Math 5301 Homework 1

The file `FTIC.csv` contains data for 9,218 first time in college freshman, admitted to TSU between 2004 and 2011. The goal of this lab is to explore the effect of increased admissions standards on enrollment and retention, while learning some basic R commands.

1. Use the "Import Dataset" tool (upper right window of R studio) to import `FTIC.csv`. Make sure to choose "Yes" for the "Heading" option.

2. The command `ls()` will list all objects in the R workspace, which will just be a dataframe called `FTIC`. Now, `dim(FTIC)` will show the number of rows and columns in this dataframe, and `head(FTIC)` will display the first few rows.

3. The column `X2nd_Fall` shows whether each student was retained until their 2nd fall at TSU. Use the command `retention=FTIC$X2nd_Fall` to store that column of data in a vector called `retention`. Now, try these commands: `ls()`, `length(retention)`, `head(retention)`, and `table(retention)`.

4. The columns `PERCENTILE` and `SAT` contain the students' percentile ranks and SAT scores. Store these columns in vectors called `rank` and `sat`, respectively, and see what happens when you apply the functions `hist()`, `summary()`, and `mean()` to these vectors.

5. It will be conventient to replace the `Y` and `N` values of `retention` with 1's and 0's. Try this:

   (a) `table(retention)`

   (b) `temp=(retention=="Y")`

   (c) `table(temp)`

   (d) `temp=(retention=="Y")*1`

   (e) `table(temp)`

   (f) Now, `retention=temp` will replace `retention` with `temp`. Of course, this entire process can be achieved with `retention=(retention=="Y")*1`.

   (g) Now that `retention` has values of 0 and 1, `sum(retention)` tells us the total number of students retained, and `mean(retention)` gives the retention rate. Yes, only 66.8% of FTIC freshman were retained until their second fall. Finding ways to increase retention is one of the most important problems faced by universities.

6. All of our variables are set up, and it's easy to do some basic statistics, as follows.

   (a) `plot(rank,sat)`

   (b) `model=lm(sat~rank)` (Creates a linear regression model for predicting `sat` using `rank`)

   (c) `summary(model)`

   (d) `plot(retention,rank)` Unfortunately, R is now treating `retention` as a quantitative variable, since it takes the values 0 and 1. We can force R to treat it as a categorical variable, that is, a *factor*, by using the `as.factor()` command.

   (e) `plot(as.factor(retention),rank)`

7. If we can figure out how to predict retention accurately, we can admit those students with the highest chance of being retained, and increase our retention rate.

   (a) `rankmodel=glm(retention~rank,family='binomial')` Creates a logistic regression model for predicting `retention` using `rank`. Because `retention` is dichotomous, a linear regression model is not appropriate, and a logistic regression model is a better choice.

   (b) `summary(rankmodel)` The p-value for `rank` in this model is less than 2×10^{-16}, so `rank` is a highly statistically significant predictor of `retention`.

   (c) `satmodel=glm(retention~sat,family='binomial')`

   (d) `summary(satmodel)` The p-value for `sat` is 1.55×10^{-8}, so `sat` is also highly statistically significant.

   (e) `model=glm(retention~rank+sat,family='binomial')` Creates a logistic regression model for predicting `retention` using `rank` and `sat`.

   (f) `summary(model)` The variable `rank` is still highly statistically significant, but `sat` no longer is, with a p-value of 0.287. This shows that `sat` was only a strong predictor of `retention`, because it is correlated with `rank`. Once `rank` is included in the model, there is no statistically significant predictive power gained by adding `sat`. This could suggest that `sat` shouldn't be used in admissions decisions if `rank` data is available.

8. The table below shows TSU's admissions requirements from fall 2012. Note that percentile ranks of 90 to 100 are not listed, because public universities are required by state law to admit anyone in the top 10% of their class. Also, the minimum score possible on the SAT is 400, so all students with ranks from 50 to 89 are admitted under this policy.

Percentile Rank	1–24	25–49	50 – 89
SAT Requirement	1030	950	400

   There are actually 547 students in our data set who were admitted, even though they don't satisfy these requirements. How can we see this? One aspect of R and Matlab that make them such powerful languages is their matrix and vector indexing.

```
index=((rank<25)&(sat>=1030))|
      ((rank>=25)&(rank<50)&(sat>=950))|
      ((rank>=50)&(rank<90)&(sat>=400))|
      (rank>=90)

                #index is a boolean vector, which is TRUE for students
                #who satisfy the admissions requirements, and FALSE for
                #those who don't.

index           #Just lists all the TRUE/FALSE values stored in index.
table(index)    #Shows us that 8671 students satisfied the requirements,
                #and 547 didn't.
```

```
retention[index]    #The retention vector, but only the components where
                    #index is true.  That is, this vector lists the
                    #retention for all students who satisfied the admissions
                    #requirements.

mean(retention[index])  #Retention rate for students satisfying the
                        #requirements.  It's a bit higher (67.4%).
```

These calculations show us that enforcing the admissions standards would have resulted in an *enrollment loss* of 547 and a *new retention rate* of 0.674. From the university's perspective, enrollment loss is bad, but higher retention is good. It would be nice if we could consider many different admissions policies, and try to find one that is optimal. This is easy in R, because we can write our own functions.

9. Example of an R function:

```
mysumanddiff=function(x,y){
                L=list(sum=x+y,diff=x-y)
                return(L)}

                #Creates the function.

mylist=mysumanddiff(8,3)  #Applies function to 8 and 3 and stores in mylist.
mylist$sum                #Returns the sum from mylist.
mylist$diff               #Returns the difference from mylist.
```

10. In problem 8, we effectively have a vector of *rank thresholds* $(25, 50)$ and a vector of *SAT thresholds* $(1030, 950, 400)$. Write a function called `evalthresholds` that accepts vectors `rank`, `sat`, `retention`, `rankthresholds`, and `satthresholds` and returns a list with `enrollmentloss` and `newretention` components. Then, the following code should produce the same results from problem 8.

```
rankthresholds=c(25,50)  #The c() command is one way to make vectors in R.
satthresholds=c(1030,950,400)
mylist=evalthresholds(rank,sat,retention,rankthresholds,satthresholds)
mylist$enrollmentloss
mylist$newretention
```

11. In fall 2012, the Office of Enrollment Management recommended no longer admitting students with percentile ranks below 33, which is equivalent to the following requirements.

Percentile Rank	1–32	33 – 49	50 – 89
SAT Requirement	1610	950	400

Find the enrollment loss and new retention for this policy.

12. Challenge: Can you find a policy with a higher new retention than the one given in problem 11 without having a higher enrollment loss? This is another powerful feature of R. Now that you have the `evalthresholds` function, you can use a brute force search to find the admissions policy that maximizes new retention, among all policies that don't increase enrollment loss.