An Interactive Tool to Visualize Results from Uncertainty Quantification

Matt Isaac
e-mail: matt.isaac@aggiemail.usu.edu

April 9, 2019

Outline

Motivation

Uncertainty Quantification (UQ):

- Is a branch of simulation analysis that is used often in engineering contexts;
- Assists engineers and decision-makers in finding a balance between design efficiency and design sufficiency (avoid under/over designing);
- Provides a quantification of how variability in a system can influence the end state of that system.

UQ Overview - Terminology

system response quantity (SRQ): the output (i.e. prediction) from an engineering model.

engineering model: A mathematical model that defines the relationship between the parameters (model inputs) and SRQ (model output).

aleatory uncertainty: Uncertainty from randomness inherent to a given parameter. This type of uncertainty is irreducible.

epistemic uncertainty: Uncertainty from a lack of knowledge about a given parameter. This type of uncertainty can be reduced by more information about the parameter.

UQ Overview - Algorithm

Example: A beam required to bear a certain load. The SRQ is a representation of the capability of the beam.

$$\text{Aleatory model inputs: } A_1, A_2, A_3$$

$$\text{Epistemic model inputs: } E_1, E_2, E_3$$

$$SRQ = E_1^{E_2} A_3 A_2^{A_1} + E_3$$

- A domain expert selects a probability distribution for each input.

Example adapted from Ewing et al. (2018)

UQ Overview - Algorithm

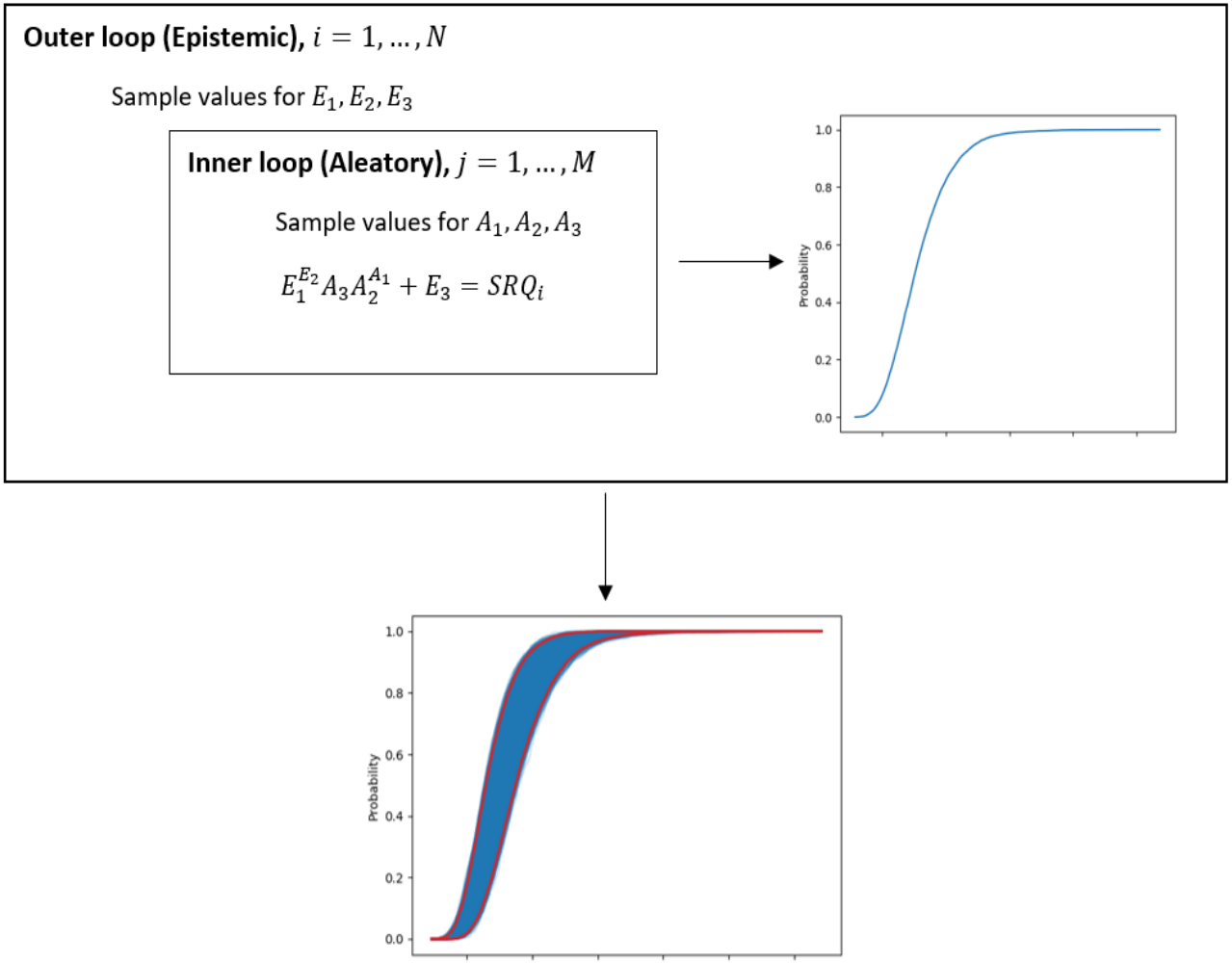
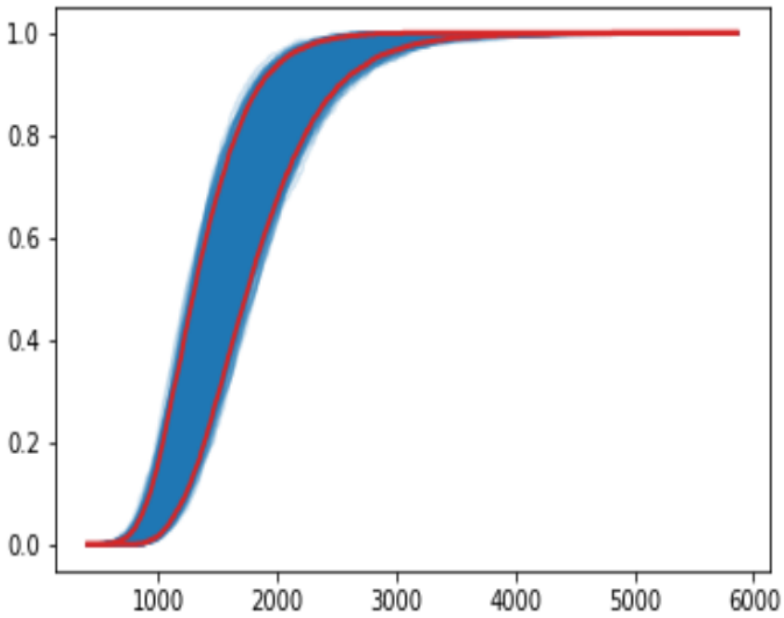


