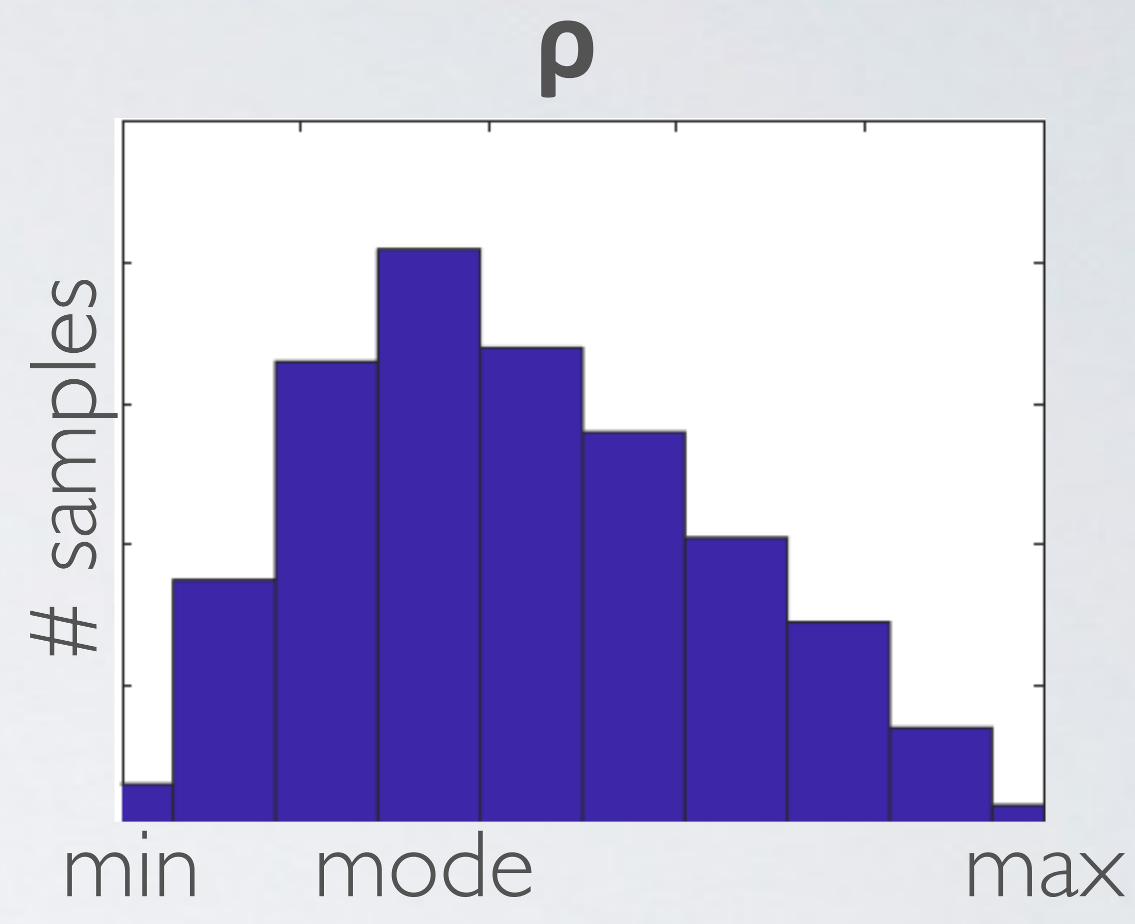
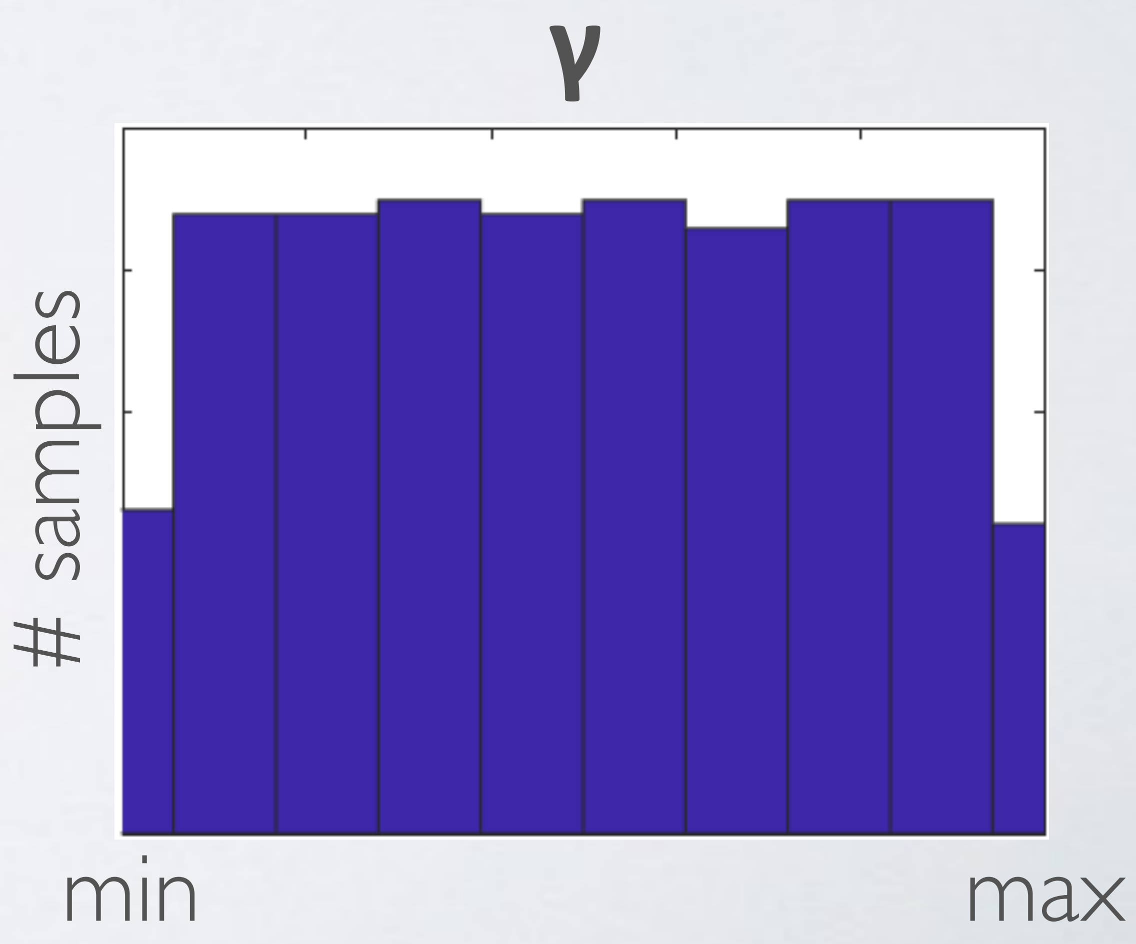
VARIANCE BASED SAMPLING

MODEL SENSITIVITY ANALYSIS

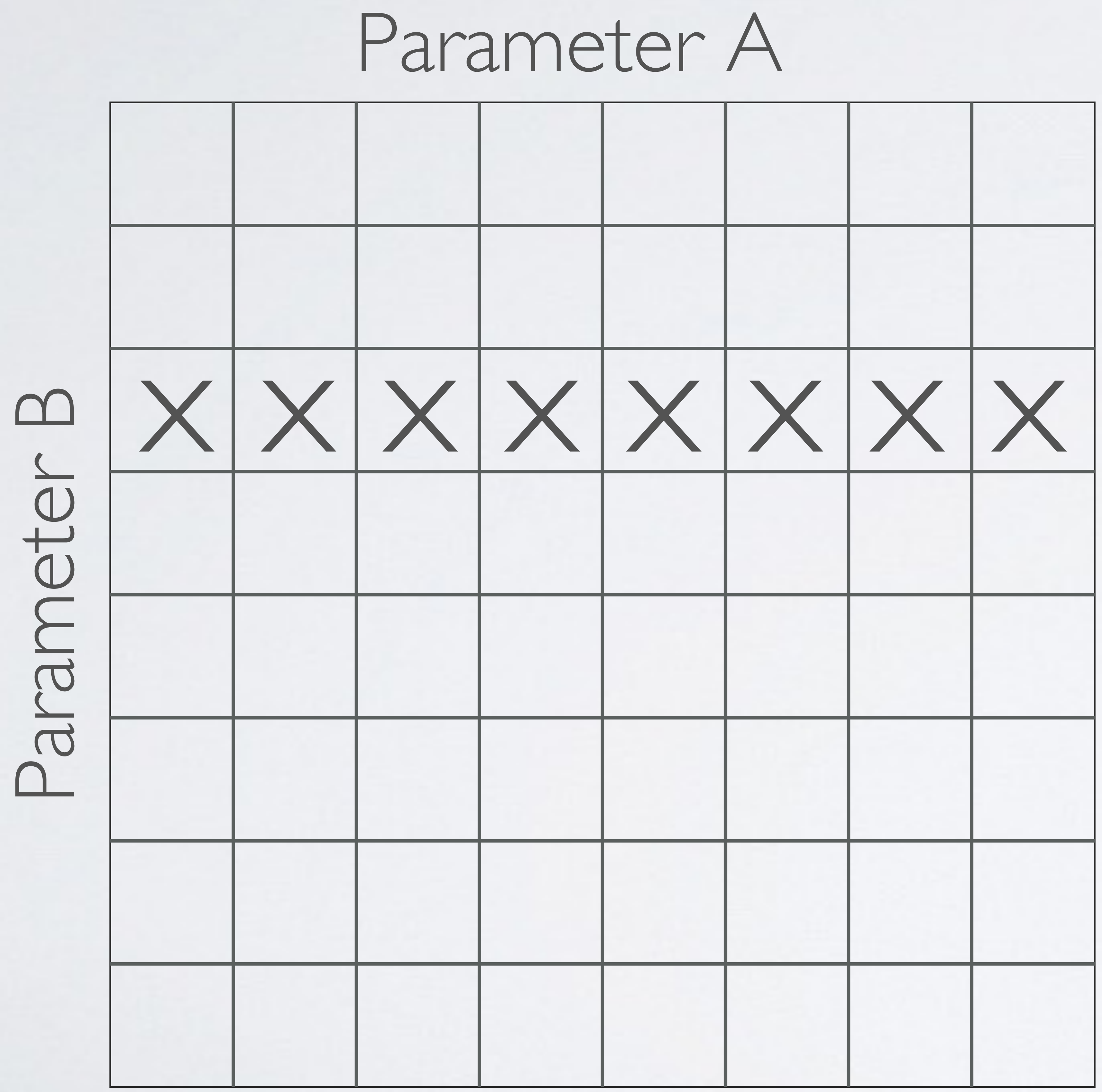
- Many parameters have natural distributions or can be approximated from observations:
  - Normally distributed observations with mean and variance, *if observations are sufficiently dense*
  - A triangle distribution with max, min and mode, *may be used to represent sparser observation data with low, high, and most frequently observed values*
- Model constraints can also help define distributions:
  - e.g., unitless scaling parameters $\in [0, 1]$ sampled uniformly





