

R Class Homework 10

Assignment

- 1) Read Part 4 of the custom functions tutorial: Developing from blank screen to complete function.
- 2) Complete the `boot_power()` custom function described below.

`boot_power()` custom function

This is the second custom function for R Class Exercise 3. This function will be used to perform a parametric bootstrap power analysis. This function will use the `sig_t()` custom function you created for HW09. This should ideally be a flexible function that could do this analysis to any two sample groups from any dataset.

The plan

Our specific interest is in the SLA (specific leaf area) dataset provided with Class Exercise 3. The data columns contain SLA values for different leaf structural groupings. The three leaf-structural groupings we are interested in are columns 'l' (lamina), 'lv' (lamina plus midvein), and 'whole' (whole leaf). The samples to compare by t-test are clay (`soil=="C"`) and sand (`soil=="S"`) soils.

We want to know how much statistical power we currently have to detect significant differences between sand and clay samples for 'l', 'lv', and 'whole'. We'd also like to know how much statistical power we *would* have if we increased our sample size. Comparing the power for the different leaf structural groupings will give us an indicator of which SLA metric best distinguishes plant communities across soil types.

The function

Arguments:

- `data`: the dataframe containing the data of interest
- `data.col`: The name of the column containing the data to analyze. (E.g., 'l', 'lv', 'whole'.)
- `group.col`: The name of the column containing the grouping variables (for our dataset, "soil").
- `group1`: The grouping variable in `group.col` for group1 (e.g., "S").
- `group2`: The grouping variable in `group.col` for group2 (e.g., "C").
- `n.boot`: the number of bootstrap iterations to run. Assign a default of 1000.
- `n1`: the sample size for the randomized sample for group1 (could be the actual sample size, or an inflated sample size to test the hypothetical power with more samples).
- `n2`: the sample size for the randomized sample for group2.
- `alpha`: The significance threshold. Give it a default value of 0.05.

Function outline:

- Make a subsetting data frame that contains only the grouping variables of interest, and no NAs in the group column or the data column of interest.
- Calculate the observed mean value for each group, store them as objects 'm1' and 'm2'.
- Calculate the degrees of freedom ($= n1 + n2 - 2$).
- Calculate the pooled variance (see equation in HW09) and store as object 'pooled.var'.
- Use a loop with 'n.boot' iterations to grow or fill a vector of p-values obtained from t-tests on randomly generated data. (See Part 3 of the For Loop tutorial for growing vectors.) For each iteration:
 - o Use function `rnorm()` to generate a random sample from a normal distribution with mean = m1 and sd = `sqrt(pooled.var)`. Save as 'rand1'.
 - o Do the same for a randomized version of group2 data and save as 'rand2'.
 - o Make a data frame with a group column and a data column. The data column contains 'rand1' on top of 'rand2' and the group column specifies their group association (mimicking the structure of the original dataset).
 - o Use the `sig_t()` function to return the p-value from a t-test on the two artificial groups.
 - o Deposit that p-value into your growing vector of p-values.
- Determine what proportion of p-values in your vector are less than 'alpha'. That proportion is your *statistical power*. Save that number as object 'power'.
- Return the 'power' object to the global environment.