Sensitivity Analysis for Component Design:

Analysis of Success Assured Data

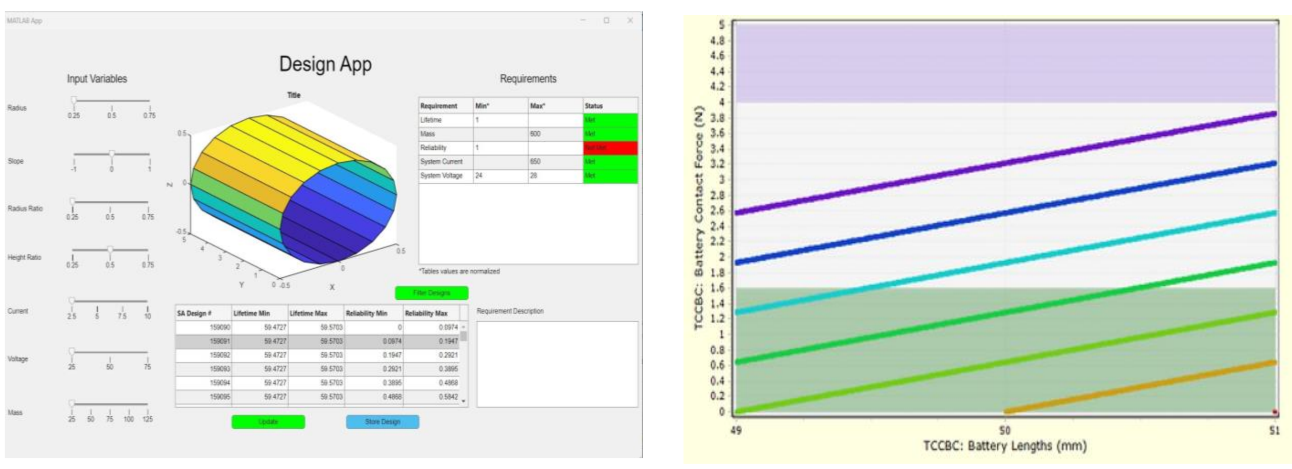
Talia Duffy, University of Illinois Urbana-Champaign
B.S in Statistics and B.S in Journalism (May 2025)
7585 | Mentor: Kate Gabet Hoffmeister

Abstract

The objective of this Advanced Digital Engineering Pathfinder is to create a mini-app that gives insight into the space of component designs and requirements.

The app is populated with data from Success Assured (SA), a set-based concurrent engineering software. Inputs and outputs must be manually chosen for export from SA, which prompts the question: Was the right set of inputs and outputs chosen to make our app useful to designers?

Sensitivity analysis is a set of statistical methods that can help answer this question.



Left: Current UI for the ADE component design app. Right: An example of output curves from SA.

Sensitivity Analysis

Sensitivity analysis is a way to quantify how changes in one variable affect changes in another variable within a system. Which inputs are driving most of the uncertainty in our outputs? Where do we need more data, and which variables can we fix?

Sobol' Method – Decomposition of variance

Decomposition of variance, or Sobol' method, quantifies how variances in the inputs affect variance in the output. There are two types of Sobol' indices:

- Main effect index, Sj, -- describes sensitivity associated with the input alone.
- Total effect index, STj -- describes sensitivity associated with the input and all interactions with other inputs

