**Summary**: Create “regressr” an R library dedicated to helping user build regression models, providing text output to ease interpretation of the model’s results, and optimizing model specifications. The purpose of this library is to make the process of running regression modeling and coefficient interpretation easier for those unfamiliar with regressions.

Descriptions of Functions

Model Builder
Function 1 Piotr:

Users input a dataset. The function returns with a table of variable description and summary statistics, with instructions on choosing a dependent variable and a set of independent variable(s). The function also returns instruction on choosing dependent and independent variables for regression, using accessible language for those unfamiliar with regressions.

1. Create a df of Variable Names and Types
    a. Describe data types to user
    b. Instruct User to Pick Dependent and Independent Var
        i. Return comment to user:
            1. Recommend use function 2
            2. Explain what a dependent variable is
2. Explain the Variable Types within a given df
3. Warn about missing values.
    a. Warn if more than 10% /NA per variable, and recommend options

Function 2 Justin Goss:

Users input a dataset and dependent variable. The function checks dependent variable for class and recommends an initial regression technique. For instance, if the dependent variable is binary, the function recommends a logit or probit regression as opposed to recommending OLS for a continuous dependent variable.

1. User Inputs (dependentVar, independentVar)
2. Outputs what type of models you can run
3. Explain the different recommended models
    a. OLS - continuous dependent
    b. Logit/Probit - logical/factor dependent

    c. Multinomial Logit/Probit -
    d. Ordered Logit/Probit -
    e. Tobit -
    f. Fixed Effect (optional) -
4. Recommend use function 3. Explain what dependent and independent variables are

Model Interpretation
Function 3:

Users input a data set and a formula of regression. The function runs the regression and outputs the coefficients of predictors and the model's diagnostics, along with text interpretation for each parameters and explanation of the model's various diagnostics. The interpretation will be based on model specifications -- log-log, lin-log, quadratics, VARs, etc. For instance, the function could describe what a coefficient means in terms of the user's independent and dependent variables, for example: "a one unit change in x correlates with a B unit change in y." Other text outputs could explain the meaning of a p-value and statistical significance, the adjusted R-squared term or the cumulative significance of the model.

1. User Inputs (modelType, dependentVar, independentVar (vector) , df, options[logDepen = T/F, logIndepen = c(independentVar), squareIndepend = c(independentVar)])
2. If statement (modelType) of regressions
3. Run designated model type based on given inputs
4. Interpret model output.
5. Interpret each variable.
    a. Example: "a one unit change in x correlates with a B unit change in y."
    b. Statistical significance
6. Output Statistical Info on the Regression (R^2, F-Stat etc).

Model Optimization
Function 4:

Users input a dependent variable and a set of potential independent variables, and specify a model diagnostic parameter. Function checks combinations of independent variables looking for those with the best model fit, such as the lowest error rate or highest adjusted R-squared,

and outputs the  top N number of model (user specified or default). The function also allows users to input options including what variables should be included in all specifications, whether to include quadratic terms, etc.

1. User has to create new var if they want squared/logged variables included in the model optimization process.
2. User Inputs (modelType, dependentVar, independentVar (vector) , df, [option(includeIndepVar = c(from step1) )])
3. Uses MAPE as the diagnostic parameter.
4. For-loop with every possible variable transformation
5. Output df with ranked model specifications

Optional Considerations for everything...Yixuan:

Allow user more front-end choices when using function 4
Consider using shiny to make this into a dashboard