RESULTS FOR AutomatedTesting.ipynb Jupyter Notebook

Determining an RSquared Value for different test setups when attempting linear regression for a formula for a 3x3 invertible matrix

Use automated testing to perform iterations of linear regression to ensure that the fitted r squared values for each cell are statistically different from one another.

The notebook is AutomatedTesting.ipynb

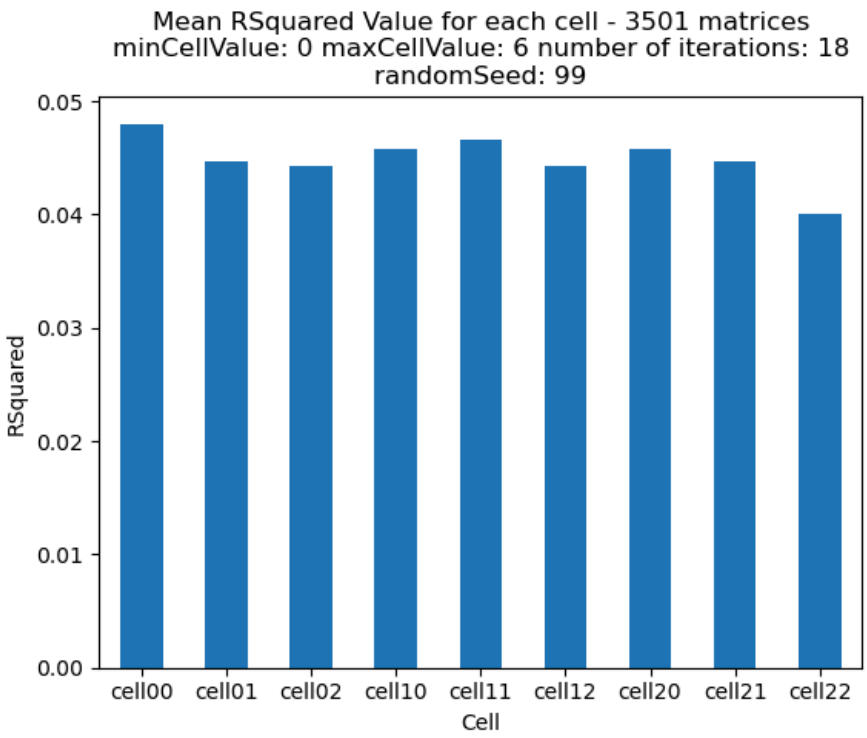
The code for this notebook is based off of LinearRegressionMatrixInverse.ipynb

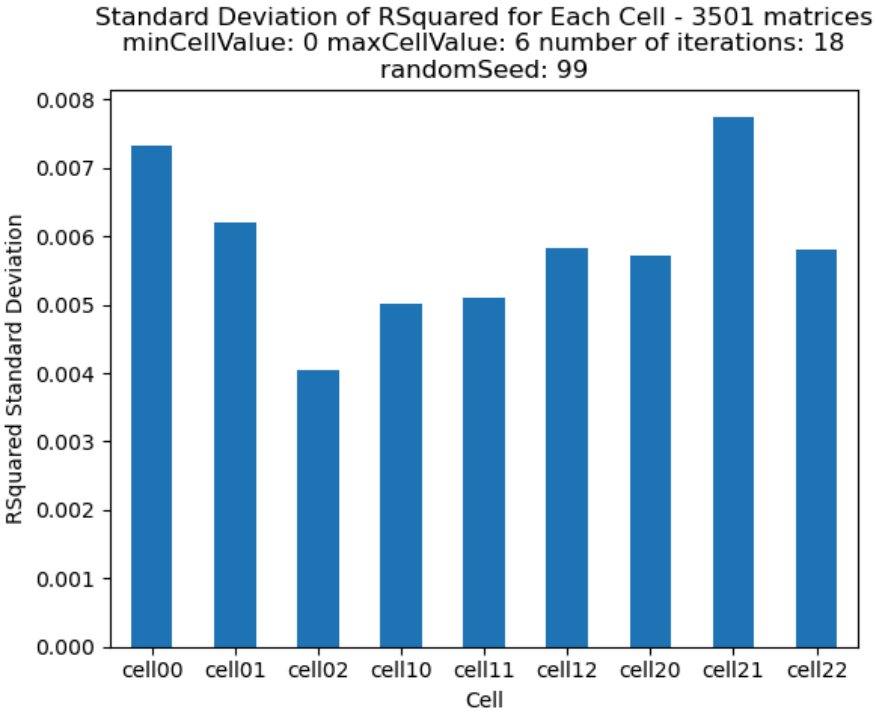
The code is condensed and functionalized to be run in a loop to continue until the ANOVA p value is 0.05 or better. This can be set as a variable as well in case it is desired to adjust this cutoff.

