Using ML to optimize a Group Sequential Design

Tom Parke
Berry Consultants
ADMTP Regensburg 2025
With Thanks to: Dr Matthew Darlington*, Dr Luke Rhodes-Leader* and Dr Peter Jacko (*Lancaster University)

BERRY

Introduction

- As trial designs become more complex, they gain parameters and thresholds that need to be set with no simple and obvious method of deriving them.
- Even for a group sequential trial , choosing the number and timing of the interim, and the stopping parameter boundaries is largely a matter of custom and practice.
- Can we optimize a group sequential design over the parameter space?
- One of the problems of conventional optimization techniques (simulated annealing and genetic algorithms) is that they assume an exact result of evaluating the “function” at any particular set of parameter choices.
- But trying to optimize a clinical trial design will require simulation to estimate the operating characteristics and so we’ll have only an approximate answer. Also running the simulations can be time consuming.
- Fortunately there are relatively new optimization techniques specifically for optimizing functions with approximate values (such as from simulation) that use Bayesian smoothing, and Python code libraries (in particular “botorch”) that implement them.
- We will illustrate the approach by applying to a group sequential design (though we could probably dispense with simulation in this case) because this class of design is well known – and the results might be interesting.

Quick intro to ML and botorch

BERRY

Bayesian Optimization

- Aims to find the values of x (a vector of parameters) that maximizes f(x).
- Each x_i is assumed to be continuous and have a bounded range. [We'll need to tweak things where some inputs are not continuous (e.g. "number of interims").]
- Fortunately f() is assumed to be expensive to evaluate and it can be allowed that the results of f() have stochastic noise.
- Importantly f() is assumed to be a black box and that no derivatives are available (ruling out other optimization methods that depend on these – gradient descent, Newton, quasi-Newton).
- F() is assumed to be continuous and that smoothing between evaluated points is (Gaussian process regression) is reasonable.

Rough outline

1. Run simulations at a number of initial parameter combinations (generated using latin hypercube sampling)
2. Evaluate the utility function (the “score”) for each runs set of results
3. Fit an “Exact Marginal Log Likelihood” model. (Gaussian smoothing)
4. Use the posterior mean and find the point with the max score.
5. Use an “acquisition function” to determine the Q most useful points to simulate and evaluate next. We use the “knowledge gradient” option available in the botorch package.
6. Repeat fixed number of times / for a given elapsed time / until converged on the max point.

Approach

- To develop some Python code (a lot of machine learning code is available as Python packages)
- That can modify an initial “.facts” file to vary parameter values.
  - Complex modifications – often fairly discrete options such as number of interims or type of dose response model will possibly remain as manual changes, each option to be optimized separately.
- That can run FACTS in command line mode and read in and evaluate the results

Open questions

- Can we frame the design question sufficiently clearly that it* can be explored automatically
  - *or at least a usefully large part of the design space
- Can we summarize the simulation results into a single value to yield a ”score”?
  - This may be harder – may be a lack of consensus on trade-offs.
- Can the space be explored in reasonable time?
  - We will likely have to repeat the exploration once we see the results, either to explore a discretely different option or to tweak the optimization.

