Alex Zhang

Part 1: Midterm Review

General Concepts:

1. What is TCGA and why is it important
    a. TCGA, short for the Cancer Genome Atlas, is collective genomic data from over 11,000 cancer patients. The aggregation of data facilitates research regarding cancer, from cancer biology to treatment to detection. TCGA is important because it has data from 33 different types of cancer, thus giving researchers opportunities to study cancer genomics.
2. What are some strengths and weaknesses of TCGA?
    a. TCGA data is publically available, which allows researchers to access and analyze the data for free. In addition, the data is very organized, consistent as well as well-documented, allowing for easy comparison and analysis of data.
    b. However, TCGA data only covers 33 types of cancer, leaving out many other types. In addition, TCGA data is also limited to only genomic data, leaving out other -omics. TCGA data also focuses largely on NA and EU populations, excluding others. Finally, TCGA data focuses mostly on somatic mutations, leaving out other aspects of cancer biology like germline mutations.

Coding Skills

1. What commands are used to save a file to your GitHub repository?
    a. In terminal: **git status** (to check differences between local and github repositories).

b. In terminal: **git add -A** (to prime all unadded changes in local repository for push).

c. In terminal: **git commit -m "[message]"** (to commit changes in local repository with message).

d. In terminal: **git push --all** (to push all committed changes to github repository).

2. What command(s) must be run in order to use a package in R?

    a. install.packages("[package name]")

    b. library([package name])

3. What command(s) must be run in order to use a Bioconductor package in R?

    a. install.packages("BiocManager")

    b. library(BiocManager)

    c. To install using Bioconductor:

        i. BiocManager::install("[package name]"), ex: BiocManager::install("TCGAbiolinks")

4. What is boolean indexing? What are some applications of it?

    a. Boolean indexing uses booleans to categorize specific values within a dataset. Boolean indexing can be used to find specific keywords or values, or to filter by these values. Boolean indexing can also be used to specifically select criteria in a data set. This allows for the selection of data variables and consequently selective analysis of data.

5. Draw a mock up (just a few rows and columns) of a sample dataframe. Show an example of the following and explain what each line of code does.

Example dataframe "dataframe":

	Age	Vital status	Age at diagnosis	Sex
Patient 1	56	Dead	50	Male
Patient 2	35	Alive	33	Female
Patient 3	47	Alive	44	Male
Patient 4	39	Dead	30	Female

a. a. an ifelse() statement

   i. **old_age_mask <- ifelse(dataframe$age > 50, TRUE, FALSE)**

   ii. In the example above, ifelse() is used to create an age mask for the variable dataframe$age. The syntax used for ifelse() is as follows: ifelse([test expression], [true result], [false result]). In this case, the test expression is **dataframe$age > 50** (is age older than 50). If true, this outputs a boolean value TRUE. If false, a FALSE value is outputted. This variable is named **old_age_mask**.

b. boolean indexing

   i. **dataframe$old_age <- subset(dataframe, old_age_mask)**

   ii. In the example above, a new column (**dataframe$old_age**) is being created using subset(). The syntax for subset() is as follows: subset([data], [subset_expression]. In this example, [data] is **dataframe**, and [subset_expression] is **ifelse(dataframe$age > 50, TRUE, FALSE)**, or the variable **old_age_mask** as defined previously.