The components for linear regression are every matrix element and all possible combinations of one matrix element squared and multiplied by every other matrix element and the cube of each matrix element.

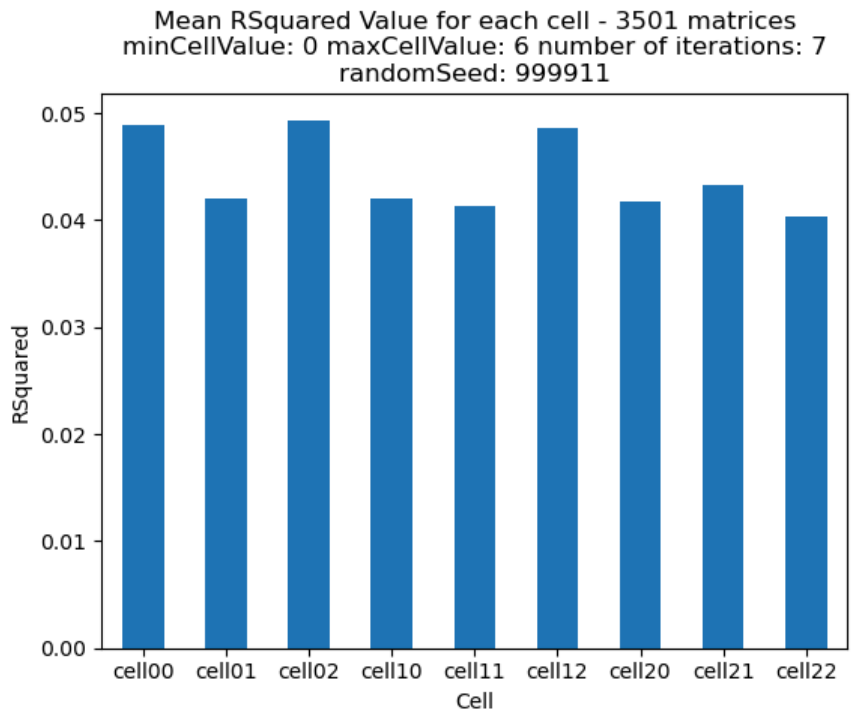
The linear regression will fit, and then generate a new set of random matrices and fit again. An ANOVA pvalue will then be calculated and more fits will continue until the ANOVA for the linear regression for each cell of the matrix inverse is statistically satisfactory.

