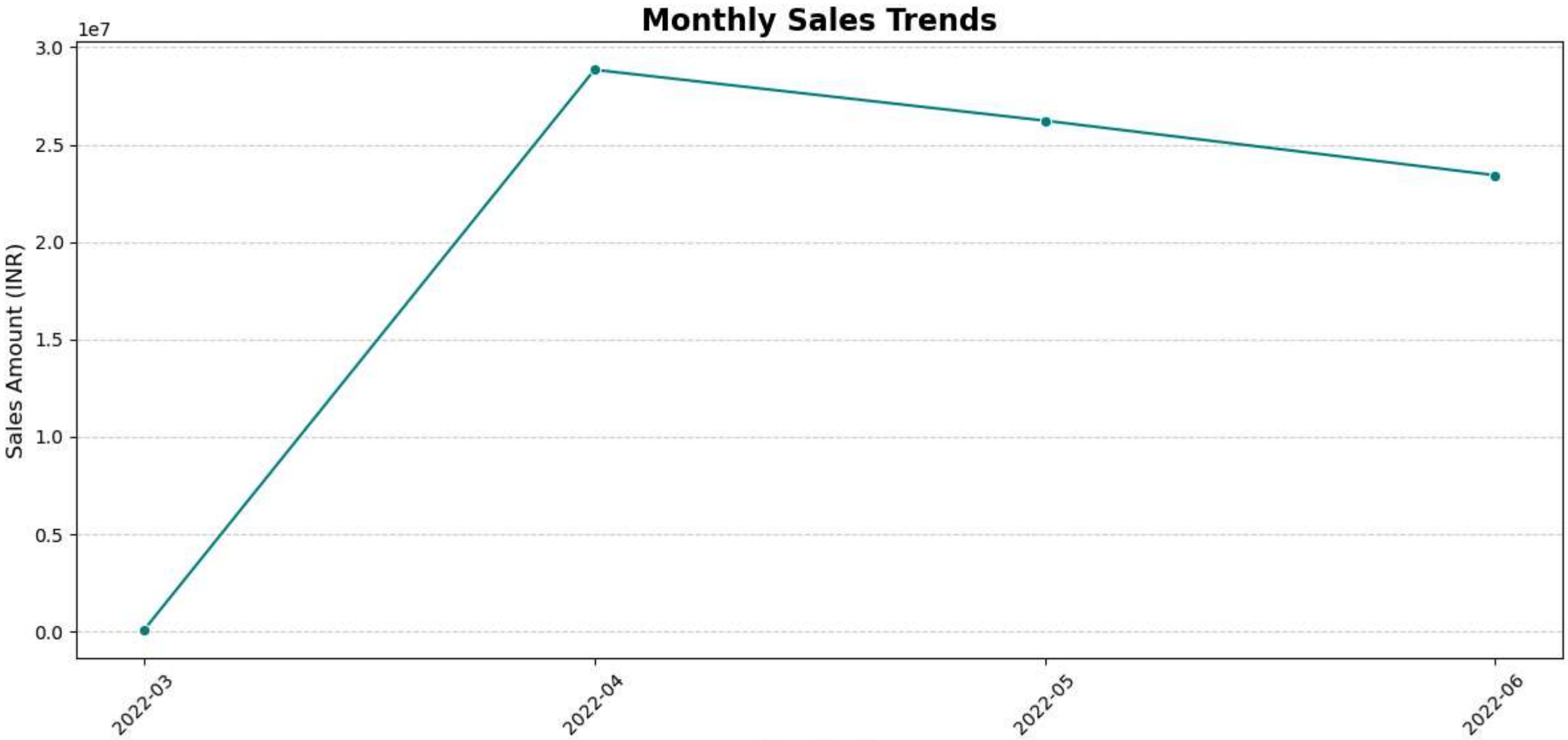
Dataset contains 128,975 rows and 24 columns, including details like order ID, date, status, fulfillment type, sales channel, product details, amounts, and shipment information. Here's what I observed:

A.Key Columns: Sales Metrics: Amount, Qty; Order Details: Order ID, Date, Status; Product Details: Style, Category, SKU; Shipment Details: ship-city, ship-state, ship-country; Some columns have missing or inconsistent data (e.g., currency, fulfilled-by, promotion-ids).