Diagram adapted from Ewing et al. (2018)

Project Goal

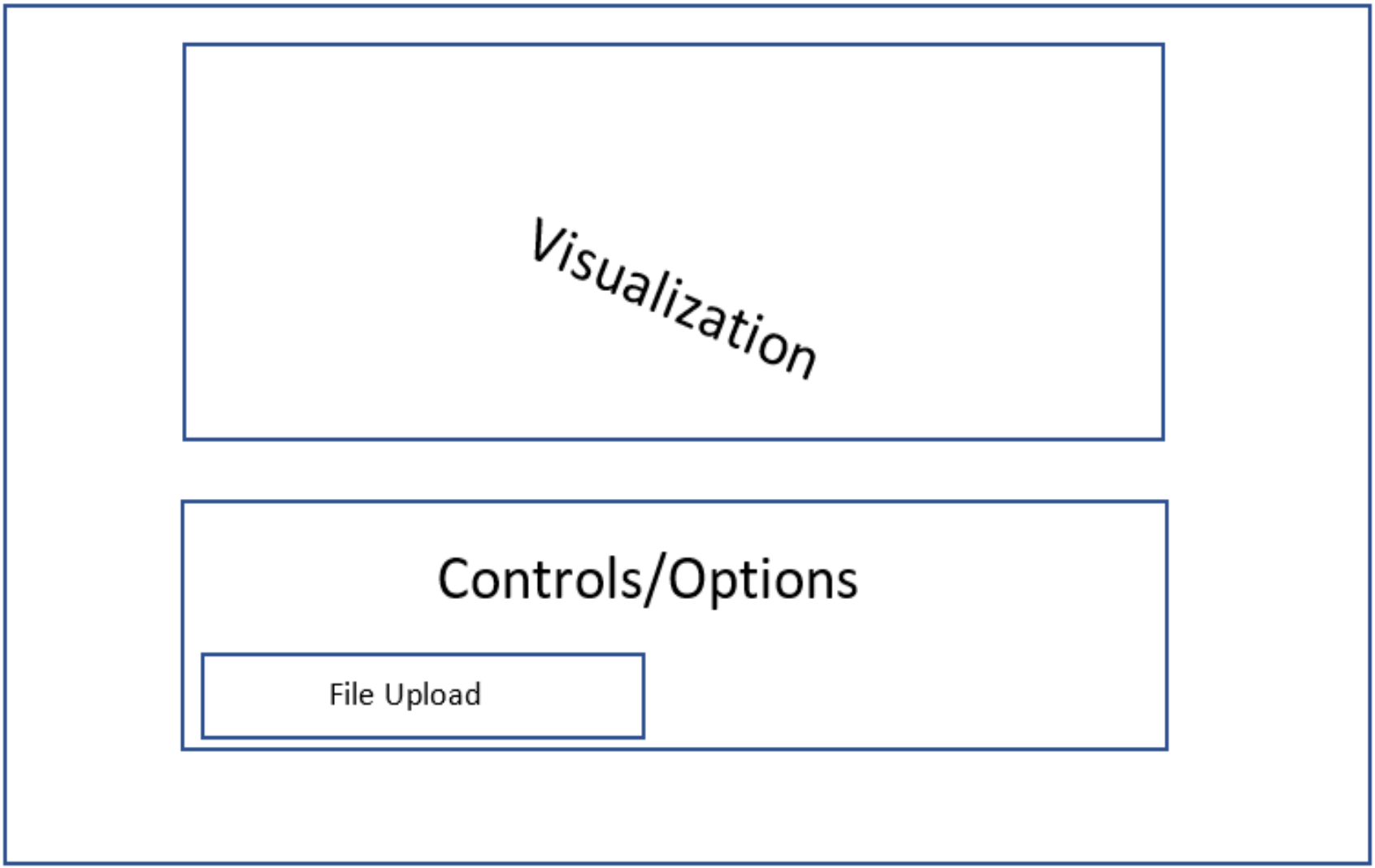
Goal: Develop an interactive tool to create meaningful visualizations (like the one below) and useful interpretations of results from a UQ simulation.



Key Features

- upload data via a `.csv` file
- toggle P-box and CDF's on/off
- select upper/lower percentiles to be used in P-box calculation
- adjustable transparency for CDF ensemble
- extract either a probability interval or an SRQ interval from P-box
- download and save visualization as a `.png` file

Planned Layout



R Packages

shiny: The `shiny` package (Chang et al. 2018) provided the framework on which the application was developed and deployed. It also contains the implementations for all user-interface components (toggle buttons, check boxes, numeric inputs, sliders, etc.).

ggplot2: The plotting functionality of the `ggplot2` package (Wickham 2016) was used to generate the actual visualization and to add, remove, or adjust components on the plot.

shinydashboard: The `shinydashboard` package (Chang and Borges Ribeiro 2018) was used as a wrapper around the `shiny` framework to give a clean, polished appearance to the application.

shinyalert: The `shinyalert` package (Attali and Edwards 2018) was used to generate pop-up messages.

shinyWidgets: The `shinyWidgets` package (Perrier et al. 2019) was used to to create custom toggle buttons.

