1. **Create a bar plot to visualize the frequency of different car brands in the mtcars dataset.**

data(mtcars)

str(mtcars)

brand\_counts <- table(mtcars$carb)

barplot(brand\_counts,

main = "Frequency of Car Brands in mtcars Dataset",

xlab = "Car Brands",

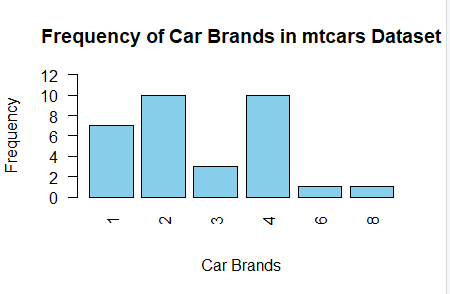
ylab = "Frequency",

col = "skyblue",

border = "black",

ylim = c(0, max(brand\_counts) + 2),

las = 2) # Rotate x-axis labels for better readability if needed



1. **Using the iris dataset, generate a scatter plot of sepal length vs. sepal width. Color the points by species.**

data(iris)

plot(iris$Sepal.Length, iris$Sepal.Width,

col = iris$Species,

pch = 19,

main = "Sepal Length vs. Sepal Width",

xlab = "Sepal Length",

ylab = "Sepal Width")

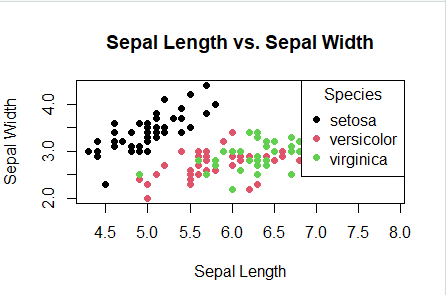
legend("topright",

legend = levels(iris$Species),

col = unique(iris$Species),

pch = 19,

title = "Species")



1. **Construct a histogram of the 'mpg' (miles per gallon) variable from the mtcars dataset. Experiment with different bin widths.**

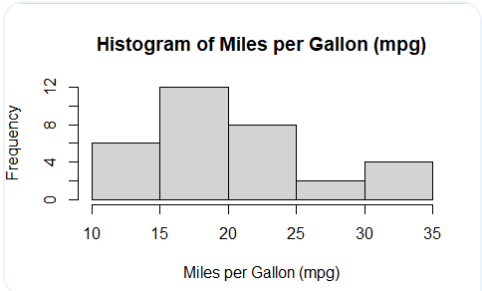
**data(mtcars)**

hist(mtcars$mpg,

main = "Histogram of Miles per Gallon (mpg)",

xlab = "Miles per Gallon (mpg)",

ylab = "Frequency")



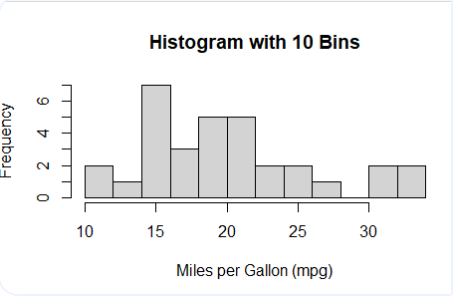
par(mfrow = c(2, 2)) # Create a 2x2 layout for multiple plots

hist(mtcars$mpg, breaks = 10,

main = "Histogram with 10 Bins",

xlab = "Miles per Gallon (mpg)",

ylab = "Frequency")

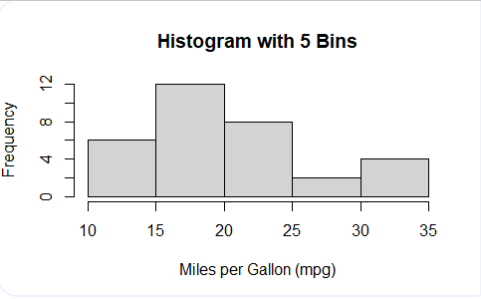


hist(mtcars$mpg, breaks = 5,

main = "Histogram with 5 Bins",

xlab = "Miles per Gallon (mpg)",

ylab = "Frequency")



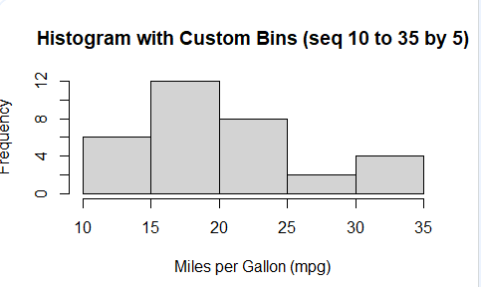
hist(mtcars$mpg, breaks = seq(10, 35, by = 5),

main = "Histogram with Custom Bins (seq 10 to 35 by 5)",

xlab = "Miles per Gallon (mpg)",

ylab = "Frequency")

par(mfrow = c(1, 1))



1. **Create two pie charts side by side using the Titanic dataset: one showing the proportion of survivors vs. non-survivors, and another showing the proportion of passengers in each class.**

data(Titanic)

Titanic\_df <- as.data.frame(Titanic)

total\_passengers <- sum(Titanic\_df$Freq)

total\_survivors <- sum(Titanic\_df[Titanic\_df$Survived == "Yes", ]$Freq)

prop\_survived <- total\_survivors / total\_passengers

prop\_not\_survived <- 1 - prop\_survived

class\_counts <- aggregate(Freq ~ Class, data = Titanic\_df, sum)

class\_counts$prop <- class\_counts$Freq / total\_passengers

par(mfrow = c(1, 2))

pie(c(prop\_survived, prop\_not\_survived),

labels = c("Survived", "Not Survived"),

col = c("lightblue", "salmon"),

main = "Survivors vs. Non-survivors",

clockwise = TRUE)

pie(class\_counts$prop,

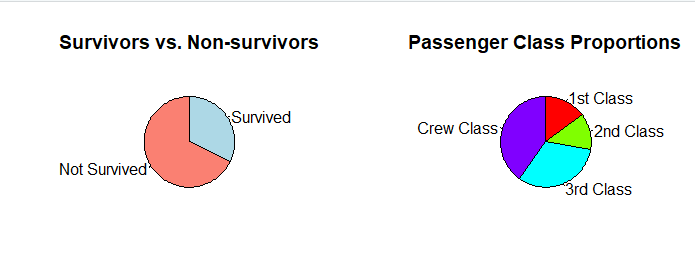
labels = paste(class\_counts$Class, "Class"),

col = rainbow(length(class\_counts$Class)),

main = "Passenger Class Proportions",

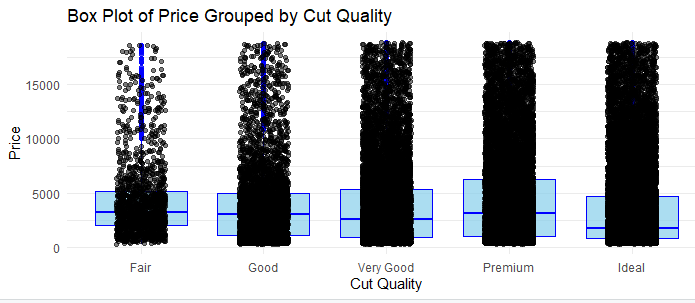
clockwise = TRUE)

par(mfrow = c(1, 1))



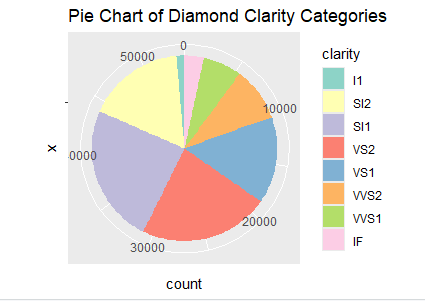
1. **Using the diamonds dataset from ggplot2, create a box plot of price grouped by cut quality. Add individual data points as a jitter plot over the box plot. (Use ggplot2 package only for loading data)**

library(ggplot2)  
  
data(diamonds)  
  
ggplot(diamonds, aes(x = cut, y = price)) +  
  geom\_boxplot(fill = "skyblue", color = "blue", alpha = 0.7) +  # Box plot aesthetics  
  geom\_jitter(position = position\_jitter(width = 0.2), alpha = 0.5) +  # Jitter plot aesthetics  
  labs(x = "Cut Quality", y = "Price", title = "Box Plot of Price Grouped by Cut Quality") +  # Axis and title labels  
  theme\_minimal()



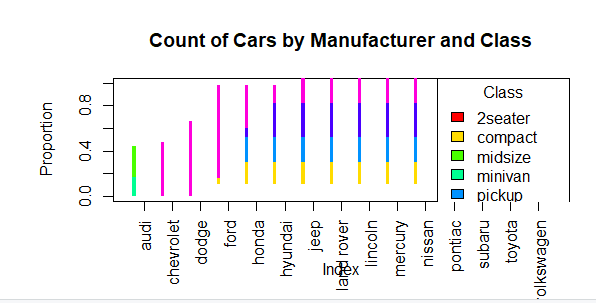
1. **Using the diamonds dataset from ggplot2, create a pie chart of diamond clarity categories. Experiment with different color palettes. (Use ggplot2 package only for loading data)**

library(ggplot2)  
  
data(diamonds)  
  
clarity\_counts <- table(diamonds$clarity)  
clarity\_df <- as.data.frame(clarity\_counts)  
names(clarity\_df) <- c("clarity", "count")  
ggplot(clarity\_df, aes(x = "", y = count, fill = clarity)) +  
  geom\_bar(stat = "identity", width = 1) +  
  coord\_polar("y") +  
  labs(title = "Pie Chart of Diamond Clarity Categories") +  
  scale\_fill\_brewer(palette = "Set3")  # Experiment with different color palettes



1. **Using the mpg dataset, create a grouped and stacked bar chart showing the count of cars by manufacturer, with each bar segment representing different classes of vehicles, and the groups representing transmission type (automatic/manual).**

library(ggplot2)  
data(mpg)  
mpg$manufacturer <- as.factor(mpg$manufacturer)  
mpg$class <- as.factor(mpg$class)  
mpg$trans <- as.factor(ifelse(grepl("auto", mpg$trans), "automatic", "manual"))  
agg\_data <- with(mpg, table(manufacturer, class, trans))  
plot\_grouped\_stacked\_bar <- function(data, group\_colors, bar\_colors, group\_labels, bar\_labels) {  
  bar\_width <- 0.8 / ncol(data)  
  barplot\_heights <- t(apply(data, 1, function(row) row / sum(row)))  
  cum\_heights <- apply(barplot\_heights, 2, cumsum)  
  num\_bars <- ncol(data)  
  num\_groups <- nrow(data)  
    
  par(mar = c(5, 6, 4, 2) + 0.1)  
  plot(0, type = "n", xlim = c(0.5, num\_groups + 0.5), ylim = c(0, 1),  
       xaxt = "n", ylab = "Proportion", main = "Count of Cars by Manufacturer and Class")  
  axis(1, at = 1:num\_groups, labels = group\_labels, las = 2)  
    
  for (i in 1:num\_bars) {  
    for (j in 1:num\_groups) {  
      if (i == 1) {  
        rect(j - 0.4, 0, j - 0.4 + bar\_width, barplot\_heights[j, i], col = bar\_colors[i], border = NA)  
      } else {  
        rect(j - 0.4, cum\_heights[j, i - 1], j - 0.4 + bar\_width, cum\_heights[j, i], col = bar\_colors[i], border = NA)  
      }  
    }  
  }  
    
  legend("topright", legend = bar\_labels, fill = bar\_colors, title = "Class")  
}  
  
# Colors  
group\_colors <- c("grey", "white")  
bar\_colors <- rainbow(ncol(agg\_data))  
plot\_grouped\_stacked\_bar(data = agg\_data,   
                         group\_colors = group\_colors,   
                         bar\_colors = bar\_colors,   
                         group\_labels = rownames(agg\_data),   
                         bar\_labels = colnames(agg\_data))



8. **Create a comparative histogram (two overlaid histograms with transparency) of highway and city mpg from the mpg dataset. (Use ggplot2 package only for loading data)**

library(ggplot2)

data(mpg)

ggplot(mpg, aes(x = hwy)) +

geom\_histogram(aes(fill = "Highway MPG"),

binwidth = 2,

alpha = 0.5,

position = "identity") +

geom\_histogram(aes(x = cty, fill = "City MPG"),

binwidth = 2,

alpha = 0.5,

position = "identity") +

labs(title = "Comparative Histogram of Highway and City MPG",

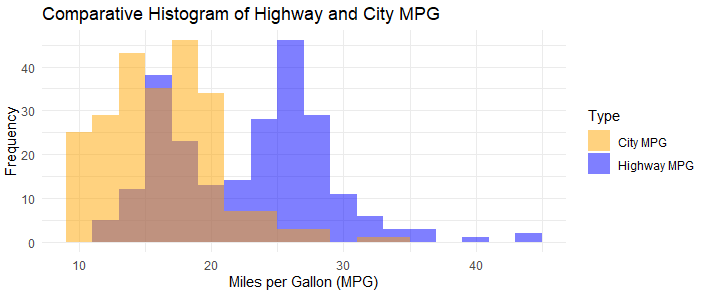
x = "Miles per Gallon (MPG)",

y = "Frequency",

fill = "Type") +

scale\_fill\_manual(values = c("Highway MPG" = "blue", "City MPG" = "orange")) + # Set fill colors

theme\_minimal()



9**. Using the airquality dataset, create a box plot of Ozone levels grouped by Month. Color the boxes by temperature ranges (you'll need to create these categories).**

library(ggplot2)

data(airquality)

str(airquality)

# Assuming temperature ranges: Low (<= 70), Medium (71-80), High (> 80)

airquality$temp\_range <- cut(airquality$Temp, breaks = c(-Inf, 70, 80, Inf),

labels = c("Low", "Medium", "High"), right = FALSE)

# Convert Month to a factor with ordered levels for proper ordering in plots

airquality$Month <- factor(airquality$Month, levels = 5:9, labels = c("May", "June", "July", "August", "September"))

# Create a box plot of Ozone levels grouped by Month and colored by temperature range

ggplot(airquality, aes(x = Month, y = Ozone, fill = temp\_range)) +

geom\_boxplot(alpha = 0.8) + # Add box plot with transparency

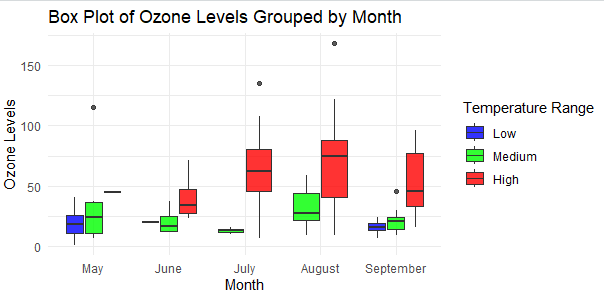
labs(title = "Box Plot of Ozone Levels Grouped by Month",

x = "Month", y = "Ozone Levels",

fill = "Temperature Range") +

scale\_fill\_manual(values = c("Low" = "blue", "Medium" = "green", "High" = "red")) + # Custom color palette

theme\_minimal()



10.**Create an animated scatter plot using the gapminder dataset, showing how the relationship between GDP per capita and life expectancy changes over time. Use continents for color and population for point size.**

install.packages("gapminder")  
install.packages("ggplot2")  
install.packages("gganimate")  
install.packages("dplyr")  
install.packages("gifski")  # Required for rendering animations  
install.packages("png")     # Required for rendering animations  
  
library(gapminder)  
library(ggplot2)  
library(gganimate)  
library(dplyr)  
  
data(gapminder)  
  
# Select required variables  
gapminder\_data <- gapminder %>%  
  select(country, year, continent, gdpPercap, lifeExp, pop) %>%  
  filter(![is.na](http://is.na)(gdpPercap), ![is.na](http://is.na)(lifeExp))  # Remove NA values if any  
  
# Convert continent to factor for categorical coloring  
gapminder\_data$continent <- factor(gapminder\_data$continent)  
  
# Create animated scatter plot  
animated\_plot <- ggplot(gapminder\_data, aes(x = gdpPercap, y = lifeExp, size = pop, color = continent)) +  
  geom\_point(alpha = 0.7) +  
  scale\_size\_continuous(range = c(1, 15)) +  # Adjust size range as needed  
  scale\_x\_log10() +  # Use log scale for GDP per capita for better visualization  
  labs(title = "GDP per Capita vs. Life Expectancy by Continent",  
       x = "GDP per Capita (log scale)",  
       y = "Life Expectancy",  
       color = "Continent",  
       size = "Population",  
       caption = 'Year: {frame\_time}') +  
  transition\_time(year) +  
  ease\_aes('linear') +  
  theme\_minimal()  
  
# Render and display the animation  
animate(animated\_plot, nframes = 200, fps = 10, width = 800, height = 600, renderer = gifski\_renderer())

