Jesse Bohannon

Isys 520

Doctor Gove Allen

Tuesday, April 15, 13

Final Project Write Up

Using Excel as a Front End to R and SQL

Task

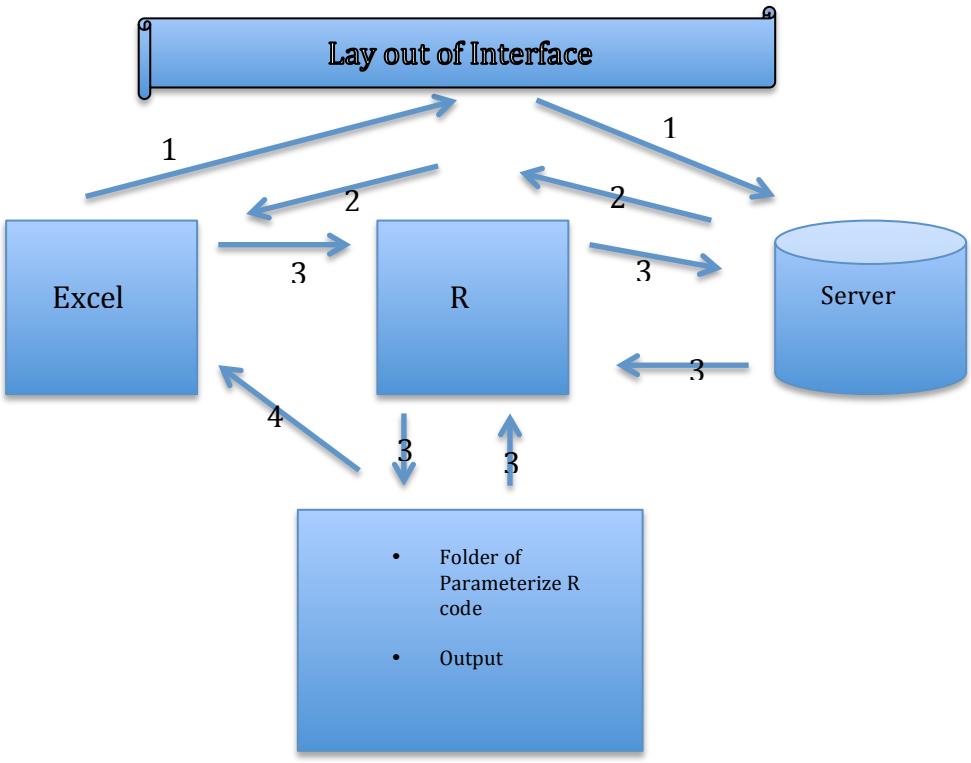
I was talking with a gentleman at an unnamed investment firm in Salt Lake City during the semester when I came across my final project. I approached him looking for internship opportunities, however the firm did not have a history of taking interns because they typically did not add value. When I mentioned I was taking a class on Visual Basic he became excited and then shared his problem with me.

The firm had numerous quantitative investment strategies with few quantitative analysts. He being one of them was responsible for any data mining or statistical analysis that took place in the program R. The firm relies heavily on this information. This causes a “bottle neck” of research, he told me. What they needed was for the non-quantitative employees to be able to do their own in depth research without typing a single line of code. “I would like them to do something like a Generalized AutoRegressive Conditional Heteroskedastic model through excel”, he said. This kind of analysis obviously could not be done in excel, but through the open source statistical software R, it could. The task would then be:

1. Use excel only as a front end interface with R
2. Use R as a backend to access the server information and perform analysis
3. To do it all remotely and robustly from a simple userform

Solution

The whole program needs to be scalable. The firm will want to add their own models, many of them proprietary, and integrate it into the overall investment strategy of the firm. The development of the application will be ongoing as strategies change depending on investment climates. There are an unlimited amount of statistical models and test to be ran, this application provides a template that could be enlarged to meet the needs of this investment firm. The program runs like so:



1. Queries the data base to populate the userform to let the user know which tables are available for analysis
2. Retrieves table names, column headings, data type or any other information wanted.
3. After the type of statistical test, table and variables are selected excel then, connects to R remotely. The parameterized R script is rewritten with the variables that where selected in the userform. This now rewritten text file is passed to R, the analysis is done and sent to another text file. The R addin package "RODBC" is installed to allow data base interaction and SQL statement compatibility. A Data Source Name must be configured in administrative tools on your control panel, then referenced in the parameterized R code. The parameterized code must coincide the structure of the data being retrieved from the server (for instance commands to a matrix requires different operations).
4. Excel then retrieves the parsed R output using the agent and displays it on a worksheet (this is scalable to retrieve graphical out).

Over the summer I will be expanding this project for the investment firm. I am presenting the program to the head quantitative analyst at the end of the semester and I am sure they will be pleased to have the beginning of a solution for the vexing problem.

Difficult Concepts

I logged around 65 hours of headphones in, no talking, billable hours on this project. I realized now that I have gotten very good at debugging. Really utilizing the

immediate and locals window, lots of stepping through with multiple breakpoints. Along with refreshing my VBA, I also learned more R and SQL code.

Remote commands to R required Shell commands. Querying the SQL server for table and column names required SQL and knowledge of the database structure. I was able to get assistance from Professor Allen for both these concepts. He generously gave his time and expertise and pay credit to him.

Another method I became familiar with is application.wait. When the program parses through the text file and sends it to a variable, the agent doesn't allow enough time for thousands of characters to be written and assigned. I only found this by realizing my code worked when I stepped through it but not when I ran it all. I then had to put different lags in my program to isolate where it needed more time to process. I created a do until loop to allow it more time.

I also gained an overall view of how computers interact with servers. I became good at configuring new ODBC DSNs. General things, like which Internet connections can route what port traffic in a database. The user form that I created was a large part of the program. It consists of four dependent combo boxes, four dependent labels, sorting checkboxes and command buttons. Populating a combobox from arrays and ranges is something I became very familiar with, as well as handling events in the userfrom like mouseup and mousedown. This became incredibly useful when many things had to happen in a short amount of time, like populating the subsequent comboboxes before the user selected them. The labels on the userform also change depending on which table is selected; this took some time to configure a loop with the .RowSource feature on the userform.

Conclusion

While this project has potentially a huge impact for the firm. I believe I have hurdled all of the main problems they might have run into and I think that the answer is rather elegant. The program allows other employees write their own parameterized r script with specific tags then just reference it in the userform. The program will eventually be a stand-alone application allowing the expansion to only be different R scripts being added to a folder. This final has been by far the most educational project this semester and I will definitely look forward to further developing the program.