$$S_j = \frac{V(E(Y|X_j))}{V(Y)} \qquad \sum_{j=1}^{k} S_j \leq 1$$

$$S_{Tj} = \frac{E(V(Y|X_{-j}))}{V(Y)} \qquad \sum_{j=1}^{k} S_{Tj} \geq 1$$

Success Assured Data

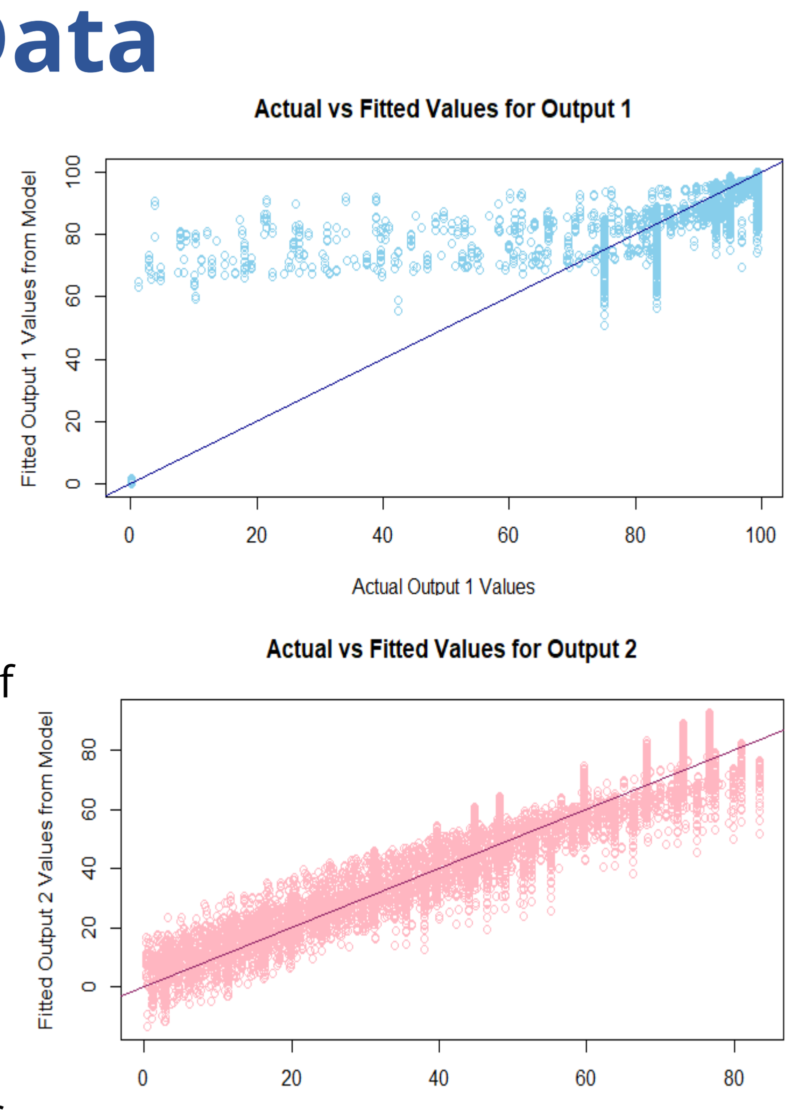
Multiple adjustments to typical methods were implemented to navigate the peculiarities of SA data.

Outputs as a curve

- SA exports data with outputs represented together as a curve because of unseen parameters
- Curves were approximated with minimum, median, and maximum of each output to make six outputs

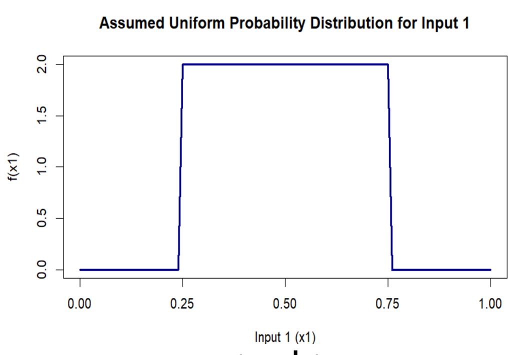
Approximating the "black box" model

- Estimation of sensitivity indices requires a function to evaluate outputs with input samples
- SA "black box" / web of relationships approximated as a model with all linear terms, quadratic terms, and two-way interactions



Above: Scatter plots of actual vs fitted values show how well our models fit the training data. While the plot for output 1 looks poor, the correlation is 0.97: only 4.7% of the data falls in the range between 10 and 70. However, one of our main objectives now is to improve model predictions.

Monte Carlo Estimation of Sobol' Sensitivity Indices



Sample construction $X_j \sim U(min(x_j), max(x_j))$

- Inputs assumed to have uniform distribution
- Two samples of size N from each input's distribution were generated to construct two matrices M1 and M2. M1 represents the "original" sample, M2 represents a "resample."

Estimates

Procedure calls for the calculation of the following estimates:

$$\hat{E}(Y) = \frac{1}{N}\sum_{r=1}^{N} f(x_1^{(r)}, x_2^{(r)}, \ldots, x_k^{(r)}) \qquad \hat{V}(Y) = \frac{1}{N-1}\sum_{r=1}^{N} f^2(x_1^{(r)}, x_2^{(r)}, \ldots, x_k^{(r)}) - \hat{E}^2(Y)$$

$$\hat{U}_j = \frac{1}{N-1}\sum_{r=1}^{N} f(x_1^{(r)}, x_2^{(r)}, \ldots, x_k^{(r)}) f(x_1^{(r')}, x_2^{(r')}, \ldots, x_j^{(r)}, \ldots, x_k^{(r')})$$

$$\hat{U}_{-j} = \frac{1}{N-1}\sum_{r=1}^{N} f(x_1^{(r)}, x_2^{(r)}, \ldots, x_k^{(r)}) f(x_1^{(r)}, x_2^{(r)}, \ldots, x_j^{(r')}, \ldots, x_k^{(r)})$$

Where *r* indicates an observation from the original sample, M1, and *r'* indicates an observation from the "resample," M2.

