Name: Geerath Bhat

Sr. No: 21292

Data Analysis with ANOVA for Gene Expression

Introduction:

The code is an implementation of an Analysis of Variance (ANOVA) for gene expression data. ANOVA is a statistical technique used to analyse the differences among group means in a dataset. In this specific case, the code has been applied to gene expression data, to identify statistically significant differences among different categories or groups of genes.

Code Overview:

The code is organized as a Python class named ANOVAAnalysis. It takes a data file as input, reads the data using Pandas, and performs ANOVA analysis on it. Here's an overview of the key components of the code:

Data Loading:

The code starts by reading data from a CSV file specified by the data_path argument. It assumes that the data is tab-separated.

Data Preparation:

The code separates the input data into two parts: data_list and gene. data_list contains the actual data used for analysis, while gene contains additional gene-related information.

Design Matrices:

The code creates two design matrices, A and B. These matrices are essential for ANOVA analysis and help in categorizing and structuring the data.

ANOVA Calculation:

The calculate_F_values method calculates the F-statistic for each data point using the provided design matrices. It performs matrix operations to compute the numerator and denominator of the F-statistic.

Handling Numerical Issues:

I was encountering the “DeprecationWarning” and “RuntimeWarning” issue. To address these warnings, I refined my code to handle the division by zero and invalid value issues more effectively.

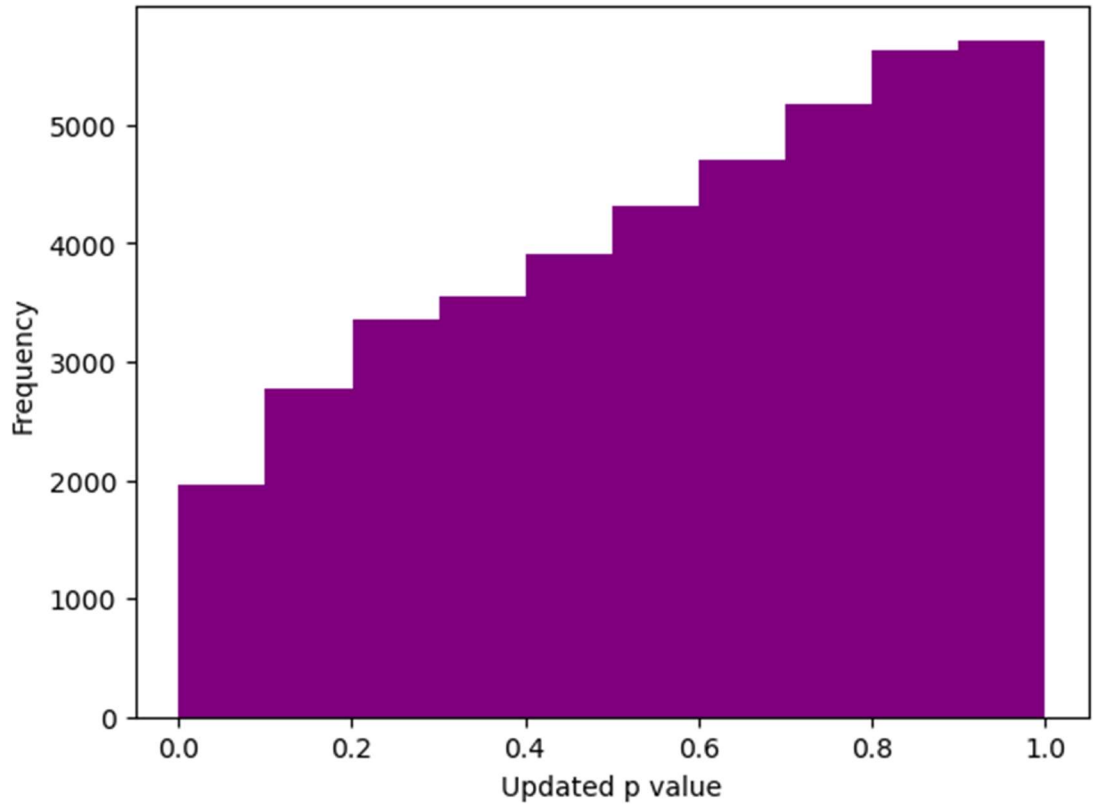
I added a check for np.abs(denominator[0][0]) < 1e-10 to identify cases where the denominator is very close to zero. If it's close to zero, it sets val to np.nan to avoid division by a nearly zero denominator.

For other cases, it performs the division as before.

The code includes error handling to address potential division by zero issues. It checks if the denominator is very close to zero and sets the F-value to NaN in such cases.

Plotting Histogram:

The plot_histogram method generates a histogram of updated p-values based on the calculated F-values. It uses the SciPy library for the F-distribution and Matplotlib for visualization.



Usage:

After initializing the ANOVAAnalysis class with the data, we can call the plot_histogram method to visualize the results.

Conclusion:

This code provides a comprehensive implementation of ANOVA analysis for gene expression data. It performs the necessary data preprocessing, calculates the F-statistic, and handles potential numerical issues. The generated histogram of updated p-values helps visualize the significance of differences among groups of genes. This code can be a valuable tool for researchers and analysts working with gene expression data to identify genes that exhibit statistically significant differences across various categories or conditions.