PH 718 Data Management and Visualization in R

Part 6: Basic Graphics in R

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

2026/03/14 23:00:15

Plotting in Base R

Why visualization matters

Before performing statistical modeling or formal inference, it is often helpful to visualize the data first. Visualization allows us to:

- Understand the distribution of variables
- Explore relationships between variables
- Compare groups
- Detect unusual observations or outliers

An illustrative example: the College dataset

- The College dataset contains information on 777 universities and colleges in the United States. Some variables include:
  - 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. degree
  - 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

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

- 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:

```r
rownames(college) <- college[, 'X']
college <- college[, names(college) != 'X']
```

- Use `summary()` function to produce a numerical summary of variables in the data set and factorize categorical variable for later use.

```r
summary(college)
college$Private = factor(college$Private)
```

Exploring relationships via `plot()`

- Suppose we want to know: Do colleges that receive more applications also accept more students?
- We can examine this relationship using a scatterplot.

```r
plot(
  college$Apps,
  college$Accept,
  xlab = "Number of Applications",
  ylab = "Number of Acceptances",
  main = "Applications vs Acceptances",
  col = "blue",
  pch = 19
)
abline(lm(college$Accept ~ college$Apps), col = "red")
```

The function `plot()` is the most basic plotting function in base `R`. `plot()` creates a scatterplot of y versus x. Common options include:

Argument	Purpose
`xlab`	label for x-axis
`ylab`	label for y-axis
`main`	title
`col`	color
`pch`	point shape
`cex`	point size

Visualizing many relationships: `pairs()`

Sometimes we want to examine relationships among many variables at once. The function `pairs()` creates a scatterplot matrix.

```r
pairs(college[,2:6])
```

- Each panel shows the relationship between a pair of variables. This allows us to quickly identify:
  - correlations
  - nonlinear relationships
  - outliers

Modern alternative to `pairs()`: `GGally::ggpairs()`

```r
GGally::ggpairs(college[,2:6])
```

Comparing groups via `boxplot()`

We may want to compare the distribution of a variable across groups. For example: Do private colleges charge higher out-of-state tuition than public colleges?

```r
boxplot(
  Outstate ~ Private,
  data = college,
  col = c("lightblue","lightgreen"),
  main = "Out-of-State Tuition by College Type",
  ylab = "Out-of-State Tuition"
)
```

What does a boxplot show? A boxplot summarizes the distribution of a variable using five key statistics (five-number summary): the minimum, 1st quartile (Q1), median, 3rd quartile (Q3), maximum. It major components include:

- The box represents the interquartile range (IQR=Q3-Q1) and contains the middle 50% of the data.
  - Bottom of box = 25th percentile (Q1)
  - Top of box = 75th percentile (Q3)
  - The median line = the median (50th percentile).
    - If the median is not centered in the box, the distribution may be skewed.
- The whiskers
  - Extend from the box to the most extreme values that are not considered outliers.
  - Typically extend to Q1-1.5xIQR and Q3+1.5xIQR, where IQR = Q3 - Q1.
- Potential outliers: Points beyond the whiskers are plotted individually. They may indicate:
  - unusual observations
  - measurement errors
  - interesting cases worth further investigation

Interpreting the boxplot:

- Interpreting the boxplot. For instance, from the tuition boxplot we can examine:
  - Differences in medians: If the median line for private schools is higher than for public schools, private schools tend to charge more.
  - Differences in variability: A wider box indicates greater variability in tuition.
  - Presence of outliers: Some schools may have extremely high tuition.

Boxplots allow us to quickly compare:

- central tendency (median)
- spread (IQR)
- skewness
- outliers

across multiple groups. Unlike a simple average, they provide a robust summary of the distribution.

Another illustrative question: Do elite colleges charge higher out-of-state tuition than others?

- 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%.

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

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

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

Understanding distributions via `hist()`

To understand the distribution of a quantitative variable, we can use a histogram.

```r
hist(
  college$Top25perc,
  main = "Distribution of Top 25% Students",
  xlab = "Percent of Students in Top 25%"
)
```

We can change the number of bins using the `breaks` argument. You may find the command `par(mfrow = c(2, 2))` useful: it will divide the window into four regions so that four plots can be shown simultaneously.

```r
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))
```

Heatmaps of correlation matrix via `image()`

When a dataset contains many numeric variables, it is useful to examine how they are related. A convenient way to visualize correlation matrix is through a heatmap, where colors represent the magnitude of correlation coefficients.

```r
num_var_idx = NULL
for (j in 1:ncol(college)) {
  if(is.numeric(college[, j])) {
    num_var_idx <- c(num_var_idx, j)
  }
}
college_num <- college[, num_var_idx] # subsetting numeric variables
cor_matrix <- cor(college_num)
col_palette <- colorRampPalette(c("green","white","red"))(100)
image(
  1:ncol(cor_matrix),
  1:ncol(cor_matrix),
  cor_matrix,
  axes = FALSE,
  xlab = "",
  ylab = "",
  main = "Heatmap of Covariance Matrix",
  col = col_palette
)
axis(1, at = 1:ncol(cor_matrix), labels = colnames(cor_matrix), las = 2)
axis(2, at = 1:ncol(cor_matrix), labels = colnames(cor_matrix), las = 2)
```

Interpretation:

- Colors close to `green` or `red` indicate larger corrlation coefficients.
- Large positive values suggest variables increase together.
- Negative values indicate variables move in opposite directions.

Heatmaps help us quickly identify:

- clusters of strongly related variables
- redundant variables
- patterns across many variables at once

This is especially useful when working with high-dimensional datasets.