

Journal Report 4

9/23/19-9/26/19

Jialin Tso

Computer Systems Research Lab

Period 2, White

Daily Log

Monday September 16

Researched SymPy's matrices and Python's statistics library and decided which functions to implement. Installed statistics library on workstation.

Tuesday September 17

Outputted a list of calculator's one-var stats (exact order) in addition to population standard deviation, population variance, and mode. Also printed out array of typed numbers using pandas DataFrame.

Thursday September 19

Converted arrays of numbers into Matrix and implemented matrix functions such as addition, multiplication, inverse, power of, determinant, and reduced row echelon form. Began researching and writing linear regression function.

Timeline

Date	Goal	Met
Sept 3	Install Wabbitemu and convert text to expression for basic functions	Emulator works and program can calculate many functions (multiple digit number words separated by dashes)
Sept 9	Install SpeechRecognition and word2number; text to expression for factorial, logarithms, and other useful functions	SpeechRecognition, word2number, and num2words installed. Factorial and logarithms work.
Sept 16	Solve integrals, derivatives, and permutations and combinations;	Indefinite and definite integrals and derivatives (not at specified value) solved. Permutations and combinations work for a certain format.
Sept 23	Solve matrices and find 1-var stats for user-inputted list.	Common matrix functions like addition, reduced row echelon, and determinant work. 1-var stats outputted.
Sept 30	Begin experimenting with SpeechRecognition and implementing speech to text. Find regression equations from user-inputted lists	

Reflection

Overall, this week went pretty smoothly since I was able to allow the program to return lots of information. A challenge I ran into was that when I first tested my onevarstats function on a list of distinct numbers, I would get an error message because statistics.mode() would only return a mode if one existed. I resolved this issue by having the program return 'None' if there was no mode. Another discrepancy I faced was with Q1 and Q3. I used 25th and 75th percentiles for the program's Q1 and Q3 respectively. However, when I compared the TI-84's and program's one var stats, the program's Q1 and Q3 were different (more precise?). A general observation I have made from my program is that my functions work as intended, but they require a strict format. I am ok with this insight because the fact that the calculator can understand my typed commands is already pretty amazing.

Example of new inputs and outputs of current program: Input: one-var-stats left parentheses left-bracket one comma two comma three right-bracket right parentheses

Expression output: onevarstats([1,2,3]) 0 0 1 1 2 2 3 (prints out array as pandas data frame)
 Answer: ('Mean:2', 'Sum:6', 'SumSquares:14', 'Sx:1.0', 'Popstd:0.816496580927726', 'SampVar:1', 'Popvar:0.816496580927726', 'Size:3', 'Min:1', 'Q1:1.5', 'Median:2', 'Q3:2.5', 'Max:3', 'Mode:None')

Input: matrix left parentheses left-bracket left-bracket one comma two comma three right-bracket comma left-bracket three comma two comma one right-bracket right-bracket right parentheses times matrix left parentheses left-bracket zero comma one comma one right-bracket right parentheses

Expression output: matrix([[1,2,3],[3,2,1]])*matrix([0,1,1]) Answer: Matrix([[5], [3]])

Input: determinant left parentheses matrix left parentheses left-bracket left-bracket one comma zero comma one right-bracket comma left-bracket two comma negative one comma three right-

bracket comma left-bracket four comma three comma two right-bracket right-bracket right parentheses right parentheses

Expression output: `det(matrix([[1,0,1],[2,-1,3],[4,3,2]]))` Answer: -1

Input: reduced-row-echelon left parentheses matrix left parentheses left-bracket left-bracket one comma zero comma one comma three right-bracket comma left-bracket two comma three comma four comma seven right-bracket comma left-bracket negative one comma negative three comma negative three comma negative four right-bracket right-bracket right parentheses right parentheses

Expression output: `rref(matrix([[1,0,1,3],[2,3,4,7],[-1,-3,-3,-4]]))` Answer: (Matrix([[1, 0, 1, 3],[0, 1, 2/3, 1/3],[0, 0, 0, 0]]), (0, 1))

Input: linreg left parentheses left-bracket zero comma one right-bracket comma left-bracket zero comma two right-bracket right parentheses

Expression output: `linreg([0,1],[0,2])` Answer: LinregressResult(slope=2.0, intercept=0.0, rvalue=1.0, pvalue=0.0, stderr=0.0)