VARIANCE BASED SAMPLING

ONE-AT-A-TIME APPROACH



Vary one parameter and fix the others

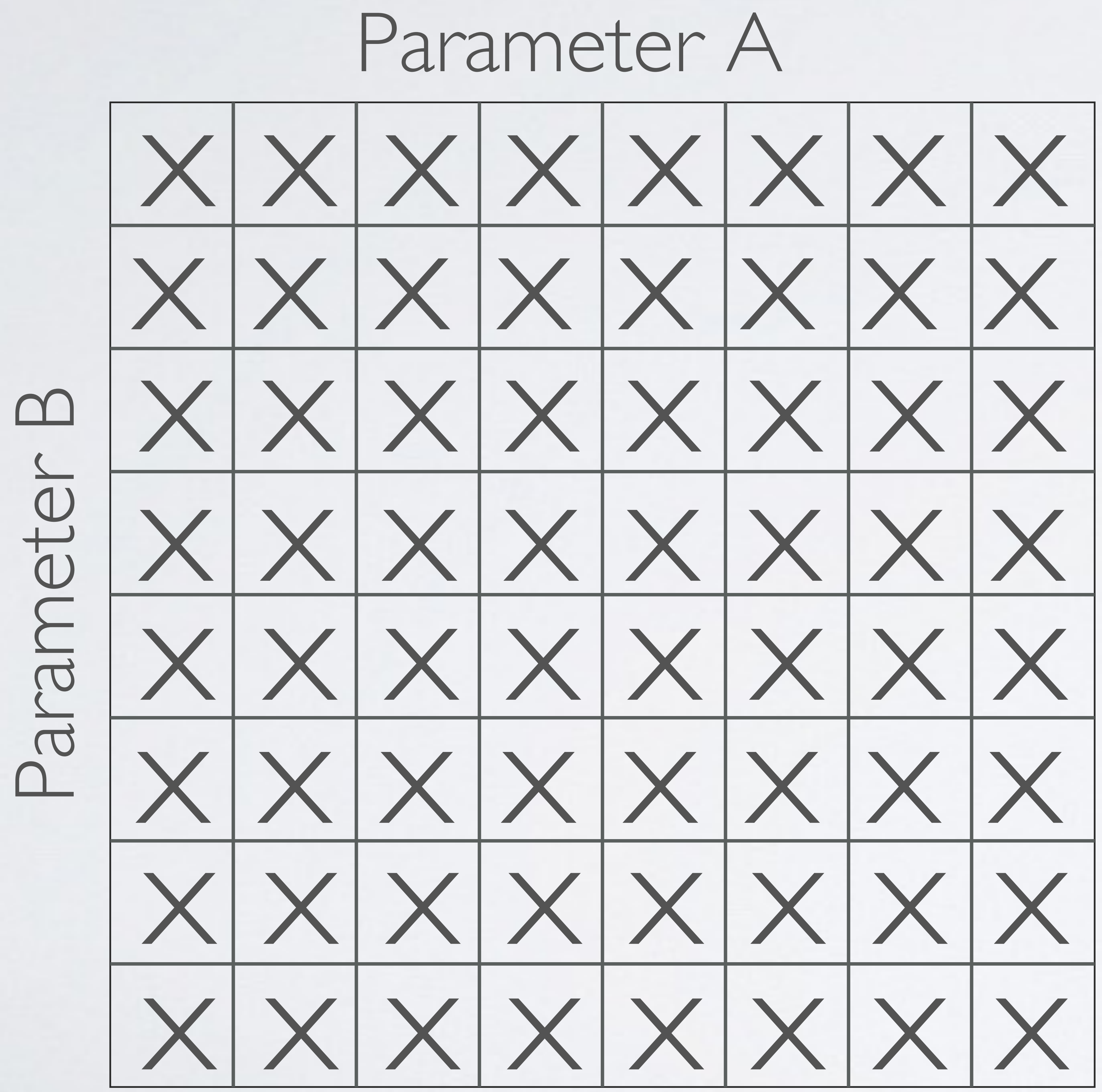
- Allows for seeing how one parameter affects the model
- Doesn't allow for interaction between multiple varied parameters
- Computationally inexpensive:

