## EPA1361 Group 3 - Assignment 2

### Part 1

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

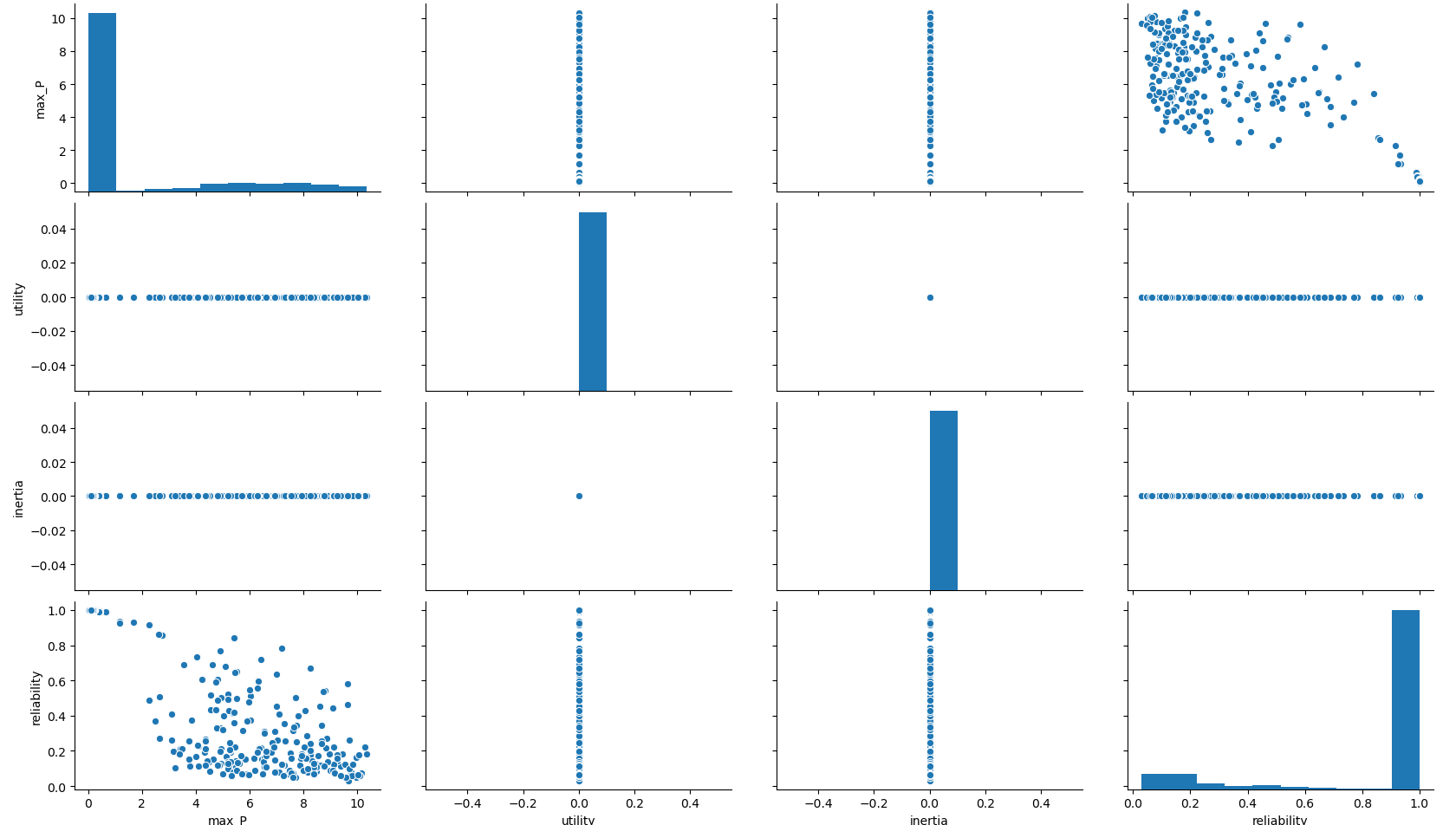
### Part 2

lake\_model\_function\_Sequential.py was used, but then the following was put when evaluating:

with SequentialEvaluator(lake\_model) as evaluator:

results = evaluator.perform\_experiments(n\_scenarios, policy, levers\_sampling=MC)

