BIO247 Exam 1

Nagendra Dhanikonda

2022-09-29

# Part 1 - Written

## Problem 1

Robust: methodology, analysis, conclusions, are formulated such that the findings can be widely applicable in similiar but different experiments. Reproducible: same methodolgy can be followed to produce the same results. ALl methodologies + resources should be open to the public so that anyone can follow the same steps.

## Problem 2

This study is not robust because the authors seem to generalize the findings even though the sample was only taken from one university. There is also no information on whether random sampling was used or not which further takes away from the generalizablity of the study. The article may be reproducible if the authors share their questionnaire and the predictors of tolerance in their methods.

## Problem 3

This study is robust because the authors used pre established, open source technology to anazyle the data from open source data bases. It is also reproducible because the authors clearly state what technology was used to do what analysis and where the data was obtained from (open sourece database).

## Problem 4

## Problem 5

N/A # Part 2 - Practical

## Problem 6

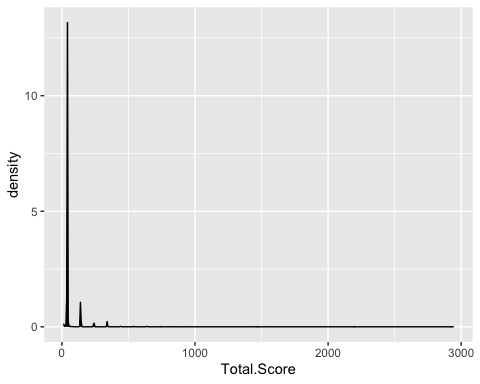
Vector of interactions was stored under the vector interact. A vector of positive and negative was then made using the repeat function(named addition). Spaces were removed in the addition vector using the gsub function and then the string split function was used to seperate the string into substrings at each comma. Unlist function was then used to revectorize the string. The paste function was then used to combined each value in the interactions vector with a positive or negative label from the addition vector.

interact <- rep("interactions",25)  
addition<- rep("suffix\_1, suffix\_0, suffix\_1, suffix\_0, suffix\_1, suffix\_0, suffix\_1, suffix\_0, suffix\_1, suffix\_0, suffix\_1, suffix\_0, suffix\_1, suffix\_0, suffix\_1, suffix\_1, suffix\_1, suffix\_0, suffix\_1, suffix\_0, suffix\_1, suffix\_0, suffix\_1, suffix\_0, suffix\_1,", 1)  
addition2 <- gsub(" ", "", addition)  
addition3 <- unlist(strsplit(addition2, ","))  
LabeledVector <- paste(interact,addition3, sep=" ")

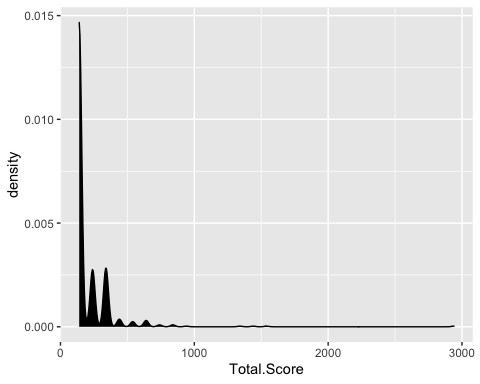
## Problem 7

Density plot code was sourced from RMD created for in class gg plot activity. A new data frame named Short\_GBM\_data was created where all rows that had a Total score of less than 140 were deleted. Total score of 140 was set as the threshold value because the majority of the interactions above 140 had a match score of 100 while scores below 140 tended to have much lower match scores. This brings into question the validity of the interactions and whether the output can really be trusted. There is also a large drop in total score between 140 to the next most abundant score (50 and lower) showing a big gap in the density between those values save a few total scores in between (not observable on the initial density plot)

#density plot pt1  
library(ggplot2)  
GBM\_data<- read.csv("//Users//nagendra//Desktop//Bioinformatics//Test\_submissions//GBM\_data.csv")  
ggplot(GBM\_data) + geom\_density(aes(x=Total.Score), fill="black")



#Threshold density plot  
Short\_GBM\_data <- GBM\_data[1:707,]  
ggplot(Short\_GBM\_data) + geom\_density(aes(x=Total.Score), fill="black")



## Problem 8

Solution was based on in-class activity done in the intro to R lecture series File was uploaded using the read.csv function. The Junk characters such as [, ], ’, and spaces were removed with the gsub function. stringsplit was then used to create a new list (Split\_ID) with each character as a single Paper ID and the unlist function was used to revectorize the list that was created. Unique function was then used to find unique paper ID’s and the length function was used to count the number of outputs from the unique function.

GBM\_data<- read.csv("//Users//nagendra//Desktop//Bioinformatics//Test\_submissions//GBM\_data.csv")  
  
List1 <- gsub("\\]", "", GBM\_data$Paper.ID)  
List2 <- gsub("\\[", "", List1)  
List3 <- gsub(" ", "", List2)  
List4 <- gsub("\\'", "", List3)  
  
Split\_ID <- unlist(strsplit(List4, ","))  
length(unique(Split\_ID))

## [1] 218