Project: E-commerce Store Sales Analysis

Introduction:

A retail store has provided us with their sales data for the past year. We are going to analyze the data to extract valuable insights that can help the store improve their sales and overall performance.

Data Description:

The data consists of sales transactions recorded at the store, containing the following fields:

- Order ID
- Order Date
- Ship Date
- Ship Mode
- Customer ID
- Country
- City
- State
- Postal Code
- Region
- Product ID
- Sales
- Quantity
- Discount
- Profit

Research Questions:

We will attempt to answer the following research questions:

- What is the overall sales trend for the year?
- What is the average sale size?
- Does the month of the year affect sales?
- Is there a correlation between profit and quantity sold?

In [104…

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [105…

```
# Load data
df = pd.read_excel('store_sales.xlsx')
```

In [106…

```
df.head()
```

Out[106]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Country	City	State	Postal Code	Region	Product ID	Sales	Quantity	Discount
0	1	CA-2016-152156	2016-08-11 00:00:00	2016-11-11 00:00:00	Second Class	CG-12520	United States	Henderson	Kentucky	42420	South	FUR-BO-10001798	261.9600	2	0.00
1	2	CA-2016-152156	2016-08-11 00:00:00	2016-11-11 00:00:00	Second Class	CG-12520	United States	Henderson	Kentucky	42420	South	FUR-CH-10000454	731.9400	3	0.00
2	3	CA-2016-138688	2016-12-06 00:00:00	6/16/2016	Second Class	DV-13045	United States	Los Angeles	California	90036	West	OFF-LA-10000240	14.6200	2	0.00
3	4	US-2015-108966	2015-11-10 00:00:00	10/18/2015	Standard Class	SO-20335	United States	Fort Lauderdale	Florida	33311	South	FUR-TA-10000577	957.5775	5	0.45
4	5	US-2015-108966	2015-11-10 00:00:00	10/18/2015	Standard Class	SO-20335	United States	Fort Lauderdale	Florida	33311	South	OFF-ST-10000760	22.3680	2	0.20

Exploratory Data Analysis

First, let's load the data and perform some exploratory data analysis (EDA) to get a better understanding of our data.

We check for any missing values and data types. We also get some basic statistics of our numerical data. From the output, we can see that there are no missing values in the data. All columns have the expected data types.

In [107…

```
# Check for missing values
df.isnull().sum()
```

Out[107]:

```
Row ID          0
Order ID        0
Order Date      0
Ship Date       0
Ship Mode       0
Customer ID     0
Country         0
City            0
State           0
Postal Code     0
Region          0
Product ID      0
Sales           0
Quantity        0
Discount        0
Profit          0
dtype: int64
```

In [108…

```
# Check data types and basic statistics
df.dtypes
```

Out[108]:

```
Row ID            int64
Order ID         object
Order Date       object
Ship Date        object
Ship Mode        object
Customer ID      object
Country          object
City             object
State            object
Postal Code       int64
Region           object
Product ID       object
Sales           float64
Quantity          int64
Discount        float64
Profit          float64
dtype: object
```

In [109…

```
df.describe()
```

Out[109]:

	Row ID	Postal Code	Sales	Quantity	Discount	Profit
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	4997.500000	55190.379428	229.858001	3.789574	0.156203	28.656896
std	2885.163629	32063.693350	623.245101	2.225110	0.206452	234.260108
min	1.000000	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	2499.250000	23223.000000	17.280000	2.000000	0.000000	1.728750
50%	4997.500000	56430.500000	54.490000	3.000000	0.200000	8.666500
75%	7495.750000	90008.000000	209.940000	5.000000	0.200000	29.364000
max	9994.000000	99301.000000	22638.480000	14.000000	0.800000	8399.976000

Descriptive Statistics:

Next, let's look at some descriptive statistics of our data.

In [110…

```
# Total Sales statistics
print('Sales Statistics')
print(f"Sales: ${df['Sales'].sum():,.2f}")
```

```
Sales Statistics
Sales: $2,297,200.86
```

In [111…

```
print(f"Sale: ${df['Sales'].mean():,.2f}")
```

```
Sale: $229.86
```

In [112…

```
print(f"Median Sale: ${df['Sales'].median():,.2f}")
```

```
Median Sale: $54.49
```

In [113…

```
print(f"Min Sale: ${df['Sales'].min():,.2f}")
```

```
Min Sale: $0.44
```

In [114…

```
print(f"Max Sale: ${df['Sales'].max():,.2f}")
```

```
Max Sale: $22,638.48
```

In [115…

```
print(f"Sales Variance: ${df['Sales'].var():,.2f}")
```

```
Sales Variance: $388,434.46
```

In [116…

```
print(f"Sales Std Dev: ${df['Sales'].std():,.2f}")
```

```
Sales Std Dev: $623.25
```

In [117…

```
print(f"Sales Skewness: {df['Sales'].skew():,.2f}")
```

```
Sales Skewness: 12.97
```

In [118…

```
print(f"Sales Kurtosis: {df['Sales'].kurt():,.2f}")
```

```
Sales Kurtosis: 305.31
```

We get some statistics for our Sales column, such as the sum, mean, median, minimum, maximum, variance, standard deviation, skewness, and kurtosis.

