**STAT-823 Midterm Exam**

You are provided with a dataset called “**cell\_clinical\_data.csv**” that contains the data about the cells collected from various cancer patients. The clinical data of the patients and the information on relevant cancer cells is recorded in this table.

**Part – I: Quality processing**

Read the data into “**cell.data**” dataframe.

This dataframe contains several unwanted columns and missing data. This requires quality processing.

Perform the following quality processing steps to make a clean dataframe.

**1**. Check the column names that are available in this dataframe

**[1] "Study.ID" "Patient.ID" "Sample.ID" "Age"**

**[5] "Annotation.Source" "Cancer.Type" "Cancer.Type.Detailed" "Cell.Line.Source"**

**[9] "DepMap.ID" "Ethnicity.Category" "Fraction.Genome.Altered" "Growth.Medium"**

**[13] "Mutation.Count" "Mutation.Rate" "Oncotree.Code" "Ploidy"**

**[17] "Primary.Tumor.Site" "Race.Category" "Sample.Type" "Sex"**

**[21] "TMB..nonsynonymous." "Tumor.Type"**

**2**. "Patient.ID" and "Sample.ID" contains several duplicates. How do you check existence of duplicates in a column? write the code and save them in the separate vectors. Don’t print them as there are several duplicates in both of them, it will fill your report.

Patient.ID\_duplicates <- *code to identify duplicates*

Sample.ID\_duplicates <- *code to identify duplicates*

**3**. "Patient.ID" and "Sample.ID" have several duplicates and cannot be used as unique record identifiers. To deal with this issue, create one column “Pt.ID” that contains the values “Pt\_n”. n should be the numbers stating from 1 to length of datraframe.

The Pt.IDs should like this

"Pt\_1" "Pt\_2" "Pt\_3" "Pt\_4" "Pt\_5" and so on

**4**. “Sample.ID” values contains multiple parts separated by underscore. Add the first part of the “Sample.ID” to the Pt.ID separated by underscore. The resultant Pt.IDs should look like this

[1] "Pt\_1\_127399" "Pt\_2\_1321N1" "Pt\_3\_1321N1" "Pt\_4\_143B"

[5] "Pt\_5\_143B" "Pt\_6\_201T" "Pt\_7\_22RV1" "Pt\_8\_22RV1"

**5**. relocate the column Pt.ID to the first position

**6**. Remove the following unwanted columns from the **cell.data** dataframe

**"Patient.ID", "Sample.ID", "Annotation.Source", "Cell.Line.Source", "Cancer.Type.Detailed", "DepMap.ID", Race.Category", "Sample.Type", "TMB..nonsynonymous.", "Tumor.Type"**

**7**. The "Age" column Na values. Replace these NAs with random values of age ranging from 1 to 100.

**8**. Replace the column name "Ethnicity.Category" to "Ethnicity"

**9**. Replace the NA values in the "Ethnicity" column with random selections from the ethnicities "White", "Hispanic", and "Black"

**10**. Replace the NA values in the "Fraction.Genome.Altered" column with random values within the range of minimum value to maximum value of that column using the runif() function in R.

**11**. Replace the values in "Growth.Medium" with the random distribution of these types of media DMEM, RPMI-1640, MEM, EMEM, FBS.

**12**. Replace the NA values in the "Mutation.count" column with random values ranging from 1 to 20000

**13**. There are several NA values in the Mutation.Rate. Fill those NA values using the formula (Mutation.Count/(Fraction.Genome.Altered\*3000000)

**14**. There are some formula-like values in the Ploidy, Ex: *3n+/-, Near-triploid 69+/- (58-80)*. remove those and make them NA. You may use conditions such as if the number of character are more than 4, assign NA.

**15**. Now find the min and max values of the ploidy. Fill the NA values with the random numbers ranging from min to max values of Ploidy. Round the random values to two digits after the decimal point

**16**. Primary.Tumor.Site contain values with multiple words separated by underscores. keep only the first word and remove the words after the first underscore including the underscore.

**17**. Replace the value "Large" with "Large Intestine" in the "Primary.Tumor.Site" column using the **ifelse()** function.

In the same way Replace; Biliary to Biliary Tract, Central to Central Nervous System, Haematopoietic And Lymphoid Tissue to Haematopoietic, Salivary to Salivary Gland, Small to Small Intestine, Soft to Soft Tissue, Upper to Upper Aerodigestive Tract and Urinary to Urinary Tract

**18**. There are Na values in the Primary.Tumor.Site, replace them with the first word of Cancer.Type column

**19**. There are NA values in the Sex column. replace them with random values of "Male" and Female". As a next step, when the Cancer.Type is "Breast Cancer" it should be Female and if Cancer.Type is "Prostate Cancer" it should be Male.

**20**. Now check if NAs are available in cell.data dataframe. At this step the dataframe should be clean and with no Nas. If you still find any Nas or missing data, go back and work on it.

**Part – II: Analysis and interpretation**

**1**. Perform some general functions on this dataframe that will familiarize you on this dataframe.

**2**. Generate a contingency table to show how many studies are involved in this dataframe. How many patient records are available in each study? Use Study.ID to pull this data.

**3**. what is the best plot you prefer to use to know the Age distribution Among these patients, and how you do that?

**4**. what is the most abundant Cancer.Type in this dataset? generate a bar graph that shows how many patients are existing in each cancer type.

**5**. What are the different types of Growth.Medium that is used for Bladder cancer?

**6**. define the relationship among Mutation.Count, Mutation.Rate using a scatter plot. ignore the values less than one to generate the plot

**7**. how the Ploidy is distributed among Males and Females. Generate two histograms in the same Figure.

**8**. Create a list object called cell.data.obj with the variables Cancer.Type, Growth.Medium, Primary.Tumor.Site and Oncotree.Code

**9**. Age Group Analysis: Categorize patients into different age groups (e.g., child (<=20), adult (<=60), senior (>60)) based on their ages. Calculate the average mutation count for each age group and identify if there are significant differences in mutation counts among age groups. use for loop and if else statements

**10**. Generate a contingency table of Study.ID by Ethnicity. Use this contingency table to generate a barplot that will show you the distribution of ethnicity among the three studies. color the bars as per the Study.ID. What are the ethnicities that are involved in all the studies.