Homework1.Rmd

2024-10-10

Packages and Data Setup

Task 1

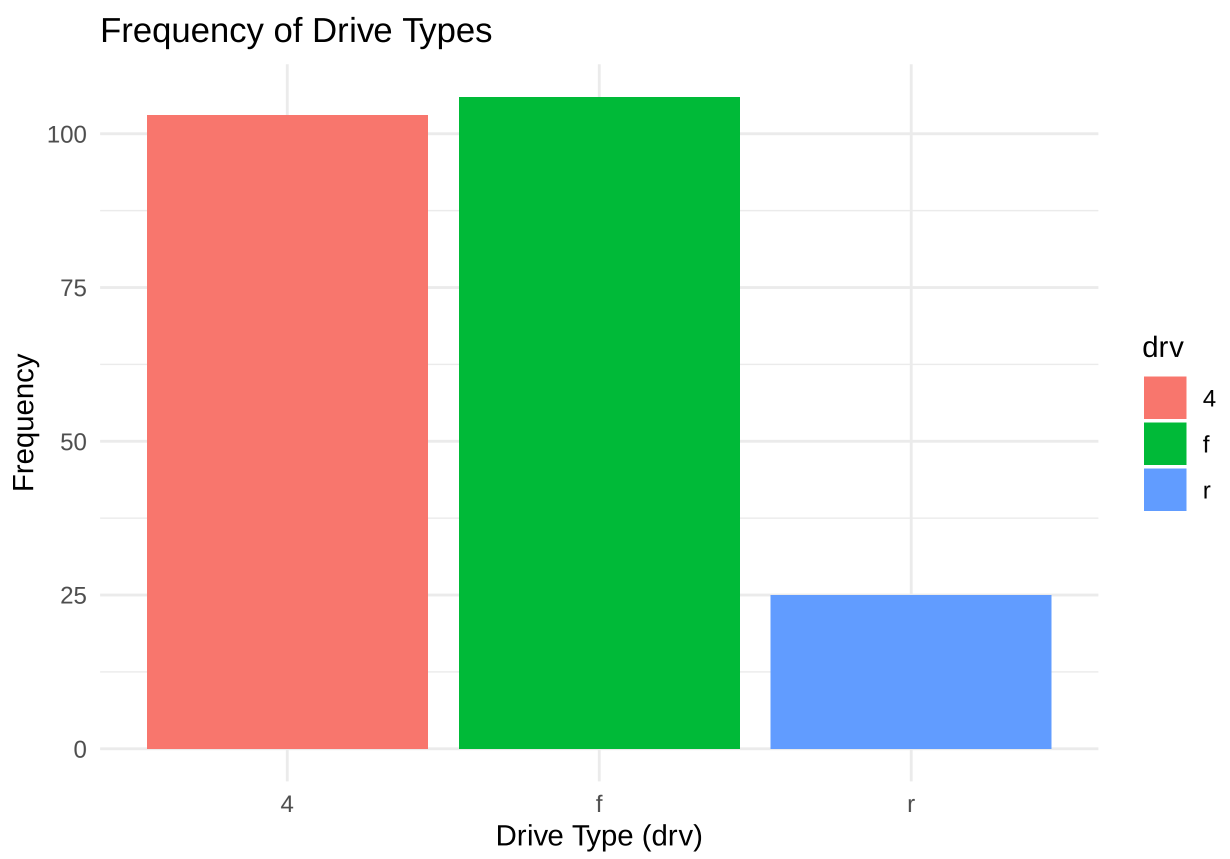
```r
# Load the mpg dataset as a Data Frame
df <- as.data.frame(mpg)

# Check the Data Frame
head(df)
```

```
##   manufacturer model displ year cyl      trans drv cty hwy fl   class
## 1         audi    a4   1.8 1999   4   auto(l5)   f  18  29  p compact
## 2         audi    a4   1.8 1999   4 manual(m5)   f  21  29  p compact
## 3         audi    a4   2.0 2008   4 manual(m6)   f  20  31  p compact
## 4         audi    a4   2.0 2008   4   auto(av)   f  21  30  p compact
## 5         audi    a4   2.8 1999   6   auto(l5)   f  16  26  p compact
## 6         audi    a4   2.8 1999   6 manual(m5)   f  18  26  p compact
```

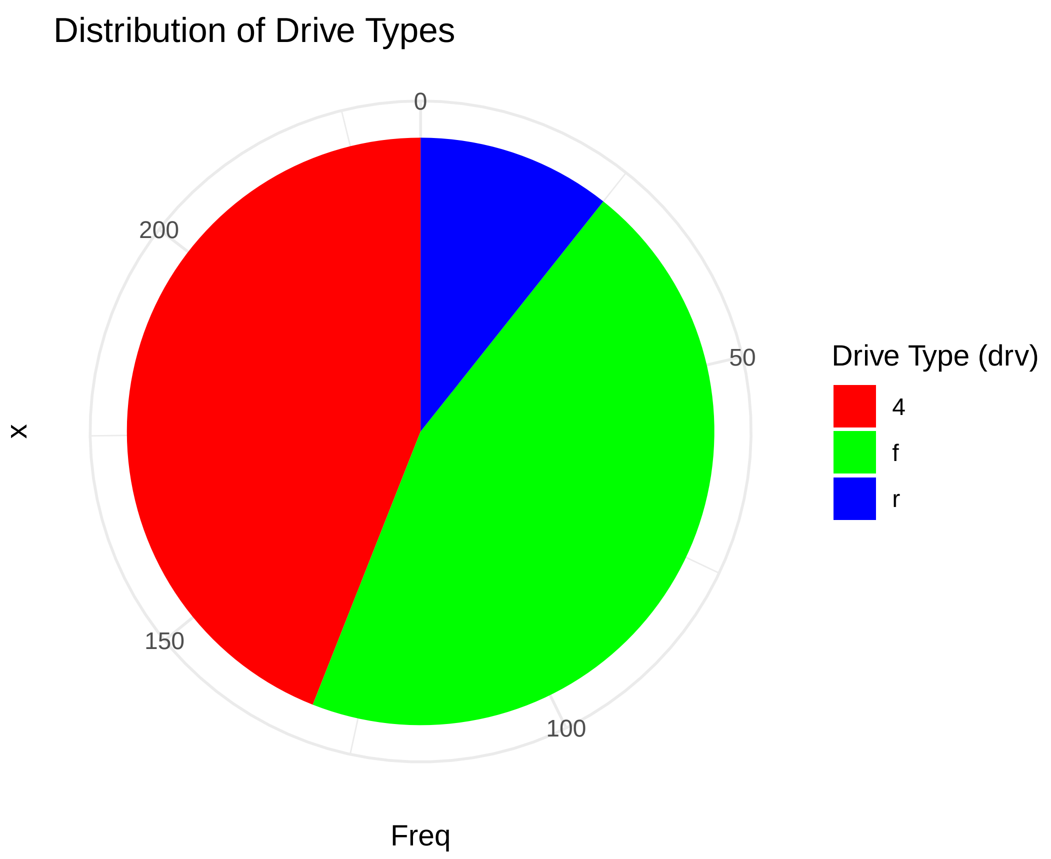
```r
# Create a frequency table for the 'drv' variable
freq_table <- as.data.frame(table(mpg$drv))

# Calculate relative frequency and percentage
freq_table$rel_Freq <- round(freq_table$Freq / sum(freq_table$Freq), 2)
freq_table$Percentage <- round(freq_table$rel_Freq * 100, 2)

# Rename the columns for clarity
colnames(freq_table) <- c("drv", "Freq", "rel_Freq", "Percentage")

# Print the frequency table
print(freq_table)
```

```
##   drv Freq rel_Freq Percentage
## 1   4  103     0.44         44
## 2   f  106     0.45         45
## 3   r   25     0.11         11
```

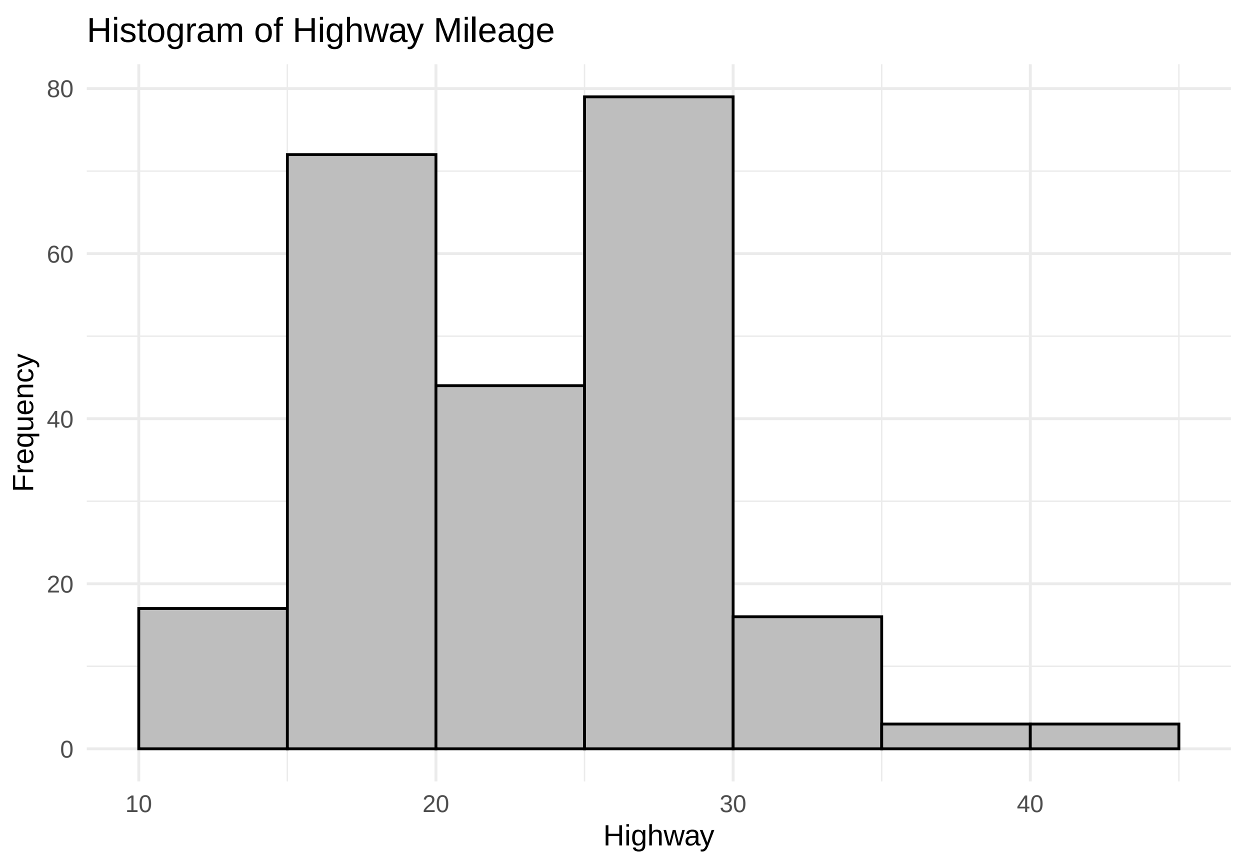
```r
# Create a bar chart
ggplot(freq_table, aes(x = drv, y = Freq, fill = drv)) +
  geom_bar(stat = "identity") +
  labs(title = "Frequency of Drive Types", x = "Drive Type (drv)", y = "Frequency") +
  theme_minimal()
```



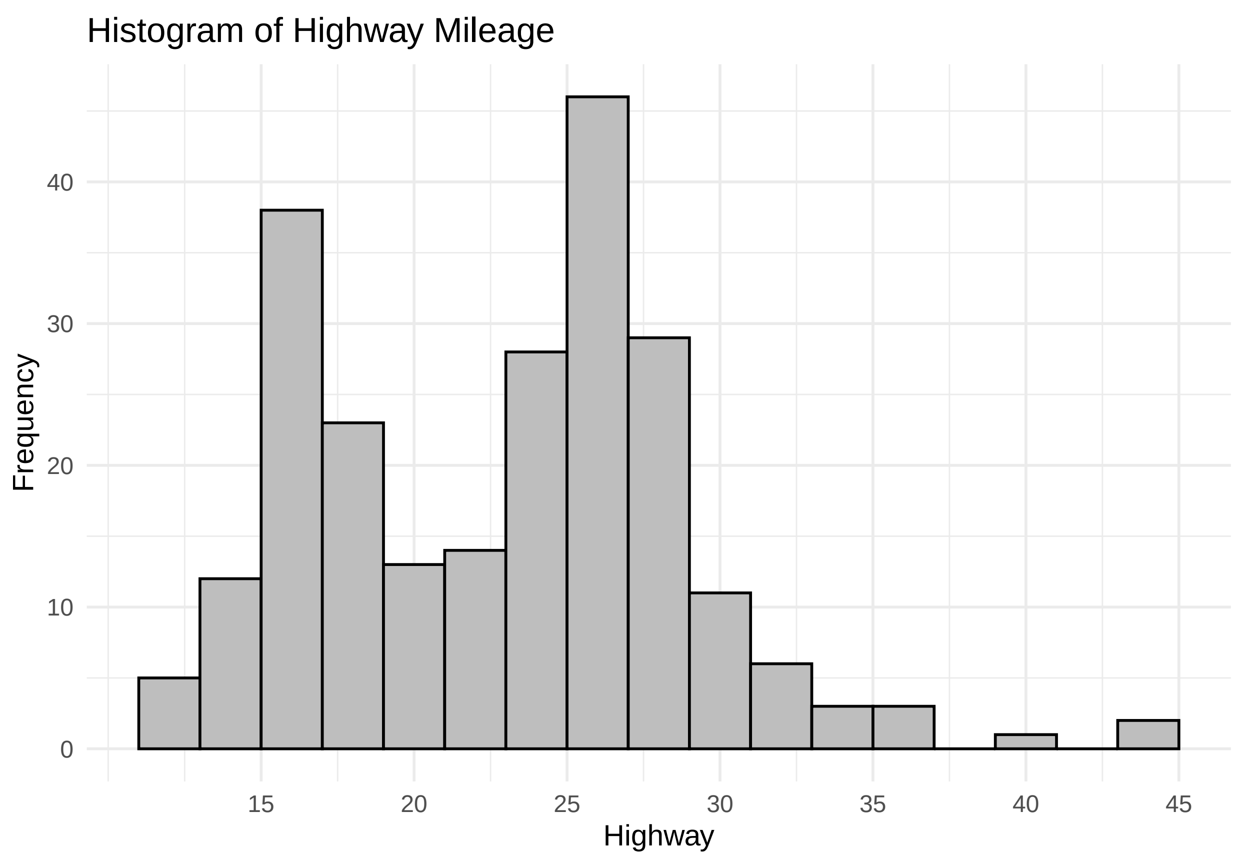
```r
# Create a pie chart
ggplot(freq_table, aes(x = "", y = Freq, fill = drv)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  scale_fill_manual(values = c("4" = "red", "f" = "green", "r" = "blue")) +
  labs(title = "Distribution of Drive Types", fill = "Drive Type (drv)") +
  theme_minimal()
```



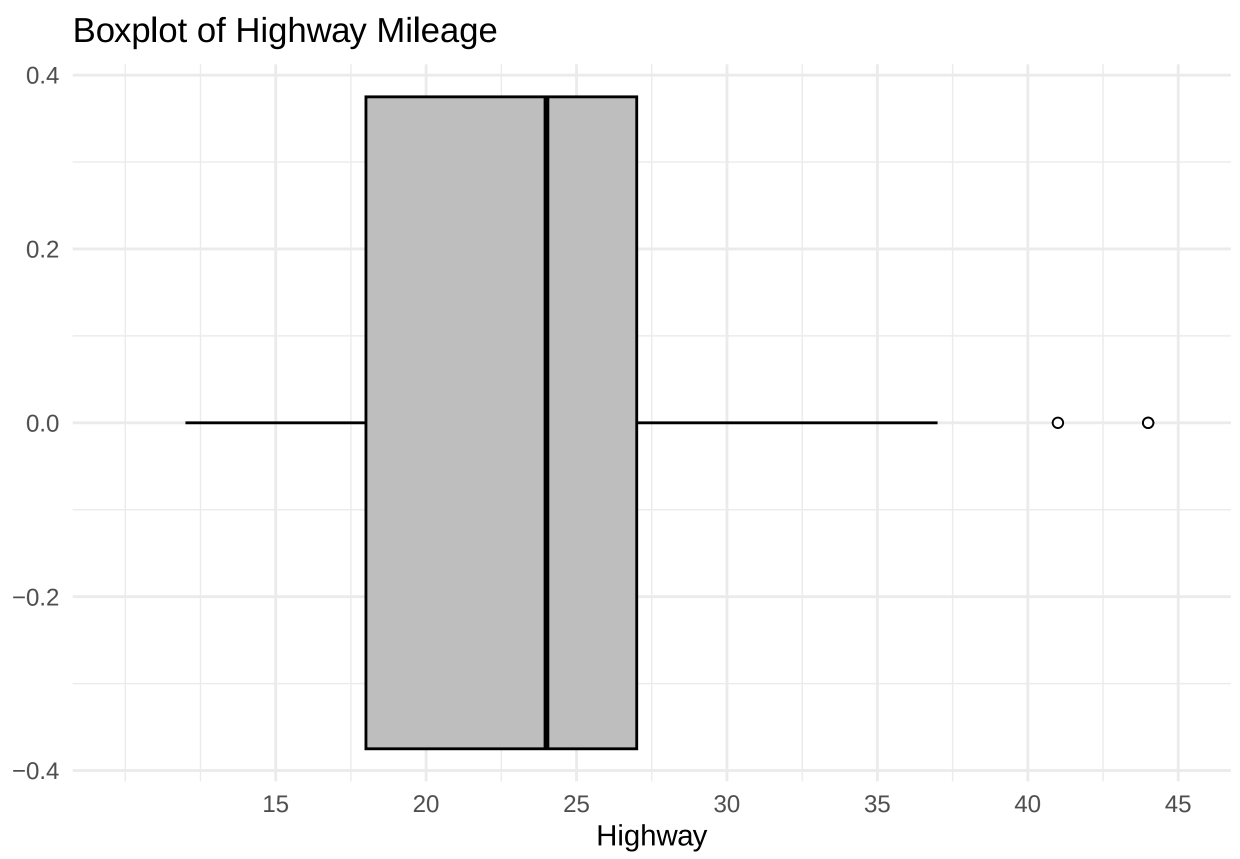
```r
# Simple histogram with custom breaks
ggplot(df, aes(x = hwy)) +
  geom_histogram(breaks = c(10, 15, 20, 25, 30, 35, 40, 45),
      fill = "grey", color = "black") +
  labs(title = "Histogram of Highway Mileage", x = "Highway", y = "Frequency") +
  theme_minimal()
```



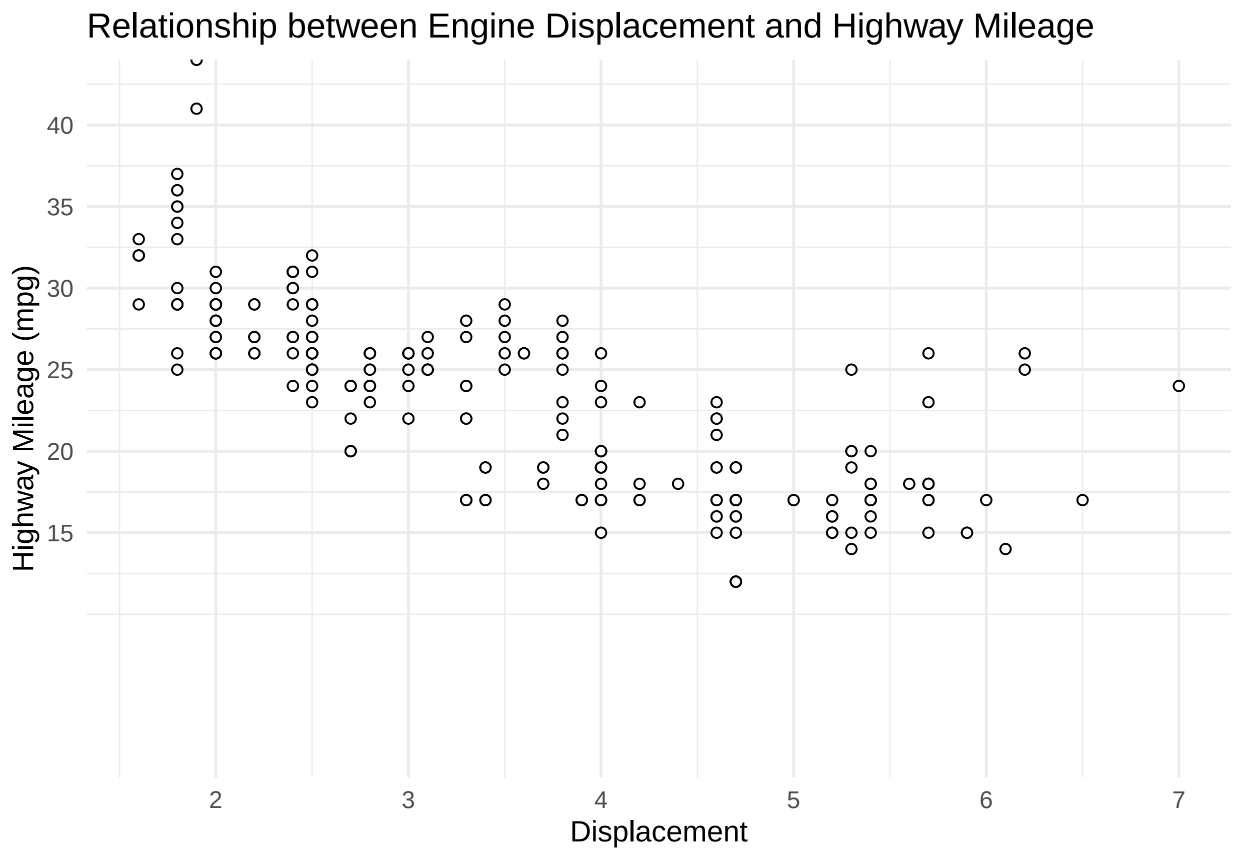
```
# Histogram with x-axis labeled from 15 to 45 in steps of 5
ggplot(df, aes(x = hwy)) +
  geom_histogram(binwidth = 2, fill = "grey", color = "black") +
  labs(title = "Histogram of Highway Mileage", x = "Highway", y = "Frequency") +
  scale_x_continuous(breaks = seq(15, 45, by = 5)) +
  theme_minimal()
```



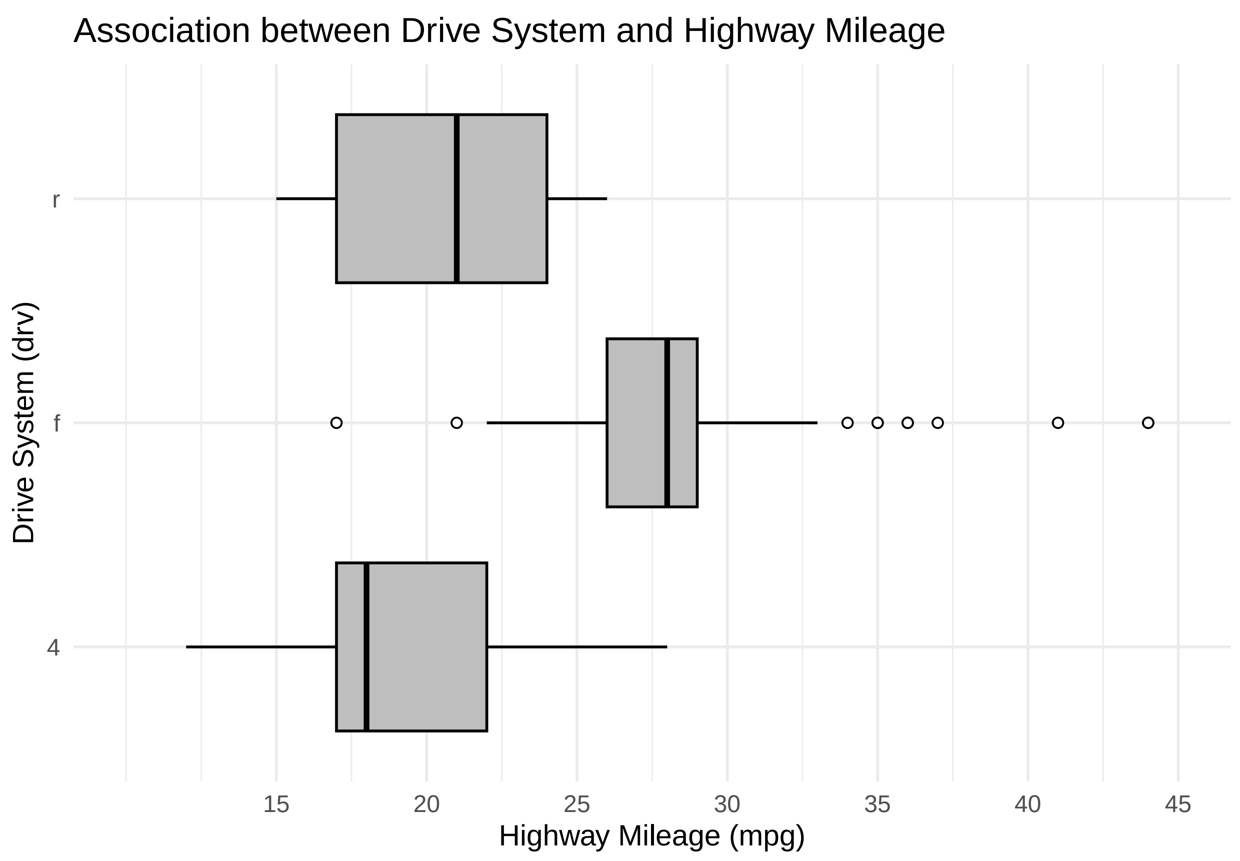
```
# Simple Boxplot
ggplot(df, aes(x = hwy)) +
  geom_boxplot(fill = "grey", color = "black", outlier.shape = 21,
               outlier.fill = "white", outlier.color = "black") +
  scale_x_continuous(breaks = seq(15, 45, by = 5), limits = c(10, 45), name = "Highway") +
  labs(title = "Boxplot of Highway Mileage") +
  theme_minimal()
```



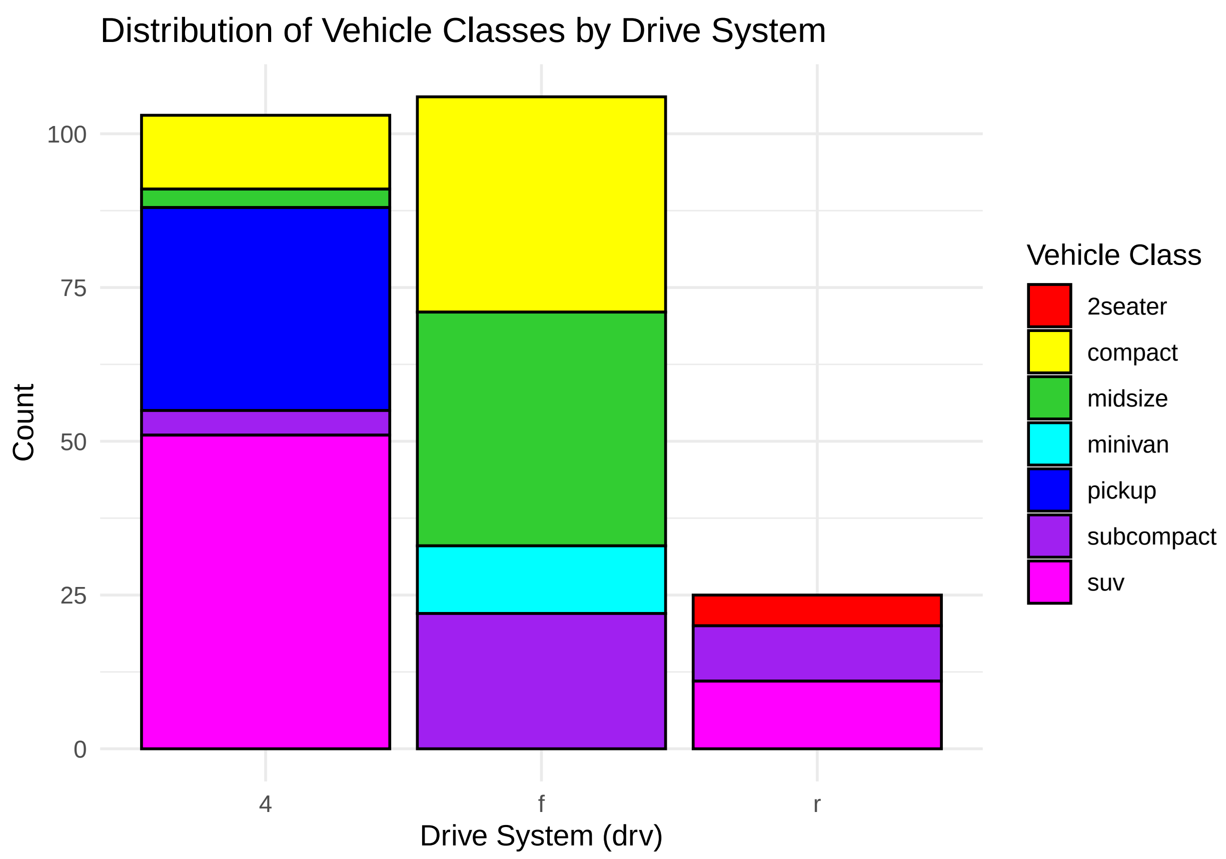
```r
# Scatter plot with adjusted y-axis limits, no grid, and white points with black borders
ggplot(df, aes(x = displ, y = hwy)) +
  geom_point(color = "black", fill = "white", shape = 21) +
  labs(title = "Relationship between Engine Displacement and Highway Mileage",
       x = "Displacement",
       y = "Highway Mileage (mpg)") +
  scale_y_continuous(breaks = seq(15, 45, by = 5), limits = c(0, NA), expand = c(0, 0)) +
  theme_minimal()
```



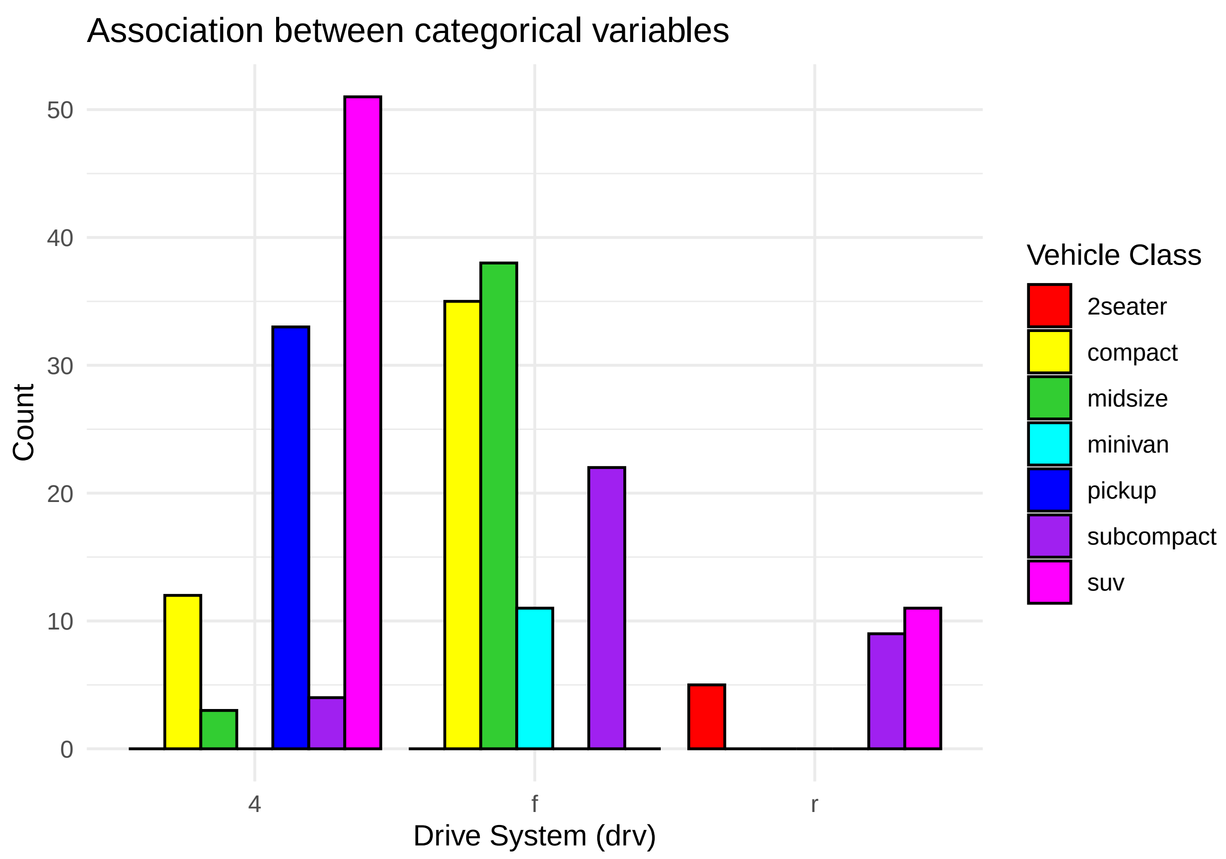
```r
# boxplot showing the association between highway mileage and drive system
ggplot(df, aes(y = drv, x = hwy)) +
  geom_boxplot(fill = "grey", color = "black", outlier.shape = 21,
               outlier.fill = "white", outlier.color = "black") +
  labs(title = "Association between Drive System and Highway Mileage",
       y = "Drive System (drv)",
       x = "Highway Mileage (mpg)") +
  scale_x_continuous(breaks = seq(15, 45, by = 5), limits = c(10, 45)) +
  theme_minimal()
```



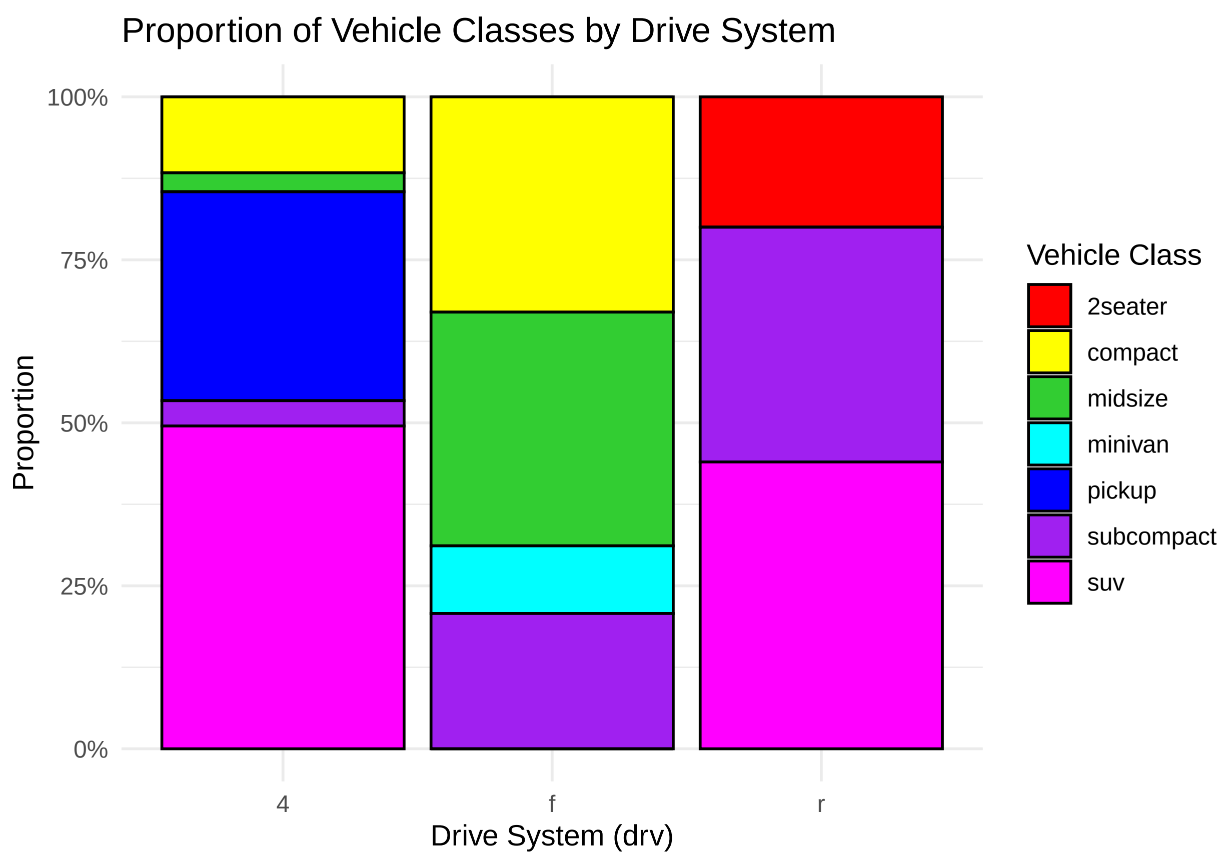
```r
# Stacked bar chart showing the distribution of vehicle classes within each drive type
ggplot(df, aes(x = drv, fill = class)) +
  geom_bar(position = "stack", color = "black") +
  labs(title = "Distribution of Vehicle Classes by Drive System",
       x = "Drive System (drv)",
       y = "Count",
       fill = "Vehicle Class") +
  scale_fill_manual(values = c("suv" = "magenta",
                               "subcompact" = "purple",
                               "pickup" = "blue",
                               "minivan" = "cyan",
                               "midsize" = "limegreen",
                               "compact" = "yellow",
                               "2seater" = "red")) +
  theme_minimal()
```



```r
# Clustered bar chart showing association between drv and class
df_complete <- as.data.frame(table(mpg$drv, mpg$class))
colnames(df_complete) <- c("drv", "class", "n")

ggplot(df_complete, aes(x = drv, y = n, fill = class)) +
  geom_bar(stat = "identity", position = "dodge", color = "black") +
  labs(title = "Association between categorical variables",
       x = "Drive System (drv)", y = "Count", fill = "Vehicle Class") +
  scale_fill_manual(values = c("2seater" = "red",
                               "compact" = "yellow",
                               "midsize" = "limegreen",
                               "minivan" = "cyan",
                               "pickup" = "blue",
                               "subcompact" = "purple",
                               "suv" = "magenta")) +
  theme_minimal()
```



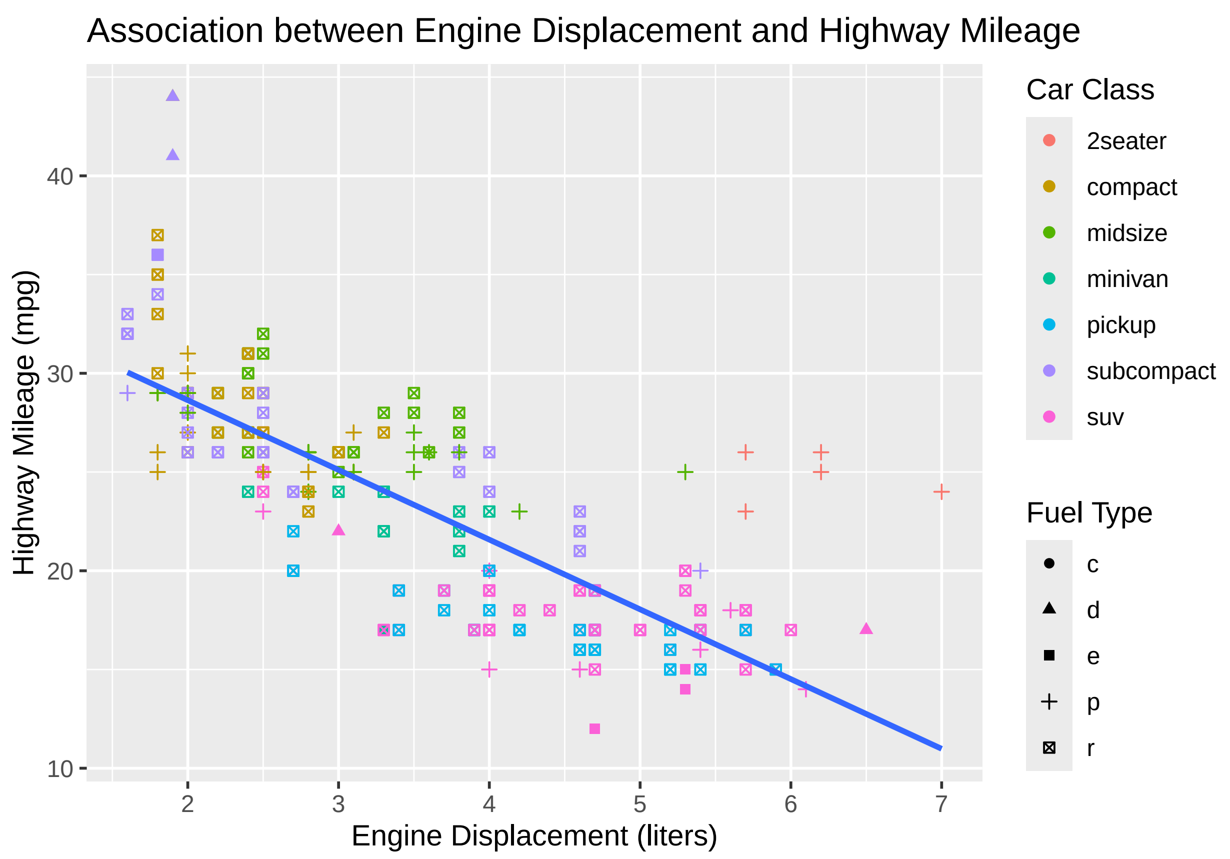
```r
# Stacked bar chart with proportions
df_propor <- df_complete
df_propor$proportion <- df_propor$n / ave(df_propor$n, df_propor$drv, FUN = sum)

ggplot(df_propor, aes(x = drv, y = proportion, fill = class)) +
  geom_bar(stat = "identity", position = "fill", color = "black") +
  labs(title = "Proportion of Vehicle Classes by Drive System",
       x = "Drive System (drv)",
       y = "Proportion",
       fill = "Vehicle Class") +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_manual(values = c("2seater" = "red",
                               "compact" = "yellow",
                               "midsize" = "limegreen",
                               "minivan" = "cyan",
                               "pickup" = "blue",
                               "subcompact" = "purple",
                               "suv" = "magenta")) +
  theme_minimal()
```



Task 2: Association between engine displacement and highway mileage

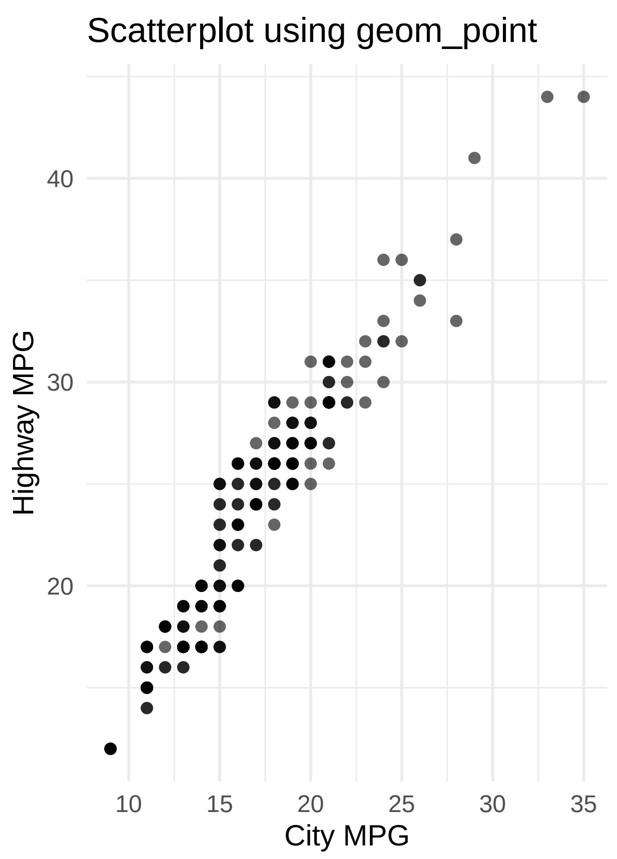
```
# Scatter plot to show the association between engine displacement and highway mileage
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(shape = fl, color = class)) +
  geom_smooth(method = "lm", se = F) +
  labs(title = "Association between Engine Displacement and Highway Mileage",
       x = "Engine Displacement (liters)",
       y = "Highway Mileage (mpg)",
       color = "Car Class",
       shape = "Fuel Type")
```

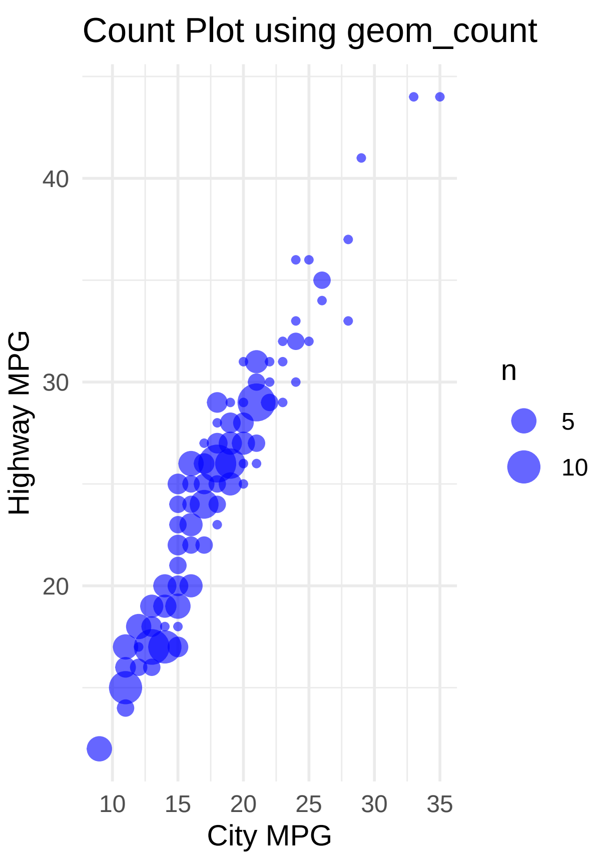
```
## `geom_smooth()` using formula = 'y ~ x'
```



Task 3: Comparison of geom_point() and geom_count()

```r
# Scatter plot using geom_point
point_plot <- ggplot(data = mpg, aes(x = cty, y = hwy)) +
  geom_point(alpha = 0.6) +  # Add slight transparency for overlapping points
  labs(title = "Scatterplot using geom_point", x = "City MPG", y = "Highway MPG") +
  theme_minimal()

# Count plot using geom_count
count_plot <- ggplot(data = mpg, aes(x = cty, y = hwy)) +
  geom_count(color = "blue", alpha = 0.6) +  # Color to enhance density visibility
  labs(title = "Count Plot using geom_count", x = "City MPG", y = "Highway MPG") +
  theme_minimal()

# Display plots side by side
grid.arrange(point_plot, count_plot, ncol = 2)
```

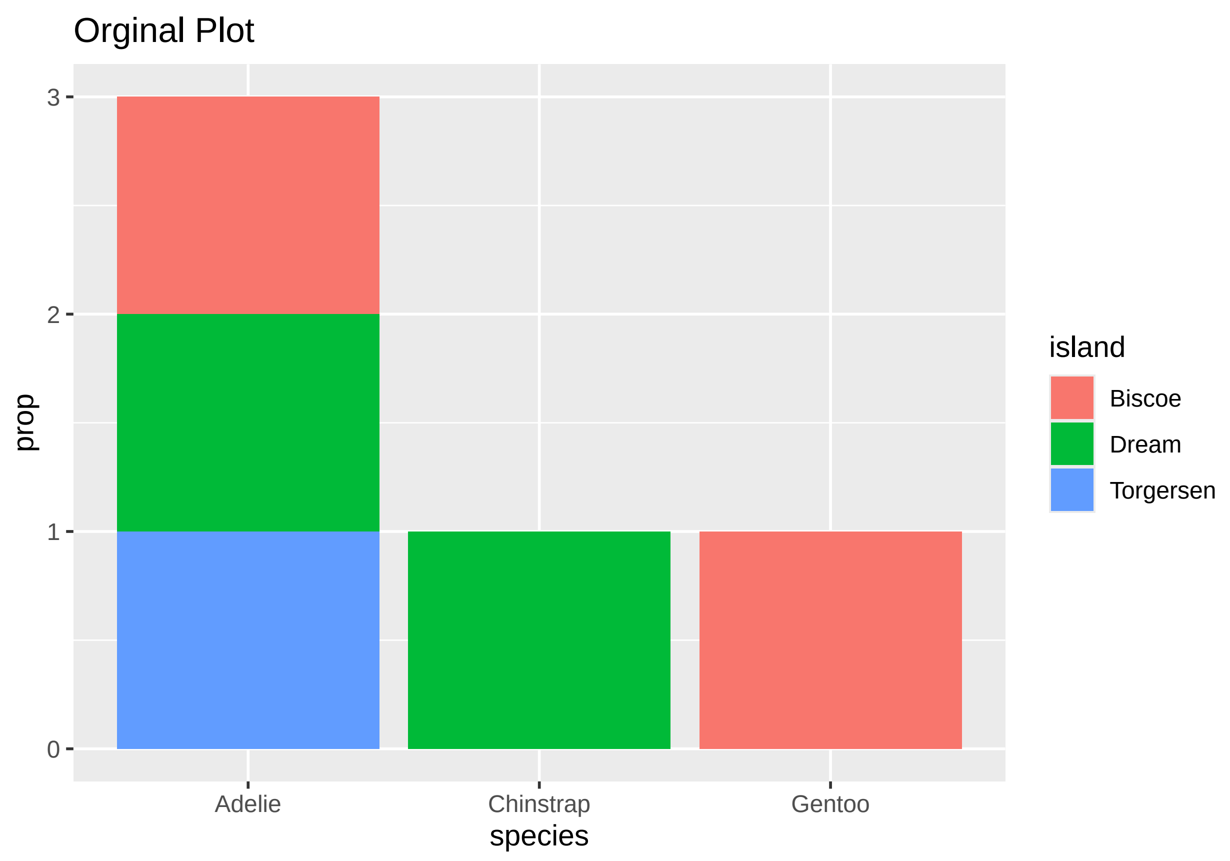




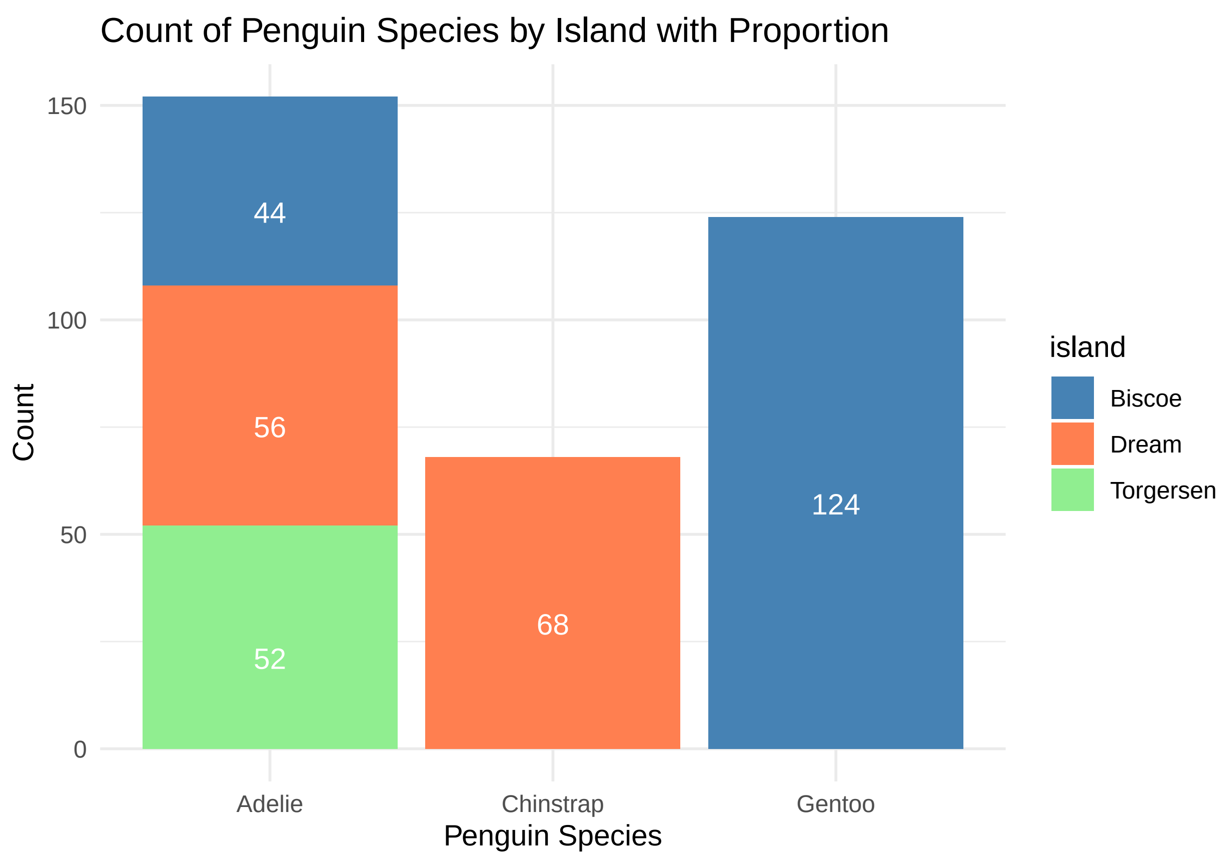
For Task 3, we use geom_point() and geom_count() to explore the relationship between cty (city MPG) and hwy (highway MPG) in the mpg dataset. In the scatter plot with geom_point() (as suggested in the slides on visualizing distributions),each point represents an observation, which can obscure dense areas if points overlap. Adding slight transparency (alpha) mitigates this and reveals overlapping data points. In contrast, geom_count() changes the size of each point based on its count,providing a clear indication of where data points are densest, a recommendation seen in the slides for visual clarity when values overlap. Overall, geom_point() is ideal for datasets with low overlap, while geom_count() effectively highlights density in datasets with repeated values. These enhancements meet the general assignment criteria for readability and clarity in data presentation.

Task 4: Penguins

```r
# Load the penguins dataset
data(penguins)

# Original (copy of the homework)
ggplot(data = penguins, aes(fill = island, x = species)) +
geom_bar(aes(y = after_stat(prop))) +
  labs(title = "Orginal Plot")
```



```
# Improved version with absolute counts displayed over proportionally filled bars
p <- ggplot(data = penguins, aes(x = species, fill = island)) +
  geom_bar(position = "stack") +
  geom_text(stat = "count",
            aes(label = after_stat(count)),
            position = position_stack(vjust = 0.5),
            color = "white",
            vjust = 1.5) +
  labs(title = "Count of Penguin Species by Island with Proportion",
       x = "Penguin Species",
       y = "Count") +
  scale_fill_manual(values = c("Biscoe" = "steelblue",
                               "Dream" = "coral",
                               "Torgersen" = "lightgreen")) +
  theme_minimal()

# Display the plot
p
```



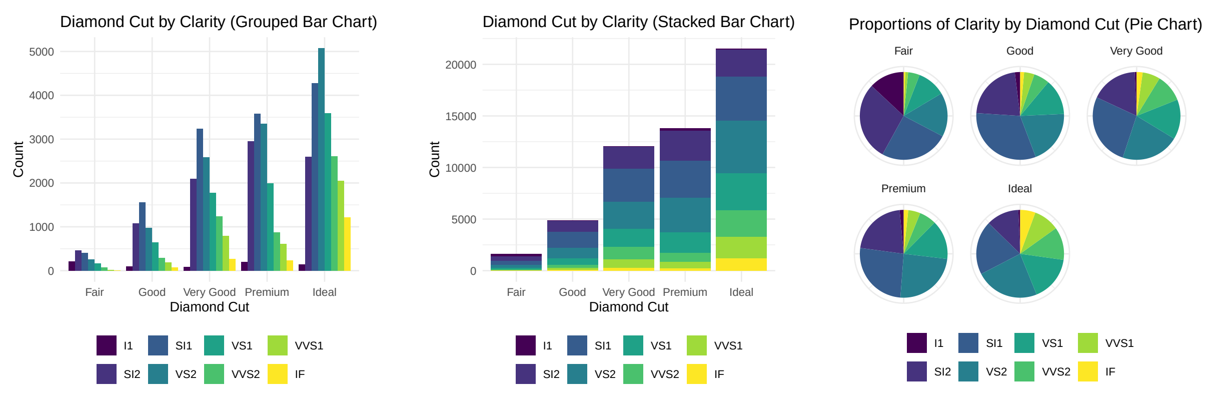
Problems with the original plot

In the original plot, you can only see which penguin species are present on each island. For example, on Torgersen Island, there are only Adelie penguins, whereas Biscoe Island has both Gentoo and Adelie penguins. However, it does not give a sense of where there are larger populations or the total number of birds on each island. That's why we added the total counts to each bar in our updated plot. Additionally, the proportions are now displayed accurately, which enhances the clarity of the visualization. For instance, you can immediately see that while Adelie penguins are the most common species, Biscoe Island has the largest overall penguin population.

Task 5: Diamond

```r
# Load the diamonds dataset
data(diamonds)

# Basic bar chart - Clarity distribution grouped by diamond cut
bar_plot <- ggplot(diamonds, aes(x = cut, fill = clarity)) +
  geom_bar(position = "dodge") +  # 'dodge' position for grouped bars
  labs(title = "Diamond Cut by Clarity (Grouped Bar Chart)",
       x = "Diamond Cut", y = "Count") +
  theme_minimal() +
  theme(legend.position = "bottom", legend.title = element_blank(),
        plot.margin = unit(c(1, 1, 1, 1), "cm"))  # Add margin for spacing
```

```r
# Stacked bar chart - Clarity distribution stacked within each cut category
stacked_bar_plot <- ggplot(diamonds, aes(x = cut, fill = clarity)) +
  geom_bar(position = "stack") +  # 'stack' position for cumulative counts
  labs(title = "Diamond Cut by Clarity (Stacked Bar Chart)",
       x = "Diamond Cut", y = "Count") +
  theme_minimal() +
  theme(legend.position = "bottom", legend.title = element_blank(),
        plot.margin = unit(c(1, 1, 1, 1), "cm"))  # Add margin for spacing

# Pie chart - Proportions of clarity within each cut category
pie_chart <- ggplot(diamonds, aes(x = "", fill = clarity)) +
  geom_bar(width = 1, position = "fill") +  # 'fill' to normalize each segment
  coord_polar("y") +  # Polar coordinates for pie chart effect
  facet_wrap(~cut) +  # Faceting to display one pie chart per cut
  labs(title = "Proportions of Clarity by Diamond Cut (Pie Chart)",
       x = NULL, y = NULL) +
  theme_minimal() +
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text = element_blank(),
        legend.position = "bottom",
        legend.title = element_blank(),
        plot.margin = unit(c(1, 1, 1, 1), "cm"))  # Add margin for spacing

# Display the charts side by side with grid.arrange
grid.arrange(bar_plot, stacked_bar_plot, pie_chart, ncol = 3)
```



comparison task 5

For this task, we created three visualizations to explore the distribution of clarity within each cut category in the diamonds dataset. The grouped bar chart, referencing slide 10, shows clarity levels side-by-side within each cut, making comparisons straightforward. The stacked bar chart, also inspired by slide 10, stacks clarity levels within each cut, emphasizing relative contributions to the total counts. Finally, the faceted pie chart, which uses guidance from slides 12 and 13 on polar coordinates, displays the proportional breakdown of clarity within each cut. While the grouped bar chart is best for clear comparisons, the stacked bar and pie charts provide insights into the relative proportions, each with their own visual strengths.

Which Slides Did We Use for Each Task

Creating R Project: Referenced EMPR_03b_Projects_Reports_AS2024.pdf.

Task 2:

For Task 2, we primarily referenced EMPR_03_Visualization1_AS2024.pdf, which covers most aspects of scatter plots. Pages 4, 7, and 14 were especially helpful -for our work.

Task 3:

For Task 3, we primarily referenced EMPR_03_Visualization1_AS2024.pdf, which covers most aspects of scatter plots. Pages 2, 4, and 5 were especially helpful -for our work.

Task 4:

For Task 4, we primarily referenced EMPR_04_Visualization2_AS2024.pdf, page 8 ("Add counts to bars: geom_text() or geom_label()"), and adapted the code to - -suit our requirements

Task 5:

For Task 5, we primarily referenced EMPR_04_Visualization2_AS2024.pdf; Pages 3 and 10 provided guidance on creating bar charts using geom_bar() and explained the use of the position argument for arranging bars (defaulting to "stack" for stacked bar charts, with "dodge" used for grouped bar charts). Page 12 introduced pie charts and explained how to create them using coord_polar().Page 13 demonstrated examples of applying coord_polar() to geom_bar() to create pie charts, similar to the approach used in our code.

Workload:

- We sat down at the beginning and divided up the tasks. The first thing we did was set up a git repository so that we could easily work together and always see what the other person had done. We organised the tasks so that Fabian did tasks 1 and 2 and Samuel did tasks 3-5. Afterwards we had a short meeting and discussion about the status of the work. We actually got on quite well. Even though we hadn't finished some things yet. We then agreed that Fabian would check and correct Samuel's tasks and the other way round.