Hypotheses:

Next, we will formulate hypotheses based on our research questions and test them.

Hypothesis 1:

- Null hypothesis: There is no significant trend in sales throughout the year.
- Alternative hypothesis: There is a significant trend in sales throughout the year.

We can test this hypothesis by plotting the sales by month.

In [119…

```
# Plot monthly sales
df['Order Date'] = pd.to_datetime(df['Order Date'])
```

In [120…

```
df['Order Date']
```

Out[120]:

```
0       2016-08-11
1       2016-08-11
2       2016-12-06
3       2015-11-10
4       2015-11-10
           ...
9989    2014-01-21
9990    2017-02-26
9991    2017-02-26
9992    2017-02-26
9993    2017-04-05
Name: Order Date, Length: 9994, dtype: datetime64[ns]
```

In [121…

```
df['Month'] = df['Order Date'].dt.month
```

In [122…

```
df['Month']
```

Out[122]:

```
0        8
1        8
2       12
3       11
4       11
        ..
9989     1
9990     2
9991     2
9992     2
9993     4
Name: Month, Length: 9994, dtype: int64
```

In [123…

```
monthly_sales = df.groupby('Month')['Sales'].sum()
```

In [124…

```
monthly_sales
```

Out[124]:

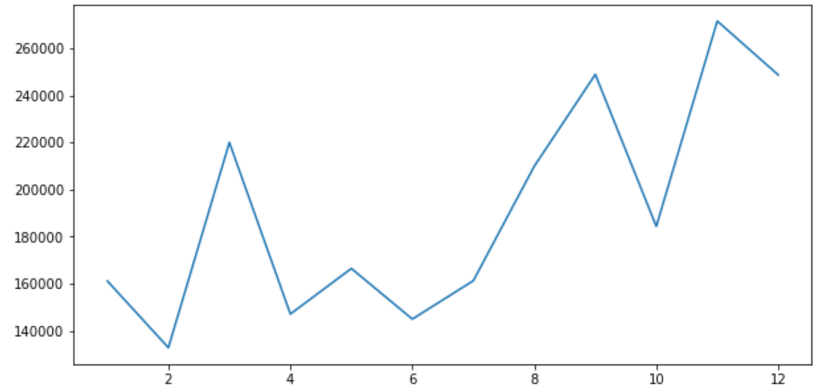
```
Month
1     161083.5874
2     132721.3594
3     220064.6460
4     147031.2641
5     166420.3167
6     144883.4973
7     161227.1045
8     209964.3679
9     248989.3031
10    184356.3342
11    271693.7525
12    248765.3272
Name: Sales, dtype: float64
```

In [125…

```
plt.figure(figsize=(10, 5))
plt.plot(monthly_sales)
```

Out[125]:

```
[<matplotlib.lines.Line2D at 0x7fb50069bb80>]
```



In [126…

```
# Compute the correlations between the variables
corr_matrix = df[['Sales', 'Quantity', 'Discount', 'Profit']].corr()
```

In [127…

```
corr_matrix
```

Out[127]:

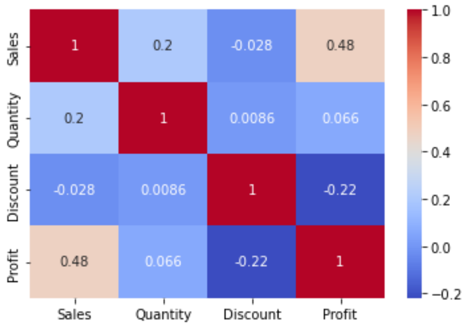
	Sales	Quantity	Discount	Profit
Sales	1.000000	0.200795	-0.028190	0.479064
Quantity	0.200795	1.000000	0.008623	0.066253
Discount	-0.028190	0.008623	1.000000	-0.219487
Profit	0.479064	0.066253	-0.219487	1.000000

In [128…

```
# Create a heatmap of the correlations
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
```

Out[128]:

```
<AxesSubplot:>
```



Based on the correlation matrix provided:

There is a strong positive correlation (0.48) between Sales and Profit. This indicates that as Sales increase, so does Profit. This is a desirable relationship for businesses as it means that they can increase their profits by increasing their sales.

