FAQ

**workflow**: NHANES DATASET ---> Table extraction--> Column selection --> row and column mutation--->table

When you have a dataset like NHANES and you are trying to customize 10-20 tables from that data, you can bet that it will be time intensive work.

Testing:

Tests on our scripts for automation (what happens if we cant launch our app?)

We will run tests on our functions in R (ex. null data sets, expected tables )

We will run tests on our connections to the database

We will run tests on our GUI (ex..range values, no selections, table handling)

Feasibility Report For Senior Design Project

December 15th, 2017

CS 4900

Antonio Ruiz

Apolinar Ortega

Jacob Darling

ABSTRACT

    This paper will talk about various aspects of the project and will summarize why the project is feasible to take on. The paper will start off with background information to introduce the problem this project is meant to solve. From there, we will cover the stories and that we can complete them within the given time frame. We will talk about . We will talk about how we have access to the resources needed to complete this project. Then we will indicate how our client is able to maintain the project after we complete it. That will be followed by a section summarizing any security concerns we may have. The next section will cover how we will do our testing. Finally, we will talk about why we won’t have legal issues to worry about for the project.

BACKGROUND

    Here is the background section of the project. This section of the paper will introduce our client, Dr. McKean, give a quick overview as to what the project is about, and talk about why the need for this project came about.

Client:

Dr. McKean is a Professor of Statistics here at Western Michigan University.  He has published several papers on robust features in linear models. These papers include theory for estimation and testing, geometries of inference He is currently an Associate Editor for the *Journal of the American Statistical Association* and the *Journal of Statistical Computation and Simulation*. In 1999, he was elected as a fellow of the American Statistical Association.

Dr. McKean will also be the sole maintainer for the project when we finish development. He has extensive experience using the R programming language along with Sass scripts. He has some experience with using the bash shell. However, he does not have experience using SQL. Aside from this one issue, Dr. McKean is completely proficient in the technologies needed to maintain the project.

Problem:

Dr.McKean’s daughter did her doctoral dissertation with the NHANES dataset. They quickly had issues that came with cleaning the data and mutating it so that they could perform analysis. They would have had to code in R just to get the proper tables they wanted. Dr. McKean wants to build an R application that would ease the process of creating tables from such a large data set.

The main issue with the current method of accessing the data directly is that the data sets are fairly large. Some of the largest tables contain more than 50 columns and there are more than 40 tables to get data from. This makes for a daunting task where it might take some time to extract data sets. The application would be designed to be a quick and easy way to access the data, combine it with other pieces of data, and select columns. There are some large applications such as Rapid DataMine that allows similar functionality. However, a light-weight, custom R application allows for more flexibility and a more user-friendly interface.

Workflow:

Dr. McKean is looking for a software package that will aid in data transformation and mutation of large datasets. We will use R Shiny, which is a library that would provide a graphical user interface. The user will use the GUI to pull data out from different tables, and then subset the data further based on the variables presented in the table. The end result will be a collection of RDA files.  Some of the functionality is as follows.

* Column extraction and merging
* Row sub setting
* Table merging and stacking
* Visualization through ggplot2

We will be working with a subset of the NHANES data sets, which is a collection of 40+ tables with 40,000 observations. But the package will be flexible enough to load any collection of data sets from the MySQL server and manipulate those data frames.

About The Dataset:

    The National Health and Examination Survey (NHANES) is a large and continuous program of government studies to assess the health and nutrition of people living in the US. This data was provided by the Center for Disease Control and Prevention under the US Department of Health and Human Services. The NHANES program dates back as far as 1959, but for the sake of keeping the datasets consistent we chose to examine datasets from 2007-2014.

    The NHANES interview includes demographic, socioeconomic, dietary, and health-related questions.  The examination component consists of medical, dental, and physiological measurements, as well as laboratory tests administered by highly trained medical personnel. We broke our datasets down into a few categories, DEMOGRAPHICS contain datasets on clients demographic information. LABORATORY contains reports of the presence of different diseases and antibodies within the client. EXAMINATIONS contain information on blood pressure and body measures. DIETARY contains information on the effects of vitamins and minerals on the body. In total we will be using about 40 tables with roughly 30,000 observations.