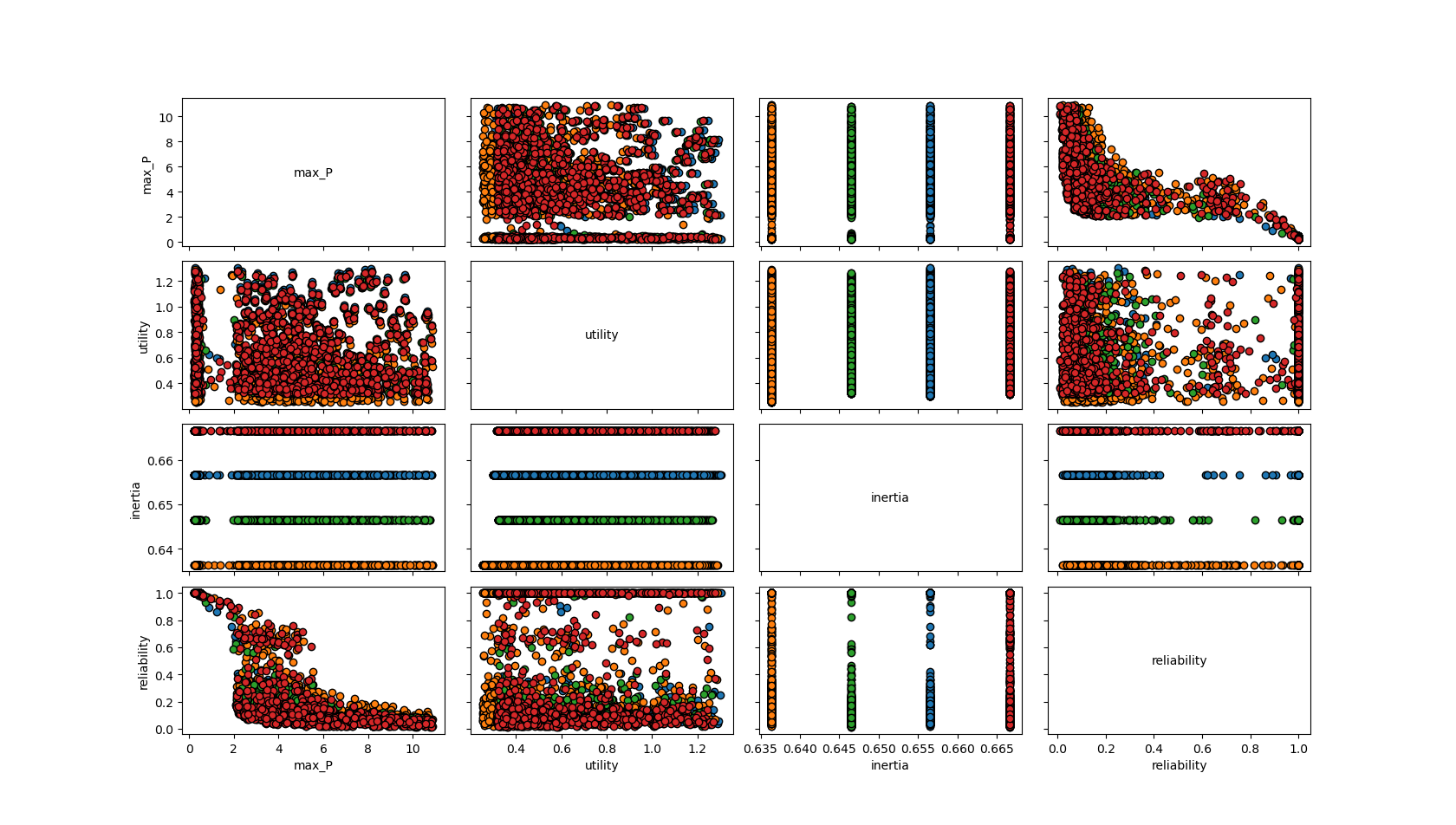
🡪 How should we visually identify the uncertainties that drive system behaviour?