2) VISUALIZATION : SALES TREND AND ANALYSIS

In [3]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

```
# Data preprocessing: Convert dates to datetime format and check for missing values

data['Date'] = pd.to_datetime(data['Date'], errors='coerce')
missing_summary = data.isnull().sum()

# Extracting a year-month column for trend analysis

data['Year-Month'] = data['Date'].dt.to_period('M')

# Summarize sales trends over time (monthly aggregation)

monthly_sales = data.groupby('Year-Month')['Amount'].sum().reset_index()
monthly_sales['Year-Month'] = monthly_sales['Year-Month'].astype(str)

# Create a sales trend visualization

plt.figure(figsize=(12, 6))
sns.lineplot(data=monthly_sales, x='Year-Month', y='Amount', marker='o', color='teal')
plt.title('Monthly Sales Trends', fontsize=16, fontweight='bold')
plt.xlabel('Year-Month', fontsize=12)
plt.ylabel('Sales Amount (INR)', fontsize=12)
plt.xticks(rotation=45, fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.tight_layout()

# Show missing summary and trends visualization

missing_summary, plt.show()
```


Monthly Sales Trends
1e7
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Sales Amount (INR)
2022-03
2022-04
2022-05
2022-06
Year-Month

Out[3]:
```
(index                   0
 Order ID                0
 Date                    0
 Status                  0
 Fulfilment              0
 Sales Channel           0
 ship-service-level      0
 Style                   0
 SKU                     0
 Category                0
 Size                    0
 ASIN                    0
 Courier Status       6872
 Qty                     0
 currency             7795
 Amount               7795
 ship-city              33
 ship-state             33
 ship-postal-code       33
 ship-country           33
 promotion-ids       49153
 B2B                     0
 fulfilled-by        89698
 Unnamed: 22         49050
 dtype: int64,
 None)
```

Findings:

Missing Data:

a)Key fields like fulfilled-by (70%), promotion-ids (38%), and Courier Status (~5%) have missing values. b)Columns such as Amount and currency have 7,795 missing entries.