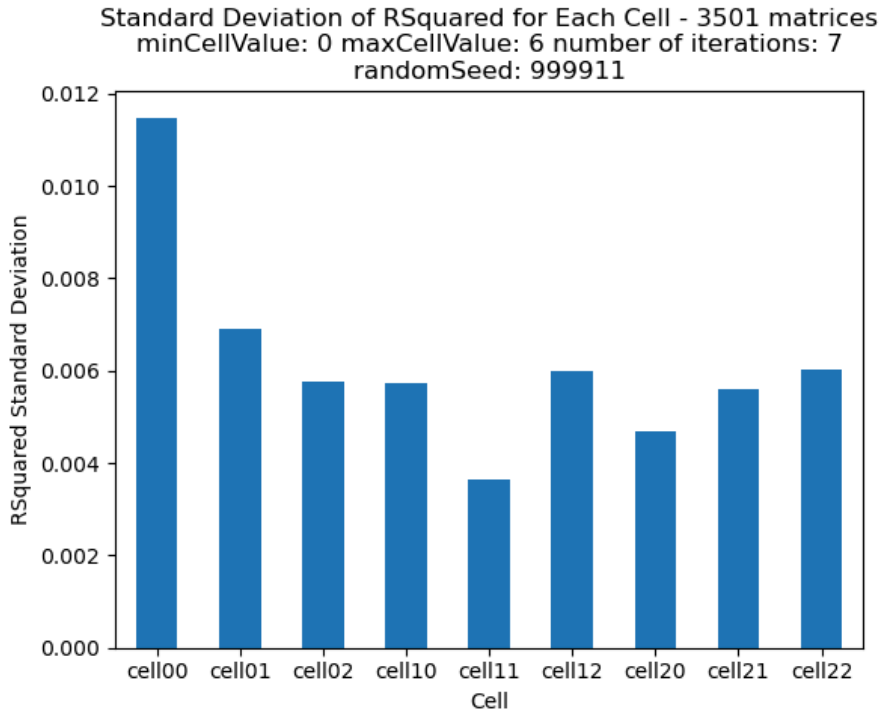
The random seed can be set to adjust the starting randomly generated matrices. The results from the AutomatedTesting.ipynb notebook will be compiled in this document.





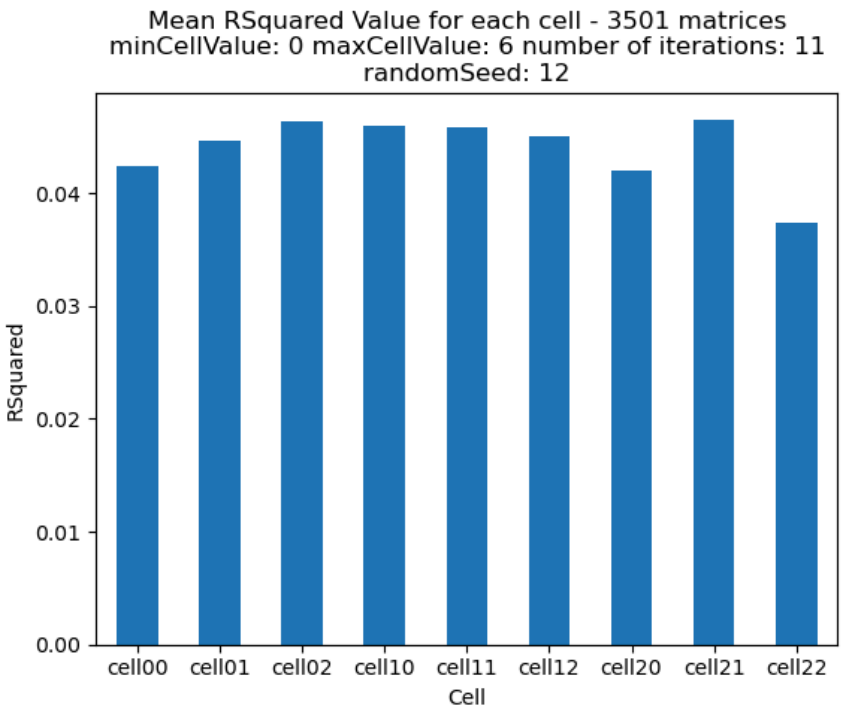
The RSquared value accross all cells is 0.044904182811598095 - 3501 matrices - minCellValue: 0 maxCellValue: 6 randomSeed: 99

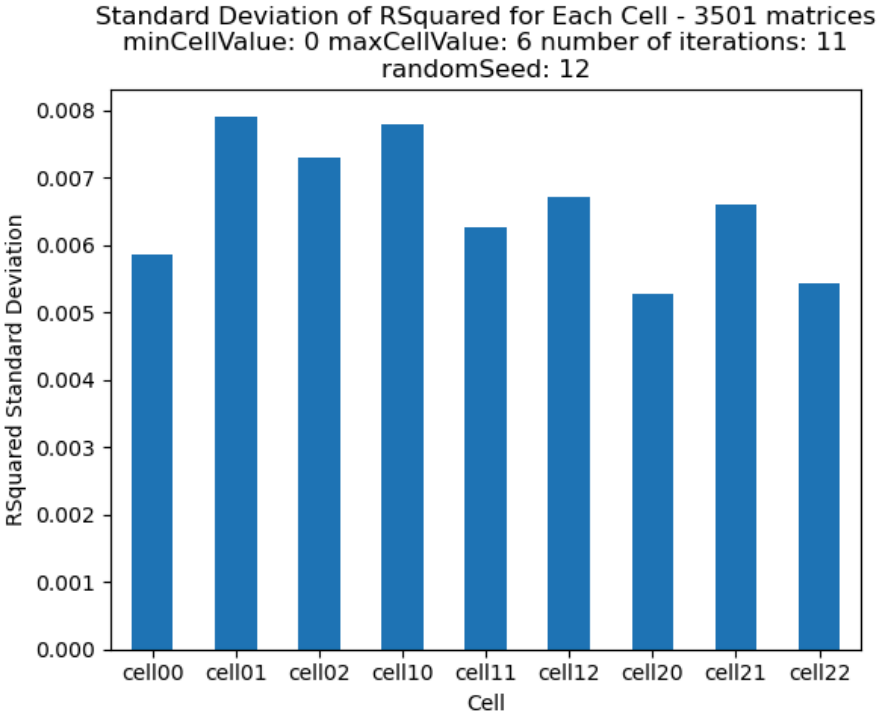




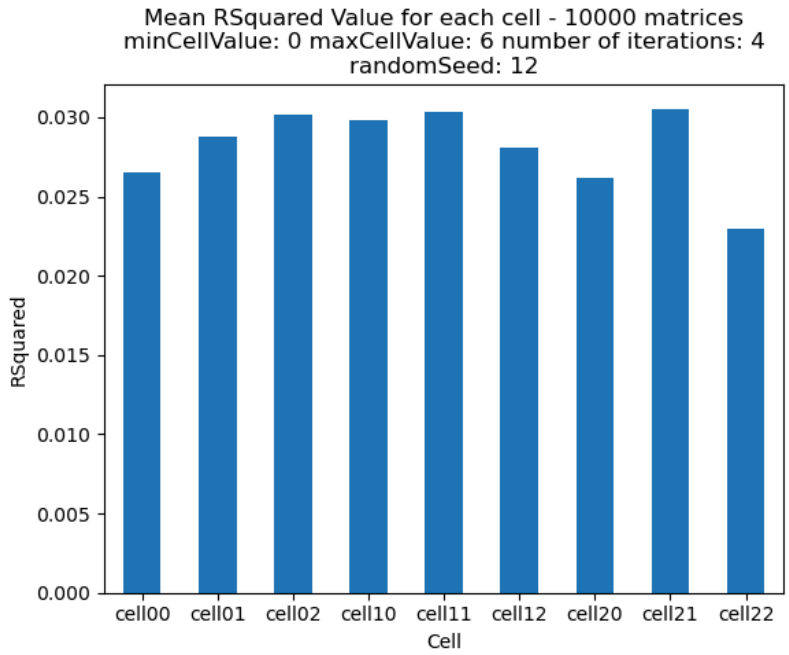
The RSquared value accross all cells is 0.04419329027637289 - 3501 matrices - minCellValue: 0 maxCellValue: 6 randomSeed: 999911

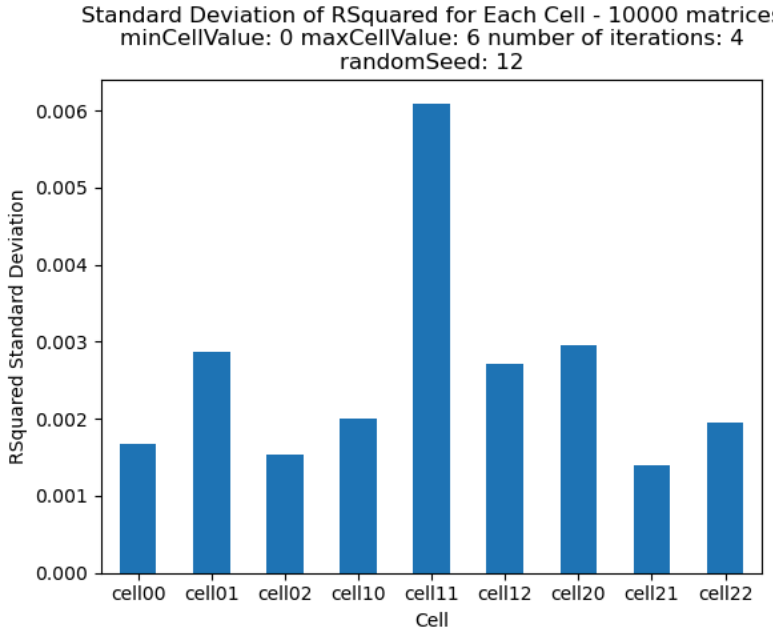
Well it only took 7 iterations for random seed 99911 to generate a sufficient p value for rsquared. It took random seed 99 18 iterations. The means across cells are not matching. 0.449 and 0.441 are the mean r squared values across the cells.