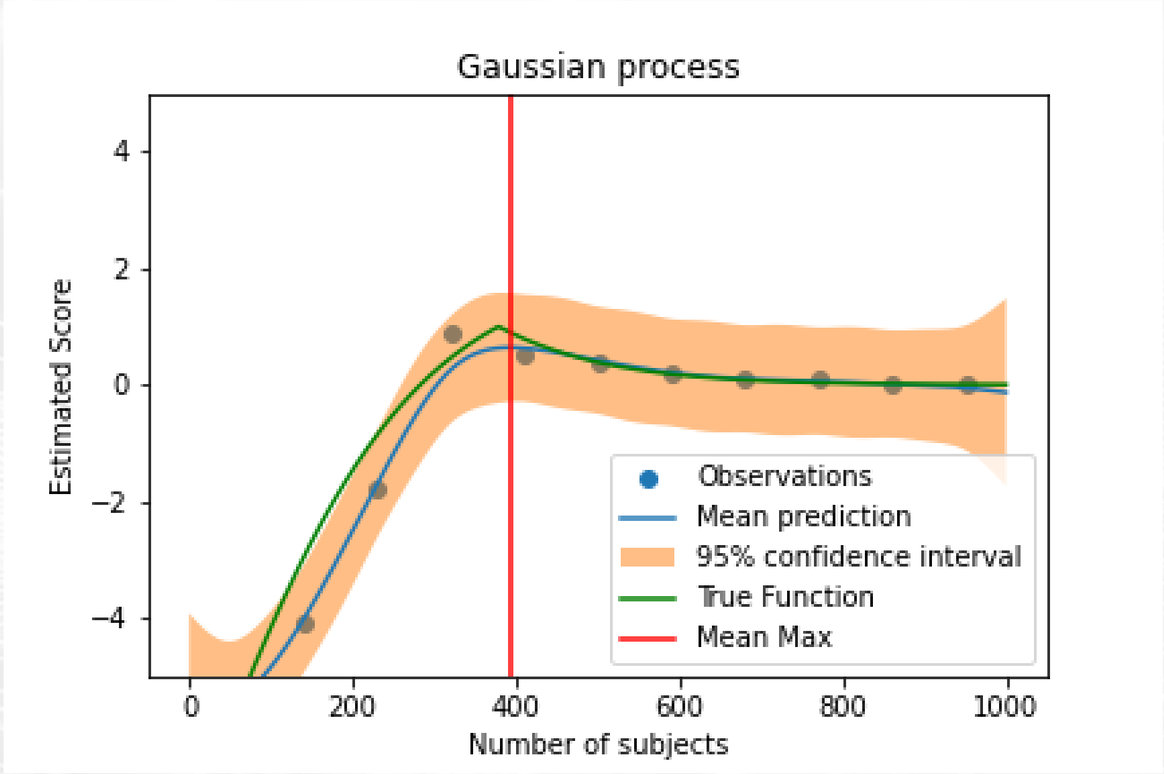
Simple Sample Size

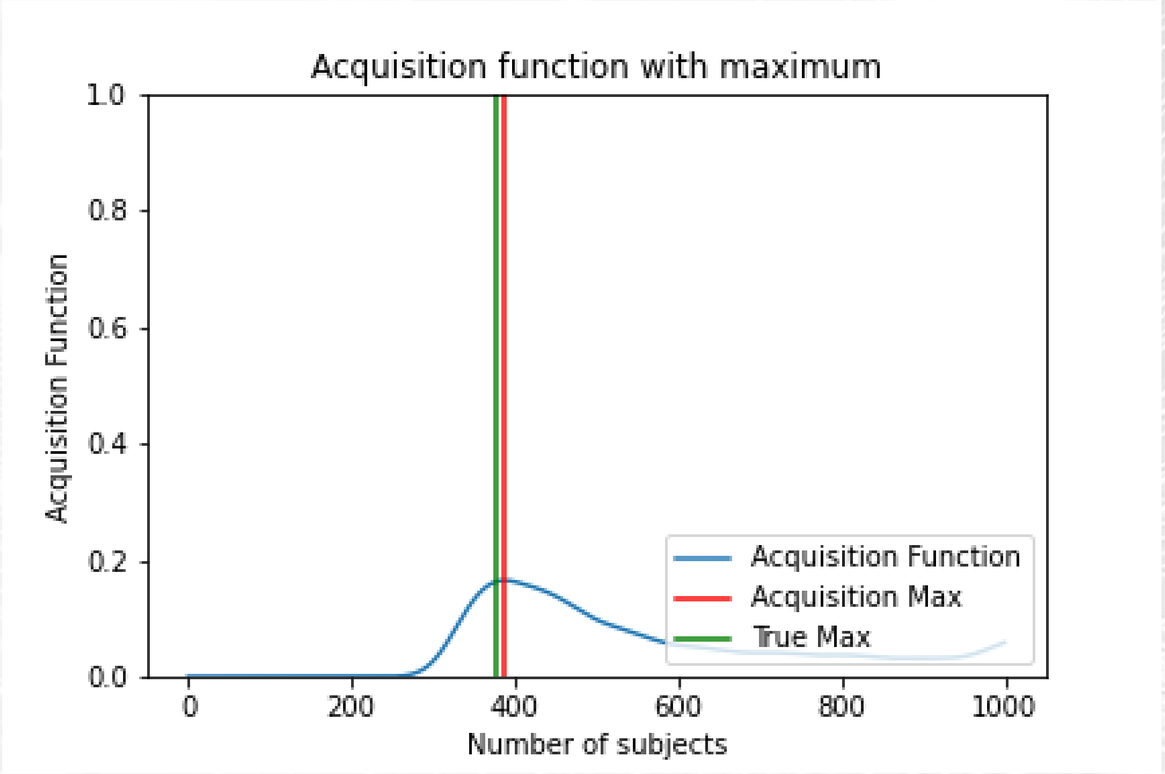
- Simple 2 fixed arm trial
- Continuous endpoint
- Target treatment effect of 1 pts
- SD of Endpoint: 3
- Required Type-1 error: 0.025 single sided
- Required Power: 0.9
- Analysis: simple t-test
- Sample size by standard sample size calculation: 190 per arm.
- Task: using FACTS simulation search the space from sample size of 25 per arm to 500 per arm (50-1,000 total sample size). To find minimum sample size with power of 0.9.
- How easy is it to reliably find the correct sample size? How many simulations? How many iterations?

Approach

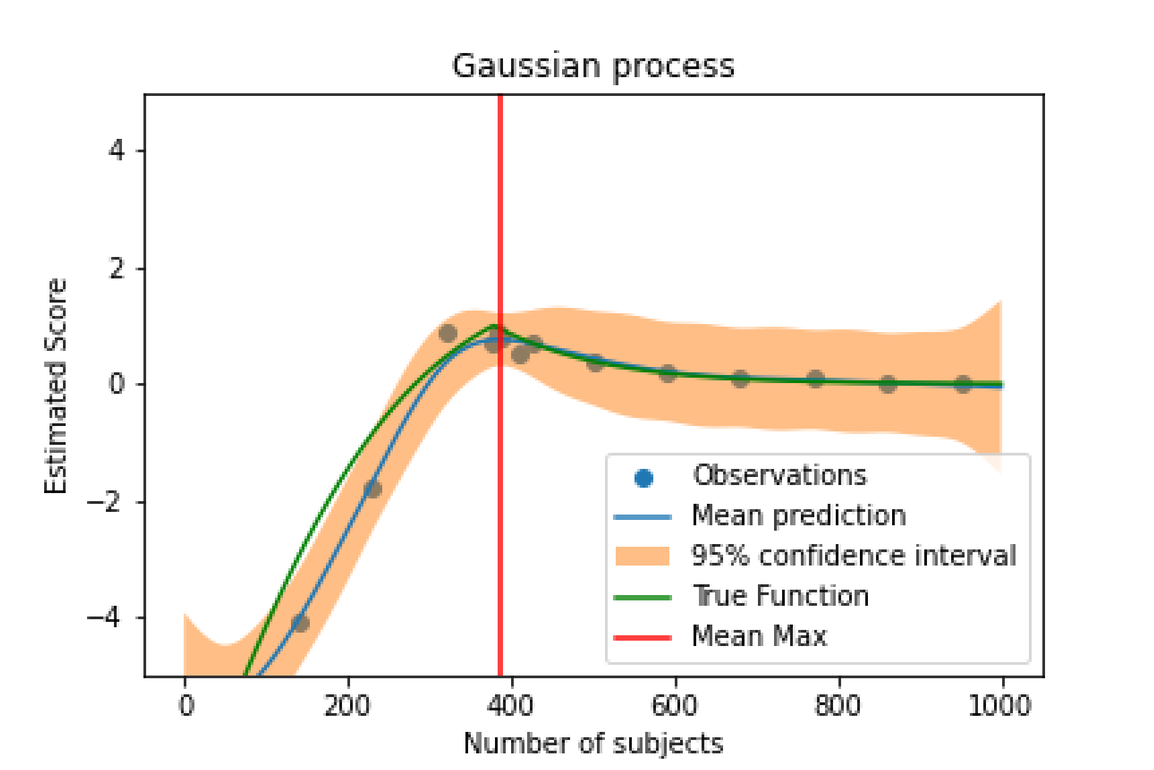
- Start with sims at 10 equally spaced intervals through the range 50-950
- The added sims sequentially at points guided by the Bayesian optimization “acquisition function” – an estimate of the expected improvement over the current maximum
- Tried running 100, sims each time and 1,000 sims each time and 3 different degrees of smoothing