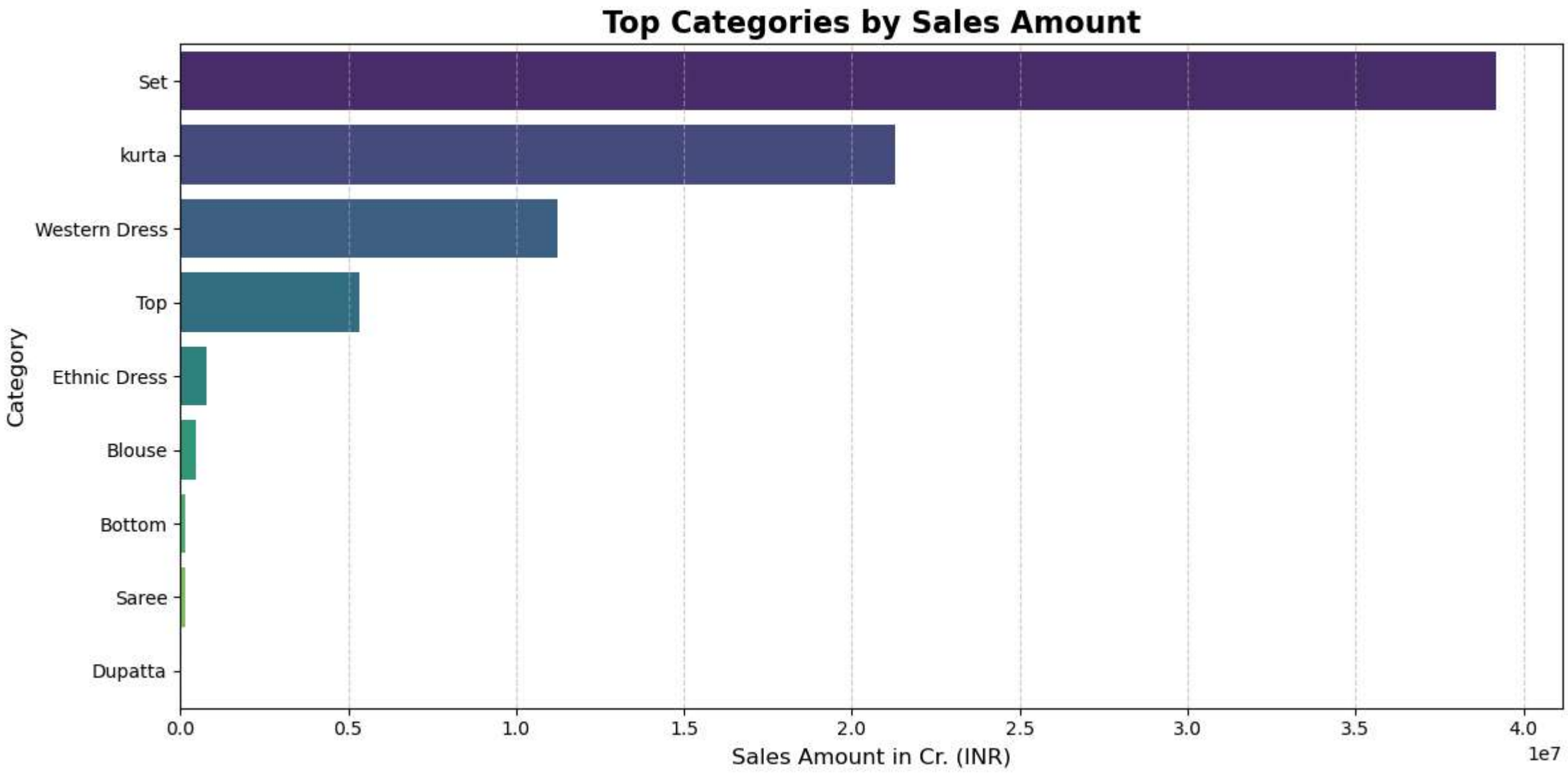
Sales Trends Visualization:

