Customer Behaviour Analysis using Python

Customer Behavior Analysis is a process that involves examining and understanding how customers interact with a business, product, or service. This analysis helps organizations make informed decisions, tailor their strategies, and enhance customer experiences. If you want to learn how to analyze customer behaviour on a platform, this article is for you. In this article, I'll take you through the task of Customer Behaviour Analysis using Python.

Customer Behaviour Analysis: Process We Can Follow

Customer Behavior Analysis is a valuable process that empowers businesses to make data-driven decisions, enhance customer experiences, and remain competitive in a dynamic market. Below is the process we can follow for the task of Customer Behaviour Analysis:

- Collect data related to customer interactions. It can include purchase history, website visits, social media engagement, customer feedback, and more.
- Identify and address data inconsistencies, missing values, and outliers to ensure the data's quality and accuracy.
- Calculate basic statistics like mean, median, and standard deviation to summarize data.
- Create visualizations such as histograms, scatter plots, and bar charts to explore trends, patterns, and anomalies in the data.
- Use techniques like clustering to group customers based on common behaviours or characteristics.

So, the process starts with collecting data based on customer behaviour on a platform. I found an ideal dataset for this task. You can download the data from here.

Customer Behaviour Analysis using Python

Now, let's get started with the task of Customer Behaviour Analysis by importing the necessary Python libraries and the dataset:

In [1]:
```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go

data = pd.read_csv(r"C:\Users\shali\OneDrive\Documents\Data analysis project\ecommerce_customer_data.csv")
print(data.head())
```

```
   User_ID  Gender  Age   Location Device_Type  Product_Browsing_Time  \
0        1  Female   23  Ahmedabad      Mobile                     60   
1        2    Male   25    Kolkata      Tablet                     30   
2        3    Male   32  Bangalore     Desktop                     37   
3        4    Male   35      Delhi      Mobile                      7   
4        5    Male   27  Bangalore      Tablet                     35   

   Total_Pages_Viewed  Items_Added_to_Cart  Total_Purchases  
0                  30                    1                0  
1                  38                    9                4  
2                  13                    5                0  
3                  20                   10                3  
4                  20                    8                2  
```

Before moving forward, let's have a look at the summary statistics for both numerical and categorical columns in the dataset:

In [2]:
```
# Summary statistics for numeric columns
numeric_summary = data.describe()
print(numeric_summary)
```