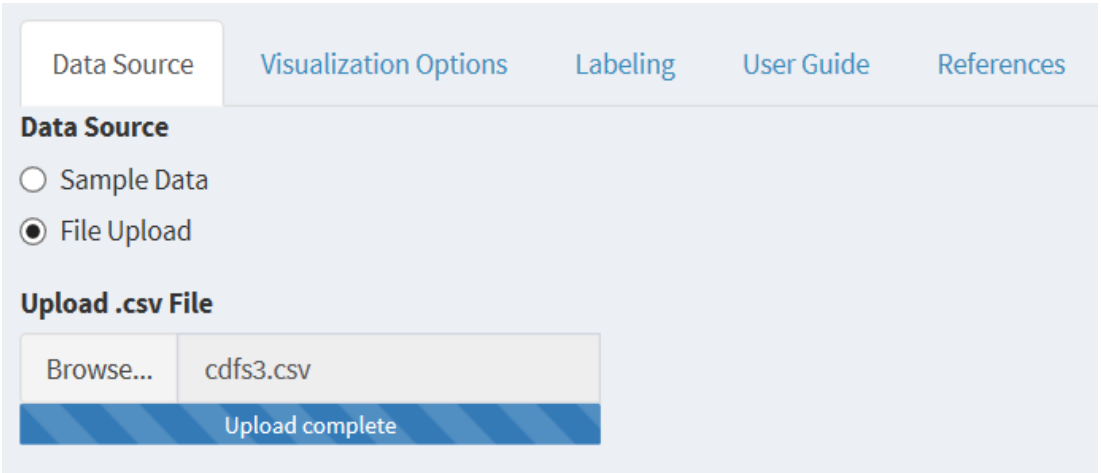
Deployment

Shiny App currently deployed at:

`https://misaac.shinyapps.io/UQViz/`

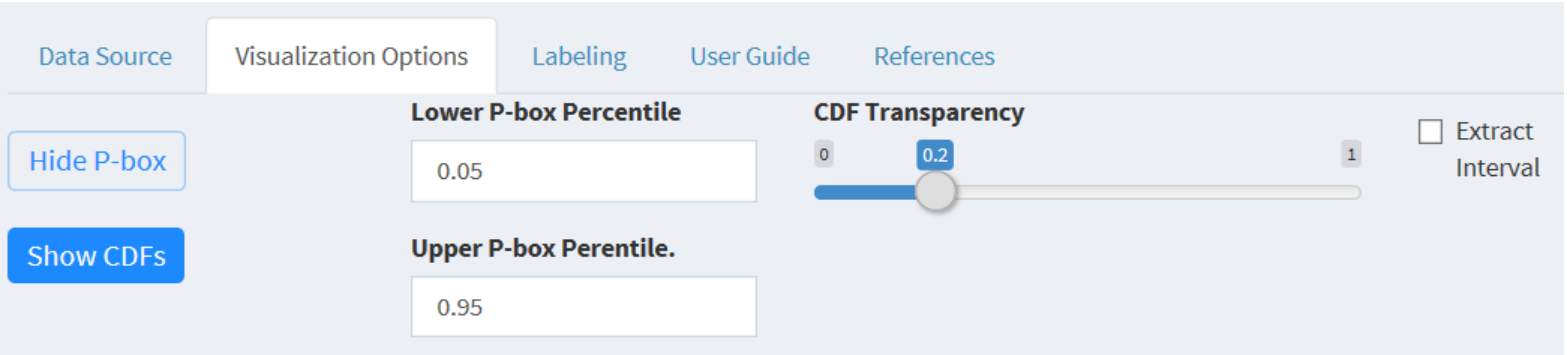
Data Source

- Users can upload `.csv` files for visualization. The files should have the x values in the 1st column, and the CDFs in subsequent columns.
- For convenience, a sample data set is also provided