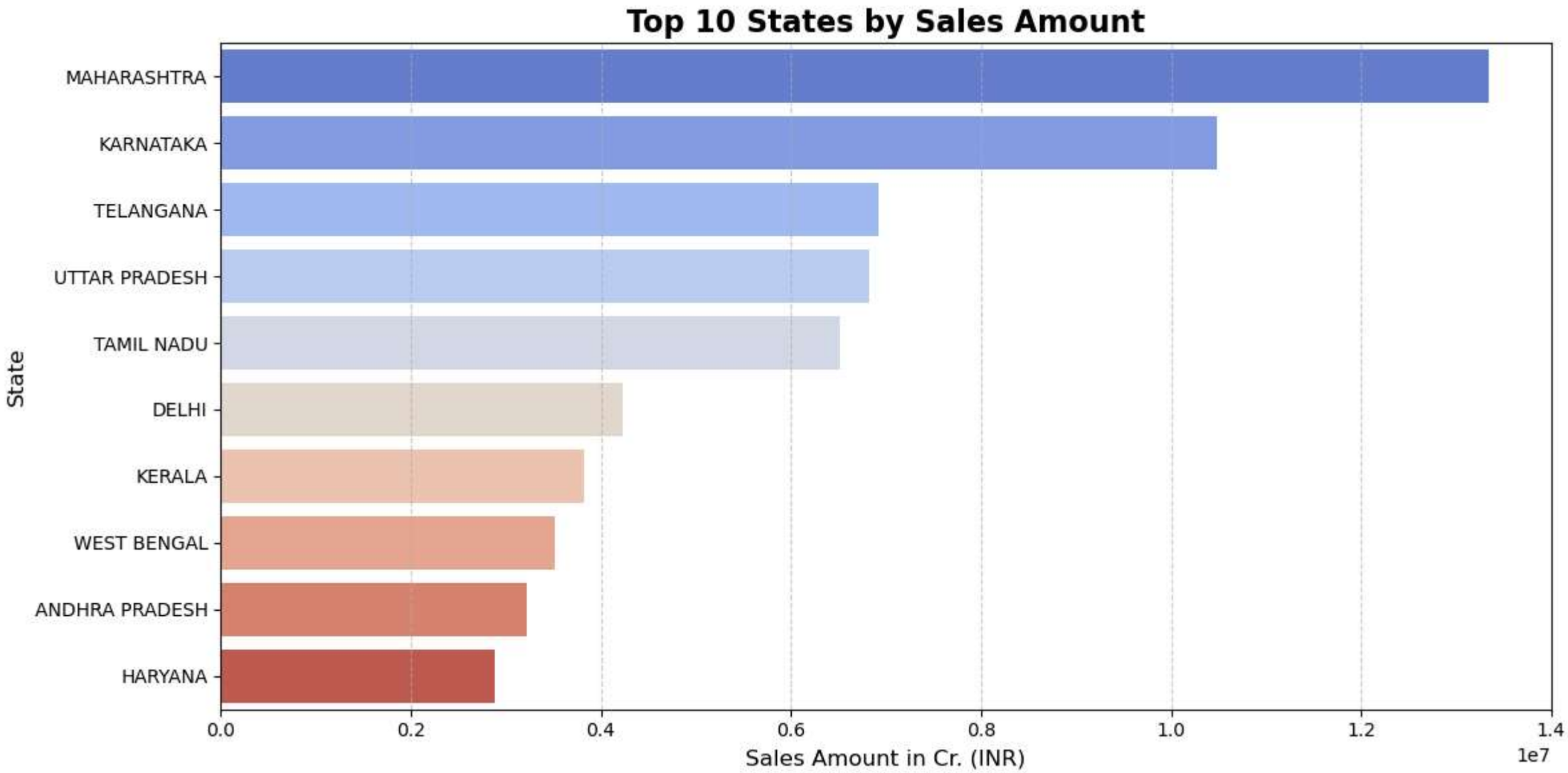
The chart above displays the monthly sales trends. We can observe peaks and troughs to identify seasonal or promotional impacts.

3) VISUALIZATION : PRODUCT PERFORMENCES & GEOGRAPHICAL INSIGHTS (SALES BY STATE) ANALYSIS

In [12]:
```
# Top Performing Categories
top_categories = data.groupby('Category')['Amount'].sum().reset_index().sort_values(by='Amount', ascending=False)

# Top Performing Products (SKUs)
top_products = data.groupby('SKU')['Amount'].sum().reset_index().sort_values(by='Amount', ascending=False).head(10)

# Geographic Insights (Sales by State)
sales_by_state = data.groupby('ship-state')['Amount'].sum().reset_index().sort_values(by='Amount', ascending=False).head(10)

# Visualization: Top Categories
plt.figure(figsize=(12, 6))
sns.barplot(data=top_categories.head(10), x='Amount', y='Category', palette='viridis')
plt.title('Top Categories by Sales Amount', fontsize=16, fontweight='bold')
plt.xlabel('Sales Amount in Cr. (INR)', fontsize=12)
plt.ylabel('Category', fontsize=12)
plt.grid(axis='x', linestyle='--', alpha=0.6)
```

```
plt.tight_layout()

# Visualization: Sales by State
plt.figure(figsize=(12, 6))
sns.barplot(data=sales_by_state, x='Amount', y='ship-state', palette='coolwarm')
plt.title('Top 10 States by Sales Amount', fontsize=16, fontweight='bold')
plt.xlabel('Sales Amount in Cr. (INR)', fontsize=12)
plt.ylabel('State', fontsize=12)
plt.grid(axis='x', linestyle='--', alpha=0.6)
plt.tight_layout()

# Display results
plt.show()

# Return tables for further insights
top_categories.head(10), top_products, sales_by_state
```




Top 10 States by Sales Amount
State
MAHARASHTRA
KARNATAKA
TELANGANA
UTTAR PRADESH
TAMIL NADU
DELHI
KERALA
WEST BENGAL
ANDHRA PRADESH
HARYANA
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1e7
Sales Amount in Cr. (INR)

