E-Commerce Sales Analysis

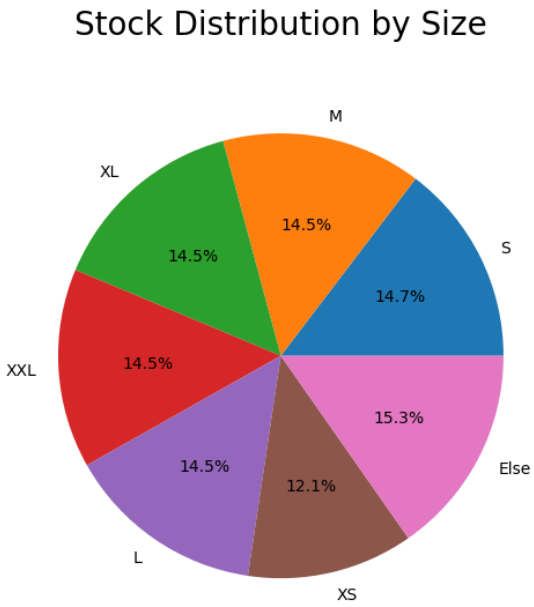
In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

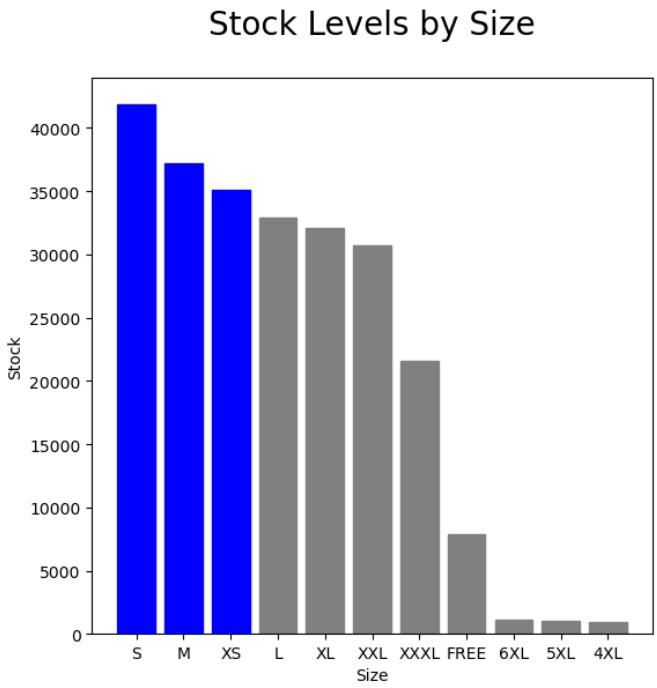
Load Data

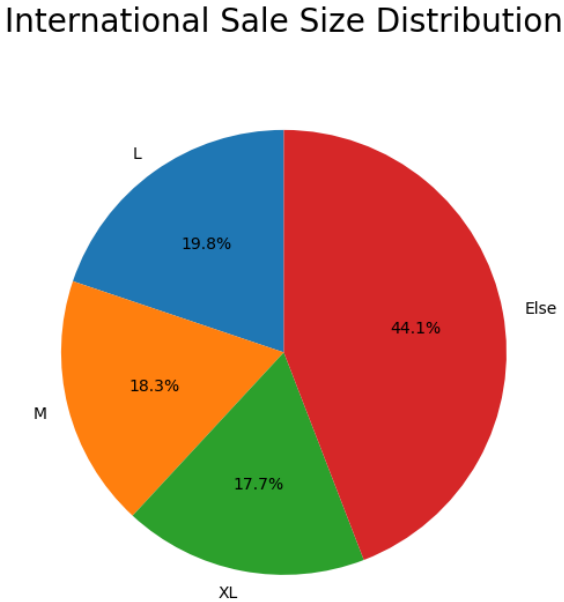
In [2]:
```
# Load the datasets
df1 = pd.read_csv('./dataset/sales/Sale Report.csv')
df2 = pd.read_csv('./dataset/sales/International sale Report.csv')
df3 = pd.read_csv('./dataset/sales/Amazon Sale Report.csv',low_memory=False)
df4 = df1
```