STORIES

This section will elaborate the stories that we’ll implement for the project. These stories will involve a user who would access our application and perform certain operations to achieve a goal.

Through this project, the user will access with the NHANES data set through a user-friendly interface. The interface would allow users to select and combine different data sets. The user would then download a file with the format of their choice with the selected data.

           The user will be able to select data sets by different properties. These properties could be any combination of the columns within the given data set. For example, the user can select data for a certain set of years.

Once the user has picked properties for a certain data set, they can pick the properties for another data set and combine the two tables. For most data sets, they will combine with the row id number.

After downloading the data sets, the user will be able to quickly view a representation of their data through plots. This part will be implemented using the GGPlot library.

The scope for each of these stories is enough to keep us busy throughout the semester, but it will not be overwhelming. The time to complete these stories is feasible.

RESOURCES

    For our project, we will need computers to develop the program with, software needed to run the R Shiny  applications with, and a server to host the programs. This application will be developed on our own laptops. The software needed to develop R Shiny applications are all freely available. Amongst this software is the R Studio IDE, R packages such as “rshiny”, and MySQL. Dr. McKean also has the hardware and software needed to maintain this project. He will maintain it using his own desktop.

    For now, while we’re still developing, MySQL and the applications will run on a localhost. For the future, we would like to access CS department servers.

    Dr. McKean and the development team all have the necessary resources to develop and maintain the project. We have the machines to develop and maintain the application with. The software needed is free and cross platform. Finally, we will have access to a server to host the application on.

MAINTENANCE

    The majority of the application will be done using the R scripting language, which Dr. McKean is very comfortable with. The automation for the project can be done either in bash scripts or R scripts. Dr. McKean also has the necessary hardware and software to maintain the project.

SECURITY

    This section will focus on any security concerns that we may have. This can apply to the application when in production or when maintaining security for the developers working on the open source project. With each security concern, we will also cover solutions for the security concerns that we will put into practice or build into the application.

    The user will be limited in what they can do. They will not interact with the database directly. Instead, they will select options provided by the GUI interface. This would make SQL injections very difficult if not impossible to execute. The user will not have the option to log into an account either.

    The security concerns would deal with the development process and assuring that the database elements have not been tampered with. Since this is an open source project, users may have the chance to make unwarranted changes to the database information. Saving the R image stat would save the command history and it can contain sensitive information for a developer. THis is troubling for function calls to something like “authorize” that takes in sensitive information such as a token and a username. In order to combat this, we will define in our practices not to save an R image state.

TESTING

    Files containing R code will have an accompanying file called “test.R” that will contain automated tests to run on that piece of the code. There will be both a bash script and an R script available that when called, it will run all of the “test.R” files.

    Our project will utilize unit testing. Some of these tests will test the database. For example, it will check for null data sets or it can test whether the received data set is the same as what received from the table.

When testing the GUI for errors, it will have to be done by hand. When testing the GUI, we will test for conditions such as when no data was selected for a given table.

    When testing the interface, we will give it to our client and observe what could be confusing or misleading with the interface. We will then adjust our interface and give it to him again to test it.

Testing of the functions are initially done in R by hand. After testing is completed in R on each functions, it will then be moved into R shiny and then test again in the same way as it was conducted in R.

The function of column extraction was done by both specifying the name of the column and then the index of each column. In both of these cases, we expect NxP dataframes to be created where N is the number of columns selected and P is the number of rows in the column. Bad inputs for these methods would be either an incorrect or not properly capitalized column name, or the use of an index that is out of the range of the dataframe. The use of an improper name is expected to successful run, but produce a new dataframe of "NULL". The use of an index outside of the range will either produce a dataframe of 0xP if the index use is 0, or the error "undefined columns selected" if the index used is greater than the number of columns in the data frame.

    The process of column merging was done by using the Cbind function and was tested with 1xP dataframes, like what would be extracted, of both similar and dissimilar data types and lengths. The merging of columns of different data types but equal lengths produces no errors as long as the columns have been properly extracted. If one of the columns being merged was improperly extracted by using an index of 0, then this column will be ignored in the output and not produce any errors. All other errors in merging columns of the same length would have been caught in the extraction portion and a similar error would be given. Unless one column is a multiple of another, then merging of two columns of unequal lengths produces an error that the columns are not multiples of each other. If the columns are not equal, but one is a multiple of the other, the shorter ones will be repeated until they are all the length of the longest one.

    The process of row subsetting is done by specifying either the index of the rows you would like, or the use of logical argument on the data in the columns. If logical arguments are used, the possible error is that none of the data in the column you are looking at satisfies the argument. If this is the case, then the subset with be done, but a dataframe with 0 rows will be produced. If subsetting is done by the index of the rows, if the index you are using is outside the range, then you will either receive a dataframe with 0 rows if an index of 0 is used, or if you are an index greater.

        The function of column extraction was done by both specifying the name of the column and then the index of each column. In both of these cases, we expect NxP dataframes to be created where N is the number of columns selected and P is the number of rows in the column. Bad inputs for these methods would be either an incorrect or not properly capitalized column name, or the use of an index that is out of the range of the dataframe. The use of an improper name is expected to successful run, but produce a new dataframe of "NULL". The use of an index outside of the range will either produce a dataframe of 0xP if the index use is 0, or the error "undefined columns selected" if the index used is greater than the number of columns in the data frame.

    The process of column merging was done by using the Cbind function and was tested with 1xP dataframes, like what would be extracted, of both similar and dissimilar data types and lengths. The merging of columns of different data types but equal lengths produces no errors as long as the columns have been properly extracted. If one of the columns being merged was improperly extracted by using an index of 0, then this column will be ignored in the output and not produce any errors. All other errors in merging columns of the same length would have been caught in the extraction portion and a similar error would be given. Unless one column is a multiple of another, then merging of two columns of unequal lengths produces an error that the columns are not multiples of each other. If the columns are not equal, but one is a multiple of the other, the shorter ones will be repeated until they are all the length of the longest one.

    The process of row subsetting is done by specifying either the index of the rows you would like, or the use of logical argument on the data in the columns. If logical arguments are used, the possible error is that none of the data in the column you are looking at satisfies the argument. If this is the case, then the subset with be done, but a dataframe with 0 rows will be produced. If subsetting is done by the index of the rows, if the index you are using is outside the range, then you will either receive a dataframe with 0 rows if an index of 0 is used, or if you are an index greater than the number of rows, you will receive a data frame with NA's in the rows that are greater than the maximum number of rows.

    Table merging and stacking will be done by the use of Cbinds and Rbinds. Possible errors in merging the tables are the same as those in merging the columns. Possible errors in stacking of the columns than the number of rows, you will receive a data frame with NA's in the rows that are greater than the maximum number of rows.

    Table merging and stacking will be done by the use of Cbinds and Rbinds. Possible errors in merging the tables are the same as those in merging the columns. The possible error in stacking is not having the same columns in both of the tables. If an Rbind is used in this case, then it will not stack and an error of “names do not match previous names” will appear.

LEGAL

    There is no Non-Disclosure associated with this project, This project’s purpose is to be an open source tool for any researcher to freely use in their studies and for anyone else interested in accessing the NHANES data sets. Our project will be open source and hosted on a public Github repository. Dr. Mckean nor any of the participants in this project claim any rights or ownership over this software or its distribution.

SUMMARY

    According to stories, we have the time. We have the resources needed. Nothing is legally keeping us from doing the project. With all of these powers combined, the project is feasible! The project will be useful to any researcher who wants to create his/her own tables to perform evaluations and run models on any complex dataset.