

Homework 4 Instructions

Kristin Privitera-Johnson

10/26/2021

Part 1 - Programming Assignment

Open a new script in RStudio and create an assignment header using comments.

```
# Name: First Last  
# Homework 4
```

Complete the tasks below. Please label each question and task with comments.

```
##Question 1a####  
# Your R code here  
  
##Question 1b####  
# Your R code here
```

When your script is complete, save it as `LastName_Homework4.R`, then clear your workspace (Workspace/Clear all) and run through your script again to make sure you don't have any `object not found` errors. Then go to the course website to submit your R script.

Question 1: An example of generating simulated “data”

Set a random seed to start the state of R's random number generator.

- Generate 1000 random variables from a Cauchy distribution with a location parameter 5 and scale parameter 1. Store these values in `randomCauchy`. What are the minimum and maximum values? What is the mean value?
- Create a sequence of numbers from 0.01 to 0.985 in increments of 0.025. Call this vector `x`.
- For each value in `x`, find the density, assuming a Beta distribution with shape parameters equal to 5 and 2 respectively. Name this vector `betaDensity`.
- Renormalize the vector `betaDensity` by dividing every observation in the vector by the sum of all the observations. Call this new vector `betaDensityNorm`. The sum of all the observations should now be 1. Check this.
- Create a sequence of integers from 12 to 90 in increments of 2. Call this vector `lengths`.
- The vectors `betaDensityNorm`, and `lengths` should all be of length 40 (check!). Create a random sample of 1000 lengths from the vector `lengths` with weighted probabilities for each length given by the vector `betaDensityNorm`. Call this vector `lengthsSample`.

- g) Create a histogram from the vector `lengthsSample`. Add a density curve. Change the default plot to make your histogram “pretty” (*This is subjective so you can add a title, color, . . . or anything that you feel this plot needs*).

Question 2: Functions & multi-panel plots

For this question we’ll be using the `Cabazon.csv` and `YERockfish.csv` data.

- a) Read in `YERockfish.csv` data and call it `Yelloweye`.
- b) Write a function called `Length.Age.Mat()` that takes at least (i.e, you may add more for plotting) three parameters (arguments): `length`, `age`, and `maturity`. The function will:
- plot the length (y-axis) at age (x-axis) data with proper labels
 - HINT!: In part c) you will reuse this function for a different species that has a different range of length and ages. This will cause issues with your `xlim` and `ylim` specifications. Think of a way to make setting the limits of your plot generalizable.
 - calculate the mean length and mean age for each stage of maturity (Immature, Mature, and Unknown) and store them as vectors called `Length_Means` and `Age_Means`
 - plot the `Length_Means` for Immature and Mature as horizontal lines (e.g., `abline()` function) on the length at age plot. Format the plot to match the plot provided below.
 - return `Length_Means` and `Age_Means` as a list.
- c) Read in `Cabazon.csv` and call it `Cabazon`. Use `Length.Age.Mat()` to create a length at age plot and find the mean lengths and mean ages for `Cabazon`!
- d) Create a multi-panel plot using `Cabazon` and `Yelloweye`’s length at age plots using `layout()` or `par(mfrow)`.

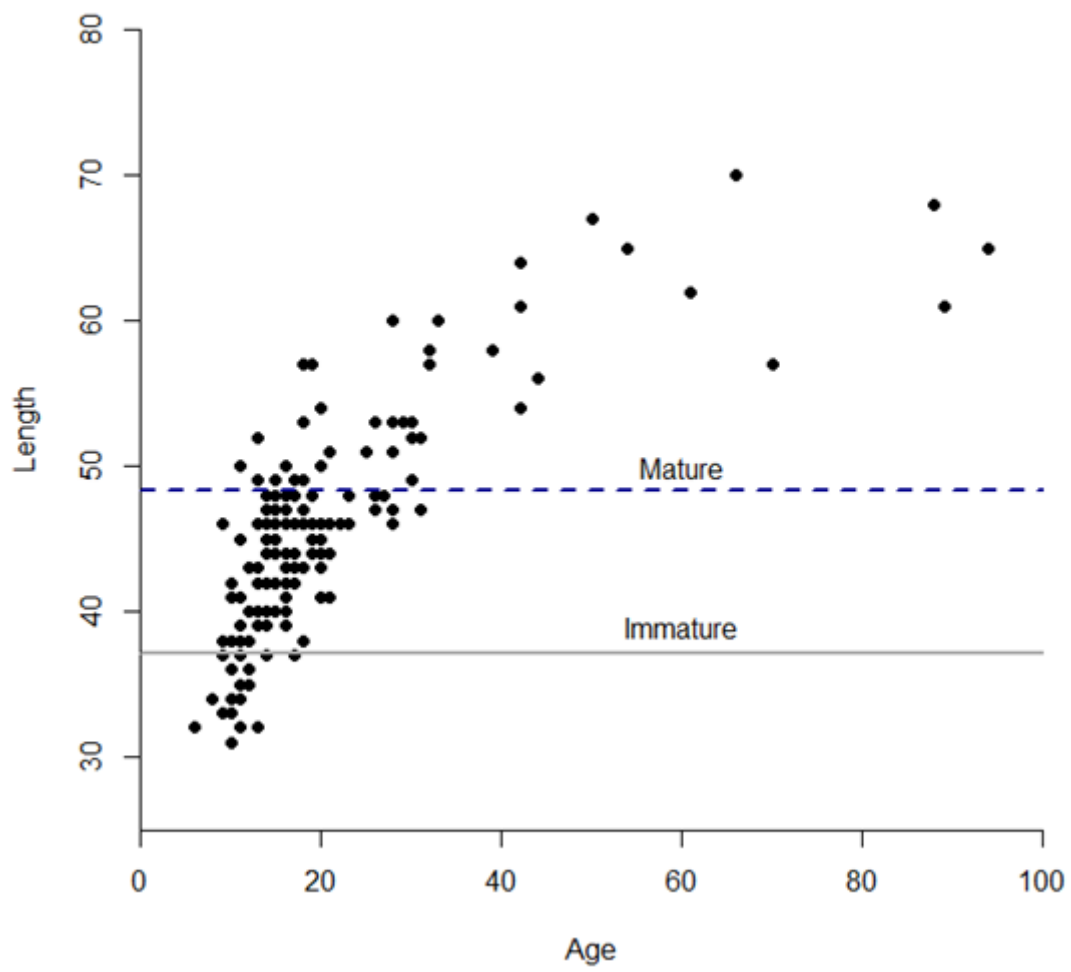


Figure 1: Figure to recreate for part b)