

SASSi Programming Language

Group Leader: Carl Morgan Paul Salama Xiaotang Zhang

1 INTRODUCTION

SASSi's purpose is to create an efficient and easy to use scheme for handling statistics. The language is designed to let the user make quick and simple programs to deal with various statistical problems.

The hope is that SASSi will be seen as an alternative to the disaster that is SAS. Despite its longer name, SASSi would be smaller than the unnecessarily bloated SAS. It would also be free, allowing those who can't stomach SAS's \$120 price tag, namely everybody. Compute variance and standard deviations graphically. Learn more from a generated pie chart with a small glance at a computer screen. Think about never having to figure out what libraries are needed and how to use them. SASSi makes it all possible.

1.1 *SASSi Overview*

SASSi: A simple, powerful, efficient, easily made graphical, customizable, and architecture neutral language.

Borrowing the form set forth by the Java white paper, we are using similar buzzwords to describe the features of our language. This only seems natural since our language will compile into Java code. SASSi has features that are helpful to any organization or student using statistics and can easily be learned and applied. The following explanation of the previous buzzwords will further help you to understand the characteristics of SASSi:

Simple

When designing a system for statistics, there are two focuses: the set up of the model and the representation of that model. Because SASSi inherently has its own data types and constructs, it is simple to set up a statistics system. Also, it is easy for any user to make additions to the model or change behavior of an entire program. Any changes that the users make are immediately noted and take effect in the GUI version. The language is simple and easy to learn because of its similarity to English. This parallel also helps in understanding exactly what each command does and simplifies how to use the language.

Powerful

SASSi gives you the power to make statistical models a reality in just a few lines of code.

Easily Made Graphical

Since SASSi is compiled into Java code, creating a GUI is simple to do. The user need not worry about creating any code for graphical representation. The compiler knows how to represent everything and where to place what where. All the user needs to do is choose what models he or she wishes to see. This was touched on briefly previously. The users will also have a text field that displays any important information. However, this does not mean that the GUIs are made by the language. The GUIs are actually made with the language through a general template.

Efficient

The main goal of this language is to take the strain off of the user and the computer. By keeping the libraries small and the manual compact, anyone can use SASSi. Using smaller libraries with SASSi allows the user to utilize more memory for the computations and other various tasks on his/her computer. The algorithms will also be efficient so as to maximize the utility of the computer.

Customizable

Many different aspects of SASSi's implementation are customizable. For example: the users can choose if they want to add variables or subtract variables from a model. The users can customize the models to behave as they wish. Also, the users can customize the messages to be displayed on screen, i.e. instead of just spitting out results the user can have the results displayed in a nice convenient manner. Also, the users will be able to design their own functions so that if they aren't satisfied with the tools we give them, they can easily implement their new function and not experience the headache of trying to find software that supports it.

Architecture Neutral

SASSi code is compiled into Java byte code which makes it architecture neutral. This means that SASSi can be developed on any platform that has the Java compiler, the JVM, and the GUI. You could write the code once for the setup of the GUI and the little details of the statistics system you design and literally use it anywhere. This will make distributing the software easier.

1.2 SASSi Functionality

Graphics

SASSi offers several nice graphical options. The users can decide how the results are seen. The user can choose from a variety of formats including but not limited to: bar chart, pie chart, and curves. The users also have an easy way of seeing the results in a text form. These of course are the default settings, the user can customize it to behave differently if they desire. This simplicity

allows every onlooker to quickly understand what is going on with the statistics system and if their model is behaving the way they desired. The message area allows users to gain feedback from the work they have done.

Data Types

SASSi will support two types of data: arrays and numbers. The numbers will follow the basic patterns that are defined in most mathematical applications, i.e. integer, floating point number, long, and so on. The arrays will be used to represent phasors and matrices. The basic algebraic functions (addition, subtraction, multiplication, and division) will all be supported for both number and array arguments.

Algorithms

It will also support a number of various algorithms. These will be imported using libraries pertaining to the functions you will be using. This will save space and make the program run more efficiently. For example: one library will have all the commands you need for doing basic statistical functions like mean, median, and mode. Another library will allow you to do variance and standard deviation functions. Regressions will have their own library: including linear, multiple, and nonlinear regressions. Various libraries will be incorporated as well for variance, expectation, and a few other functions needed for statistics.

1.3 Code Samples

SASSi is programmed at

```
sampleprogram      /* the name of the program*/
begin             /* the beginning of our sample program*/
    number n = 6, m; /* define variable n, m to
number and set n's value to 6 */
    array A , B, c, d; /* define variable A,B,c,d
to array*/

    A = [1,2,3];
    B = [4,5,6];
    c = A * n;
    d = B / n;

    m = median(A);

    print "Hey! This is our sample program!"; /*
output a string */
    print m; /* output a number 2 */
    print A; /* output 1,2,3 */
    print d bar; /* output array D in bar chart */
end             /* end of our sample program */
```

1.4 Conclusion

In conclusion, SASSi is a language designed to help people using basic statistics. The name SASSi is just the name SAS with an extra Si for humor. For all intensive purposes, you still pronounce as if it had a y and not an i. SASSi makes statistics simple to see and simple to modify. Variations in it and on it can be made quickly. The interface allows the user to truly interact with the statistics system that it is designed for. Since the code is compiled into Java, it can be run on any machine that has the JVM or Java Virtual Machine. The goal of SASSi is to take a statistics system and make it as efficient as possible while simplifying the commands and keeping a user-friendly interface.