Introduction to Predicitve Analytics with R part 1

Michael Brett

2023-01-26

This course covers

- Exploratory Data Analysis and visualization
- Building the following Supervised learning Regression models:
  - Linear Regression
  - Multiple Linear Regression
  - Polynomial Regression
  - Logistic Regression
  - Multiple Logistic Regression
  - Decision Tree
  - Random Forest
- Interpretation of model output
- Making predictions using the models
- Creating professional reports using R Markdown

Data Visualisation

One of the most important aspects of data analysis is the ability to effectively communicate your findings. Creating tables and plots to visualize your data can help make your findings more understandable and engaging for your audience.

We will be using two built-in datasets in R, “mtcars” and “iris”, to demonstrate how to create tables and plots to visualize and present data.

First, let’s load the datasets:

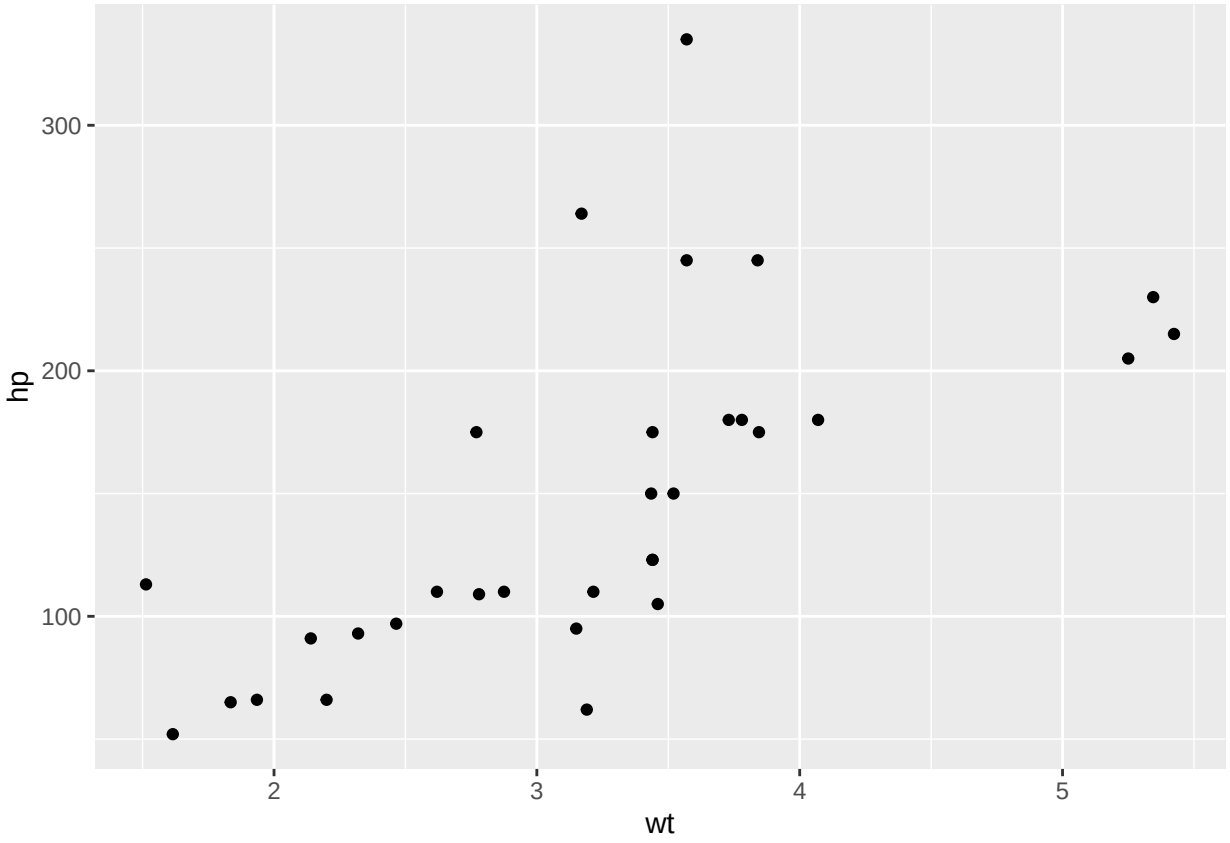
```
data("mtcars")
data("iris")
data("Titanic")
```

In predictive analytics key plot used include: Scatter plots: to visualize the relationship between the predictor and the target variables. Line plots: to visualize the relationship between the target variable and time. You need to install the ggplot2 package first by using the command install.packages(“ggplot2”), and then load it by using the command library(ggplot2)

```
#install.packages("ggplot2")
library(ggplot2)
```

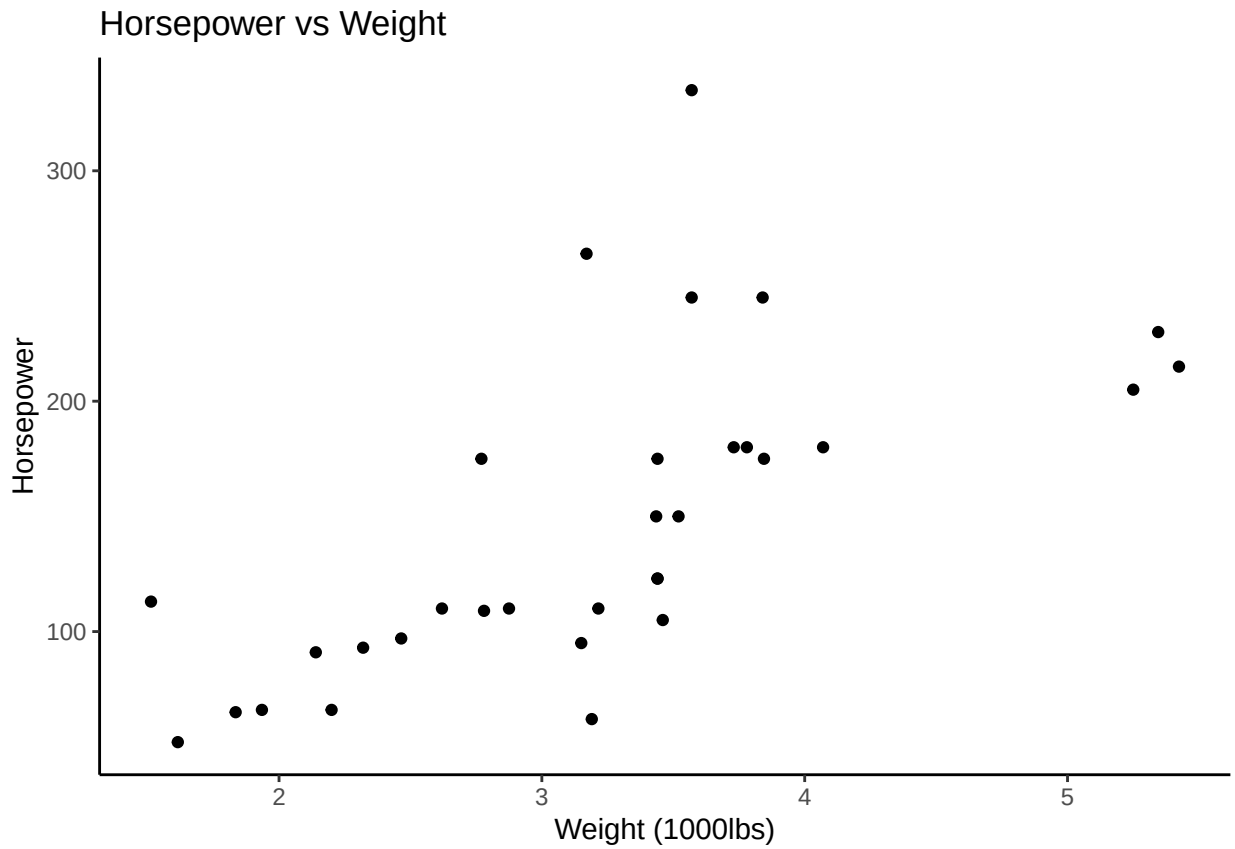
Next, we can use the built-in mtcars data set to create a scatter plot of horsepower versus weight:

```
ggplot(data = mtcars) +
  geom_point(mapping = aes(x = wt, y = hp))
```



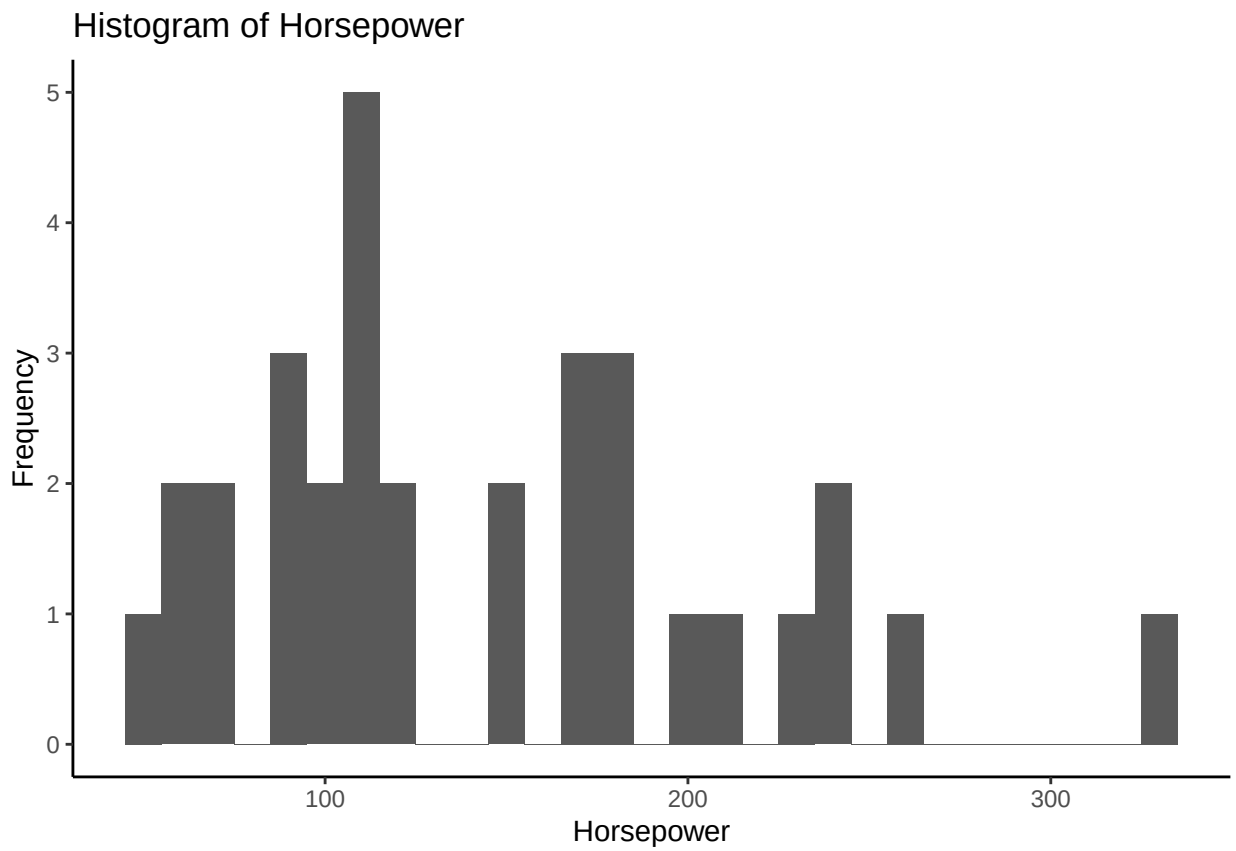
We can add plot labels and customize other options by adding additional layers to the plot:

```
ggplot(data = mtcars) +
  geom_point(mapping = aes(x = wt, y = hp)) +
  ggtitle("Horsepower vs Weight") +
  xlab("Weight (1000lbs)") +
  ylab("Horsepower") +
  theme_classic()
```



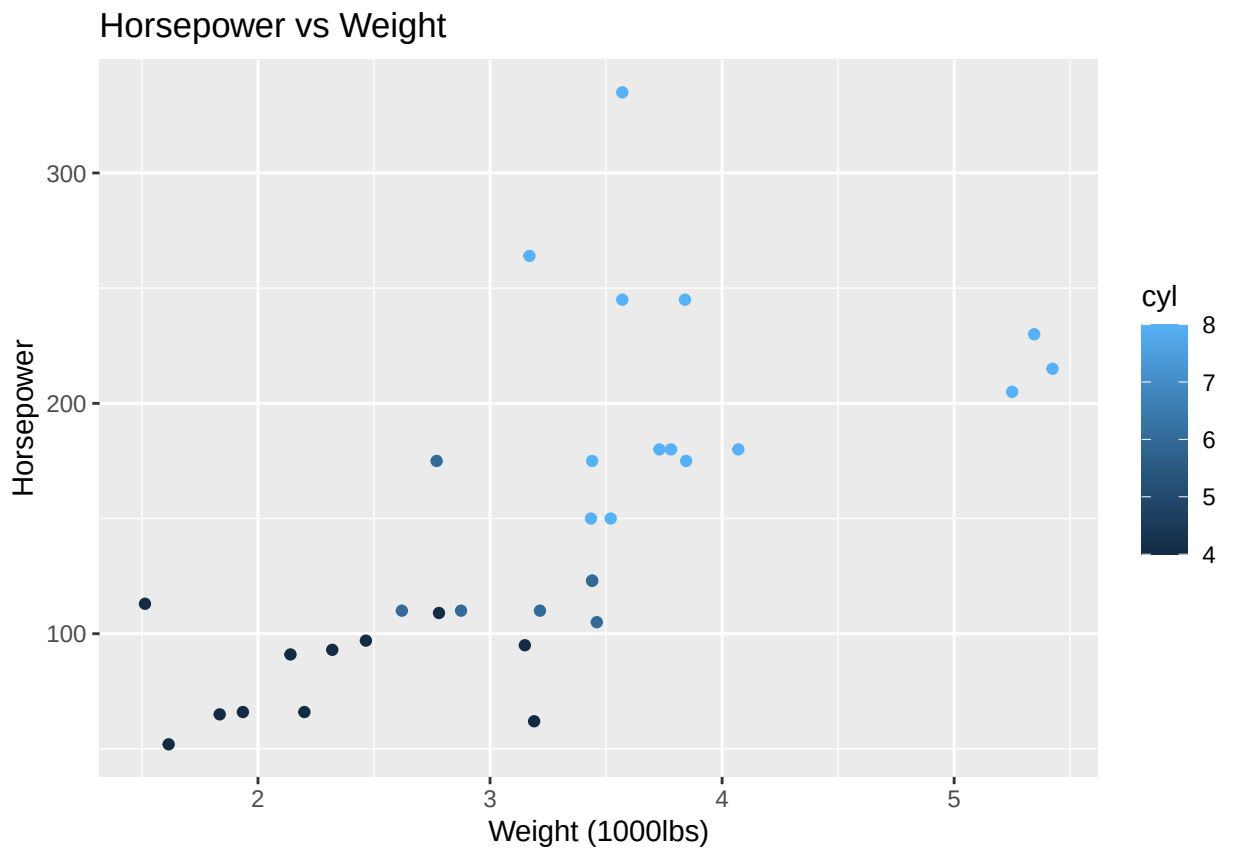
we can create a histogram of the horsepower data:

```
ggplot(data = mtcars) +
  geom_histogram(mapping = aes(x = hp), binwidth = 10) +
  ggtitle("Histogram of Horsepower") +
  xlab("Horsepower") +
  ylab("Frequency") +
  theme_classic()
```



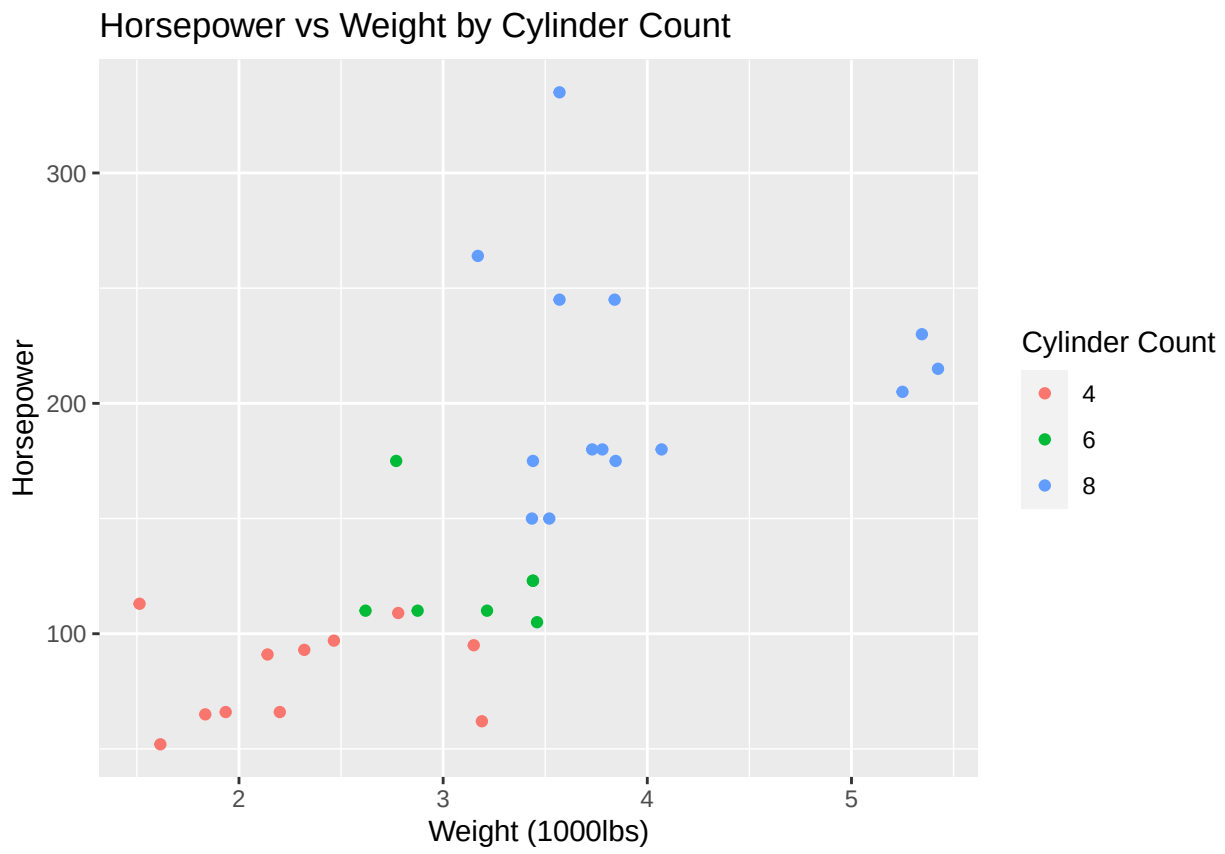
You can colour the scatter plot by other variables, by adding color = cly, the points plotted are differentiated by colour.

```
ggplot(data = mtcars) +
  geom_point(mapping = aes(x = wt, y = hp, colour = cyl)) +

  ggtitle("Horsepower vs Weight") +

  xlab("Weight (1000lbs)") +

  ylab("Horsepower")
```



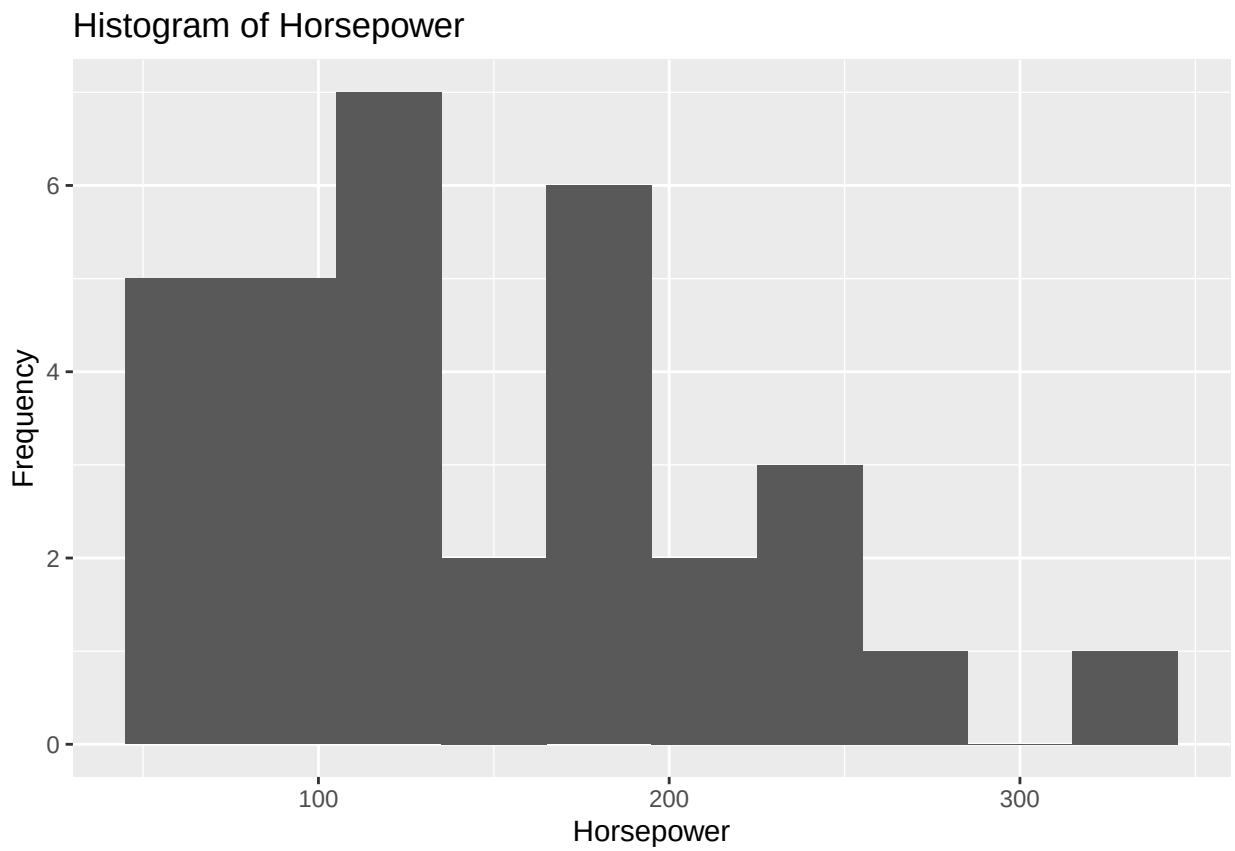
To add a legend, use the aes() function to map variables to the color or shape of the points, and then use the scale_color_discrete() or scale_shape_discrete() functions to specify the legend labels:

```
ggplot(data = mtcars) +
  geom_point(mapping = aes(x = wt, y = hp, color = factor(cyl))) +

  ggtitle("Horsepower vs Weight by Cylinder Count") +

  xlab("Weight (1000lbs)") +

  ylab("Horsepower") +

  scale_color_discrete(name = "Cylinder Count")
```



Changing the width of the bins may be required for your particular data set, in this case its set to 30.

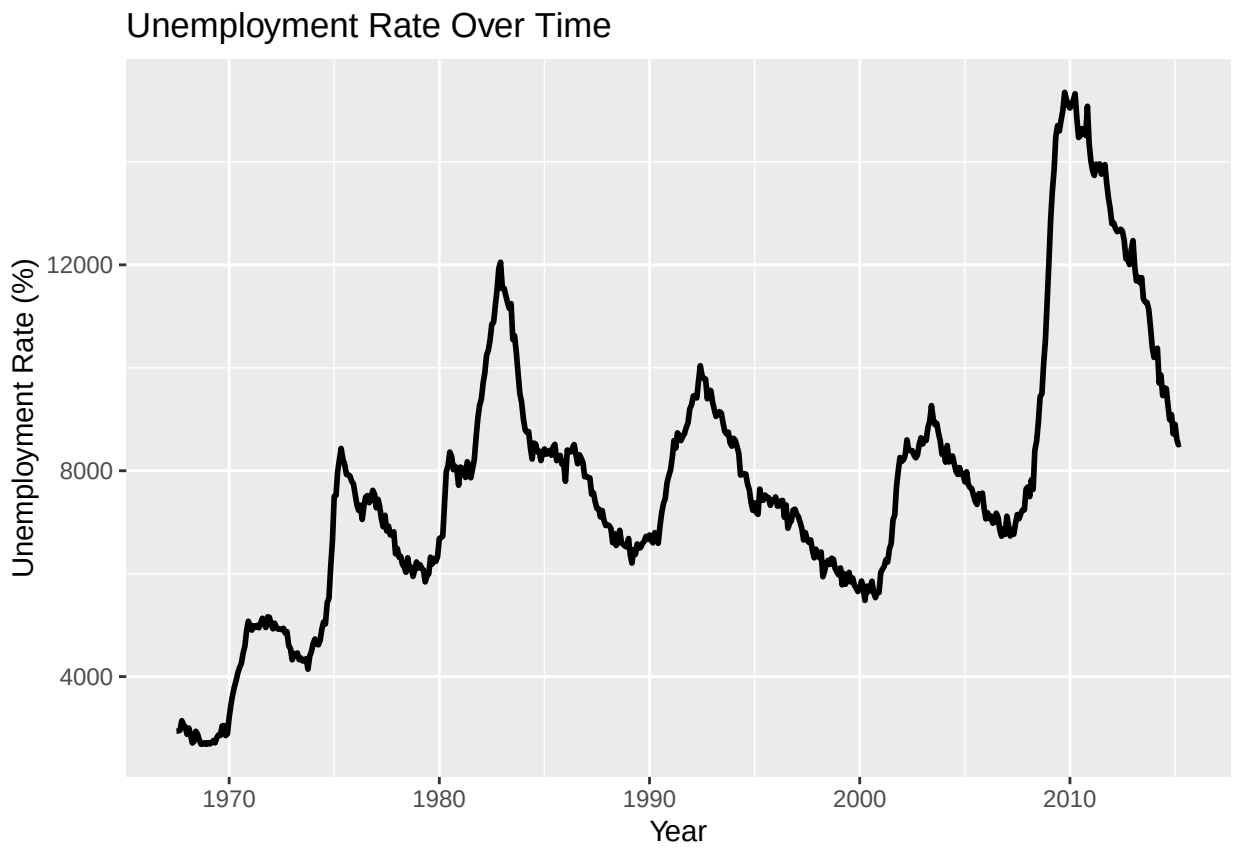
```
library(ggplot2)
ggplot(data = mtcars) +
  geom_histogram(mapping = aes(x = hp), binwidth = 30) +
  ggtitle("Histogram of Horsepower") +
  xlab("Horsepower") +
  ylab("Frequency")
```



It's important to note that the ggtitle, xlab, ylab are the functions to be used to add the titles, labels to the plots respectively. Also, in case you want to change the theme

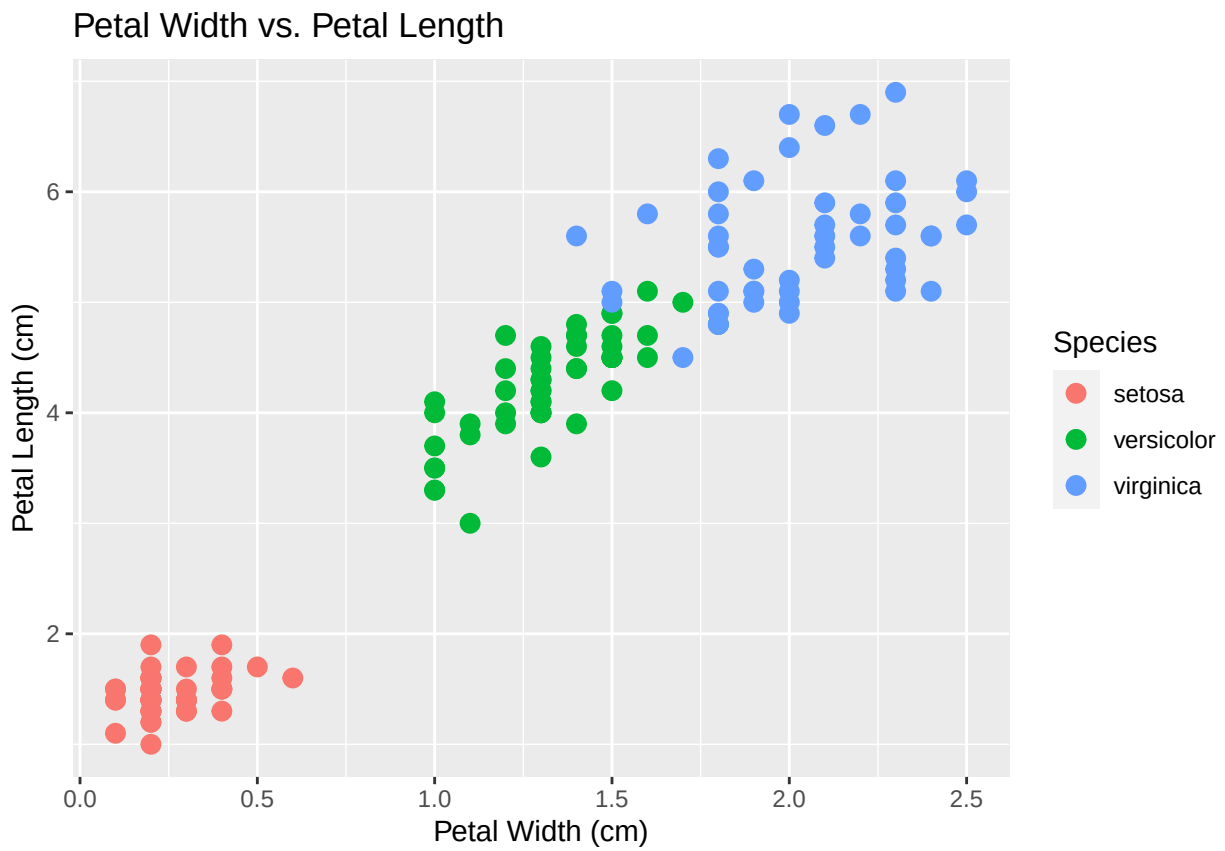
Here are some examples of how you can use the ggplot2 package to create each of the common plots using a built-in dataset in R:

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

# Create a line plot of the unemployment rate over time
ggplot(economics, aes(x = date, y = unemploy)) +
  geom_line(linewidth=1) +
  ggtitle("Unemployment Rate Over Time") +
  xlab("Year") +
  ylab("Unemployment Rate (%)")
```



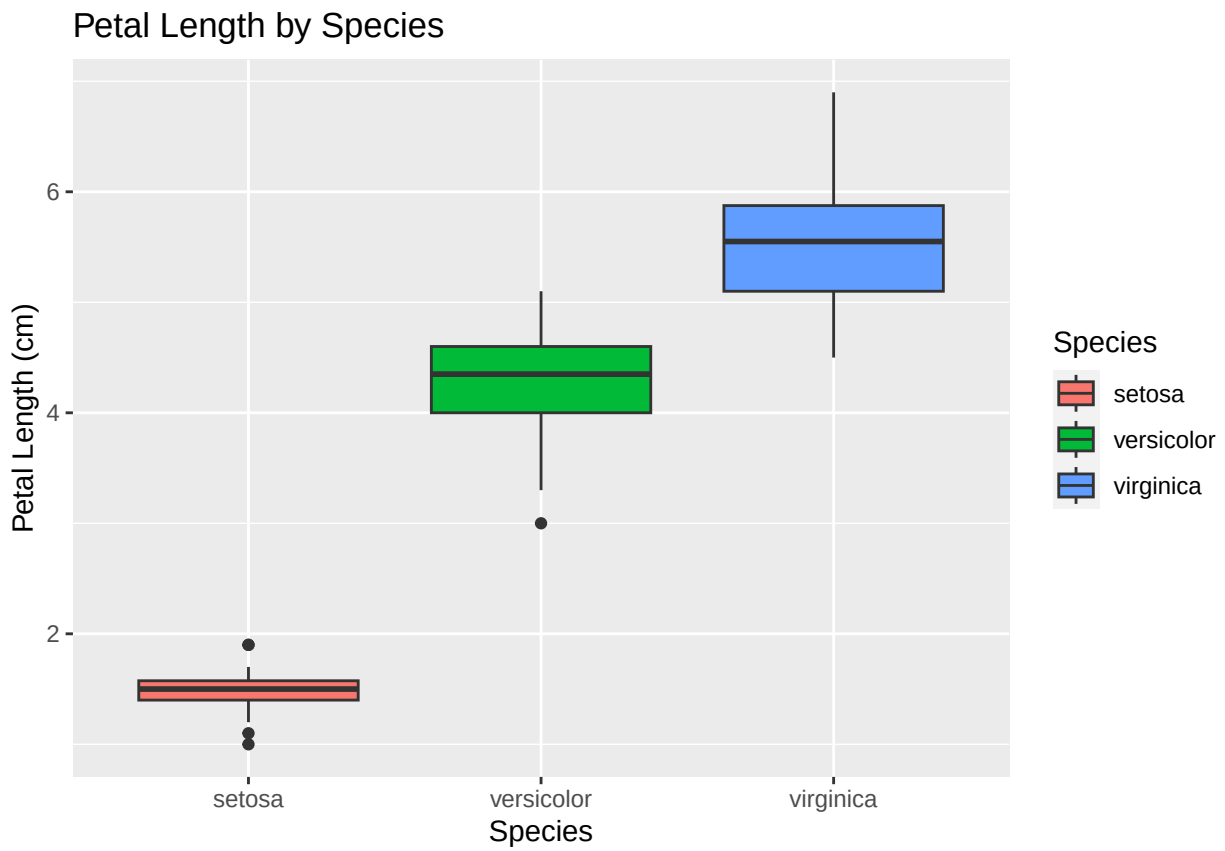
Scatter plots:

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

# Create a scatter plot of petal length vs. petal width
ggplot(iris, aes(x = Petal.Width, y = Petal.Length, colour = Species)) +
  geom_point(size=3) +
  ggtitle("Petal Width vs. Petal Length") +
  xlab("Petal Width (cm)") +
  ylab("Petal Length (cm)")
```



Box plots:

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

# Create a box plot of petal length by species
ggplot(iris, aes(x = Species, y = Petal.Length, fill = Species)) +
  geom_boxplot() +
  ggtitle("Petal Length by Species") +
  xlab("Species") +
  ylab("Petal Length (cm)")
```



You can customize the plots by changing the colors, adding more variables, adding more geoms, etc.

exporting as high res pdf:

```
pdf('boxplots_examlpe.pdf', height = 8.27, width = 11.69, paper = "a4r")

library(ggplot2)

  ggplot(iris, aes(x = Species, y = Petal.Length, fill = Species)) +
  geom_boxplot() +
  ggtitle("Petal Length by Species") +
  xlab("Species") +
  ylab("Petal Length (cm)")

  dev.off()
```

```
## pdf
##   2
```