Importing Required Libraries

In [37]:

```
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
import pandas as pd
import plotly.express as px
import pycountry
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

In [38]:

```
pip install pycountry
```

```
Requirement already satisfied: pycountry in /Users/harshitha/anaconda3/lib/python3.11/site-packages (2
3.12.11)
Note: you may need to restart the kernel to use updated packages.
```

Reading Dataset

In [39]:

```
df = pd.read_csv("/Users/harshitha/Desktop/Datasets/Supply_Chain_Dataset.csv", encoding='latin-1')
```

In [40]:

```
df.head()
```

Out[40]:

	Type	Days for shipping (real)	Days for shipment (scheduled)	Benefit per order	Sales per customer	Delivery Status	Late_delivery_risk	Category Id	Category Name	Customer City	...	Order Zipcode	Pro Ca
0	DEBIT	3	4	91.250000	314.640015	Advance shipping	0	73	Sporting Goods	Caguas	...	NaN	
1	TRANSFER	5	4	-249.089996	311.359985	Late delivery	1	73	Sporting Goods	Caguas	...	NaN	
2	CASH	4	4	-247.779999	309.720001	Shipping on time	0	73	Sporting Goods	San Jose	...	NaN	
3	DEBIT	3	4	22.860001	304.809998	Advance shipping	0	73	Sporting Goods	Los Angeles	...	NaN	
4	PAYMENT	2	4	134.210007	298.250000	Advance shipping	0	73	Sporting Goods	Caguas	...	NaN	

5 rows × 53 columns

In [41]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180519 entries, 0 to 180518
Data columns (total 53 columns):
 #   Column                         Non-Null Count   Dtype  
---  ------                         --------------   -----  
 0   Type                           180519 non-null  object 
 1   Days for shipping (real)       180519 non-null  int64  
 2   Days for shipment (scheduled)  180519 non-null  int64  
 3   Benefit per order              180519 non-null  float64
 4   Sales per customer             180519 non-null  float64
 5   Delivery Status                180519 non-null  object 
 6   Late_delivery_risk             180519 non-null  int64  
 7   Category Id                    180519 non-null  int64  
 8   Category Name                  180519 non-null  object 
 9   Customer City                  180519 non-null  object 
 10  Customer Country               180519 non-null  object 
 11  Customer Email                 180519 non-null  object 
 12  Customer Fname                 180519 non-null  object 
 13  Customer Id                    180519 non-null  int64  
 14  Customer Lname                 180511 non-null  object 
 15  Customer Password              180519 non-null  object 
 16  Customer Segment               180519 non-null  object 
 17  Customer State                 180519 non-null  object 
 18  Customer Street                180519 non-null  object 
 19  Customer Zipcode               180516 non-null  float64
 20  Department Id                  180519 non-null  int64  
 21  Department Name                180519 non-null  object 
 22  Latitude                       180519 non-null  float64
 23  Longitude                      180519 non-null  float64
 24  Market                         180519 non-null  object 
 25  Order City                     180519 non-null  object 
 26  Order Country                  180519 non-null  object 
 27  Order Customer Id              180519 non-null  int64  
 28  order date (DateOrders)        180519 non-null  object 
 29  Order Id                       180519 non-null  int64  
 30  Order Item Cardprod Id         180519 non-null  int64  
 31  Order Item Discount            180519 non-null  float64
 32  Order Item Discount Rate       180519 non-null  float64
 33  Order Item Id                  180519 non-null  int64  
 34  Order Item Product Price       180519 non-null  float64
 35  Order Item Profit Ratio        180519 non-null  float64
 36  Order Item Quantity            180519 non-null  int64  
 37  Sales                          180519 non-null  float64
 38  Order Item Total               180519 non-null  float64
 39  Order Profit Per Order         180519 non-null  float64
 40  Order Region                   180519 non-null  object 
 41  Order State                    180519 non-null  object 
 42  Order Status                   180519 non-null  object 
 43  Order Zipcode                  24840 non-null   float64
 44  Product Card Id                180519 non-null  int64  
 45  Product Category Id            180519 non-null  int64  
 46  Product Description            0 non-null       float64
 47  Product Image                  180519 non-null  object 
 48  Product Name                   180519 non-null  object 
 49  Product Price                  180519 non-null  float64
 50  Product Status                 180519 non-null  int64  
 51  shipping date (DateOrders)     180519 non-null  object 
 52  Shipping Mode                  180519 non-null  object 
dtypes: float64(15), int64(14), object(24)
memory usage: 73.0+ MB
```

Data Cleaning

In [42]:
```
df.isnull().sum()
```

Out[42]:
```
Type                                  0
Days for shipping (real)              0
Days for shipment (scheduled)         0
Benefit per order                     0
Sales per customer                    0
Delivery Status                       0
Late_delivery_risk                    0
Category Id                           0
Category Name                         0
Customer City                         0
Customer Country                      0
Customer Email                        0
Customer Fname                        0
Customer Id                           0
Customer Lname                        8
Customer Password                     0
Customer Segment                      0
Customer State                        0
Customer Street                       0
Customer Zipcode                      3
Department Id                         0
Department Name                       0
Latitude                              0
Longitude                             0
Market                                0
Order City                            0
Order Country                         0
Order Customer Id                     0
order date (DateOrders)               0
Order Id                              0
Order Item Cardprod Id                0
Order Item Discount                   0
Order Item Discount Rate              0
Order Item Id                         0
Order Item Product Price              0
Order Item Profit Ratio               0
Order Item Quantity                   0
Sales                                 0
Order Item Total                      0
Order Profit Per Order                0
Order Region                          0
Order State                           0
Order Status                          0
Order Zipcode                    155679
Product Card Id                       0
Product Category Id                   0
Product Description              180519
Product Image                         0
Product Name                          0
Product Price                         0
Product Status                        0
shipping date (DateOrders)            0
Shipping Mode                         0
dtype: int64
```

In [43]:
```
df.drop(columns=['Customer Email', 'Product Image', 'Customer Password', 'Order Zipcode', 'Product Desc
```

