Phase 2, Part 1: Data Preparation

1. Import Pandas and Load the Data

In [11]:
```
import pandas as pd

# Load the core datasets
orders = pd.read_csv('olist_orders_dataset.csv')
items = pd.read_csv('olist_order_items_dataset.csv')
products = pd.read_csv('olist_products_dataset.csv')
customers = pd.read_csv('olist_customers_dataset.csv')
payments = pd.read_csv('olist_order_payments_dataset.csv')
translations = pd.read_csv('product_category_name_translation.csv')
```

2. Merge the DataFrames

In [12]:
```
# --- Merging Process ---

# Merge orders with order items (one order can have multiple items)
df = pd.merge(orders, items, on='order_id', how='left')

# Merge product information
df = pd.merge(df, products, on='product_id', how='left')

# Merge customer information
df = pd.merge(df, customers, on='customer_id', how='left')

# Merge payment information
df = pd.merge(df, payments, on='order_id', how='left')

# Merge the English category names
df = pd.merge(df, translations, on='product_category_name', how='left')
```

3. Initial Inspection

In [13]:
```
# --- Initial Inspection ---

print("DataFrame Info after all merges:")
df.info()

print("\nFirst 5 rows:")
print(df.head())
```

```
DataFrame Info after all merges:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 118434 entries, 0 to 118433
Data columns (total 31 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   order_id                       118434 non-null  object
 1   customer_id                    118434 non-null  object
 2   order_status                   118434 non-null  object
 3   order_purchase_timestamp       118434 non-null  object
 4   order_approved_at              118258 non-null  object
 5   order_delivered_carrier_date   116360 non-null  object
 6   order_delivered_customer_date  115037 non-null  object
 7   order_estimated_delivery_date  118434 non-null  object
 8   order_item_id                  117604 non-null  float64
 9   product_id                     117604 non-null  object
 10  seller_id                      117604 non-null  object
 11  shipping_limit_date            117604 non-null  object
 12  price                          117604 non-null  float64
 13  freight_value                  117604 non-null  float64
 14  product_category_name          115906 non-null  object
 15  product_name_lenght            115906 non-null  float64
 16  product_description_lenght     115906 non-null  float64
 17  product_photos_qty             115906 non-null  float64
 18  product_weight_g               117584 non-null  float64
 19  product_length_cm              117584 non-null  float64
 20  product_height_cm              117584 non-null  float64
 21  product_width_cm               117584 non-null  float64
 22  customer_unique_id             118434 non-null  object
 23  customer_zip_code_prefix       118434 non-null  int64
 24  customer_city                  118434 non-null  object
 25  customer_state                 118434 non-null  object
 26  payment_sequential             118431 non-null  float64
 27  payment_type                   118431 non-null  object
 28  payment_installments           118431 non-null  float64
 29  payment_value                  118431 non-null  float64