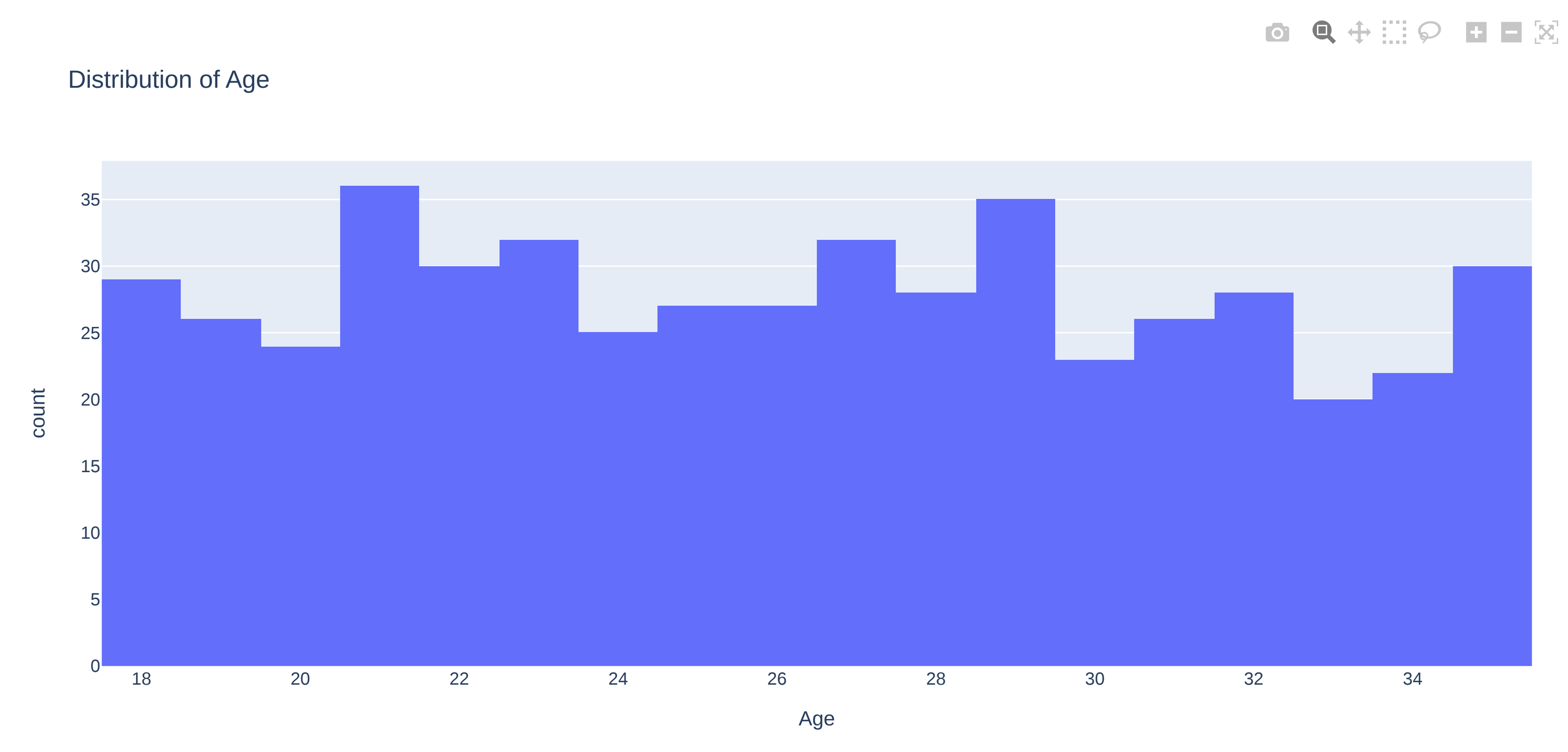
```
          User_ID         Age  Product_Browsing_Time  Total_Pages_Viewed  \
count  500.000000  500.000000             500.000000          500.000000   
mean   250.500000   26.276000              30.740000           27.182000   
std    144.481833    5.114699              15.934246           13.071596   
min      1.000000   18.000000               5.000000            5.000000   
25%    125.750000   22.000000              16.000000           16.000000   
50%    250.500000   26.000000              31.000000           27.000000   
75%    375.250000   31.000000              44.000000           38.000000   
max    500.000000   35.000000              60.000000           50.000000   

       Items_Added_to_Cart  Total_Purchases  
count           500.000000       500.000000  
mean              5.150000         2.464000  
std               3.203127         1.740909  
min               0.000000         0.000000  
25%               2.000000         1.000000  
50%               5.000000         2.000000  
75%               8.000000         4.000000  
max              10.000000         5.000000  
```

In [3]:
```
# Summary for non-numeric columns
categorical_summary = data.describe(include='object')
print(categorical_summary)
```

```
       Gender Location Device_Type
count     500      500         500
unique      2        8           3
top      Male  Kolkata      Mobile
freq      261       71         178
```

