# AMS 325 Final Project: Sudoku

By Rohan Basavaraju & Patrick McLoughlin

# Table of Contents

# Introduction

- The “Rubik’s Cube of the 21st Century”

    - Euler’s Latin Squares (Smith, 2005), “Diabolical Magic Squares” (Boyer, 2006)

- Modern Origins and Worldwide Popularity (Smith, 2005)

    - New York’s *Dell Magazines* publishes “Number Place” in 1984, credits Howard Garns

    - Japan’s *Nikoli* in 1984 prints with title Su Doku (“the numbers which are single”)

    - Wayne Gould programs game generator, pitch to *Times* of London, rapid growth in US & UK

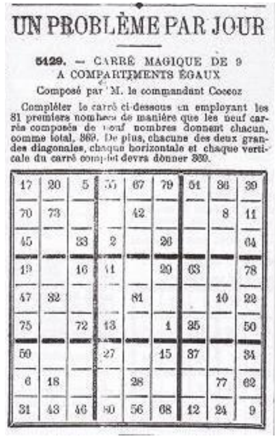
- How to play (Easybrain LTD, 2018)

    - Each ***row***, ***column***, and 3 x 3* ***sub-grid*** of a 9 x 9* array contains one instance of each character

    - Difficulty scales with number of given cells populated, characters traditionally numbers 1 through 9

** traditional dimensions, many variants exist; “Japanese” has 2 x 2’s within 4 x 4*

*From* **Le Siécle** *(Bordeaux, FR), Nov. 19, 1892* :



UN PROBLÈME PAR JOUR

5129. — CARRÉ MAGIQUE DE 9 A COMPARTIMENTS ÉGAUX

Composé par M. le commandant Coccoz

Compléter le carré ci-dessous en employant les 81 premiers nombres de manière que les neuf carrés composés de neuf nombres donnent chacun, comme total, 369. De plus, chacune des deux grandes diagonales, chaque horizontale et chaque verticale du carré complet devra donner 369.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 17 | 20 | 5 | 55 | 67 | 79 | 51 | 36 | 39 |
| 70 | 73 | | | 42 | | | 8 | 11 |
| 45 | | 33 | 2 | | 26 | | | 64 |
| 19 | | 16 | 11 | | 29 | 63 | | 78 |
| 47 | 32 | | | 81 | | | 10 | 22 |
| 75 | | 72 | 13 | | 1 | 35 | | 50 |
| 59 | | | 27 | | 15 | 37 | | 34 |
| 6 | 18 | | | 28 | | | 77 | 62 |
| 31 | 43 | 46 | 80 | 56 | 68 | 12 | 24 | 9 |

## Project Overview

Objective: Create a program to test Sudoku boards

- Generate a state-based Sudoku board
- Identify if an element is viable for a given position
- Test if a filled Sudoku board is a working solution
- Analyze and visualize the Sudoku board’s accuracy
- Scale the implementation across normal, letter, and Japanese Sudoku

Team Contributions:

- Rohan: Code (algorithms, testing, documentation), Presentation & Report
- Patrick: Presentation (slides), Report (citations), Code (debugging, testing)

## Techniques

Software Tools:

- Python 3 Programming Language: NumPy, Matplotlib (PyPlot, Colors)
- Atom, Visual Studio, Google Colab IDEs
- GitHub Version Control System (VCS)

Sudoku Rules (Easybrain LTD, 2018):

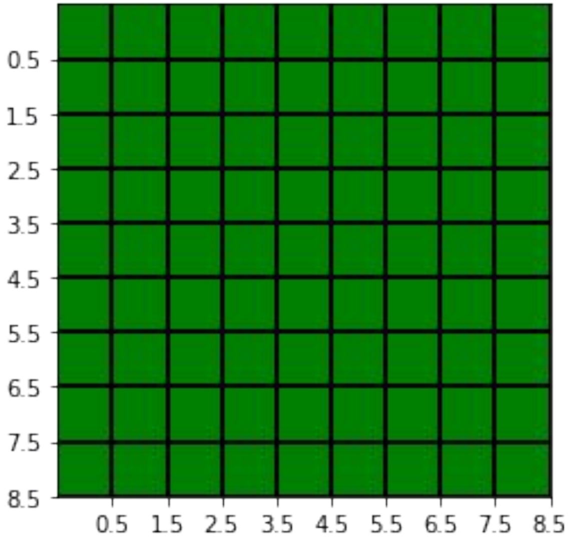
- All values must be valid (e.g. 1-9 [integer], 1-4 [integer], ‘A’-‘I’ [string])
- Only 1 value per column & row (normal/letter: 9x9, japanese: 4x4)
- Only 1 value per sub-box (normal/letter: 9 -> 3x3, japanese: 4 -> 2x2)

## Coded Functions

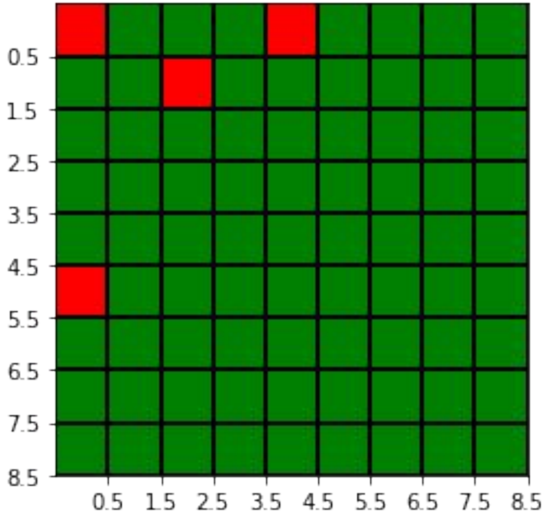
```
createEmptyBoard(version="normal") -> board: 2-D array initially filled with zeroes
checkElement(board, element, row, column, version="normal") -> boolean: [in-]valid
value
possibleElements(board, row, column, version="normal") -> rangeVals: list of valid
values
addElement(board, element, row, column) -> board: updated with added element
removeElement(board, row, column) -> board: updated with removed element
testBoard(board, version="normal") -> boolean: [in-]valid Sudoku solution
accuracyBoard(board, version="normal") -> percentage of correct boxes
viewAccuracyPlot(board, version="normal") -> colorful visual plot
```

## Results & Discussion - Normal Sudoku

Of the 81 total values in the board, 81 values are correct. This means 100% of the values are correct.

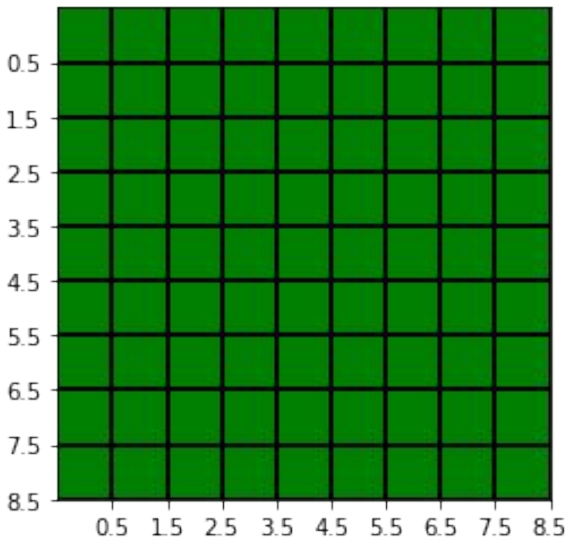


Of the 81 total values in the board, 77 values are correct. This means 95% of the values are correct.

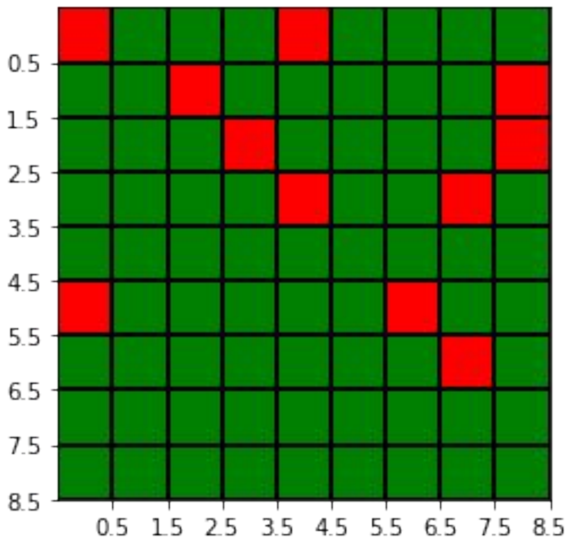


## Results & Discussion - Letter Sudoku

Of the 81 total values in the board, 81 values are correct. This means 100% of the values are correct.

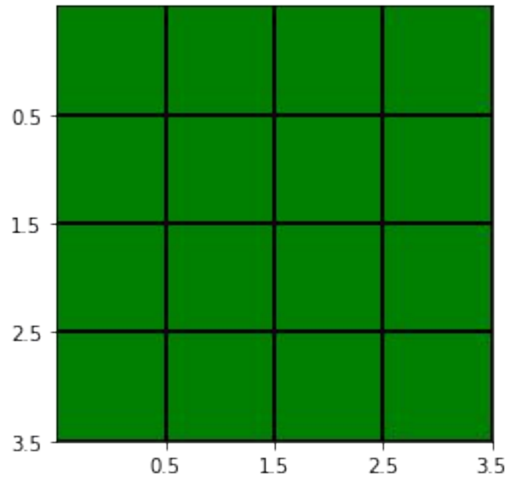


Of the 81 total values in the board, 70 values are correct. This means 86% of the values are correct.

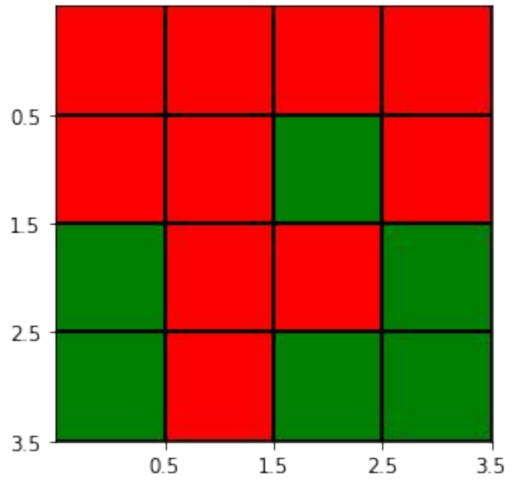


## Results & Discussion - Japanese Sudoku

Of the 16 total values in the board, 16 values are correct. This means 100% of the values are correct.



Of the 16 total values in the board, 6 values are correct. This means 38% of the values are correct.



## Conclusions

- We successfully developed and tested a suite of Python functions to evaluate, analyze, and visualize success in Sudoku (normal, letter, Japanese)
- Rohan: I learned how to properly document my code using docstrings & comments along with creating test cases
- Patrick: The Python skills taught have enabled development of a tool to help play a fun game (possibly saving time and frustration!)

Future Investigations:

- Implementing a Sudoku board generator and solver
- Expanding the Sudoku tester's scalability to analyze a Sudoku board with any number of dimensions (e.g. n x n) or values (e.g. integers, characters, etc.)

**Thank you!**

## Bibliography

- Smith, D. (2005, May 14). ‘So you thought sudoku came from the Land of the Rising Sun’ *The Guardian*

- Boyer, C. (2006, May 24). ‘Les ancêtres français du sudoku’ *Pour La Science*

- Easybrain LTD (2018, August)  ‘Sudoku Rules’ [<https://sudoku.com/sudoku-rules/>](https://sudoku.com/sudoku-rules/)