 30  product_category_name_english  115881 non-null  object
dtypes: float64(13), int64(1), object(17)
memory usage: 28.0+ MB

First 5 rows:
                           order_id                       customer_id  \
0  e481f51cbdc54678b7cc49136f2d6af7  9ef432eb6251297304e76186b10a928d
1  e481f51cbdc54678b7cc49136f2d6af7  9ef432eb6251297304e76186b10a928d
2  e481f51cbdc54678b7cc49136f2d6af7  9ef432eb6251297304e76186b10a928d
3  53cdb2fc8bc7dce0b6741e2150273451  b0830fb4747a6c6d20dea0b8c802d7ef
4  47770eb9100c2d0c44946d9cf07ec65d  41ce2a54c0b03bf3443c3d931a367089

  order_status order_purchase_timestamp    order_approved_at  \
0    delivered      2017-10-02 10:56:33  2017-10-02 11:07:15
1    delivered      2017-10-02 10:56:33  2017-10-02 11:07:15
2    delivered      2017-10-02 10:56:33  2017-10-02 11:07:15
3    delivered      2018-07-24 20:41:37  2018-07-26 03:24:27
4    delivered      2018-08-08 08:38:49  2018-08-08 08:55:23
```

```
  order_delivered_carrier_date order_delivered_customer_date  \
0          2017-10-04 19:55:00           2017-10-10 21:25:13
1          2017-10-04 19:55:00           2017-10-10 21:25:13
2          2017-10-04 19:55:00           2017-10-10 21:25:13
3          2018-07-26 14:31:00           2018-08-07 15:27:45
4          2018-08-08 13:50:00           2018-08-17 18:06:29

  order_estimated_delivery_date  order_item_id  \
0           2017-10-18 00:00:00            1.0
1           2017-10-18 00:00:00            1.0
2           2017-10-18 00:00:00            1.0
3           2018-08-13 00:00:00            1.0
4           2018-09-04 00:00:00            1.0

                         product_id  ... product_width_cm  \
0  87285b34884572647811a353c7ac498a  ...             13.0
1  87285b34884572647811a353c7ac498a  ...             13.0
2  87285b34884572647811a353c7ac498a  ...             13.0
3  595fac2a385ac33a80bd5114aec74eb8  ...             19.0
4  aa4383b373c6aca5d8797843e5594415  ...             21.0

                customer_unique_id  customer_zip_code_prefix  customer_city  \
0  7c396fd4830fd04220f754e42b4e5bff                      3149      sao paulo
1  7c396fd4830fd04220f754e42b4e5bff                      3149      sao paulo
2  7c396fd4830fd04220f754e42b4e5bff                      3149      sao paulo
3  af07308b275d755c9edb36a90c618231                     47813      barreiras
4  3a653a41f6f9fc3d2a113cf8398680e8                     75265     vianopolis

  customer_state  payment_sequential  payment_type  payment_installments  \
0             SP                 1.0   credit_card                   1.0
1             SP                 3.0       voucher                   1.0
2             SP                 2.0       voucher                   1.0
3             BA                 1.0        boleto                   1.0
4             GO                 1.0   credit_card                   3.0

   payment_value  product_category_name_english
0          18.12                     housewares
1           2.00                     housewares
2          18.59                     housewares
3         141.46                      perfumery
4         179.12                           auto

[5 rows x 31 columns]
```