There is a weak positive correlation (0.20) between Sales and Quantity. This indicates that there is some relationship between the two variables, but it is not very strong. This could be due to various factors such as pricing, seasonality, etc.

There is a weak positive correlation (0.07) between Quantity and Profit. This indicates that there is some relationship between the two variables, but it is not very strong. This could be due to various factors such as the cost of goods sold, pricing, etc.

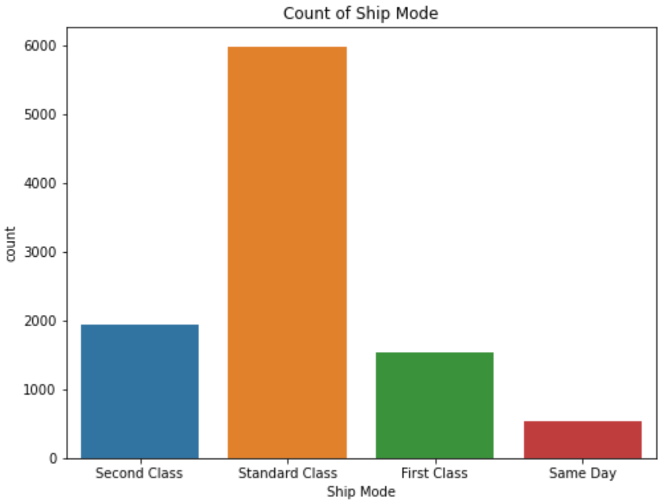
There is a weak negative correlation (-0.02) between Sales and Discount. This indicates that there is some relationship between the two variables, but it is not very strong. This could be due to various factors such as pricing strategy, promotions, etc.

There is a weak negative correlation (-0.22) between Discount and Profit. This indicates that as Discount increases, Profit decreases. This is an undesirable relationship for businesses as it means that they are sacrificing their profits in order to make sales.

Overall, the correlation matrix suggests that Sales is the most important variable in determining Profit. However, the relationships between the variables are not very strong, which suggests that there are other factors that could be influencing Profit as well. Further analysis and modeling would be needed to identify these factors and to develop a more accurate model for predicting Profit based on Sales, Quantity, and Discount.

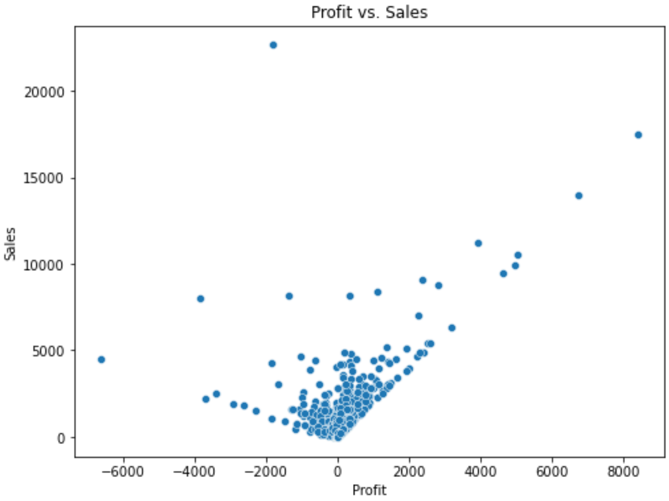
In [129…

```
# bar chart for Ship Mode
plt.figure(figsize=(8, 6))
sns.countplot(x='Ship Mode', data=df)
plt.title('Count of Ship Mode')
plt.show()
```



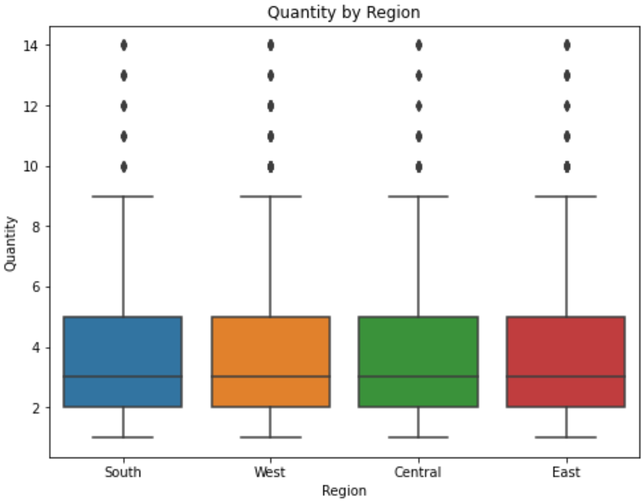
In [130…

```
# scatter plot for Profit vs. Sales
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Profit', y='Sales', data=df)
plt.title('Profit vs. Sales')
plt.show()
```



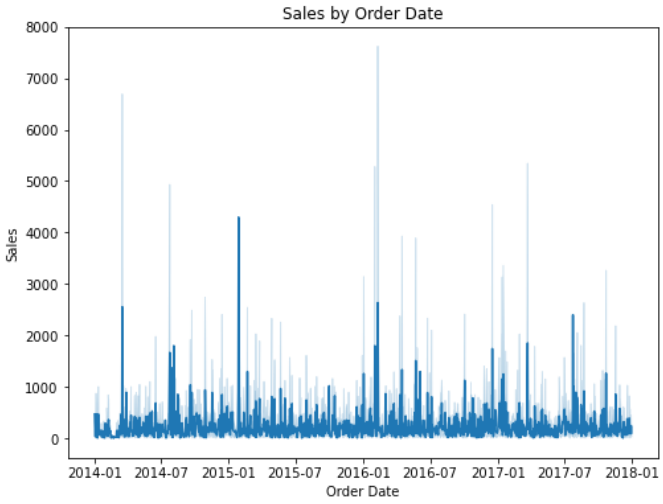
In [131…

```
# box plot for Quantity by Region
plt.figure(figsize=(8, 6))
sns.boxplot(x='Region', y='Quantity', data=df)
plt.title('Quantity by Region')
plt.show()
```



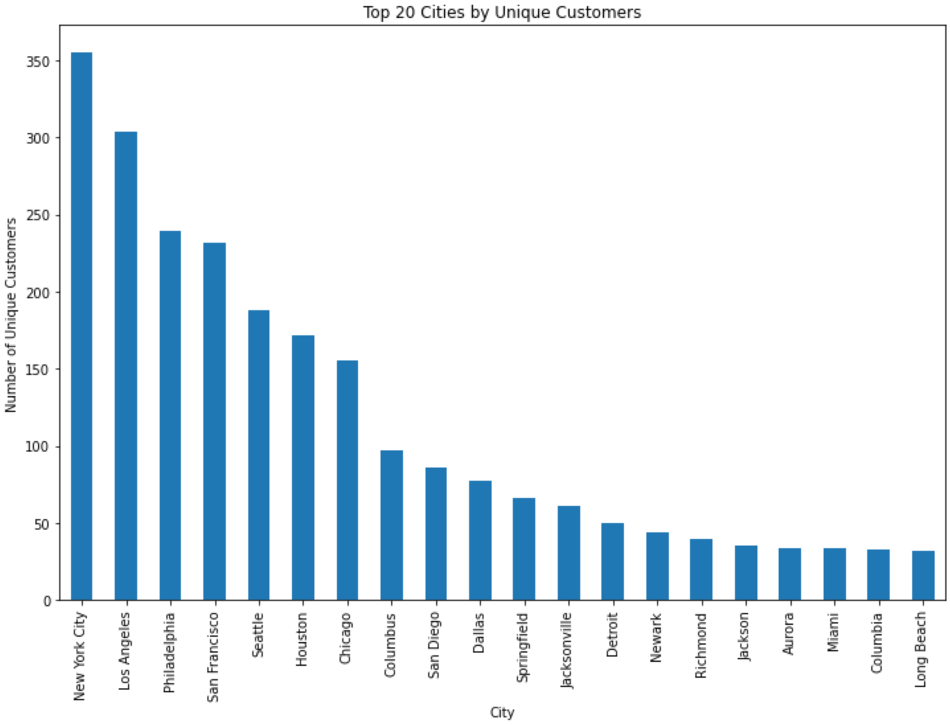
In [132…

```
# line plot for Sales by Order Date
plt.figure(figsize=(8, 6))
sns.lineplot(x='Order Date', y='Sales', data=df)
plt.title('Sales by Order Date')
plt.show()
```



In [133…

```
# bar chart for Customer ID by City
plt.figure(figsize=(12, 8))
df.groupby(['City'])['Customer ID'].nunique().sort_values(ascending=False).head(20).plot(kind='bar')
plt.title('Top 20 Cities by Unique Customers')
plt.xlabel('City')
plt.ylabel('Number of Unique Customers')
plt.show()
```

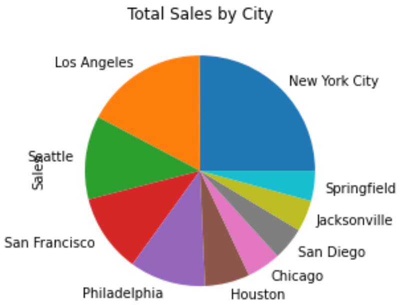


In [134…

```
# Pie chart of sales by city
sales_by_city = df.groupby('City')['Sales'].sum().sort_values(ascending=False)[:10]
sales_by_city.plot(kind='pie', title='Total Sales by City')
```

Out[134]:

```
<AxesSubplot:title={'center':'Total Sales by City'}, ylabel='Sales'>
```

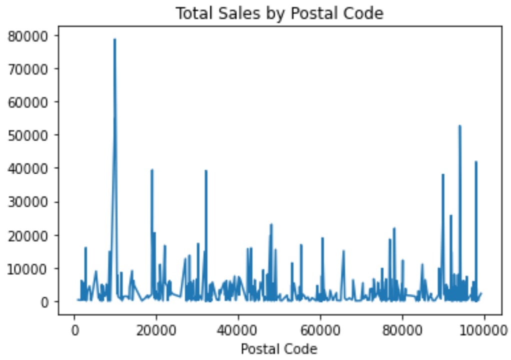


In [135…

```
# Line chart of sales by postal code
sales_by_postal = df.groupby('Postal Code')['Sales'].sum()
sales_by_postal.plot(kind='line', title='Total Sales by Postal Code')
```

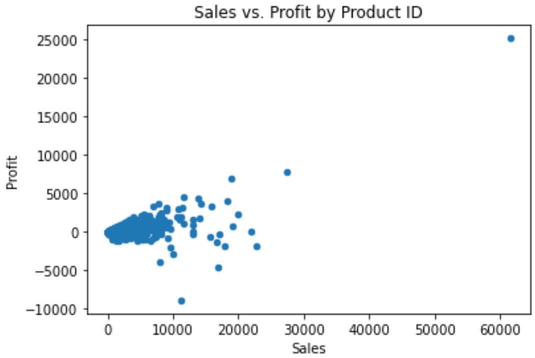
Out[135]:

```
<AxesSubplot:title={'center':'Total Sales by Postal Code'}, xlabel='Postal Code'>
```



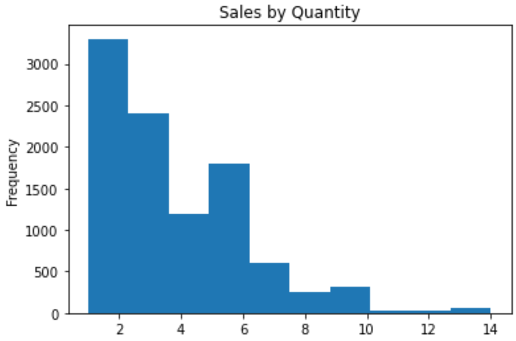
```
In [136…  # Scatter plot of sales and profit by product ID
          sales_profit_by_product = df.groupby('Product ID')[['Sales', 'Profit']].sum()
          sales_profit_by_product.plot(kind='scatter', x='Sales', y='Profit', title='Sales vs. Profit by Product ID')
```

```
Out[136]: <AxesSubplot:title={'center':'Sales vs. Profit by Product ID'}, xlabel='Sales', ylabel='Profit'>
```



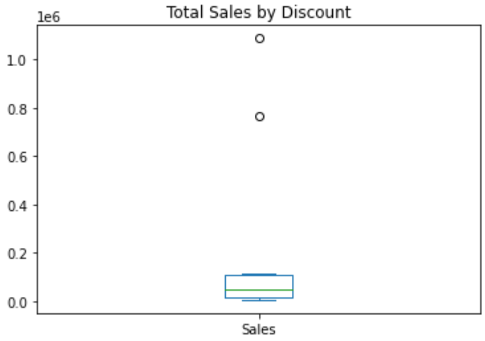
```
In [137…  # Histogram of sales by quantity
          sales_by_quantity = df['Quantity']
          sales_by_quantity.plot(kind='hist', title='Sales by Quantity')
```

```
Out[137]: <AxesSubplot:title={'center':'Sales by Quantity'}, ylabel='Frequency'>
```



```
In [138…  # Box plot of sales by discount
          sales_by_discount = df.groupby('Discount')['Sales'].sum()
          sales_by_discount.plot(kind='box', title='Total Sales by Discount')

          plt.show()
```


Total Sales by Discount
1e6
Sales