## Assignment 2

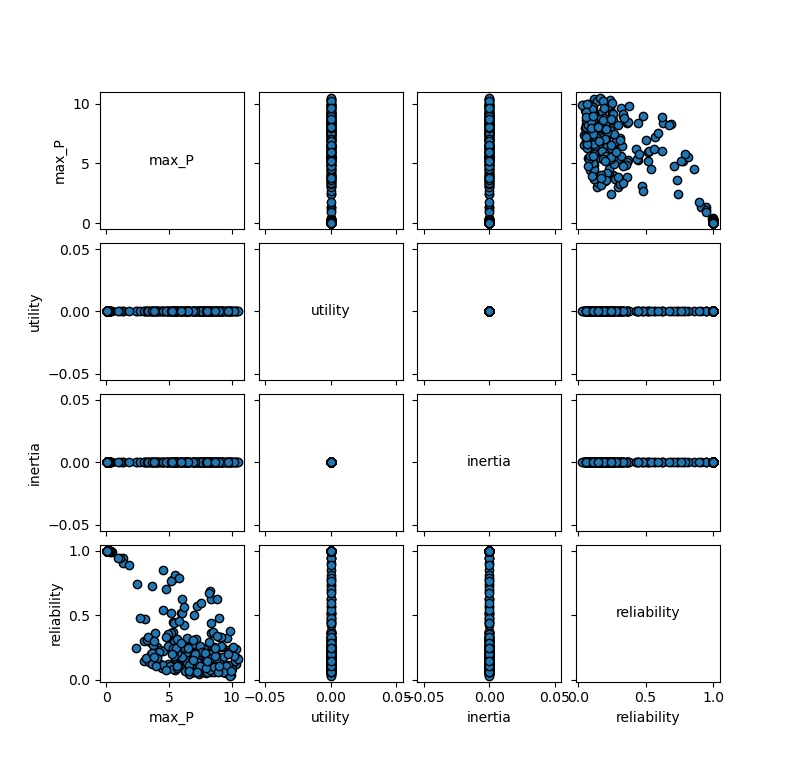
### Part 1

The code we made and used is in the .ipynb file

### Part 2

Plot with default sampling, assume absence of any release: how can we implement this assumption into our code? We thought of maybe setting the lever (‘decisions’) to zero, but this gives error messages, as a lever that cannot be adjusted is no lever I assume?

🡪 model.levers = [RealParameter('decisions', 0, 0.1)] could maybe be changed but we don’t know what to change it into



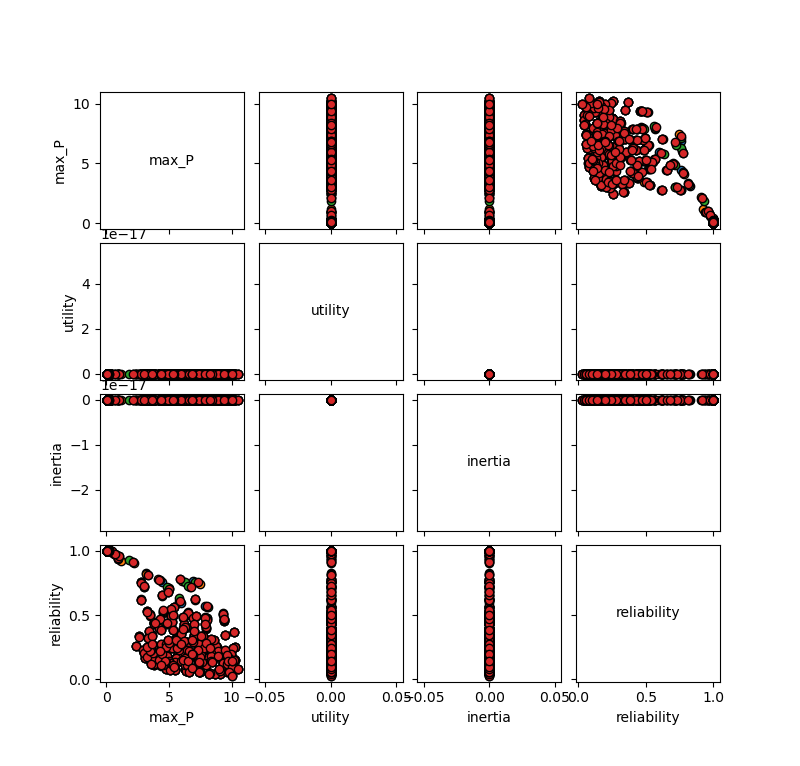
The only trade-off that can be seen in this graph is between reliability and max\_P.

🡪 How should we visually identify the uncertainties that drive system behaviour?

### Part 3

(scenarios=1000, policies=4)

What does the colour mean in the plots (red, orange, green)? And also in part 2 (the blue colour)?



### Part 4

**Difference MultiprocessingEvaluator and IpyparallelEvaluator**

Ipyparallel combines the EMA workbench with the IPyton parallel package and Multiprocessing supports the usage of the intern multiprocessing library.

Ipyparallel can distribute work across many machines, which multiprocessing cannot. Ipyparallel has some advantages over Multiprocessing, but if the code is not that lengthy it is fine to use the Multiprocessing tool.

**Runtimes**

Multiprocessing runtime in minutes = 1.006902559598287 (1000 scenarios, 4 policies, n\_processes=7)

Sequential runtime in minutes = 2.7093281070391337 (1000 scenarios, 4 policies)