In [44]:
```
df.drop(columns=['Order Profit Per Order'], inplace=True)
```

In [45]:
```
df.isnull().sum()
```

Out[45]:
```
Type                             0
Days for shipping (real)         0
Days for shipment (scheduled)    0
Benefit per order                0
Sales per customer               0
Delivery Status                  0
Late_delivery_risk               0
Category Id                      0
Category Name                    0
Customer City                    0
Customer Country                 0
Customer Fname                   0
Customer Id                      0
Customer Lname                   8
Customer Segment                 0
Customer State                   0
Customer Street                  0
Customer Zipcode                 3
Department Id                    0
Department Name                  0
Latitude                         0
Longitude                        0
Market                           0
Order City                       0
Order Country                    0
Order Customer Id                0
order date (DateOrders)          0
Order Id                         0
Order Item Cardprod Id           0
Order Item Discount              0
Order Item Discount Rate         0
Order Item Id                    0
Order Item Product Price         0
Order Item Profit Ratio          0
Order Item Quantity              0
Sales                            0
Order Item Total                 0
Order Region                     0
Order State                      0
Order Status                     0
Product Card Id                  0
Product Category Id              0
Product Name                     0
Product Price                    0
Product Status                   0
shipping date (DateOrders)       0
Shipping Mode                    0
dtype: int64
```

In [46]:
```
df['Customer Zipcode'].fillna('Unknown',inplace=True)
df['Customer Lname'].fillna('Unknown',inplace=True)
```

In [47]:
```
df.isnull().sum()
```

Out[47]:
```
Type                             0
Days for shipping (real)         0
Days for shipment (scheduled)    0
Benefit per order                0
Sales per customer               0
Delivery Status                  0
Late_delivery_risk               0
Category Id                      0
Category Name                    0
Customer City                    0
Customer Country                 0
Customer Fname                   0
Customer Id                      0
Customer Lname                   0
Customer Segment                 0
Customer State                   0
Customer Street                  0
Customer Zipcode                 0
Department Id                    0
Department Name                  0
Latitude                         0
Longitude                        0
Market                           0
Order City                       0
Order Country                    0
Order Customer Id                0
order date (DateOrders)          0
Order Id                         0
Order Item Cardprod Id           0
Order Item Discount              0
Order Item Discount Rate         0
Order Item Id                    0
Order Item Product Price         0
Order Item Profit Ratio          0
Order Item Quantity              0
Sales                            0
Order Item Total                 0
Order Region                     0
Order State                      0
Order Status                     0
Product Card Id                  0
Product Category Id              0
Product Name                     0
Product Price                    0
Product Status                   0
shipping date (DateOrders)       0
Shipping Mode                    0
dtype: int64
```

Converted to DateTime format

In [48]:
```
date_columns = ['order date (DateOrders)', 'shipping date (DateOrders)']
for col in date_columns:
    df[col] = pd.to_datetime(df[col])

print(df.dtypes)
```

```
Type                                          object
Days for shipping (real)                       int64
Days for shipment (scheduled)                  int64
Benefit per order                            float64
Sales per customer                           float64
Delivery Status                               object
Late_delivery_risk                             int64
Category Id                                    int64
Category Name                                 object
Customer City                                 object
Customer Country                              object
Customer Fname                                object
Customer Id                                    int64
Customer Lname                                object
Customer Segment                              object
Customer State                                object
Customer Street                               object
Customer Zipcode                              object
Department Id                                  int64
Department Name                               object
Latitude                                     float64
Longitude                                    float64
Market                                        object
Order City                                    object
Order Country                                 object
Order Customer Id                              int64
order date (DateOrders)               datetime64[ns]
Order Id                                       int64
Order Item Cardprod Id                         int64
Order Item Discount                          float64
Order Item Discount Rate                     float64
Order Item Id                                  int64
Order Item Product Price                     float64
Order Item Profit Ratio                      float64
Order Item Quantity                            int64
Sales                                        float64
Order Item Total                             float64
Order Region                                  object
Order State                                   object
Order Status                                  object
Product Card Id                                int64
Product Category Id                            int64
Product Name                                  object
Product Price                                float64
Product Status                                 int64
shipping date (DateOrders)            datetime64[ns]
Shipping Mode                                 object
dtype: object
```

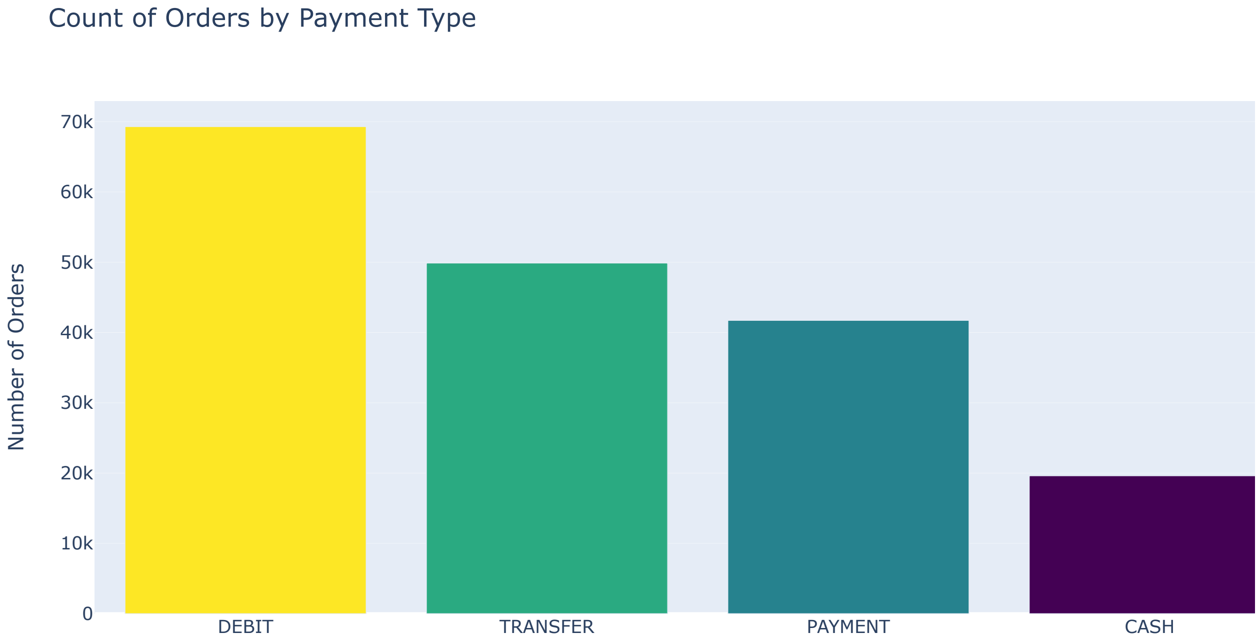
In [61]:
```
# Ensuring it is in dateTime format and extracting "Order Year" column

df['order date (DateOrders)'] = pd.to_datetime(df['order date (DateOrders)'])
df['Order Year'] = df['order date (DateOrders)'].dt.year
df['year_from_date'] = df['shipping date (DateOrders)'].dt.year.astype(int)
```

