1. Preprocessing
   1. Reads the data
   2. Formats the data correctly
   3. Removes columns with sparsely populated categories
      1. Sets threshold
      2. Returns the data with the columns that are sparse removed –DON’T DO THIS
         1. Recurses through each column in the data frame – this should be a simple nested function –use sapply
            1. Finds the most common factor integer in the column

Generates a list of factor levels present tied with their frequency

Finds the integers associated with every factor – as.numeric

Finds the number of each integer integer (tabulate)

Finds the most common of those number (which.max)

Finds the factor level associated with that integer (levels)

* + - * 1. Levels(f)[which.max(tabulate(as.numeric(f))))]
      1. Sapply(df, funcation(x) {levels(x)[which.max(tabulate(as.numeric(x)))]}
      2. Finds the proportion of factor variable integer values that differ from the most common factor variable value
         1. Creates a Boolean vector that indicates each factor f variable integer value is the same as the most common factor variable integer value x
         2. x == as.numeric(f)
         3. Finds the mean of the numeric version of the Boolean vector and compares it the sparse.thresholds using mean and as.numeric
         4. Mean(as.numeric(x == as.numeric(f)) < sparse.threshold
      3. Mean(as.numeric(Levels(f)[which.max(tabulate(as.numeric(f))))] == as.numeric(f)) < sparse.threshold
      4. Mean(as.numeric(Levels(f)[which.max(tabulate(as.numeric(f))))] == as.numeric(f)) < sparse.threshold
      5. As.numeric(levels(f)[x] === x
      6. Mean(which.max(tabulate(as.numeric(f)) == as.numeric(f)) < sparse.threshold
      7. Sapply(df, function(x) { Mean(which.max(tabulate(as.numeric(f)) == as.numeric(f)) < sparse.threshold}
      8. Df[Sapply(df, function(x) { Mean(which.max(tabulate(as.numeric(f)) == as.numeric(f)) < sparse.threshold}]
  1. Creates model matrix

1. Problem 1
2. Runs GLM on the model matrix