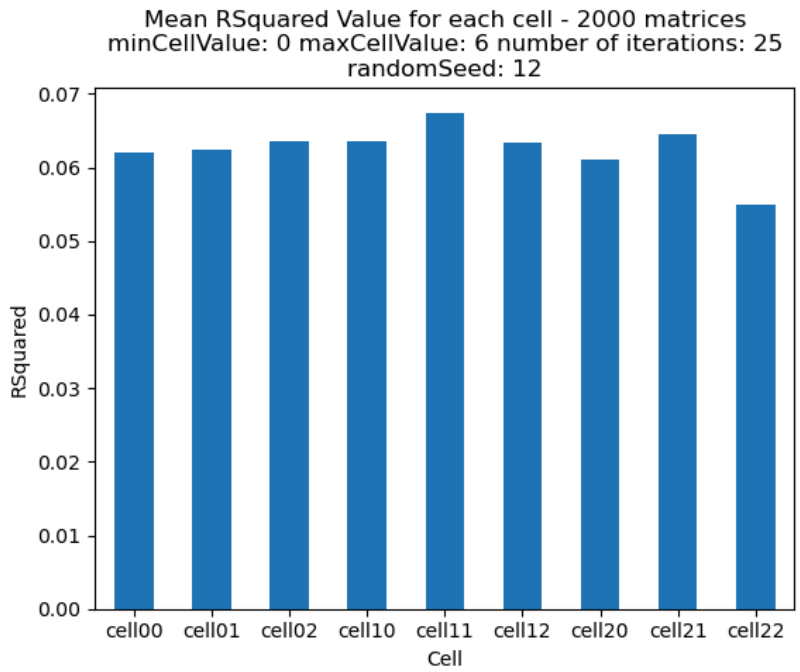
The RSquared value accross all cells is 0.04402159240054801 - 3501 matrices - minCellValue: 0 maxCellValue: 6 randomSeed: 12

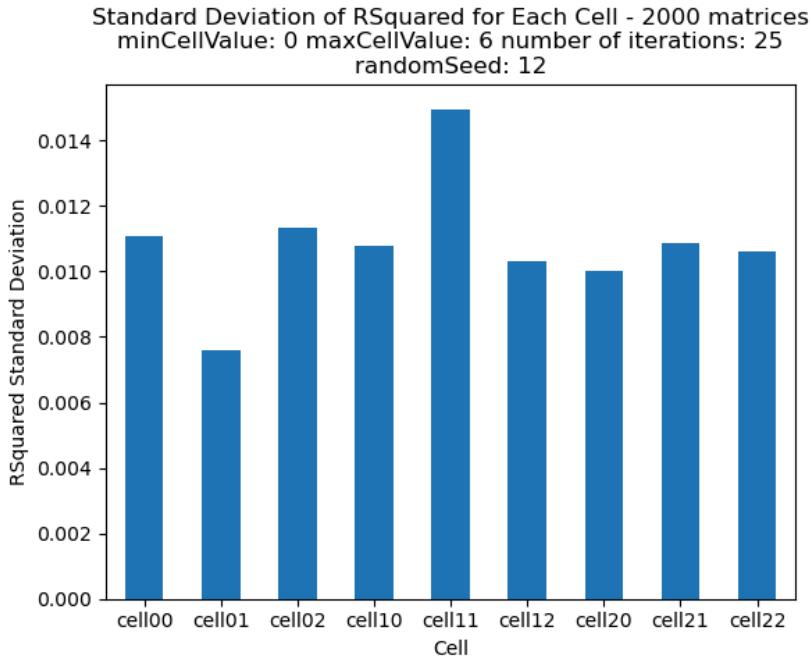




The RSquared value accross all cells is 0.028154899128897737 - 10000 matrices - minCellValue: 0 maxCellValue: 6 randomSeed: 12