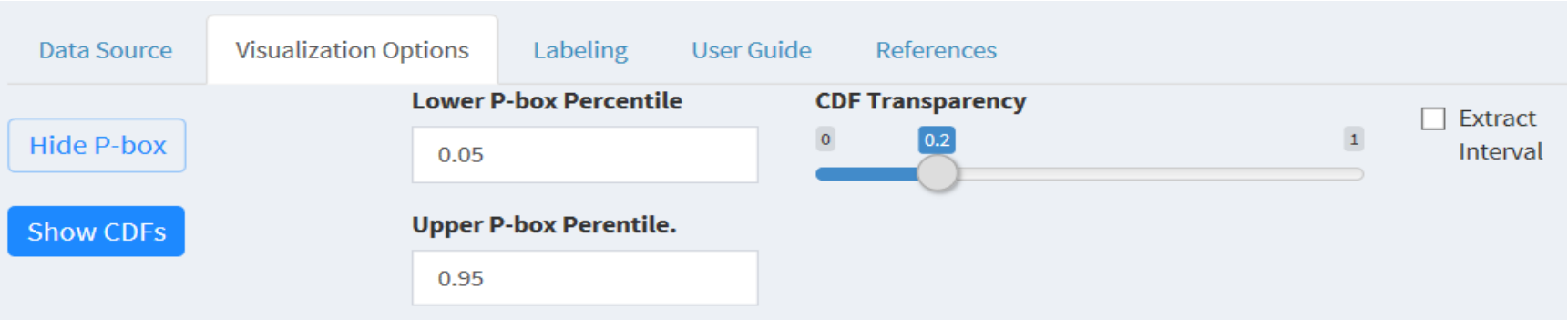


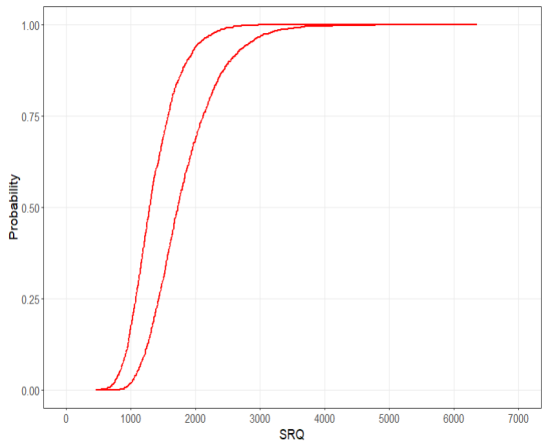
Visualization Options

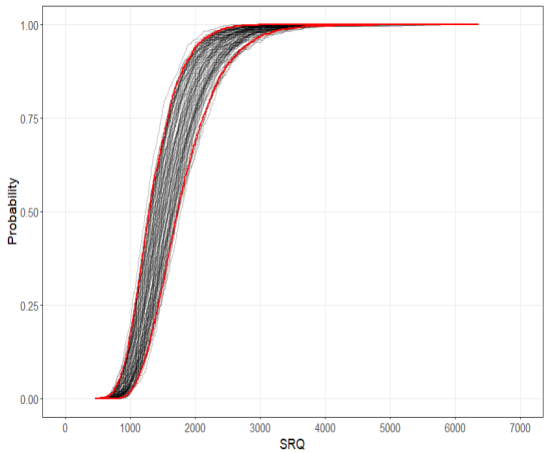
- Several options are provided to allow customized visualizations