  M parameters, N samples

  => N simulations

VARIANCE BASED SAMPLING

FULL FACTORIAL APPROACH



Vary all the parameters against each other

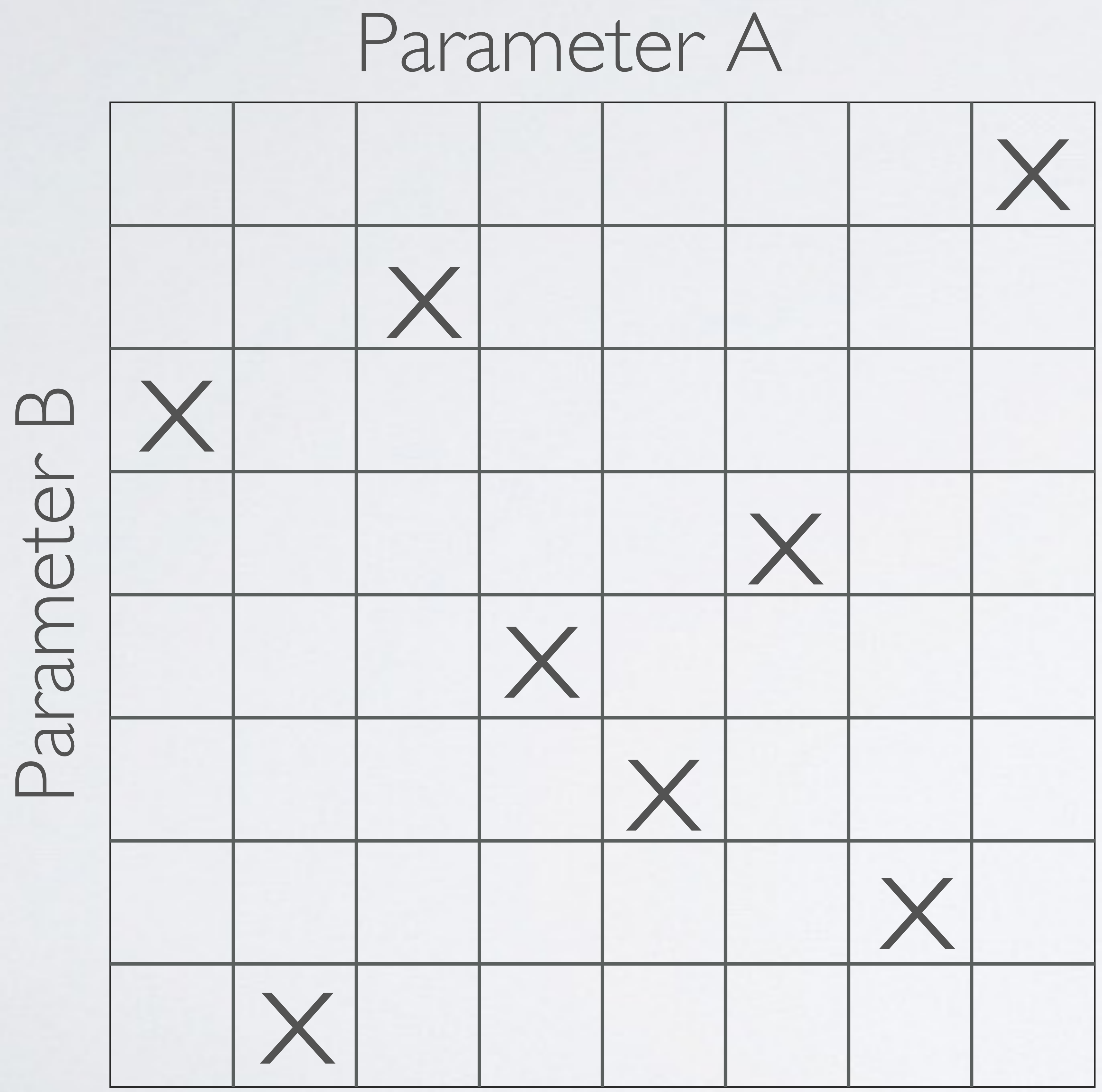
- Allows for seeing the interaction of multiple varied parameters values in the model
- Computationally expensive:

