Week 5 Lab problems

EEB 429
Bhaskar Kumawat

Note: You might find this R-markdown cheat sheet useful. Refer to this cheat sheet on equations in R-markdown for writing nice mathematical equations.

A. Using R-markdown

1. Create a new R-markdown file titled “LastnameFirstname_Lab5” with the author name as your name, and output as “html_document”.

2. Create a level-1 (largest) header called “Week 5 Problems”

3. Below that, write your full name in **bold lettering.**

4. Below that, create a level-2 header (second largest) called “Problem A”.

5. Recreate the following equation below the heading. (See equations cheat sheet above)

$$f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

6. Start a new R code block below the equation (using ```` ```{r} <code> ``` ````)

7. Inside the code block, define a function called `Gaussian` that takes in three arguments, `x`, `mean`, and `std`. Code the function to calculate the value as defined in the equation above (where, `x`=x, `mean`=μ, and `std`=σ). You might have to use the in-built R functions `exp()` and `sqrt()`.

8. Below the code block, create a level-3 header called “Gaussian plot”.

9. Below the header, create another R code block with the `echo=FALSE` parameter.

10. Inside the code block, define two variables `mu` (with value 2) and `stdev` (with value 1).

11. After that, create a vector called `xvals` that goes from -1 to 5 in steps of 0.01. (Using `seq`)

12. Then, call the function `Gaussian` with the arguments `xvals`, `mu`, and `stdev` and store the result in a variable called `yvals`.

13. Plot `xvals` (x-axis) against `yvals` (y-axis) using the plot function.

B. Using apply on a data-frame

In the **same R-markdown file** as the above problem:

1. Create a level-2 header called “Problem B” below the answers to the previous question.

2. Start a new R code block below the equation (using ` ```{r} <code> ``` `)

3. Use the `apply` function on the `mtcars` dataframe to get the mean value in each column of the dataframe and store these in a variable called `all_means`.

4. Use the `apply` function on the `mtcars` dataframe to get the standard deviation of values in each column of the dataframe and store these in a variable called `all_stds`.

5. Use the `apply` function on the `mtcars` dataframe to get the minimum value in each column of the dataframe and store these in a variable called `all_mins.`

6. Use the `apply` function on the `mtcars` dataframe to get the maximum value in each column of the dataframe and store these in a variable called `all_maxs`.

7. Index the above four vectors to obtain the minimum, maximum, mean value, and standard deviation of the `wt` column.

Submit the entire R-Markdown file AND the generated HTML file on Canvas for full credit.