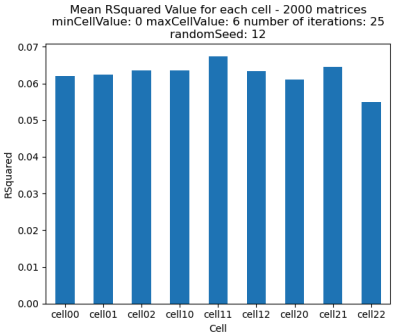
10000 matrices took only 3 iterations to reach the desired p value! 3500 matrices took 7-18 for the results so far

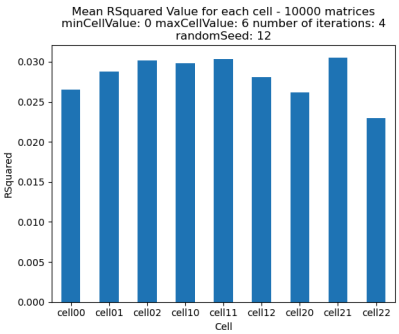


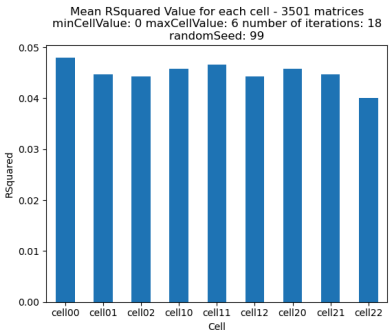


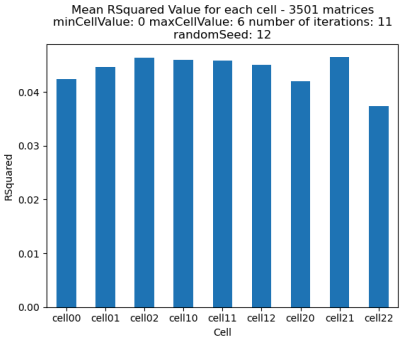
The RSquared value accross all cells is 0.0625457952716379 - 2000 matrices - minCellValue: 0 maxCellValue: 6 randomSeed: 12

Looking at every graph of the mean R Squared values across cells we can see a pattern starting to develop. 7 iterations random seed 999911 is the only graph that does not match patterns.







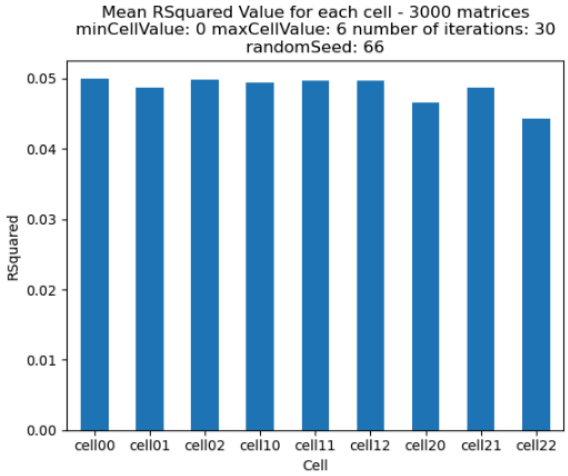


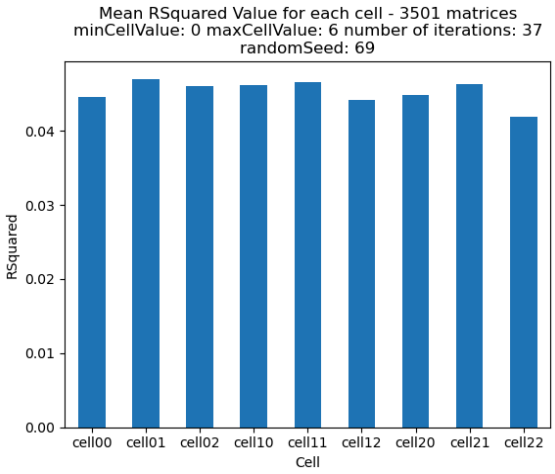
We can see a bit of a pattern emerging. However I will run the Notebook with many results twice and see if the graphs line up this will give us a pattern.

