R Assignment

Your R assignment will consist of three parts:

1. Replicating your UNIX assignment in R
2. Additional analysis and visualization
3. Reviewing two assignments from your peers

The final outcome of this project should be a well organized GitHub repository that contains a README.md file describing its general organization, a separate file in the **"R Markdown" format** that contains both the code and the description of the workflow, and an output file in either HTML or PDF format. The repository should also include the files you generated in part1. If you new to "R Markdown", check this website for more information!

You will be given email addresses of two randomly selected participants of the class. Please send them the url address of the GitHub (public) repository you've created **by 11:59pm on Friday, March 10**. In turn, you will receive links to two repositories to review. When you receive a link, first fork the repository, then clone the forked repository on your computer and write a review inside it named [your lastname]_review.Rmd.

Push your review to the forked repository and submit a Pull request **by 1pm on Monday, March 20**. Accept the pull requests of your reviewers. It's up to you if you make any changes recommended by the reviewers. If you do, create a new R Markdown document with implemented changes and name it accordingly.

Finally, submit your assignment in Canvas **by 1pm on Wednesday, March 22**.

Notices

- There will be significant time involved in completing this assignment, especially if you are new to R. Start early, look for additional resources, don't hesitate to ask for help. Google is your friend as are the other people in the class!
- Make sure that your code in the R Markdown document works. One should be able to replicate all your results by simply running it with the `Run all` command. Remember to remove all the objects you've created before running the code.
- It is your responsibility to send a link to your reviewers as well as to submit a review. It is not your responsibility to solicit either the links to other students' repositories or reviews of your project. If you haven't received the link on time, you don't have to review the project. If you sent a link, but haven't received the review, it's the reviewer's problem. The quality of your reviews will influence (increase) your grade.

Part I

Data Inspection

Load the two data files you used for your UNIX assignment in R as dataframes and inspect their context. Use appropriate functions to describe their structure and their dimensions (file size, number of columns, number of lines, ect...). You don't have to limit yourselves to the functions we learned in class.

As a reminder, the files are:

1. `fang_et_al_genotypes.txt`: a published SNP data set including maize, teosinte (i.e., wild maize), and Tripsacum (a close outgroup to the genus *Zea*) individuals
2. `snp_position.txt`: an additional data file that includes the SNP id (first column), chromosome location (third column), nucleotide location (fourth column) and other information for the SNPs genotyped in the `fang_et_al_genotypes.txt` file.

Data Processing

Manipulate the two files in R in order to format them for a downstream analysis. During this process, we will need to join these data sets so that we have both genotypes and positions in a series of input files. All our files will be formatted such that the first column is "SNP_ID", the second column is "Chromosome", the third column is "Position", and subsequent columns are genotype data from either maize or teosinte individuals.

For maize (Group = ZMMIL, ZMMLR, and ZMMMR in the third column of the `fang_et_al_genotypes.txt` file) we want 20 files in total:

- 10 files (1 for each chromosome) with SNPs ordered based on increasing position values and with missing data encoded by this symbol: ?
- 10 files (1 for each chromosome) with SNPs ordered based on decreasing position values and with missing data encoded by this symbol: -

For teosinte (Group = ZMPBA, ZMPIL, and ZMPJA in the third column of the `fang_et_al_genotypes.txt` file) we want 20 files in total:

- 10 files (1 for each chromosome) with SNPs ordered based on increasing position values and with missing data encoded by this symbol: ?
- 10 files (1 for each chromosome) with SNPs ordered based on decreasing position values and with missing data encoded by this symbol: -

A total of 40 files will therefore be produced.

A few notes and hints:

- In order to join these files, you may need to transpose your genotype data so that the columns become rows. You just have to know one letter to do this in R: `t()`.
  However, check the results carefully, as there could be surprises 😉
- As in the UNIX assignment, it might help to write out the entire workflow that will be necessary to produce the files described above before doing the actual analysis.
- Try to avoid loops in R. Especially nested loops. They usually take a lot of time. Try using `lapply` and `sapply` functions instead. We'll talk about them in class, but you can read this tutorial in advance.
- If you get stuck or confused, first, use the `help()` function; second, search the Internet; and, finally, post to the "scripting_help" channel on Slack.

Part II Visualization

In this part, you use `ggplot` to investigate (by visualization) the following questions:

SNPs per chromosome

What is the distribution of SNPs on and across chromosomes? Are there more SNP positions in maize or teosinte individuals?

Missing data and amount of heterozygosity

What is the proportion of homozygous and heterozygous sites as well as missing data in each sample and each group?
Hints: Create a new column to indicate whether a particular site is homozygous (has the same nucleotide on both chromosomes (i.e., A/A, C/C, G/G, T/T) or heterozygous (otherwise)). Normalize the height of individual bars using one of the `ggplot` "position adjustments" options.

Your own visualization

Visualize one other feature of the dataset. The choice is up to you!

Note, that it may be easier to reshape the original data (make it tidy) using the `pivot_longer()` function in the `tidyr` package within the `tidyverse` collection.