Homework1.Rmd

2024-10-10

Packages and Data Setup

Task 1 - Data Frame

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

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

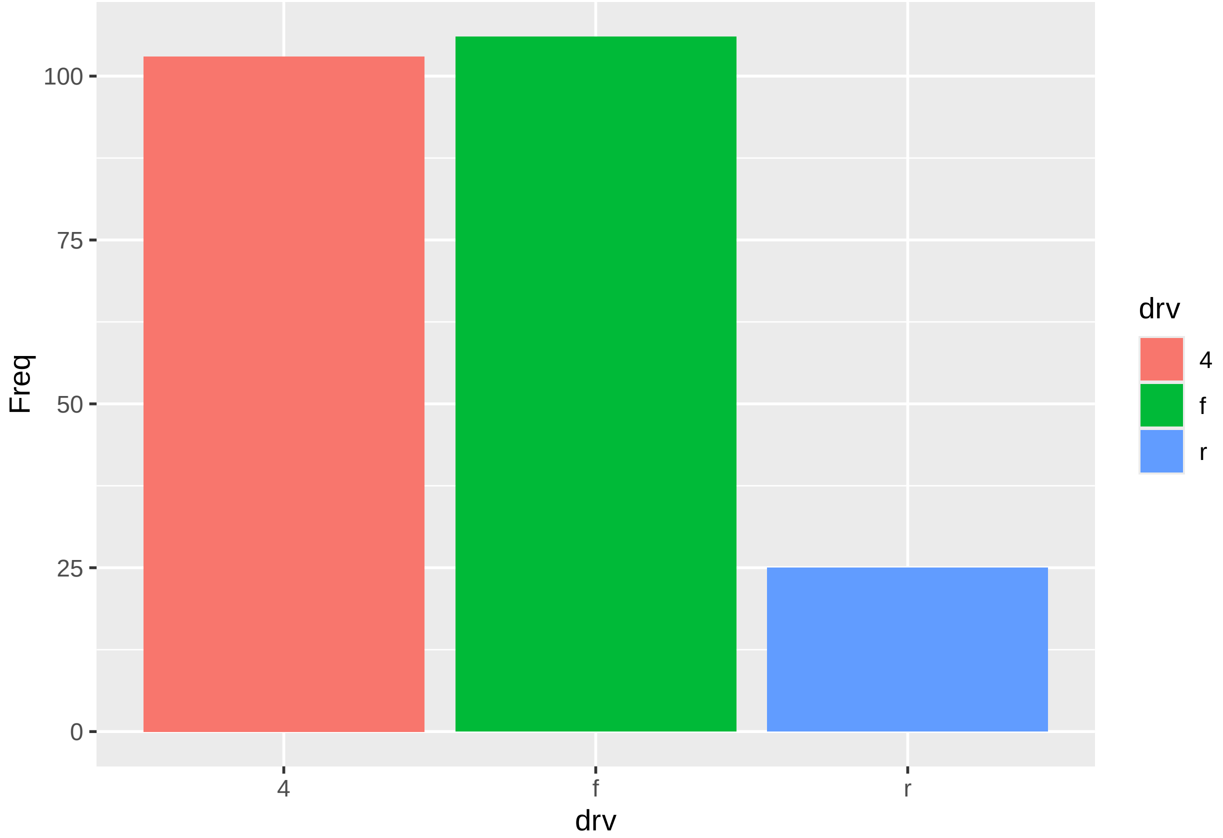
```
##   manufacturer model displ year cyl      trans drv cty hwy fl   class
## 1         audi    a4   1.8 1999   4   auto(l5)   f  18  29  p compact
## 2         audi    a4   1.8 1999   4 manual(m5)   f  21  29  p compact
## 3         audi    a4   2.0 2008   4 manual(m6)   f  20  31  p compact
## 4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

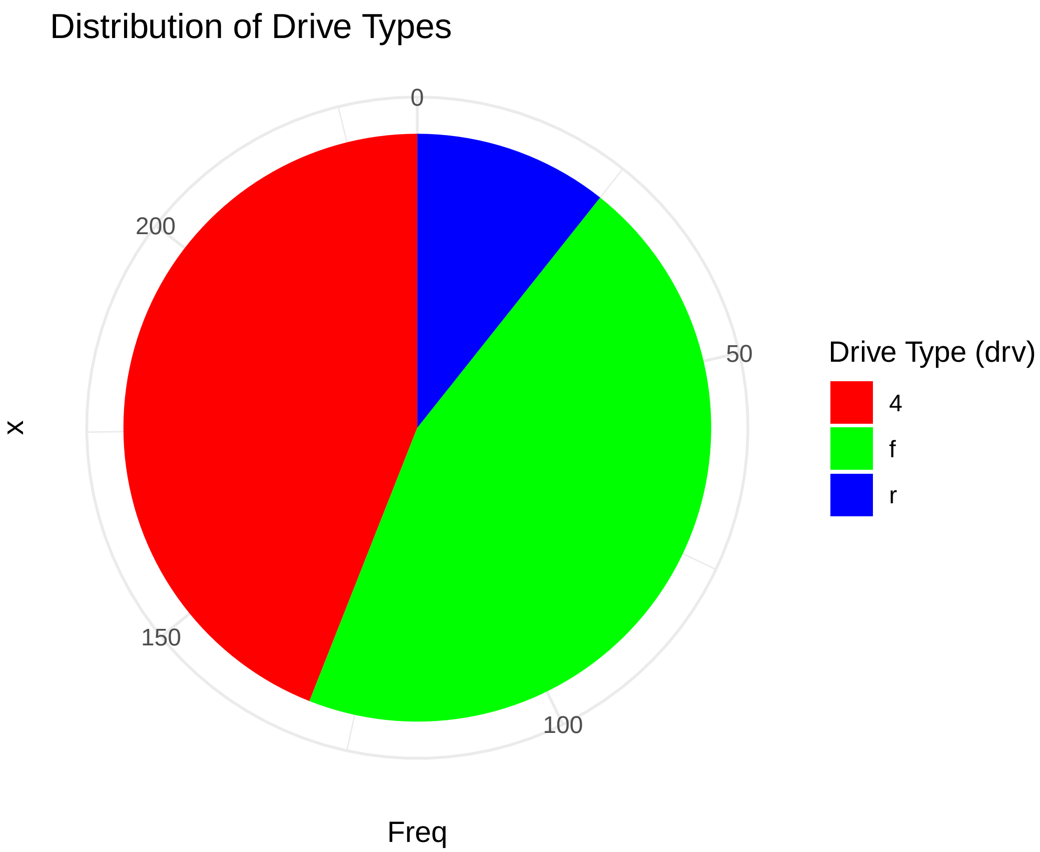
```
##   drv Freq rel_Freq Percentage
## 1   4  103     0.44         44
## 2   f  106     0.45         45
## 3   r   25     0.11         11
```

Visualizations

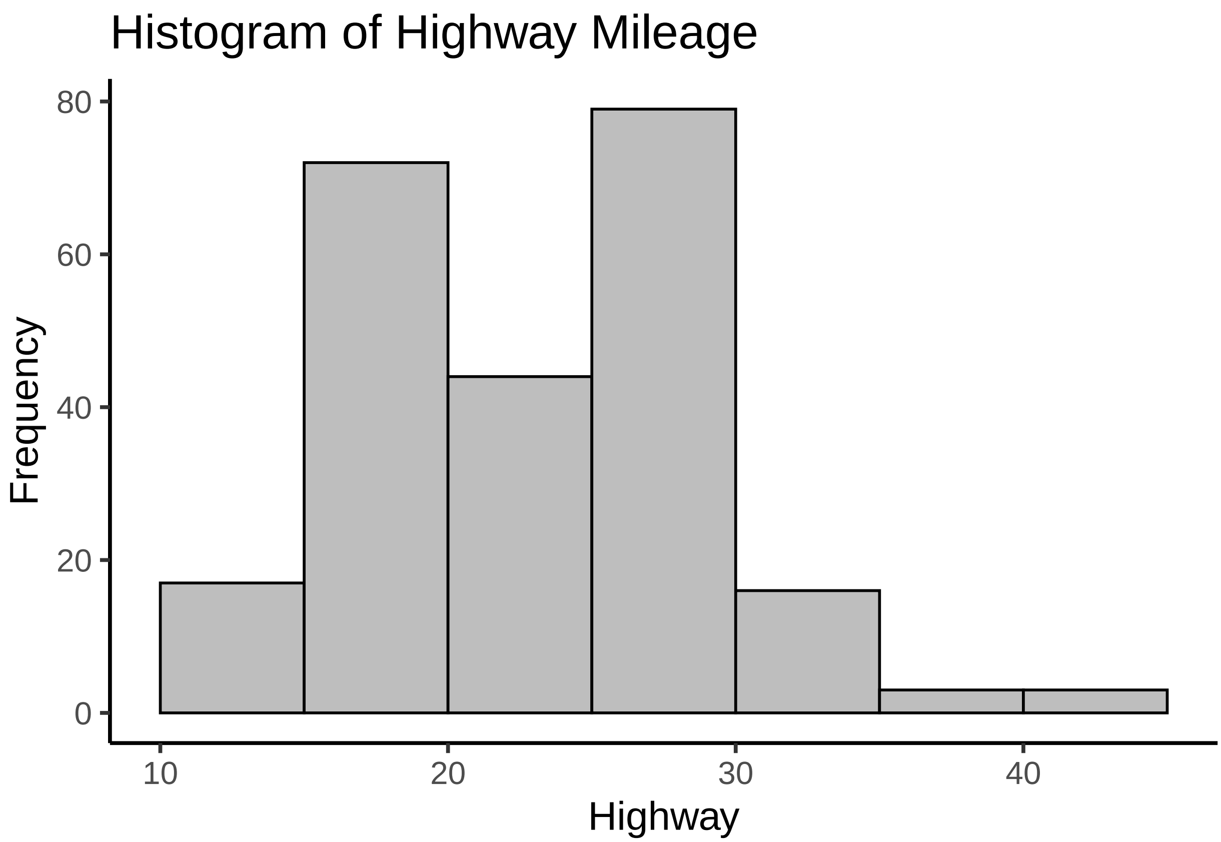
```r
# Create a bar chart
ggplot(freq_table, aes(x = drv, y = Freq, fill = drv)) +
  geom_bar(stat = "identity")
```



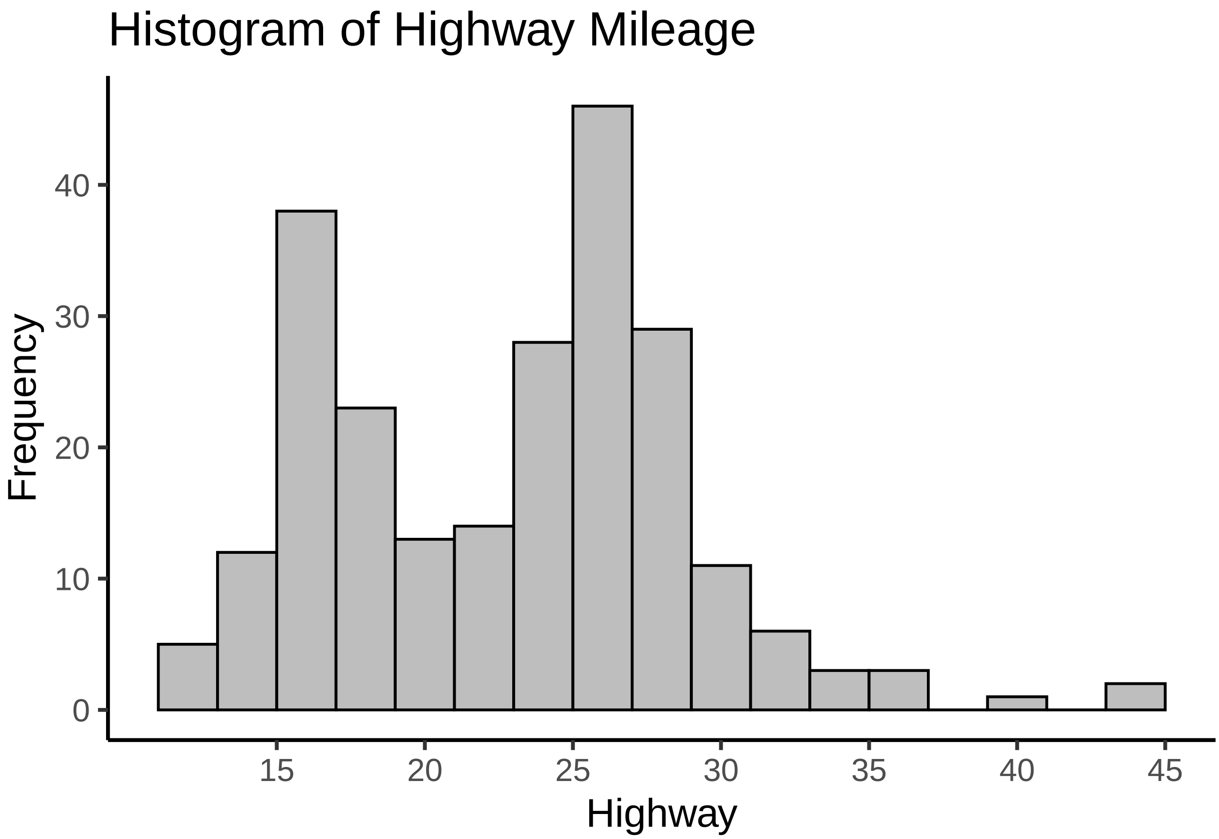
```r
# Create a pie chart
ggplot(freq_table, aes(x = "", y = Freq, fill = drv)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  scale_fill_manual(values = c("4" = "red", "f" = "green", "r" = "blue")) +
  labs(title = "Distribution of Drive Types", fill = "Drive Type (drv)")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



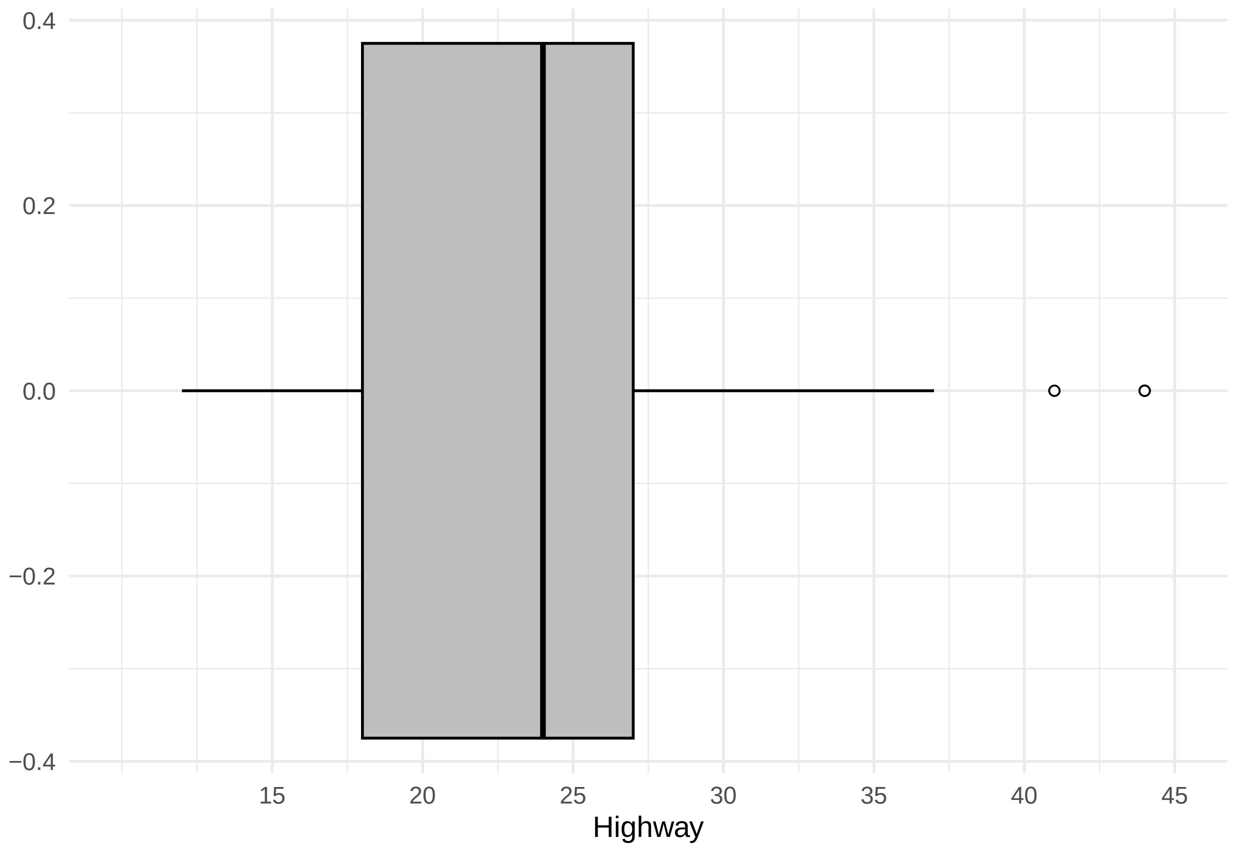
```r
# Simple histogram with custom breaks
ggplot(df, aes(x = hwy)) +
  geom_histogram(breaks = c(10, 15, 20, 25, 30, 35, 40, 45), fill = "grey", color = "black") +
  labs(title = "Histogram of Highway Mileage", x = "Highway", y = "Frequency") +
  theme_classic(base_size = 15)
```



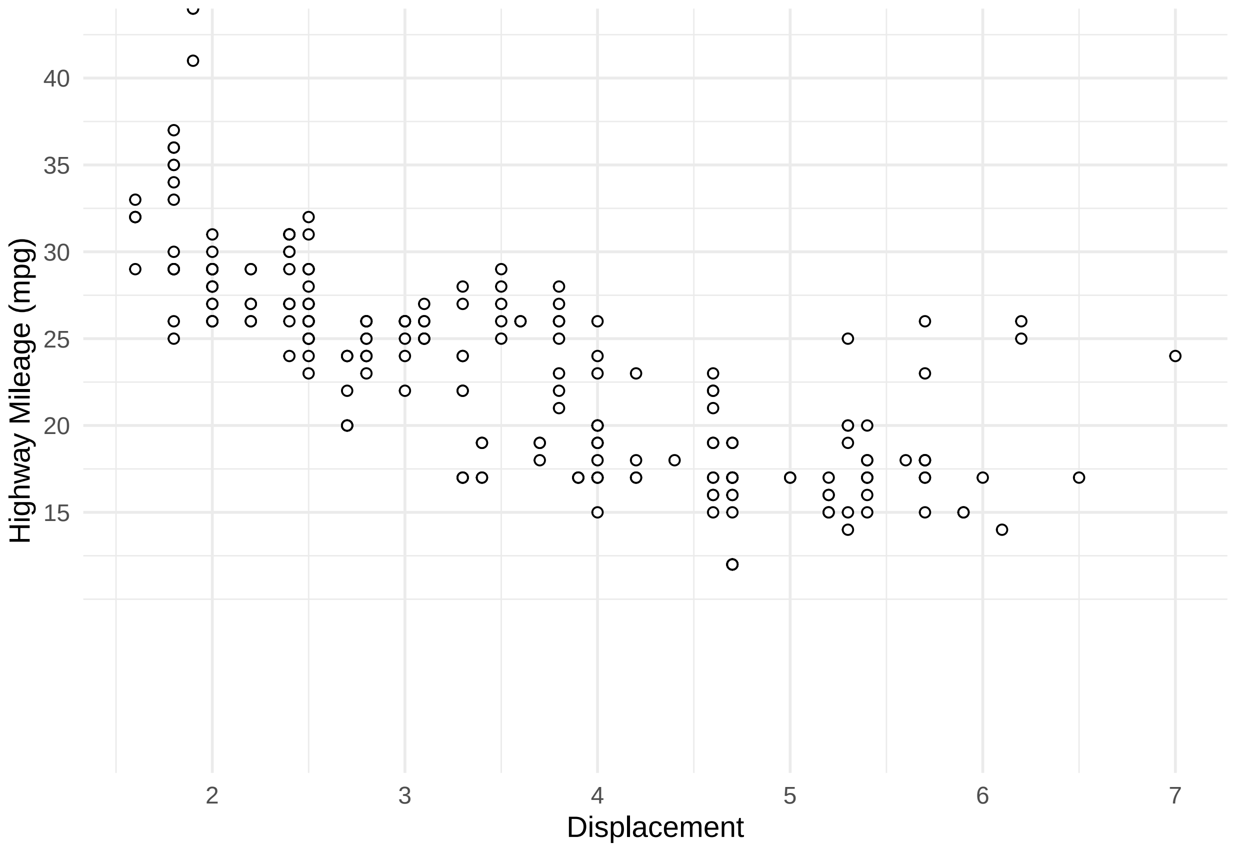
```r
# Histogram with x-axis labeled from 15 to 45 in steps of 5
ggplot(df, aes(x = hwy)) +
  geom_histogram(binwidth = 2, fill = "grey", color = "black") +
  labs(title = "Histogram of Highway Mileage", x = "Highway", y = "Frequency") +
  scale_x_continuous(breaks = seq(15, 45, by = 5)) +
  theme_classic(base_size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



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

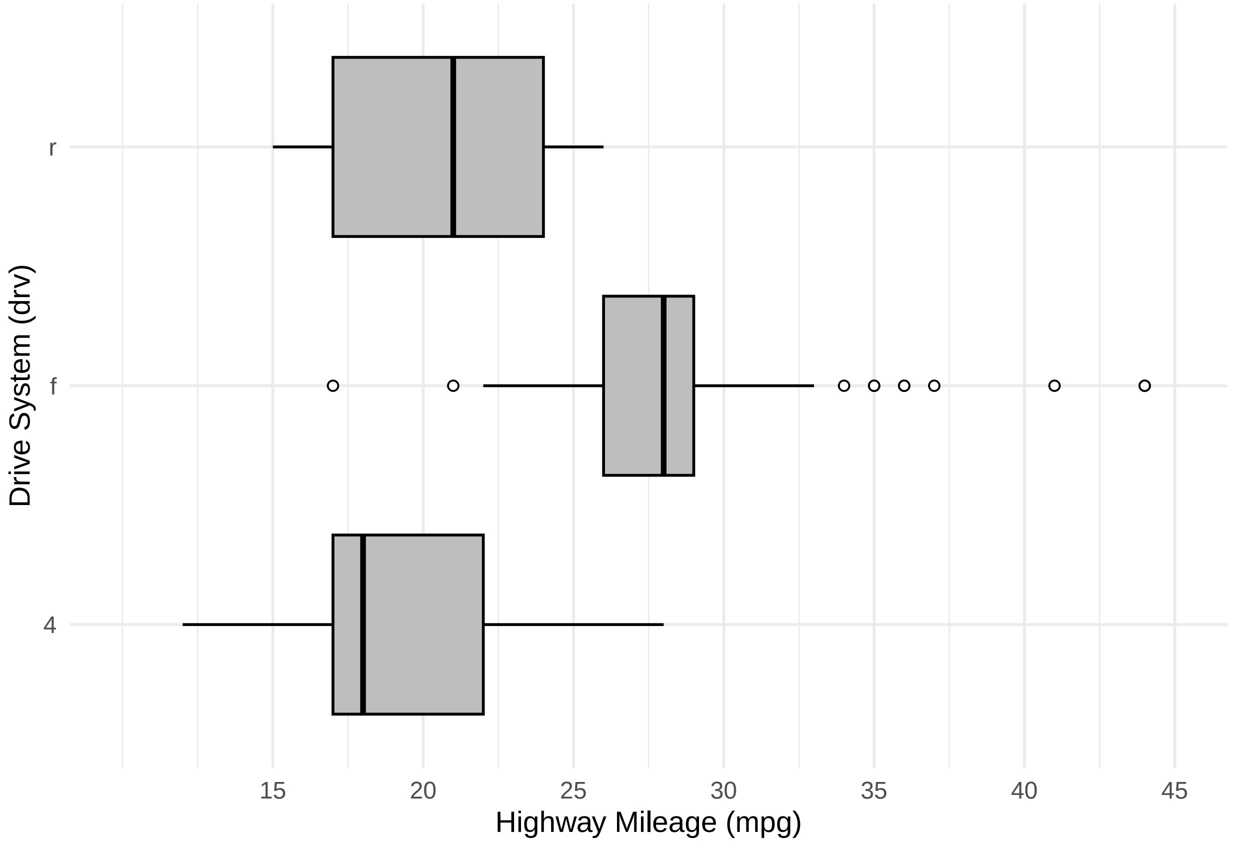


```
# Scatter plot with adjusted y-axis limits, no grid, and white points with black borders
ggplot(df, aes(x = displ, y = hwy)) +
  geom_point(color = "black", fill = "white", shape = 21) +
  labs(x = "Displacement", y = "Highway Mileage (mpg)") +
  scale_y_continuous(breaks = seq(15, 45, by = 5), limits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

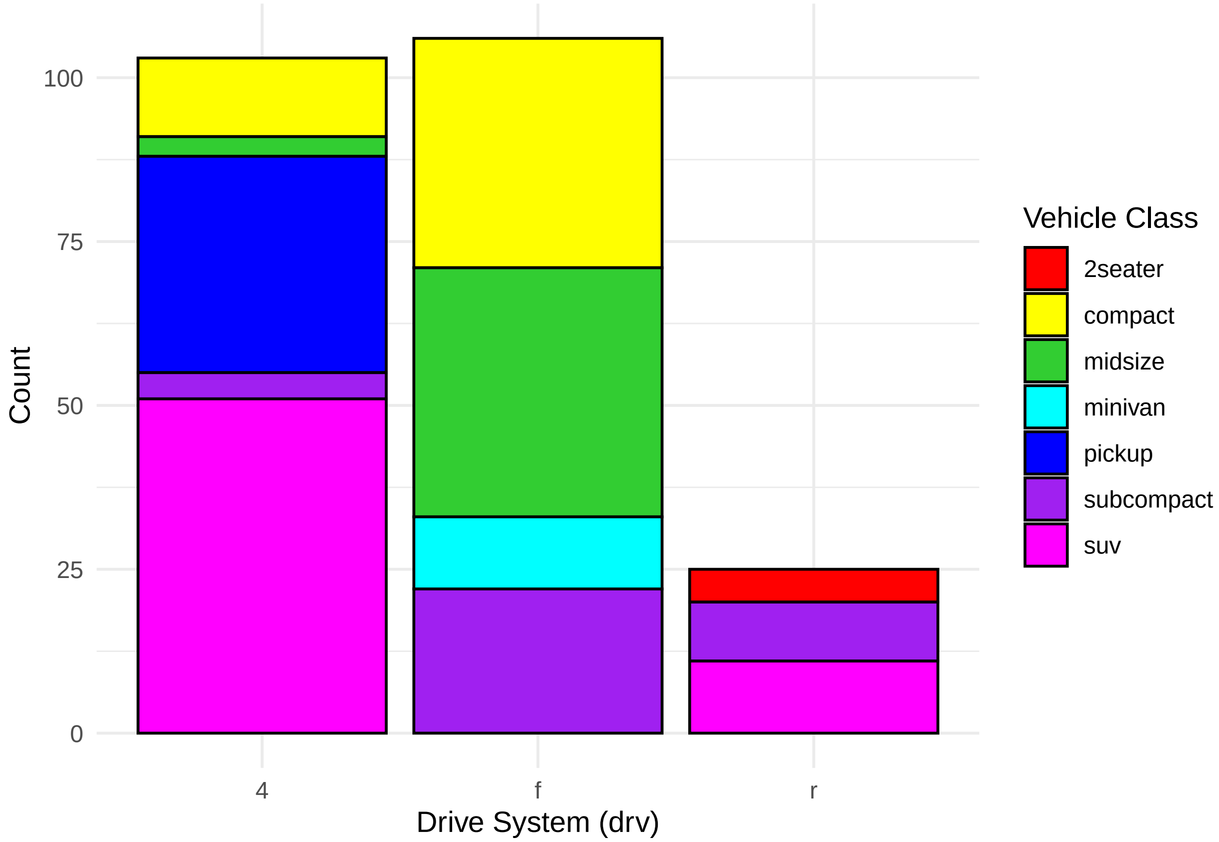


Additional Plots and Analyses

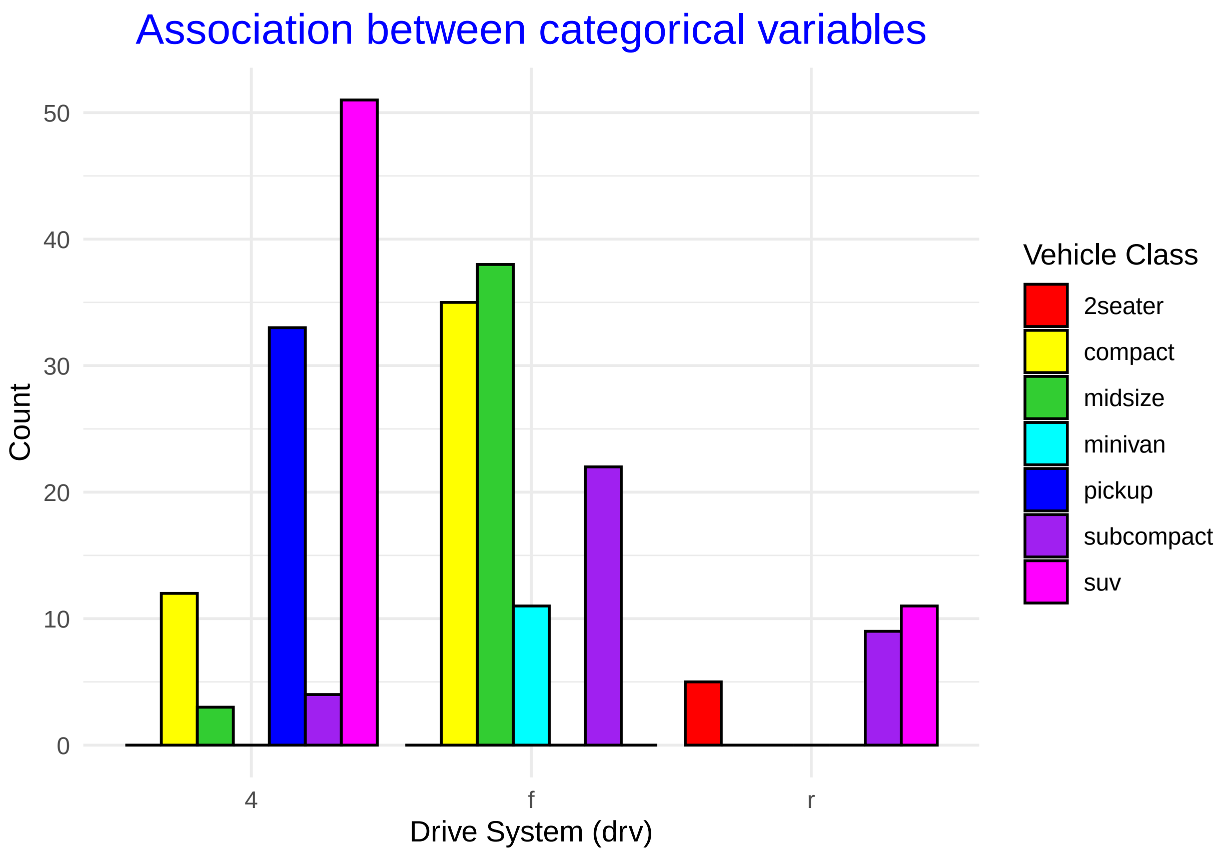
```r
# Horizontal boxplot showing the association between highway mileage (hwy) and drive system (drv)
ggplot(df, aes(y = drv, x = hwy)) +
  geom_boxplot(fill = "grey", color = "black", outlier.shape = 21, outlier.fill = "white", outlier.colo
  labs(y = "Drive System (drv)", x = "Highway Mileage (mpg)")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



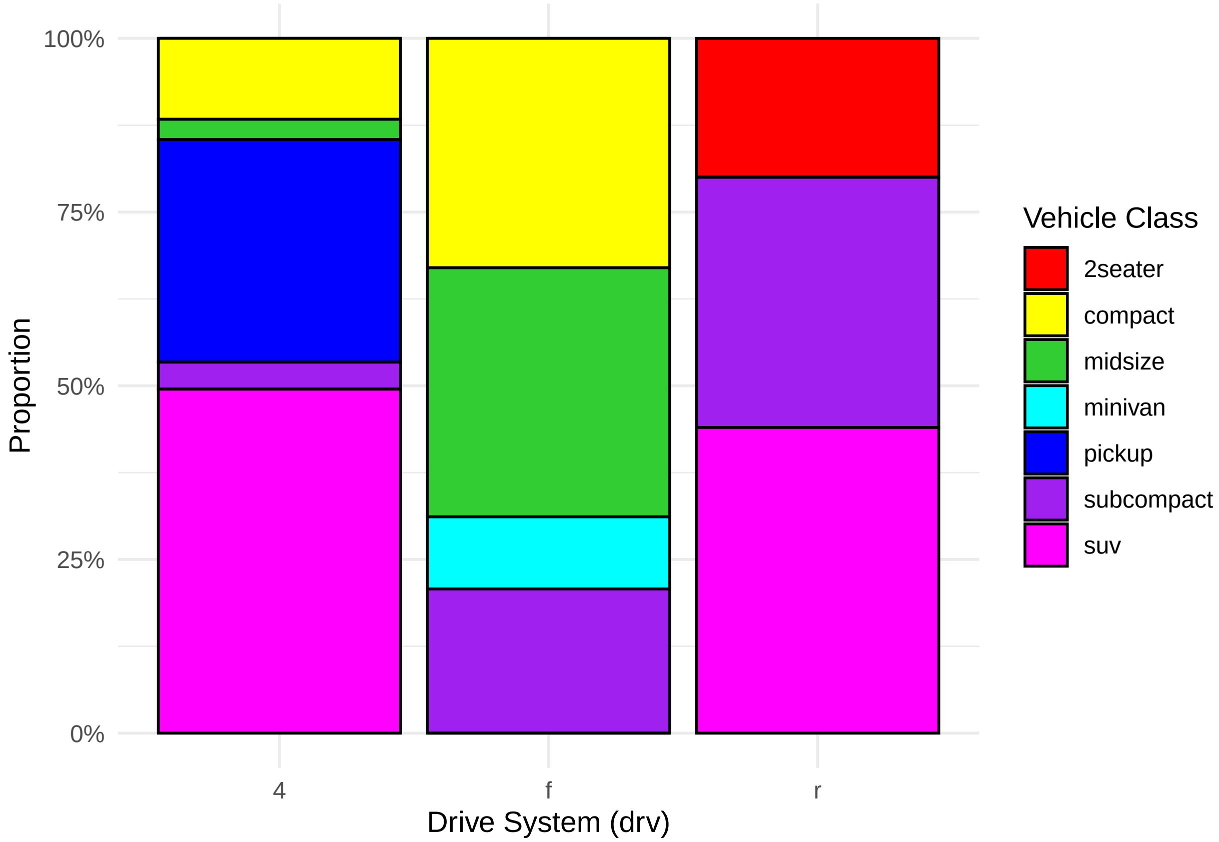
```
# Stacked bar chart showing the distribution of vehicle classes within each drive type
ggplot(df, aes(x = drv, fill = class)) +
  geom_bar(position = "stack", color = "black") +
  labs(x = "Drive System (drv)", y = "Count", fill = "Vehicle Class") +
  scale_fill_manual(values = c("suv" = "magenta",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



```
# 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
  scale_fill_manual(values = c("2seater" = "red", "compact" = "yellow", "midsize" = "limegreen",
                               "minivan" = "cyan", "pickup" = "blue", "subcompact" = "purple", "suv" =
  theme_minimal() +
  theme(plot.title = element_text(color = "blue", size = 16, hjust = 0.5))
```



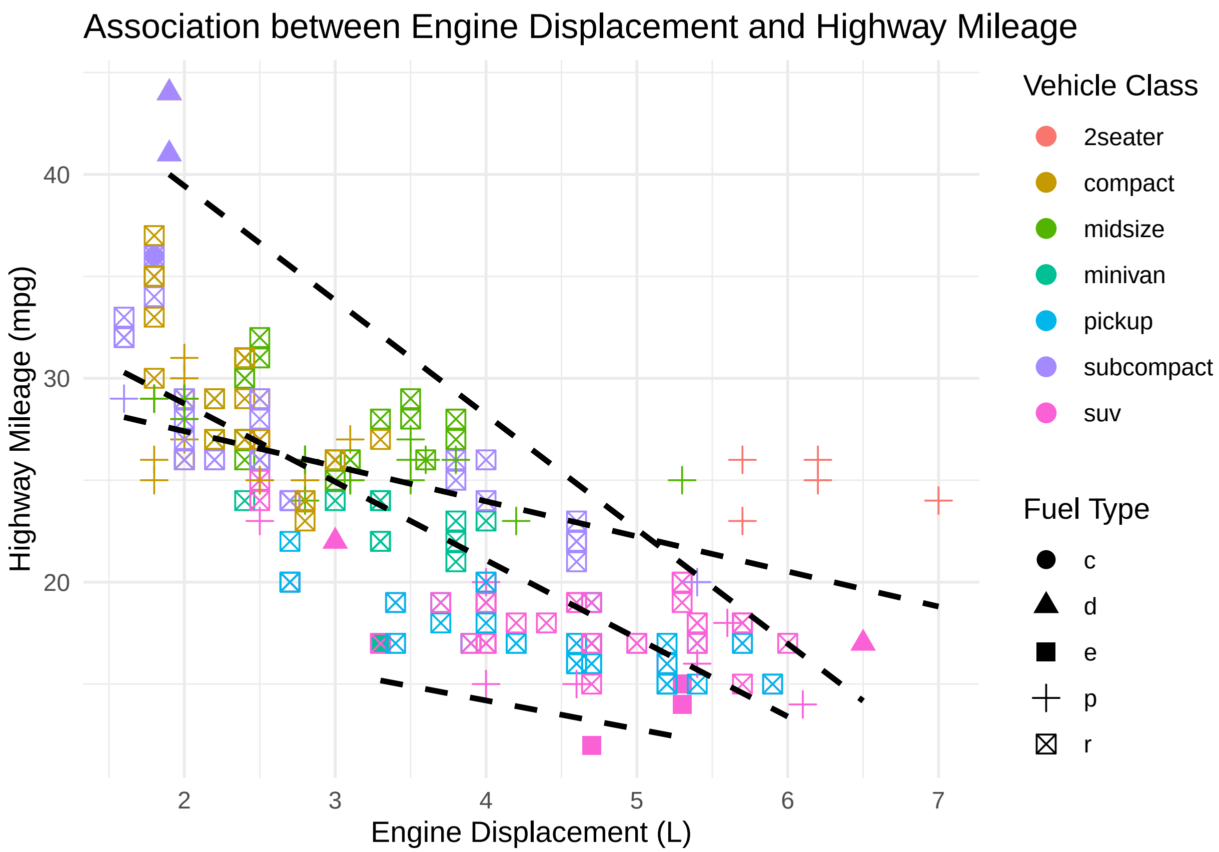
```r
# Stacked bar chart with proportions
df_proportions <- df_complete
df_proportions$proportion <- df_proportions$n / ave(df_proportions$n, df_proportions$drv, FUN = sum)

ggplot(df_proportions, aes(x = drv, y = proportion, fill = class)) +
  geom_bar(stat = "identity", position = "fill", color = "black") +
  labs(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" =
  theme_minimal()
```



Scatter Plot with Smoothing

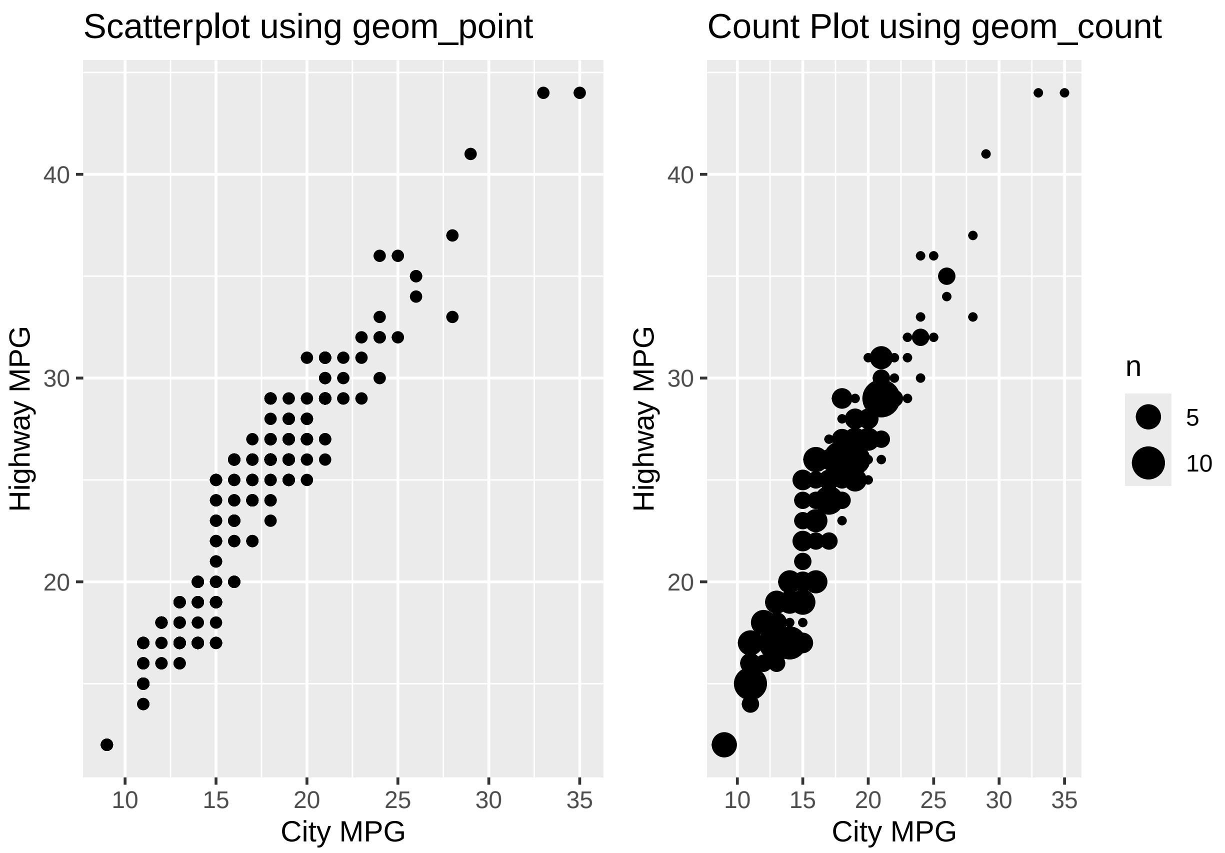
```
# Scatter plot to show the association between engine displacement and highway mileage
ggplot(mpg, aes(x = displ, y = hwy, shape = fl, color = class)) +
  geom_point(size = 3) +
  geom_smooth(method = "lm", linetype = "dashed", color = "black", se = FALSE) +
  labs(title = "Association between Engine Displacement and Highway Mileage", x = "Engine Displacement
  theme_minimal()
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



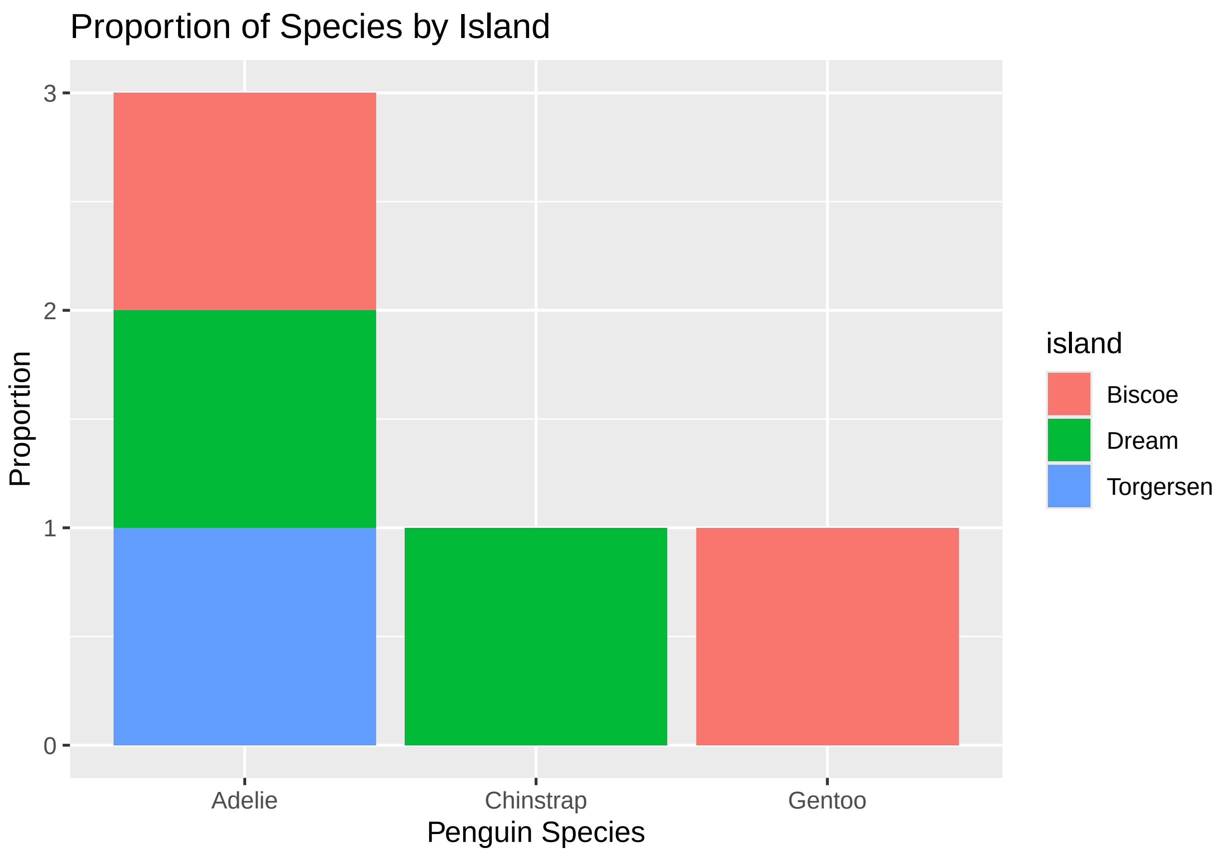
Task 3: Comparison of geom_point() and geom_count()

```
# Create geom_point plot
point_plot <- ggplot(data = mpg, aes(x = cty, y = hwy)) +
              geom_point() +
              labs(title = "Scatterplot using geom_point", x = "City MPG", y = "Highway MPG")

# Create geom_count plot
count_plot <- ggplot(data = mpg, aes(x = cty, y = hwy)) +
              geom_count() +
              labs(title = "Count Plot using geom_count", x = "City MPG", y = "Highway MPG")

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



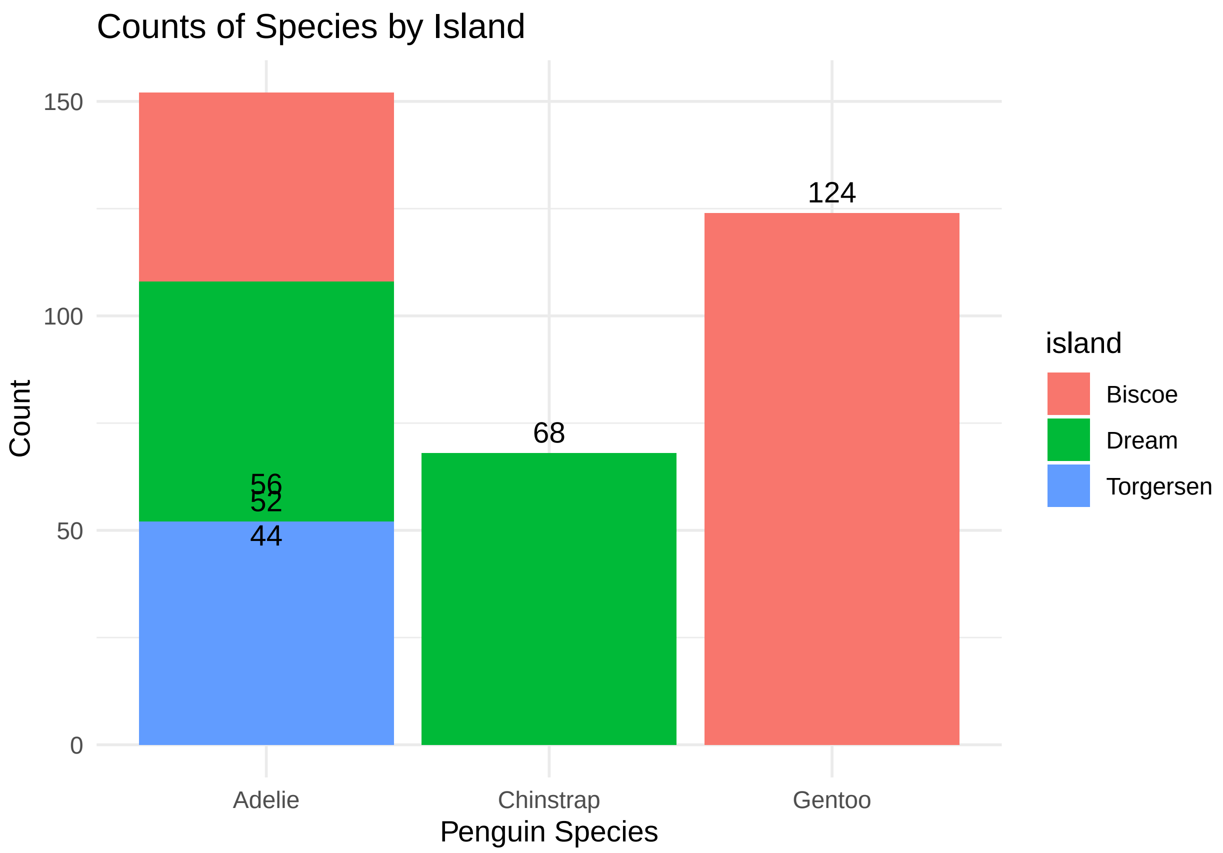
Task 4

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

# Initial bar plot for proportion
ggplot(data = penguins, aes(fill = island, x = species)) +
geom_bar(aes(y = after_stat(prop))) +
labs(title = "Proportion of Species by Island", x = "Penguin Species", y = "Proportion")
```



```
# Absolute counts with labels
ggplot(data = penguins, aes(fill = island, x = species)) +
geom_bar(position = "stack") +
geom_text(stat = "count", aes(label = ..count..), vjust = -0.5) +
labs(title = "Counts of Species by Island", x = "Penguin Species", y =

 "Count") +
theme_minimal()
```

```
## Warning: The dot-dot notation (‘..count..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(count)‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

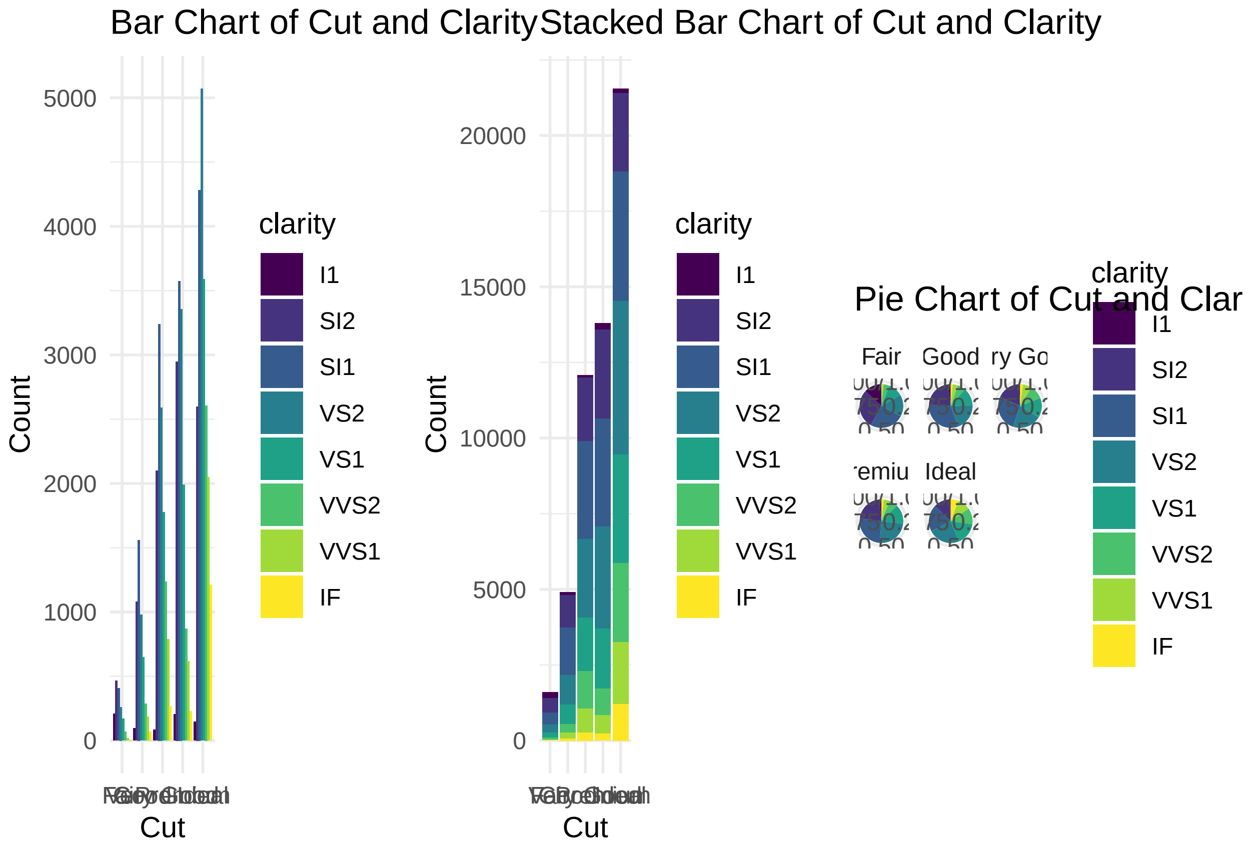


Task 5

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

# Basic bar chart
bar_plot <- ggplot(diamonds, aes(x = cut, fill = clarity)) +
            geom_bar(position = "dodge") +
            labs(title = "Bar Chart of Cut and Clarity", x = "Cut", y = "Count") +
            theme_minimal()
```

```
# Stacked bar chart
stacked_bar_plot <- ggplot(diamonds, aes(x = cut, fill = clarity)) +
                    geom_bar(position = "stack") +
                    labs(title = "Stacked Bar Chart of Cut and Clarity", x = "Cut", y = "Count") +
                    theme_minimal()
```

```
# Pie chart
pie_chart <- ggplot(diamonds, aes(x = "", fill = clarity)) +
             geom_bar(width = 1, position = "fill") +
             coord_polar("y") +
             facet_wrap(~cut) +
             labs(title = "Pie Chart of Cut and Clarity") +
            theme_minimal() +
            theme(axis.title.x = element_blank(), axis.title.y = element_blank())

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


Bar Chart of Cut and Clarity
Count
Cut
Fair
Good
Very Good
Premium
Ideal
clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF
Stacked Bar Chart of Cut and Clarity
Count
Cut
clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF
Pie Chart of Cut and Clar
Fair
Good
ry Go
emiu
Ideal
clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF