Lab 12

Statistical Computing, 36-350 Friday December 4, 2015

Today's agenda: practicing simulation tasks, in the context of fun and games: darts!

Background. Darts is a game where you throw metal darts at a dartboard, and receive different scores for landing the dart in different regions.



Recall, as explained in lecture:

- Land in a pie slice, obtain a score of the corresponding number (e.g., dotted region is worth 20 points)
- Land in the "double ring", get double the score (e.g., solid region is worth 60 points)
- Land in the "triple ring", get triple the score (e.g., striped region is worth 40 points)
- Land in the "double bullseye" (DB), get 50 points
- Land in the "single bullseye" (DB), get 25 points

For this lab, you'll need the functions scorePositions() and drawBoard() that you used in lecture. These functions are available in the file http://www.stat.cmu.edu/~ryantibs/statcomp/labs/darts-helpers.R, and to load them into your R session (instead of copying and pasting the function code), you can simply call:

source("http://www.stat.cmu.edu/~ryantibs/statcomp/labs/darts-helpers.R")

Now you should be able to call scorePositions() and drawBoard(), right in this R session. You'll also note that the list board, containing dart board measurements (all in mm) has also been loaded into memory. So, for example:

drawBoard(board)



scorePositions(0,0,board)

[1] 50

scorePositions(0,80,board)

[1] 20

scorePositions(0,171,board)

[1] 0

Let X, Y denote the horizontal and vertical positions of a dart throw, measured from the center of the board. In this lab, we'll consider the model:

$$X \sim N(\mu_x, \sigma_x^2), \quad Y \sim N(\mu_y, \sigma_y^2)$$

so that (μ_x, μ_y) is the aiming location, measured in mm (with (0,0) denoting the board's center), and σ_x, σ_y are the standard deviations of in the x and y directions, again in mm, respectively. We will refer to this as the "normal dart model".

1. Generate 100 throws from the normal dart model, aimed at the center of the board, and with $\sigma_x = \sigma_y = 50$. Plot the dart board, and plot the resulting positions on top, with pch=20 (to make small solid dots). Then compute the scores of your 100 throws, and draw in text, next to each throw, its score. (Hint: for this last part, use text().)

- 2. Write a function, compute.average.score(), that takes in the following five arguments: n, mu.x, mu.y, sigma.x, sigma.y. Your function should generate n throws from the normal dart model, with mean given by mu.x, mu.y, and standard deviations sigma.x, sigma.y in each direction. It should then compute their scores, and return the average score. (No plotting performed by this function.) Try it out by calling compute.average.score(n=100,mu.x=0,mu.y=0,sigma.x=50,sigma.y=50).
- 3. Write a function compute.score.curve(), that takes in the following five arguments: n, mu.x, mu.y, sigma.min, sigma.max, with the last two having default values of 0 and 100, respectively. Your function should carry out the following steps:
 - Create a vector sigmas of standard deviations, with 100 values in between sigma.min and sigma.max. Create a new vector scores of the same length.
 - For every standard deviation value in the vector sigmas, set sigma.x and sigma.y equal to this value, and compute the average score of n throws, under the normal dart model, with aiming point mu.x, mu.y, and the given standard deviations sigma.x, sigma.y. Save this average score in the appropriate entry of the scores vector.
 - Return a list with two components, sigmas (the standard deviation values you inspected) and scores (the average scores you computed).
- 4. Call your function compute.score.curve(), with n=5000, the default values of sigma.min and sigma.max, and three different settings for the aiming point:
 - mean.x=0, mean.y=0;
 - mean.x=0, mean.y=103;
 - mean.x=-32, mean.y=-98.

Explain in words, where are these aiming points on the dart board? Then, produce a plot that shows your three score curves (corresponding to the three aiming points), as functions of the standard deviation. You should draw them as different colored lines, and they should all appear on the same plot. Label the axes appropriately, and include a legend that specifies what the different colors portray.

5. Now let's interpret your plot from the last question. According to average score, what's the best place to aim, among the three locations you considered, when the standard deviations in the x and y directions are both 8? 25? 80? Can you offer (rough) explanations of why this would be the case?

Bonus 1. Consider the following alternative model for dart throws:

$$X \sim \text{Unif}[\mu_x - \sigma_x, \mu_x + \sigma_x], \quad Y \sim \text{Unif}[\mu_y - \sigma_y, \mu_y + \sigma_y]$$

where Unif[a, b] denotes the uniform distribution between a and b. Repeat the tasks in questions 2 through 5. Have the high-level results changed?

Bonus 2. Install the darts package (available on CRAN), and load it into your R session. Play around with the tools that this package makes available. Use it to check your results from question 5, about where is the best place to aim. Test out its functionality to estimate the standard deviation in the normal dart throw model, based on scores alone. (Hint: test it out by running a simulation where you know the standard deviation, compute the scores, and then pass the scores to its estimation procedure to see how it does.)