We can then estimate the sensitivity indices.

$$\hat{S}_j = \frac{\hat{U}_j - \hat{E}^2(Y)}{\hat{V}(Y)}$$

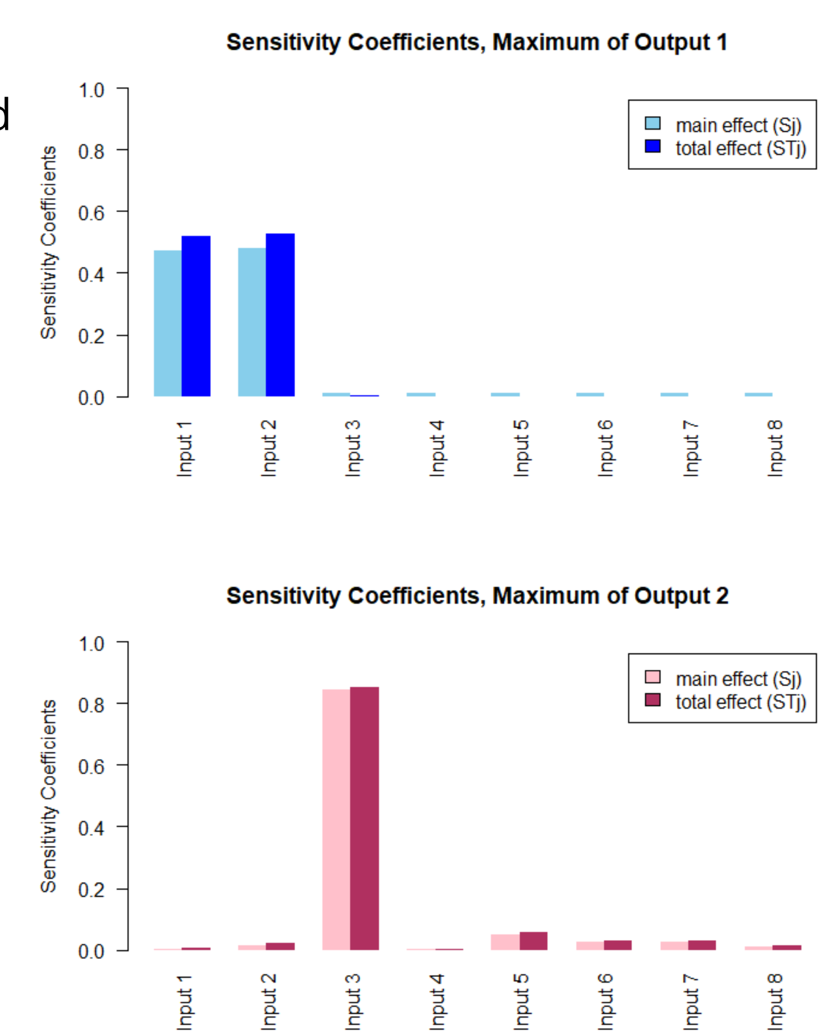
$$\hat{S}_{Tj} = 1 - \frac{\hat{U}_{-j} - \hat{E}^2(Y)}{\hat{V}(Y)}$$

Implementation

This process was first tested on one dataset, and then generalized and structured in an R Markdown file. Users can read in their dataset; define inputs, outputs, and sample size N; and tweak the output models as necessary.

After setup, users run the script, which is generalized to calculate Sobol' sensitivity indices for any dataset. The script produces tables and visualizations of the sensitivity indices.

Input	Out 1: Min	Out 1: Med	Out 1: Max	Out 2: Min	Out 2: Med	Out 2: Max
Input 1	NaN	0.5212	0.5192	0.0133	0.0012	.0075
Input 2	NaN	0.5174	0.5265	0.0012	0.0063	0.0217
Input 3	NaN	0.0023	0.0014	0.8088	0.8607	0.8505
Input 4	NaN	-0.00003	-0.00002	0.0023	0.0024	0.0022
Input 5	NaN	0.00009	0.0002	0.1135	0.0681	0.0563
Input 6	NaN	0.0004	0.0003	0.0313	0.0308	0.0313
Input 7	NaN	0.0002	0.0002	0.0297	0.0290	0.0284
Input 8	NaN	0.0001	-0.000009	0.0118	0.0123	0.0123
Sum	NaN	1.0418	1.0476	1.0119	1.0108	1.0101



Above: Bar charts visualize the sensitivity associated with different inputs. Currently, the script makes six charts.
Left: This table displays Monte Carlo estimates of the total effect sensitivity indices for the minimum, median, and maximum of each output. Highest sensitivities are highlighted.

Summary and Future Work

The methods developed so far have promising results for test datasets. SMEs and other apps have supported our results, but some tasks remain:

- Improving our models so we get more accurate predictions and sensitivity estimates
- Partially automating model construction based on properties of the outputs
- Translating the script to MATLAB
- Integrating with the design mini-app

References

1. Saltelli, Andrea. *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*. Wiley, 2007.
2. Sobol', I.M. "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates." *Mathematics and Computers in Simulation*, vol. 55, no. 1–3, Feb. 2001, pp. 271–280, https://doi.org/10.1016/s0378-4754(00)00270-6.

Special thanks to Steve Crowder (5574) for providing lots of advice and letting me BORROW some books on modeling, sensitivity analysis, and Ernest Hemingway :)