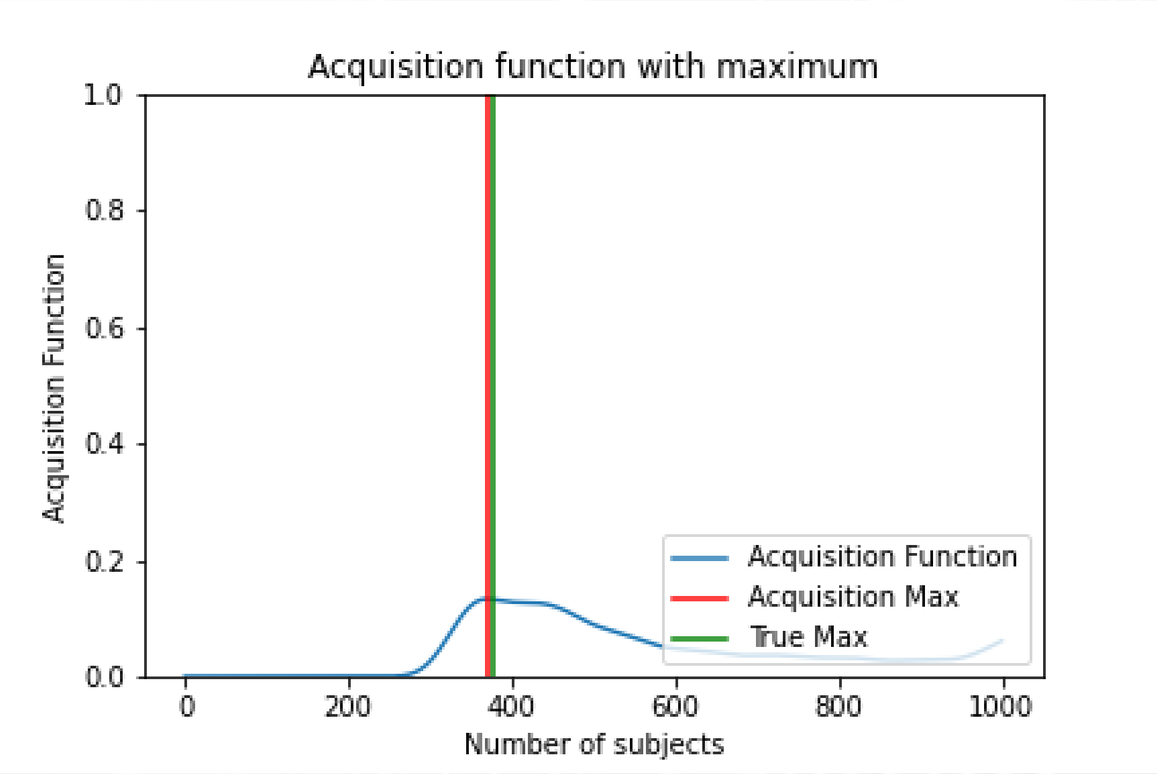
Medium smoothing nu=1.5, 100 sims per point, initial state



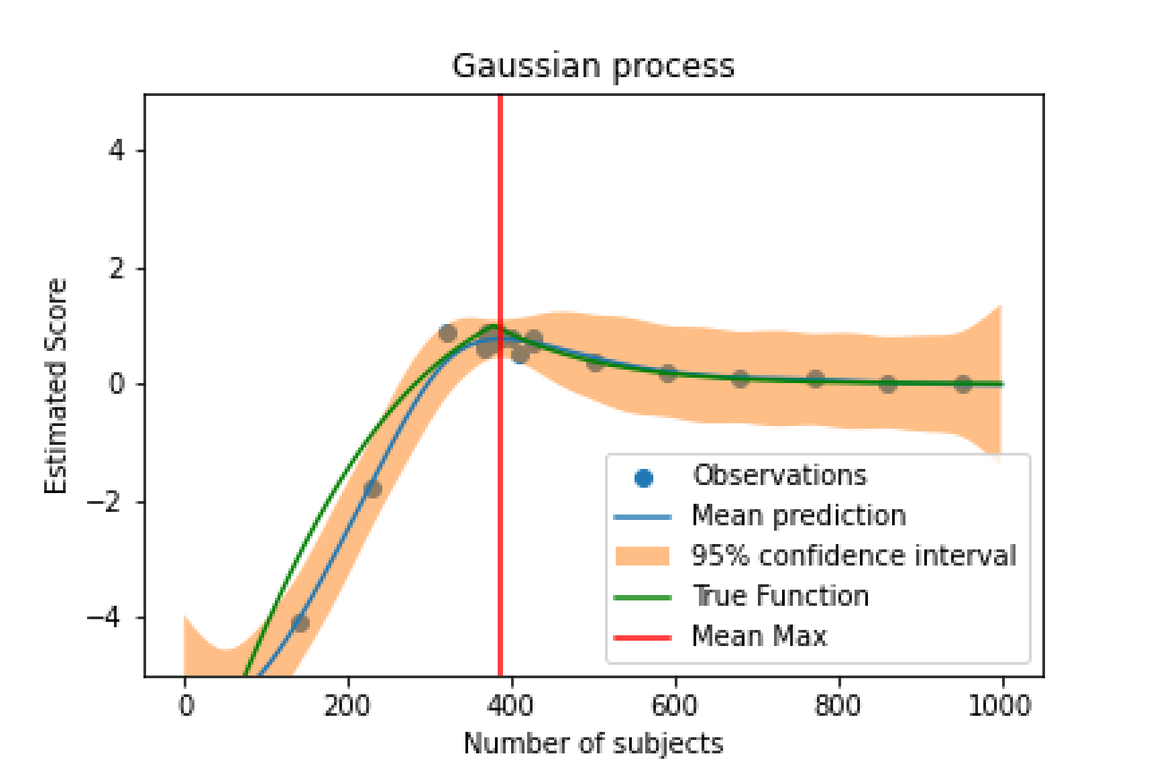


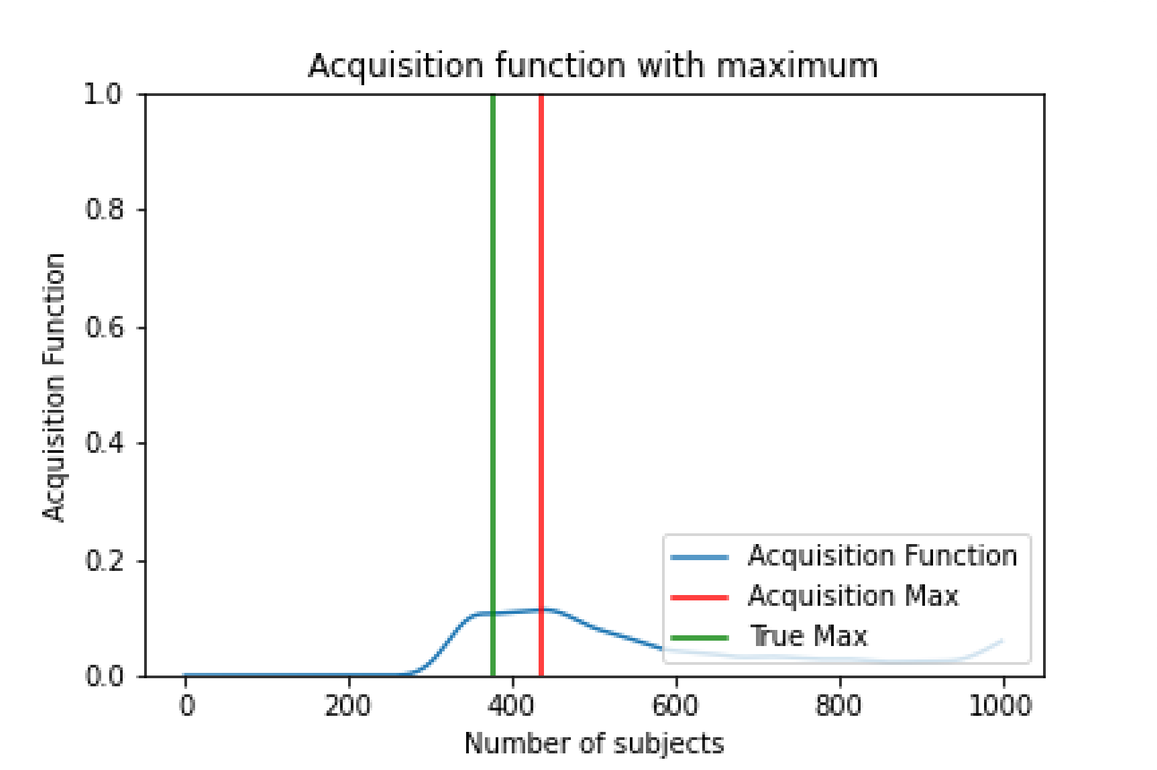
Medium smoothing nu=1.5, 100 sims per point, iteration 5