Data Visualization

1. Payment Type Analysis

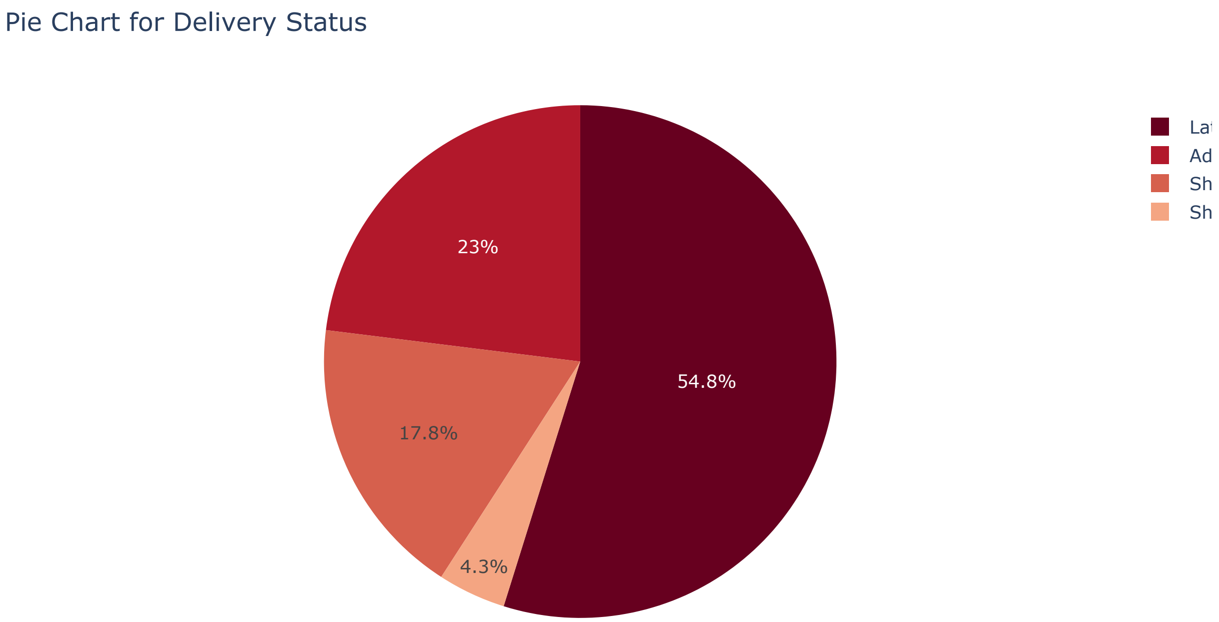
In [50]:

```
type_counts = df['Type'].value_counts().reset_index()
type_counts.columns = ['Payment Type', 'Count']

fig = px.bar(type_counts, x='Payment Type', y='Count',
             title='Count of Orders by Payment Type',
             color='Count',
             color_continuous_scale=px.colors.sequential.Viridis)

fig.update_layout(
    xaxis_title="Payment Type",
    yaxis_title="Number of Orders",
    coloraxis_showscale=False
)
fig.show()
```



