Non-FA Track Project 3 Process

1) Import data into a data frame, call it df (or whatever you want…)
2) Create the data n x (m+1) data matrix using model.matrix(y~.,data=df)
3) Create a list with just the y values from the data frame
4) For a specific value of k, and a specific fold we now need to formulate the optimization problem
    a. For this fold's training data let X be the appropriate rows of the matrix output from step 2
    b. Calculate $X^T X$
    c. Calculate $-2 y^T X$
    d. Set Q to be a (2m+1) x (2m+1) matrix of all zeros
    e. Assign the first m+1 rows and columns of Q to be the output from step b
    f. Set obj to be a list of (2m+1) zeros
    g. Set the first m+1 values of obj to be the output from step c
    h. Create a constraint matrix, direction, and rhs to satisfy the 2m constraints for the big-M (one for the left inequality and one for the right inequality, for each value of j) and the constraint that the sum of the z's is less than or equal to k
    i. Create a variable-type array that assigns the betas to be continuous, and the zs to be binary
    j. Assign the lower bound of the betas to be -M
    k. Plug this all into a gurobi list and solve the problem
    l. If it were me writing this code, I would put this inside a function that takes X,y,k as input so that I could just call the function every time I need to solve the MIQP. In total, you'll need to solve the MIQP 101 times: 100 for the cross-validation, and once on the entire training dataset with the correct k.
5) HINT: You can check if your code is working correctly by comparing your betas to the betas found using R's lm where you restrict the columns to just the ones where $z_j = 1$