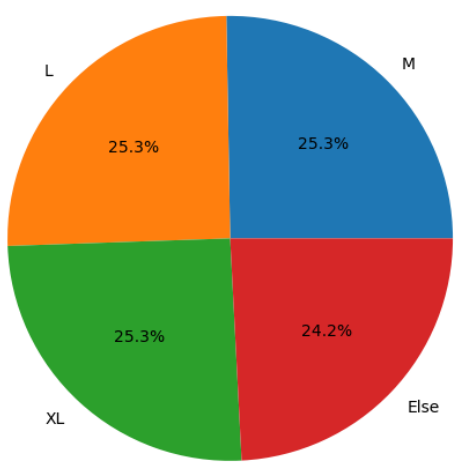
In [6]:
```
# Set up the plot grid
fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(15, 15))

# Plot the first chart
sizes = df1['Size'].value_counts()
sizes['Else'] = sizes[['XXXL', '4XL', '5XL', '6XL', 'FREE']].sum()
sizes = sizes.drop(labels=['XXXL', '4XL', '5XL', '6XL', 'FREE'])
total_stock = sizes.sum()
sizes_percentage = sizes / total_stock * 100
axs[0, 0].pie(sizes, labels=sizes.index, autopct='%1.1f%%')
axs[0, 0].set_title('Stock Distribution by Size',fontsize=20, x=0.5, y=1.05, pad=10)

# Load the data from the CSV file
data = pd.read_csv('./dataset/sales/Sale Report.csv')

# Map sizes to aggregated sizes
size_map = {'S': 'S', 'M': 'M', 'L': 'L', 'XL': 'XL', 'XXL': '2XL', 'XXXL': '3XL', '4XL': '4XL'
data['AggregatedSize'] = data['Size'].map(size_map)

# Calculate the sum of stock levels for each aggregated size
#data_by_aggregated_size = data.groupby('AggregatedSize').sum().reset_index()
data_by_aggregated_size = data.groupby('Size').sum().reset_index()

# Sort the data by stock levels in descending order
#data_by_aggregated_size = data_by_aggregated_size.sort_values('Stock', ascending=False)
data_by_aggregated_size = data_by_aggregated_size.sort_values('Stock', ascending=False)

# Highlight the top 3 aggregated sizes
data_by_aggregated_size['Highlighted'] = False
data_by_aggregated_size.loc[0:2, 'Highlighted'] = True

# Create the bar plot
bars = axs[0,1].bar(data_by_aggregated_size['Size'], data_by_aggregated_size['Stock'])
for i in range(len(bars)):
    if i < 3:
        bars[i].set_color('blue')
    else:
        bars[i].set_color('gray')
axs[0,1].set(xlabel='Size', ylabel='Stock')
axs[0,1].set_title('Stock Levels by Size', fontsize=20, x=0.5, y=1.05, pad=10)


# Plot the second chart
valid_sizes = ['XS', 'S', 'M', 'L', 'XL', 'XXL', 'XXXL', '4XL', '5XL', '6XL', 'FREE']
df2 = df2[df2['Size'].isin(valid_sizes)]
df2['PCS'] = pd.to_numeric(df2['PCS'], errors='coerce')
size_counts = df2.groupby('Size').size().reset_index(name='count')
size_counts = size_counts.sort_values(by='count', ascending=False)
top_sizes = size_counts.head(3)
other_sizes_count = size_counts.iloc[3:]['count'].sum()
other_sizes = pd.DataFrame({'Size': ['Else'], 'count': [other_sizes_count]})
top_sizes = pd.concat([top_sizes, other_sizes], ignore_index=True)
total_sales = df2['PCS'].sum()
top_sizes['percentage'] = top_sizes['count'] / total_sales * 100
axs[1, 0].pie(top_sizes['percentage'], labels=top_sizes['Size'], autopct='%1.1f%%', startangle=
axs[1, 0].set_title('International Sale Size Distribution',fontsize=20, x=0.5, y=1.05, pad=10)
```

```
# Plot the third chart
sizes = df3['Size']
allowed_sizes = ['XS', 'S', 'M', 'L', 'XL', 'XXL', '3XL', '4XL', '5XL', '6XL', 'FREE']
sizes = sizes[sizes.isin(allowed_sizes)]
size_counts = sizes.value_counts()
total_sales = df3['Qty'].sum()
top_sizes = size_counts[:3].index.tolist()
top_sizes_count = size_counts[top_sizes].sum()
top_sizes_percentage = top_sizes_count / total_sales * 100
else_count = size_counts.drop(top_sizes).sum()
else_percentage = else_count / total_sales * 100
labels = top_sizes + ['Else']
values = [top_sizes_percentage] * len(top_sizes) + [else_percentage]
axs[1, 1].pie(values, labels=labels, autopct='%1.1f%%')
axs[1, 1].set_title('Amazon Sale Size Distribution',fontsize=20, x=0.5, y=1.05, pad=10)

# Remove the fourth subplot
#fig.delaxes(axs[0, 1])

# Adjust the spacing between subplots
plt.subplots_adjust(hspace=0.4, wspace=0.4)

# Show the plot
plt.show()
#fig.savefig('/input/Final Size Analysis.png', dpi=300)
```