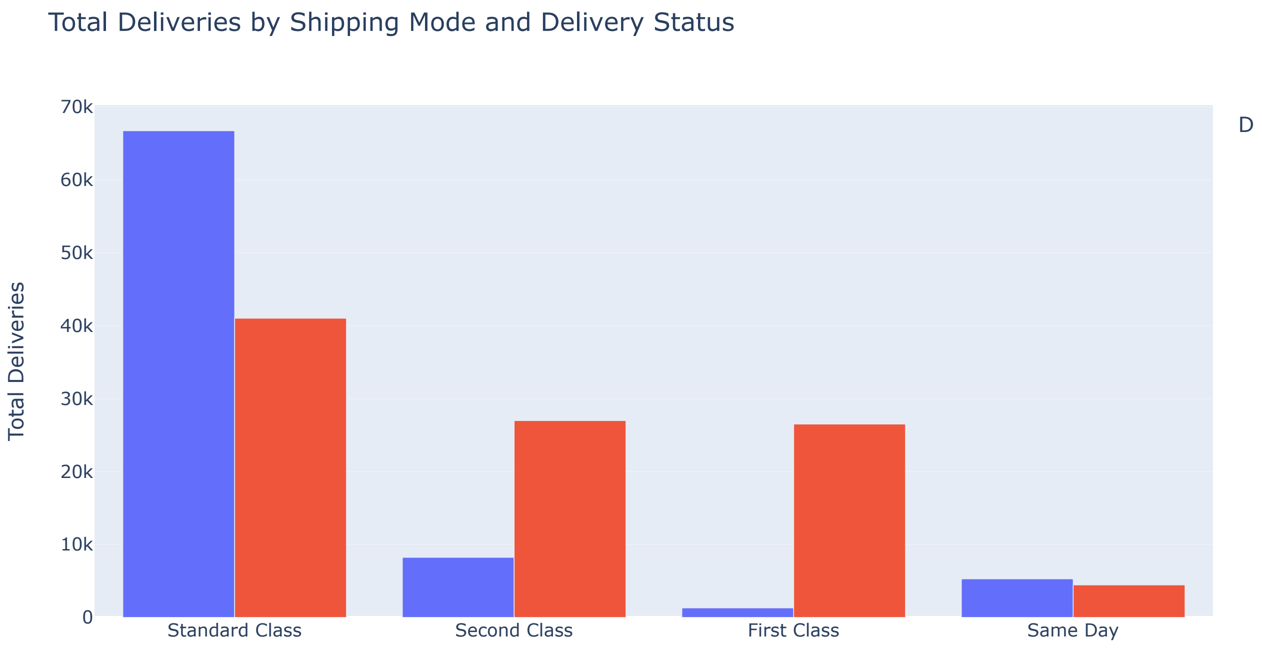
2. Delivery Status Analysis

In [51]:
```
import pandas as pd
import plotly.express as px

# Counting the occurrences of each delivery status for the pie chart
delivery_status_counts = df['Delivery Status'].value_counts().reset_index()
delivery_status_counts.columns = ['Delivery Status', 'Count']
pie_chart_fig = px.pie(delivery_status_counts, values='Count', names='Delivery Status',
                       title='Pie Chart for Delivery Status', color_discrete_sequence=px.colors.sequent

pie_chart_fig.show()
```


Pie Chart for Delivery Status

3. Shipping Mode Analysis

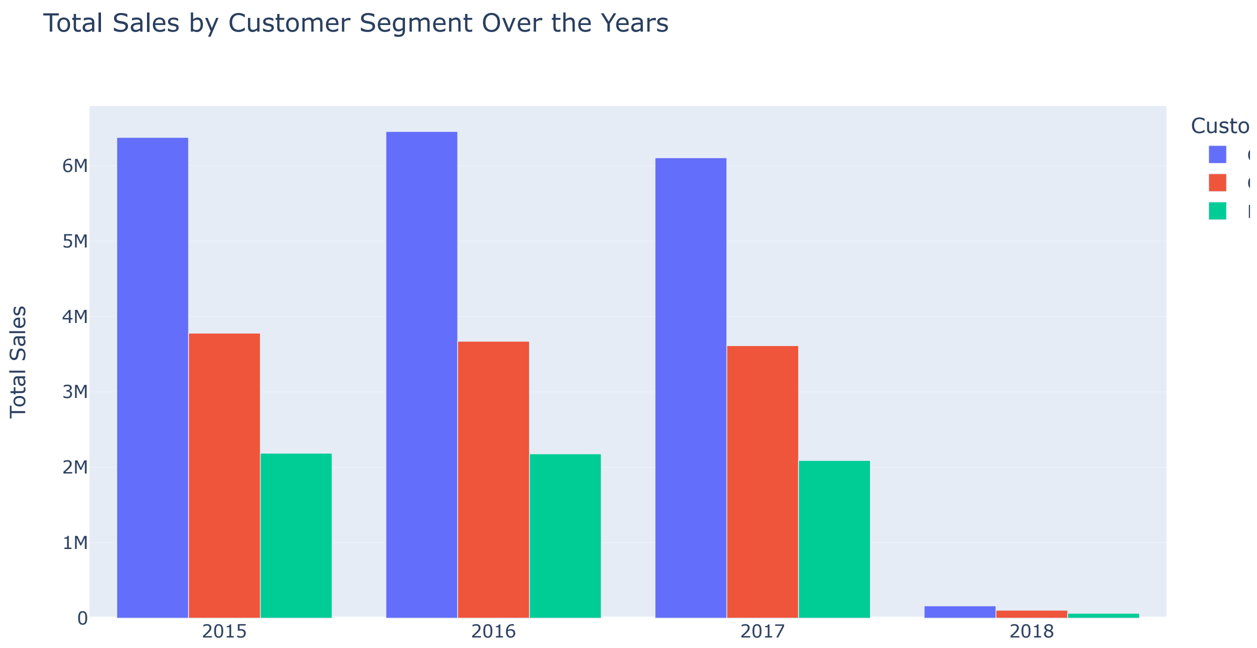
In [52]:
```
df['Delivery Status'] = df['Late_delivery_risk'].apply(lambda x: 'On Time' if x == 1 else 'Late')
delivery_counts = df.groupby(['Shipping Mode', 'Delivery Status']).size().reset_index(name='Count')

fig = px.bar(delivery_counts, x='Shipping Mode', y='Count', color='Delivery Status',
             barmode='group', title='Total Deliveries by Shipping Mode and Delivery Status')

fig.update_layout(xaxis={'categoryorder':'total descending'},
                  yaxis_title='Total Deliveries',
                  xaxis_title='Shipping Mode')
fig.show()
```



4. Sales by Customer Segment over the years

In [53]:

```
segment_yearly_sales = df.groupby(['Order Year', 'Customer Segment'])['Sales'].sum().reset_index()
fig = px.bar(segment_yearly_sales, x='Order Year', y='Sales', color='Customer Segment',
             title='Total Sales by Customer Segment Over the Years')

fig.update_layout(
    xaxis_title="Order Year",
    yaxis_title="Total Sales",
    barmode='group'
)

fig.show()
```