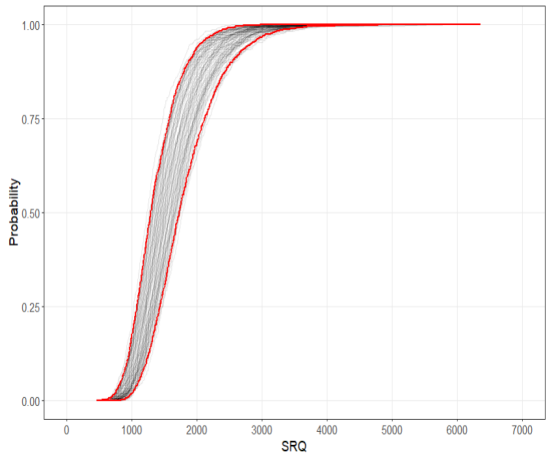


Visualization Options & Examples



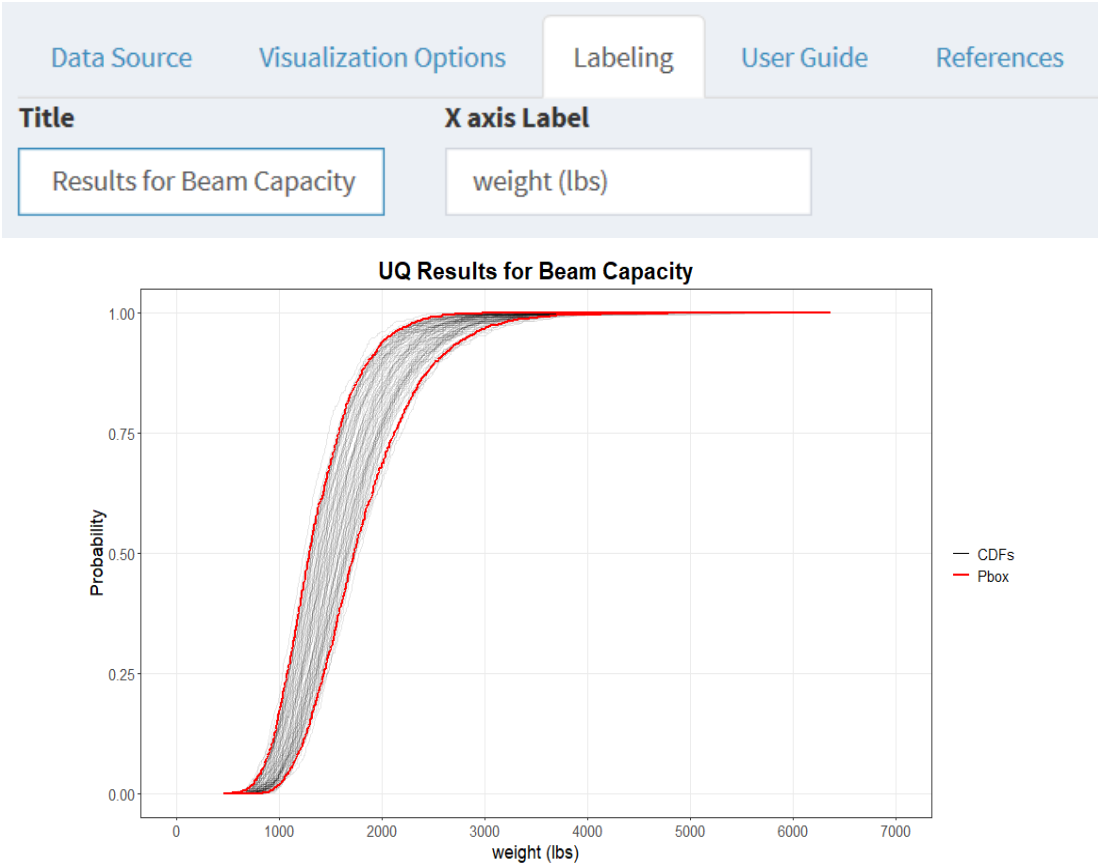






Labeling

- Custom title and x-axis label can be added to adapt visualization to domain.

