**Methods:**

**Packages:**

import pandas as pd

import matplotlib.pyplot as plt

from time import time

from scipy.stats import (pearsonr, spearmanr, ttest\_rel, ttest\_ind)

from statsmodels.stats.multitest import multipletests

**Steps:**

are found in the jupyter note book called Galloul\_final on github to be completed by BADRA

You will find very descriptive comments will help u write the process of our code

**Results and Discussion:**

Setting our confidence level to be 99% (alpha = 0.01) meaning that we are 99% confident about the genes we found to be paired or independent or both.

Our analysis leads to the following:

First result: 

We found the number of paired DEGs before we apply the FDR correction method to be 11,855 genes, while the number of paired DEGs after we apply FDR correction is 11,389 which corresponds to almost 60.3% and 58% of all the genes (19,648), respectively.

This means that the FDR correction method helped us reject 457 genes that we thought paired DEGs which represent about 3.85% of paired DEGs before we apply FDR.

Second result:



We found the number of independent DEGs before we apply the FDR correction method to be 11,779 genes, while the number of independent DEGs after we apply FDR correction is 11,318 which corresponds to almost 59.95% and 57.6% of all the genes (19,648), respectively.

This means that the FDR correction method helped us reject 461 genes that we thought were independent DEGs which represent about 3.91% of independent DEGs before we apply FDR.

Third result:



We found that there are 11,180 happen to be paired and independent at the same time DEGs after applying the FDR correction method which corresponds to almost 56.9% of all the genes (19,648) where the distinct paired DEGs and distinct independent DEGs found to be 209 and 138 respectively which corresponds to 1.06% and 0.7% respectively all the genes (19,648).

How to improve our results?

By preprocessing our data to:

1. Drop rows (genes) that have zero value in the majority of its 50 GE columns

e.g.



1. Drop rows (fake genes) that have zero `Enterz\_Gene\_Id ` and/or any weird `Hugo\_Symbol`

e.g.



These kinds of issues in the datasets could be cleaned using two approaches:

1. Manually cleaning and droping the rows with the specified issues above.

Which would be a hard and long job especially with this kind of long data. (almost 20K rows)

2. Programmatically cleaning the data

This sounds like the nicer and gentle approach to take, unfortunately, this approach would cost us high Big O notation as we will iterate on the whole dataset gene by gene and then check the number of zeros in the 50 GE columns for each gene would produce Big O notation of O(n2) (nested for loop for example)

So in this case we are talking about execution time of hours to apply this nested drop the rows (genes) with the issues.