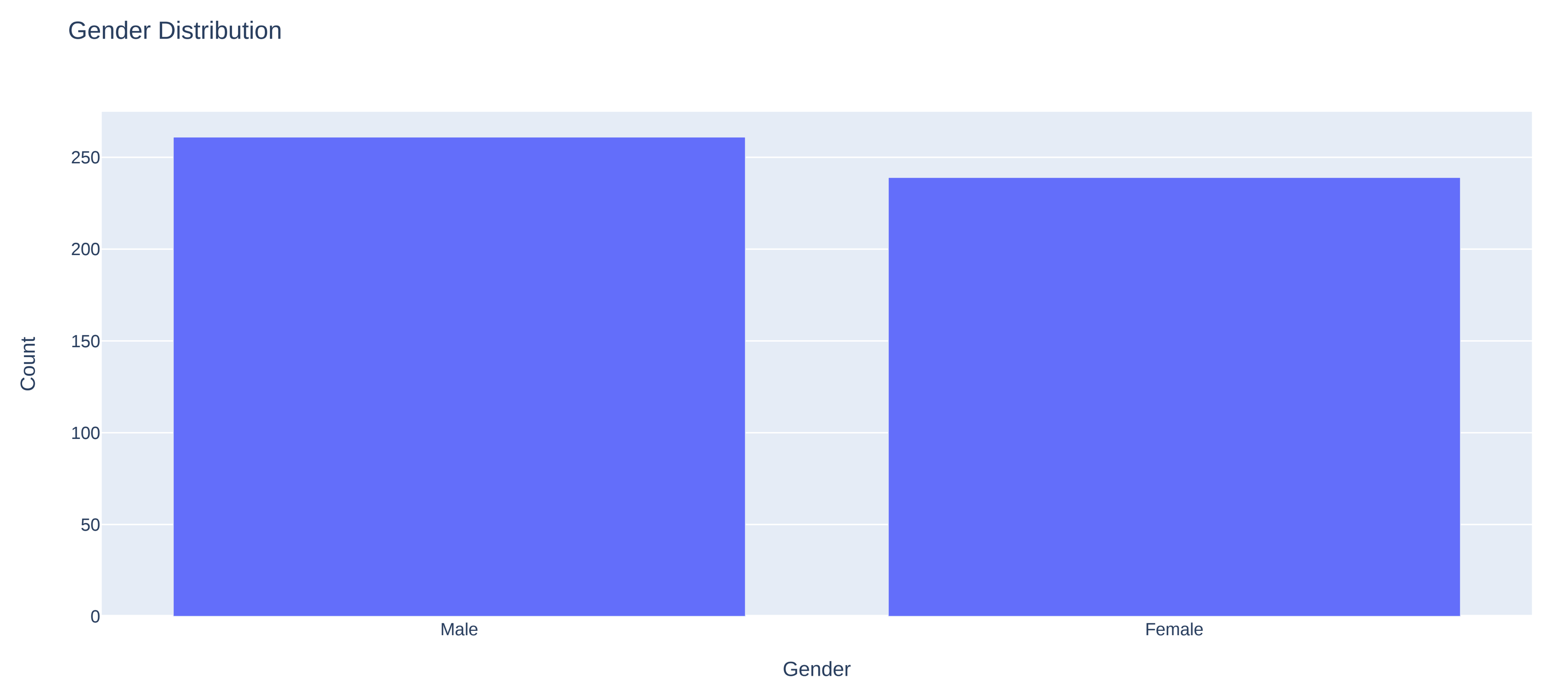
Now, let's have a look at the distribution of age in the dataset:

In [4]:
```
# Histogram for 'Age'
fig = px.histogram(data, x='Age', title='Distribution of Age')
fig.show()
```



Now, let's have a look at the gender distribution:

In [5]:
```
# Bar chart for 'Gender'
gender_counts = data['Gender'].value_counts().reset_index()
gender_counts.columns = ['Gender', 'Count']
fig = px.bar(gender_counts, x='Gender',
             y='Count',
             title='Gender Distribution')
fig.show()
```



Analyzing Customer Behaviour

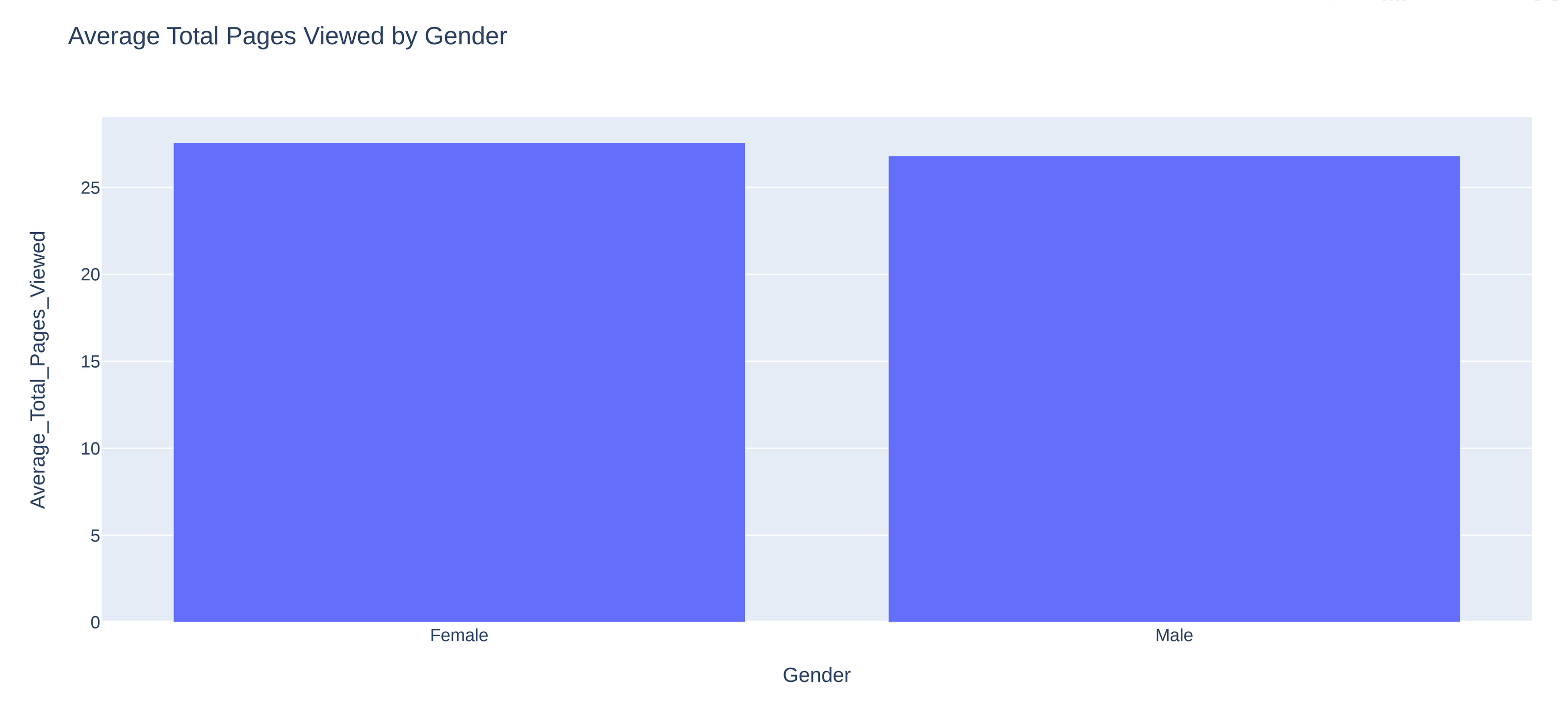
Now, let's have a look at the relationship between the product browsing time and the total pages viewed:

In [6]:
```
# 'Product_Browsing_Time' vs 'Total_Pages_Viewed'
fig = px.scatter(data, x='Product_Browsing_Time', y='Total_Pages_Viewed',
                 title='Product Browsing Time vs. Total Pages Viewed',
                 trendline='ols')
fig.show()
```

The above scatter plot shows no consistent pattern or strong association between the time spent browsing products and the total number of pages viewed. It indicates that customers are not necessarily exploring more pages if they spend more time on the website, which might be due to various factors such as the website design, content relevance, or individual user preferences.

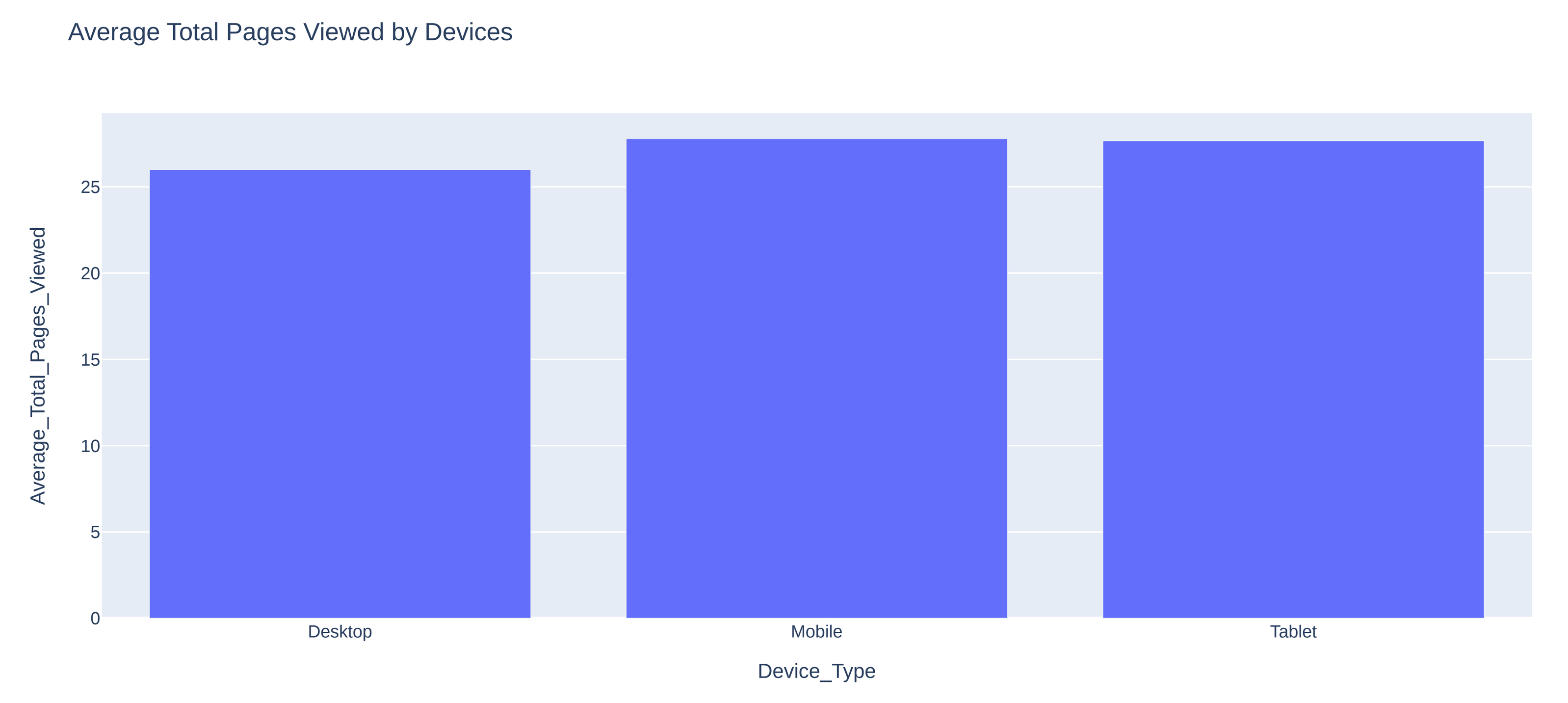
Now, let's have a look at the average total pages viewed by gender:

In [7]:
```
# Grouped Analysis
gender_grouped = data.groupby('Gender')['Total_Pages_Viewed'].mean().reset_index()
gender_grouped.columns = ['Gender', 'Average_Total_Pages_Viewed']
fig = px.bar(gender_grouped, x='Gender', y='Average_Total_Pages_Viewed',
             title='Average Total Pages Viewed by Gender')
fig.show()
```



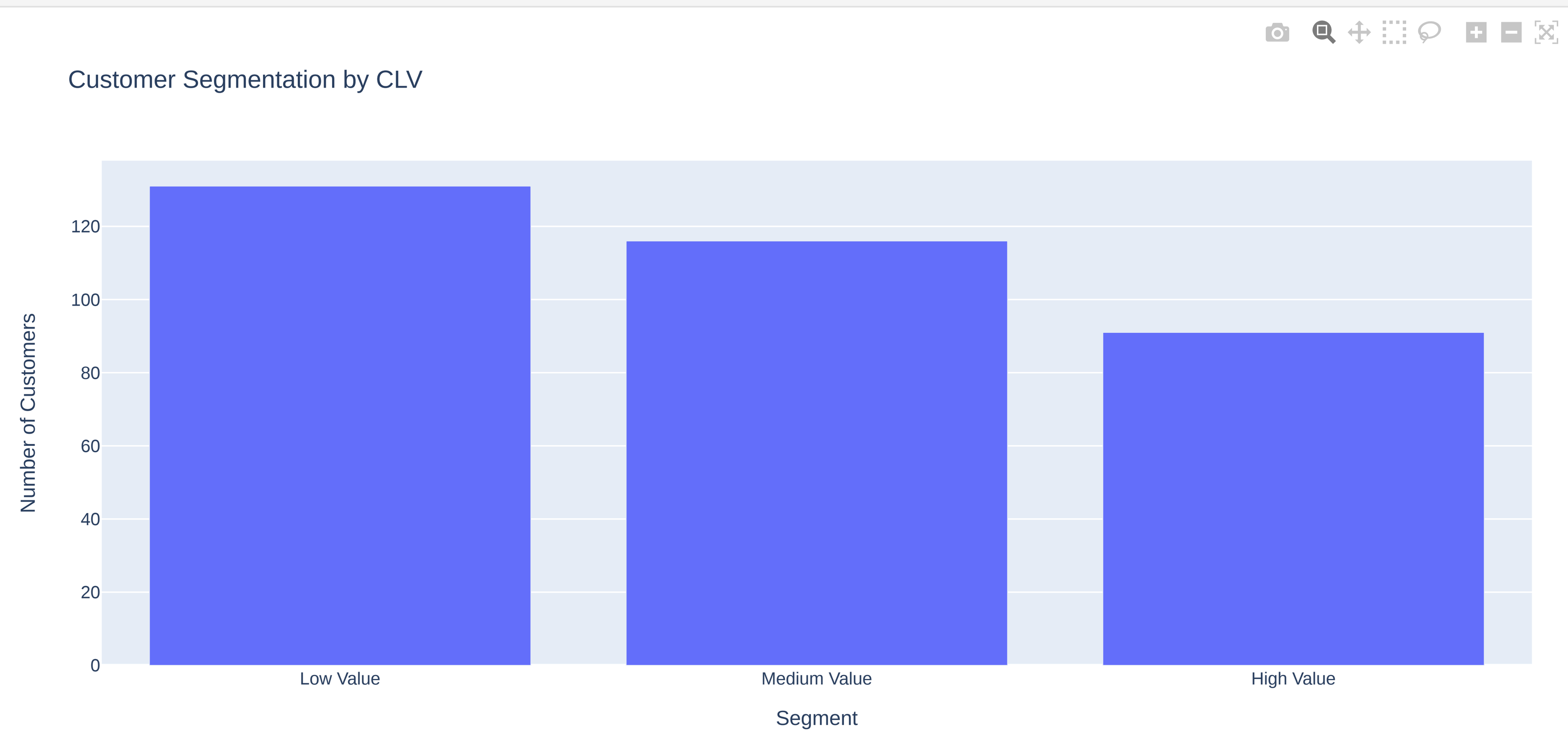
Now, let's have a look at the average total pages viewed by devices:

In [8]:
```
devices_grouped = data.groupby('Device_Type')['Total_Pages_Viewed'].mean().reset_index()
devices_grouped.columns = ['Device_Type', 'Average_Total_Pages_Viewed']
fig = px.bar(devices_grouped, x='Device_Type', y='Average_Total_Pages_Viewed',
             title='Average Total Pages Viewed by Devices')
fig.show()
```