M parameters, N samples

=> M^N simulations

VARIANCE BASED SAMPLING

LATIN HYPERCUBE SAMPLING (LHS) APPROACH



Strategically vary all the parameters

- Allows for seeing the interaction of multiple varied parameters values in the model
- Computationally in expensive:

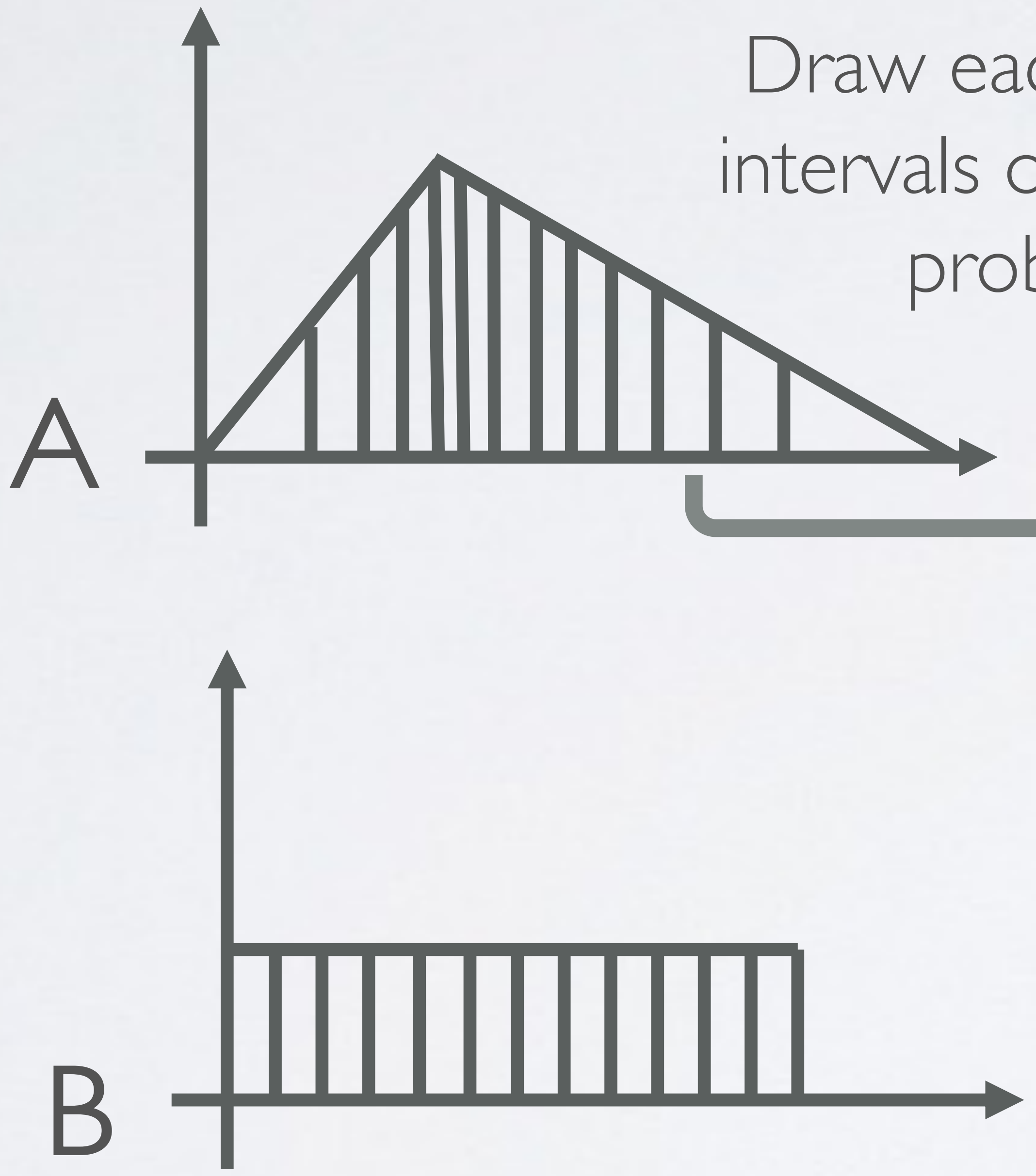
M parameters, N samples

=> N simulations

LATIN HYPERCUBE SAMPLING

STEP 1: DRAW EQUIPROBABLE SAMPLES

Probability Distributions of Observed Parameter values



Draw each of N samples randomly from N intervals of equal area under the parameter's probability density function curve

