

Handed out: 02/08/2014

Due by 17:25PM on Friday, 02/14/2014

Problem 1. Present the duration and waiting time of Old Faithful eruptions as two colorful pie charts using breaks at every 0.5 minutes.

Problem 2. Let us finish the plot of the correlation between waiting times and durations of Old Faithful data. Recreate the scatter plot of waiting vs. duration times. As we mentioned in class, the best linear assessment in the sense of the least squares fit of a relationship (proportionality) between two or many variables can be achieved with function `lm()`, where `lm` stands for the linear model. The first argument of `lm()` is called `formula` accepts a model which starts with the response variable, `waiting` in our case, followed by a tilde (symbol `~`, read as “is modeled as”) followed by the (so called Wilkinson-Rogers) model on the right. In our case we simply assume that waiting time is proportional to the duration time and that “model” reads: `formula = waiting ~ duration`. The second argument of function `lm()` is called `data` and, in our case, will take value `faithful`, the data set containing our data. Store the result of function `lm()` in a variable. The name of that variable is not essential. Call it `model`. Print the variable. The first component of that variable is the intercept of calculated line with the vertical axis (waiting, here) and the second if the slope of the line. Convince yourself that line with those parameters will truly lie on your graph. Function `abline()` adds a line to the previously created graph. Next, pass the variable `model` to the function `abline()`. Make that line somewhat thicker and red. Use `help(functionName)` to find details about invocations of both `lm()` and `abline()` functions.

Problem 3. You noticed that eruptions clearly fall into two categories, short and long. Let us say that short eruptions are all which have duration shorter than 3.1 minute. Add a new column to data frame `faithful` called `type`, which would have value `'short'` for all short eruptions and value `'long'` for all long eruptions. Next use `boxplot()` function to provide your readers with some basic statistical measures for waiting and then in a separate plot for duration times. Please note that `boxplot()` function also accepts as its first argument a formula such as `waiting ~ type`, where `waiting` is the numeric vector of data values to be split in groups according to the grouping variable `type`. The second argument of function `boxplot()` is called `data`, which in our case will take the name of our dataset, i.e. `faithful`. Find a way to add meaningful legends to your graphs.

Problem 4. Find a way to generate a random variable with a triangular probability. If you do not find your way, please install package “triangle” and use its function `rtriangle()` to generate a vector of random variables with values between 0 and 1. The triangular distribution has a zero probability that the variable will have a value less than 0 and greater than 1. Probability that the variable will have a value between 0 and 0.5 grows linearly with the value of the variable. Similarly, the probability that the variable will have a value between 0.5 and 1 falls linearly with the value of the variable.

Create a histograms with 20 bars to convince yourself that generated values truly fall under a triangular distribution.

Problem 5. Create a matrix with 20 columns and 100 rows. Populate each column with random variable of the type created in problem 4. Do this by writing a function in R. Do not create each vector manually. Try to find a way to present two distributions contained in any two of the columns of your matrix on a single plot. To do that you might want to export the distribution data from two columns into two stand-alone vectors of equal length, e.g. `y1` and `y2`. Plot one distribution first using a call to `plot(x, y1)`, where vector `x` contains the “predictor” or the parameter vector with values between 0 and 1 you selected above. To add the next curve (distribution `y2`) try invoking function `lines(x, y2)`. To improve your diagram, present two curves in different colors and add labels on x and y axis, as well as the title to your graph.

Problem 6. Go back to your matrix from problem 5. Add yet another column to that matrix and populate that column with the sum of original 20 columns. Create a histogram of values in the new column showing that the distribution starts to resemble the Gaussian curve. Add a true, calculated, Gaussian curve to that diagram with the parameters you expect from the sum of 20 random variables of triangular distribution with values between 0 and 1.

Problem 7. Plot the binomial distribution for $p = 0.3$, $p = 0.5$ and $p = 0.8$ and the total number of trials $n = 60$ as a function of k the number of successful trials. For each value of p , determine 1st Quartile, median, mean, standard deviation and the 3rd Quartile. Present those values as a vertical box plot with the probability p on the horizontal axis.

SUBMISSION INSTRUCTIONS:

Your main submission should be an MS Word document containing your code, results produced by that code and brief textual descriptions of what you did and why. Typically, you just copy your code and results from the R console and past them into this Word document. In other words start with text of this homework assignment as the template. Please add any other files that you might have used or generated.

Package everything into a ZIP archive called `E63_LastNameFirstNameHW02.zip`. Naming your file properly is important. We download many files and if they are all named `Assignment02.zip` it becomes hard not to overwrite and loose them. Please do not use archiving tools which do not produce ZIP files. Please do not submit rar or tar archives.

If you are using a Mac, please make sure that your files are **READABLE** to users of Windows. You are welcome to save your work as a PDF file, but please, always submit a Word document, as well. Upload your ZIP archive to the course web site. Every assignment has its own drop box. If you miss the deadline, please submit your solution into the `00_AnyHW_WayLate Drop Box`. Those assignments will be graded as well. **We will chop 5% of your grade for every day you are late.** Your grade for every assignment will be entered as a comment next to your submission.

If you have issues with the formulation of the assignment or the software you are using, please FIRST go to the Discussion Forum on the class web site: <http://isites.harvard.edu/icb/icb.do?keyword=k102025> and check whether someone else raised the same issue and whether the answer is already there. If not, raise the issues yourself. A person from the class or a member of the teaching stuff will respond.

If the issue is not address for a day or two, please send an inquiry to cscie63@fas.harvard.edu. The discussion forum is a very important tool. We all learn from the discussions on the forum.

If we respond to your inquiry to class email address or any email address of the teaching stuff, PLEASE DO NOT RESPOND WITH A THANK YOU NOTE. This is not a joke. We will take 2% of your grade for that week's assignment for every "Thank you note".

We will apply the same penalty to any trivial email. Please do not complain when you loose a few points on your assignment.

If you have issues with the class web site, please let us know right away. In the past, we experienced issues with the visibility of various folders, upload permissions and so on. We will try to resolve such issues as soon as we hear about them. For some issues we depend on the university support services and delays are possible.