Authors: Brett Schlesinger (bgs59), Natalie Isak (ngi2), Lillian Joyce (lhj32)

Vision: Our vision for this project was to create a database management system. Our system has the capacity to read a CSV file and can perform a variety of functions. In particular, we can read/add/delete rows/columns, encrypt/decrypt the csv, create scatter plots based on any two rows within the CSV, generate variables, change the data file through multiplication/addition, perform functions like mean/median/mode/standard deviation/variance on the data file or variables, and perform statistical analysis on our data. Additionally, it updates the current file as functions are performed on it and saves the current file. All of these functionalities can be accessed by "make run." We accomplished all of the tasks up to and including excellent scope for MS2. The largest way this vision has evolved from previous sprints is that we implemented encryption/decryption of our data instead of implementing a data scraping functionality because we believed that this was a more applicable and interesting addition to our program.

Summary of Progress: In this sprint, we accomplished many new features. For satisfactory scope, we added new mathematical functions: mean, median, mode, standard deviation, and add/subtract/multiply/divide elements. This included tests and corresponding commands as well. For good scope, we changed the import command so you can only import one CSV file at once. This greatly simplified our command functionality and overall improved the user experience. Also, we wrote a rep_ok function to ensure that all uploaded CSV files had an equal number of rows and columns. This was a crucial addition since it ensured that the representation invariant (that all rows had the same number of elements and all the columns had the same number of elements) was satisfied. Finally, for excellent scope we added in data visualization functionality (using Ocaml's library Gnuplot), RSA encryption of CSV files, and statistical analysis (generate variable, Z statistical tests, and regression tests). In our demo, we showed make test, and make run (including the graphing, encryption/decryption, various statistical tests, and various mathematical tests).

Activity Breakdown:

Brett:

- Changed command to only import one CSV file at once
- Integrated Gnuplot with command and main files to enable graphing functionality
- Documenting/testing corresponding additions

Lill:

- Mathematical functions (mean, median, mode, standard deviation, and add/subtract/multiply/divide elements)
- Statistical functions (generate variable, Z statistical tests, and regression tests)
- Documenting/testing corresponding additions

Natalie:

- Rep_ok function
- RSA encryption/decryption
- Documenting/testing corresponding additions

Productivity Analysis: Overall, we believed we were very efficient with our time. We accomplished precisely what we planned for each scope, all in a timely manner. In MS1, we exceeded the expectations we set for ourselves for each scope. Therefore, for MS2, we set more ambitious guidelines for ourselves. Through hard work, we accomplished satisfactory scope, good scope, and excellent scope.

Scope Grade: Excellent Scope
In our plan for MS2 that we created after MS1, we had three scopes: satisfactory scope, good scope, and excellent scope. For satisfactory scope, we successfully added mathematical functions. For good scope, we added in rep_ok and fixed the way CSV files were imported based on feedback from our demo. Finally, for excellent scope we added graphing functionality, RSA encryption, and statistical tests. For each scope, we added test cases and commands (so that the user could “make run” and access the new functionality). Finally, we thoroughly documented each section.