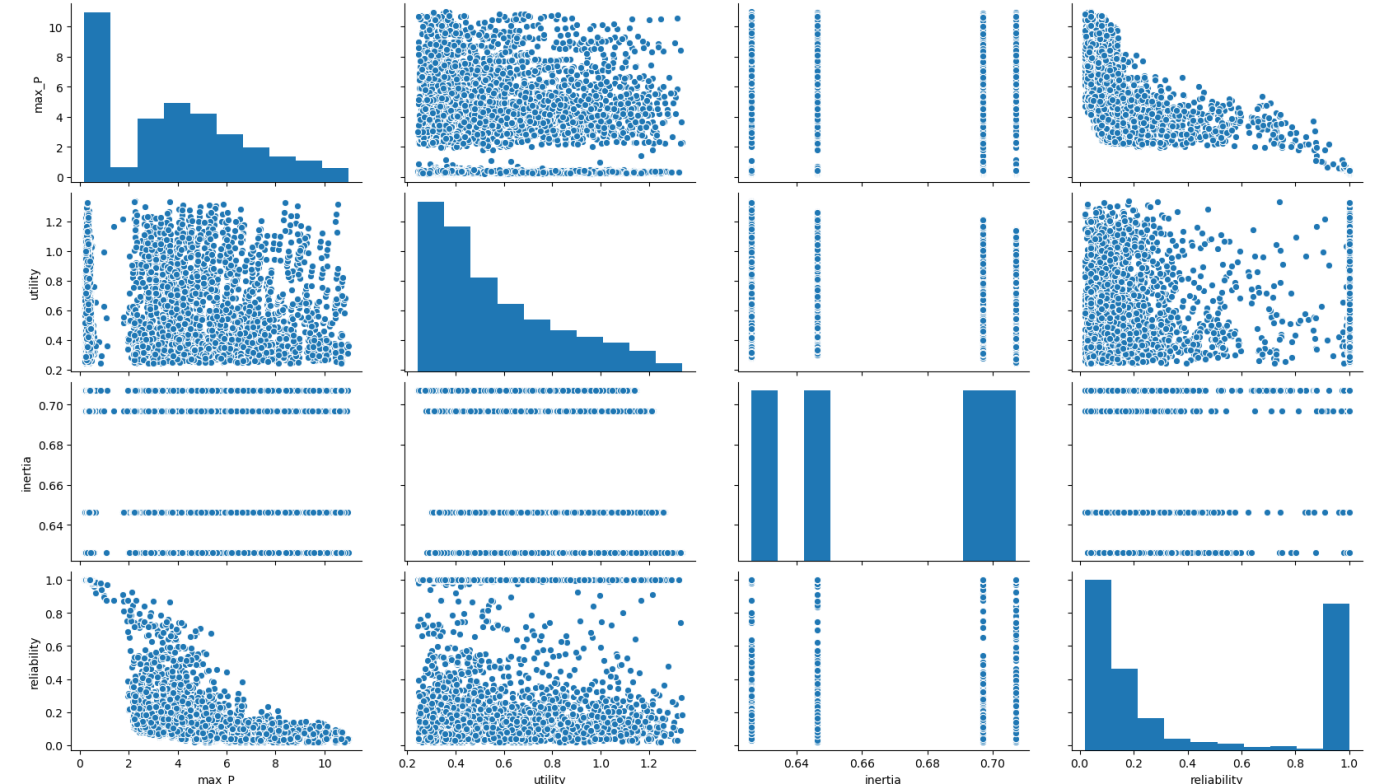


### Part 3

lake\_model\_function\_Sequential.py was used

(scenarios=1000, policies=4)





### Part 4

wk12\_lake\_model\_solo.py & wk12\_assignment2\_code.py (put them both in the same folder and run the wk12\_assignment2\_code.py file).

**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.

**Runtime differences**

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

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