Phase 2, Part 2: Data Cleaning

1.Convert to Datetime

In [4]:
```
# Convert all timestamp columns to datetime objects
```

```
date_columns = ['order_purchase_timestamp', 'order_approved_at',
                'order_delivered_carrier_date', 'order_delivered_customer_date
                'order_estimated_delivery_date']

for col in date_columns:
    df[col] = pd.to_datetime(df[col])
```

2.Handle Missing Values

In [7]:
```
# Remove rows where product_id is null and re-assign the DataFrame
df = df.dropna(subset=['product_id'])

# Fill missing English product category names and re-assign the column
df['product_category_name_english'] = df['product_category_name_english'].fill

print("Missing values handled successfully using the corrected method.")
```

```
Missing values handled successfully using the corrected method.
```

3. Feature Engineering: Create New Time-Based Columns

In [9]:
```
# Extract year, month, and day of week for trend analysis
df['order_purchase_year'] = df['order_purchase_timestamp'].dt.year
df['order_purchase_month'] = df['order_purchase_timestamp'].dt.month
df['order_purchase_dayofweek'] = df['order_purchase_timestamp'].dt.dayofweek #
```

4. Export the Cleaned Data

In [10]:
```
# Save the final, cleaned dataset
df.to_csv('olist_cleaned_master_data.csv', index=False)

print("\nCleaning complete! 'olist_cleaned_master_data.csv' is ready for Power
print("\nFinal data info:")
df.info()
```

```
Cleaning complete! 'olist_cleaned_master_data.csv' is ready for Power BI.

Final data info:
<class 'pandas.core.frame.DataFrame'>
Index: 117604 entries, 0 to 118433
Data columns (total 34 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   order_id                       117604 non-null  object
 1   customer_id                    117604 non-null  object
 2   order_status                   117604 non-null  object
 3   order_purchase_timestamp       117604 non-null  datetime64[ns]
 4   order_approved_at              117589 non-null  datetime64[ns]
 5   order_delivered_carrier_date   116359 non-null  datetime64[ns]
 6   order_delivered_customer_date  115037 non-null  datetime64[ns]
 7   order_estimated_delivery_date  117604 non-null  datetime64[ns]
 8   order_item_id                  117604 non-null  float64
 9   product_id                     117604 non-null  object
 10  seller_id                      117604 non-null  object
 11  shipping_limit_date            117604 non-null  object
 12  price                          117604 non-null  float64
 13  freight_value                  117604 non-null  float64
 14  product_category_name          115906 non-null  object
 15  product_name_lenght            115906 non-null  float64
 16  product_description_lenght     115906 non-null  float64
 17  product_photos_qty             115906 non-null  float64
 18  product_weight_g               117584 non-null  float64
 19  product_length_cm              117584 non-null  float64
 20  product_height_cm              117584 non-null  float64
 21  product_width_cm               117584 non-null  float64
 22  customer_unique_id             117604 non-null  object
 23  customer_zip_code_prefix       117604 non-null  int64
 24  customer_city                  117604 non-null  object
 25  customer_state                 117604 non-null  object
 26  payment_sequential             117601 non-null  float64
 27  payment_type                   117601 non-null  object
 28  payment_installments           117601 non-null  float64
 29  payment_value                  117601 non-null  float64
 30  product_category_name_english  117604 non-null  object
 31  order_purchase_year            117604 non-null  int32
 32  order_purchase_month           117604 non-null  int32
 33  order_purchase_dayofweek       117604 non-null  int32
dtypes: datetime64[ns](5), float64(13), int32(3), int64(1), object(12)
memory usage: 30.1+ MB
```

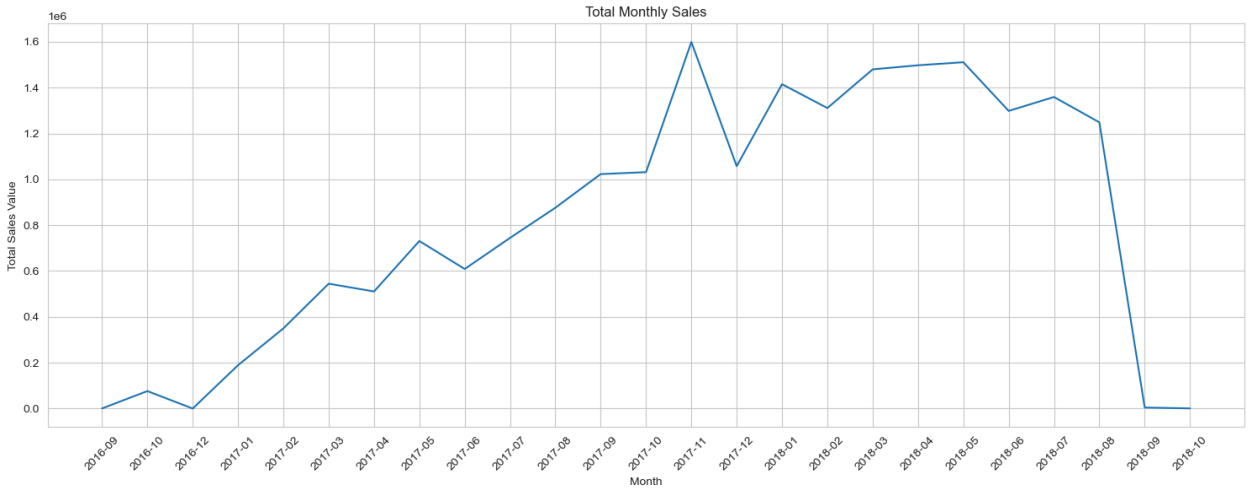
Phase 3: Exploratory Data Analysis (EDA)

