Data visualizationLab

Bibek Sapkota

Data Visualization (Week-2 Lab)

Task: loading ggplot2 amd dplyr packages

```
library(ggplot2)
library(dplyr)
```

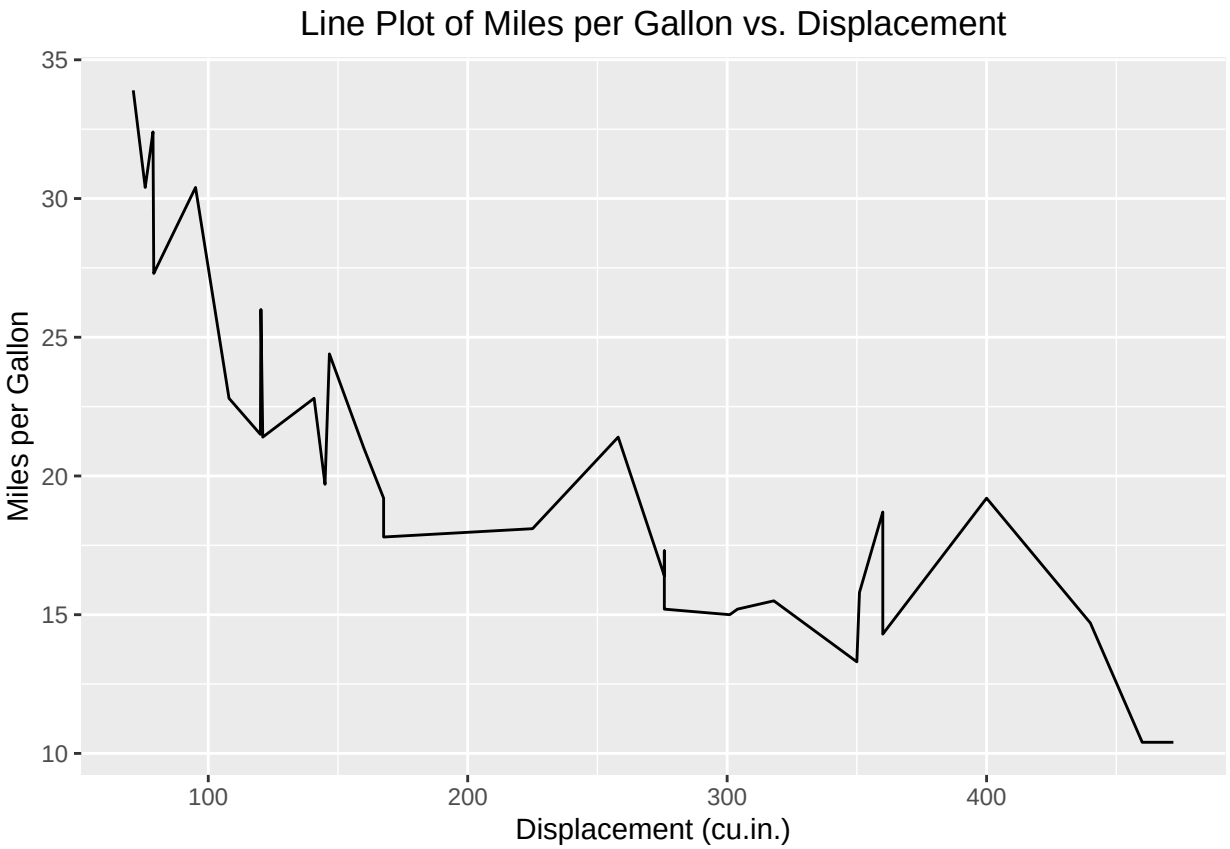
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Question-1: Create a simple line plot using ggplot2 with mpg (miles per gallon) on the y-axis and disp (displacement) on the x-axis from the 'mtcars' dataset.

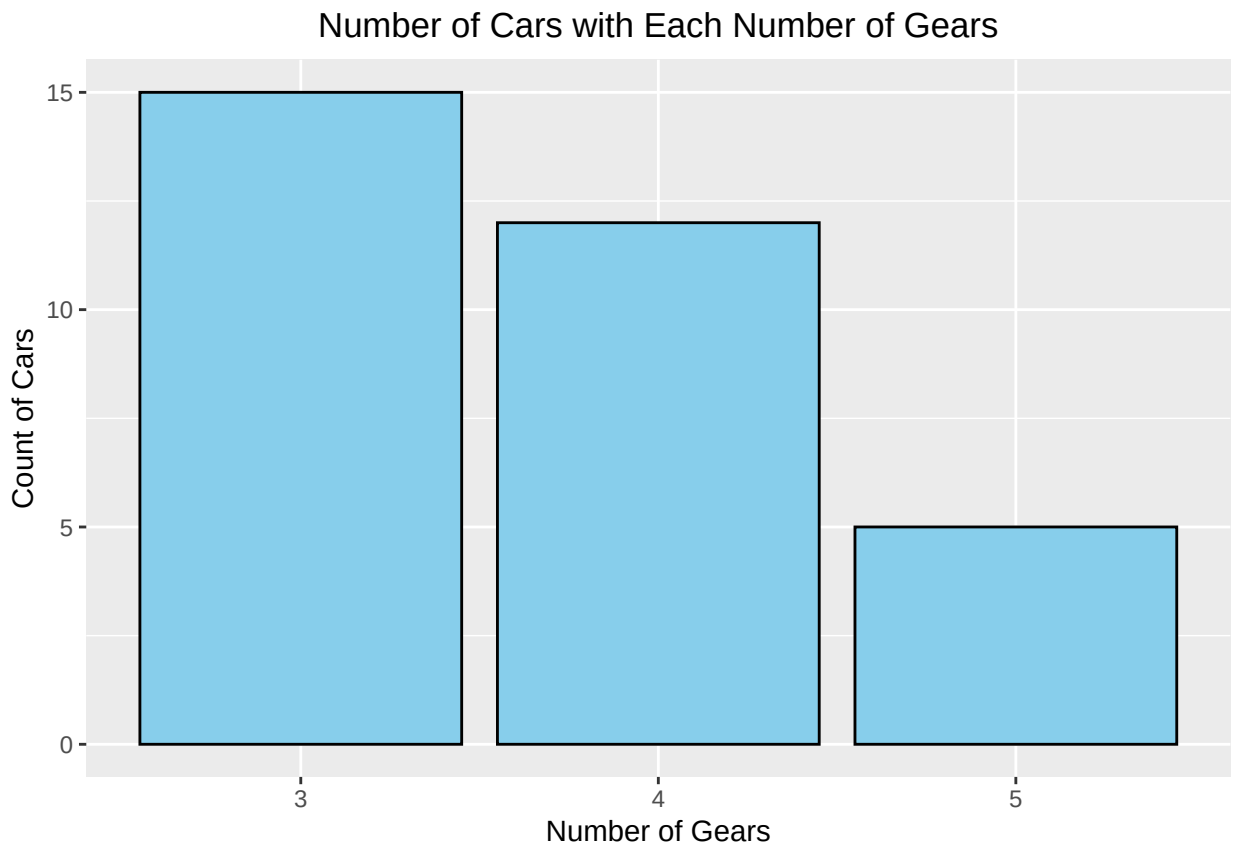
```
data(mtcars) #loading the required mtcars datset

ggplot(mtcars, aes(x = disp, y = mpg)) +
  geom_line() +
  labs(
    title = "Line Plot of Miles per Gallon vs. Displacement",
    x = "Displacement (cu.in.)",
    y = "Miles per Gallon"
  ) +
  theme(
    plot.title = element_text(hjust = 0.5)
  )
```



Interpretation: Increasing engine displacement generally leads to lower miles per gallon due to a negative correlation. However, some instances deviate from this pattern, showing inconsistent relationships between displacement and fuel efficiency.

Question-2: Using the 'mtcars' dataset, create a bar plot showing the number of cars with each number of gears (gear).

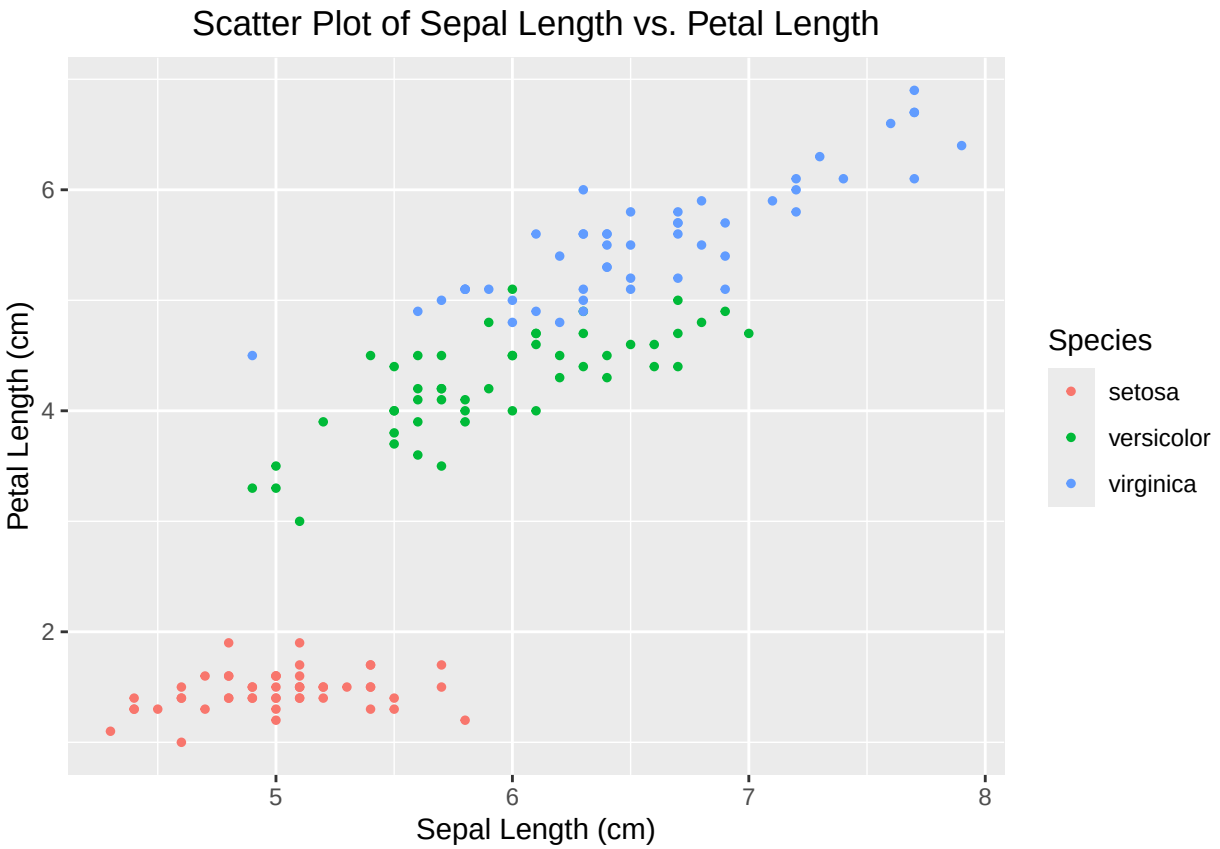
```
ggplot(mtcars, aes(x = factor(gear))) +
  geom_bar(fill = "skyblue", color = "black") +
  labs(
    title = "Number of Cars with Each Number of Gears",
    x = "Number of Gears",
    y = "Count of Cars"
  ) +
  theme(
    plot.title = element_text(hjust = 0.5)
  )
```



Interpretation: The bar plot illustrates how cars are distributed according to the number of gears. It reveals that 3 gears are the most prevalent, followed by 4 gears, while cars with 5 gears are the least frequently encountered.

Question-3: Create a scatter plot using the iris dataset, mapping Sepal.Length to x, Petal.Length to y, and Species to color.

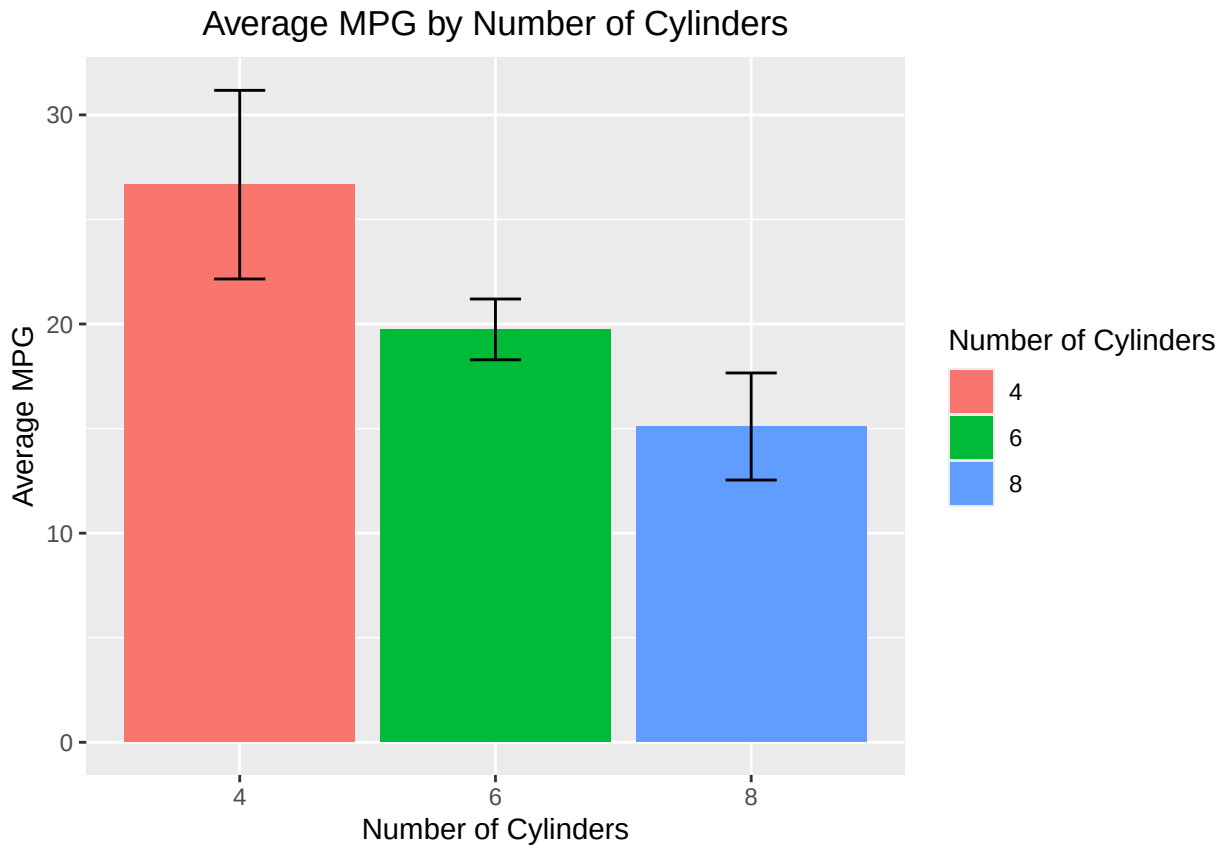
```
data("iris") #loading the required iris datset

ggplot(iris, aes(x = Sepal.Length, y = Petal.Length, color = Species)) +
  geom_point(size = 1) +
  labs(
    title = "Scatter Plot of Sepal Length vs. Petal Length",
    x = "Sepal Length (cm)",
    y = "Petal Length (cm)"
  ) +

  theme(
    plot.title = element_text(hjust = 0.5)
  )
```



Interpretation:: Sepal length shows a positive correlation with petal length, implying that as sepal length increases, petal length generally increases. However, the relationship isn't perfect, with noticeable scatter in the data. Iris setosa typically has shorter sepals and petals compared to Iris versicolor and Iris virginica, which generally exhibit longer sepals and petals with more variability.

Question-4: Create a bar plot showing the average mpg (miles per gallon) for each 'cyl' (number of cylinders) in the 'mtcars' dataset. Color bars based on 'cyl' and add error bars to represent the standard deviation

```
mpg_summary <- mtcars %>%
  group_by(cyl) %>%
  summarise(mean_mpg = mean(mpg),
            sd_mpg = sd(mpg))

ggplot(mpg_summary, aes(x = factor(cyl), y = mean_mpg, fill = factor(cyl))) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_errorbar(aes(ymin = mean_mpg - sd_mpg, ymax = mean_mpg + sd_mpg),
                width = 0.2, position = position_dodge(width = 0.9)) +
  labs(x = "Number of Cylinders", y = "Average MPG", fill = "Number of Cylinders") +
  ggtitle("Average MPG by Number of Cylinders") +
  theme(
    plot.title = element_text(hjust = 0.5)
  )
```

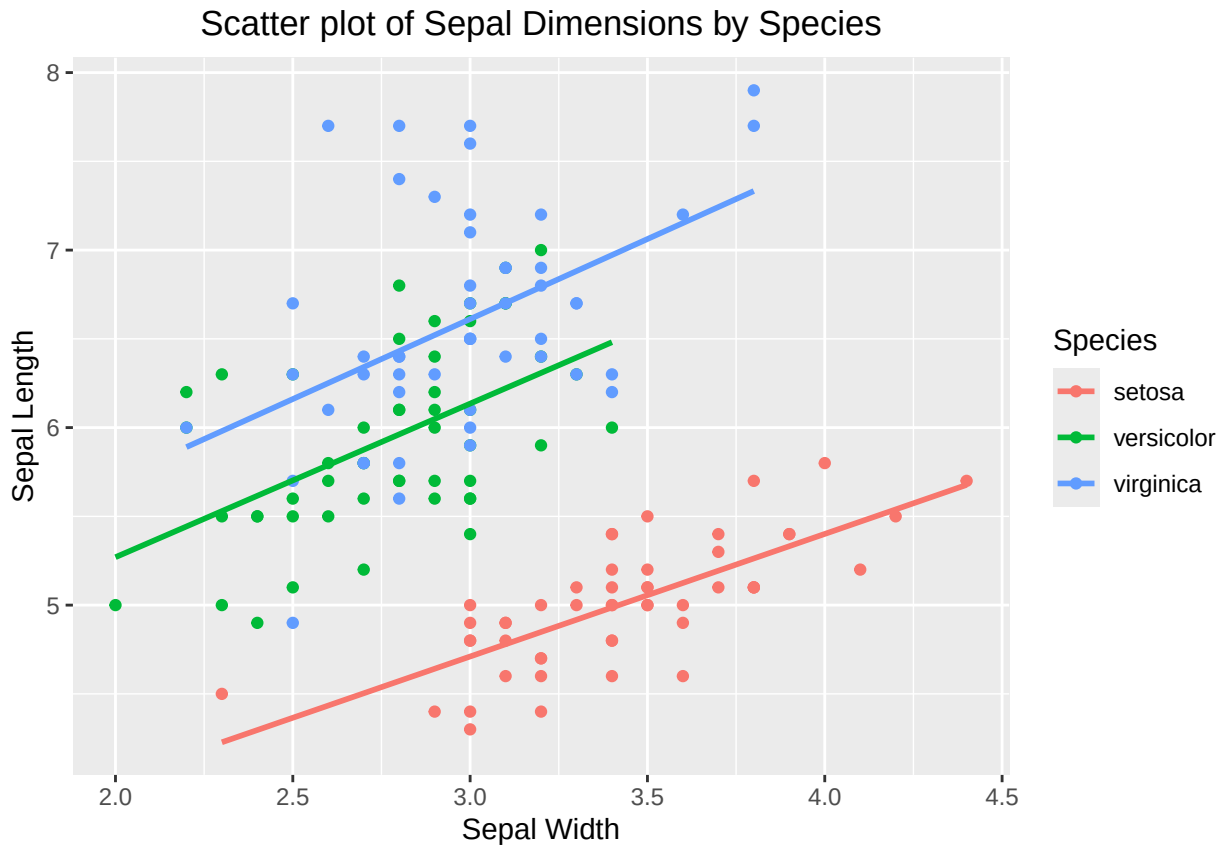


Interpretation: The chart illustrates that cars with fewer cylinders tend to have better mileage than those with more cylinders. This is because smaller engines in cars with fewer cylinders use less fuel. Four-cylinder cars achieve the highest mileage at around 30 MPG, while six-cylinder cars average about 20 MPG, and eight-cylinder cars have the lowest mileage at around 10 MPG.

Question-5: Using the iris dataset, create a scatter plot mapping Sepal.Width to x and Sepal.Length to y, colored by Species. Add a linear regression line for each species.

```
ggplot(iris, aes(x = Sepal.Width, y = Sepal.Length, color = Species)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Scatter plot of Sepal Dimensions by Species",
       x = "Sepal Width",
       y = "Sepal Length") +
 theme(
    plot.title = element_text(hjust = 0.5)
  )
```

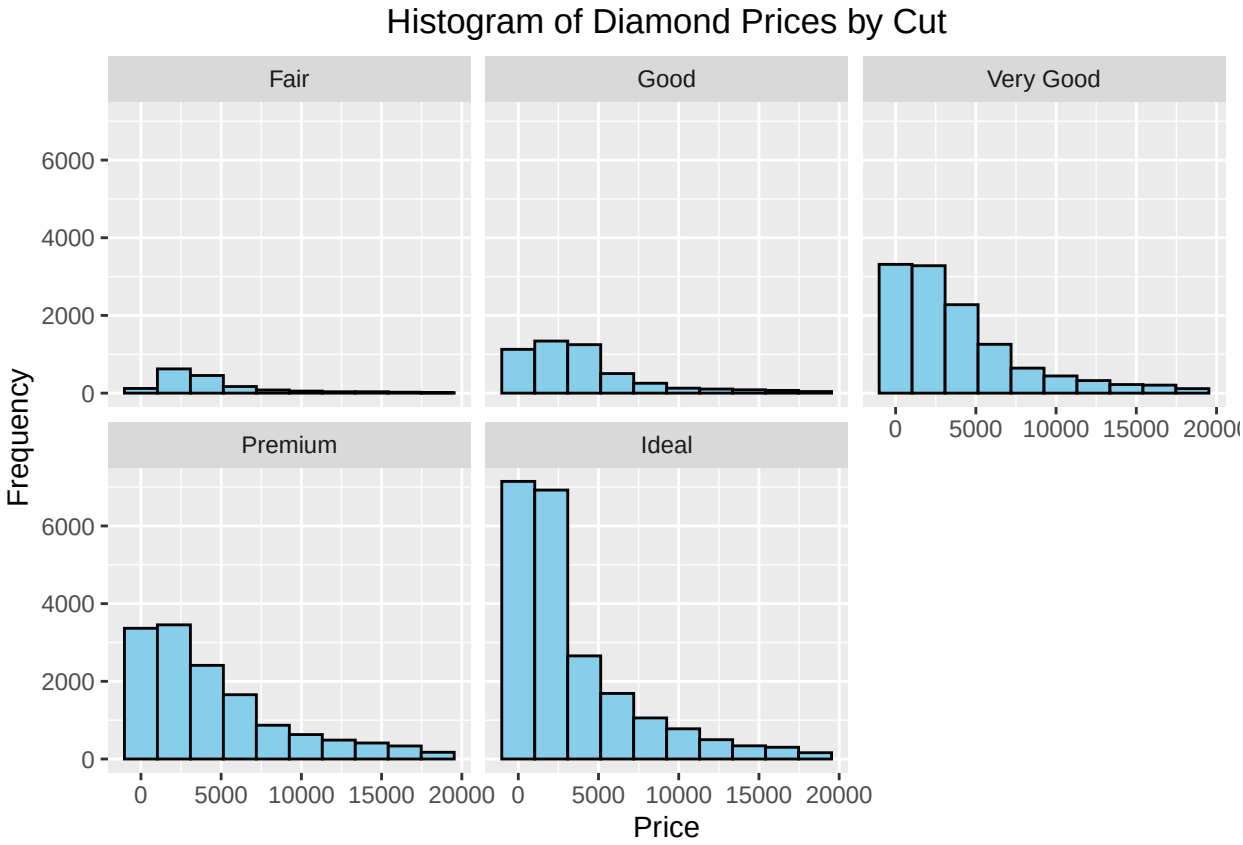
```
## `geom_smooth()` using formula = 'y ~ x'
```



Interpretation: The regression line for all three variables slopes positively, suggesting that as Sepal.Width increases, Sepal.Length tends to increase as well. Each species exhibits a positive correlation between these variables, but the strength and range vary. Setosa generally has the shortest Sepal.Length, versicolor intermediate, and virginica the longest.

Question-6: Using the diamonds dataset, create faceted histograms for price, split by cut. Adjust the bins so they are suitable for the data's distribution.

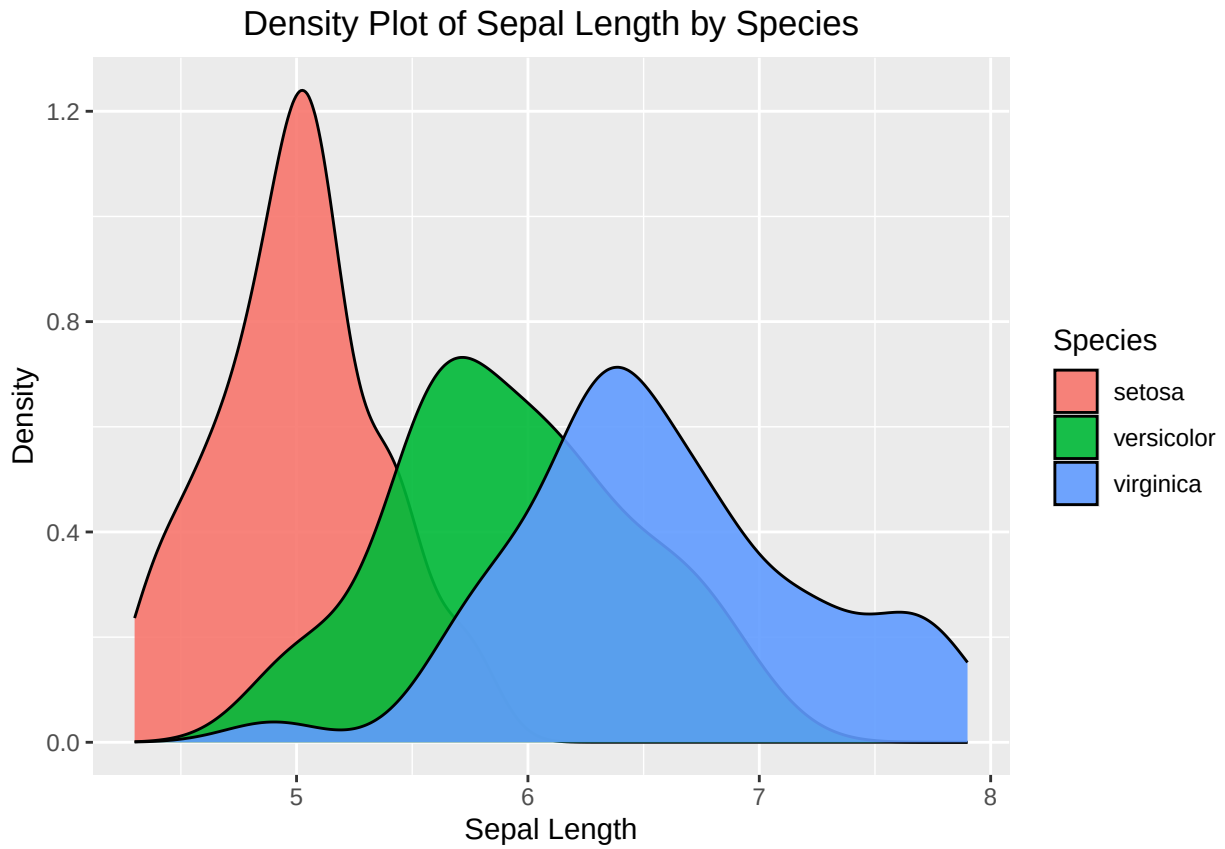
```
ggplot(diamonds, aes(x = price)) +
  geom_histogram(bins = 10, fill = "skyblue", color = "black") +  # Adjust bins as needed
  facet_wrap(~ cut) +
  labs(title = "Histogram of Diamond Prices by Cut",
       x = "Price",
       y = "Frequency") +
  theme(
    plot.title = element_text(hjust = 0.5)
  )
```



Interpretation: The histogram illustrates varying price distributions across different diamond cuts, revealing a clear relationship between diamond cut and price. Diamonds with superior cut grades (Premium, Ideal) exhibit a broader range of prices, potentially higher than those with lower grades (Fair, Good, Very Good).

Question-7: Create a density plot for Sepal.Length from the iris dataset, fill based on Species, and apply a different color palette.

```
ggplot(iris, aes(x = Sepal.Length, fill = Species)) +
  geom_density(alpha = 0.9) +
  labs(title = "Density Plot of Sepal Length by Species",
       x = "Sepal Length",
       y = "Density") +
  theme(
    plot.title = element_text(hjust = 0.5)
  )
```



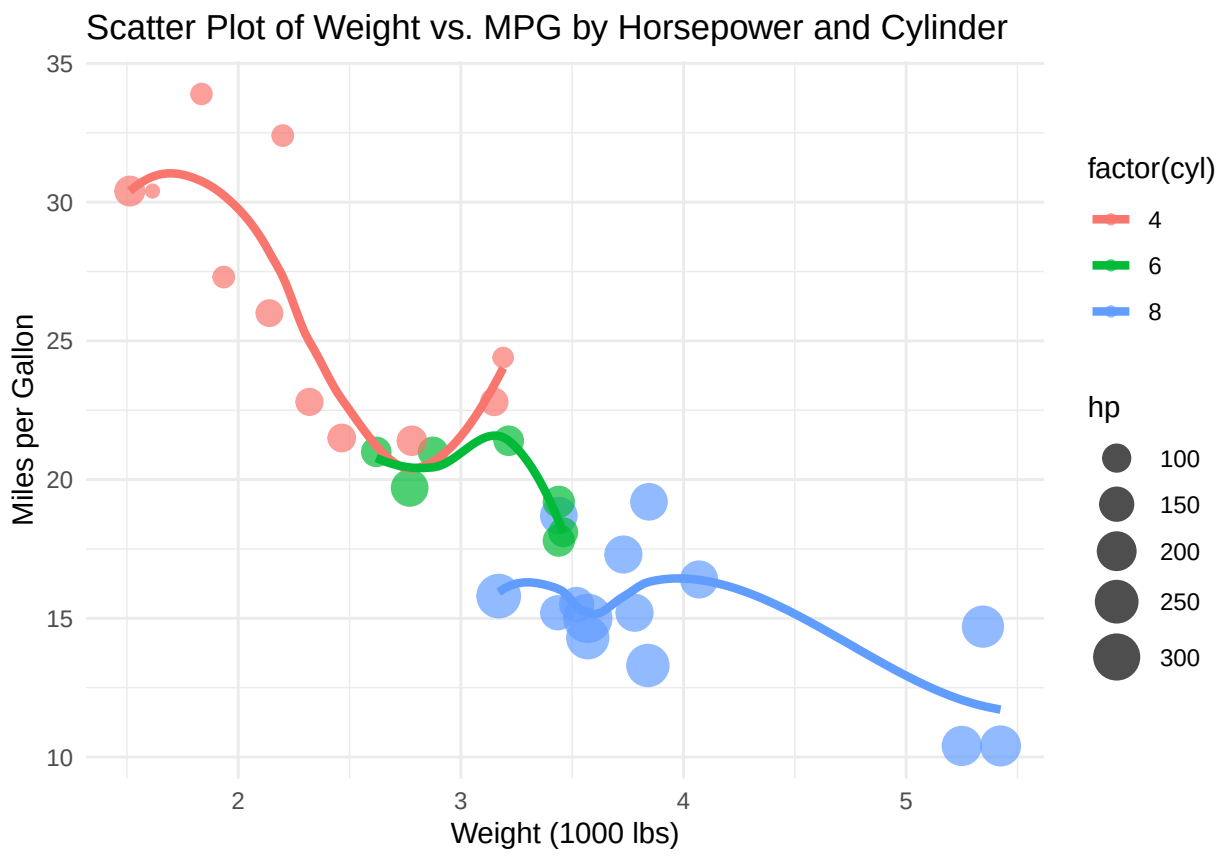
Interpretation: The density plot shows that Setosa typically has the smallest sepals, Versicolor has intermediate sepal lengths, and Virginica has the longest sepals on average. There may be some overlap in sepal length distribution between Versicolor and Setosa, but there is generally minimal overlap between Versicolor and Virginica.

Question-8: Using the 'mtcars' dataset, create a scatter plot of wt (weight) vs. mpg (miles per gallon), size the points by hp (horsepower), and color them by 'cyl' (cylinders). Add smooth lines to show the trend for each cylinder group.

```
ggplot(mtcars, aes(x = wt, y = mpg, size = hp, color = factor(cyl))) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "loess", se = FALSE, aes(group = cyl), size = 1.5) +
  scale_size_continuous(range = c(2, 8)) +
  labs(x = "Weight (1000 lbs)", y = "Miles per Gallon") +
  ggtitle("Scatter Plot of Weight vs. MPG by Horsepower and Cylinder") +
  theme_minimal() +
  theme(
    legend.position = "right",
  )
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

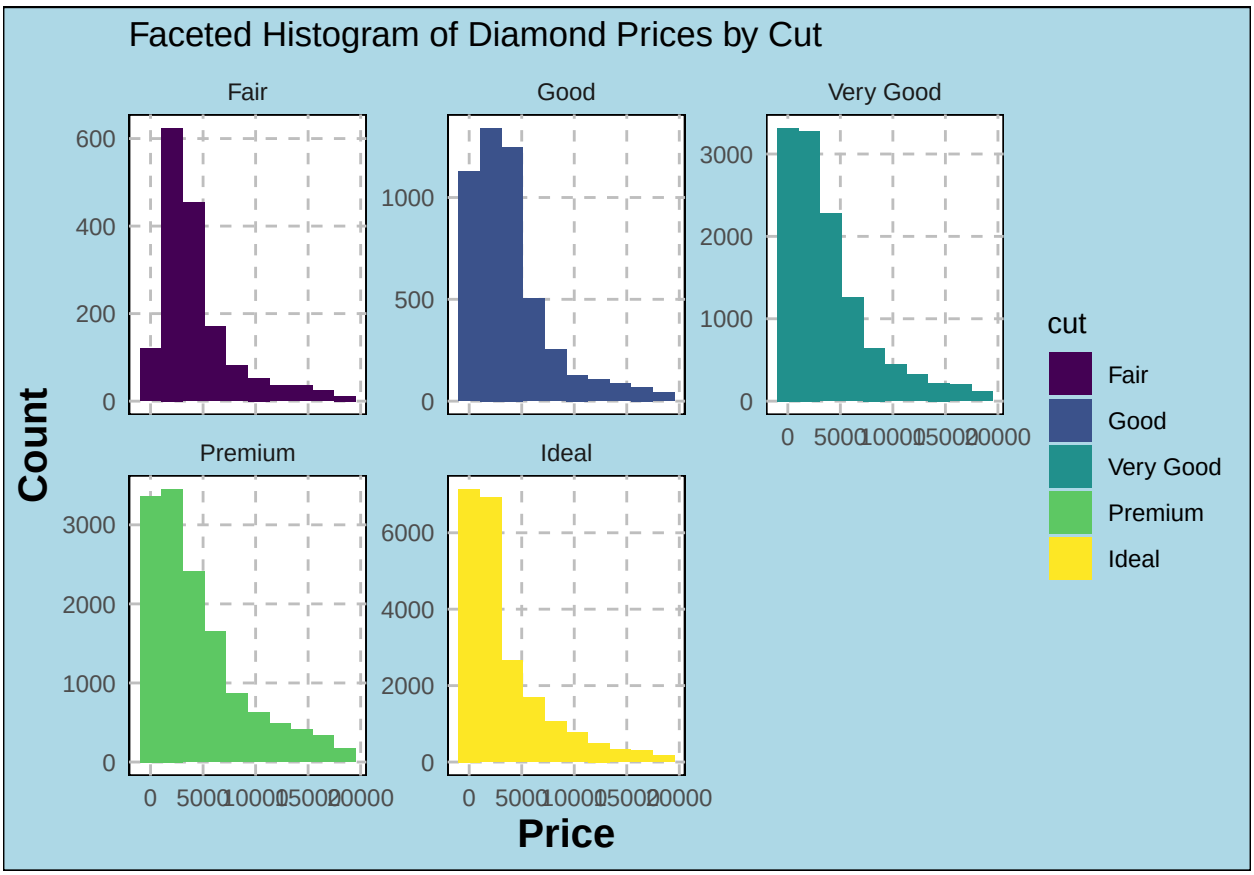


Interpretation: The scatter plot indicates a negative correlation between MPG and weight, meaning as car weight increases, fuel efficiency decreases. It also shows that both weight and horsepower influence vehicle fuel efficiency, with lighter and lower horsepower vehicles generally achieving higher MPG.

Question-9: Create any plot of your choice using the ggplot2 package and then customize it using a theme() function. Modify at least five different aspects of the theme (e.g., text size, background color, grid lines).

```r
# Create the basic plot
p <- ggplot(diamonds, aes(x = price, fill = cut)) +
  geom_histogram(bins = 10) +
  facet_wrap(~ cut, scales = "free_y") +
  labs(x = "Price", y = "Count") +
  ggtitle("Faceted Histogram of Diamond Prices by Cut") +

  # Customize the plot using theme()
  theme_minimal() +
  theme(
    plot.background = element_rect(fill = "lightblue"),
    panel.background = element_rect(fill = "white"),
    panel.grid.major = element_line(color = "gray", linetype = "dashed"),
    panel.grid.minor = element_blank(),
```

```
    axis.title = element_text(size = 15, face = "bold"),
    legend.position = "right"
  )

# Print the plot
print(p)
```



Question-10: How can you create your custom theme in ggplot2?Carry out the following steps and analyze the output.

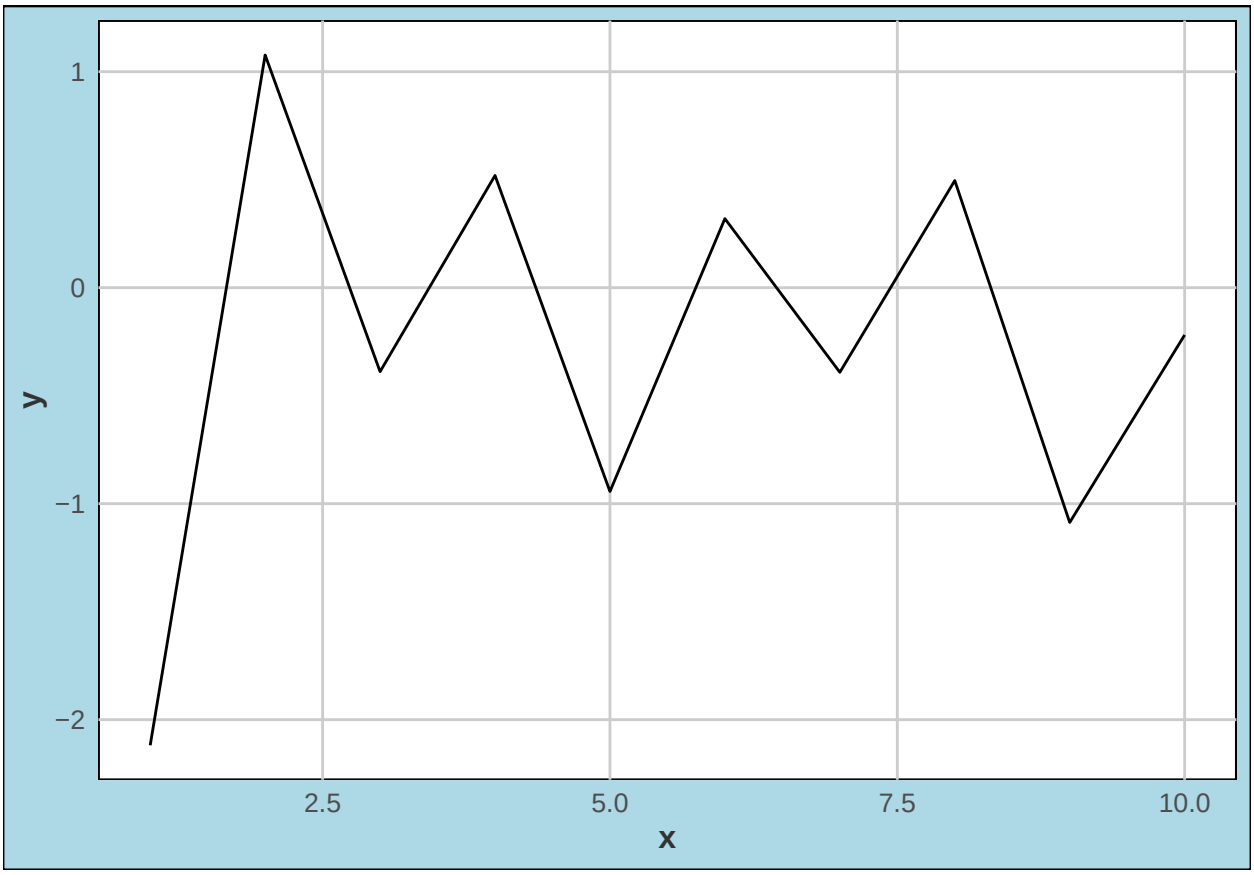
A. DefinetheCustomTheme:

```
my_custom_theme <- function() {
theme_minimal() + # Start with a minimal theme
  theme(
    text = element_text(family = "Helvetica", color ="#333333"),
    plot.background = element_rect(fill = "lightblue"),
    panel.background = element_rect(fill = "white"),
    panel.grid.major = element_line(color = "grey80"),
    panel.grid.minor = element_blank(),
    axis.title = element_text(size = 12, face = "bold"),
    axis.text = element_text(size = 10),
    legend.background = element_rect(fill = "white"),
```

```
    legend.title = element_text(face = "bold"),
    legend.text = element_text(size = 9)
  )
}
```

B. ApplytheCustomTheme:

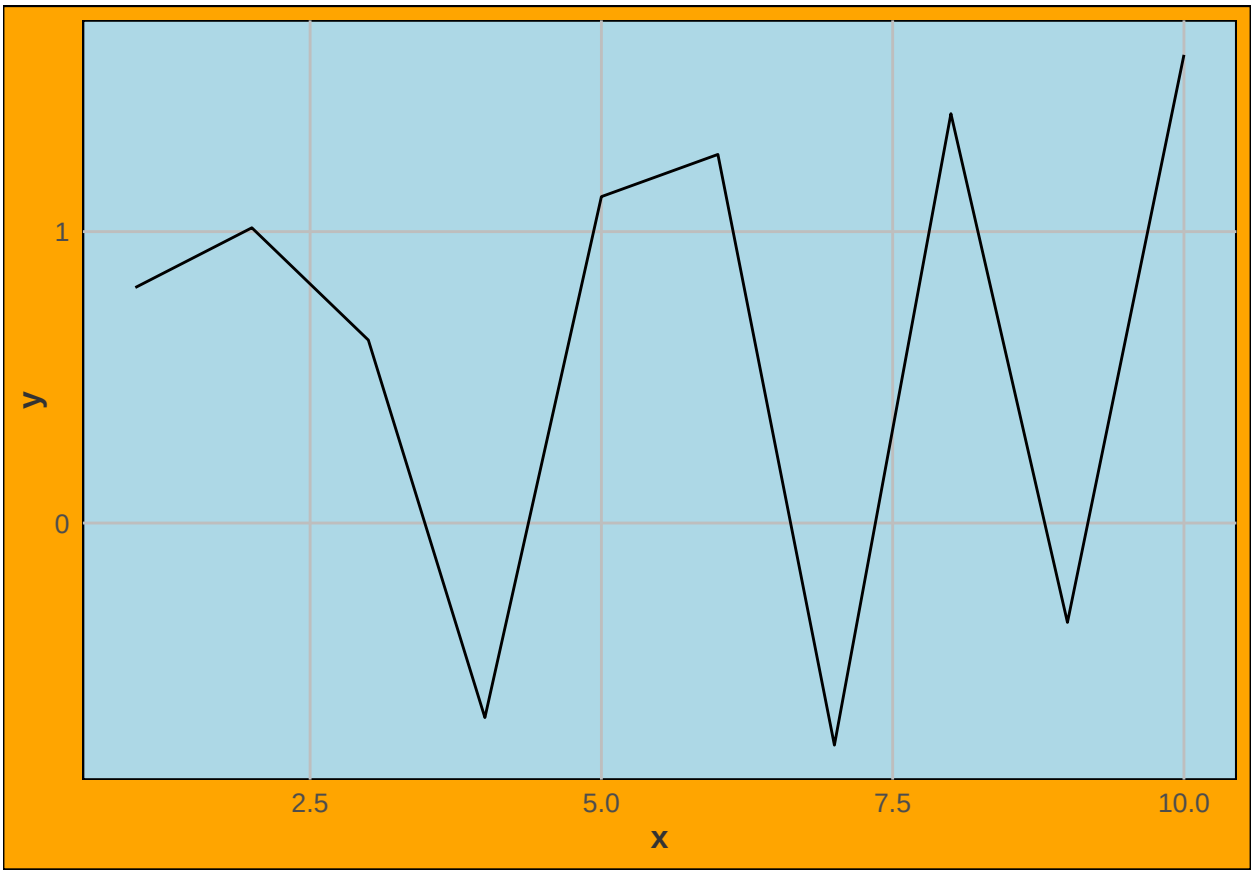
```
data <- data.frame(
  x = 1:10,
  y = rnorm(10)
)

ggplot(data, aes(x, y)) +
  geom_line() +
  my_custom_theme()  # Apply your custom theme
```



C. ModifyandReuseYourTheme:

```
my_custom_theme <- function() {
  theme_minimal() +
    theme(
      text = element_text(family = "Helvetica", color = "#333333"),
      plot.background = element_rect(fill = "orange"),
      panel.background = element_rect(fill = "lightblue"),
```

```
      panel.grid.major = element_line(color = "grey"),
      panel.grid.minor = element_blank(),
      axis.title = element_text(size = 12, face = "bold"),
      axis.text = element_text(size = 10),
      legend.background = element_rect(fill = "white"),
      legend.title = element_text(face = "bold"),
      legend.text = element_text(size = 9)
    )
}

data <- data.frame(
  x = 1:10,
  y = rnorm(10)
)

ggplot(data, aes(x, y)) +
  geom_line() +
  my_custom_theme()  # Apply your custom theme
```



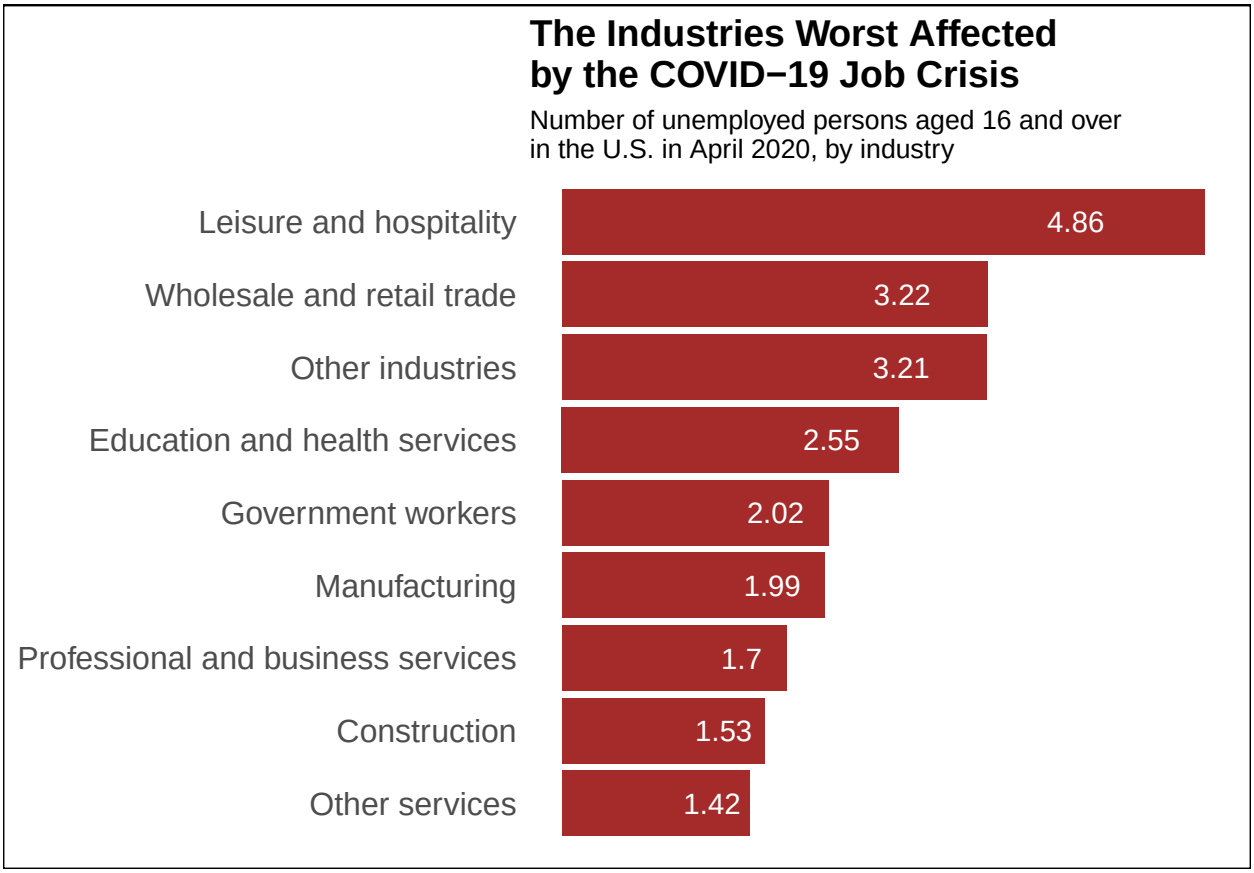
Question-10:

Dataset

```
# Define the vectors
industries <- c("Leisure and hospitality", "Wholesale and retail trade", "Other industries", "Education
                "Government workers", "Manufacturing",
                "Professional and business services", "Construction", "Other services")
unemployed_numbers <- c(4.86, 3.22, 3.21, 2.55, 2.02, 1.99, 1.70, 1.53, 1.42) # in million
unemployment_rates <- c(0.39, 0.17, 0, 0.11, 0.09, 0.13, 0.10, 0.17, 0.23)  # Convert percentages to pr

# Create the data frame
job_crisis_data <- data.frame(Industry = industries, UnemployedNumbers = unemployed_numbers,
                              UnemploymentRate = unemployment_rates)

# Print the data frame
print(job_crisis_data)
```

```
##                              Industry UnemployedNumbers UnemploymentRate
## 1            Leisure and hospitality              4.86              0.39
## 2         Wholesale and retail trade              3.22              0.17
## 3                   Other industries              3.21              0.00
## 4      Education and health services              2.55              0.11
## 5                 Government workers              2.02              0.09
## 6                      Manufacturing              1.99              0.13
## 7 Professional and business services              1.70              0.10
## 8                       Construction              1.53              0.17
## 9                     Other services              1.42              0.23
```

```
ggplot(job_crisis_data, aes(x = reorder(Industry, UnemployedNumbers), y = UnemployedNumbers)) +
  geom_bar(stat = "identity", fill = "brown") +
  coord_flip() +
  geom_text(aes(label = UnemployedNumbers),
            position = position_stack(vjust = 0.8),
            hjust = 0.5, color = "white", size = 4) +
  labs(title = "The Industries Worst Affected
by the COVID-19 Job Crisis",
       subtitle = "Number of unemployed persons aged 16 and over
in the U.S. in April 2020, by industry",
       x = NULL, y = NULL) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, face = "bold"),
    plot.subtitle = element_text(size = 10),
    axis.text.x = element_blank(),
    axis.text.y = element_text(size = 12),
    plot.background = element_rect(fill = "white"),
    panel.grid = element_blank()
  )
```



Interpretation: The graph shows that the leisure and hospitality sector was hardest hit by the COVID-19 job crisis, experiencing markedly higher unemployment rates than other industries. Wholesale and retail trade, alongside various other sectors, also saw significant job losses. Other industries, while impacted, reported relatively fewer unemployed individuals.