Data analysis and visualization using R (1)

R basics - plotting

Michiel Noback

September 2019

Plotting

Basic embedded plot types

- Looking at numbers is boring - people want to see pictures!
- There are a few plot types supported by core R that deal with (combinations of) vectors:
  - scatter (or line-) plot
  - barplot
  - histogram
  - boxplot

We'll only look at the bare basics because we are going to do it good with ggplot2 in the next course.

Scatter and line plots

- Meet `plot()` - the workhorse of R plotting.

```
time <- c(1, 2, 3, 4, 5, 6)
response <- c(0.09, 0.30, 0.41, 0.48, 0.72, 1.12)
plot(x = time, y = response)
```

Plot decoration

- By passing arguments to `plot()` you can modify or add many features of your plot
- Basic decoration includes
  - adjusting markers (`pch = 19`, `col = "blue"`)
  - adding connector lines (`type = "b"`) or removing points (`type = "l"`)
  - adding axis labels and title (`xlab = "Time (hours)"`, `ylab = "Systemic response"`, `main = "Systemic response to agent X"`)
  - adjusting axis limits (`xlim = c(0, 8)`)

Here is a more complete plot using a variety of arguments.

```
plot(x = time, y = response, pch = 19, type = "b", xlim = c
     xlab = "Time (hours)", ylab = "Systemic response (a.u
     main = "Systemic response to agent X", col = "blue")
```

Adjusting the plot symbol

When you have many data points they will overlap. Using transparency with the `rgb(„ alpha=)` color definition and/or smaller plot symbols (`cex=`) solves this.

```
x <- rnorm(1000, 10, 2); y <- x + rnorm(1000, 0.5, 0.5)
plot(x, y, pch = 19, cex = 0.6,
     col = rgb(red = 0, green = 0, blue = 1, alpha = 0.2))
```

Barplots

- Barplots can be generated in two ways
  - By passing a factor to plot() - simple of level frequencies
  - By using barplot()

```
persons <- as.factor(sample(c("male", "female"), size = 100
plot(persons)
```

barplot() with a vector

- barplot() can be called with a vector of heights (frequencies) or a table() object

```
frequencies <- c(22, 54, 12, 29)
barplot(frequencies, names = c("one", "two", "three", "four
```

barplot() with a table object

```
table(persons)
```

```
persons
female   male
    57     43
```

```
barplot(table(persons))
```

barplot() with a 2D table object

Suppose you have these data:

```
set.seed(1234)
course <- rep(c("biology", "chemistry"), each = 10)
passed <- sample(c("Passed", "Failed"), size = 20, replace
tbl <- table(passed, course) # the order matters!
tbl
```

```
        course
passed   biology chemistry
  Failed       6         8
  Passed       4         2
```

You can create a stacked bar chart like this. (The `xlim=` setting is a trick to get the legend beside the plot)

```
barplot(tbl, col=c("red", "darkblue"),
        xlim=c(0, ncol(tbl) + 2), legend = rownames(tbl))
```

Using the `beside=TRUE` argument, you get them side by side:

```
barplot(tbl, col=c("red", "darkblue"), beside = TRUE,
        xlim=c(0, ncol(tbl)*2 + 3), legend = rownames(tbl))
```

Later, we'll see another data structure to feed to barplot: matrix.

Histograms

Histograms help you visualise the distribution of your data

```
male_weights <- c(rnorm(500, 80, 8)) ## create 500 random
hist(male_weights)
```

Using the `breaks` argument, you can adjust the bin width. Always explore this option when creating histograms!

```
par(mfrow = c(1, 2)) # make 2 plots to sit side by side
hist(male_weights, breaks = 5, col = "gold", main = "Male w
hist(male_weights, breaks = 25, col = "green", main = "Male
```

Boxplots

This is the last of the basic plot types. A boxplot is a visual representation of the *5-number summary* of a variable.

```
persons <- rep(c("male", "female"), each = 100)
weights <- c(rnorm(100, 80, 6), rnorm(100, 75, 8))
#print 6-number summary (5-number + mean)
summary(weights[persons == "female"])
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  57.68   69.36   74.12   73.99   78.97   92.23
```

```r
par(mfrow = c(1, 2)) # make 2 plots to sit side by side
# create boxplots of weights depending on sex
boxplot(weights ~ persons, ylab = "weight")
boxplot(weights ~ persons, notch = TRUE, col = c("yellow",
```

Use `varwidth=TRUE` when you want to visualize the difference in group sizes.

Plotting rules

Plots should always have these decorations

- Axis labels indicating measurement type and its units
- If multiple data series are plotted: a legend
- Often a title as well