Data Visualization with Seaborn

Introduction

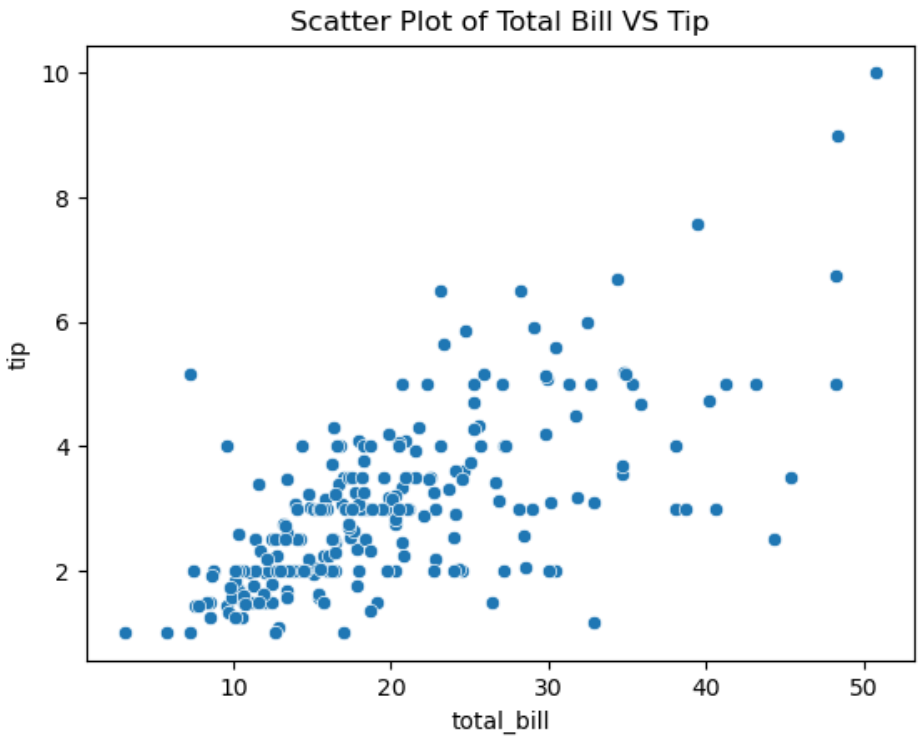
Seaborn is a Python visualization library based on Matplotlib that provides a high-level interface for drawing attractive and informative statistical graphics. Seaborn helps in creating complex visualizations with just a few lines of code. In this lesson, we will cover the basics of Seaborn, including creating various types of plots and customizing them.

Data Source: Tips Dataset

The 'tips' dataset is included in the Seaborn library and contains information about restaurant tips. It includes the following columns:

Column	***Description***
total_bill	Total bill (in USD)
tip	Tip amount (in USD)
sex	Gender of the person paying the bill (Male/Female)
smoker	Whether the person is a smoker (Yes/No)
day	Day of the week (Sun, Sat, Thur, Fri)
time	Time of day (Lunch/Dinner)
size	Size of the party (number of people)

Scatter Plot

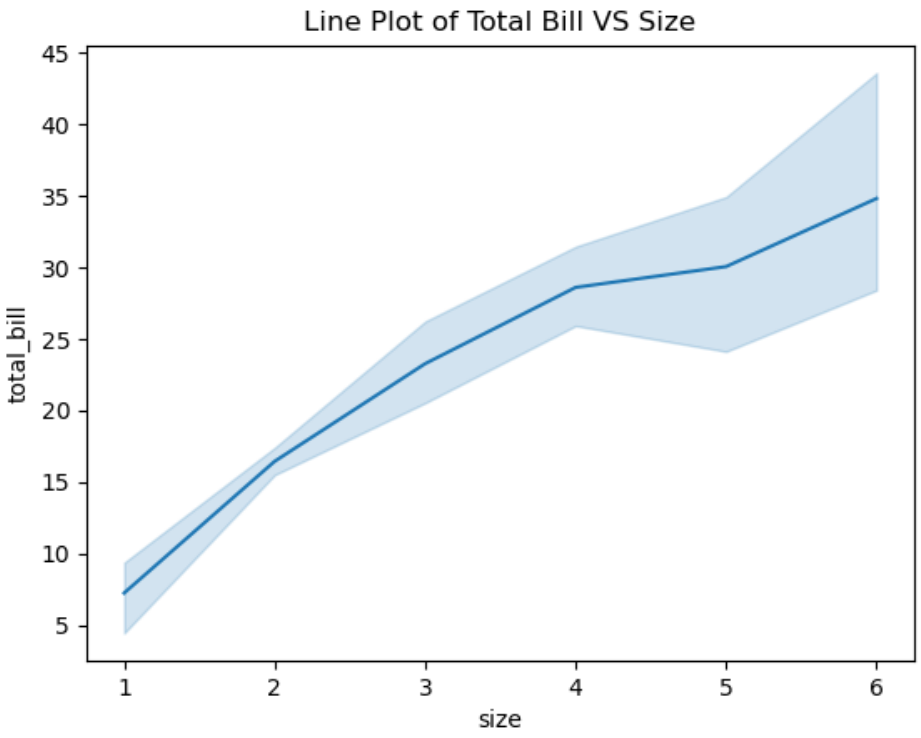


Code:

```
sns.scatterplot(x='total_bill', y='tip', data=tips)
```

Key Definition: A scatter plot displays values for typically two variables for a set of data. Each point represents an observation in the dataset.

Line Plot

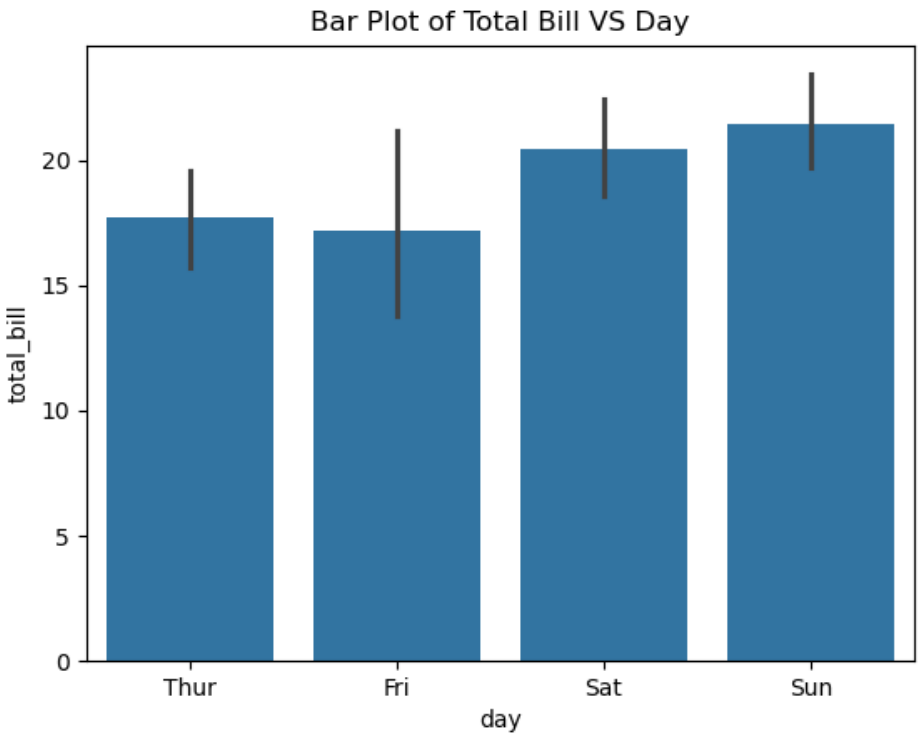


Code:

```
sns.lineplot(x='size', y='total_bill', data=tips)
```

Key Definition: A line plot is used to display information as a series of data points called 'markers' connected by straight line segments.

Bar Plot

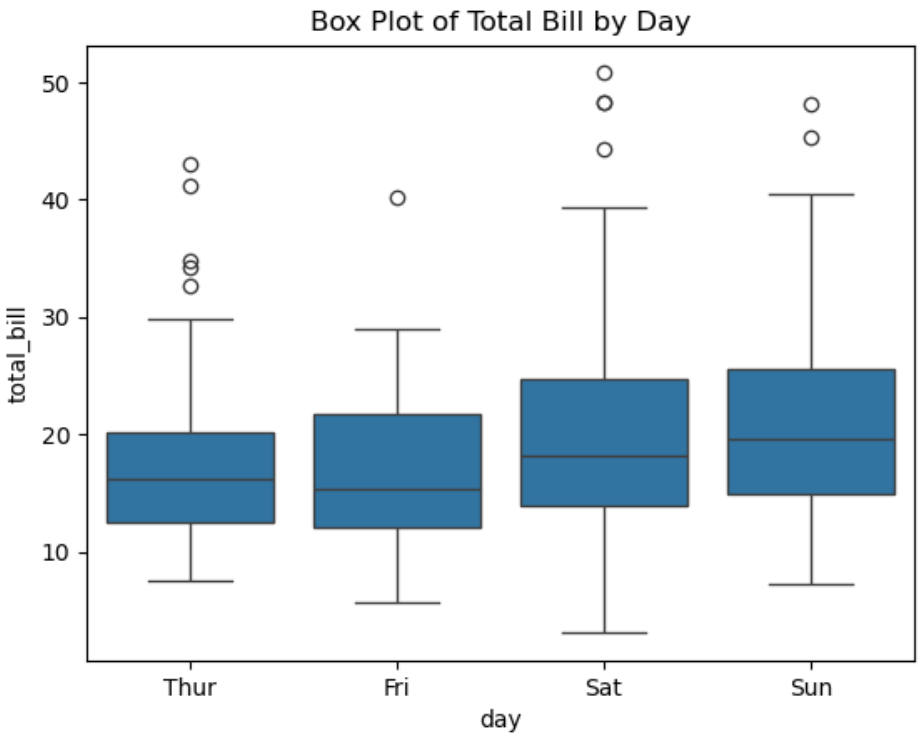


Code:

```
sns.barplot(x='day', y='total_bill', data=tips)
```

Key Definition: A bar plot is a chart that presents categorical data with rectangular bars. The lengths of the bars are proportional to the values they represent.

Box Plot

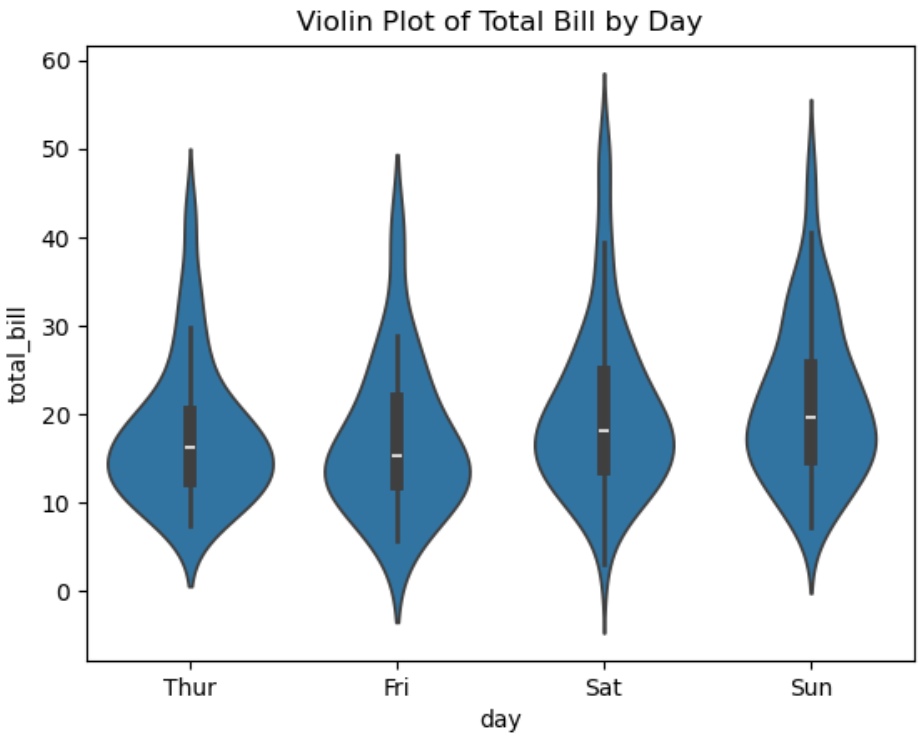


Code:

```
sns.boxplot(x='day', y='total_bill', data=tips)
```

Key Definition: A box plot (or whisker plot) displays the distribution of data based on a five-number summary ('minimum', first quartile (Q1), median, third quartile (Q3), and 'maximum').

Violin Plot

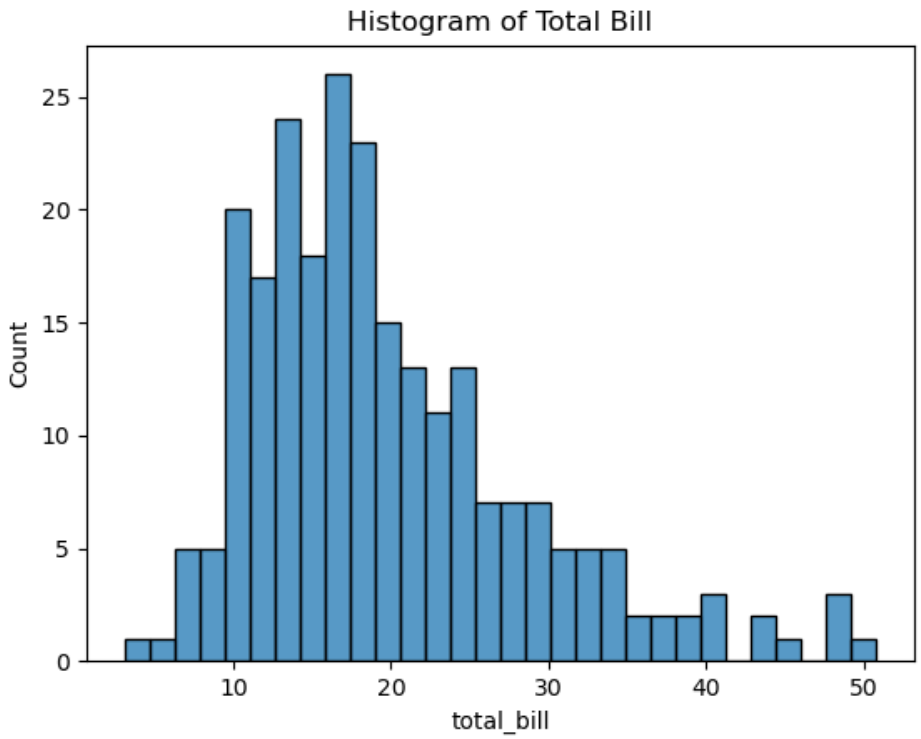


Code:

```
sns.violinplot(x='day', y='total_bill', data=tips)
```

Key Definition: A violin plot is similar to a box plot, but it also shows the probability density of the data at different values, which is useful for visualizing the distribution of the data.

Histogram

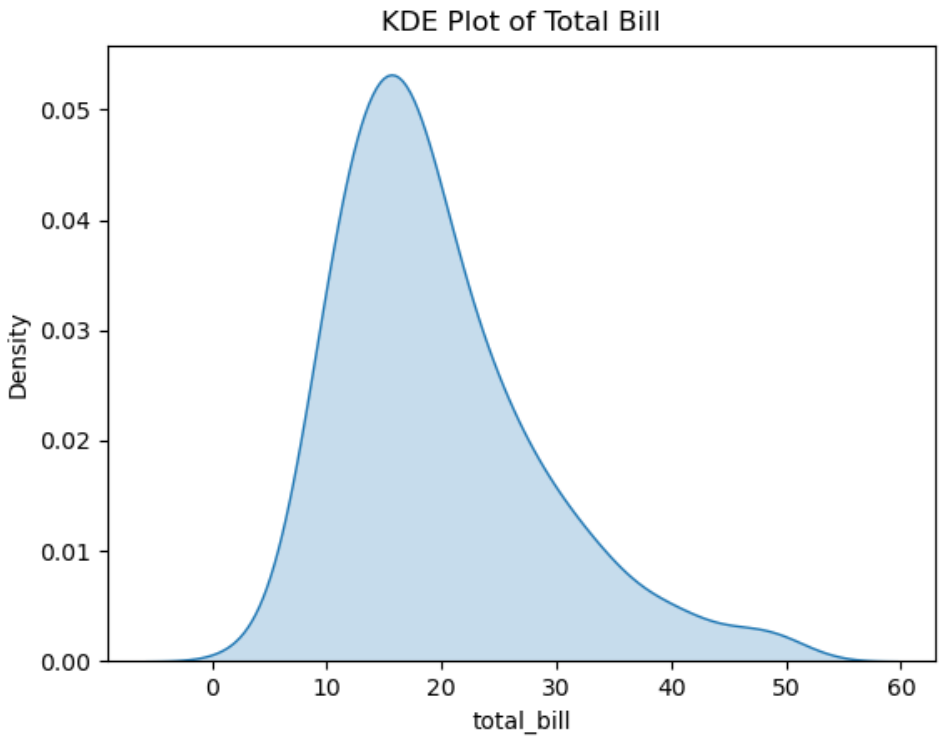


Code:

```
sns.histplot(tips['total_bill'], bins=30, kde=False)
```

Key Definition: A histogram is a graphical representation of the distribution of numerical data, showing the number of data points that fall within a specified range of values (bins).

KDE Plot

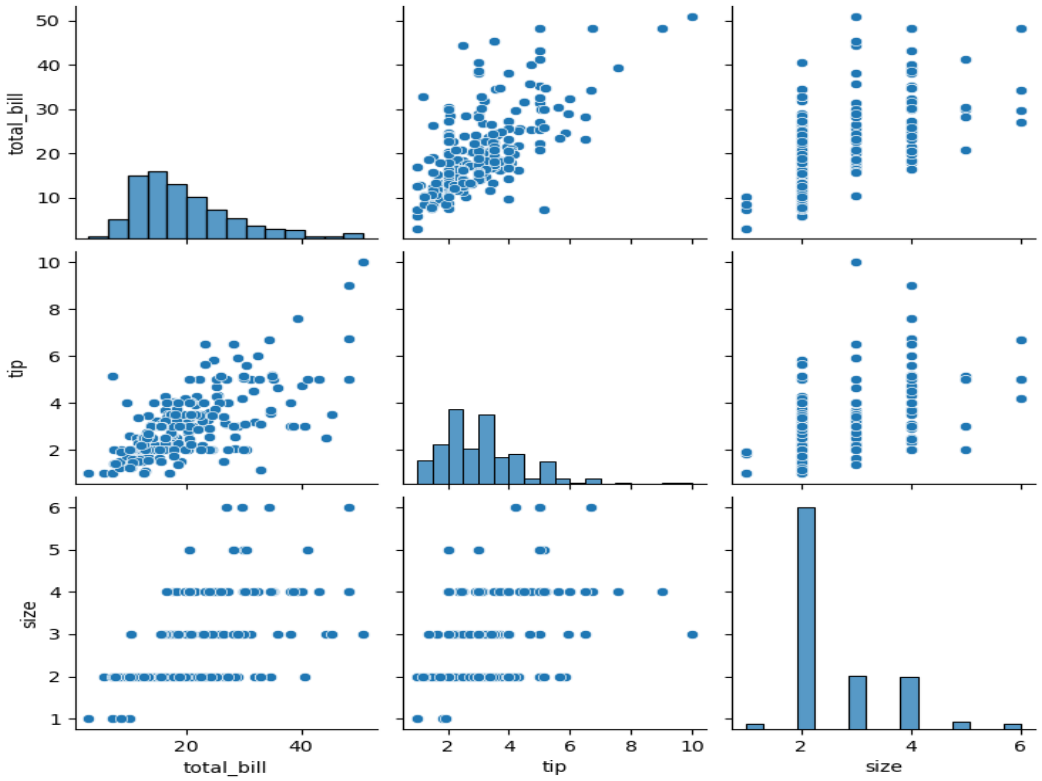


Code:

```
sns.kdeplot(tips['total_bill'], fill=True)
```

Key Definition: A Kernel Density Estimate (KDE) plot is a way to estimate the probability density function of a random variable, providing a smooth curve that represents the distribution of the data.

Pair Plot

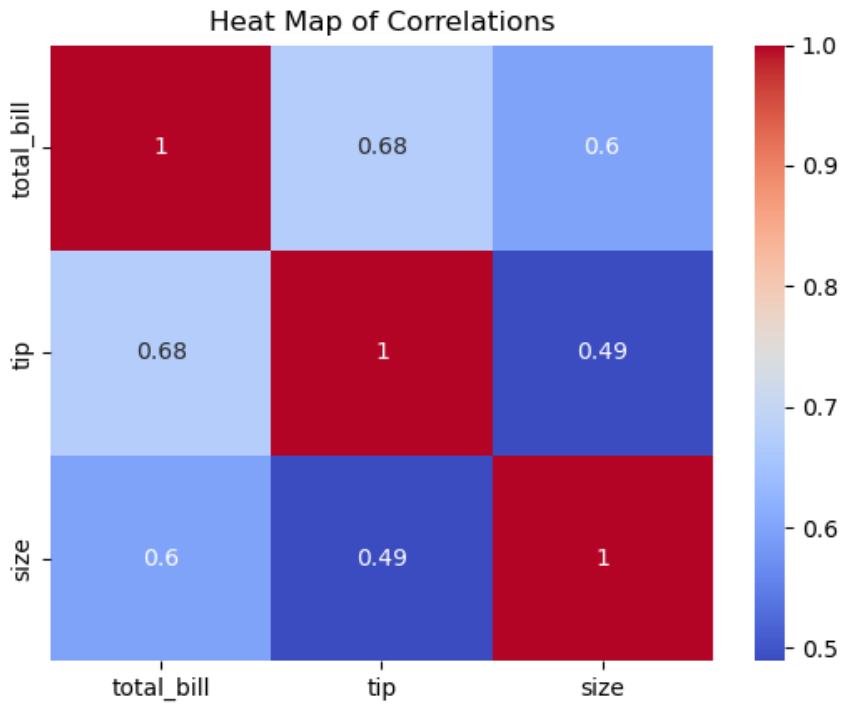


Code:

```
sns.pairplot(tips)
```

Key Definition: A pair plot is a matrix of scatter plots that shows the relationships between multiple variables in a dataset, allowing for easy visualization of correlations.

Heat Map

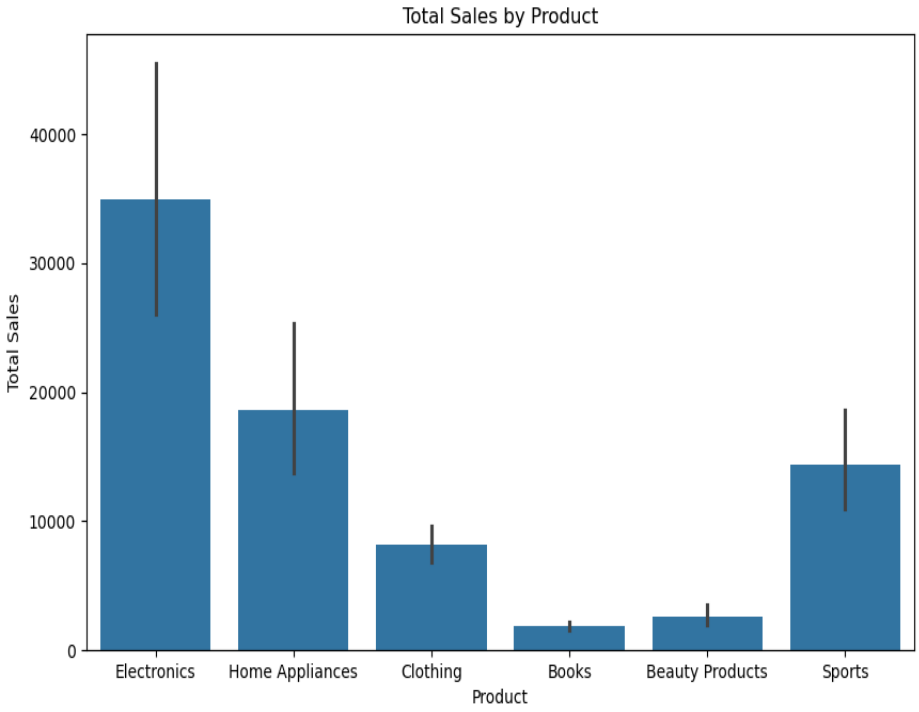


Code:

```
corr=tips[['total_bill','tip','size']].corr() sns.heatmap(corr, annot=True,
cmap='coolwarm')
```

Key Definition: A heat map is a data visualization technique that shows the magnitude of a phenomenon as color in two dimensions, allowing for easy identification of patterns and correlations.

Total Sales by Product

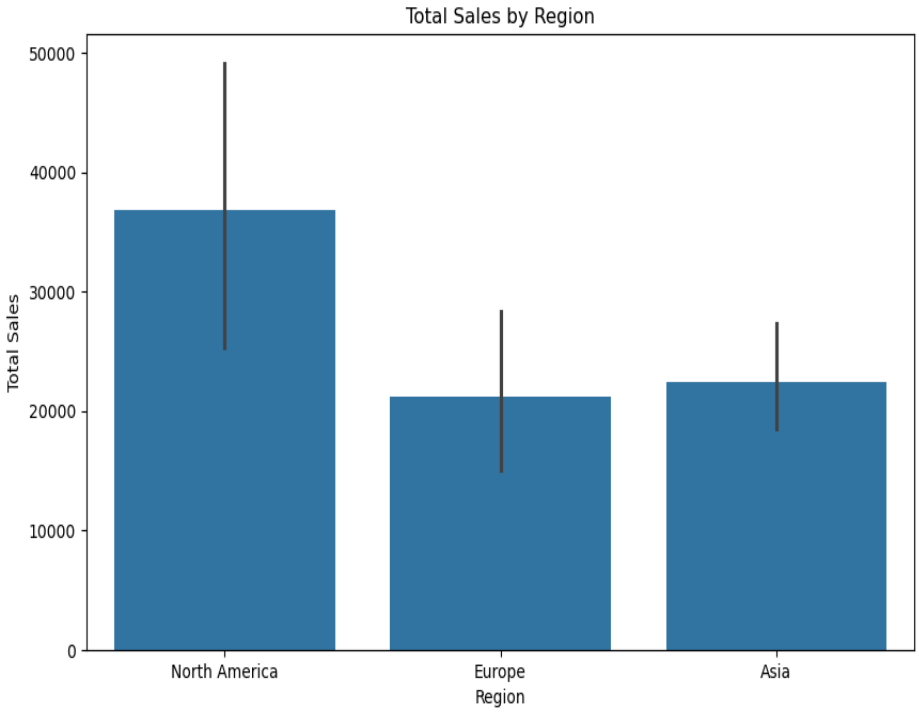


Code:

```
plt.figure(figsize=(10,6)) sns.barplot(x='Product Category',y='Total
Revenue',data=sales_df,estimator=sum) plt.title('Total Sales by Product')
plt.xlabel('Product') plt.ylabel('Total Sales') plt.show()
```

Key Definition: This bar plot shows the total sales revenue for each product category, providing insights into which categories are performing best.

Total Sales by Region



Code:

```
plt.figure(figsize=(10,6)) sns.barplot(x='Region',y='Total
Revenue',data=sales_df,estimator=sum) plt.title('Total Sales by Region')
plt.xlabel('Region') plt.ylabel('Total Sales') plt.show()
```

Key Definition: This bar plot illustrates the total sales revenue across different regions, highlighting regional performance.