Creativity1.r: Explanation

Read the Rstart document for information on reading data into R and how R stores variables and data frames.

Goals of code:

- Display the creativity data (histogram, boxplot, dotplot)
  - As one group or for each treatment
- Calculate summary statistics for each group

These notes are meant to be used while you have R open and are executing the code in creativity1.r. Each command (or group of related commands) will explained and important parts of the result will be discussed. It will be easiest to use these notes if you execute each command, then read the notes.

Notes (reminders):
1) I am describing the essential arguments to obtain the sort of results we need for this class. Almost all functions have many more options, specified by additional arguments, that are described in the help files.

2) In many cases, there are multiple ways to accomplish something in R. For example, there are three different graphics platforms (base graphics, lattice graphics and ggplot graphics). I do not illustrate all ways. I illustrate what I believe are the best for beginners. For example, I illustrate the use of base graphics. If you know and are more comfortable with other approaches, feel free to use them.

If you use some other approach, please check that there is not some subtle difference (there are many in R). You can do this by running your approach on my examples or the self-assessment. Or, ask me or the TA.

**Reading data**: `creativity <- read.csv('creativity.csv', as.is=T)`
Described in the Rstart document.

**Histogram**: `hist(creativity$score)`
draws a histogram of the variable named in the function. Not easy (at least in base graphics) to draw side-by-side histograms as in the book.

**Boxplot**: `boxplot(creativity$score)`
Draws a boxplot. The format follows the description in section 1.5.1, except that "far-away" points, > 3 IQR, are not distinguished from "not-so-far-away" points, > 1.5 IQR.

**Side-by-side Boxplots**: `boxplot(split(creativity$score, creativity$treatment))`
Functions can be nested, as illustrated here. The split() function splits the first argument (all 47 scores) into groups defined by the value of the second argument (the treatment name). The result of the split function go to boxplot(), which plots the groups side-by-side.

**Dot plots**: `stripchart(score~treatment, data=creativity, pch=1)`
When there aren't too many data points, a dot plot portrays the data without summarizing it first. The stripchart() function draws a horizontal dot plot. It accepts a formula. The left side of the `~` is the response to be plotted; the right side is the grouping variable. The `data=` tells R where to find the two variables. Further description of `data=` is in my introduction to R document.

The `pch=19` argument changes the plotting character to an filled circle. The default for stripchart is pch=0, which is a square box. I like circles better. `pch=1` is an open circle. There are 25 plot symbols, as well as any character (e.g., `pch='a'`). To see the 25 symbols, type `plot(1:25, rep(1,25), pch=1:25)`. If you prefer square boxes, omit the pch= argument.

**Numeric summaries**: `summary(), mean(), sd(), var(), IQR()` These five functions produce numeric summaries of the data.
`summary()` gives a 6 values: minimum and maximum, 1st quartile, median and 3rd quartile and the average.
`mean(), sd(), var(), IQR()` give the average, standard deviation, variance, and inter-quartile range.

**Numeric summary of multiple groups**:
`tapply(creativity$score, creativity$treatment, mean)`
The tapply() function applies the function you specify (last argument, here mean), to the values in the first argument, categorized by the second argument. The example command will compute the mean for each of the two treatment groups. To get the sd for each group, replace mean with sd. You can use tapply for functions that return more than one value. Ask if you don't understand the format of that output.