Continuous x Continous data (1 of 2)

Aug 7, 2023.

Exploring bivariate Continuous x Continuous data

THIS CHAPTER explores how to summarize and visualize the interaction between *bivariate continuous data* using correlation analysis, scatter plots, scatter plot matrices and other such techniques.

Data: **Data**: Let us work with the same `mtcars` data from the previous chapter. Suppose we run the following code to prepare the data for subsequent analysis and save it in a tibble called `tb`:

```r
# 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)
# Convert several numeric columns into factor variables
tb$cyl <- as.factor(tb$cyl) # cyl = {4,6,8}, number of cylinders
tb$am <- as.factor(tb$am) # am = {0,1}, 0:automatic, 1: manual transmission
tb$vs <- as.factor(tb$vs) # vs = {0,1}, v-shaped engine, 0:no, 1:yes
tb$gear <- as.factor(tb$gear) # gear = {3,4,5}, number of gears
# Directly access the data columns of tb, without tb$mpg
attach(tb)
```

Scatterplots

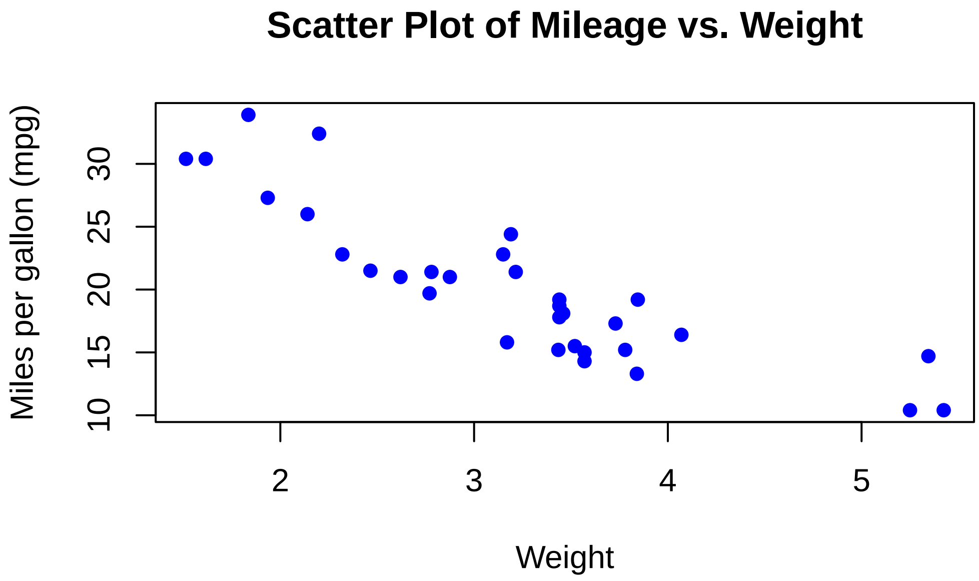
1. A scatter plot is a type of graph used to display the relationship between two continuous variables. It is a graphical representation of a bivariate distribution, where the values of two variables are plotted as points on a two-dimensional coordinate system.

2. 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. [1]

3. To create a scatter plot of `mpg` (miles per gallon) against `wt` (weight) in the mtcars data set, we can use the following code:

Scatterplot using `plot()`

```
plot(tb$wt,
     tb$mpg,
     main = "Scatter Plot of Mileage vs. Weight",
     xlab = "Weight", ylab = "Miles per gallon (mpg)",
     pch = 16,
     col="blue")
```



4. Discussion:

- This code will first load the mtcars data set, then create a scatter plot of `mpg` against `wt` using the `plot()` function.

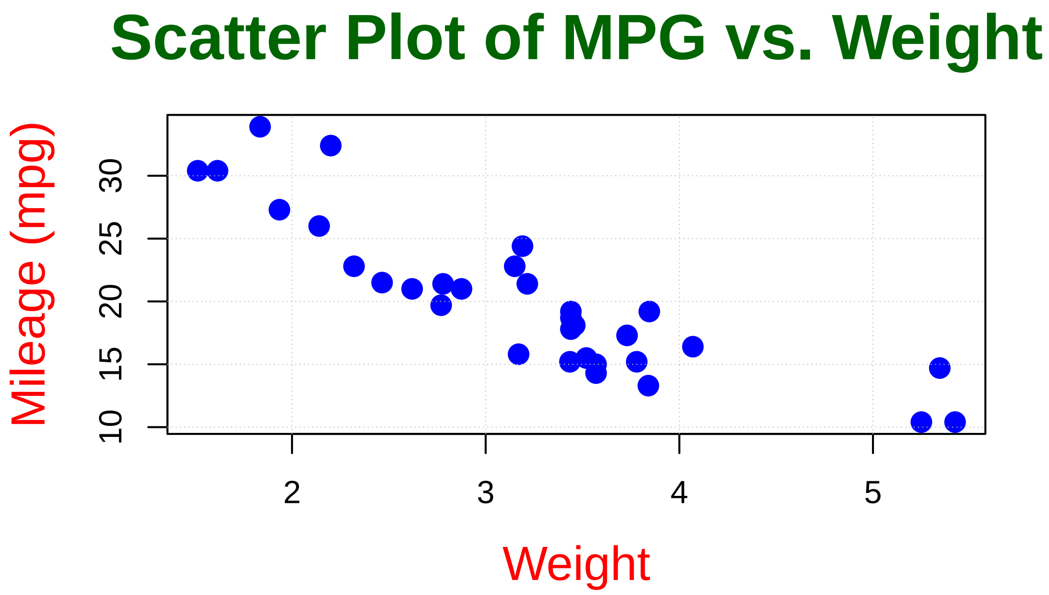
- The `main` argument adds a title to the plot, the `xlab` and `ylab` arguments add axis labels

- the `pch` argument sets the shape of the points to a solid circle. pch = 15 gives a filled square. Here are popular values: pch = 16 gives a filled circle, pch = 17 gives a filled triangle (pointing upwards), pch = 18 gives a filled diamond, pch = 19 gives a solid circle, pch = 20 gives a filled bullet (smaller than pch = 19)

- the `col` argument specifies the color of the data points. We can use any named color in R, or we can use hexadecimal color codes. For instance, `col = "#FF0000"` would give us red points. [2]

5. We can personalize the appearance of the scatterplot in a variety of additional ways.

```
# Create the scatterplot
plot(tb$wt,
     tb$mpg,
     main = "Scatter Plot of MPG vs. Weight",
     xlab = "Weight", ylab = "Mileage (mpg)",
     pch = 16, cex = 1.5, col="blue",
     col.lab="red", cex.lab=1.5,
     col.main="darkgreen", cex.main=2,
     bg = "gray")

# Add a grid
grid(col = "gray", lty = "dotted", lwd = 0.5)
```



6. Discussion

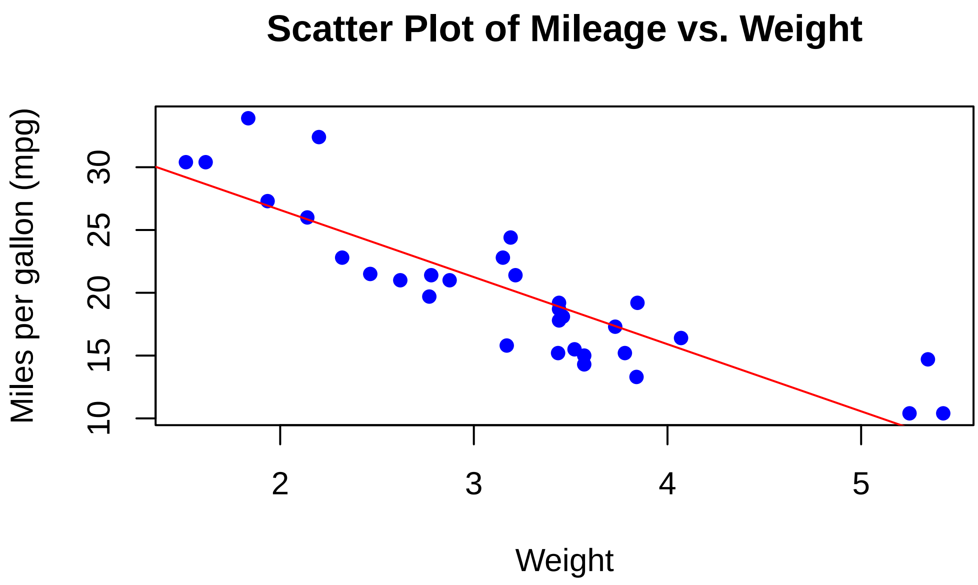
- Point Size: In the second plot, the size of the points is 1.5 times the default size (`cex = 1.5`), while in the first plot, the size of the points is the default size as `cex` is not specified.

- Axis Labels' Color and Size: The second plot has red-colored, larger size axis labels (`col.lab="red", cex.lab=1.5`), while the first plot uses the default color and size as these parameters are not specified.
- Title's Color and Size: The second plot has a dark green title that is twice the default size (`col.main="darkgreen", cex.main=2`), while the first plot uses the default color and size for the title as these parameters are not specified.
- Background Color: The second plot has a light gray background (`bg = "lightgray"`), while the first plot uses the default background color as the bg parameter is not specified.
- Grid: The second plot includes a grid with gray dotted lines (`grid(col = "gray", lty = "dotted", lwd = 0.5)`), while the first plot does not have a grid as the grid() function is not called. [2]

7. Scatterplot with best fit line

- We can add a line of best fit (a regression line) to your scatterplot using the `abline()` and `lm()` functions. `lm()` is used to fit linear models, and `abline()` adds a straight line to the plot.

```r
# Create the scatterplot
plot(tb$wt,
     tb$mpg,
     main = "Scatter Plot of Mileage vs. Weight",
     xlab = "Weight", ylab = "Miles per gallon (mpg)",
     pch = 16,
     col="blue")

# Fit a linear model
fit <- lm(tb$mpg ~ tb$wt)
# Add a regression line
abline(fit, col = "red")
```

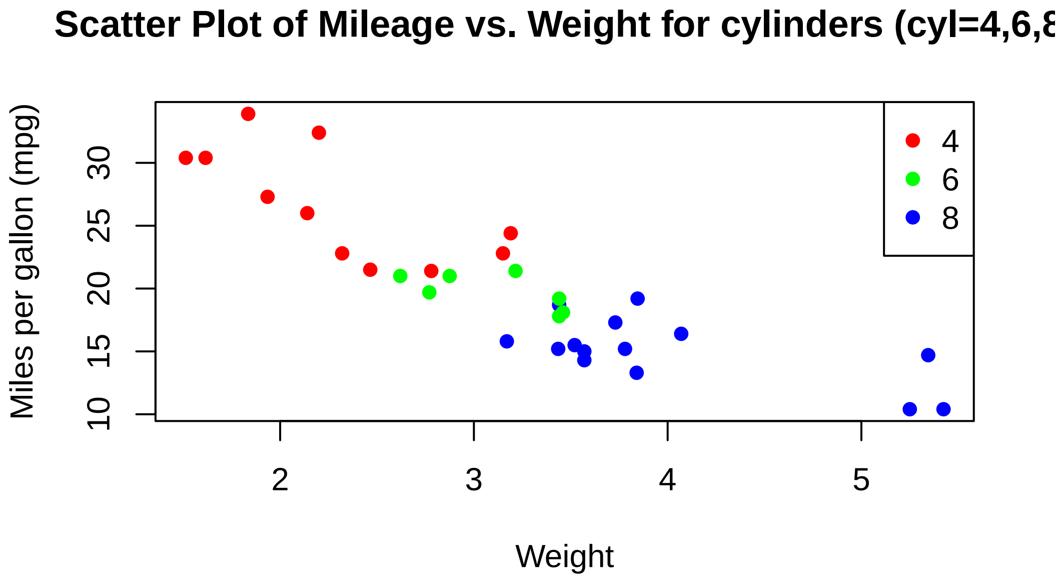


8. Discussion:

- `lm(tb$mpg ~ tb$wt)` fits a linear model predicting `mpg` from `wt`. The `~` operator is a formula operator in R that separates the response variable (on the left of `~`) from the predictor variables (on the right of `~`). `abline(fit, col = "red")` subsequently adds the regression line to the plot, drawn in red color. [3]

9.

```
# Create the scatterplot with points colored by cyl
plot(tb$wt,
     tb$mpg,
     main = "Scatter Plot of Mileage vs. Weight for cylinders (cyl=4,6,8)",
     xlab = "Weight", ylab = "Miles per gallon (mpg)",
     pch = 16,
     col=c("red","green","blue")[tb$cyl])

# Add a legend
legend("topright",
       legend = levels(tb$cyl),
       col = c("red", "green", "blue"),
       pch = 16)
```



10. Discussion

- In the snippet `col=c("red","green","blue")[tb$cyl]`, we assign the color of the data points according to the `cyl` variable. It translates to us giving distinct colors for different `cyl` values in the plot.
- Moreover, when we incorporate the `legend()` function with parameters such as `"topright"`, we place a legend at the top-right corner of the plot. The identifiers in the legend are corresponding to the unique values of the `cyl` variable, and we've made sure the color scheme and point design in the legend are consistent with those in our plot. [2]

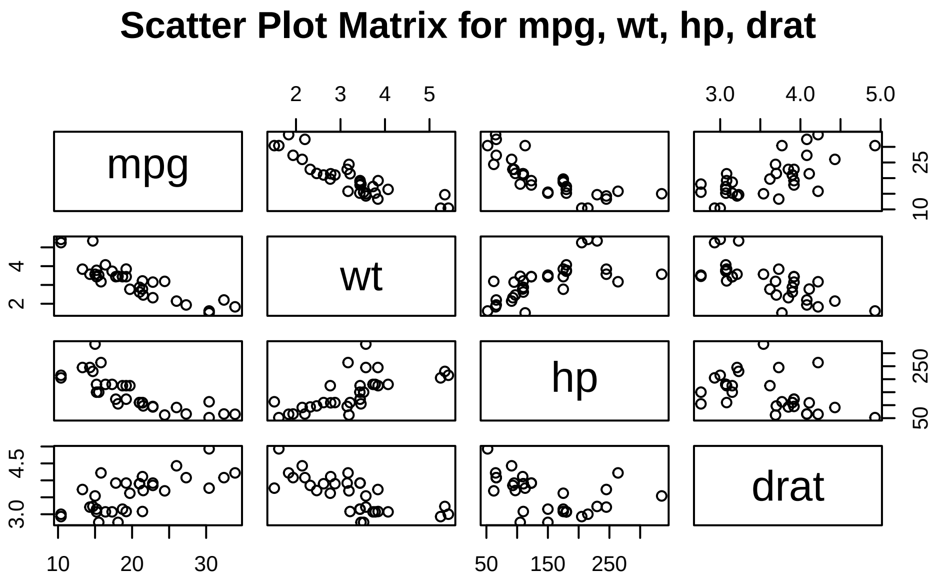
Scatterplot Matrix

1. A scatter plot matrix, also known as a pairs plot or SPLOM, is a powerful visual tool that helps us in depicting the pair-wise relationships among a group of variables. In this matrix, every distinct variable from the dataset is charted against each other in a grid-like structure. This enables us to delve into the associations between variable pairs and identify possible trends or patterns in the dataset.
2. When dealing with multivariate datasets, scatter plot matrices can be remarkably handy. They equip us with a rapid way to discern potential correlations among variable pairs – whether they are strong, weak, or non-existent. It's also a convenient method to spot non-linear relationships between variables. In addition, it's a beneficial tool to recognize outliers or peculiar data points and to observe clusters or collections of observations, as suggested by the patterns within the scatter plots. [4]

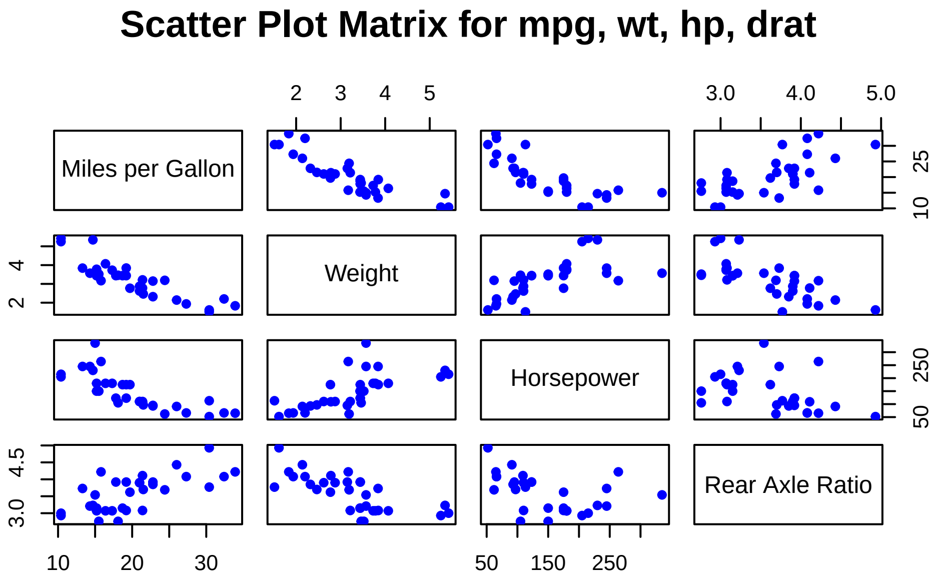
Scatterplot Matrix Using `pairs()`

1.

```
# scatter plot matrix for mpg, wt, hp, drat
pairs(tb[,c("mpg","wt","hp","drat")],
      main = "Scatter Plot Matrix for mpg, wt, hp, drat"
      )
```



```
# scatter plot matrix for mpg, wt, hp, drat
pairs(tb[,c("mpg","wt","hp","drat")],
      main = "Scatter Plot Matrix for mpg, wt, hp, drat",
      pch = 19,
      labels = c("Miles per Gallon", "Weight", "Horsepower", "Rear Axle Ratio"),
      col = c("blue"),
      cex = 0.8
      )
```



Discussion:

Symbol: The pch parameter is used to specify the symbol that represents data points in a plot.

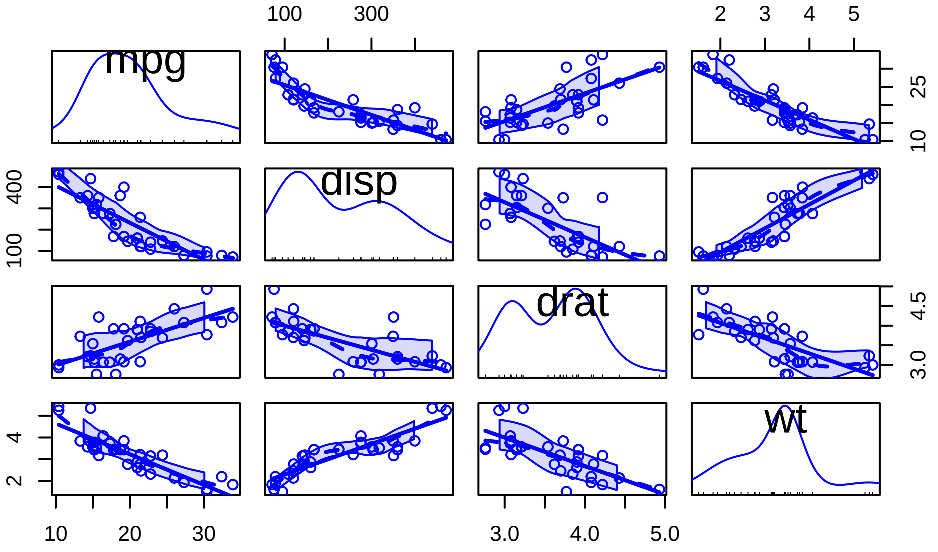
Labels: The labels parameter lets you customize the variable labels that appear on the diagonal:

Color: You can specify different colors for the points in each scatter plot with the col parameter.

Point Size: The cex parameter controls the size of the points in the scatter plot.

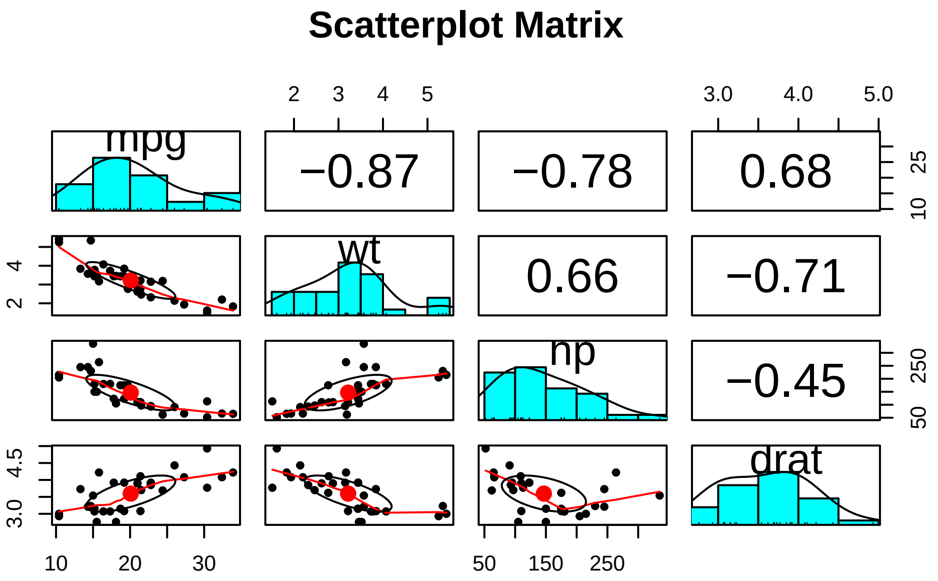
Scatterplot Matrix Using scatterplotMatrix()

```r
# 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")
```



References

[1]

Everitt, B. S., & Hothorn, T. (2014). A Handbook of Statistical Analyses Using R. Chapman and Hall/CRC. (

[2]

R Core Team. (2023). R: A language and environment for statistical computing. R Foundation for Statistical Computing. https://www.R-project.org/

[3]

Fox, J., & Weisberg, S. (2019). An R Companion to Applied Regression (3rd ed.). Sage.

[4]

Everitt, B., & Hothorn, T. (2011). An introduction to applied multivariate analysis with R. Springer.