PH 718 Data Management and Visualization in R

Part 3: Basic Graphics in R

Zhiyang Zhou (zhou67@uwm.edu, zhiyanggeezhou.github.io)

2025/01/26 12:26:23

---

plot()

- Primary way to plot data in R.
- `plot(x, y)` produces a scatterplot of the numbers in `x` vs. the numbers in `y`.
- Additional options available. For example, passing in the argument `xlab` will result in a label on the x-axis. To find out more information about the `plot()` function, type `?plot`.

```
set.seed(718) # to make the subsequent randomization reproducible
x <- rnorm(100) # yield 100 random numbers following N(0,1)
y <- rnorm(100)
plot(x, y,
     xlab = "this is the x-axis", ylab = "this is the y-axis",
     main = "Plot of X vs Y")
```

contour()

- Producing a *contour plot* in order to represent three-dimensional data, like a topographical map. It takes three arguments:
  - a vector of the `x` values (the first dimension),
  - a vector of the `y` values (the second dimension), and
  - a matrix whose elements correspond to the `z` value (the third dimension) for each pair of (`x`, `y`) coordinates.

```
x <- seq(-pi, pi, length = 50)
y <- x
f <- outer(x, y, function(x, y){x^2+y^2}) # outer product of x and y
contour(x, y, f)
```

image()

- Working the same way as `contour()`, and producing a *heatmap* (a color-coded plot whose colors depend on the `z` value).

```
image(x,y,f)
```

persp()

- Producing a three-dimensional plot.

```
persp(x, y, f)
persp(x, y, f, theta = 30) # theta and phi control the angle at which the plot is viewed
persp(x, y, f, theta = 30, phi = 20)
persp(x, y, f, theta = 30, phi = 70)
persp(x, y, f, theta = 30, phi = 40)
```

An example (ISL, pp. 54)

- The `College` data set can be found on the book website (https://www.statlearning.com/s/College.csv). It contains a number of variables for 777 different universities and colleges in the US. The variables are
  - `Private`: Public/private indicator
  - `Apps`: Number of applications received
  - `Accept`: Number of applicants accepted
  - `Enroll`: Number of new students enrolled
  - `Top10perc`: New students from top 10% of high school class
  - `Top25perc`: New students from top 25% of high school class
  - `F.Undergrad`: Number of full-time undergraduates
  - `P.Undergrad`: Number of part-time undergraduates
  - `Outstate`: Out-of-state tuition
  - `Room.Board`: Room and board costs
  - `Books`: Estimated book costs
  - `Personal`: Estimated personal spending
  - `PhD`: Percent of faculty with Ph.D.'s
  - `Terminal`: Percent of faculty with terminal degree
  - `S.F.Ratio`: Student/faculty ratio
  - `perc.alumni`: Percent of alumni who donate
  - `Expend`: Instructional expenditure per student
  - `Grad.Rate`: Graduation rate
- Use the `read.csv()` function to read the data into `R`. Call the loaded data `college`.

```
college = read.csv(file = "https://www.statlearning.com/s/College.csv")
```

- The first column is the name of each university. We don't really want `R` to treat this as a variable. Instead, it is better to have `row.names` recording these names before removing them. Try the following commands:

```
rownames(college) <- college[, 1]
college <- college[, -1]
View(college)
```

- Use `summary()` function to produce a numerical summary of variables in the data set.

```
summary(college)
```

- Use `pairs()` function to produce a scatterplot matrix of the first ten variables of the data.

```
pairs(data.matrix(college[,1:10])) # data.matrix() converting a data frame to a numeric matrix
```

- Use `boxplot()` function to produce side-by-side boxplots of `Outstate` versus `Private`.

```
boxplot(Outstate ~ Private, data = college)
```

- Create a new qualitative variable, called `Elite`, by dividing universities into two groups based on whether or not the proportion of students coming from the top 10% of their high school classes exceeds 50%.

```
Elite <- rep("No", nrow(college))
Elite[college$Top10perc > 50] <- "Yes"
Elite <- as.factor(Elite)
college <- data.frame(college, Elite)
```

- Use `summary()` function to see how many elite universities there are.

```
summary(college$Elite)
```

- Use `boxplot()` function to produce side-by-side boxplots of `Outstate` vs. `Elite`.

```
boxplot(Outstate ~ Elite, data = college)
```

- Use `hist()` function to produce histograms with different numbers of bins for quantitative variables. You may find the command `par(mfrow = c(2, 2))` useful: it will divide the print window into four regions so that four plots can be made simultaneously.

```
par(mfrow = c(2, 2))
hist(college$Top25perc, breaks = 1, freq = T, main = 'breaks=1')
hist(college$Top25perc, breaks = 10, freq = T, main = 'breaks=10')
hist(college$Top25perc, breaks = 100, freq = T, main = 'breaks=100')
hist(college$Top25perc, breaks = 1000, freq = T, main = 'breaks=1000')
par(mfrow = c(1, 1))
```