

Seaborn

What is Seaborn?

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Seaborn makes it easier to create complex visualizations with less code and is especially powerful when working with pandas DataFrames.

Why is Seaborn Needed?

- It simplifies creating statistical plots.
- It comes with built-in themes and color palettes for better-looking charts.
- It integrates well with pandas, allowing easy plotting of DataFrame columns.
- It supports complex visualizations like heatmaps, violin plots, and categorical scatterplots with very little effort.

Plots in Seaborn:

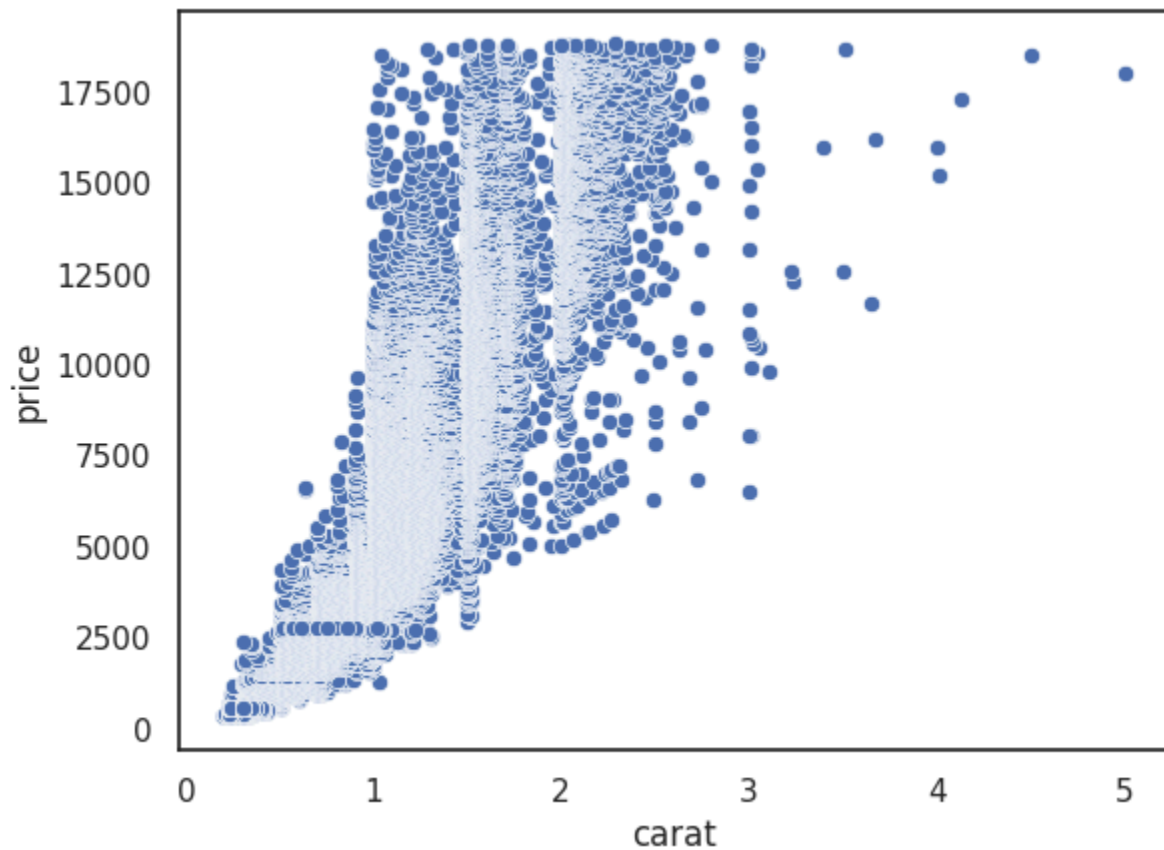
1. Scatter Plot (sns.scatterplot)

A scatter plot shows the relationship between two numerical variables by plotting dots on a coordinate plane. It's great for visualizing correlations or clustering between values like carat and price.

```
import seaborn as sns
import matplotlib.pyplot as plt

diamonds = sns.load_dataset('diamonds')

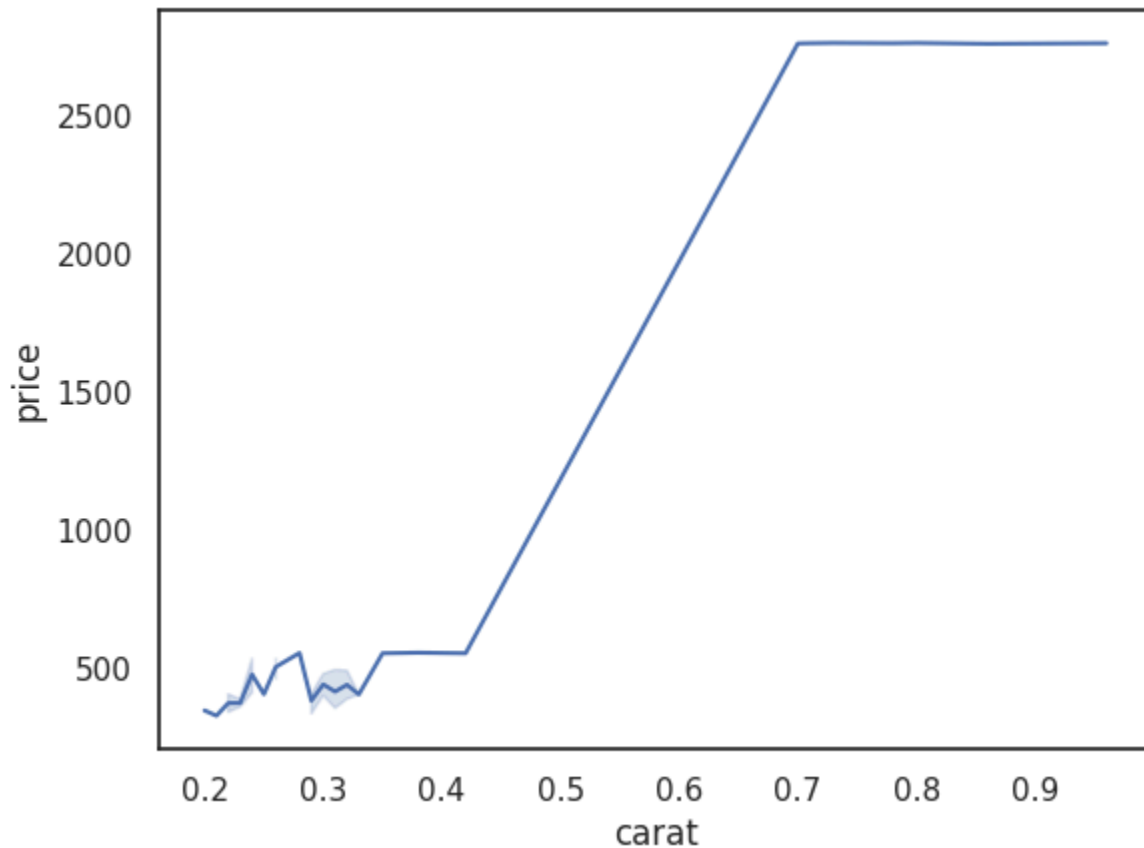
sns.scatterplot(data=diamonds, x='carat', y='price')
plt.show()
```



Abinaya Vina
RA2211031010145 (U2)
Data Science
2. Line Plot (sns.lineplot)

A line plot connects data points with a line to show trends over a continuous variable, such as time, size, or quantity. It helps in analyzing how a variable changes over an interval.

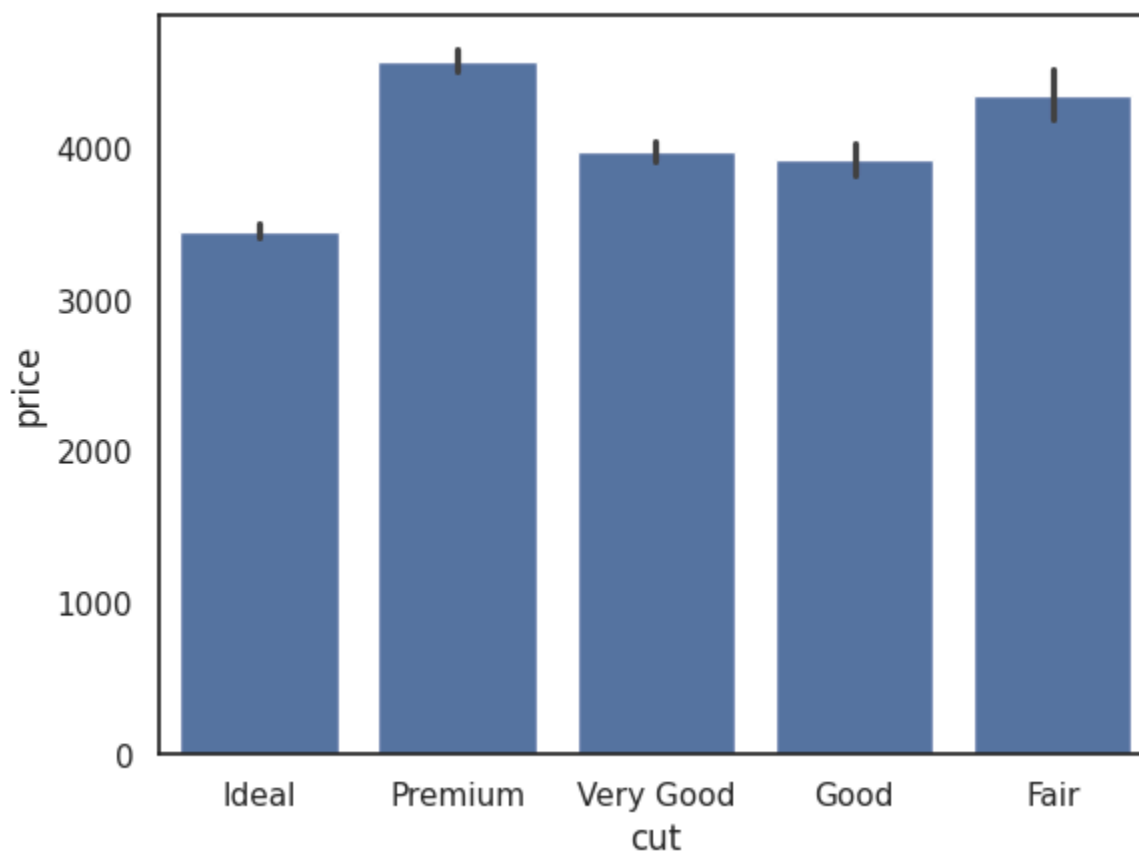
```
sns.lineplot(data=diamonds.head(100), x='carat', y='price')  
plt.show()
```



3. Bar Plot

A bar plot displays the average or other aggregate values of a numerical variable across different categories. It's useful for comparing group-wise statistics like average price by cut quality.

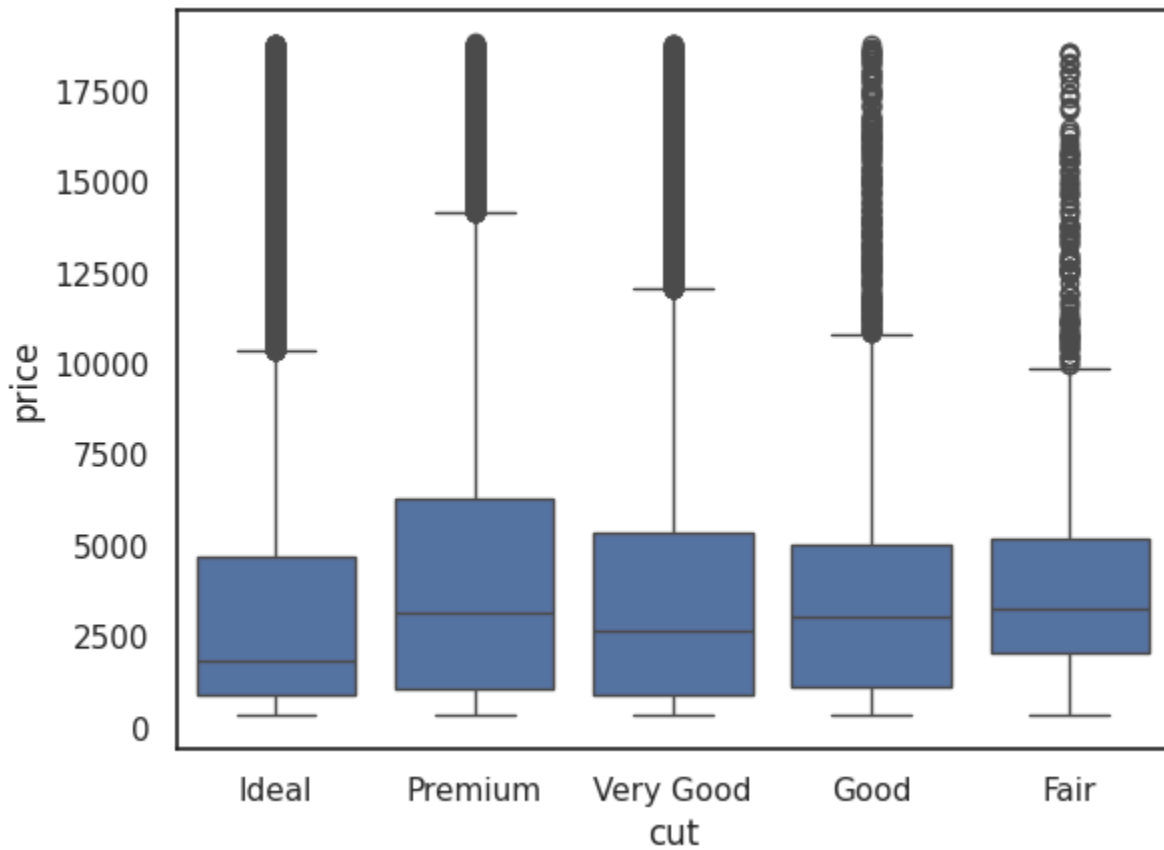
```
sns.barplot(data=diamonds, x='cut', y='price')  
plt.show()
```



4. Box Plot

A box plot shows the distribution of data using five summary statistics: minimum, first quartile, median, third quartile, and maximum. It also highlights outliers, making it ideal for comparing spread across group

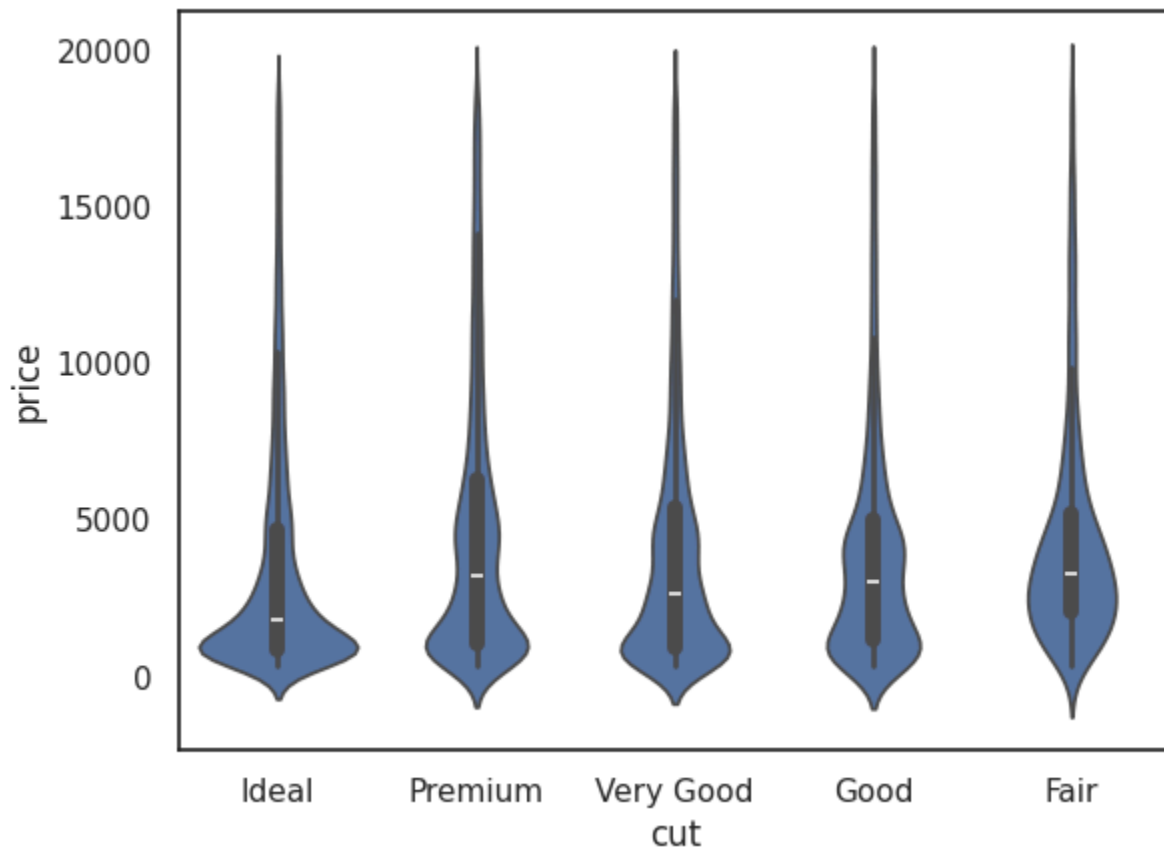
```
sns.boxplot(data=diamonds, x='cut', y='price')  
plt.show()
```



5. Violin Plot

A violin plot combines aspects of a box plot and a KDE plot. It shows the distribution's shape along with the central values, giving a richer picture of the data's spread and density.

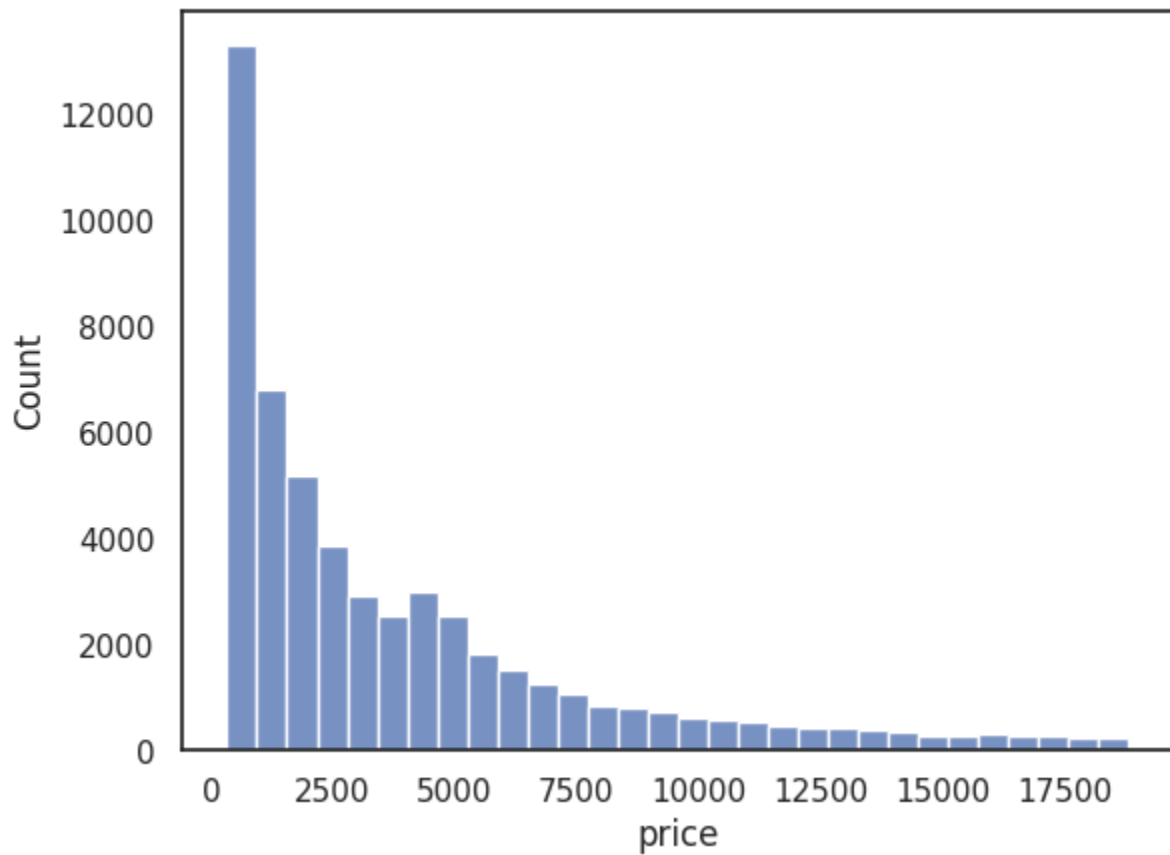
```
sns.violinplot(data=diamonds, x='cut', y='price')  
plt.show()
```



6. Histogram

A histogram groups numerical data into bins and shows the frequency of data points in each bin. It helps understand the distribution, skewness, and modality of a variable.

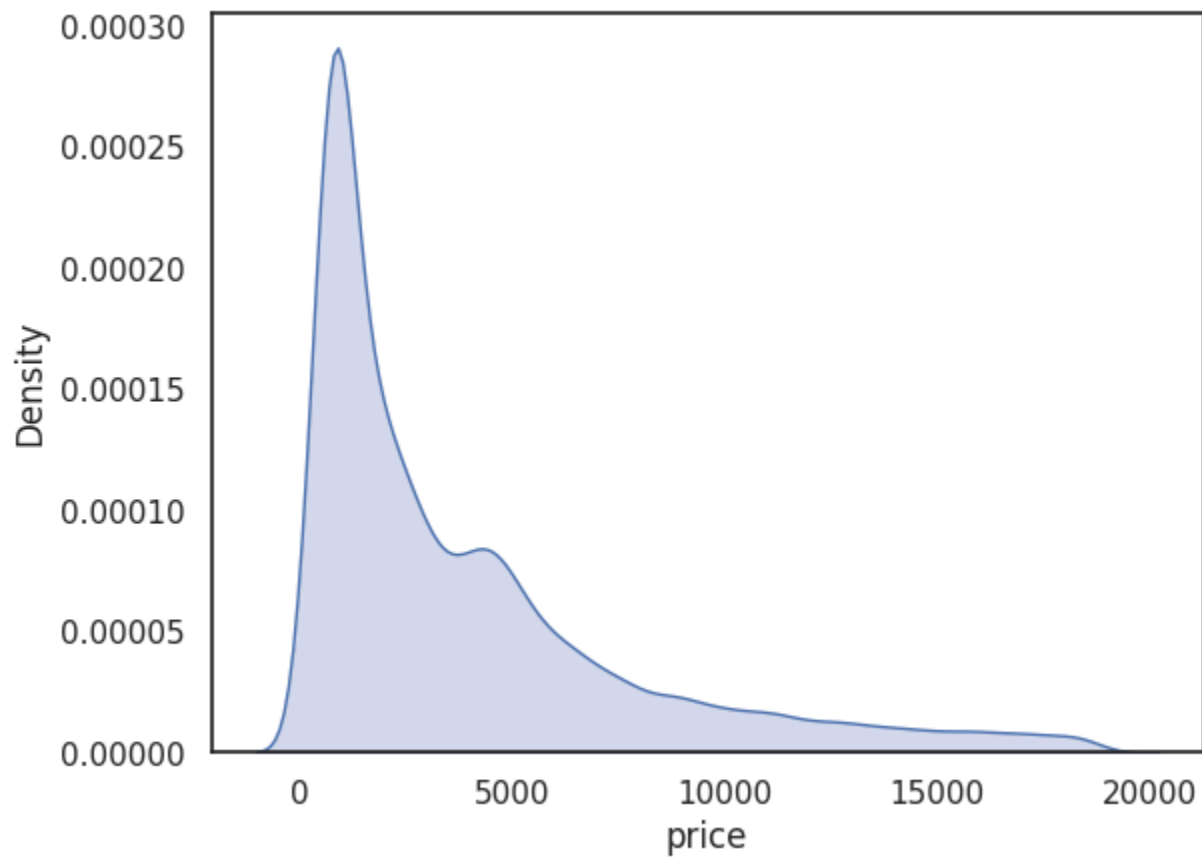
```
sns.histplot(data=diamonds, x='price', bins=30)  
plt.show()
```



7. KDE Plot

A KDE (Kernel Density Estimate) plot is a smooth version of a histogram. It estimates the probability density function of a variable and is useful for visualizing the underlying distribution.

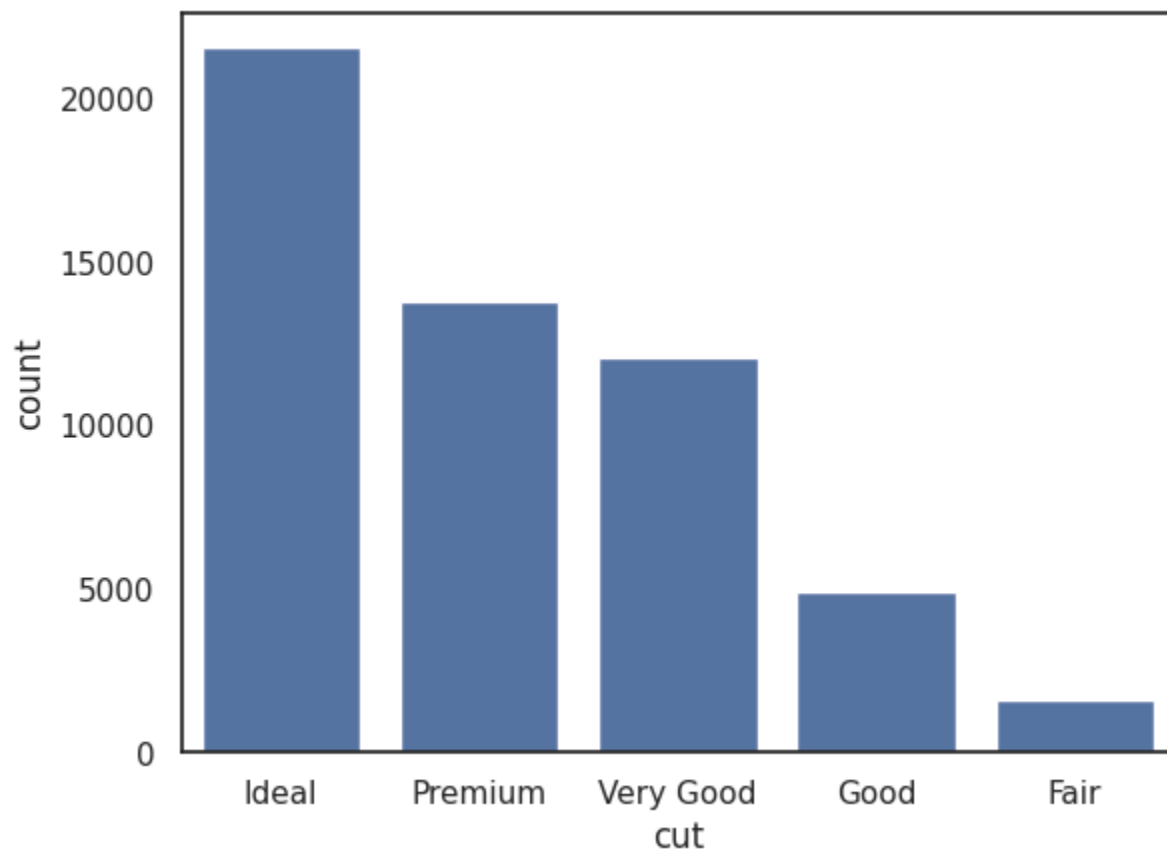
```
sns.kdeplot(data=diamonds['price'], shade=True)  
plt.show()
```



8. Count Plot

A count plot shows the count of observations for each category in a categorical variable. It's a quick way to understand how frequently each category appears in the data.

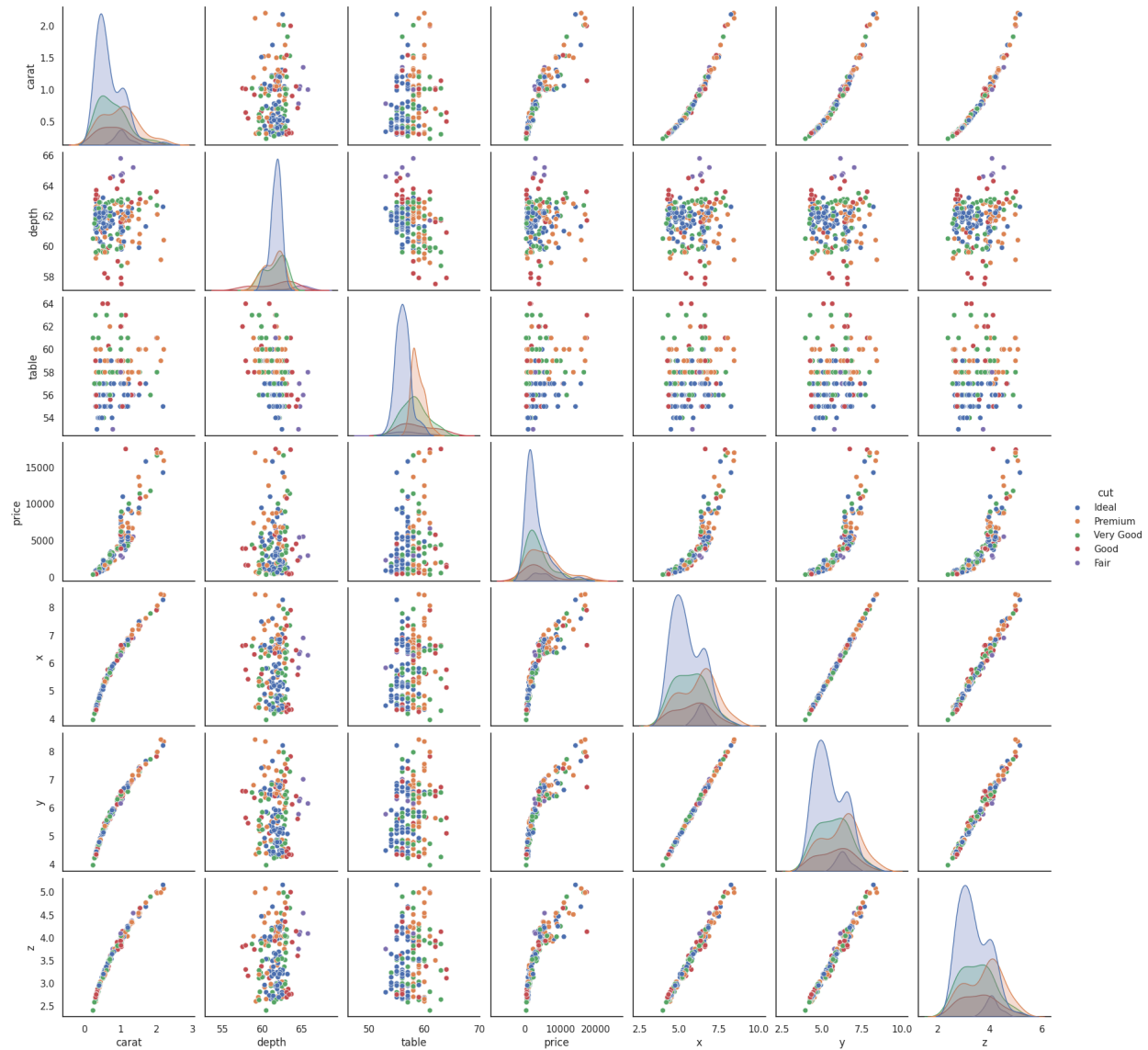
```
sns.countplot(data=diamonds, x='cut')  
plt.show()
```

9. Pair Plot

A pair plot displays scatter plots for all combinations of numerical variables in a dataset. It's excellent for exploring patterns, relationships, and clusters among variables.

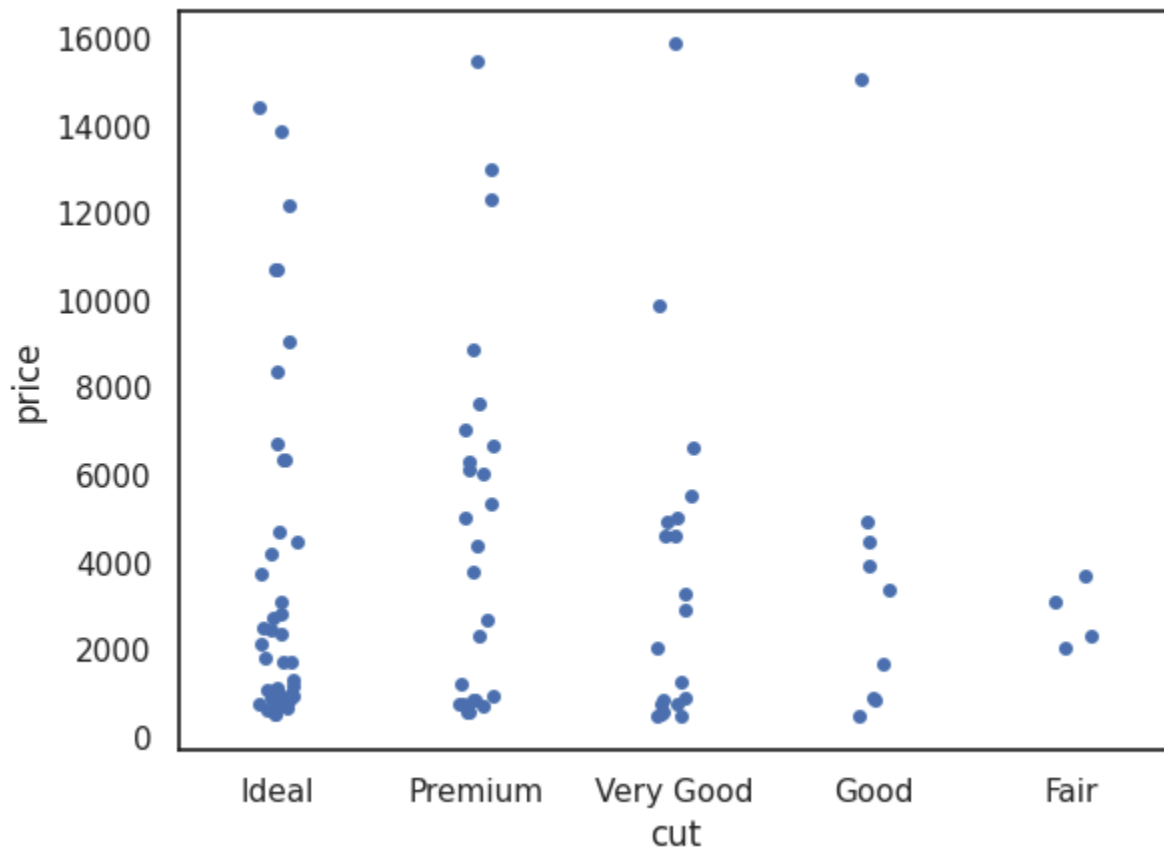
```
sns.pairplot(diamonds.sample(200), hue='cut')  
plt.show()
```



10. Strip Plot

A strip plot places all individual data points along a categorical axis, with a small jitter to avoid overlap. It helps to visualize the spread and distribution of raw data across categories.

```
sns.stripplot(data=diamonds.sample(100), x='cut', y='price')  
plt.show()
```



11. Swarm Plot

A swarm plot is similar to a strip plot but arranges points to prevent overlap entirely. It clearly shows the distribution of values without sacrificing individual data visibility.

```
sns.swarmplot(data=diamonds.sample(100), x='cut', y='price')  
plt.show()
```

