

Lecture 11 – Sensitivity Analysis

Machine Learning and the Physical World

Nicolas Durrande (Monumo) – 17th of November 2025

Introduction

Why sensitivity analysis?

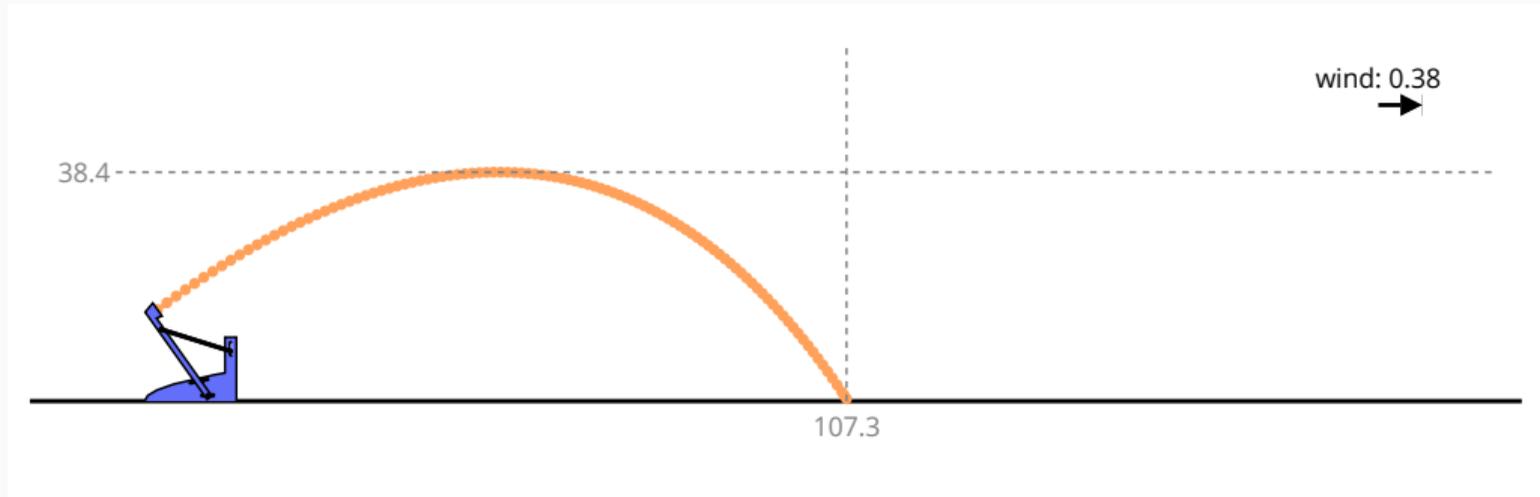
The purpose of sensitivity analysis is to *quantify how much* input variables influence the output of a function/model.

Typical questions include:

- Which variables are the most influential?
- Are there interactions between variables?
- How robust is the output to changes in inputs?
- Where should we reduce uncertainty (better sensors, more data, tighter tolerances)?

Spoiler: surrogate models will be useful to answer such questions!

We'll use a catapult simulator as a running example:

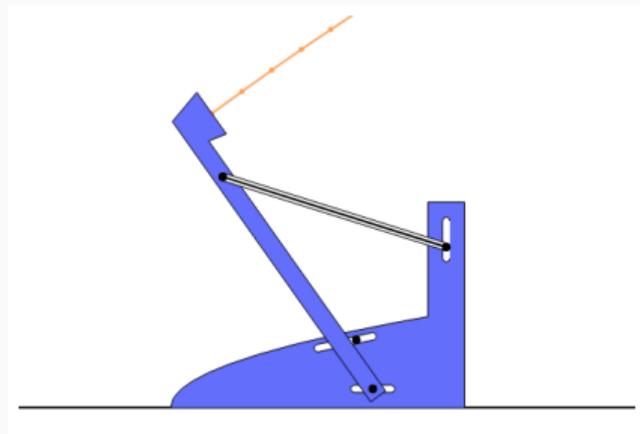


Inputs: catapult settings in $[0, 1]^5$.

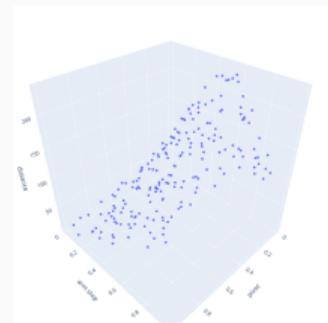
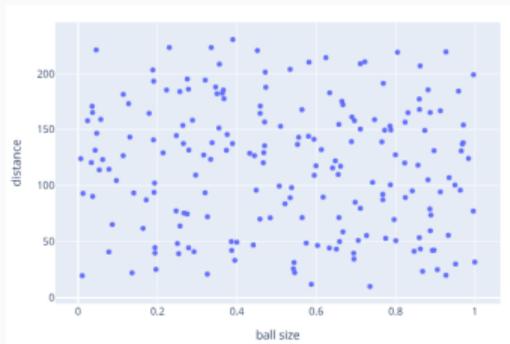
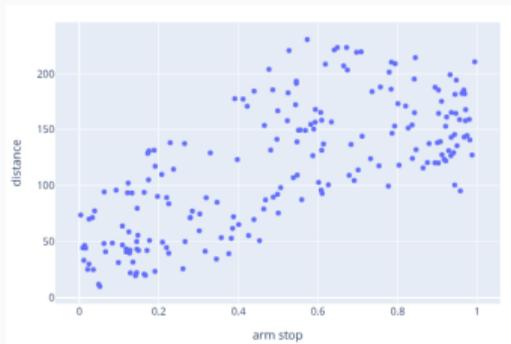
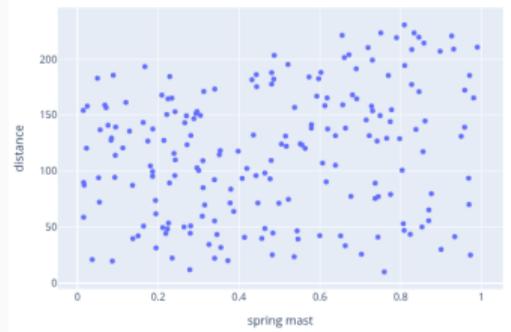
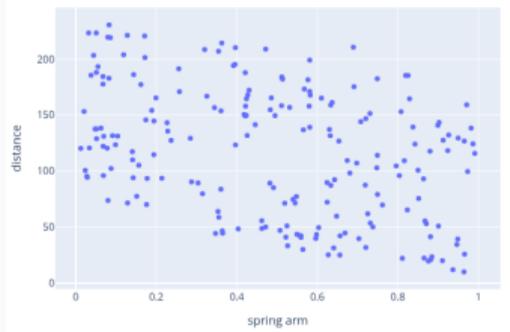
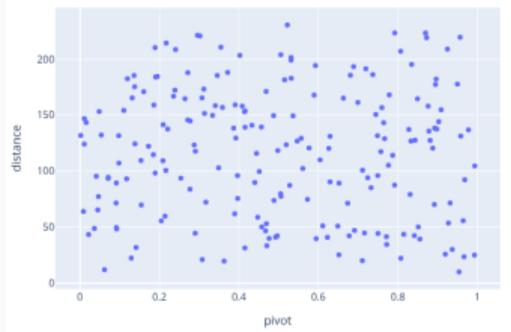
Output: distance reached by the projectile.

In more detail, the inputs are:

- x_1 spring attachment point on the arm
- x_2 arm pivot position
- x_3 spring attachment point on the mast
- x_4 arm stop position
- x_5 projectile size.



To get insights into how inputs influence the output, one can plot the input/output relationship for some random designs:



It's already possible to see some trends, but it's hard to be quantitative and to capture interactions above the first order...

Mathematically speaking, we write the function of interest as

$$f : \mathbb{R}^d \rightarrow \mathbb{R}, \quad y = f(x_1, \dots, x_d).$$

and assume it is deterministic.

Two kinds of sensitivity analysis can be distinguished:

- **Local sensitivity analysis:** effect of small perturbations around a nominal input value.
- **Global sensitivity analysis:** effect of input variations over their entire range.

Furthermore, one can consider model-free and model-based approaches.

Local sensitivity analysis

Local derivatives as sensitivity measures

Fix a nominal input $x^{(0)} = (x_1^{(0)}, \dots, x_d^{(0)})$ (design point, operating point).

The **partial derivative**

$$S_i^{\text{loc}} = \frac{\partial f}{\partial x_i}(x^{(0)})$$

quantifies the change in f when only x_i is perturbed.

Example (Catapult Simulator)

If we choose $x^{(0)} = (0.5, 0.5, 0.5, 0.5, 0.5)$ as the nominal point, we get

input variable	x_1 spring arm	x_2 pivot	x_3 spring mast	x_4 arm stop	x_5 projectile size
S_i^{loc}	-5.4	-209.7	57.1	256.9	0.0

Derivatives depend on units and scales. We sometimes prefer dimensionless indices.

A common choice is

$$S_{i,rel}^{loc}(x^{(0)}) = \frac{\partial f}{\partial x_i}(x^{(0)}) \cdot \frac{x_i^{(0)}}{f(x^{(0)})},$$

which measures the *relative* change in f for a relative change in x_i .

In our catapult example all inputs and outputs are dimensionless, so this is less relevant. (It would however make sense if, e.g. some inputs were distances and other angles...)

Several options:

- **Analytic derivatives:** if f is given in closed form, differentiate symbolically.
- **Finite differences** (black-box model):

$$\frac{\partial f}{\partial x_j}(x^{(0)}) \approx \frac{f(x^{(0)} + he_j) - f(x^{(0)})}{h}$$

- **Automatic differentiation / backprop:**
 - With automatic differentiation, gradients come for “free” (same order of cost as forward pass).

Interpretation as Taylor expansion

Since the local sensitivity is based on derivatives, it can be interpreted via a first-order Taylor expansion around $x^{(0)}$:

$$f(x) \approx f(x^{(0)}) + \sum_{i=1}^d \frac{\partial f}{\partial x_i}(x^{(0)})(x_i - x_i^{(0)}).$$

which makes it clear that

- only linear effects are captured,
- interactions between variables are ignored.

Note: This view also gives a recipe on how to include higher-order effects by using e.g. the Hessian!

Consider a minimisation problem, and let x^* be a local optimum of f :

$$\nabla f(x^*) = 0.$$

For small perturbations δ around x^* , a 2nd-order Taylor expansion gives

$$f(x^* + \delta) \approx f(x^*) + \frac{1}{2} \delta^\top H(x^*) \delta,$$

where $H(x^*)$ is the Hessian.

- Local robustness is now governed by the curvature matrix $H(x^*)$.
- Large curvature $\Rightarrow f(x^*)$ is sensitive to small perturbations of x .
- “Flat” minima (small curvature) are more robust.

Global sensitivity analysis

Global sensitivity analysis aims at capturing the influence of the inputs over their entire range of variation. In other words, it relates the "variability" in inputs to the "variability" in the output.

This requires a **probabilistic model** for the inputs:

$$X = (X_1, \dots, X_d) \sim p_X,$$

which can be propagated through f to induce variability in Y .

$$Y = f(X).$$

In the following, we assume that X_1, \dots, X_d are independent, i.e. $p_X = \prod_{i=1}^d p_{X_i}$.

The **variance** is typically used as a measure of global variability:

$$\text{var}(Y) = \text{E} [(Y - \text{E}[Y])^2] .$$

It is built on the intuition:

- “Important” inputs should explain a large fraction of $\text{var}(Y)$.
- Interactions between inputs should appear as extra variance beyond main effects.

The law of total variance gives interesting insights here:

$$\text{var}(Y) = \text{E} [\text{var}(Y | X_j)] + \text{var} (\text{E}[Y | X_j]).$$

The second term quantifies how much of $\text{var}(Y)$ is explained by X_j alone. To make this precise, we need the functional ANOVA decomposition.

Functional ANOVA decomposition

Assume X_1, \dots, X_d are independent and $Y = f(X)$ is square-integrable. The Hoeffding–Sobol expansion (or FANOVA) is the unique decomposition

$$f(x) = f_0 + \sum_i f_i(x_i) + \sum_{i < j} f_{ij}(x_i, x_j) + \dots + f_{1\dots d}(x_1, \dots, x_d),$$

where all the terms are orthogonal in L^2 . That is, for all $u, v \subset \{1, \dots, d\}$:

$$\int f_u(x_u) f_v(x_v) p_X(x) dx = 0 \quad \text{if } u \neq v.$$

In particular, orthogonality with respect to the constant function implies that

$$\int f_u(x_u) p_X(x) dx = 0 \quad \text{if } u \neq \emptyset.$$

For a 2D input space $[0, 1]^2$ with a uniform measure we get:

$$f(x_1, x_2) = f_0 + f_1(x_1) + f_2(x_2) + f_{12}(x_1, x_2)$$

and the terms are defined as

$$f_0 = \int f(x) dx_1 dx_2,$$

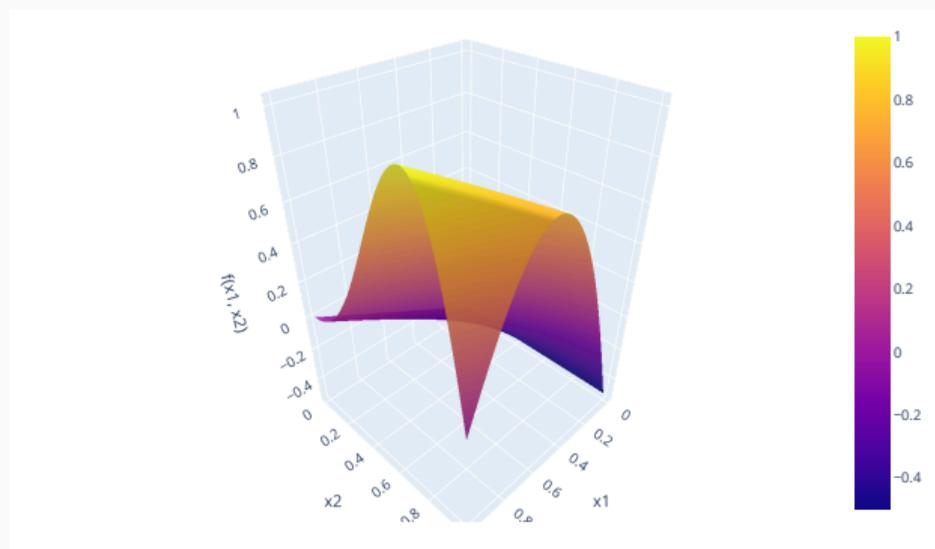
$$f_1(x_1) = \int f(x) dx_2 - f_0,$$

$$f_2(x_2) = \int f(x) dx_1 - f_0,$$

$$f_{12}(x_1, x_2) = f(x) - f_0 - f_1(x_1) - f_2(x_2),$$

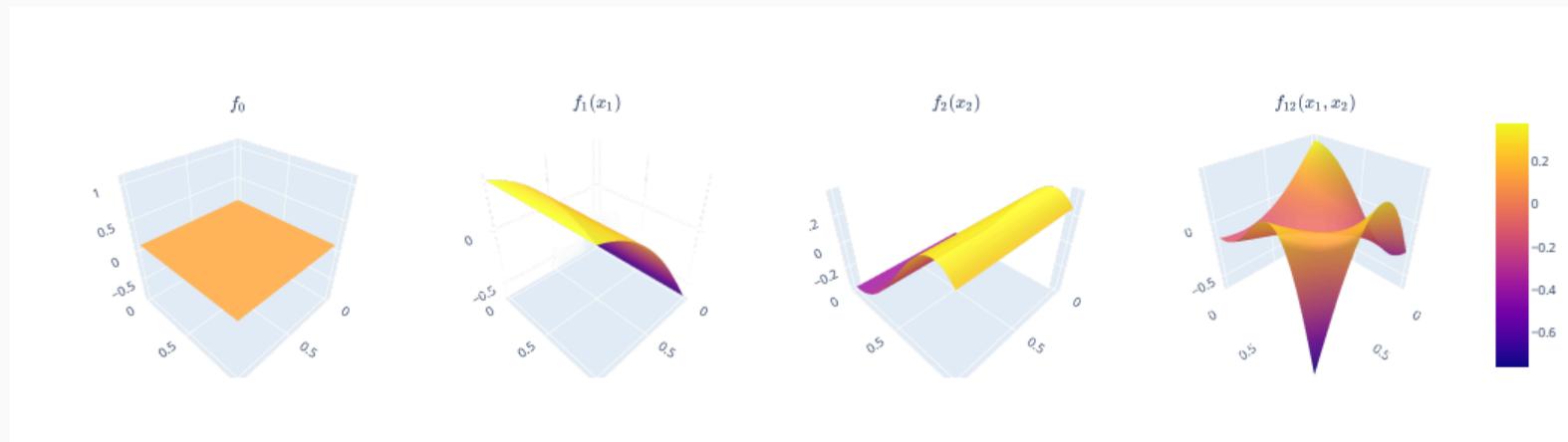
Example (1/2)

Let's consider the toy test function $f(x_1, x_2) = 0.5x_1 + \sin(\pi x_1 x_2^2) - 0.5$



Example (2/2)

We get the following FANOVA decomposition:



Variance decomposition

The L^2 orthogonality of the FANOVA terms implies that they are uncorrelated random variables when evaluated at X :

$$\text{cov}(f_u(X_u), f_v(X_v)) = \mathbb{E}[f_u(X_u)f_v(X_v)] - \mathbb{E}[f_u(X_u)]\mathbb{E}[f_v(X_v)] = 0 \quad \text{if } u \neq v.$$

Applying $\text{var}(\cdot)$ to the FANOVA expansion gives

$$\text{var}(Y) = \sum_i V_i + \sum_{i < j} V_{ij} + \cdots + V_{1\dots d},$$

where $V_u = \text{var}(f_u(X_u)) \geq 0$ for each subset $u \subseteq \{1, \dots, d\}$.

Interpretation:

V_i : variance contribution from X_i acting alone.

V_{ij} : contribution from interaction between X_i and X_j .

...

Sobol' sensitivity indices normalise each variance component by $\text{var}(Y)$.

- First-order (main-effect) index:

$$S_i = \frac{V_i}{\text{var}(Y)} = \frac{\text{var}(\mathbb{E}[Y | X_i])}{\text{var}(Y)}.$$

- Second-order index:

$$S_{ij} = \frac{V_{ij}}{\text{var}(Y)}.$$

- Higher-order indices $S_u = V_u / \text{var}(Y)$ for any subset u .

By constructions, we have $S_u \in [0, 1]$ and the indices sum to one!

Main-effect indices S_i ignore interactions. To capture “everything involving X_i ”, define the **total-effect** index:

$$S_{T_i} = \sum_{u \ni i} S_u = 1 - \frac{\text{var}(\mathbf{E}[Y \mid X_{-i}])}{\text{var}(Y)},$$

where X_{-i} denotes all variables except X_i .

- S_{T_i} captures the variance that would be lost if X_i were fixed.
- The gap $S_{T_i} - S_i$ quantifies how much of X_i 's influence comes from interactions.

In general, the model f is a black box and the integrals defining $\mathbb{E}[Y]$, $\text{var}(Y)$, S_i , etc. cannot be computed analytically.

- Sample $X^{(1)}, \dots, X^{(N)} \sim p_X$.
- Evaluate $Y^{(n)} = f(X^{(n)})$.
- Approximate expectations/variances by empirical averages:

$$\mathbb{E}[Y] \approx \frac{1}{N} \sum_{n=1}^N Y^{(n)}, \quad \text{var}(Y) \approx \frac{1}{N-1} \sum_{n=1}^N (Y^{(n)} - \bar{Y})^2.$$

More structured sampling schemes are required for Sobol' indices since we want to estimate conditional expectations.

Monte Carlo estimation of Sobol' indices (Saltelli scheme)

One popular strategy (Saltelli):

1. Draw two independent sample matrices with rows sampled from p_X .

$$A, B \in \mathbb{R}^{N \times d}$$

2. For each i , build a mixed matrix $A_B^{(i)}$ by taking all columns from A except the i th, which comes from B .
3. Evaluate the quantities $f(A)$, $f(B)$, $f(A_B^{(1)})$, \dots , $f(A_B^{(d)})$.

Then, for example, a first-order index can be estimated as

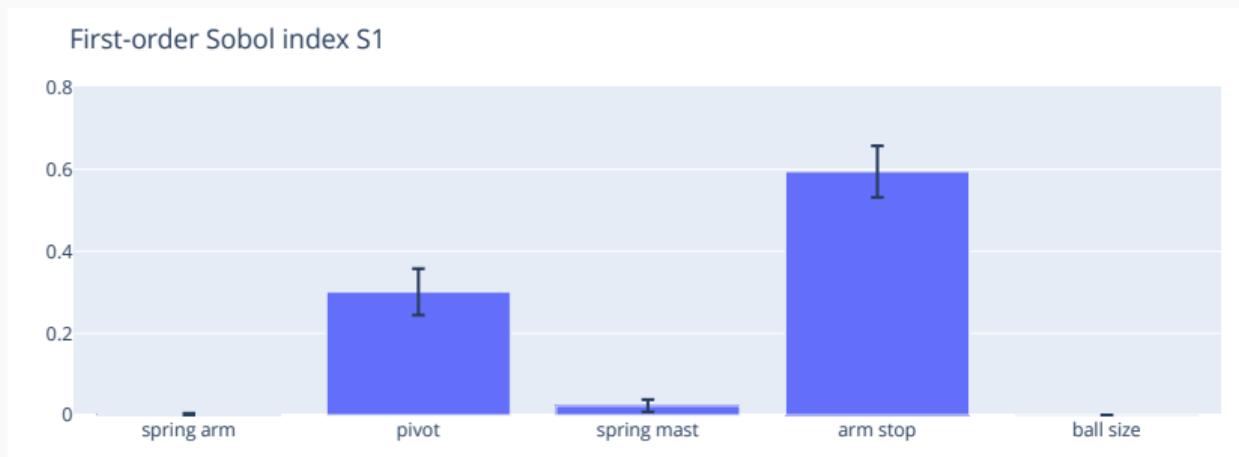
$$S_i \approx \frac{\frac{1}{N} \sum_{n=1}^N f(B_n) (f(A_{B,n}^{(i)}) - f(A_n))}{\text{var}(Y)},$$

where $\text{var}(Y)$ is estimated from $f(A)$ and $f(B)$.

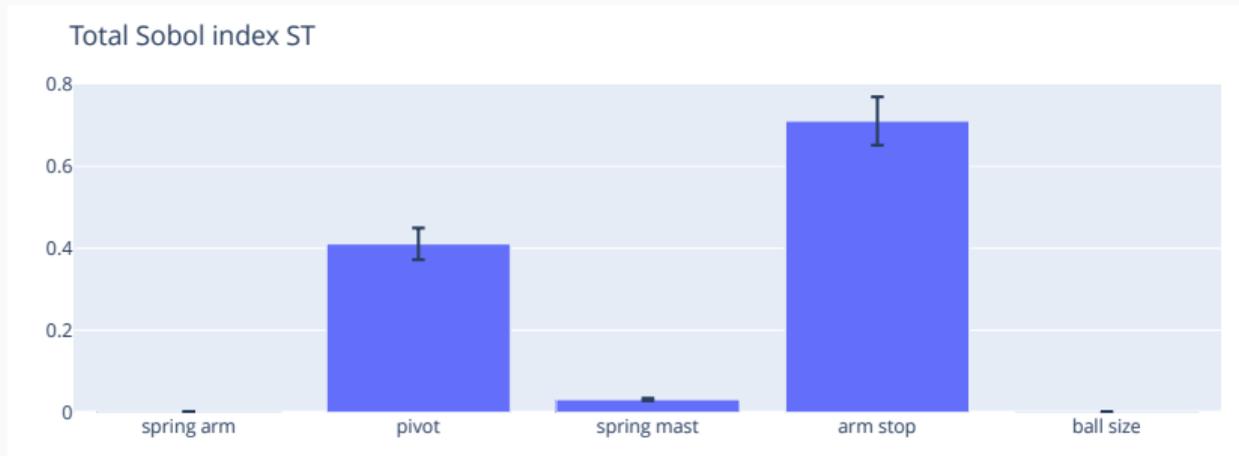
The **SALib** Python library implements various sampling and analysis methods.

Example

For the catapult, using $N \approx 1000$ samples and the Saltelli scheme, we get:

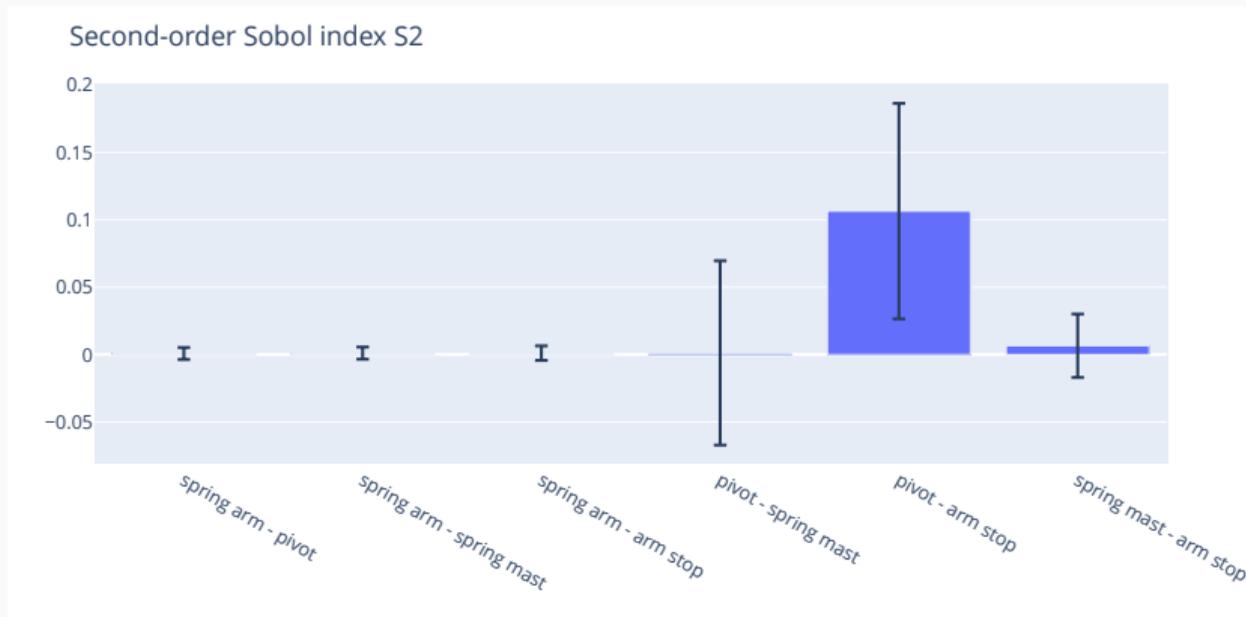


To further assess the importance of each variable, we can also look at the total-effect indices.



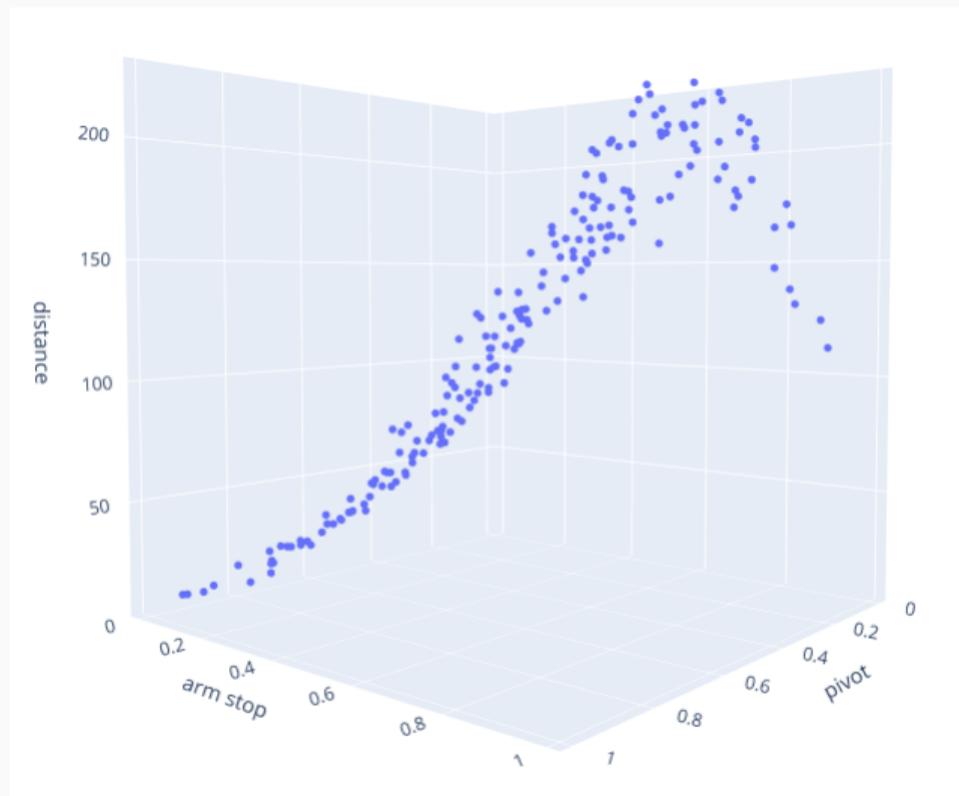
This tells us that x_2 (pivot) and x_4 (arm stop) have some higher order interactions. It also tells us that we can discard x_5 (projectile size) from the analysis without any loss!

Finally, we can look at the second-order indices.



In the end, we found two dominant variables (x_2 and x_4) with some interactions between them.

3D scatter plot in the (x_2, x_4) space:



Model-based sensitivity analysis

Why model-based sensitivity analysis?

Full Monte Carlo global SA may be too expensive when:

- each evaluation of f is costly (e.g. CFD, FEA, multi-physics solvers),
- or we want to explore many different sensitivity measures.

Idea:

- Evaluate f on a carefully chosen **design of experiments**:

$$\{(x^{(n)}, y^{(n)})\}_{n=1}^N, \quad y^{(n)} = f(x^{(n)}),$$

with N as small as possible.

- Fit a **surrogate model** \hat{f} .
- Perform sensitivity analysis on \hat{f} (cheap to evaluate).

Assume we have a linear model:

$$y(x) = \beta_0 + \sum_{i=1}^d \beta_i x_i,$$

and, as previously, that the distributions of all X_i are independent.

Then, the Sobol' indices are simply

$$S_i = \frac{\beta_i^2 \text{var}(X_i)}{\text{var}(Y)}.$$

Gaussian process regression is a flexible, probabilistic surrogate:

- Place a GP prior on f :

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot)),$$

where m is a mean function and k a covariance kernel.

- Condition on observed data $(x^{(n)}, y^{(n)})$ to obtain a posterior distribution:

$$f(\cdot) \mid \mathcal{D} \sim \mathcal{GP}(\mu(\cdot), c(\cdot, \cdot)).$$

- μ is the posterior mean; c is the posterior covariance function.

Pros:

- Non-parametric model.
- Comes with built-in uncertainty quantification.

Once a GP surrogate is fitted, the Sobol' indices can be obtained in various ways:

1. Plug-in Monte Carlo:

- Replace f by $\mu(x)$ (GP mean).
- Use standard Monte Carlo (e.g. Saltelli scheme) with $\mu(x)$ instead of $f(x)$.
- Very cheap: GP predictions are fast.

2. Fully Bayesian:

- Treat f as random with GP posterior.
- Sample functions from the GP and compute Sobol' indices for each.
- This gives a *distribution* over Sobol' indices, reflecting surrogate uncertainty.

3. Kernel enthusiast:

- Choose the kernel wisely...
- ... and get all the Sobol' indices analytically!

We've seen that the FANOVA decomposition was at the core of variance-based sensitivity analysis.

If the kernel k can be decomposed as

$$k(x, x') = \prod_{i=1}^d (1 + k_i(x_i, x'_i)),$$

where each k_i is a 1D kernel that integrate to zero for any of its inputs, then the GP can be decomposed as

$$f(x) = f_0 + \sum_i f_i(x_i) + \sum_{i < j} f_{ij}(x_i, x_j) + \cdots + f_{1\dots d}(x_1, \dots, x_d),$$

with f_u orthogonal in L^2 !

To get kernels with the desired properties, we can start from any GP f_i over \mathbb{R} with kernel k_i^{base} and compute the distribution of:

$$f_i^{\text{base}} \left| \int f_i^{\text{base}}(x) dx_i = 0. \right.$$

This yields the posterior covariance

$$k_i(x_i, x'_i) = k_i^{\text{base}}(x_i, x'_i) - \frac{\int k_i^{\text{base}}(x_i, s) ds \int k_i^{\text{base}}(s', x'_i) ds'}{\iint k_i^{\text{base}}(s, s') ds ds'}$$

which by construction generates samples that integrate to zero!

Given some data, the main effects and interactions can be inferred from the GP posterior:

$$f_u \mid \mathcal{D} \sim \mathcal{GP}(\mu_u(\cdot), c_u(\cdot, \cdot)),$$

with

$$\begin{aligned}\mu_u(x_u) &= k_u(x_u, x_u) K^{-1} y, \\ c_u(x_u, x'_u) &= k_u(x_u, x'_u) - k_u(x_u, x_u) K^{-1} k_u(x_u, x'_u),\end{aligned}$$

where $k_u = \prod_{i \in u} k_i$. The expected variance components are then

$$V_u = y^\top K^{-1} \left(\int k_u(x_u, x_u) k_u(x_u, x_u) dx_u \right) K^{-1} y.$$

from which the Sobol' indices follow immediately.

Catapult Example

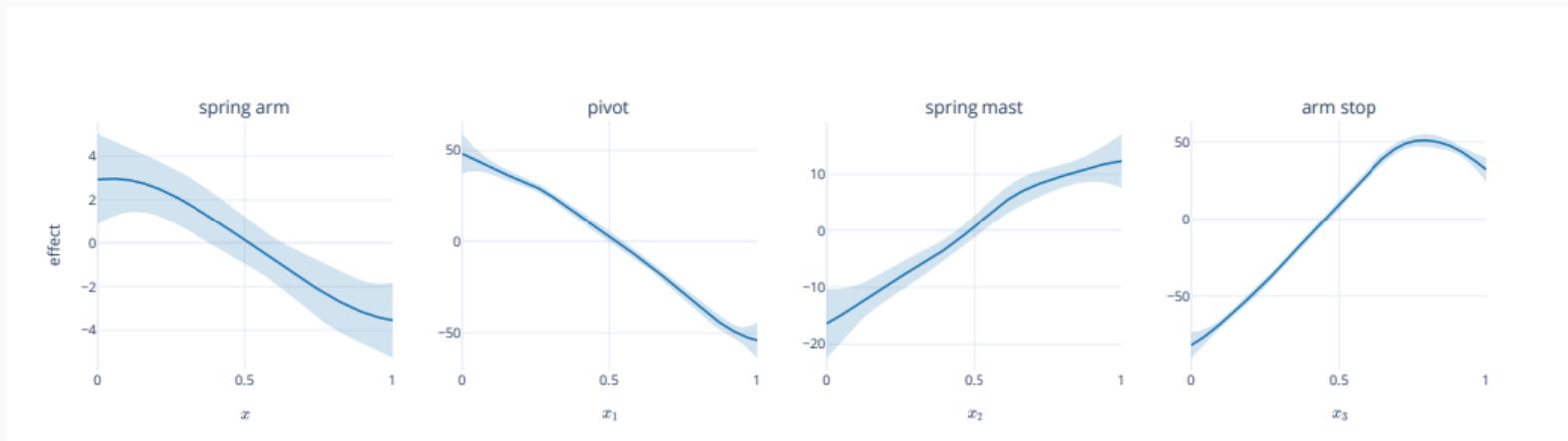
If we derive the zero-integral kernel from a Matérn 5/2 base kernel, and train a model on 50 datapoints (!) we get the following main effects:



(shared y-axis for better comparison)

Catapult Example

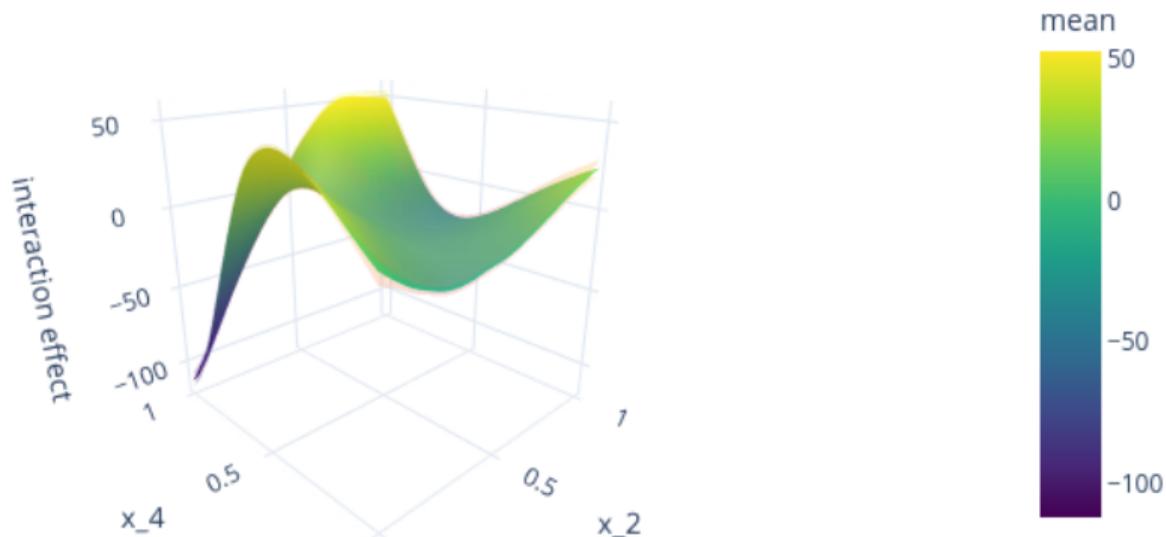
If we derive the zero-integral kernel from a Matérn 5/2 base kernel, and train a model on 50 datapoints (!) we get the following main effects:



(individual y-axis for better visibility)

Example

We can also plot the first interaction order



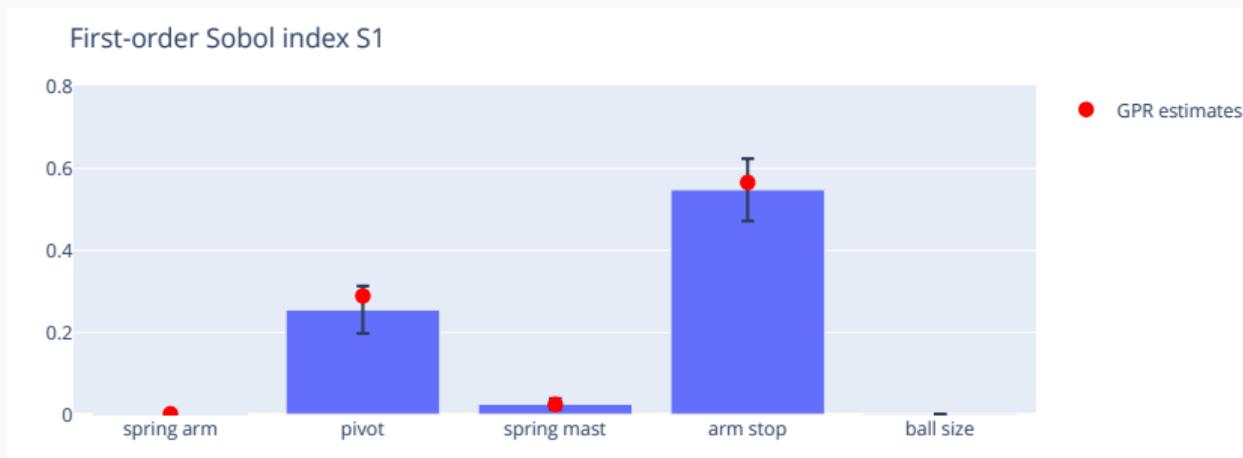
Example

Using the analytical formula for the predicted mean variance components, we get

input variable	x_1 spring arm	x_2 pivot	x_3 spring mast	x_4 arm stop
V_j	5.0	1007.1	82.7	1973.0
S_j	0.001	0.289	0.024	0.566

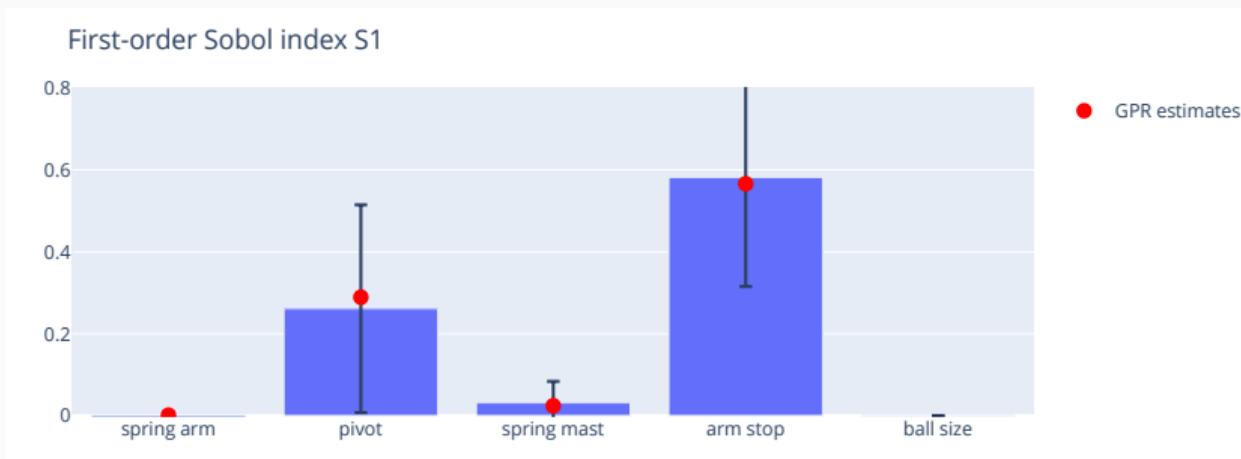
Example

We successfully recovered the same Sobol' indices as with Monte Carlo using 50 simulation evaluations instead of 1000!



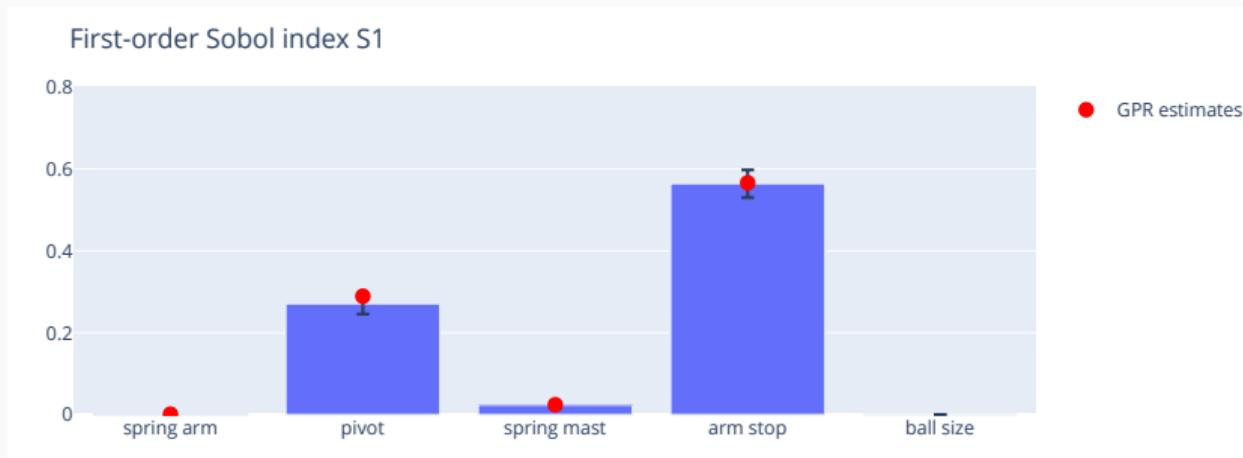
Example

As a comparison:
MC analysis with a budget of 64 simulation



Example

As a comparison:
MC analysis with a budget of 4096 simulation



Wrap up

- **Local sensitivity** (i.e. gradients!):
 - Simple, cheap, and useful near a nominal point.
 - Miss nonlinearities and interactions. (or not...)
- **Global sensitivity**:
 - Sobol' indices decompose $\text{var}(Y)$ into main and interaction effects.
 - Monte Carlo methods are flexible but can be expensive.
- **Model-based SA**:
 - Linear models link regression coefficients to Sobol' indices in simple settings.
 - GPs and other surrogates enable global SA when direct Monte Carlo is too costly.