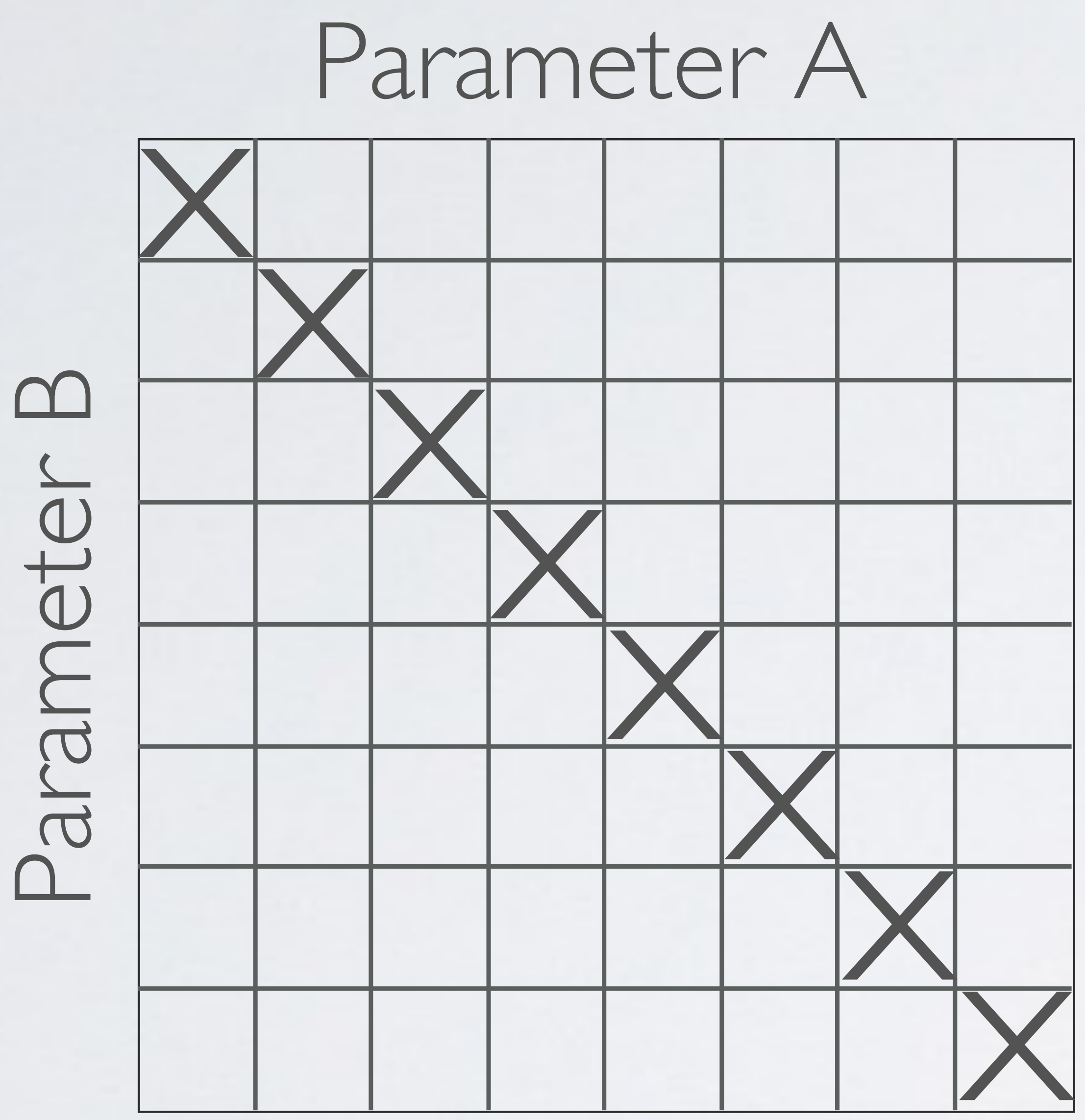
Store the i-th sample as A_i in the parameter matrix

$$\begin{bmatrix} A_1 & B_1 \\ A_2 & B_2 \\ \vdots & \vdots \\ A_{N-1} & B_{N-1} \\ A_N & B_N \end{bmatrix}$$