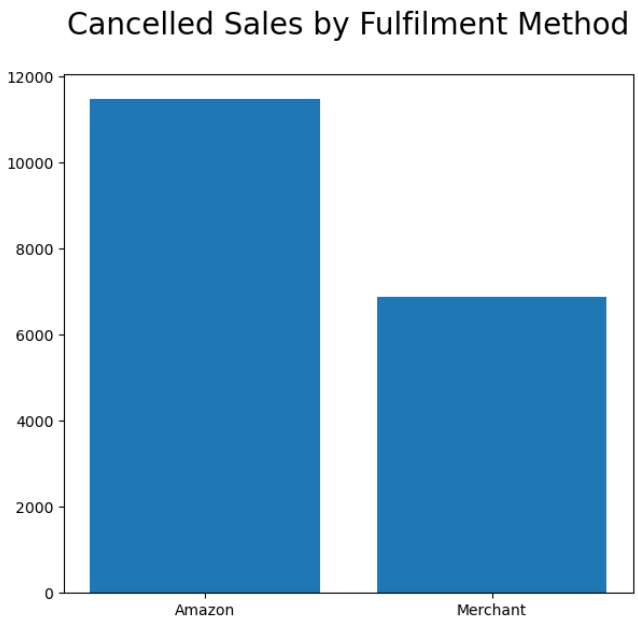


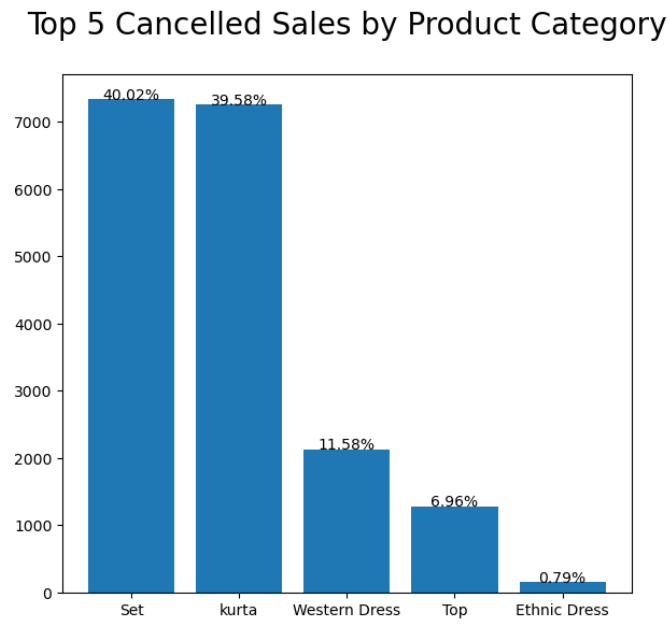


In [5]:

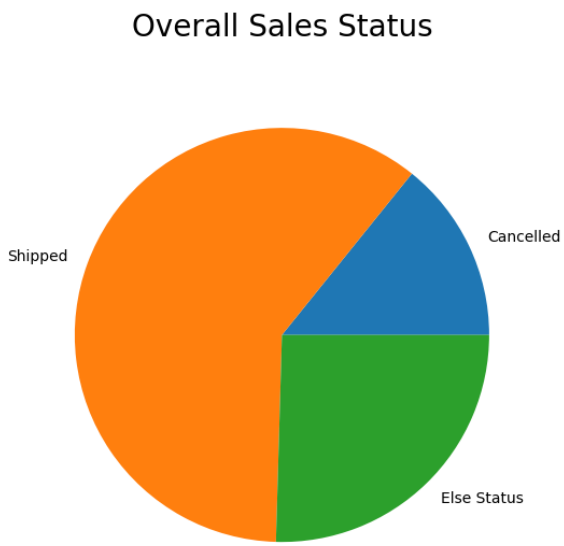
```
import pandas as pd
import matplotlib.pyplot as plt

# read the csv file into a pandas dataframe
df = pd.read_csv('./dataset/sales/Amazon Sale Report.csv')

# filter the dataframe to only include shipped and cancelled sales
filtered_df= df[df['Status'].isin(['Shipped', 'Cancelled'])]

# group the filtered dataframe by status and calculate the count
by_status = filtered_df.groupby('Status').count()['SKU']

# create a new column called 'Grouped Status' to group all other status types into 'Else Status
df['Grouped Status'] = df['Status'].apply(lambda x: x if x in ['Shipped', 'Cancelled'] else 'El

# group the original dataframe by the new 'Grouped Status' column and calculate the count
by_grouped_status = df.groupby('Grouped Status').count()['SKU']

# create a figure with 2 rows and 2 columns
fig, axs = plt.subplots(2, 2, figsize=(15, 15))

# create a bar chart to show the cancelled sales by fulfilment method
cancelled_df = filtered_df[filtered_df['Status'] == 'Cancelled']
by_fulfilment = cancelled_df.groupby('Fulfilment').count()['SKU']
axs[0, 0].bar(by_fulfilment.index, by_fulfilment.values)
axs[0, 0].set_title('Cancelled Sales by Fulfilment Method',fontsize=20, x=0.5, y=1.05, pad=10)

# group cancelled sales by product category and calculate the count
by_category = cancelled_df.groupby('Category').count()['SKU']

# sort the categories by the count of cancelled sales
by_category = by_category.sort_values(ascending=False)

# keep only the top 5 categories
top_category = by_category[:5]

# calculate the percentage of cancelled sales for each top category
category_percentages = (top_category / cancelled_df.shape[0]) * 100

# create a bar chart to show the cancelled sales by top 5 product categories
axs[0, 1].bar(top_category.index, top_category.values)
axs[0, 1].set_title('Top 5 Cancelled Sales by Product Category',fontsize=20, x=0.5, y=1.05, pad

# add the percentage labels to the bar chart
for i, v in enumerate(top_category.values):
    axs[0, 1].text(i, v+5, f'{category_percentages.values[i]:.2f}%', ha='center')



# create a bar chart to show the cancelled sales by shipping service level
by_shipping = cancelled_df.groupby('ship-service-level').count()['SKU']
axs[1, 0].bar(by_shipping.index, by_shipping.values)
axs[1, 0].set_title('Cancelled Sales by Shipping Service Level',fontsize=20, x=0.5, y=1.05, pad

# create a pie chart to show the overall status of sales
by_status = filtered_df.groupby('Status').count()['SKU']
by_status['Else Status'] = df.groupby('Grouped Status').count()['SKU']['Else Status']
axs[1, 1].pie(by_status.values, labels=by_status.index)
axs[1, 1].set_title('Overall Sales Status',fontsize=20, x=0.5, y=1.05, pad=10)

# adjust the spacing between the subplots
plt.subplots_adjust(hspace=0.4)

# show the combined chart
plt.show()
#fig.savefig('/input/Cancelled Sale Analysis.png', dpi=300)
```

```
/tmp/ipykernel_49427/582662589.py:5: DtypeWarning: Columns (23) have mixed types. Specify dtype
option on import or set low_memory=False.
  df = pd.read_csv('./dataset/sales/Amazon Sale Report.csv')
```


Cancelled Sales by Fulfilment Method
12000
10000
8000
6000
4000
2000
0
Amazon
Merchant


Top 5 Cancelled Sales by Product Category
40.02%
39.58%
11.58%
6.96%
0.79%
7000
6000
5000
4000
3000
2000
1000
0
Set
kurta
Western Dress
Top
Ethnic Dress


Cancelled Sales by Shipping Service Level
10000
8000
6000
4000
2000
0
Expedited
Standard


Overall Sales Status
Shipped
Cancelled
Else Status

In [ ]: