**General instruction:**

This challenge is about statistical analysis and/or data science. There is no standard solution, so it is important to “show your work” along the analysis, i.e., explain your logic, discuss on potential issues, and interpret the result. Therefore, a report with programming codes, figures, tables, together with comments is expected. There are many ways to make such report, e.g., by using “R Markdown” or “Python Jupyter Notebook”. Please submit a PDF file for the final report.

**Challenge #1:**

The file data1.txt is a tab-delimited text file with 4 columns: (1) cancer sample ID, (2) y, the response variable, (3) y\_w, unscaled/relative sample weight of y, and (4) x1, the explanatory variable. All variables are continuous, missing values are indicated by “NA”. The goal is to build an appropriate regression model for y using x1.

(a) Please start with a simple regression model using only y and x1, and demonstrate the relationship between y and x1 by using a figure as well as some statistic like the t-statistic and p-value (which indicate whether the true regression coefficient is not zero)

(b) Next, fit the model again but considering the sample weights, and compare to (a)

(c) A step further: try to fit a robust linear regression, and compare to (b)

(d) Please make comments/interpretations for the comparisons.

**Challenge #2:**

Here we will combine the data in data1.txt and data2.txt (unzip data2.txt.zip). The data2.txt is gene expression data and it is also a tab-delimited text file: its first column is the cancer sample ID which you can match with those IDs in data1.txt, and the rest columns are different genes. This analysis is using random forest regression model to look for predictive variables.

(a) Combine data1 and data2 by matching the sample IDs, and then you have all gene expressions as explanatory variables in addition to x1. Let’s use G = (g1, g2, g3, …) to denote those explanatory variables of gene expressions, and y is the same as above with sample weights y\_w.

(b) First, similarly as in challenge #1, perform robust linear regression modeling for y using the expression of each gene, but with x1 as a covariate. E.g., for the first gene g1, the model is E(y) = a + b\*x1 + c\*g1, and then repeat this for every gene to get the p-value of the regression coefficient of each gene. Rank the genes based on the p-values.

(c) Next, similarly as above, but fit this larger model instead: E(y) = a + b\*x1 + c\*g1 + d\*(x1\*g1), where the multiplication of x1\*g1 is the interaction term, and here “d” is the regression coefficient of interest. Like above, repeat this model for every gene to get the p-value of “d” of each gene. Rank the genes based on the p-values.

(d) Last, build a random forest regression model using x1 and G = (g1, g2, g3, …) to predict y. Sample weights y\_w should be considered. As computation can become intensive, parameter tuning, e.g., using cross-validation, is not required, and you can just set most parameters to their “default” or commonly used values. After the model is fit, use a “feature importance” measurement to rank the genes. Compare the top-ranked genes here with those top-ranked genes identified in (b) and (c), and please make comments/interpretations.