The number of columns is equal to the number of sampled parameters

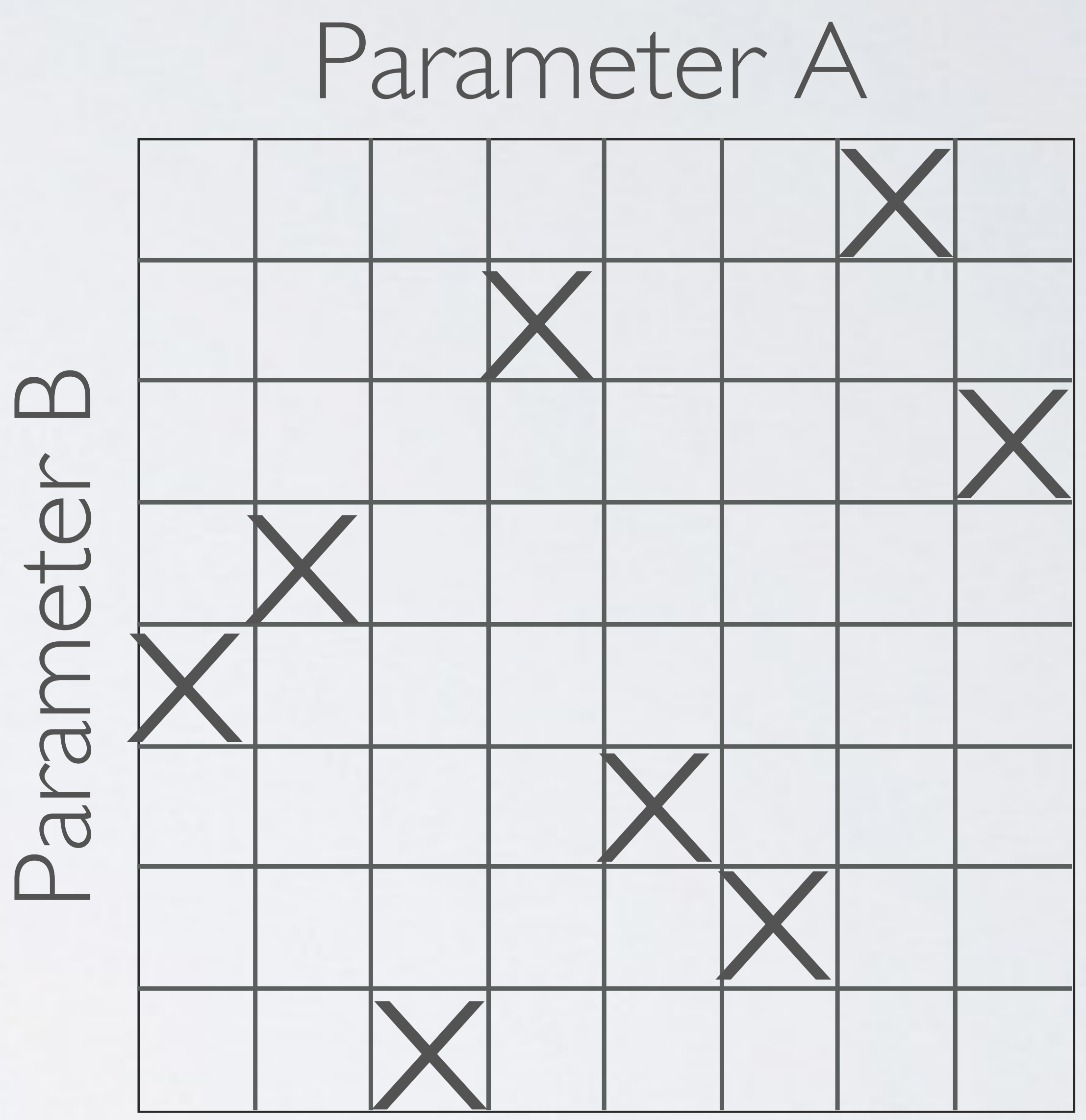
LATIN HYPERCUBE SAMPLING

STEP 2: RANDOMLY PAIR SAMPLES FOR MONTE CARLO SIMULATIONS



Randomly permute the entries in each column

$$\begin{bmatrix} A_1 & B_1 \\ A_2 & B_2 \\ \vdots & \vdots \\ A_{N-1} & B_{N-1} \\ A_N & B_N \end{bmatrix} \longrightarrow \begin{bmatrix} A_6 & B_{N-1} \\ A_4 & B_2 \\ \vdots & \vdots \\ A_1 & B_5 \\ A_N & B_3 \end{bmatrix}$$