1. Imports for Plotting

In [14]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Set a style for our plots
sns.set_style('whitegrid')
```

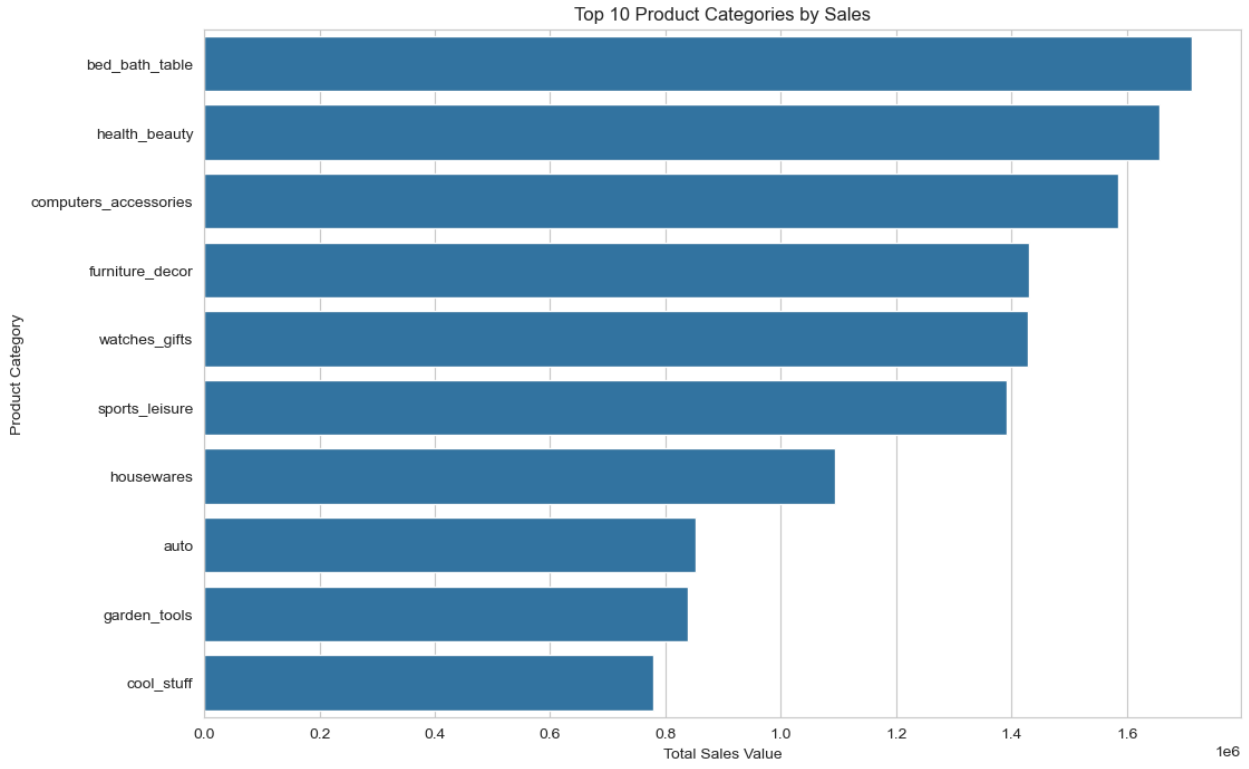
2. Question 1: What are the monthly sales trends?

In [16]:
```
# First, ensure order_purchase_timestamp is a datetime type
df['order_purchase_timestamp'] = pd.to_datetime(df['order_purchase_timestamp']

# Group data by month and sum the payment value
df['order_purchase_year_month'] = df['order_purchase_timestamp'].dt.to_period(
monthly_sales = df.groupby('order_purchase_year_month')['payment_value'].sum()

# Convert Period to timestamp for plotting
monthly_sales['order_purchase_year_month'] = monthly_sales['order_purchase_yea