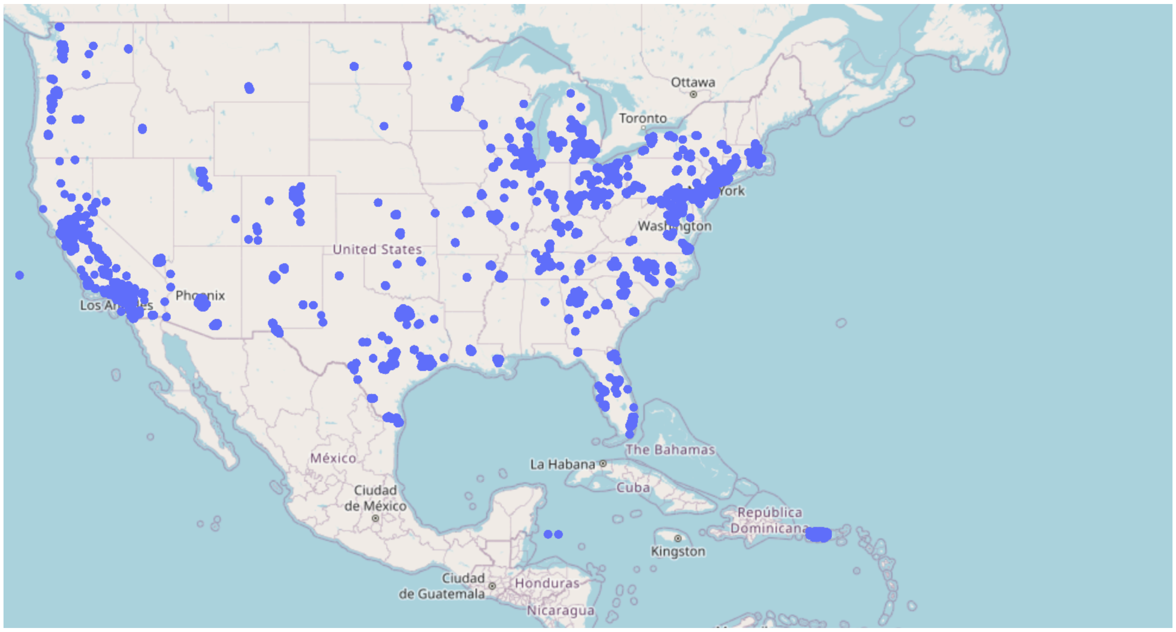


Total Sales by Customer Segment Over the Years

5. Geographic Distribution of Orders - Scatter Mapbox

In [54]:
```
fig = px.scatter_mapbox(df,
                        lat='Latitude',
                        lon='Longitude',
                        hover_name='Category Name',
                        zoom=3,
                        height=600,
                        title='Geographic Distribution of Orders')

fig.update_layout(mapbox_style='open-street-map')
fig.show()
```

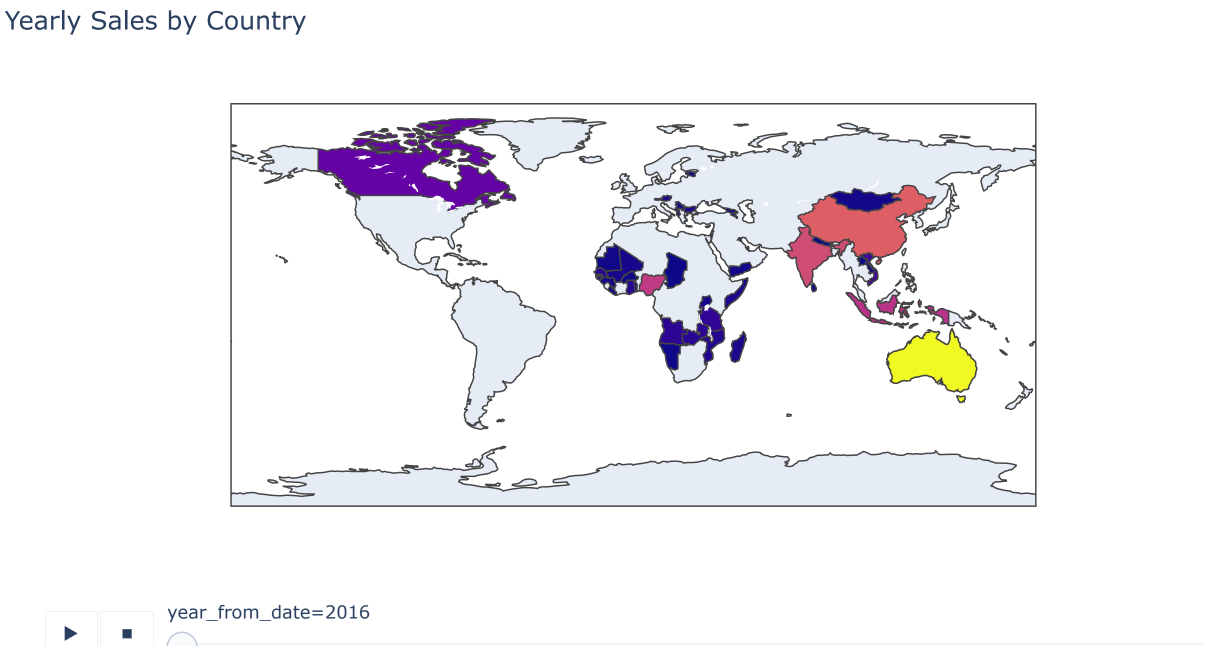
Geographic Distribution of Orders



6. Yearly Sales by Country - Choropleth Map

In [62]:

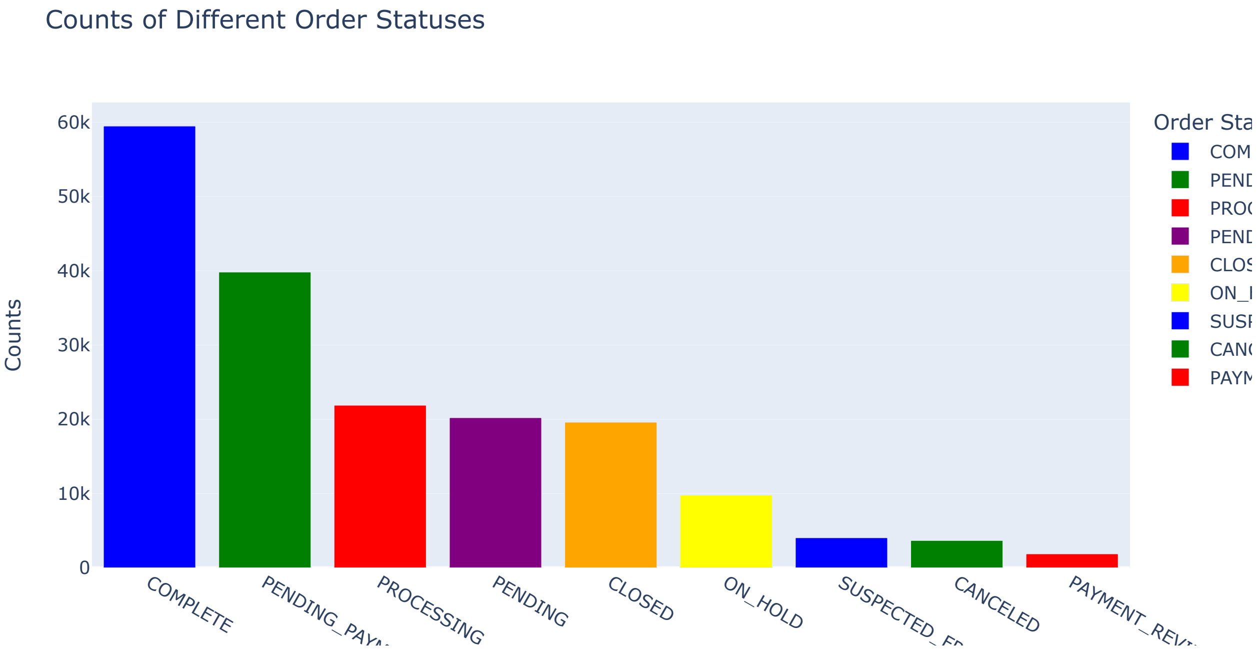
```
country_yearly_sales = df.groupby(['Order Country', 'year_from_date'])['Sales'].sum().reset_index()

# function to convert country names to ISO Alpha-3 codes
def convert_to_iso_alpha(country_name):
    try:
        return pycountry.countries.lookup(country_name).alpha_3
    except LookupError:
        return None

# Applying the function
country_yearly_sales['iso_alpha'] = country_yearly_sales['Order Country'].apply(convert_to_iso_alpha)

# Filtering out any rows that did not get an ISO code
country_yearly_sales = country_yearly_sales.dropna(subset=['iso_alpha'])

fig = px.choropleth(country_yearly_sales,
                    locations='iso_alpha',
                    color='Sales',
                    hover_name='Order Country',
                    animation_frame='year_from_date',
                    color_continuous_scale=px.colors.sequential.Plasma,
                    title='Yearly Sales by Country')

fig.layout.updatemenus[0].buttons[0].args[1]['frame']['duration'] = 3000
fig.layout.updatemenus[0].buttons[0].args[1]['transition']['duration'] = 500
fig.show()
```



7. Order Status Count

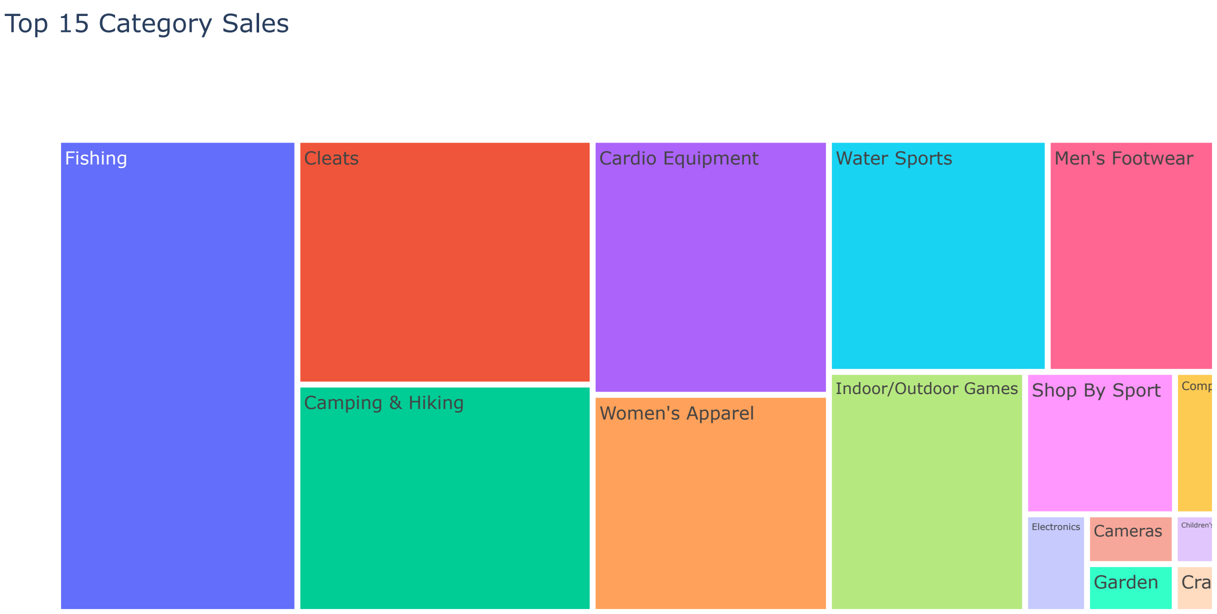
In [56]:

```
order_status_counts = df['Order Status'].value_counts().reset_index()
order_status_counts.columns = ['Order Status', 'Counts']
colors = ['blue', 'green', 'red', 'purple', 'orange', 'yellow']
bar_chart_fig = px.bar(order_status_counts, x='Order Status', y='Counts',
                       title='Counts of Different Order Statuses',
                       color='Order Status',
                       color_discrete_sequence=colors)
bar_chart_fig.show()
```