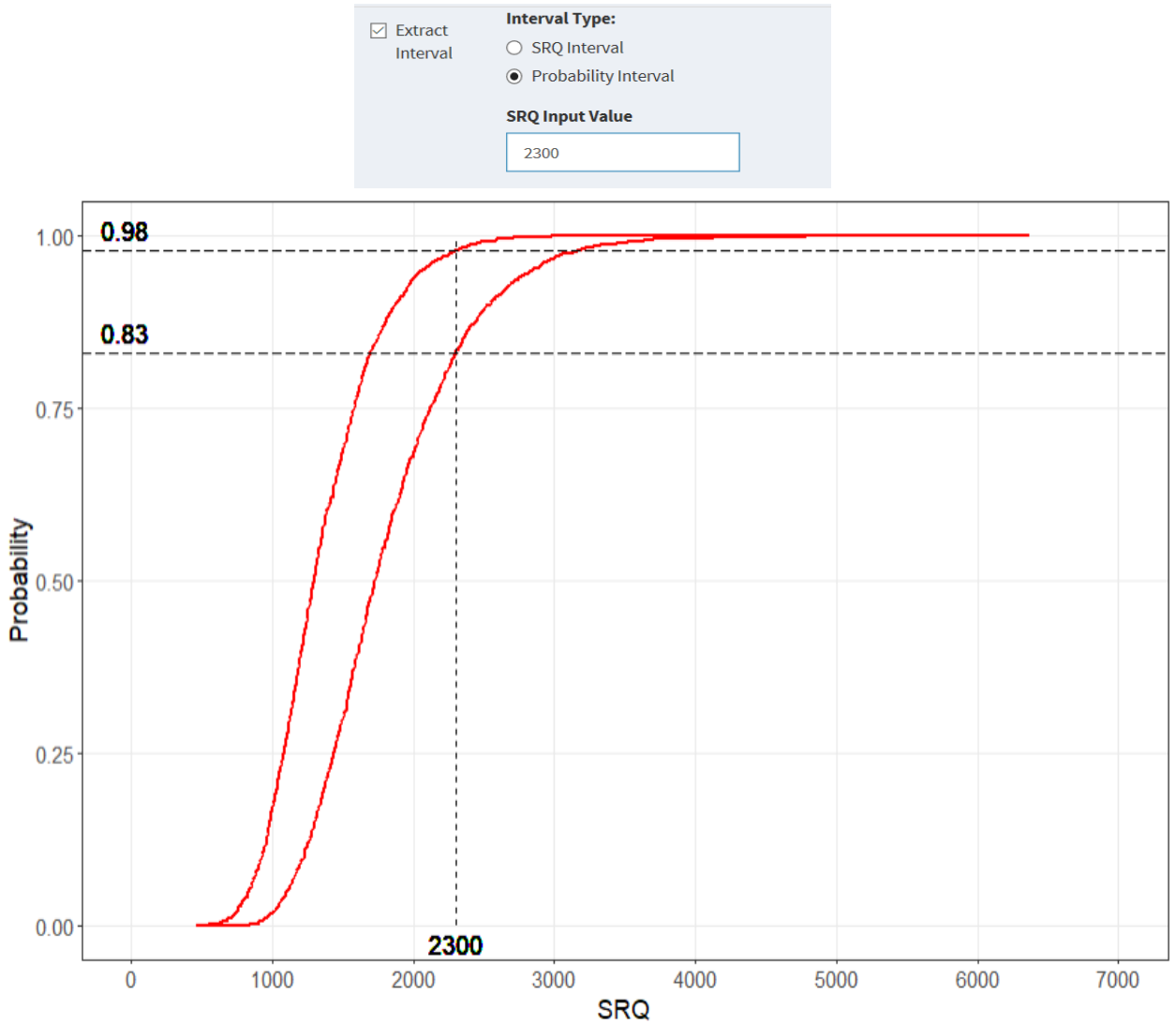


Interpretation - Extract Probability or SRQ Intervals

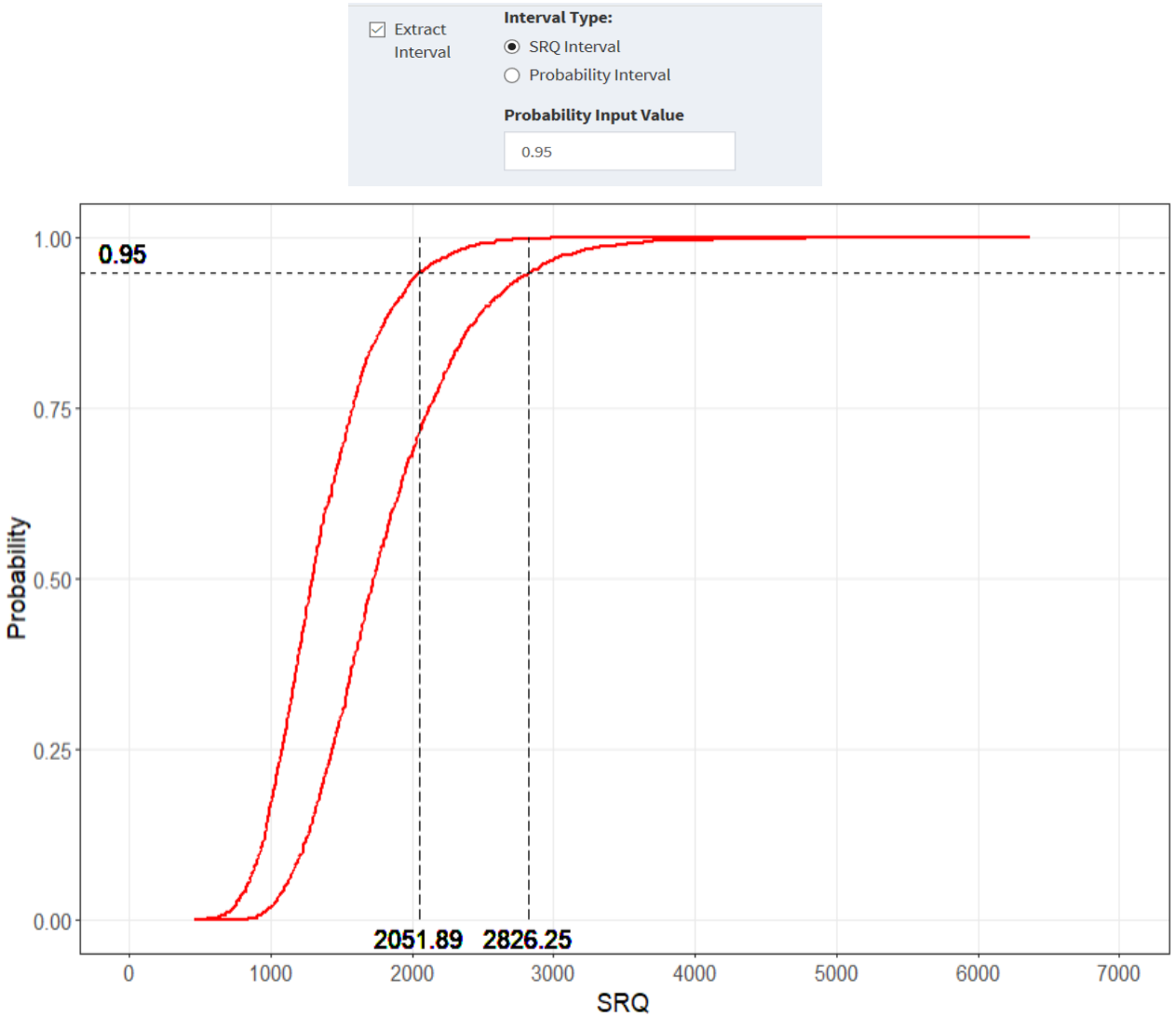
There are two ways a P-box can be used for interpretation:

- Input SRQ value to extract probability interval
- Input probability value to extract SRQ interval

Interpretation - Extract Probability Interval



Interpretation - Extract SRQ Interval



Challenges

- Long render time for hundereds (or thousands!) of traces on a plot
- Reasonable workaround: If the user wishes to display the CDFs on the final version of the plot, adjust all visualization options and controls while the CDFs are hidden, keeping the plot rendering times to a reasonable length. Then, as the last step, show the CDF ensemble.

Future Work

- Research and implement other possible ways to visualize UQ results
- Create a self contained UQ simulation tool – perform entire process, start to finish
  - Define a mathematical model
  - Select parameters as aleatory or epistemic
  - Specify the number of inner and outer loop iterations
  - Choose visualization type
  - Interact with and customize dynamic visualization

Questions ?? —

or e-mail: `matt.isaac@aggiemail.usu.edu`

Sources

Attali, D., Edwards, T., 2018. shinyalert: Easily Create Pretty Popup Messages (Modals) in 'Shiny'. R package version 1.0 (https://CRAN.R-project.org/package=shinyalert).

Chang, W., Borges Ribeiro, B., 2018. shinydashboard: Create Dashboards with 'Shiny'. R package version 0.7.1 (https://CRAN.R-project.org/package=shinydashboard).

Chang, W., Cheng, J., Allaire, J., Xie, Y., McPherson, J., 2018. shiny: Web Application Framework for R. R package version 1.2.0 (https://CRAN.R-project.org/package=shiny).

Ewing, M., Liechty, B. C., Black, D., 2018. A General Methodology for Uncertainty Quantification in Engineering Analyses Using a Credible Probability Box. Journal of Verification, Validation, and Uncertainty Quantification 3 (2), https://doi.org/10.1115/1.4041490.

Perrier, V., Meyer, F., Granjon, D., 2019. shinyWidgets: Custom Inputs Widgets for Shiny. R package version 0.4.8 (https://CRAN.R-project.org/package=shinyWidgets).

Wickham, H., 2016. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, (http://ggplot2.org).