Each row j of sample values in the matrix is passed to the model for the j-th Monte Carlo simulation

LHS & PARTIAL RANK CORRELATION

STEP 3: COMPARE SAMPLES WITH OUTPUT FROM THE SIMULATIONS

Add output of interest to matrix with corresponding parameter values

$$\begin{bmatrix} A_6 & B_{N-1} & x_1 \\ A_4 & B_2 & x_2 \\ \vdots & \vdots & \vdots \\ A_1 & B_5 & x_{N-1} \\ A_N & B_3 & x_N \end{bmatrix} \longrightarrow$$

Replace each matrix entry with it's ordinal rank among the column elements

$$\begin{bmatrix} 6 & N-1 & 2 \\ 4 & 2 & N \\ \vdots & \vdots & \vdots \\ 1 & 5 & N-1 \\ N & 3 & 6 \end{bmatrix} \longrightarrow$$

Use the entries $r_{i,j}$ of the rank matrix to compute Pearson correlation coefficients

$$c_{i,j} = \frac{\sum_{n=1}^{N} (r_{n,i} - \mu)(r_{n,j})}{\sqrt{\sum_{n=1}^{N} (r_{n,i} - \mu)^2 \sum_{k=1}^{N} (r_{k,j} - \mu)^2}},$$

$$i, j = 1, 2, ..., M, M+1$$

So in our example, C is a 3x3 matrix

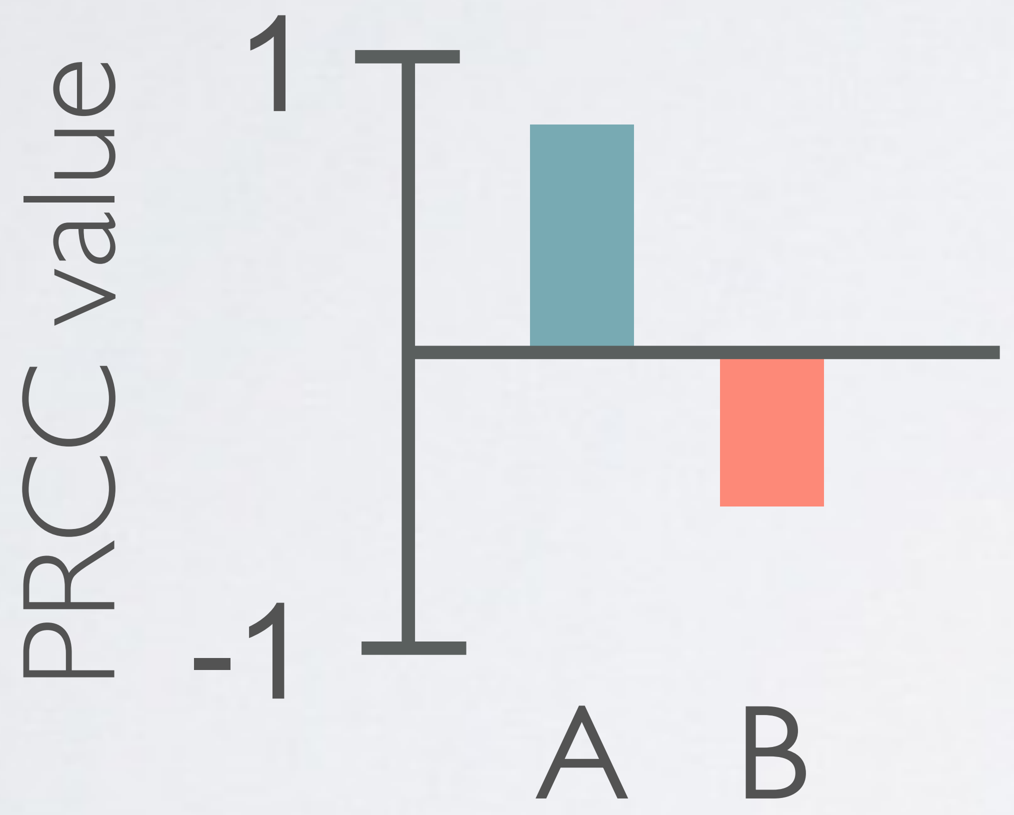
Invert C and use the entries to compute Partial Rank Correlation Coefficients

$$C^{-1} = Q = [q_{i,j}] \longrightarrow \phi_i = \frac{-q_{i,M+1}}{\sqrt{q_{i,i} q_{M+1,M+1}}}$$

LHS & PARTIAL RANK CORRELATION

STEP 4: EXAMINE AND INTERPRET THE RESULTS!

One common visualization is a waterfall plot



Can also examine how correlations with simulation results change over time or space