Exercise 7 - Analysis and plotting

Work with a partner to complete the tasks below and submit your results via a pull request on GitHub by the beginning of tutorial next Friday.

To begin this week, one of the partners should fork the TA's Exercise 7 Github repo and provide collaborative access to the other partner. Clone the forked repo so that you have the required files. Be sure to commit regularly to show how you arrived at your solutions and demonstrate coding effort by both partners.

1. Write a script that creates two plots summarizing information about "Lecture11.fasta". In the first plot, present a histogram of sequence lengths, and in the second plot present a histogram of GC content of the sequences. There is code that Stuart provided last week after lecture in your cloned repo to help process these data if you need help with that, but I encourage you to try to translate the pseudocode we wrote in lecture last week into real code for this problem.

2. Find some data on two variables that you would expect to be related to each other. These data can come from your own research, your daily life, or the internet. Enter those data into a text file or Excel and then save a text file, and write a script that loads this text file and produces a scatter plot of those two variables that includes a trend line.

3. Given the data in "data.txt". Write a script that generates a two figures that sumamrize the data. First, show a barplot of the means of the four populations. Second, show a scatter plot of all of the observations. You may want to "jitter" the points (`geom_jitter()`) to make it easier to see all of the observations within a population in your scatter plot. Alternatively, you could also try setting the `alpha` argument in `geom_scatterplot()` to 0.1. Do the bar and scatter plots tell you different stories? Why?

Devise a plan for splitting up the work and generating the required code. Do this in parrallel, not sequentially. Don't forget to check and edit each other's code. Remember to frequently `add-commit` locally and `push-pull` to GitHub to avoid conflicts. Also, remember you don't have to be in the same place at the same time to work on this collaboratively thanks to GitHub!!!

Turning in your assignment via GitHub

Once you have committed all changes to your local Git repos and pushed all of those commits to the forked repo on GitHub, you can "turn in" your assignment using a `pull request`. This can be done from the GitHub repo website. When viewing the forked repo, select "Pull requests" in the upper middle of the screen, then click the green "New pull request" button in the upper right. You'll then see a screen with a history of commits for you and your collaborator, select the green "Create pull request button". In the text box next to your user icon near the top of the page, remove whatever text is there and add "owner's last name - collaborator's last name submission", but obviously substitute your last names. If I and Ann Raiho worked on the project together the text would read "jones-raiho submission". Then click the green "Create pull request" button. **Only one of you will need to create a pull request.**