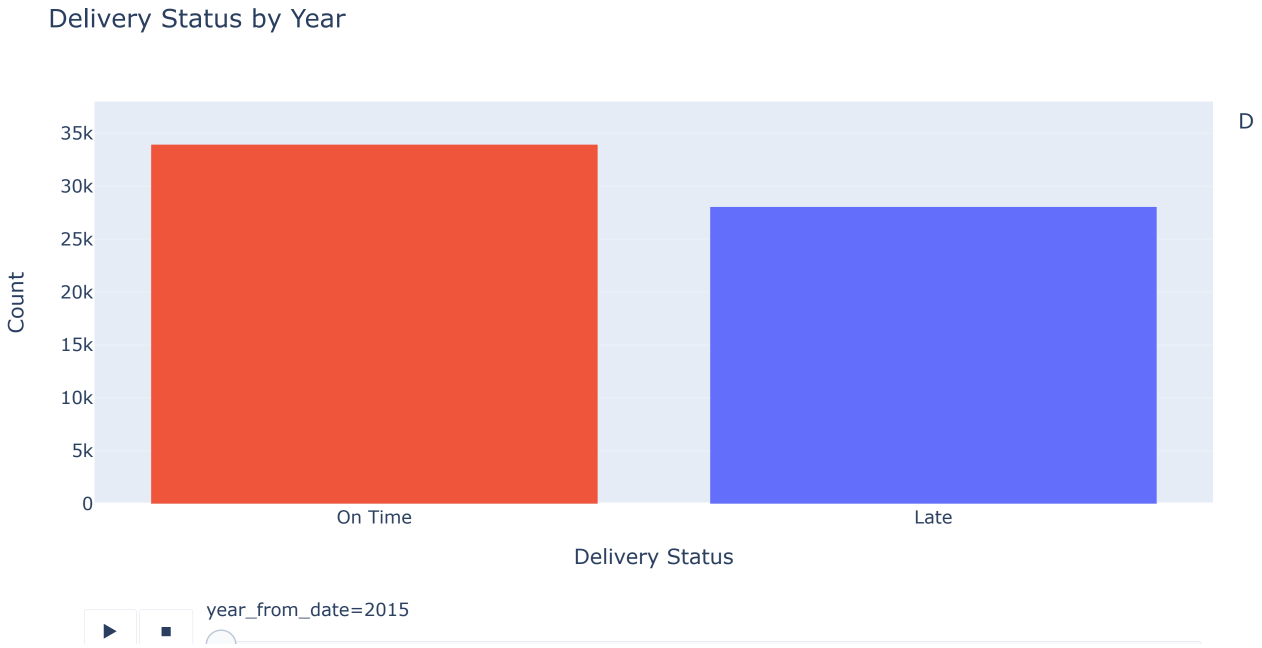
8. Sales for Top 15 Category - Treemap

In [57]:
```
category_sales = df.groupby('Category Name')['Sales'].sum().nlargest(15)
category_sales = category_sales.reset_index()

fig = px.treemap(category_sales, path=['Category Name'], values='Sales',
                 hover_data={'Category Name': True, 'Sales': ':.2f'},
                 title='Top 15 Category Sales')
fig.show()
```



