Clinical Data Partner Activity

Last time, we showed you one example of how to analyze survival data and compare different phenotypic variables. This worksheet will be more of a self-guided activity!

Deliverables:

1. Answer the following questions as it relates to the clinical data frame. Submit your answers by adding the document to your GitHub.
2. Create an Rscript file called `week4_clinical_activity.R` that contains the R code that you write to answer the following questions.
3. Your code should also include informative comments throughout, explaining what *each* line of code does.
4. Though this is a partner activity, each student needs to commit both their answers and their code to GitHub. It is okay if your answers are the same as you are working together!
5. Submit your updated repository by the start of the next meeting, Wednesday February 9 (8 am).

A Few Steps Before Starting

1. Make sure you have copied over the `.gitignore` file from the class folder to your personal repository. Please come to Office Hours or ask if you need assistance with this!
2. Ensure that your data is in your `analysis_data` folder. If you do not have this folder, at the start of the clinical tutorial we recommended that you store all your data in an `analysis_data` folder. Please come to Office Hours or ask if you need assistance with this!
3. Read in `clinic.csv` file with the following: `clinic <- read.csv("PATH/TO/FILE_NAME.csv")`. Remember that the `<-` is the same as the assignment = in R. You can choose either! This is the file that you should have saved at the end of the clinical tutorial. Saving this file allows you to skip the `GDCquery` step every time you want to read in your data.

Written Activity

1. Define the following: *categorical variable, discrete variable, continuous variable.* Provide examples of each.

2. Look at the different column names of your `clinic` dataframe. Choose one that is interesting to you and your partner. Ensure that there are not too many NAs in this

column by using `is.na(clinic$COLUMN_NAME)`. Remember that in coding, True is equal to 1 and False is equal to 0. You can then use the `sum()` function to find how many True's exist. Which variable have you chosen?

3. Google your chosen variable. How is your variable measured or collected? Is your variable categorical, discrete, or continuous?

4. Find two research articles that mention your clinical variable. Provide the links and a brief description of the findings.

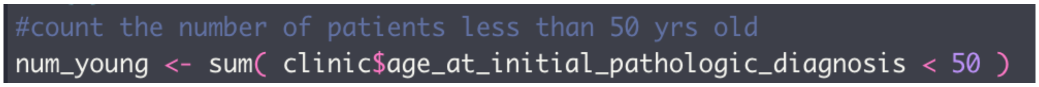
5. Choose a second variable. Which variable have you chosen? Provide a brief description of the variable and how it is determined or measured.

6. Scientists generate hypotheses before experimenting or exploring data. Generate three hypotheses: (1) Relate your variables to each other, (2) Relate your first variable to survival in colorectal cancer, (3) Relate your second variable to survival in colorectal cancer.

7. After you finish coding, summarize what you learned from your graphs!

Coding Activity:

While coding, add comments using the hashtag (#) describing what each line of code does. Example:



```
#count the number of patients less than 50 yrs old
num_young <- sum( clinic$age_at_initial_pathologic_diagnosis < 50 )
```

1. Perform an analysis looking at the two variables that you chose. First brainstorm and sketch out a plot that contains both variables. Feel free to get creative, if you are struggling, feel free to ask for ideas!
    a. TIP: Sometimes it can be hard to plot a continuous variable with another variable. You can convert a continuous variable to a categorical one. For example, we previously defined age < 50yrs old as “Young” and age >= 50yrs old as “Old.” Here we have converted age, a continuous variable, to young and old, a categorical variable.

2. Perform a survival analysis, following the steps of the clinical tutorial with the first variable.
    a. As with the previous TIP, the survival analysis needs a categorical variable. If you have a continuous variable, use an `ifelse()` statement to create a new column with a categorical version of the variable.

3. Repeat with the second variable.

Preparation for next week:
Just install the `maftools` package!

```
BiocManager::install(“maftools”)
```