Continuous Data (4 of 4)

Aug 2, 2023. V1.1 -=- This chapter is being heavily edited; It is very much Work in Progress

Continuous Data Across Categories using ggplot2

1. This chapter takes us a step further in our exploration of continuous data. Here, we delve into the use of ggplot2.
2. **Data**: Let us work with the same mtcars data from the previous chapter. Suppose we have run the following code:

```
# Load the required libraries, suppressing annoying startup messages
library(tibble)
suppressPackageStartupMessages(library(dplyr))
# Read the mtcars dataset into a tibble called tb
data(mtcars)
tb <- as_tibble(mtcars)
attach(tb)
# Convert several numeric columns into factor variables
tb$cyl <- as.factor(tb$cyl)
tb$vs <- as.factor(tb$vs)
tb$am <- as.factor(tb$am)
tb$gear <- as.factor(tb$gear)

attach(tb)
```

```
The following objects are masked from tb (pos = 3):

    am, carb, cyl, disp, drat, gear, hp, mpg, qsec, vs, wt
```

Summarizing Continuous Data by one Factor (using ggplot2)

1. We investigate the bivariate Relationship between Miles Per Gallon (`mpg`) and Cylinders (`cyl`) using ggplot2.

```
library(dplyr)

agg <- tb %>%
  group_by(cyl) %>%
  summarise(Avg_MPG = mean(mpg, na.rm = TRUE),
            SD_MPG = sd(mpg, na.rm = TRUE))
agg
```

```
# A tibble: 3 x 3
  cyl   Avg_MPG SD_MPG
  <fct>   <dbl>  <dbl>
1 4        26.7   4.51
2 6        19.7   1.45
3 8        15.1   2.56
```

2. Discussion:

- In this code, we are using the pipe operator %>% to perform a series of operations. We first group the data by the cyl column using the group_by() function. We then use summarise() to apply both the mean() and sd() functions to the mpg column.
- The results are stored in new columns, Avg_MPG and SD_MPG.
- Note that na.rm = TRUE is used in both mean() and sd() to remove missing values before calculation. This ensures that our calculations won't be disrupted by any missing data. If you are sure your data has no missing values, you may omit this.

Bivariate Continuous Data: Unveiling Relationships with Scatterplots and Categorical Breakdowns

This chapter marks the exploration of relationships between two continuous variables. We start by summarizing bivariate continuous data, showing how to calculate correlation coefficients and other related statistics using R and dplyr.

Following this, we dive into the visualization of bivariate continuous data. We introduce scatterplots as the primary tool for this purpose, illustrating how to create them using ggplot2.

In addition, we discuss how to add trend lines to scatterplots, and the interpretations that can be made from them.

In the later sections, we add another layer of complexity by breaking down bivariate continuous data by categorical variables. We demonstrate how to use color, shape, and facets in ggplot2 to create scatterplots that can visualize the interaction between two continuous variables across different categories.

By the end of this chapter, you will be equipped with robust techniques to analyze and visualize interactions between two continuous variables, while considering the impact of categorical variables.

6. **Data**: Let us work with the same mtcars data from the previous chapter. Suppose we have run the following code:

```
# Load the required libraries, suppressing annoying startup messages
library(tibble)
suppressPackageStartupMessages(library(dplyr))
# Read the mtcars dataset into a tibble called tb
data(mtcars)
tb <- as_tibble(mtcars)
attach(tb)
```

```
The following objects are masked from tb (pos = 3):

    am, carb, cyl, disp, drat, gear, hp, mpg, qsec, vs, wt

The following objects are masked from tb (pos = 4):

    am, carb, cyl, disp, drat, gear, hp, mpg, qsec, vs, wt
```

```
# Convert several numeric columns into factor variables
tb$cyl <- as.factor(tb$cyl)
tb$vs <- as.factor(tb$vs)
tb$am <- as.factor(tb$am)
tb$gear <- as.factor(tb$gear)

attach(tb)
```

```
The following objects are masked from tb (pos = 3):

    am, carb, cyl, disp, drat, gear, hp, mpg, qsec, vs, wt

The following objects are masked from tb (pos = 4):

    am, carb, cyl, disp, drat, gear, hp, mpg, qsec, vs, wt

The following objects are masked from tb (pos = 5):

    am, carb, cyl, disp, drat, gear, hp, mpg, qsec, vs, wt
```

Bivariate relationships between Continuous data

Scatterplot

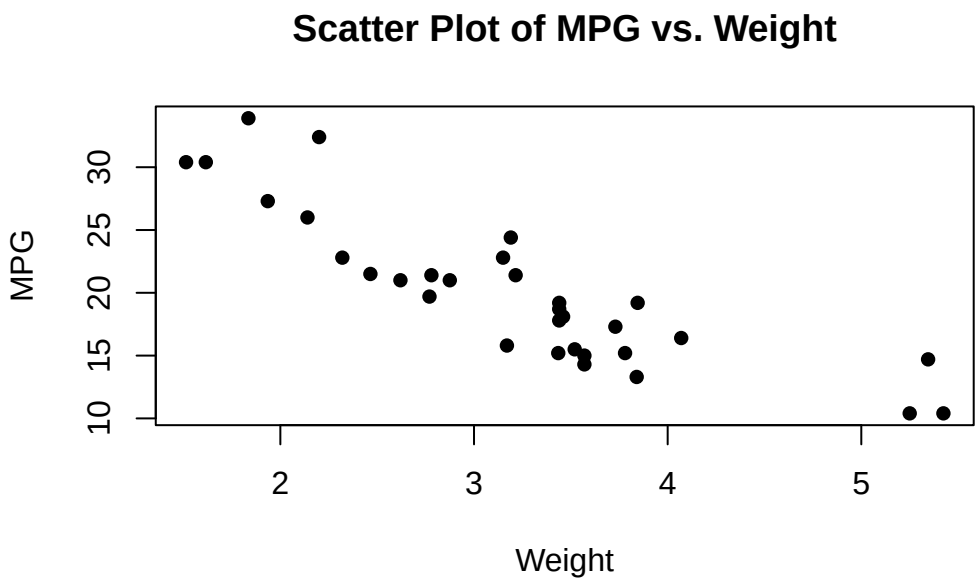
A scatter plot is a type of graph used to display the relationship between two continuous variables. It is a graphical representation of a bivariate distribution, where the values of two variables are plotted as points on a two-dimensional coordinate system.

A scatter plot can be used to identify trends, clusters, outliers, and other patterns in the data. It is also useful for detecting the presence of any outliers or influential observations that may affect the analysis.

The mtcars data set in R is a built-in data set that contains data on various car models. To create a scatter plot of mpg (miles per gallon) against wt (weight) in the mtcars data set, you can use the following code:

Scatterplot using plot()

```r
plot(tb$wt, tb$mpg, main = "Scatter Plot of MPG vs. Weight",
     xlab = "Weight", ylab = "MPG", pch = 16)
```



This code will first load the mtcars data set, then create a scatter plot of `mpg` against `wt` using the `plot()` function. The main argument adds a title to the plot, the `xlab` and `ylab` arguments add axis labels, and the pch argument changes the shape of the points to a solid circle. The resulting scatter plot will show the relationship between `mpg` and `wt` in the `mtcars` data set.

Scatterplot using ggplot2

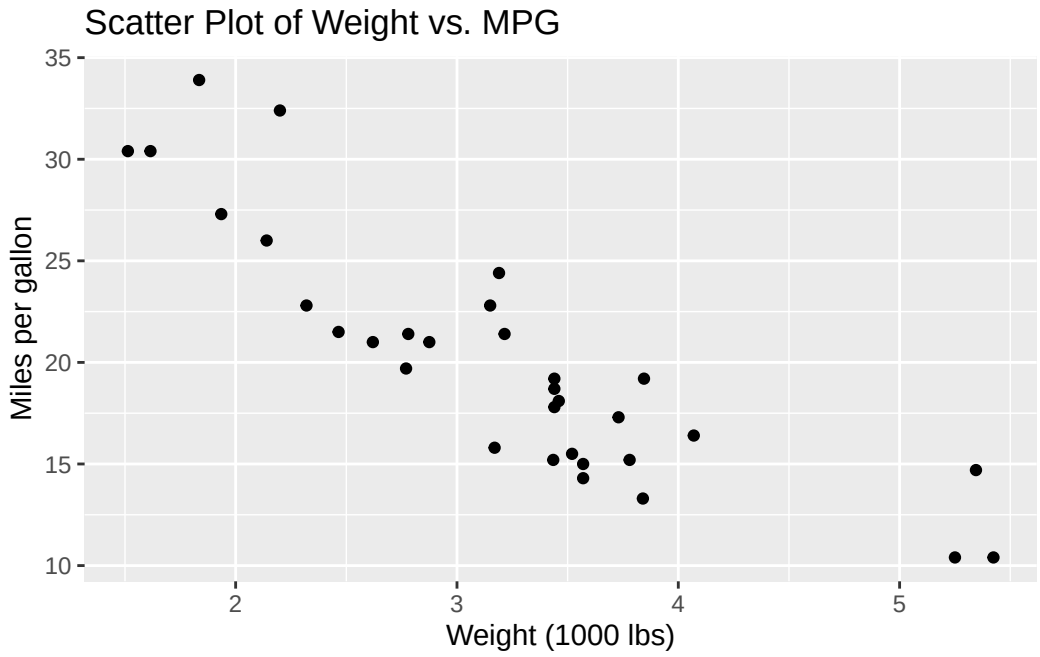
```
# Load the ggplot2 package
library(ggplot2)
```

```
Attaching package: 'ggplot2'
```

```
The following object is masked from 'tb':

    mpg
```

```
# Create the scatter plot
ggplot(tb, aes(x = wt, y = mpg)) +
  geom_point() +
  xlab("Weight (1000 lbs)") +
  ylab("Miles per gallon") +
```

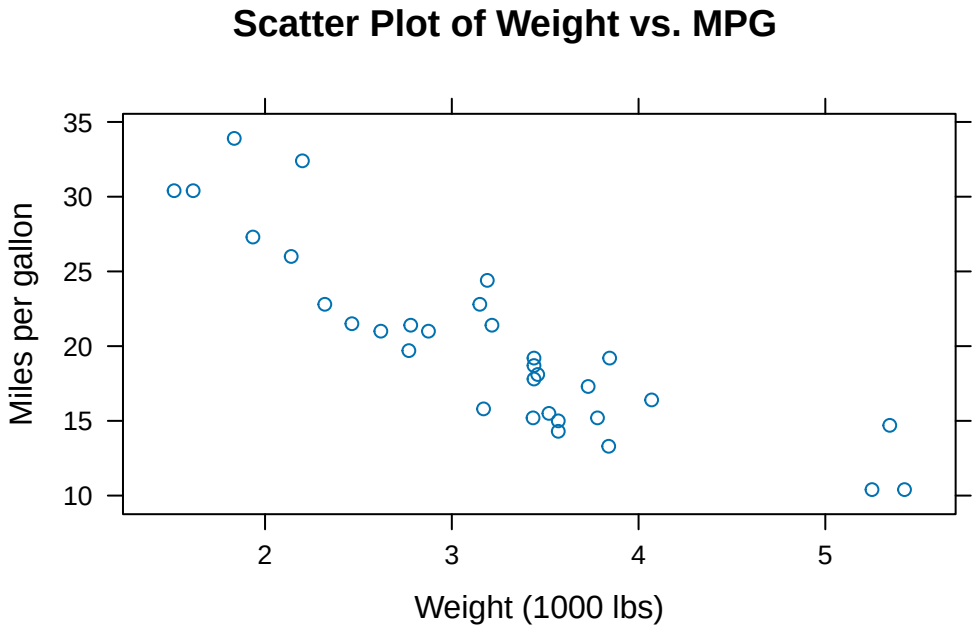
```
  ggtitle("Scatter Plot of Weight vs. MPG")
```



This code creates a scatter plot of the wt variable (weight in 1000 lbs) on the x-axis and the mpg variable (miles per gallon) on the y-axis. The geom_point() function is used to add the points to the plot, and xlab(), ylab(), and ggtitle() are used to add axis labels and a plot title, respectively. You can adjust the aesthetics of the plot, such as the color and size of the points, by adding additional arguments to the geom_point() function.

Scatterplot using Lattice

```
# Load the Lattice package
library(lattice)

# Create the scatter plot
xyplot(mpg ~ wt, data = tb,
       xlab = "Weight (1000 lbs)",
       ylab = "Miles per gallon",
       main = "Scatter Plot of Weight vs. MPG")
```



This code creates a scatter plot of the wt variable (weight in 1000 lbs) on the x-axis and the mpg variable (miles per gallon) on the y-axis using the xyplot() function. The data argument specifies the data frame to use, and xlab, ylab, and main are used to add axis labels and a plot title, respectively. You can also add additional arguments to adjust the aesthetics of the plot, such as the size and color of the points or the type of line connecting the points, depending on your data and preferences.

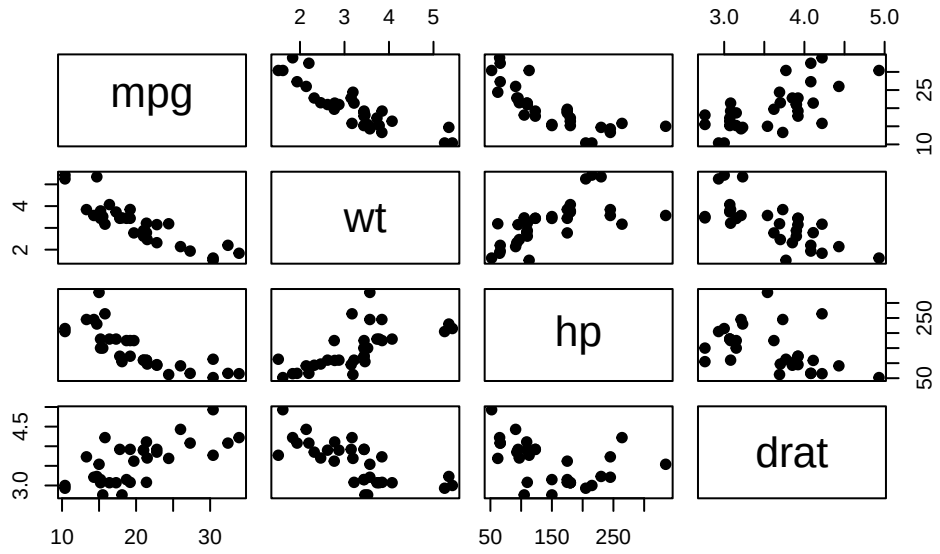
Scatterplot Matrix

A scatter plot matrix (also called a pairs plot or a SPLOM) is a graphical display of pairwise scatter plots of a set of variables. In a scatter plot matrix, each variable in the dataset is plotted against every other variable in a matrix format. This allows us to visualize the relationships between pairs of variables and explore potential patterns or trends in the data.

A scatter plot matrix is particularly useful for exploring multivariate datasets, as it allows us to quickly identify which pairs of variables may be strongly correlated, which may have weak or no correlation, and which may exhibit nonlinear relationships. It can also be used to identify outliers or unusual observations, and to visualize clusters or groups of observations based on patterns in the scatter plots.

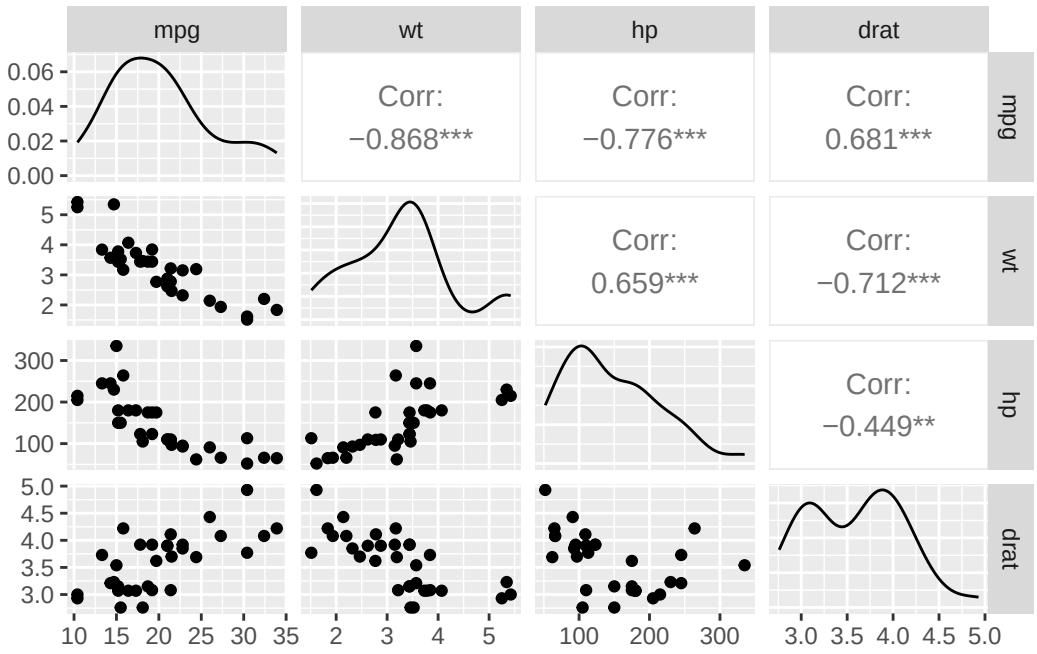
Scatterplot Matrix Using pairs()

```
# scatter plot matrix for mpg, wt, hp, drat
pairs(tb[,c("mpg","wt","hp","drat")], pch = 19)
```



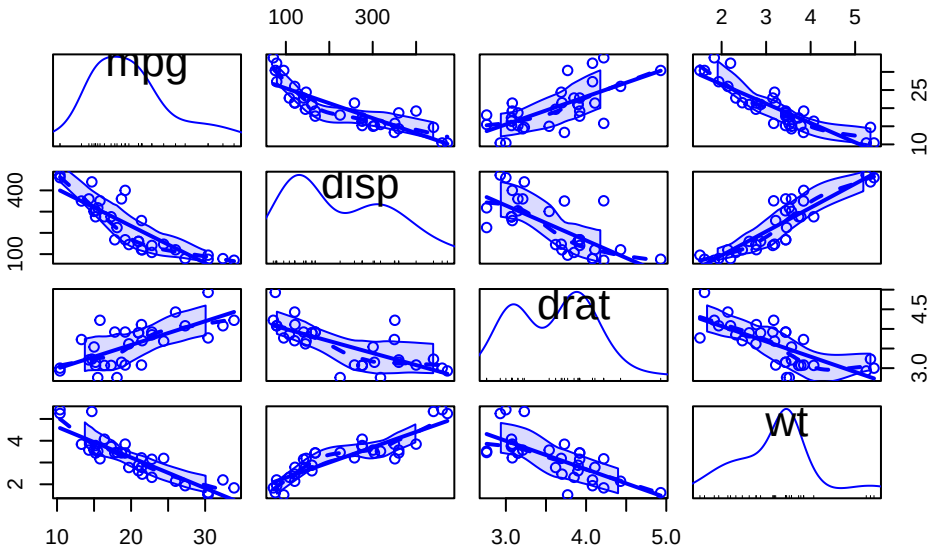
Scatterplot Matrix Using ggpairs()

```
# Load the GGally package
library(GGally)

# Create a scatterplot matrix using ggpairs()
ggpairs(tb[,c("mpg","wt","hp","drat")])
```



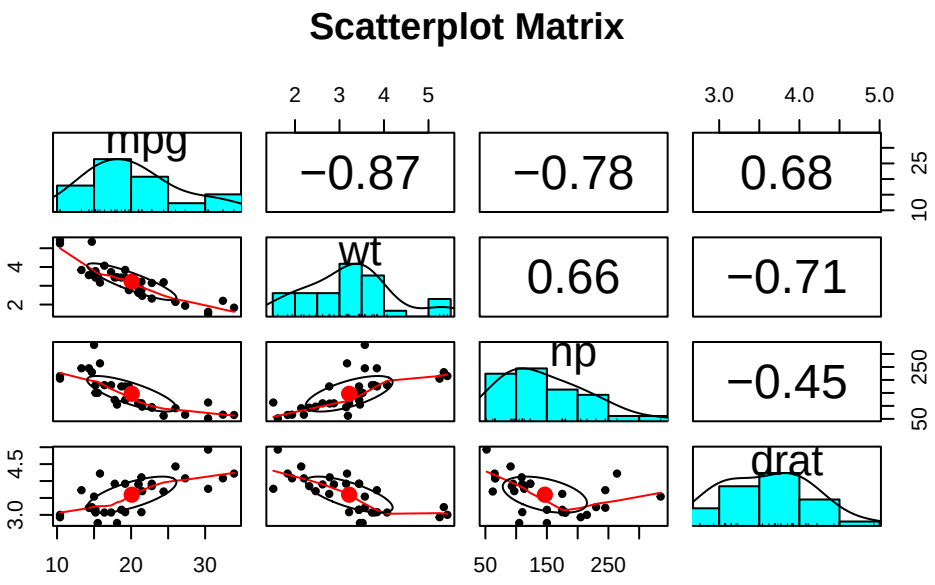
Scatterplot Matrix Using scatterplotMatrix()

```
# Load the car package
library(car)

# Create a scatterplot matrix using scatterplotMatrix()
scatterplotMatrix(~ mpg + disp +drat +wt,
                  data = tb, col = c("blue", "red"))
```



Scatterplot Matrix Using pairs.panels()

```
# Load the psych package
library(psych)

# Create a scatterplot matrix using pairs.panels()
pairs.panels(tb[,c("mpg","wt","hp","drat")],
             main = "Scatterplot Matrix")
```

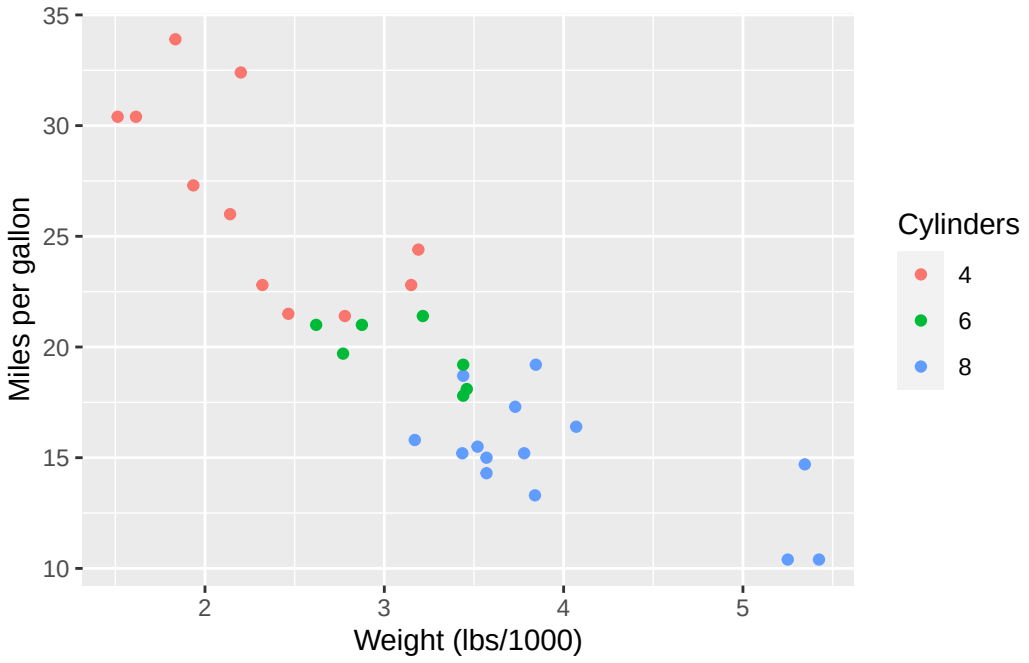


Scatterplots broken down by Categorical Variables

Scatterplot with colored by Categorical Variable Using ggplot()

This will create a scatterplot of miles per gallon (mpg) against weight, with each point colored according to the number of cylinders in the engine (cyl).

```
# Load the ggplot2 package
library(ggplot2)

# Create a scatterplot of mpg vs. wt, colored by cyl
ggplot(tb, aes(x = wt, y = mpg, color = factor(cyl))) +
  geom_point() +
  labs(x = "Weight (lbs/1000)", y = "Miles per gallon") +
  scale_color_discrete(name = "Cylinders")
```



Scatterplot with broken down by Categorical Variable Using ggplot()

This will create a scatterplot of miles per gallon (mpg) against weight, with each plot faceted by the number of cylinders in the engine (cyl).

```r
# Load the ggplot2 package
library(ggplot2)

# Create a scatterplot matrix using ggplot()
ggplot(tb, aes(x = mpg, y = disp)) +
  geom_point() +
  facet_grid(. ~ cyl)
```