**Project Proposal:**

*The Problem:*

Financial analysts today need sophisticated statistical tools in order to analyze trends in the data. We need a statistical package that financial analysts unversed in statistics can use. This means it must be able to do regressions for predictions, statistical inference for drawing conclusions from data, and basic data descriptions in order to describe or visualize the data. This also means much of the work involved in choosing the right test or regression must be automated.

*A Description of the Solution:*

My program is a statistical package that caters towards people who do not know statistics. All they need is the data they are working with and what they want to do from the data. This means that Ithe programs needs ask the user a series of questions to understand the nature of the problem they are looking at before running the appropriate statistical techniques to answer the said question. For conducting regression, I intend to use sklearn’s linear\_model.LinearRegression class in order to make predictions with the data. For statistical inference, I plan to use the various functions in Pandas to conduct statistical tests on the data. For data visualization and basic statistical summaries, I plan to use the plot() function in Pandas to create data visualizations, and basic functions in Pandas to create statistical summaries.

*Breaking Down the Problem:*

1. Finding and creating the appropriate functions or algorithms that will conduct the tasks expected
   1. Regression
      1. Leveraging sklearn to conduct multiple types of regressions
      2. Doing model checking so user doesn’t have to check for validity
      3. Separate training and test data to fine tune models
      4. Making models easily usable
         1. User can extrapolate or interpolate from model using inputs
   2. Inference
      1. Leveraging pandas to do statistical tests
      2. Model checking so user doesn’t have to check if model is appropriate
      3. Outputting a formulaic “do not reject/reject null” response based on user input
   3. Visualizations and summary
      1. Leveraging matplotlib to create visualizations
         1. User can choose what kind of visualization to create
      2. Leverage pandas to find basic traits of the data
         1. Average, quartiles, standard deviation, sample size
2. Developing the logic so that user with no knowledge can easily pick appropriate task
   1. Figure out what the user wants to do
      1. Determine what kind of regression user wants to do
         1. Examine number of variables in dataset
         2. Ask about nature of the data
      2. Determine what kind of test user wants to do
         1. Ask about what user wants to learn
      3. Determine what kind of visualization user wants to create
         1. Create visualizations automatically on data, and user chooses
3. Developing a GUI
   1. Create simple interface that is clear
   2. Provide a lot of tips and information so users can be better informed
   3. Easily interaction

*The Trickiest Problem:*

* Using sklearn to do time series analysis on prices to predict where prices will go (e.g. stock prices)

*The Algorithmic Solution:*

1. User indicates that they want to predict things
2. Indicates that time is a variable
3. Loading stock data from Google Finance using a webscraper built into Pandas, or load Excel files with historical stock data
4. Put stock data into a CSV file
5. Convert CSV file into a data frame
6. Create a lag plot for autocorrelation to examine if model is appropriate
7. Train an autoregression model on data, either automatically or user chooses how far to go back
8. Test model against test data from same data set
9. Return model with lowest mean square errors, with data points at specified intervals
10. User can use model now to predict prices in the future

*Modules Used:*

Sklearn

Pandas

*Update 1:*

Given the feedback received, my project will focus on being a stock recommender. This means that a user can create a porfolio and receive recommendations on stocks to choose. This means a recommendation algorithm must be implemented to find similar stocks to the user and time series analysis to find optimal times to sell.

Algorithm:

If new user:

Ask assets willling to put in

More assets == more risky stocks recommended

Ask how frequently trading

More frequent == less risky stocks recommended

Ask how risk averse

More risk averse == less risky stocks recommended

Create new profile

List new stocks of the day webscraped from finviz

List stocks recommended by algo

Recommending algo:

Match profile to other users with similar profiles

Examine successful trades with other users made

Recommend stocks in such trades

Examine stocks in user portfolio:

Predict future prices of stocks

Find best time to sell

Update 2:

The focus of the project has shifted from recommendation to visualization. The GUI has been modified so that there is only one entry field for the user to enter a stock ticker name. The meat of the program is the stock visualizer in the form of a graph that the user can interact with using the keyboard and mouse to gain different perspectives and knowledge.