# Plot the monthly sales
plt.figure(figsize=(15, 6))
sns.lineplot(x='order_purchase_year_month', y='payment_value', data=monthly_sa
plt.title('Total Monthly Sales')
plt.xlabel('Month')
plt.ylabel('Total Sales Value')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.tight_layout()  # Adjust layout to make room for rotated labels
plt.show()
```



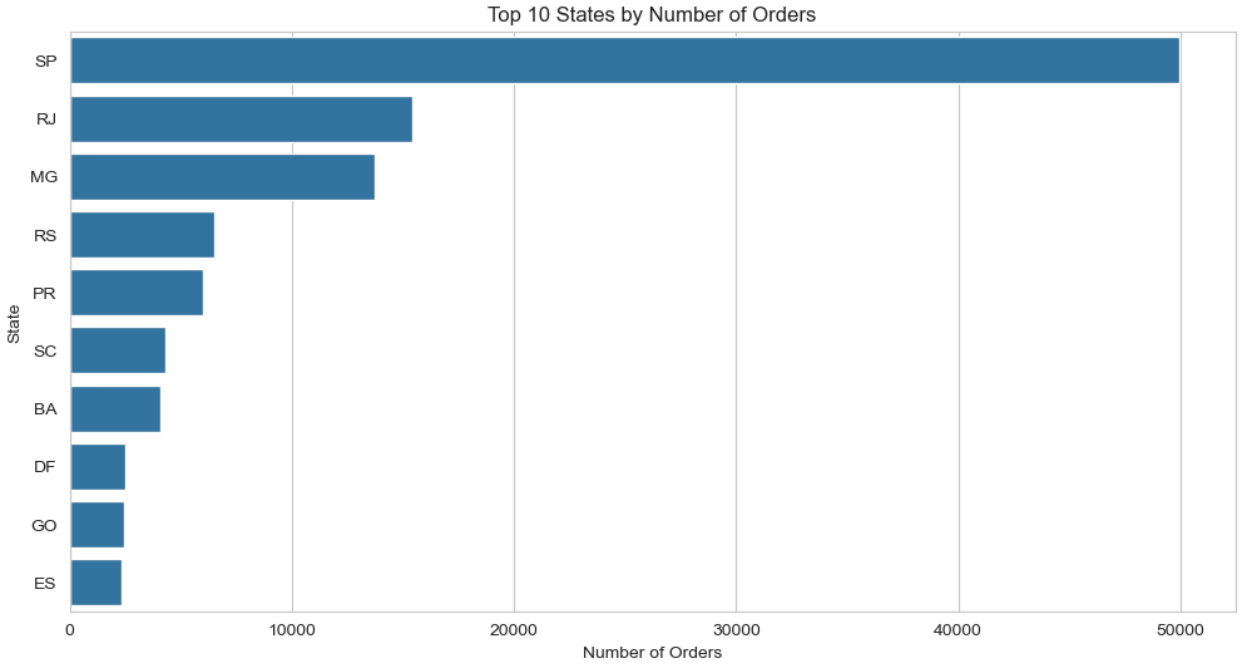
3. Question 2: Which are the top 10 product categories by sales?

In [17]:
```
# Group by product category and sum payment value
top_categories = df.groupby('product_category_name_english')['payment_value'].

# Plot the top 10 categories
plt.figure(figsize=(12, 8))
sns.barplot(x='payment_value', y='product_category_name_english', data=top_cat
plt.title('Top 10 Product Categories by Sales')
plt.xlabel('Total Sales Value')
plt.ylabel('Product Category')
plt.show()
```



4. Question 3: Which are the top 10 states by number of orders?

In [18]:
```
# Group by customer state and count the orders
top_states = df['customer_state'].value_counts().nlargest(10).reset_index()
top_states.columns = ['state', 'number_of_orders']

# Plot the top 10 states
plt.figure(figsize=(12, 6))
sns.barplot(x='number_of_orders', y='state', data=top_states, orient='h')
plt.title('Top 10 States by Number of Orders')
plt.xlabel('Number of Orders')
plt.ylabel('State')
plt.show()
```



In [ ]: