Lab ggplot2(ggplot -themes)

Pradip Basnet

code converts the dose variable in the ToothGrowth dataset from a numeric or integer type to a factor type

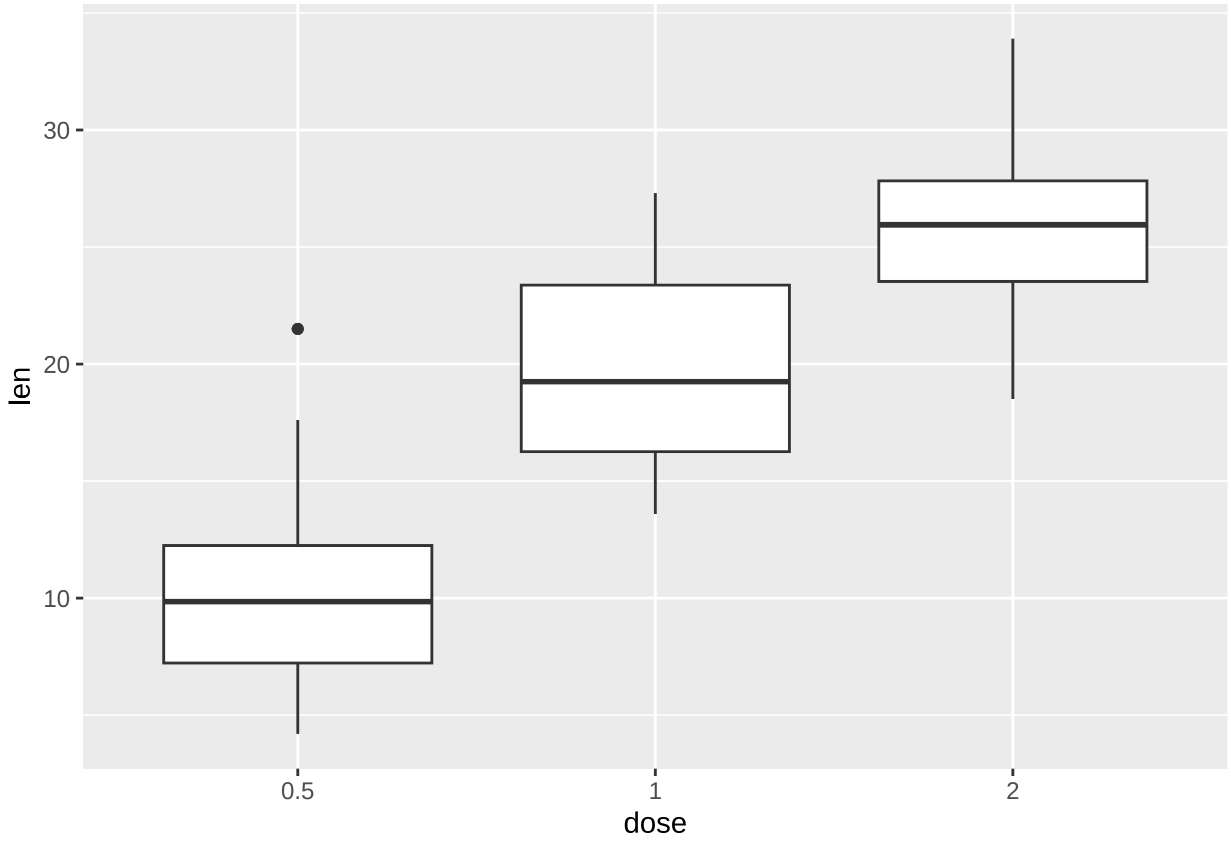
```
ToothGrowth$dose <- as.factor(ToothGrowth$dose)
```

importing/loading the necessary packages

```
library(ggplot2)
```

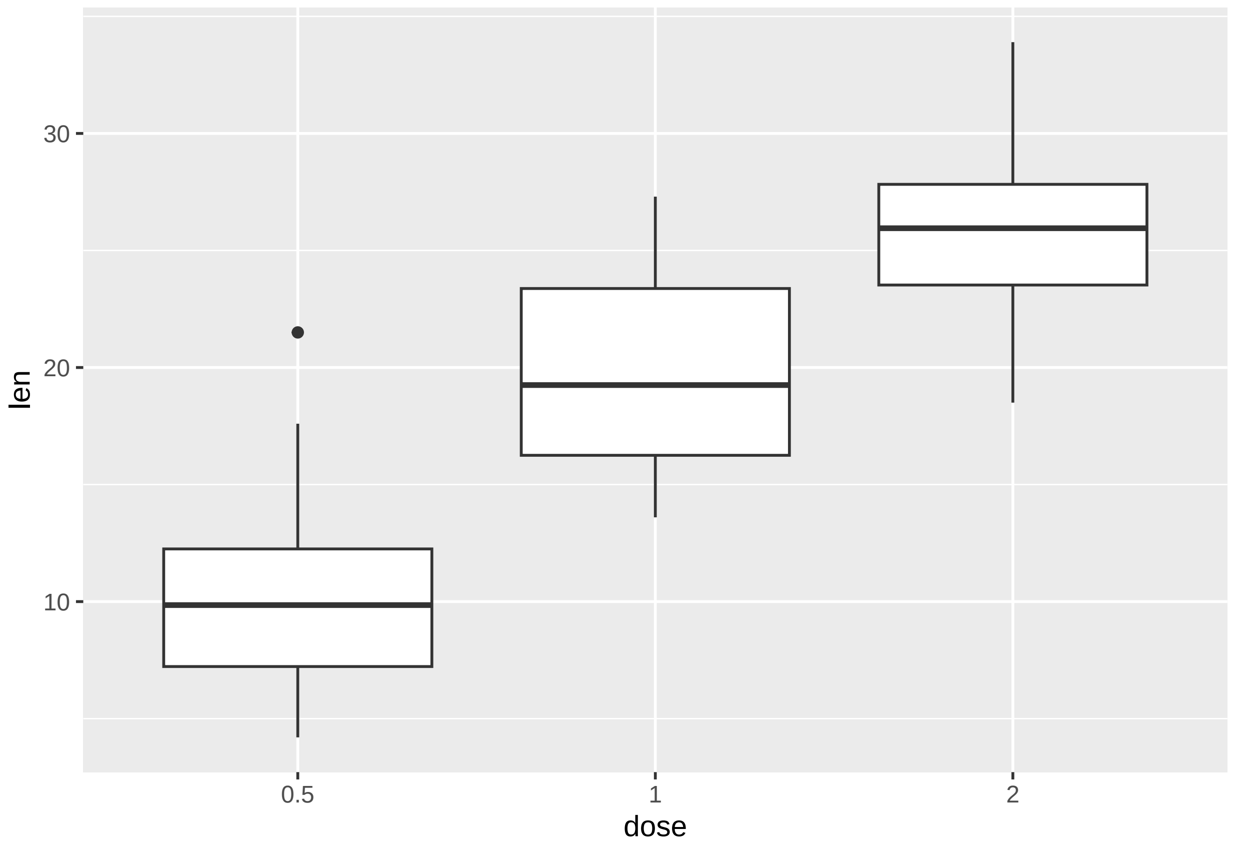
code creates and displays a boxplot to visualize the distribution of tooth length (len) across different dose levels (dose) in the ToothGrowth dataset

```
p <- ggplot(ToothGrowth, aes(x=dose, y=len)) + geom_boxplot()
print(p)
```



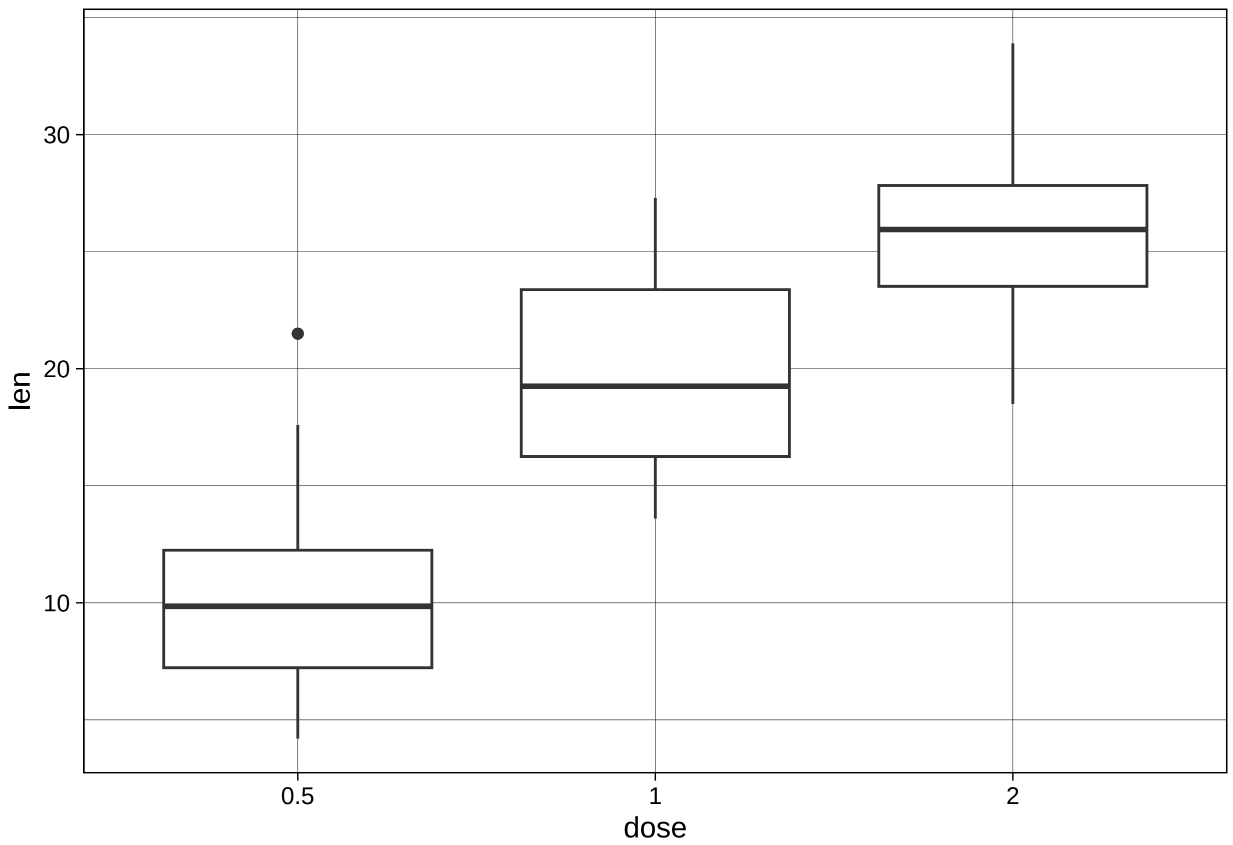
applying the default gray theme (theme_gray()), giving the plot a gray background with white grid lines

```r
p + theme_gray()
```



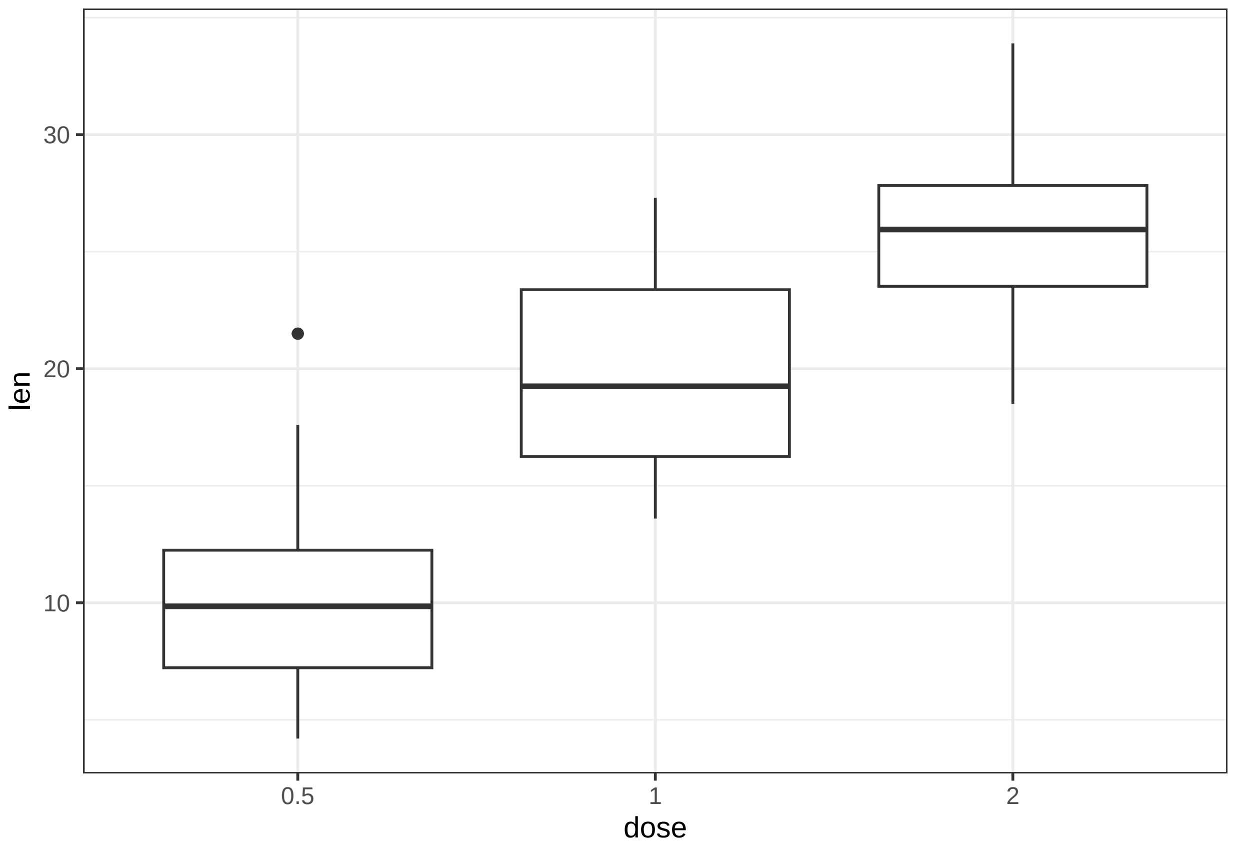
code changes the visual appearance of the plot by applying the theme_linedraw(), which features a simple and clear design with a white background and black grid lines.

```r
p + theme_linedraw()
```

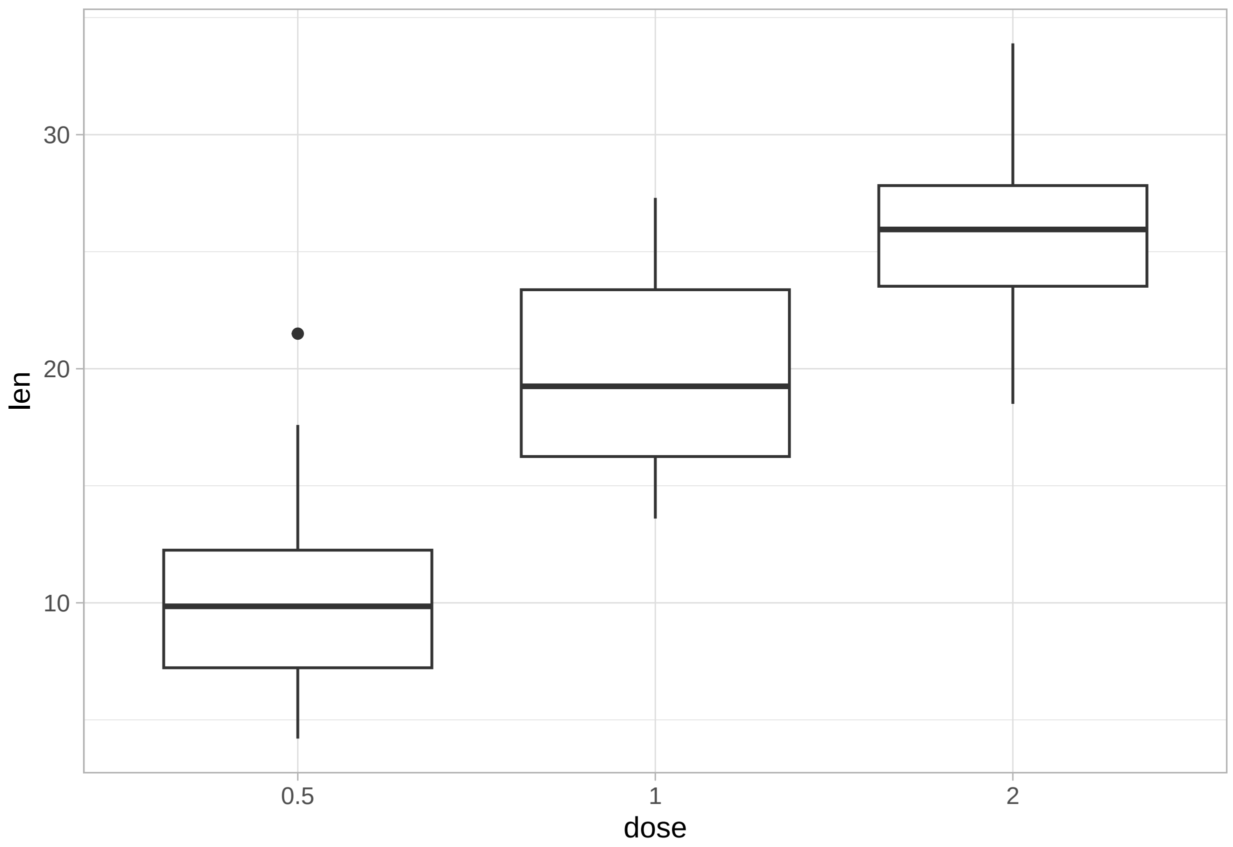


code updates the plot with the theme_bw(), giving it a clean, white background with black grid lines and borders, suitable for a more formal or publication-ready appearance.

```
p + theme_bw()
```

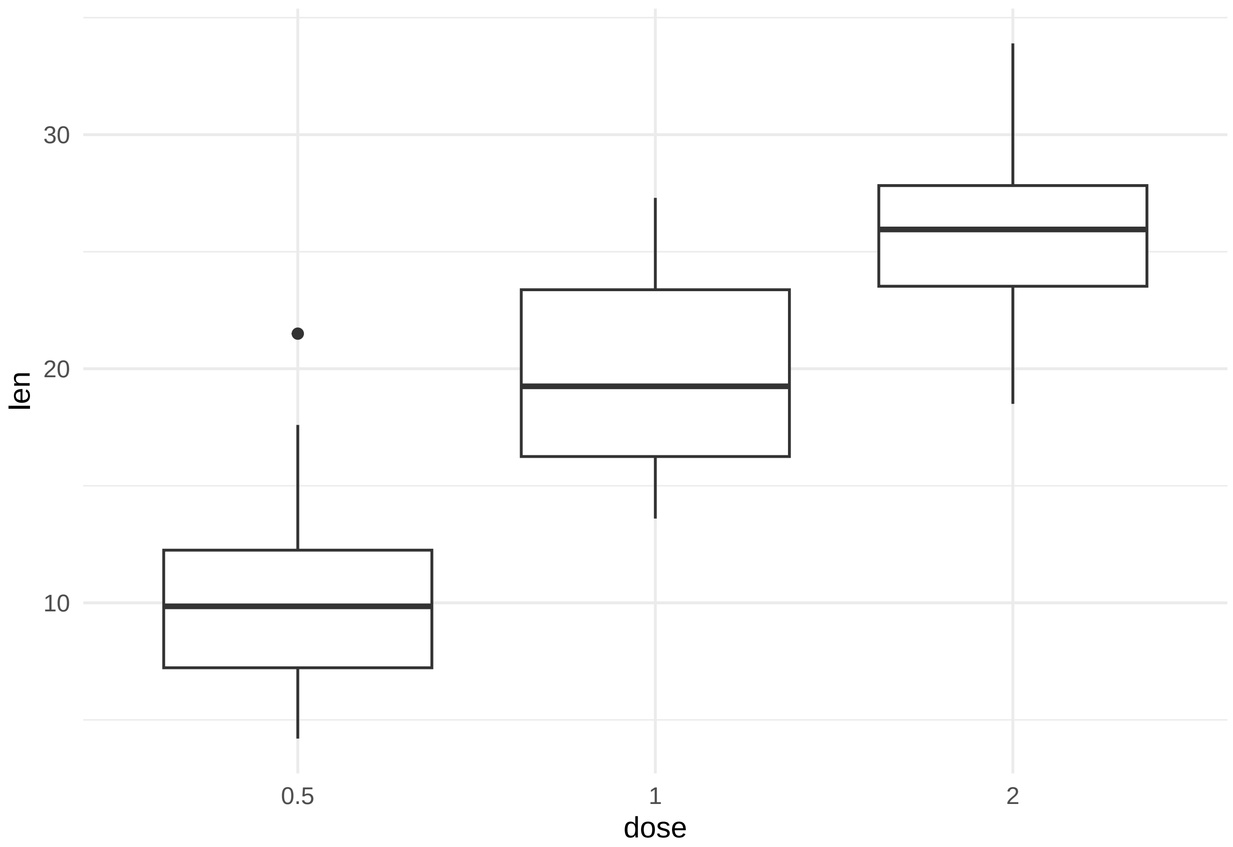


```
p + theme_light()
```



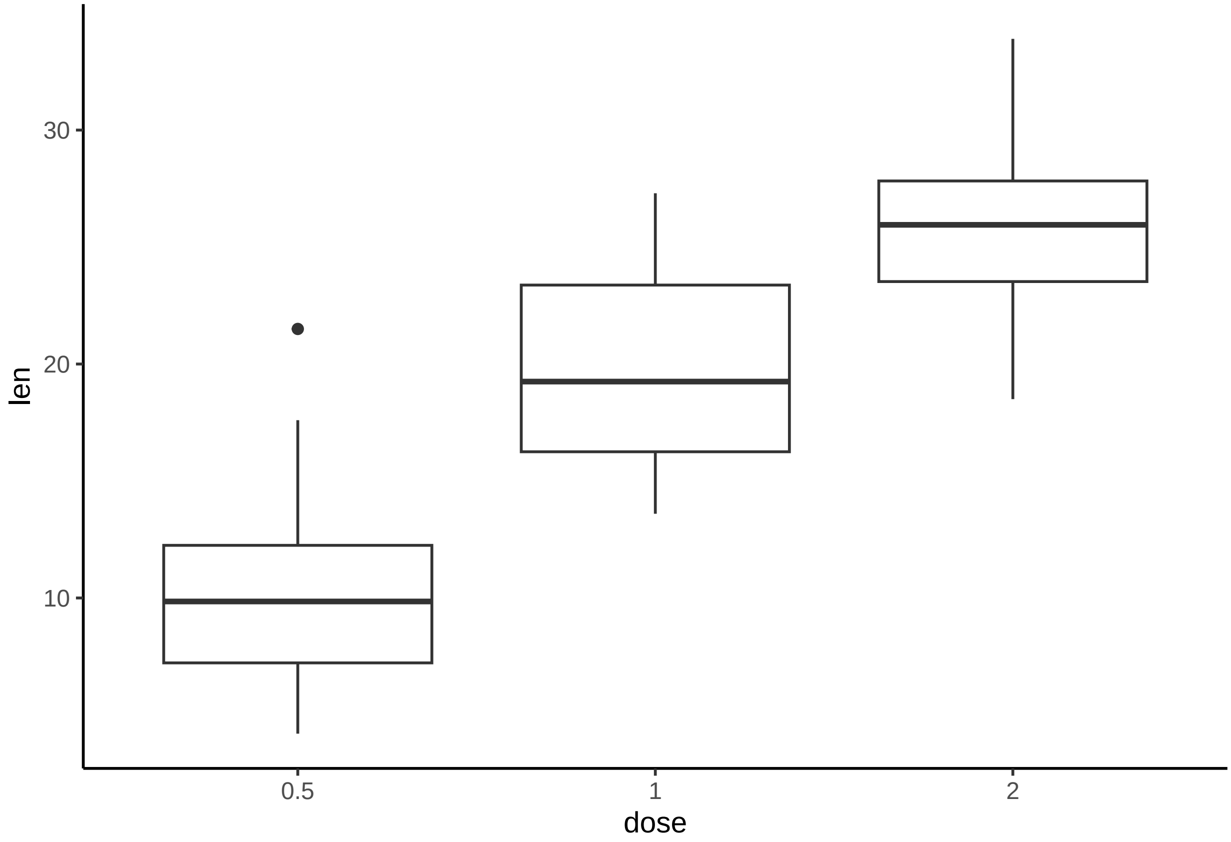
code updates the plot with the theme_minimal(), giving it a clean, stripped-down appearance with a focus on the data and minimal visual clutter

```r
p + theme_minimal()
```



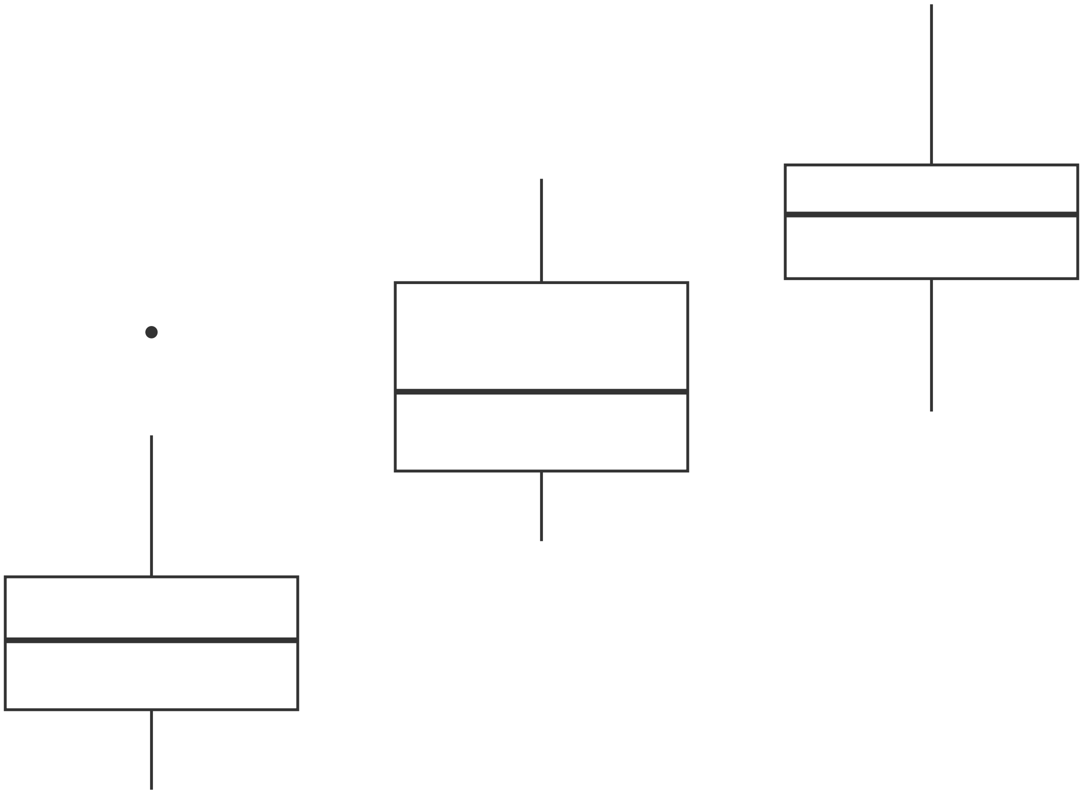
code updates the plot with the theme_classic(), giving it a simple, traditional appearance with a white background, no grid lines, and clear axis lines

```
p + theme_classic()
```



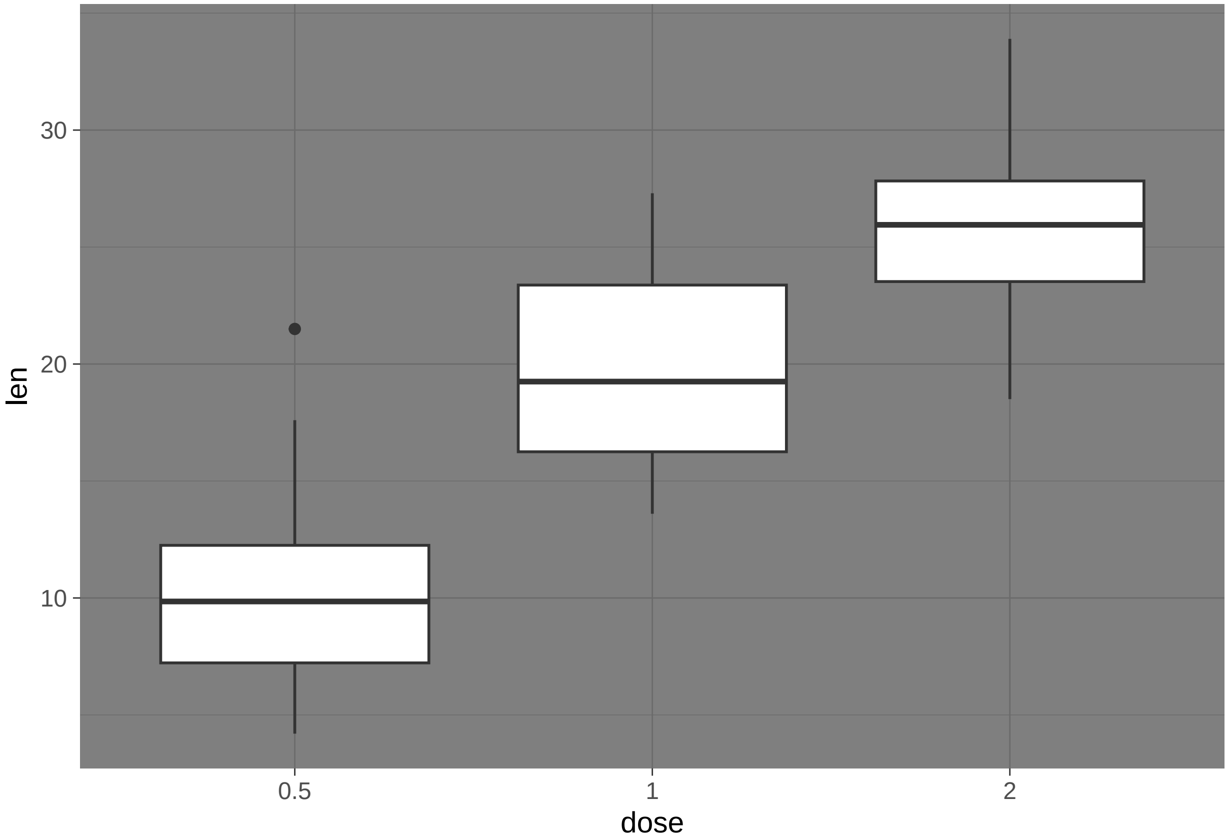
code updates the plot with the theme_void(), resulting in a plot with no background, axis lines, labels, or grid lines, showing only the data elements.

```
p + theme_void()
```



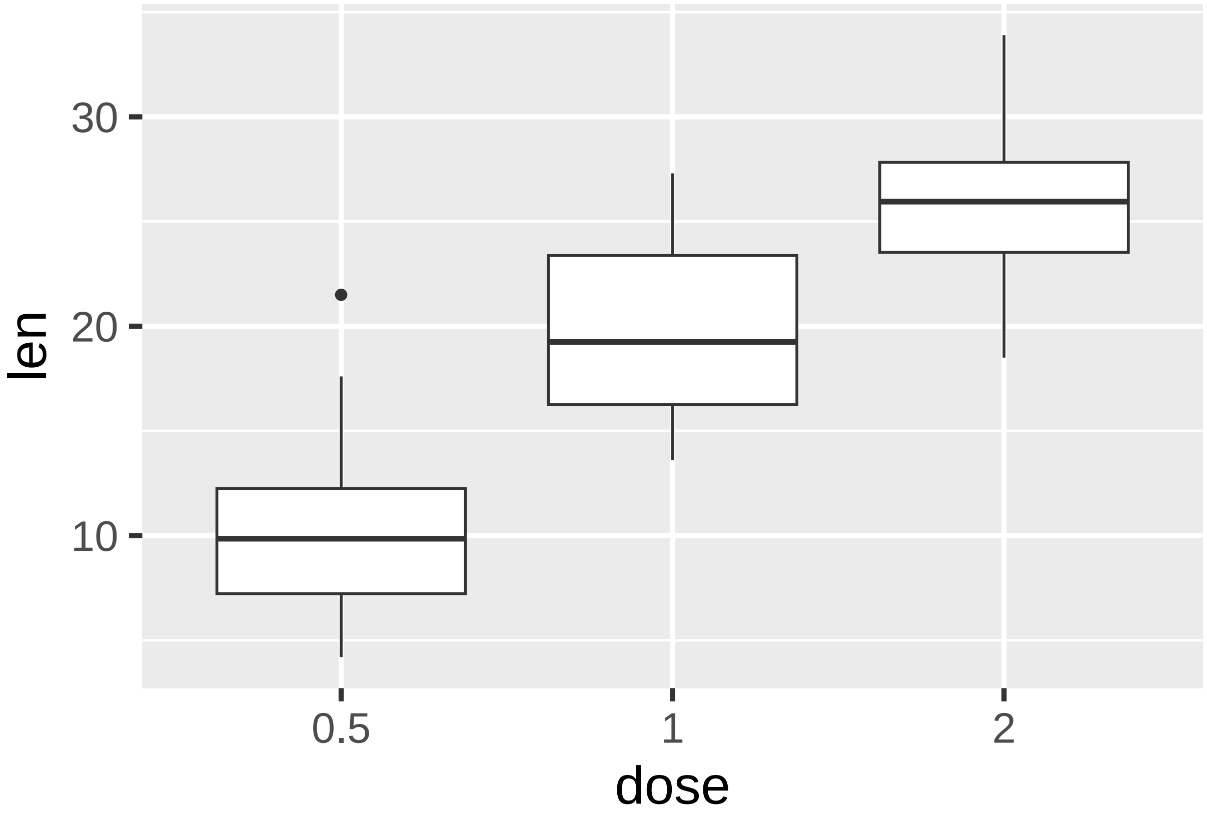
code updates the plot with the theme_dark(), resulting in a plot with a dark background and light-colored text and grid lines for a high-contrast, visually striking appearance.

```
p + theme_dark()
```



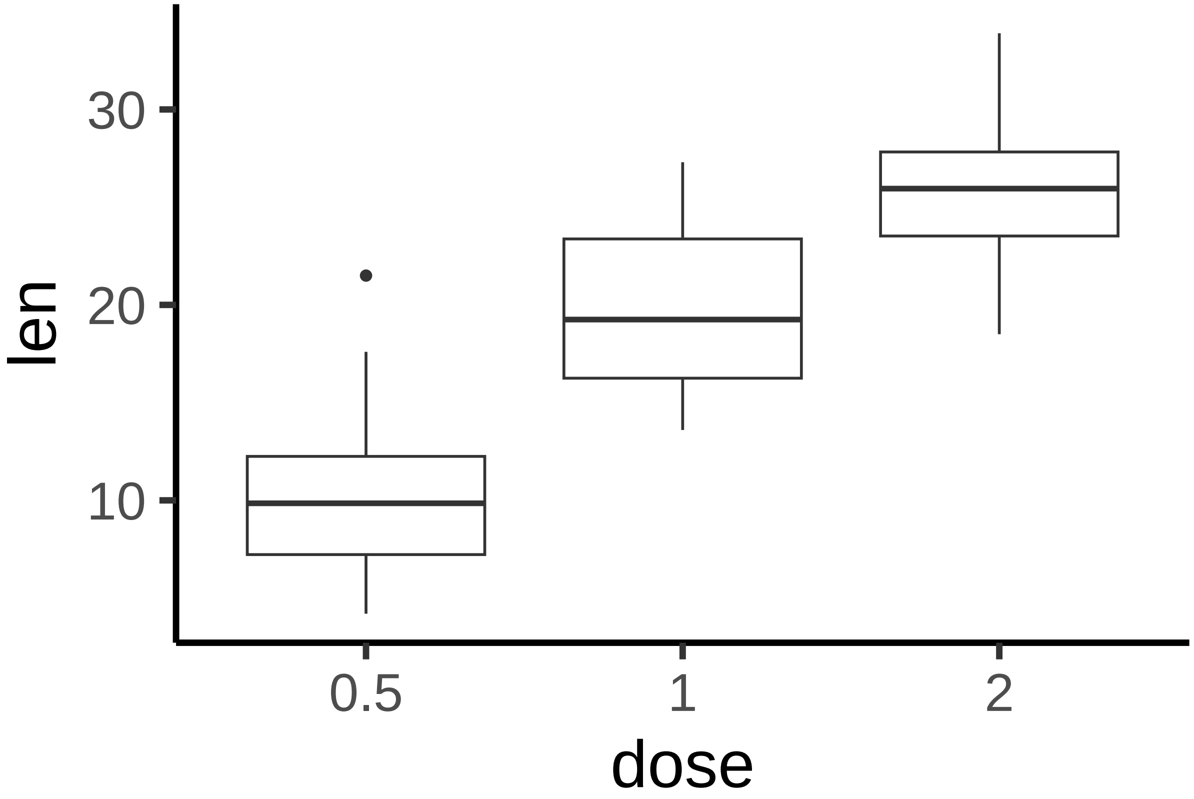
code sets the default ggplot2 theme to theme_gray with a base font size of 20, and then creates a boxplot of tooth length (len) across different dose levels (dose) using this theme. This ensures that all text elements in the plot are larger due to the increased base font size.

```
theme_set(theme_gray(base_size = 20))
ggplot(ToothGrowth,aes(x=dose,y=len)) + geom_boxplot()
```



code creates a boxplot of tooth length across different dose levels using a classic theme with a white background, and increases the font size of text elements to 25.

```
ggplot(ToothGrowth, aes(x=dose,y=len)) + geom_boxplot() +
  theme_classic(base_size = 25)
```

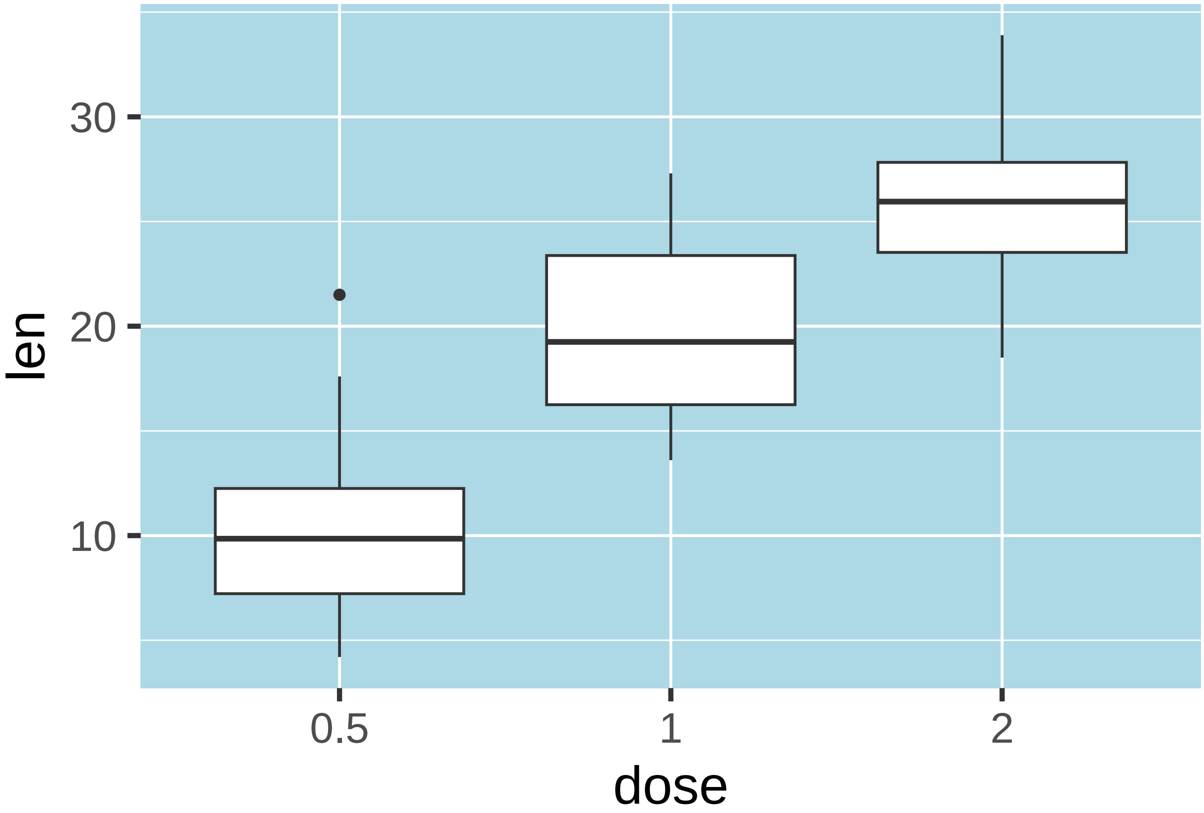


code customizes the plot by setting the panel background to light blue, and adjusting the major and minor grid lines to be solid, white, with specific thicknesses. This gives the plot a distinctive appearance with a light blue background and contrasting white grid lines.

```
p + theme(
  panel.background = element_rect(fill = "lightblue",colour = "lightblue", size = 0.5,linetype = "solid
  panel.grid.major = element_line(size=0.5,linetype = "solid",colour = "white"),
  panel.grid.minor = element_line(size = 0.25, linetype = "solid",colour = "white")
)
```

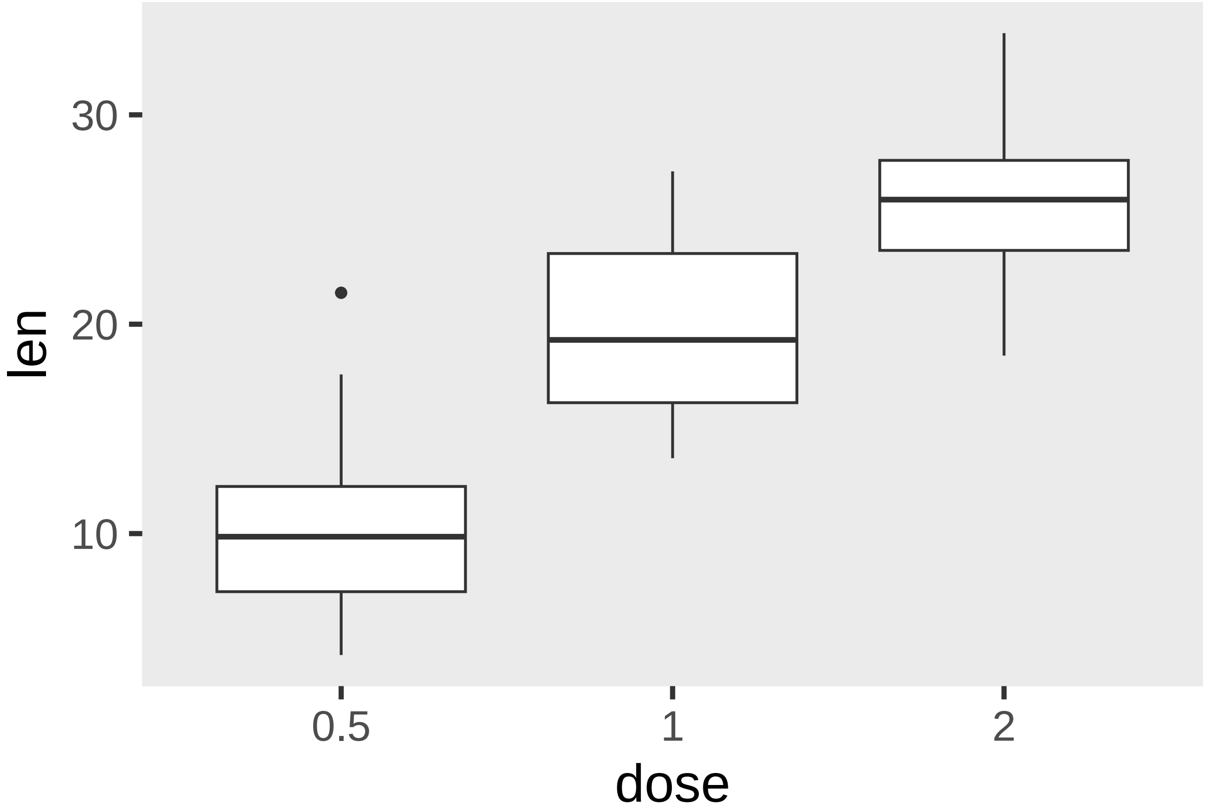
```
## Warning: The ‘size‘ argument of ‘element_rect()‘ is deprecated as of ggplot2 3.4.0.
## i Please use the ‘linewidth‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## Warning: The ‘size‘ argument of ‘element_line()‘ is deprecated as of ggplot2 3.4.0.
## i Please use the ‘linewidth‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



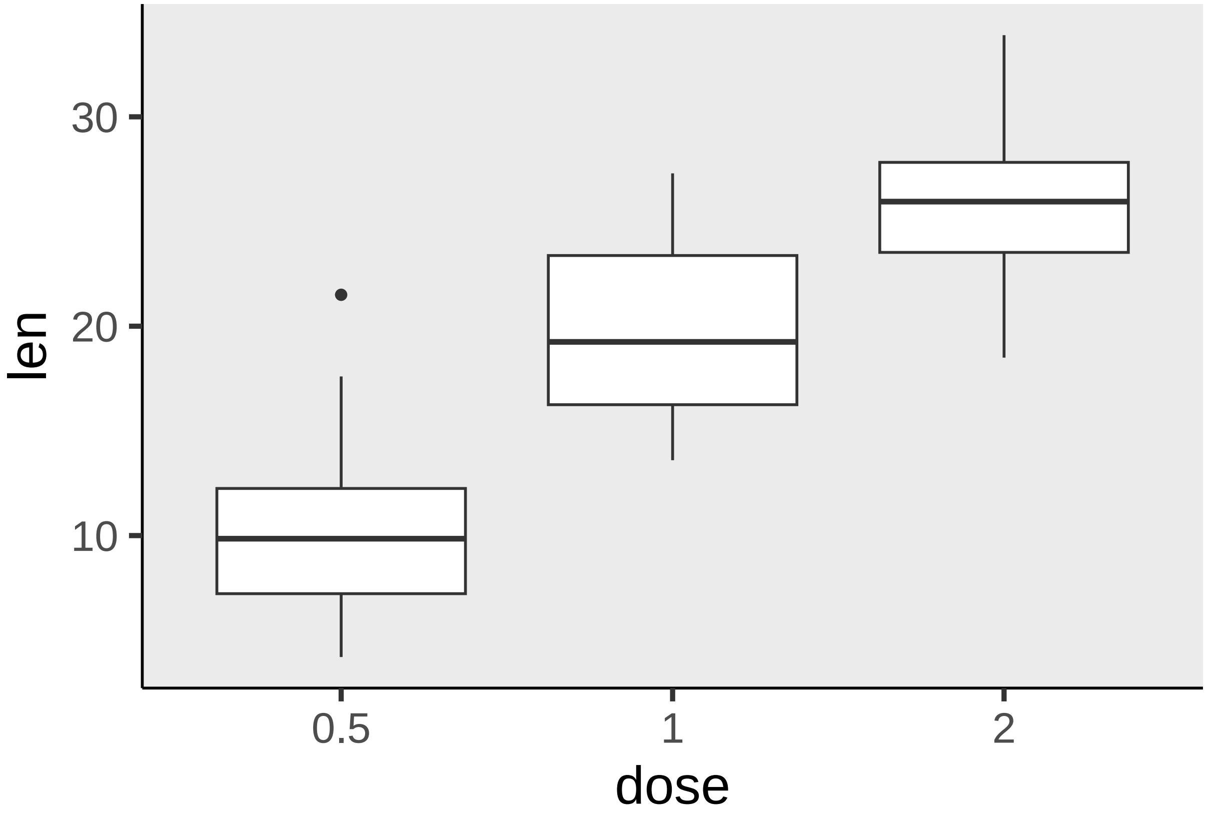
code customizes the plot p by removing the panel border and both major and minor grid lines, resulting in a cleaner plot with fewer visual distractions

```
p + theme(panel.border = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank())
```



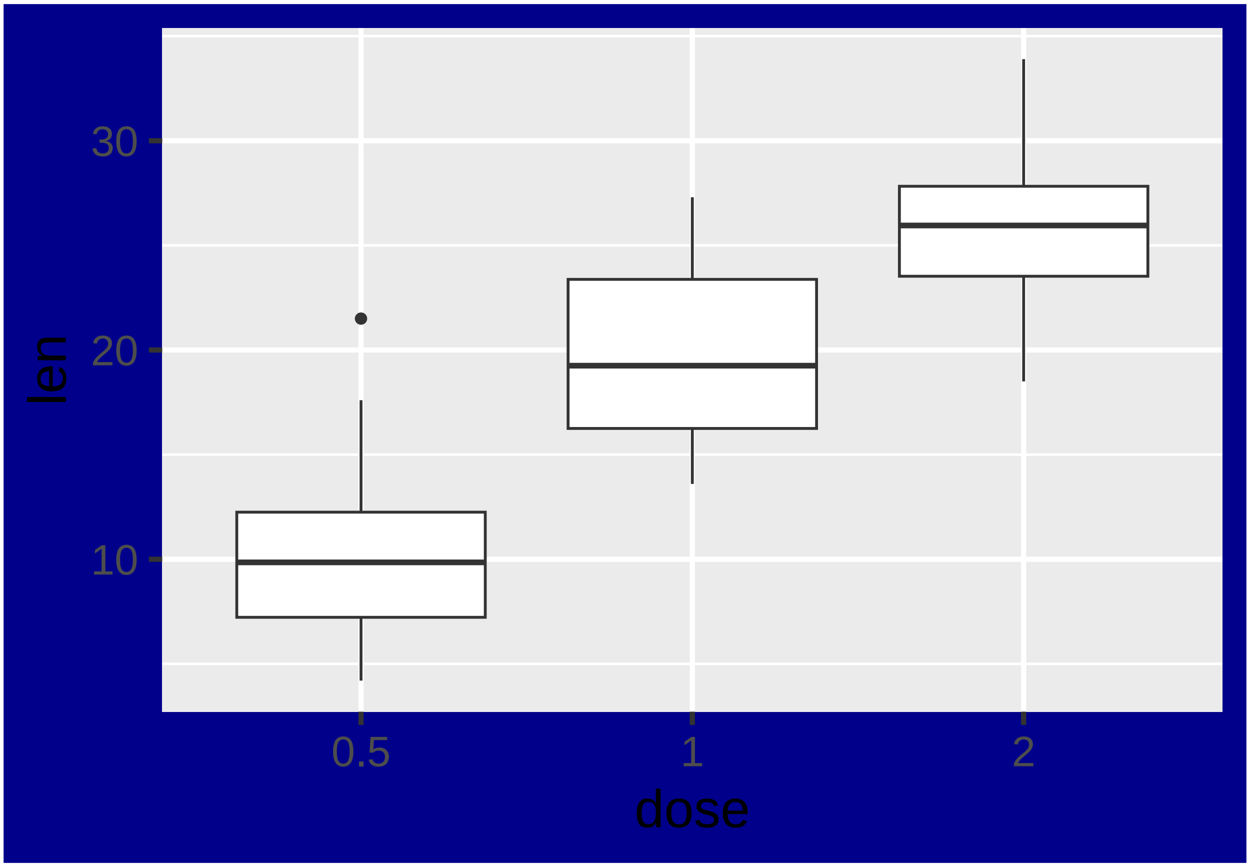
The code customizes the plot by removing the panel border, major and minor grid lines, and then adds solid black axis lines with a thickness of 0.5, resulting in a plot with a clean appearance and distinct axis lines.

```r
p + theme(panel.border = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          axis.line = element_line(size = 0.5,linetype = "solid",colour = "black"))
```



The code changes the background color of the entire plot area to dark blue, providing a visually distinct backdrop behind the plot and its elements

```
p + theme(plot.background = element_rect(fill = "darkblue"))
```

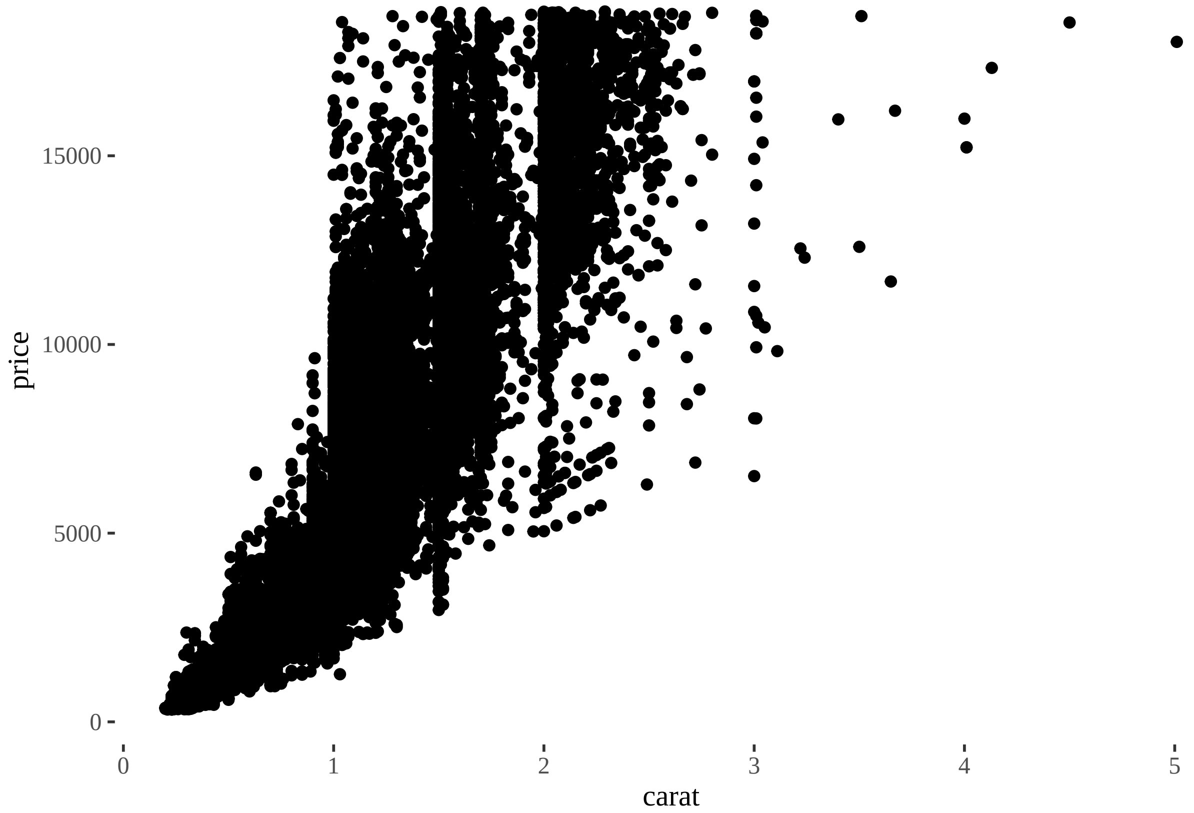


installing the necessary packages

```
#install.packages("ggthemes")
library(ggthemes)
```

The code creates a scatter plot of diamond carat versus price and applies the theme_tufte(), resulting in a minimalist plot with a clean, simple design that emphasizes the data without distracting background elements or grid lines

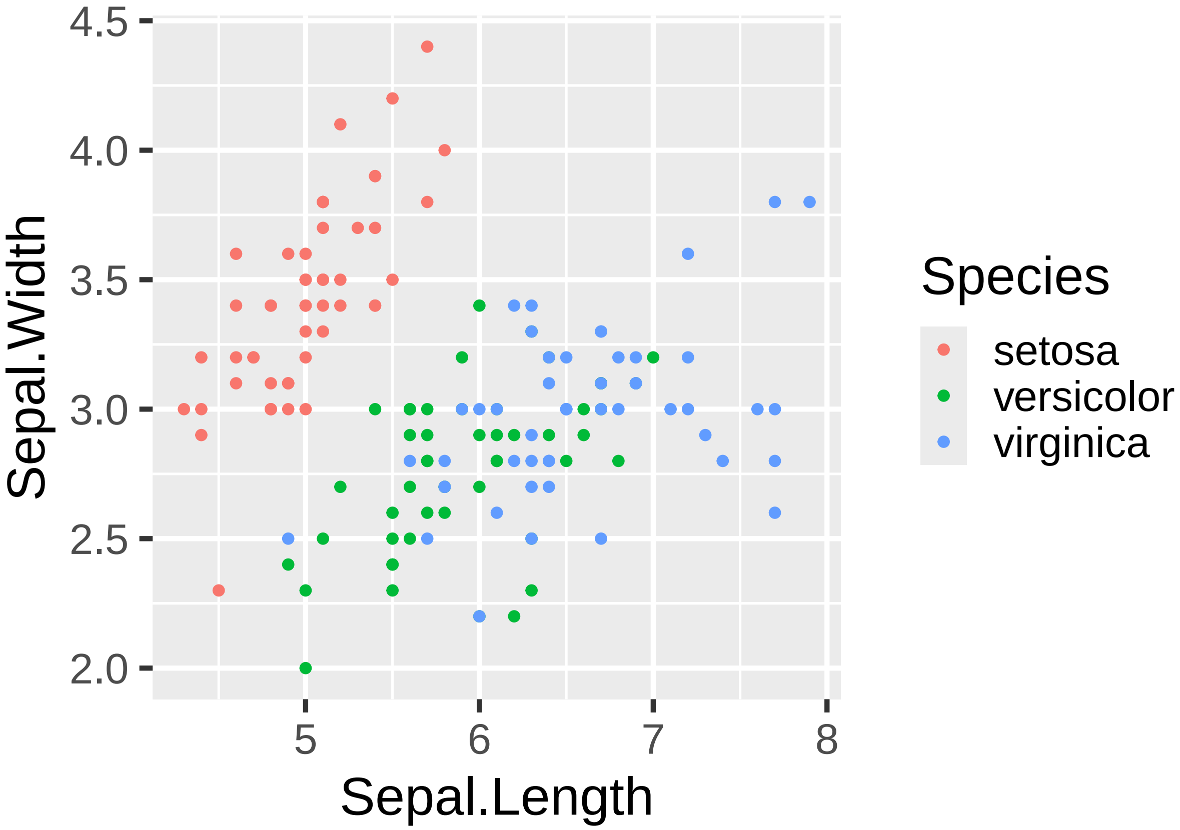
```
ggplot(diamonds,aes(carat,price)) +
  geom_point() +
  theme_tufte()
```



The code creates a scatter plot of sepal length versus sepal width for different species in the iris dataset, with points colored by species. The plot is then displayed using the variable p

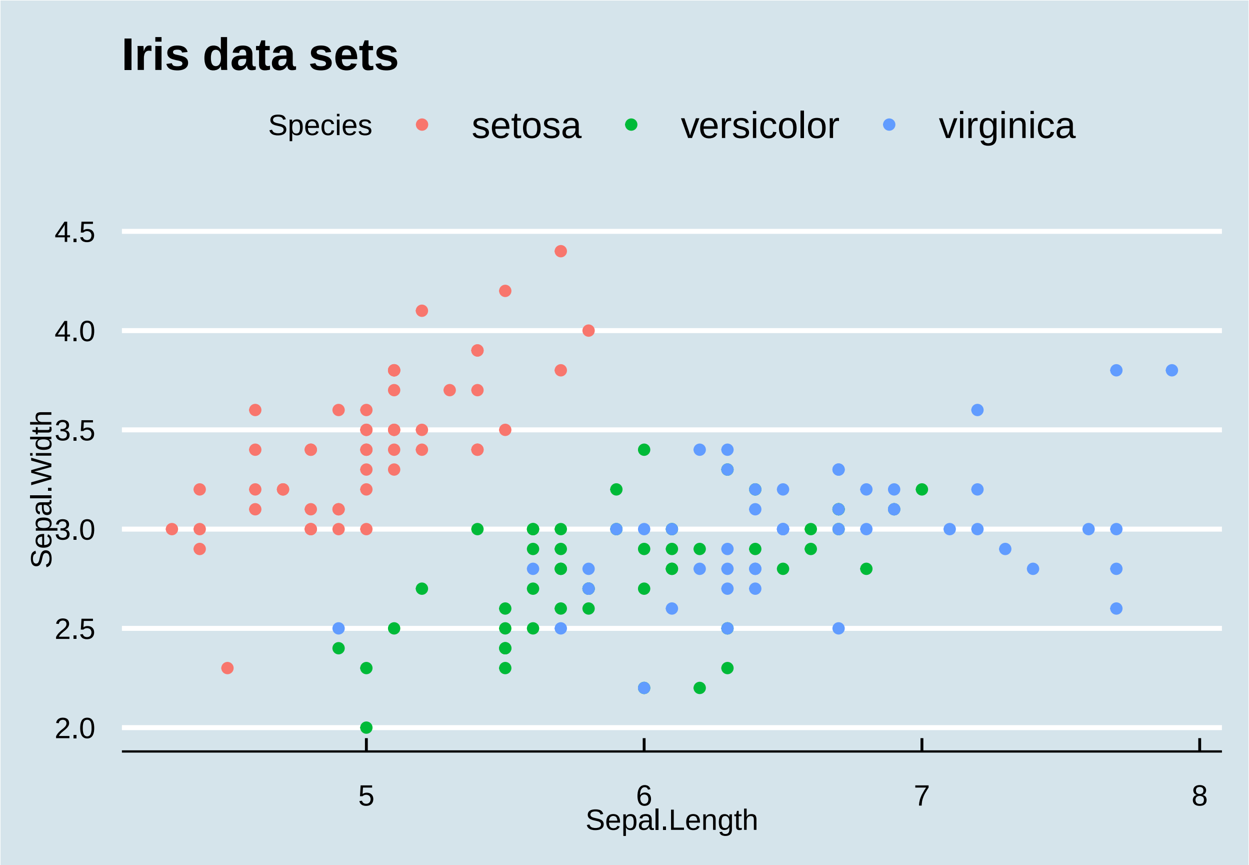
```
p <- ggplot(iris, aes(Sepal.Length, Sepal.Width, colour = Species)) +
 geom_point()

p
```



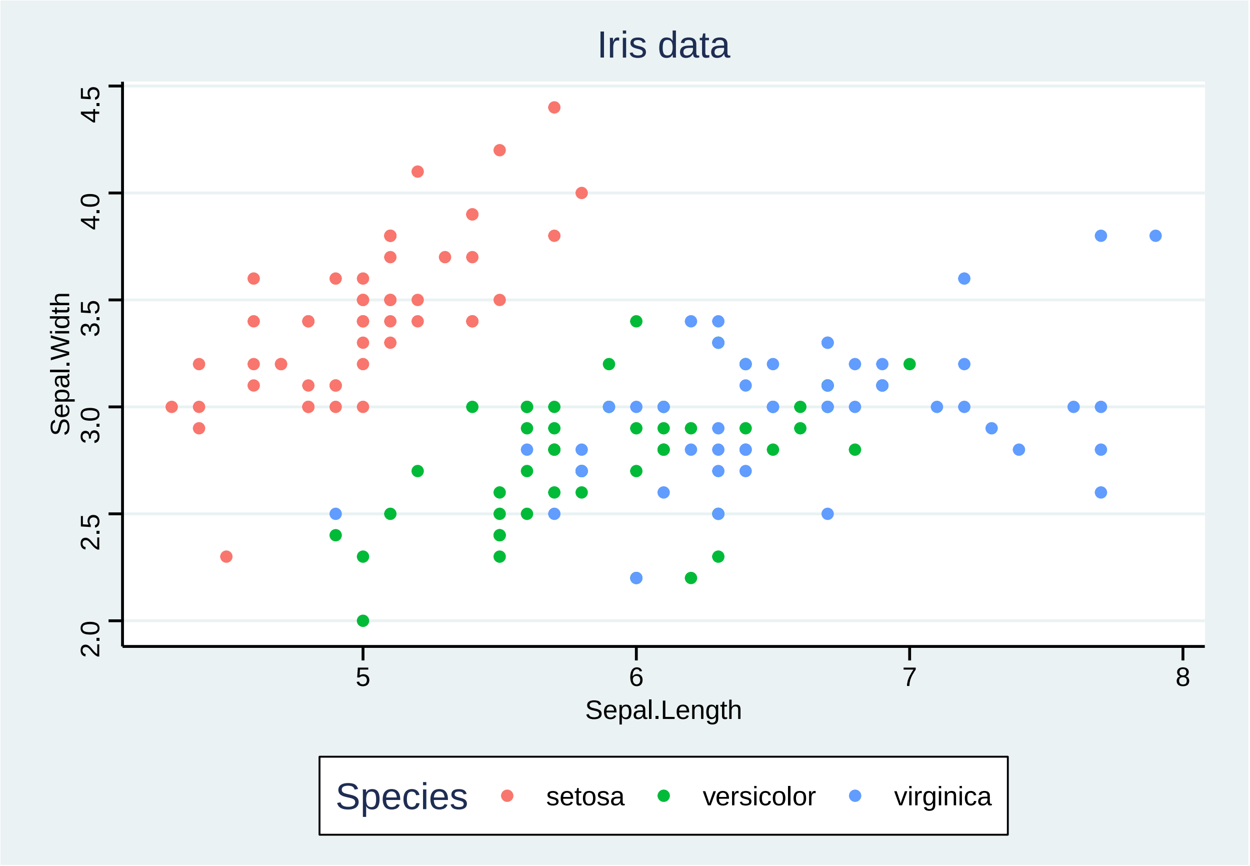
Applying the Economist theme to the plot, which features a clean and professional look with a dark background, white grid lines, and a minimalistic style similar to the design used in The Economist magazine.

```
p + theme_economist() +
 ggtitle("Iris data sets")
```

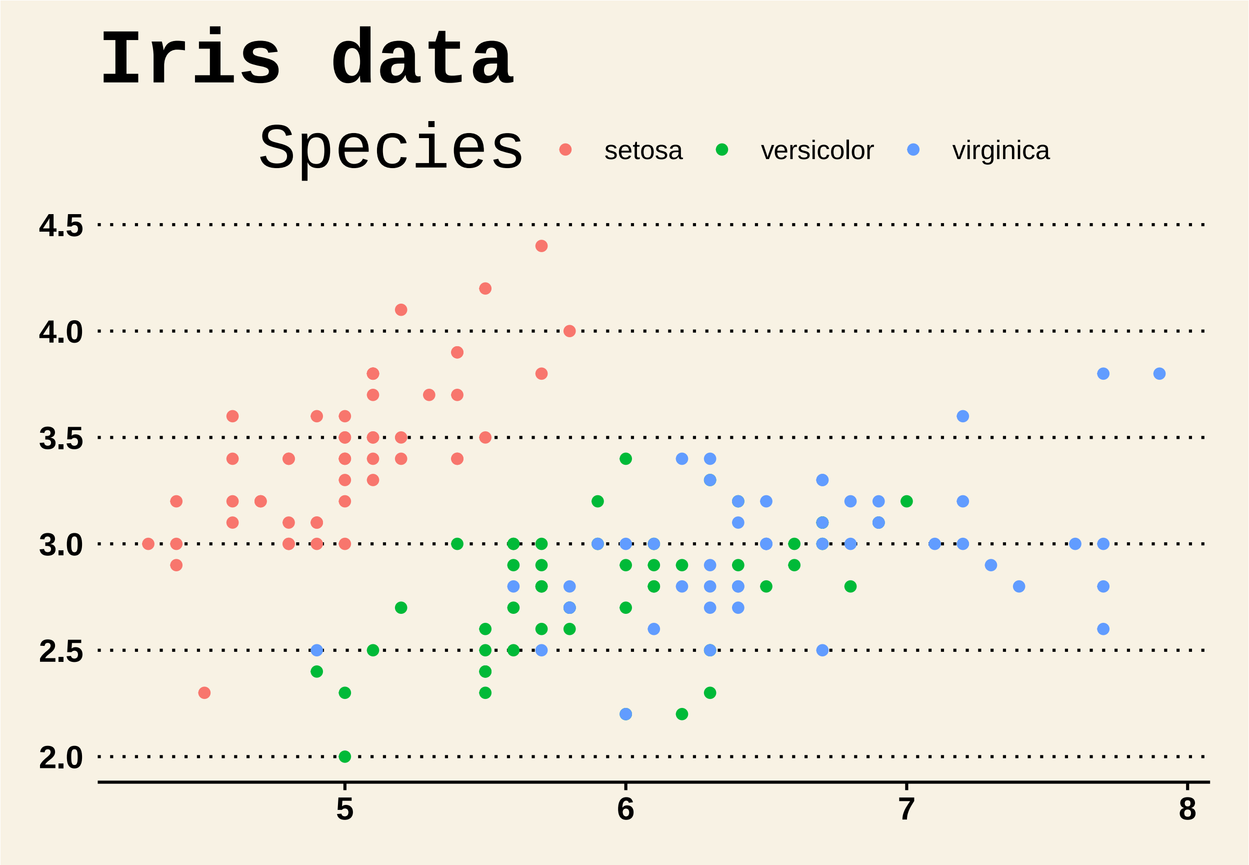


The theme_stata() function provides a polished, professional look to the plot, reminiscent of graphics from Stata software, with a clean background and minimal grid lines.

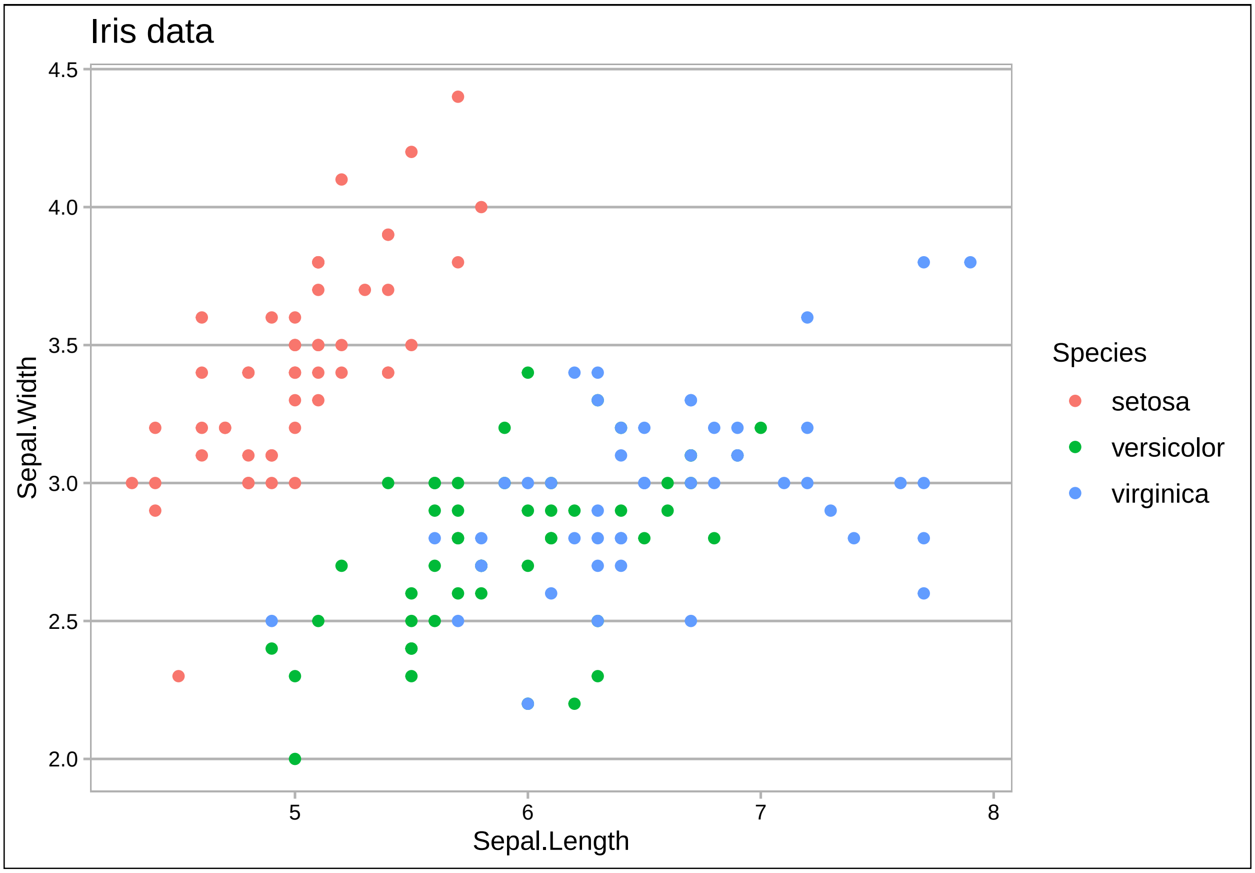
```
p + theme_stata() +
 ggtitle("Iris data")
```



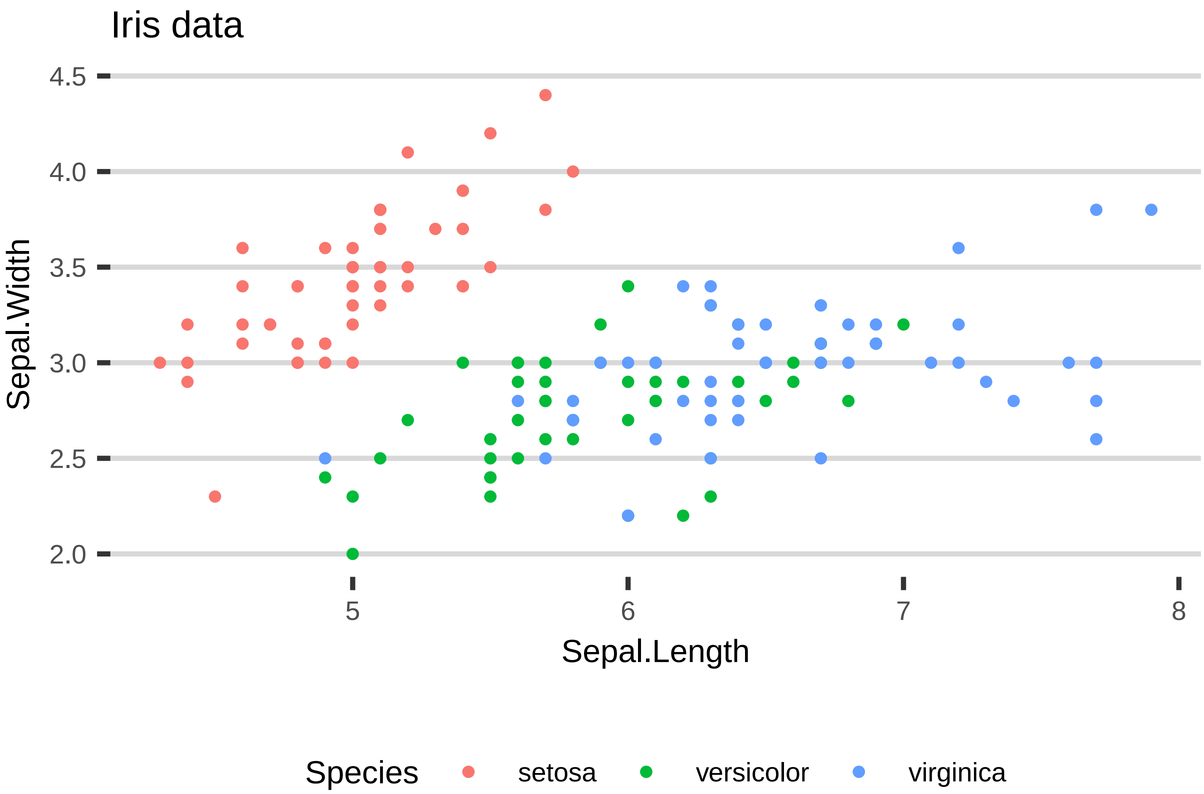
```
p + theme_wsj()+
 ggtitle("Iris data")
```



```
p + theme_calc()+
 ggtitle("Iris data")
```

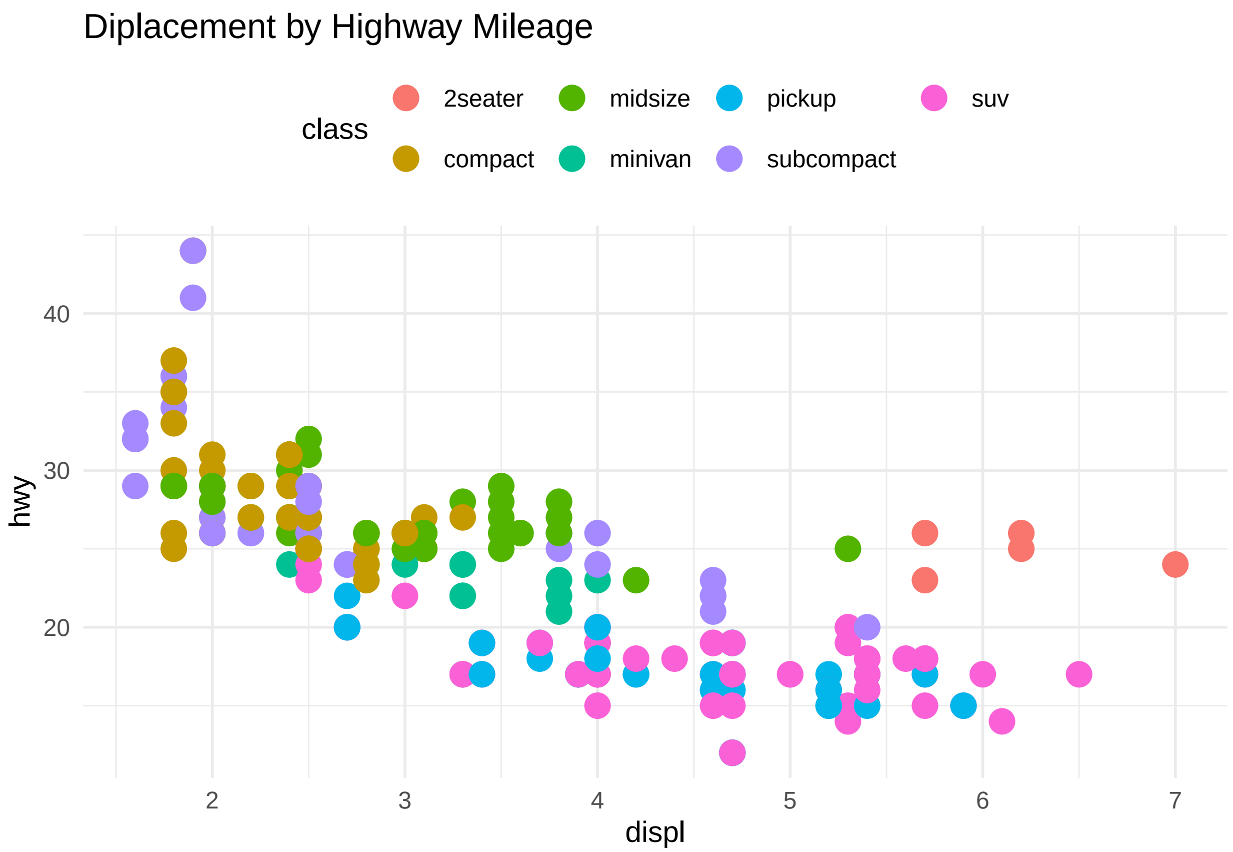


```
p + theme_hc()+
 ggtitle("Iris data")
```



The code creates a scatter plot showing engine displacement versus highway mileage from the mpg dataset, with points colored by vehicle class. It uses a minimalistic theme for a clean look, adds a title, and positions the legend at the top of the plot

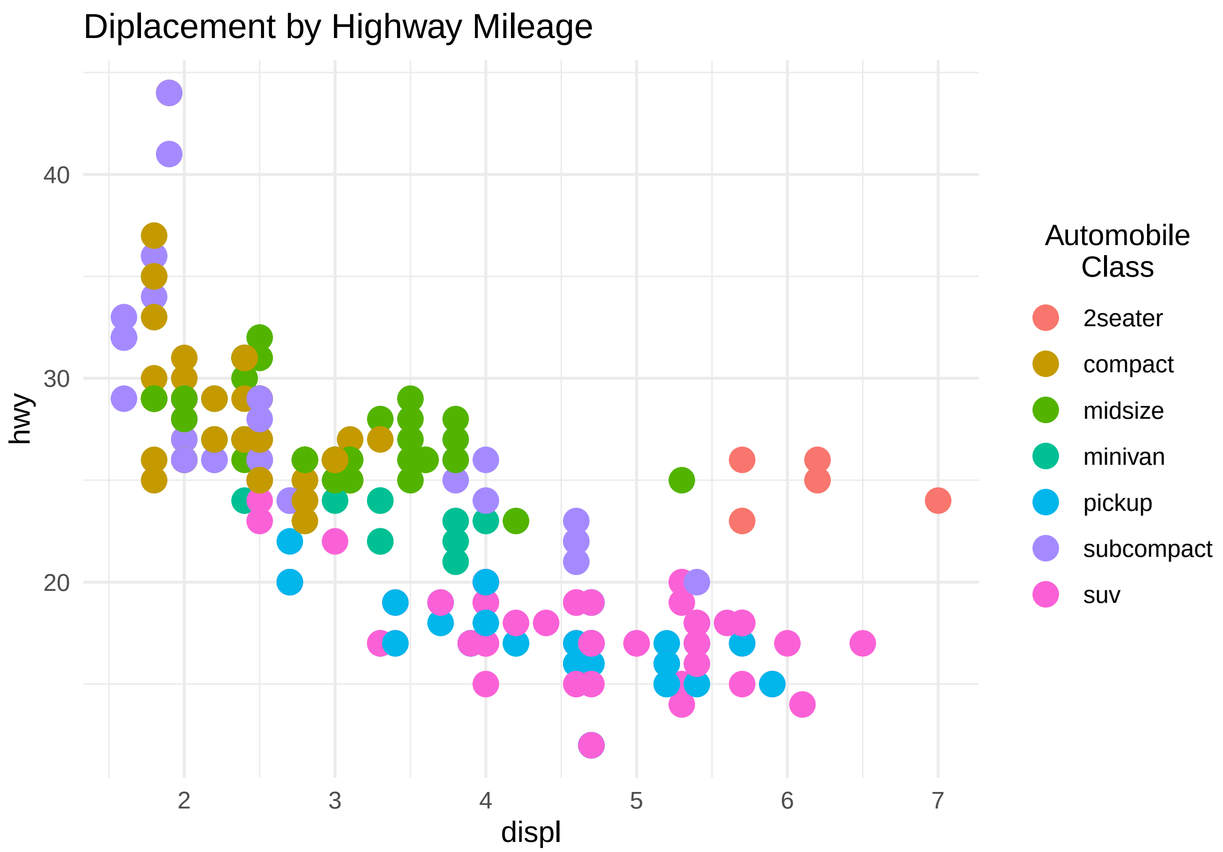
```
ggplot(mpg,
 aes(x = displ, y=hwy, color = class)) +
 geom_point(size = 4) +
 labs(title = "Diplacement by Highway Mileage") +
 theme_minimal() +
 theme(legend.position = "top")
```



The code generates a scatter plot displaying engine displacement versus highway mileage, with points colored by automobile class. It applies a minimalistic theme, adds a title and customizes the legend title to be centered and split into two lines (“Automobile” and “Class”).

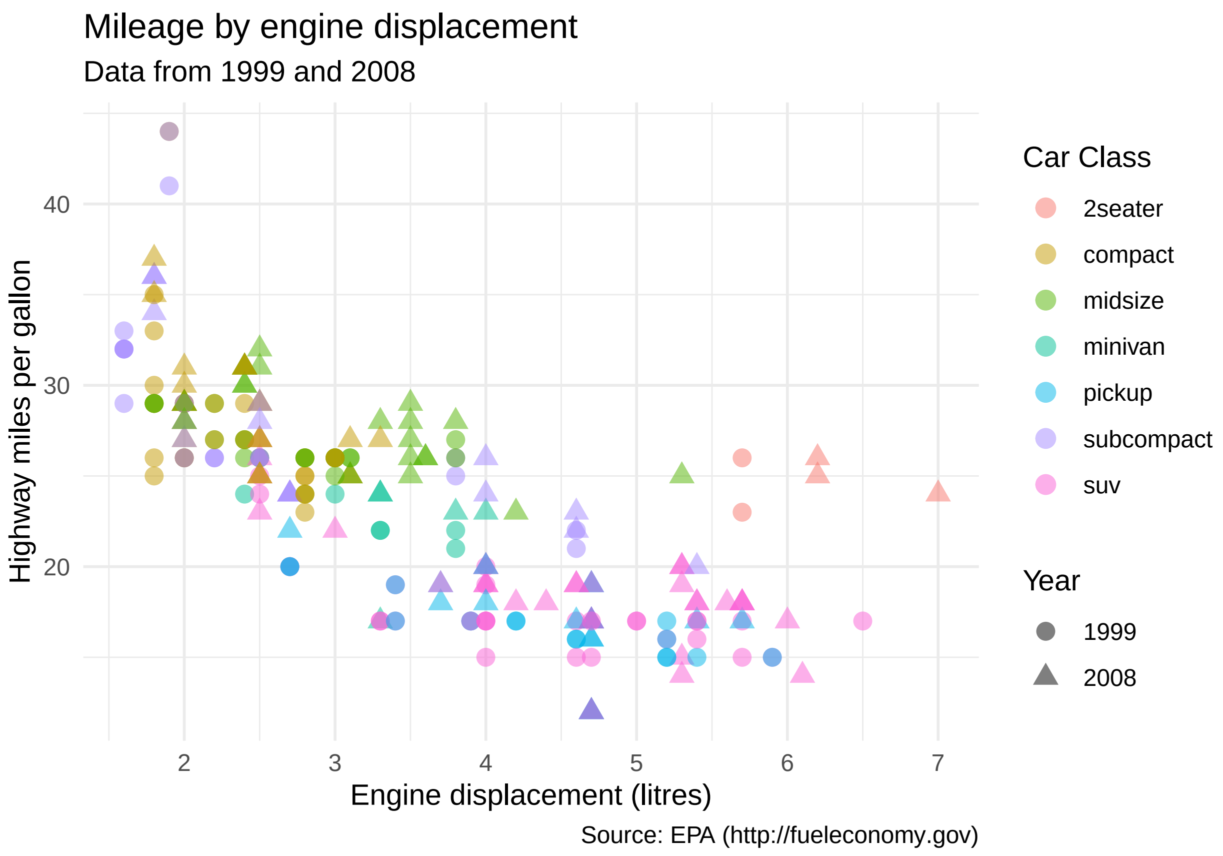
```
ggplot(mpg,
 aes(x = displ, y=hwy, color = class)) +
 geom_point(size = 4) +
 labs(title = "Diplacement by Highway Mileage",
 color = "Automobile\nClass") +
 theme_minimal() +
 theme(legend.title.align=0.5)
```

```
## Warning: The ‘legend.title.align‘ argument of ‘theme()‘ is deprecated as of ggplot2
## 3.5.0.
## i Please use theme(legend.title = element_text(hjust)) instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



The code creates a scatter plot showing the relationship between engine displacement and highway mileage, with points colored by car class and shaped by the year (1999 or 2008). It adds a title, subtitle, caption, and axis labels, and uses a minimalistic theme for a clean presentation. The points are semi-transparent to help visualize overlapping data

```
ggplot(mpg,
 aes(x = displ, y=hwy,
 color = class,
 shape = factor(year))) +
 geom_point(size = 3,
 alpha = .5) +
 labs(title = "Mileage by engine displacement",
 subtitle = "Data from 1999 and 2008",
 caption = "Source: EPA (http://fueleconomy.gov)",
 x = "Engine displacement (litres)",
 y = "Highway miles per gallon",
 color = "Car Class",
 shape = "Year") +
 theme_minimal()
```


Mileage by engine displacement
Data from 1999 and 2008
Highway miles per gallon
40
30
20
2
3
4
5
6
7
Engine displacement (litres)
Car Class
2seater
compact
midsize
minivan
pickup
subcompact
suv
Year
1999
2008
Source: EPA (http://fueleconomy.gov)