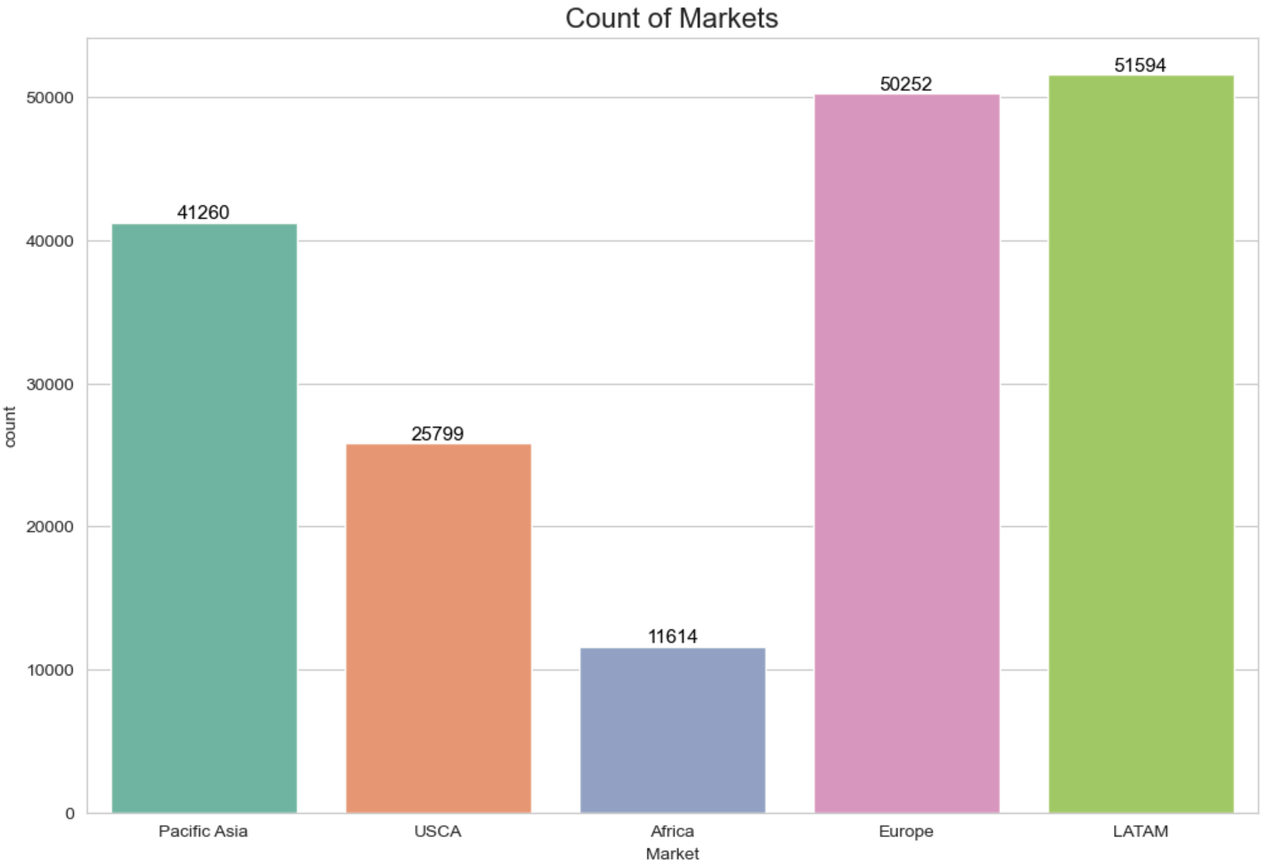
9. Delivery Status by Year

In [58]:
```
df['month_name'] = df['shipping date (DateOrders)'].dt.strftime('%B')
df_long = df.groupby(['year_from_date', 'Delivery Status']).size().reset_index(name='Count')

fig = px.bar(df_long, x='Delivery Status', y='Count',
             color='Delivery Status',
             animation_frame='year_from_date',
             animation_group='Delivery Status',
             range_y=[0, df_long['Count'].max()*1.1],
             title='Delivery Status by Year')

fig.update_layout(xaxis={'categoryorder':'total descending'},
                  yaxis_title='Count',
                  xaxis_title='Delivery Status')

fig.show()
```



10. Market Popularity

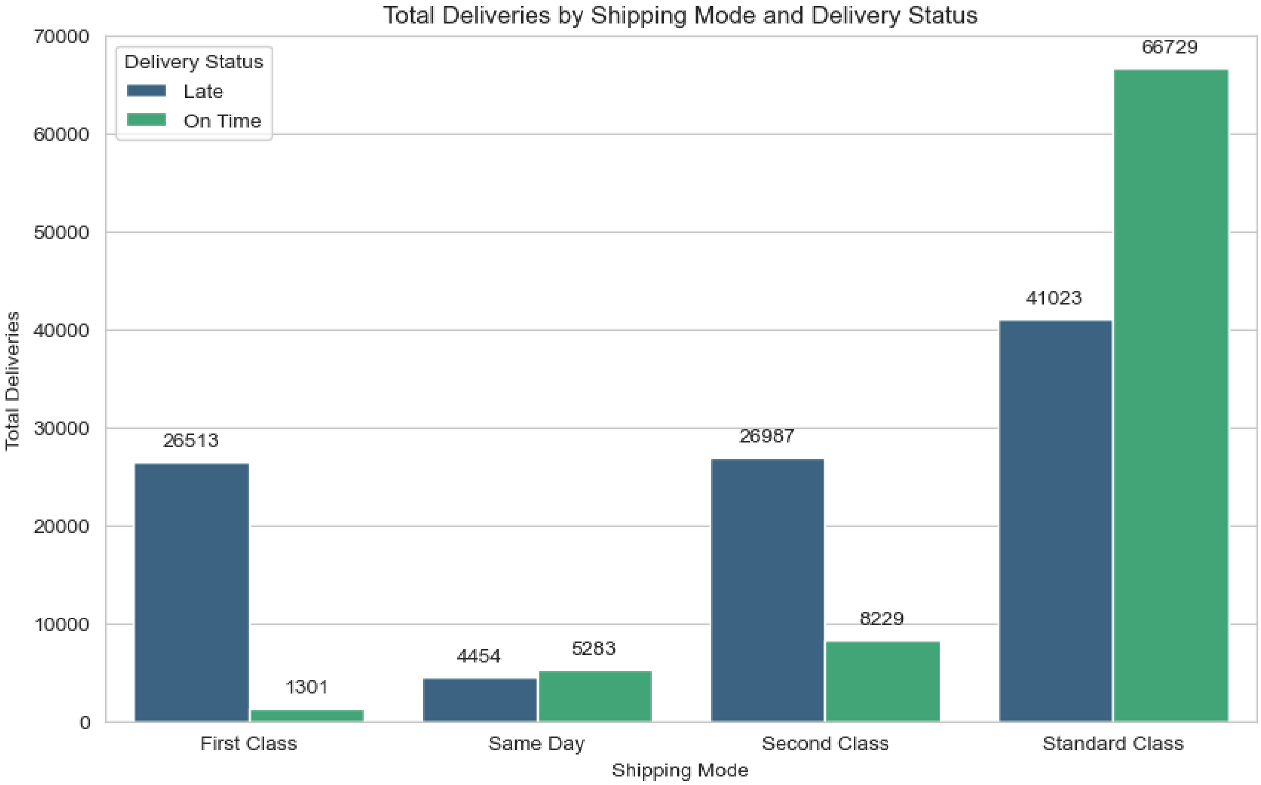
In [59]:
```
sns.set_style('whitegrid')
plt.figure(figsize=(12, 8))
ax = sns.countplot(data=df, x='Market', palette='Set2')
ax.set_xticklabels(ax.get_xticklabels(), rotation=0)

for p in ax.patches:
    ax.annotate(f'{int(p.get_height())}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', fontsize=11, color='black', xytext=(0, 5),
                textcoords='offset points')

plt.title('Count of Markets', fontsize=16)
plt.show()
```



11. Total Deliveries by Shipping Mode and Delivery Status

In [60]:

```
df['Delivery Status'] = df['Late_delivery_risk'].map({1: 'Late', 0: 'On Time'})
delivery_counts = df.groupby(['Shipping Mode', 'Delivery Status']).size().reset_index(name='Count')

plt.figure(figsize=(10, 6))
sns.barplot(data=delivery_counts, x='Shipping Mode', y='Count', hue='Delivery Status', palette='viridis
plt.title('Total Deliveries by Shipping Mode and Delivery Status')
plt.xlabel('Shipping Mode')
plt.ylabel('Total Deliveries')

for p in plt.gca().patches:
    plt.gca().annotate(format(p.get_height(), '.0f'),
                       (p.get_x() + p.get_width() / 2., p.get_height()),
                       ha = 'center',
                       va = 'center',
                       xytext = (0, 10),
                       textcoords = 'offset points')

plt.show()
```



In [ ]: