CAP 5768: Homework Assignment 1

Place name here: Krushal Kalkani

Preliminary instructions

All analyses must be performed in Python using the packages we discussed in class. Fill in all your solutions in the appropriate spaces in this Word document, and then upload a PDF copy to Canvas. **Only PDF copies will be graded**.

Brief overview of the assignment

In this assignment, you will analyze the **diamonds** data frame available in Canvas under the modules section, which contains **53,930** observations on ten features related to diamonds. A large portion of this assignment is to get to know Python and the learned visualization packages better, so you will need to use resources such as the **`seaborn`** or **`plotly`** cheat sheet on Canvas and other resources from the textbook and online to learn how to make certain types of plots.

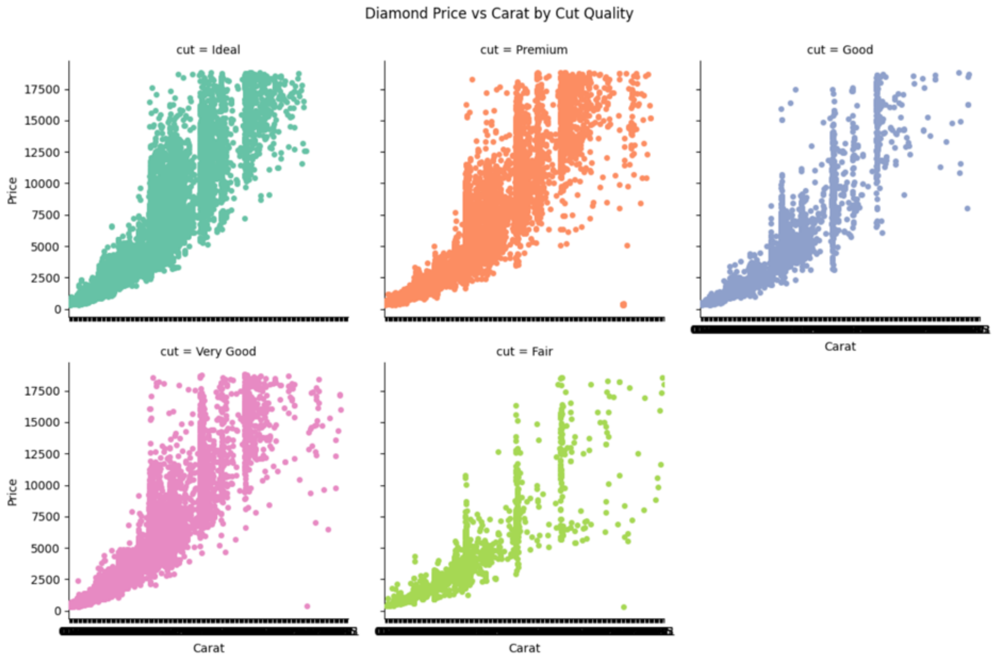
Questions and problems

1. [20%] Using **`Seaborn`** or **`Plotly`**, create a jittered scatter plot to visualize the relationship between diamond price and diamond weight (carat). The scatter plot should be divided into five sub-panels, arranged in two rows, where each sub-panel represents a different diamond cut quality. Additionally, color the points based on the cut quality. Use appropriate functions from Seaborn's **`FacetGrid`** or Plotly to achieve this visualization.

Provide the code below:

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

diamonds = pd.read_csv('diamonds.csv')

data = sns.FacetGrid(diamonds, col="cut", hue="cut", col_wrap=3,
height=4, palette="Set2")

data.map(sns.stripplot, "carat", "price", jitter=True)

data.set_axis_labels("Carat", "Price")
data.fig.subplots_adjust(top=0.9)
data.fig.suptitle('Diamond Price vs Carat by Cut Quality')

plt.show()
```

Provide the figure below:



2. [10%] What is a rug plot, and what function can be used in **`seaborn`** to generate one?

A rug plot is a type of visualization that displays individual data points along the axis of a plot, often used to complement other visualizations like histograms or density plots. Each observation is represented as a short vertical or horizontal tick mark, depending on the axis it's plotted on. Rug plots are useful for understanding the distribution of data points and identifying any clustering or gaps.

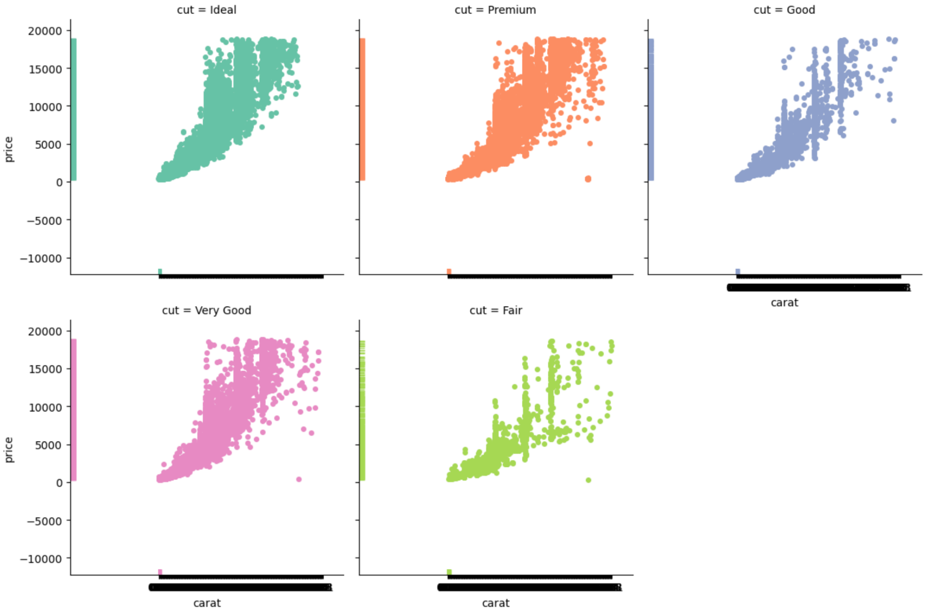
In Seaborn, the function used to generate a rug plot is sns.rugplot(). This function can be added to other plots to visualize the actual data points along with the overall distribution.

3. [20%] Add a rug plot to the figure from question 1.

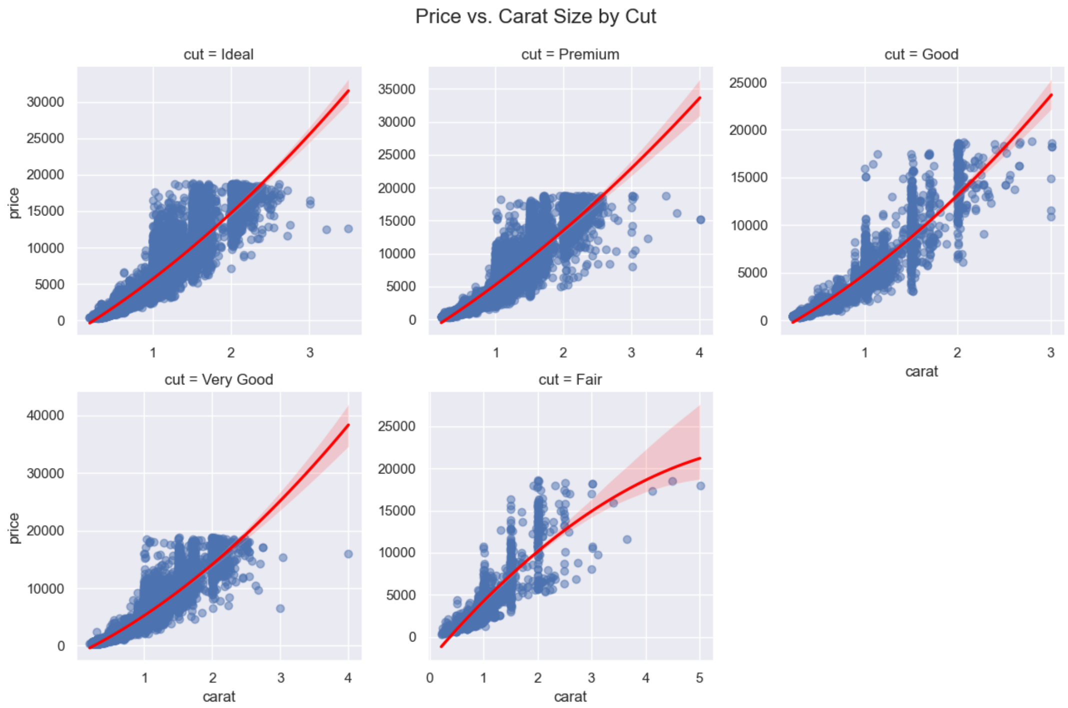
Provide the code below:

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

diamonds = pd.read_csv('diamonds.csv')

data = sns.FacetGrid(diamonds, col="cut", hue="cut", col_wrap=3,
height=4, palette="Set2")

data.map(sns.stripplot, "carat", "price", jitter=True)
data.map(sns.rugplot, "carat", "price", height=0.02)

plt.show()
```

Provide the figure below:



4. [20%] Provide the code to generate the following plot, where the bands around the fitted lines are 95% confidence intervals.



Provide code below:

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

diamonds = pd.read_csv('diamonds.csv')

plot   =   sns.lmplot(data=diamonds,   x="carat",   y="price",
col="cut", order=2, col_wrap=3, line_kws={"color":"red"})

for axis in plot.axes.flat:
    axis.set_facecolor('#f2f2f2')
    axis.grid(True,  which='both',  linestyle='-',  linewidth=0.5,
color='gray')

plot.set(xlabel="Carat", ylabel="Price")
plot.set_titles(col_template="{col_name}")

plt.subplots_adjust(top=0.9)
plot.fig.suptitle('Price vs Carat Size by Cut')

plt.show()
```

5. [10%] From your solution to question 4, does there appear to be a relationship between diamond price and diamond weight? If there is a relationship, then what is it?

Based on the analysis from question 4, it is clear that there is a significant relationship between diamond price and weight (carat). The scatter plots with fitted regression lines reveal that as the diamond's weight increases, the price generally increases as well, highlighting a positive correlation between these two factors.

This positive correlation is reinforced by the quadratic (second-order polynomial) regression applied to the data. The regression lines show an upward curve, particularly noticeable for diamonds with larger carat weights. This suggests that the price does not rise in a straight line but accelerates as the carat size increases. In other words, for larger diamonds, even a small increase in weight results in a more substantial price increase. This trend aligns with the common understanding of diamond pricing, where larger diamonds are valued more highly per carat compared to smaller ones.

Additionally, the relationship between weight and price seems to vary depending on the quality of the diamond cut. In some cases, the price increase with weight is more pronounced, indicating that the cut quality impacts the price's responsiveness to weight changes. For instance, diamonds with superior cuts might show a stronger correlation between weight and price, while diamonds with lower-quality cuts might exhibit a more moderate price increase as the carat size grows.

In summary, there is a distinct and positive relationship between diamond price and weight, with prices rising at an accelerating rate as the carat size increases. This relationship is also influenced by the quality of the diamond's cut.

6. [10%] In the plot from question 4, why are the confidence intervals much narrower for diamonds weighing less than three carats than for diamonds weighing greater than three carats?

In the plot from question 4, the confidence intervals for diamonds weighing less than three carats are noticeably narrower compared to those for diamonds weighing more than three carats. This difference is primarily due to variations in data distribution and volume across these weight categories.

Key Factors:

Data Density: The dataset has a higher concentration of diamonds weighing less than three carats. With more data points in this weight range, the model can more accurately estimate the relationship between weight and price. This dense data leads to more reliable estimates and, consequently, narrower confidence intervals.

Sparse Data for Larger Carats: Diamonds weighing over three carats are less common. The limited number of observations in this weight range results in less precise estimates of the weight-price relationship. With fewer data points, the regression model faces greater uncertainty, leading to wider confidence intervals. Essentially, the model has less information to make accurate predictions for larger diamonds, causing the intervals to broaden.

Increased Price Variability for Larger Carats: As the weight of diamonds increases, the price variability tends to rise. For larger diamonds, small differences in attributes (such as cut, clarity, or color) can lead to significant price fluctuations. This increased variability contributes to wider confidence intervals for higher carat weights.

Summary:

In conclusion, the narrower confidence intervals for diamonds weighing less than three carats are due to the higher concentration of data and lower price variability in this range, which allows for more precise estimates. In contrast, the wider confidence intervals for diamonds over three carats result from the scarcity of data and greater price variability, leading to less precise predictions.

7. [10%] What is a violin plot, and what function in **`seaborn`** can be used to generate one?

A violin plot is a type of chart that helps visualize the distribution of data across different categories. It combines features of both box plots and Kernel Density Estimates (KDE), offering a comprehensive view of both summary statistics and data density.

What a Violin Plot Shows:

Data Density: The plot displays the density of the data on each side, which is similar to a smoothed histogram. This helps you understand the shape of the distribution, whether it's skewed, uniform, or has multiple peaks.

Summary Statistics: It also includes key summary statistics, such as the median (represented by a white dot) and the interquartile range (shown as a black bar inside the violin shape).

Why Use a Violin Plot?

Violin plots are particularly useful when you want to see how data is distributed and also get an overview of summary statistics all in one chart. They are great for comparing the distributions across different categories, giving you a clear picture of variations and trends.

Creating a Violin Plot in Seaborn:

To create a violin plot in Seaborn, you use the sns.violinplot() function. This function generates the visual representation of data distributions and helps in making comparisons between different categories.

8. [10%] How are violin plots different from box plots?

Violin plots and box plots are both used to visualize the distribution of data, but they differ in how much information they convey and how they represent the distribution.

Key Differences:

Shape of Distribution:

Box Plot: Summarizes the distribution with five key statistics: minimum, first quartile (Q1), median, third quartile (Q3), and maximum. It doesn’t show the detailed shape of the data distribution; it only provides a snapshot through these summary stats.

Violin Plot: In addition to summary statistics, it shows the full distribution shape using a kernel density estimate (KDE), which smooths out the data to create a continuous curve. This allows you to see if the data is skewed, bimodal, or has outliers, giving a deeper understanding of the distribution.

Density Estimate:

Box Plot: Does not show density. You can’t see how common different values are or if there are multiple peaks in the data.

Violin Plot: Shows the density of the data on each side of the plot. The wider the plot, the more data points fall in that range. This helps to understand the concentration of data across the range.

Details about Outliers:
Box Plot: Explicitly marks outliers as individual points outside the “whiskers” (which typically extend 1.5 times the interquartile range).

Violin Plot: Doesn’t emphasize individual outliers as clearly as a box plot. However, the shape of the violin may suggest where extreme values are, even if they aren’t individually marked.

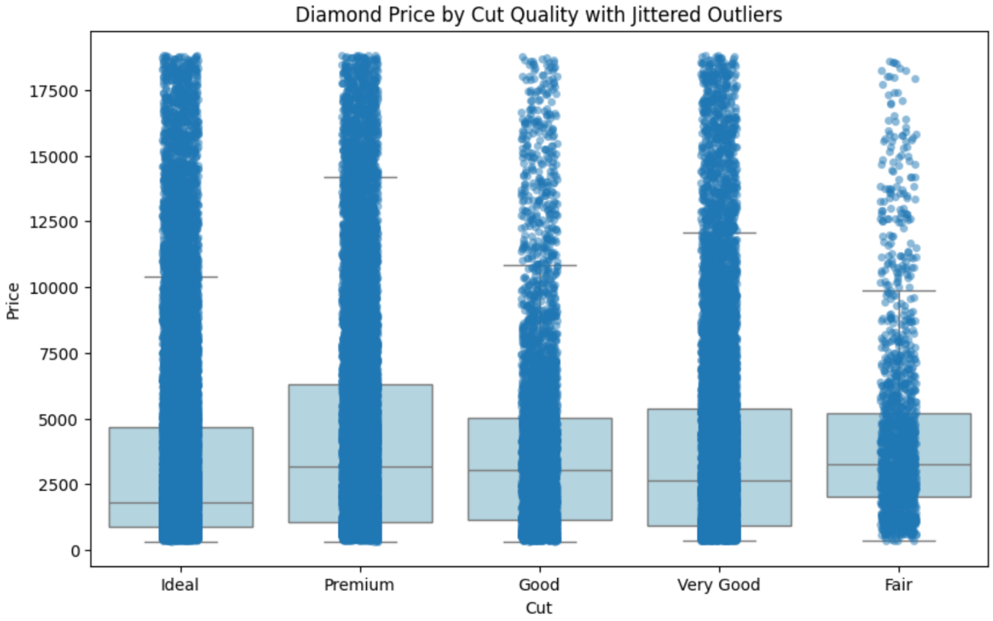
Visual Complexity:
Box Plot: Simpler and quicker to interpret, especially for a quick comparison of medians and spread.

Violin Plot: More detailed, allowing for richer interpretation but can be more complex to read if you are only interested in summary statistics.

9. [30%] In Seaborn boxplots, outliers are typically shown as individual points. However, when there are multiple outliers in the same category, they may overlap, obscuring important details. How can we address the issue of overlapping outliers in Seaborn boxplots to improve clarity? Consider using an additional Seaborn plot that adds a jitter layer on top of the boxplot to enhance the visualization of individual data points. Which specific function and parameters would you use to achieve this, and how would you incorporate it into the existing boxplot code provided below?

```
sns.boxplot(x="cut", y="price", data=diamonds)
```

Provide the cod and figure below:



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

diamonds = pd.read_csv('diamonds.csv')

plt.figure(figsize=(10, 6))
sns.boxplot(x="cut",            y="price",            data=diamonds,
color='lightblue', fliersize=0)

sns.stripplot(x="cut",   y="price",   data=diamonds,   alpha=0.5,
jitter=True)
```

```
plt.title('Diamond Price by Cut Quality with Jittered Outliers')
plt.xlabel('Cut')
plt.ylabel('Price')

plt.show()
```

Based on the figure, what is the purpose of the following using `jitter` here, since the *x*-axis is categorical?

Provide your answer below:

Improving Visibility: Jitter adds a small amount of random noise to the position of data points along the x-axis. This prevents points from overlapping exactly on top of each other, which is particularly useful when there are many outliers in the same category. By spreading out the points slightly, jitter helps to reveal the distribution and density of data more clearly.

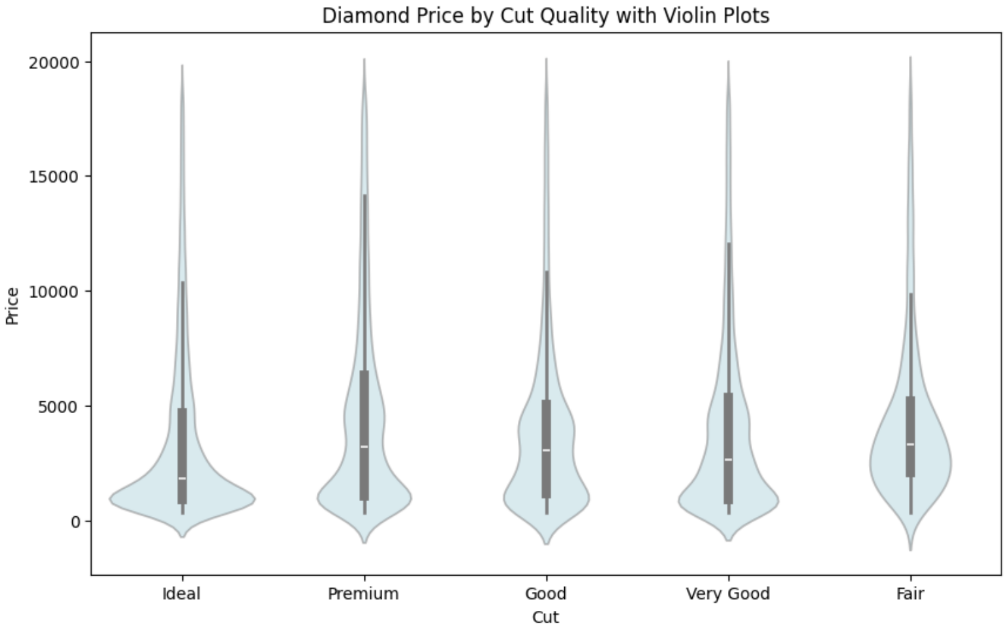
Enhancing Clarity: When dealing with categorical x-axes, jitter helps to show the spread of individual data points within each category. Without jitter, overlapping points might be obscured, making it difficult to see how many outliers there are or where they are concentrated. Jitter ensures that each point is visible and provides a better sense of the data's distribution.

10. [20%] Replace the box plots with violin plots in your figure from question 9, giving them the same level of transparency as the box plots without adding a `jitter` layer.

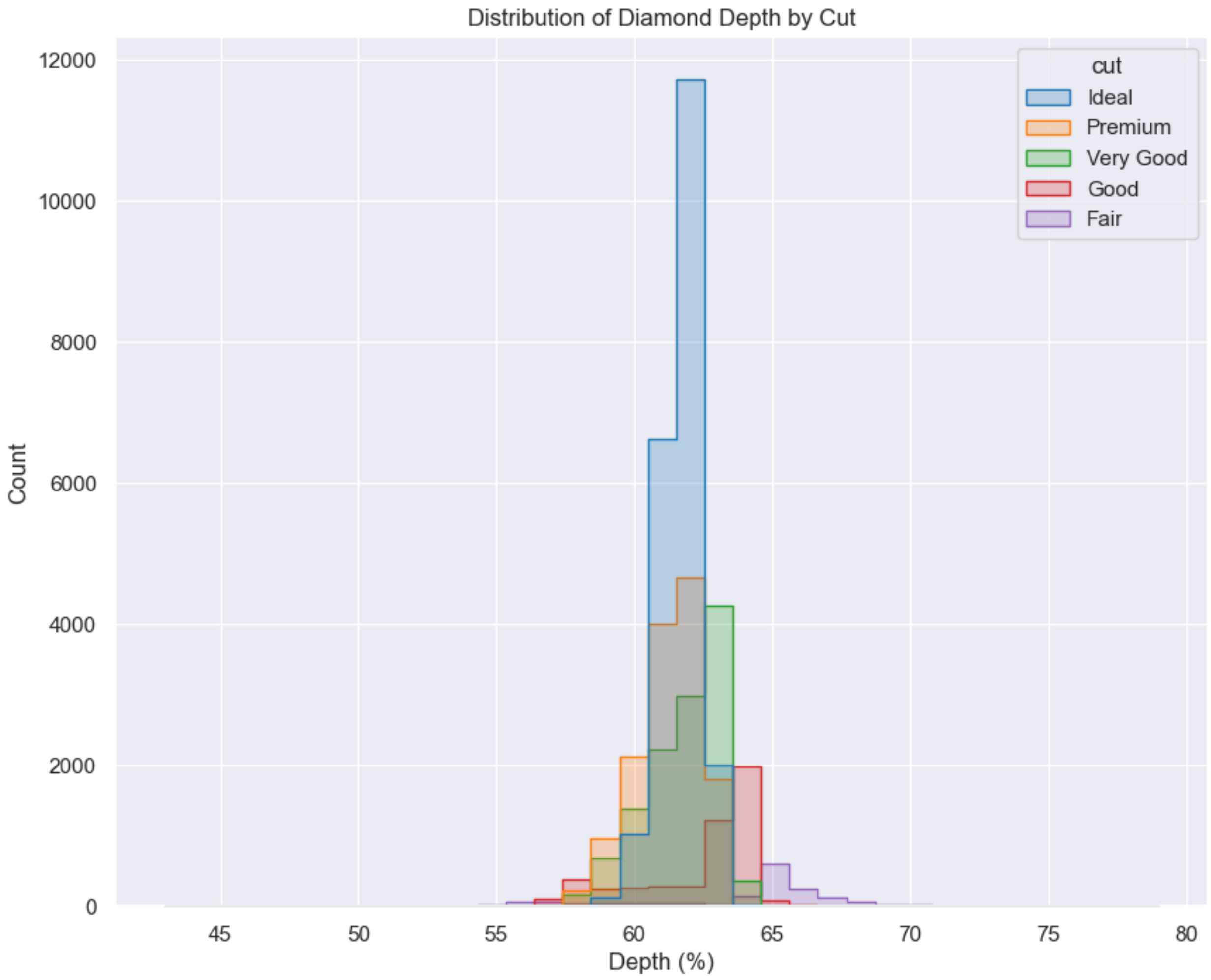
Provide the code below:

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

diamonds = pd.read_csv('diamonds.csv')

plt.figure(figsize=(10, 6))

sns.violinplot(x="cut", y="price", data=diamonds, color='lightblue', alpha=0.5)

plt.title('Diamond Price by Cut Quality with Violin Plots')
plt.xlabel('Cut')
plt.ylabel('Price')

plt.show()
```

Provide the figure below:



11. [20%] Provide the code to generate the following plot.



Provide code below:

```
diamonds = pd.read_csv('diamonds.csv')

sns.histplot(diamonds, x="depth", hue="cut", element="step",
binwidth=1)

plt.gca().set_facecolor('#f2f2f2')

plt.xlabel("Depth%")

plt.title("Distribution of Diamond Depth by Cut")

plt.grid(True, which='both', linestyle='-', linewidth=0.5,
color='white')

plt.show()
```

12. [20%] Data visualization is often described as a bridge between data and decision-making.

a. Discuss the role of visualization in the data science pipeline.

Data visualization plays a crucial role in the data science pipeline by making complex data more accessible and understandable. Here’s how:

Data Exploration: Visualization is often the first step in understanding data. Tools like scatter plots, histograms, and box plots allow data scientists to explore and detect patterns, outliers, and trends in the data. This helps in forming hypotheses and guiding further analysis.

Data Cleaning: When dealing with messy or incomplete data, visualizations can highlight missing values, anomalies, or inconsistencies. For example, a heatmap can show where data is missing or irregularly distributed, making it easier to address these issues.

Model Building: During model building, visualizations such as feature importance plots and learning curves help in evaluating model performance and understanding the impact of different variables. This assists in refining models and improving accuracy.

Result Interpretation: After building models, visualizations like ROC curves and confusion matrices provide insights into model performance. They help in interpreting results and making sense of the model’s predictions.

Communication: Visualizations are essential for communicating findings to stakeholders. They translate complex results into intuitive graphics, making it easier to share insights and inform decision-making.

b. How does visualization help improve understanding, communication, and decision-making processes?

Improving Understanding:

Clarity: Visualizations simplify complex data, making it easier to grasp the underlying patterns and relationships. For instance, a line graph can clearly show trends over time, which might be obscured in a table of numbers.

Pattern Recognition: By presenting data graphically, it’s easier to spot trends, correlations, and anomalies. For example, a scatter plot can reveal relationships between two variables that might not be obvious from raw data.

Enhancing Communication:

Accessibility: Visualizations make data accessible to non-technical stakeholders. A well-designed chart can convey key insights quickly, helping stakeholders understand and act on the information.
Engagement: Interactive visualizations, such as dashboards, allow users to explore data dynamically, engaging them in the analysis process and improving their understanding.

Supporting Decision-Making:

Informed Decisions: Visualizations provide a clear view of data trends and patterns, which supports better decision-making. For example, a bar chart comparing sales across different regions can help in strategic planning and resource allocation.

Scenario Analysis: Visualizations can help in comparing different scenarios or forecasting future trends, aiding in planning and risk assessment.

c. Provide examples where visualizing data can significantly change the way insights are perceived compared to statistical summaries alone.

COVID-19 Pandemic:

Statistical Summary: Raw numbers of cases and deaths may not convey the full story.
Visualization: Interactive maps and time series charts showed the spread and trends of the virus more effectively, allowing for better understanding and response.

Economic Data:

Statistical Summary: Tables of economic indicators like GDP growth and unemployment rates.
Visualization: Line graphs and bar charts can illustrate economic trends and comparisons across countries or time periods, making it easier to grasp economic health and performance.

Customer Feedback:

Statistical Summary: Average satisfaction scores from survey data.
Visualization: Heatmaps and word clouds can reveal patterns in customer feedback, highlighting areas of concern or satisfaction more clearly than summary statistics.

d. What are some common ways that visualizations can be misleading

Misleading Scales:

Issue: Using a non-zero baseline or manipulated scales can exaggerate or minimize differences.
Example: A bar chart with a truncated y-axis can make small differences look larger than they are.

Cherry-Picking Data:

Issue: Selecting specific data points or periods to support a particular narrative can distort the overall picture.
Example: Showing only a recent uptrend in stock prices while ignoring long-term performance trends.

Overcomplicated Visualizations:
Issue: Using too many variables or complex chart types can confuse rather than clarify.
Example: A 3D plot with multiple dimensions can be difficult to interpret, leading to potential misinterpretation.