Out[12]: (Category Amount
5 Set 39204124.03
8 kurta 21299546.70
7 Western Dress 11216072.69
6 Top 5347792.30
3 Ethnic Dress 791217.66
0 Blouse 458408.18
1 Bottom 150667.98
4 Saree 123933.76
2 Dupatta 915.00,
SKU Amount
1346 J0230-SKD-M 527699.20
4548 JNE3797-KR-L 524581.77
1347 J0230-SKD-S 479937.14
4549 JNE3797-KR-M 454290.16
4550 JNE3797-KR-S 407302.57
4551 JNE3797-KR-XL 332155.24
1345 J0230-SKD-L 305616.95
4552 JNE3797-KR-XS 303616.70
6305 SET268-KR-NP-XL 284058.96
4554 JNE3797-KR-XXXL 276375.80,
ship-state Amount
28 MAHARASHTRA 13335534.14
23 KARNATAKA 10481114.37
57 TELANGANA 6916615.65
59 UTTAR PRADESH 6816642.08
56 TAMIL NADU 6515650.11
14 DELHI 4235215.97
24 KERALA 3830227.58
61 WEST BENGAL 3507880.44
1 ANDHRA PRADESH 3219831.72
19 HARYANA 2882092.99)

Insights from Analysis.

1. Top Performing Categories The leading categories by sales amount are:

Set: ₹39.2M;

Kurta: ₹21.3M;

Western Dress: ₹11.2M;

1. Best-Selling Products (Top 10 SKUs) Some standout SKUs include:

J0230-SKD-M: ₹527,699;

JNE3797-KR-L: ₹524,582;

1. Geographic Insights (Top 10 States) Key states contributing to sales:

Maharashtra: ₹13.3M;

Karnataka: ₹10.4M;

Telangana: ₹6.9M;

4) VISUALIZATION : STATISTICAL ANALYSIS (1)

In [14]:
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

# Prepare data for regression analysis: Analyze factors affecting 'Amount'
# Selecting numeric and relevant categorical data for encoding
regression_data = data[['Amount', 'Qty', 'ship-state', 'Category']].dropna()

# One-hot encoding for categorical variables
regression_data = pd.get_dummies(regression_data, columns=['ship-state', 'Category'], drop_first=True)

# Define independent variables (X) and dependent variable (y)
X = regression_data.drop('Amount', axis=1)
y = regression_data['Amount']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Apply linear regression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predict on test set
y_pred = regressor.predict(X_test)

# Evaluate model performance
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Feature importance (coefficients)
coefficients = pd.DataFrame({'Feature': X.columns, 'Importance': regressor.coef_}).sort_values(by='Importance', ascending=False)


print('Minimum Sum of errors :',mse)
print('Goodness of fit : ',r2)

coefficients.head(10)
```

```
Minimum Sum of errors : 1.2694288558234903e+26
Goodness of fit :  -1.611829444225318e+21
```

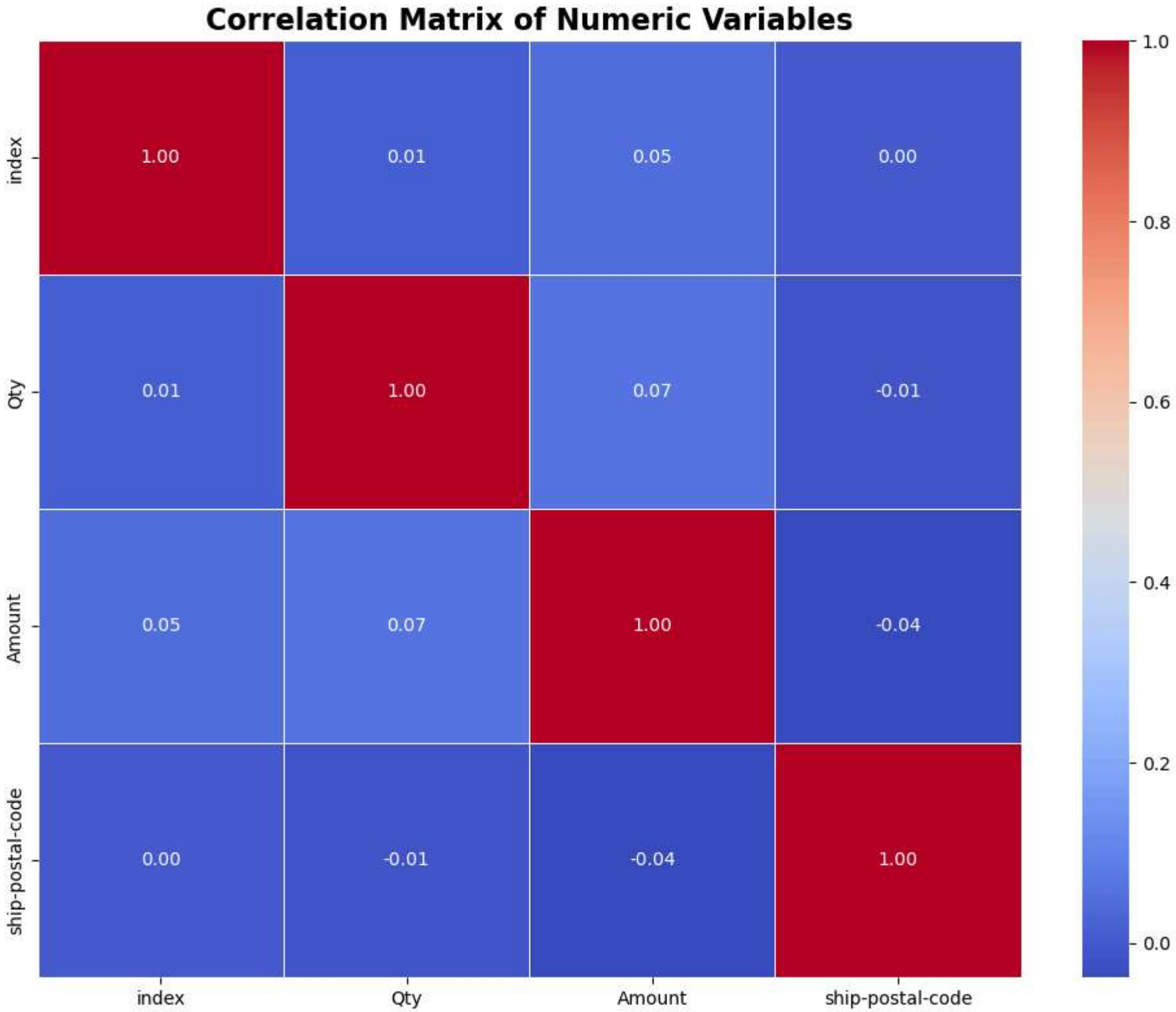
Out[14]:

	Feature	Importance
61	ship-state_bihar	596.066810
72	Category_Set	311.961884
71	Category_Saree	273.975721
11	ship-state_Chandigarh	262.344127
74	Category_Western Dress	242.762119
70	Category_Ethnic Dress	202.231916
54	ship-state_Sikkim	147.687758
24	ship-state_LADAKH	147.335790
47	ship-state_Punjab/Mohali/Zirakpur	109.256184
62	ship-state_delhi	105.896387

VISUALIZATION : STATISTICAL ANALYSIS (2)

In [15]:
```
# File path for the dataset
file_path = r'C:\Users\LENOVO\OneDrive\Desktop\Independent projects\archive\Amazon Sale Report.csv'

# Reload the data
data = pd.read_csv(file_path)

# Select numerical columns for correlation analysis
numeric_columns = data.select_dtypes(include=[np.number])

# Compute the correlation matrix
correlation_matrix = numeric_columns.corr()

# Plot the correlation matrix as a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Correlation Matrix of Numeric Variables', fontsize=16, fontweight='bold')
plt.tight_layout()
plt.show()
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_19048\1293156215.py:5: DtypeWarning: Columns (23) have mixed types. Specify dtype option on import or set low_memory=False.
  data = pd.read_csv(file_path)
```



Key Observations:

Weak correlations between all parameters choosen.