It would be best to write automated code to generate this pattern.

Conclusion

With 3501 matrices @ 30 iterations and 3000 matrices @ 37 iterations I am unable to generate a solid pattern as we can see for these two cell12 and cell00 have different locations to the group.



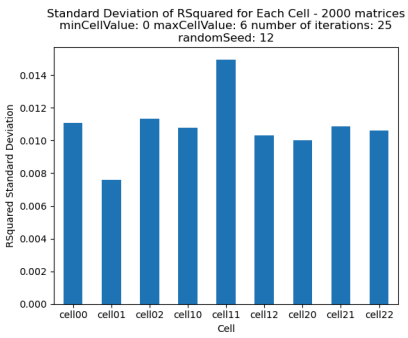


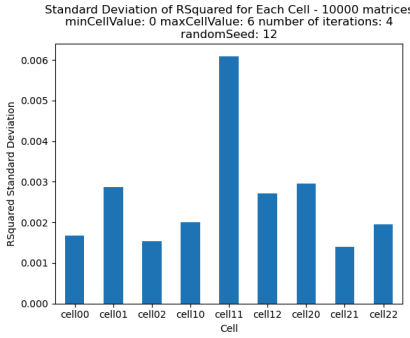
We can say that cell22 fits less well, cell21 beats cell20. Further investigation is required to determine why this is but the general pattern is shown in the above two graphs.

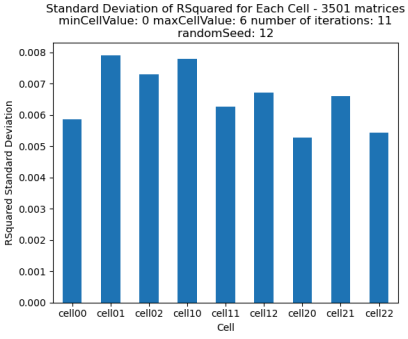
The accepted rsquared value for the average across all cells for the matrix inverse from every matrix element and all possible combinations of one matrix element squared and multiplied by every other matrix element and the cube of each matrix element is 0.044

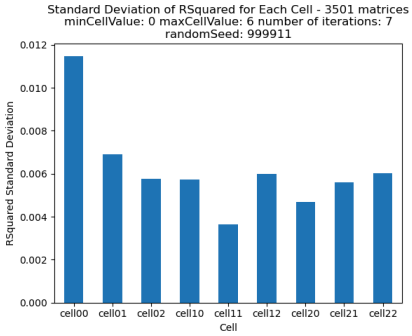
An rsquared value of 0.044 is not good enough for an accurate linear regression.

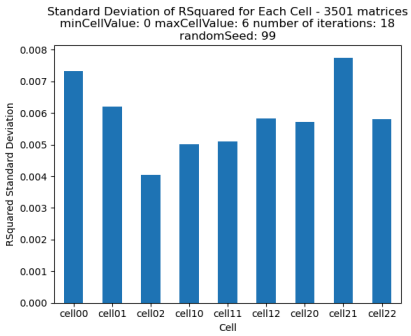
The standard deviations vary much more than the mean rsquared values for each cell. Below are 5 standard deviation graphs next to one another. Matching a pattern to these will be very difficult. This show that attempting to do a linear regression is highly nonlinear for this data set.











Next Steps

Try this automated testing for 4x4 matrices and 5x5 matrices. Perform analysis for larger numbers and floating point numbers in the original matrices. Generate code to generate patterns that are acceptable for rsquared values across cells.

Determine a mathematical reason for why the patterns for rsquared exist across different cells and why standard deviation of r squared values in iterations is so non linear.