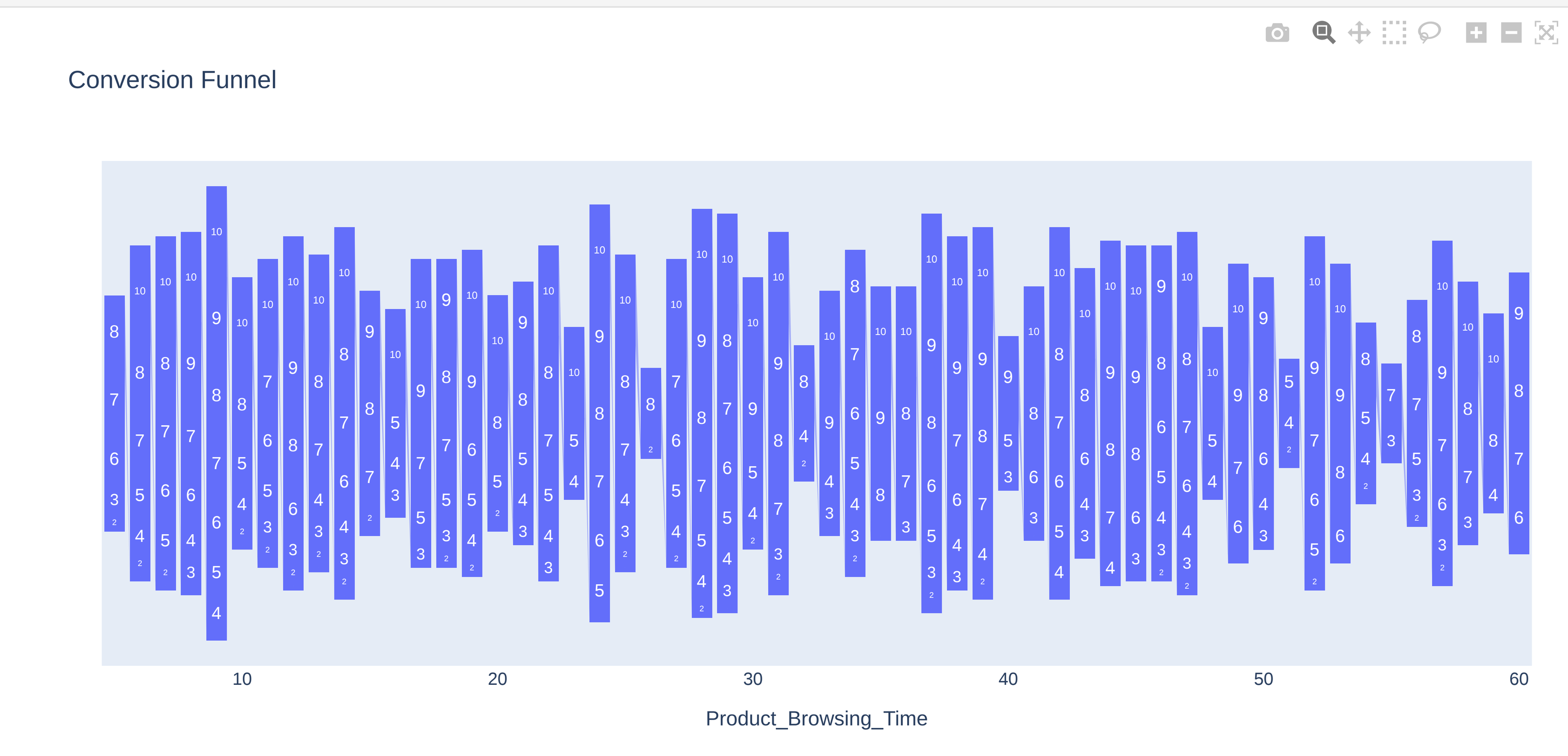
Now, let's calculate the customer lifetime value and visualize segments based on the customer lifetime value:

In [10]:
```
data['CLV'] = (data['Total_Purchases'] * data['Total_Pages_Viewed']) / data['Age']

data['Segment'] = pd.cut(data['CLV'], bins=[1, 2.5, 5, float('inf')],
                         labels=['Low Value', 'Medium Value', 'High Value'])

segment_counts = data['Segment'].value_counts().reset_index()
segment_counts.columns = ['Segment', 'Count']

# Create a bar chart to visualize the customer segments
fig = px.bar(segment_counts, x='Segment', y='Count',
             title='Customer Segmentation by CLV')
fig.update_xaxes(title='Segment')
fig.update_yaxes(title='Number of Customers')
fig.show()
```



Now, let's have a look at the conversion funnel of the customers:

In [11]:
```
# Funnel analysis
funnel_data = data[['Product_Browsing_Time', 'Items_Added_to_Cart', 'Total_Purchases']]
funnel_data = funnel_data.groupby(['Product_Browsing_Time', 'Items_Added_to_Cart']).sum().reset_index()

fig = px.funnel(funnel_data, x='Product_Browsing_Time', y='Items_Added_to_Cart', title='Conversion Funnel')
fig.show()
```



In the above graph, the x-axis represents the time customers spend browsing products on the e-commerce platform. The y-axis represents the number of items added to the shopping cart by customers during their browsing sessions.

Now, let's have a look at the churn rate of the customers:

In [12]:
```
# Calculate churn rate
data['Churned'] = data['Total_Purchases'] == 0

churn_rate = data['Churned'].mean()
print(churn_rate)
```

```
0.198
```

A churn rate of 0.198 indicates that a significant portion of customers has churned, and addressing this churn is important for maintaining business growth and profitability.

So, this is how you can analyze customer behaviour on a platform using Python. You can find many more Data Analysis projects solved and explained using Python here.

Summary

Customer Behavior Analysis is a process that involves examining and understanding how customers interact with a business, product, or service. This analysis helps organizations make informed decisions, tailor their strategies, and enhance customer experiences. I hope you liked this article on Customer Behaviour Analysis using Python. Feel free to ask valuable questions in the comments section below.

In [ ]: