In-Class Computing Project 5

Math 253: Statistical Computing & Machine Learning

Drawing a histogram

In today's programming task, you'll recreate a basic graphics functionality in R, drawing a histogram. Of course, in actual practice, you would never do this. The histogram programs in base graphics, lattice graphics, and ggplot2 are easy to use and likely more powerful than something you would produce. The purpose here is to give you experience with graphics-related calculations.

Background

The histogram will be drawn by two base graphics functions, `plot()` and `polygon()`. Almost all of the activity is the data manipulation needed to create a data frame appropriate for specifying a histogram.

The objects that you create will live in the "global environment", named `.GlobalEnv`. This has been true for all the previous programming activities as well. Starting in the next task, Task 06, we'll work with objects existing in an environment of our own creation.

Task 1

Load the `mosaicData` package, so that you have access to the `Galton` data table.

Create an object `nbins` and give it the value 10.

Task 2

Create an object, `evenly_spaced`, that consists of `nbins + 1` evenly spaced values between the minimum and maximum of the `height` variable in `Galton`.

Task 3

Create an object, `bin_counts`, that contains the number of cases in `Galton` where `height` falls into each bin. Hint: You might find `cut()` and `table()` useful functions.

Task 4

Make a data frame, `hist_basics`, that contains three variables, `xL`, `xR`, and `count`. The `xL` variable should be the values of `evenly_spaced` except for the last one. The `xR` variable should be the values of `evenly_spaced` except for the *first* one. `count` will be the same as `as.numeric(bin_counts)`.

Task 5

Here are two functions that will take each row in `hist_basics` and turn it into five rows, with the four points that define each bar in the histogram and a row containing `NA`. Retype them into your `.R` script.

```
make_one_bar <- function(point) {
    xLeft <- point$xL
    xRight <- point$xR
    height <- point$count
    res <- data.frame(x = c(xLeft, xLeft, xRight,
        xRight, NA), y = c(0, height, height,
        0, NA))
    res
}
# ================
one_to_five <- function(hist_data) {
    bars <- NULL
    for (k in 1:nrow(hist_data)) {
        new_bar <- make_one_bar(hist_data[k, ])
        bars <- rbind(bars, new_bar)
    }
    bars
}
```

Use these two functions (you'll have to include them in your script) to create a data frame called `My_bars` that has five rows for each row in `hist_basics`.

Task 6

Draw the histogram.

- `plot(My_bars, type="n")` will make an empty plotting frame to hold the histogram.
- `lines(My_bars)` will draw the bounding lines of each bar.
- `polygon(My_bars)` will draw a polygon. The `col =` argument will set the color.