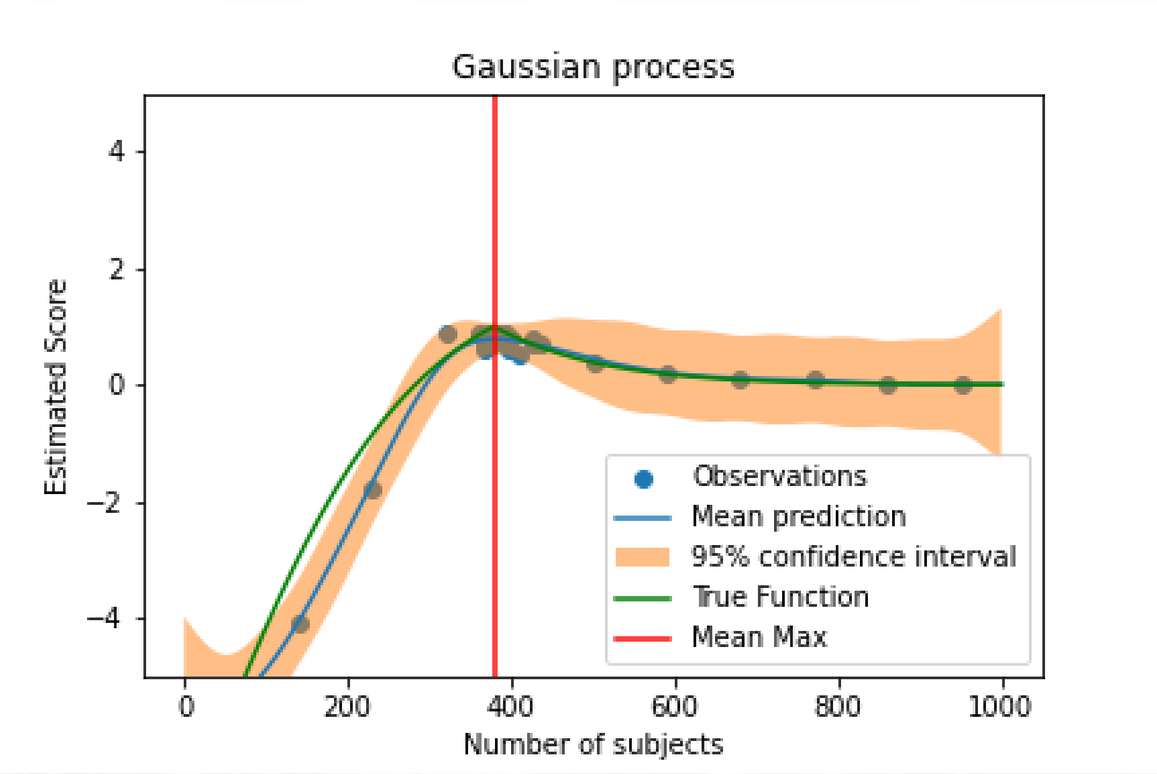


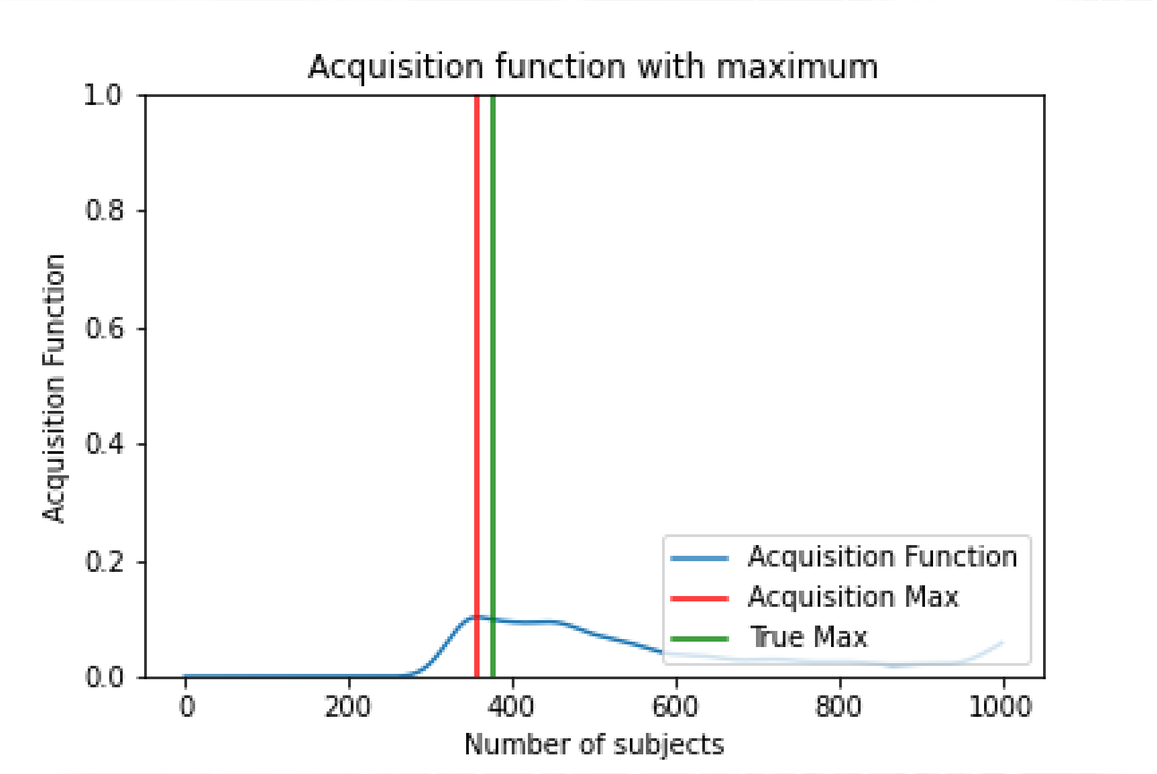
Medium smoothing nu=1.5, 100 sims per point, iteration 10



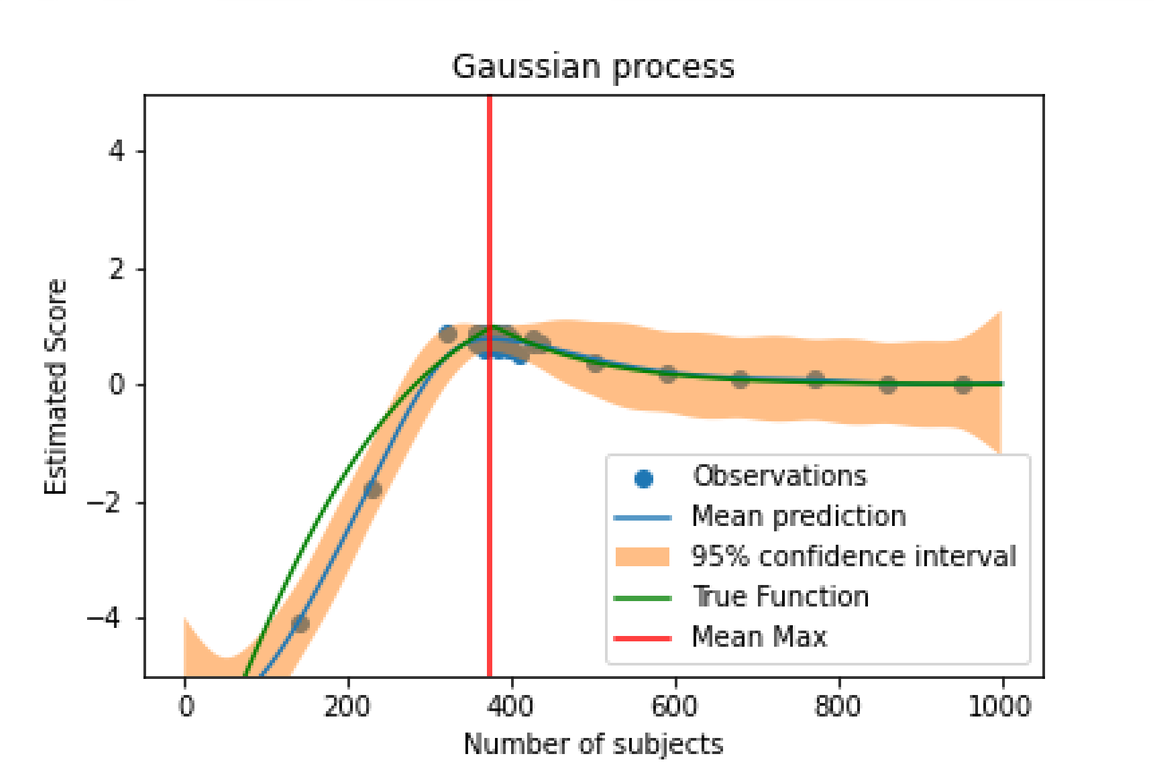


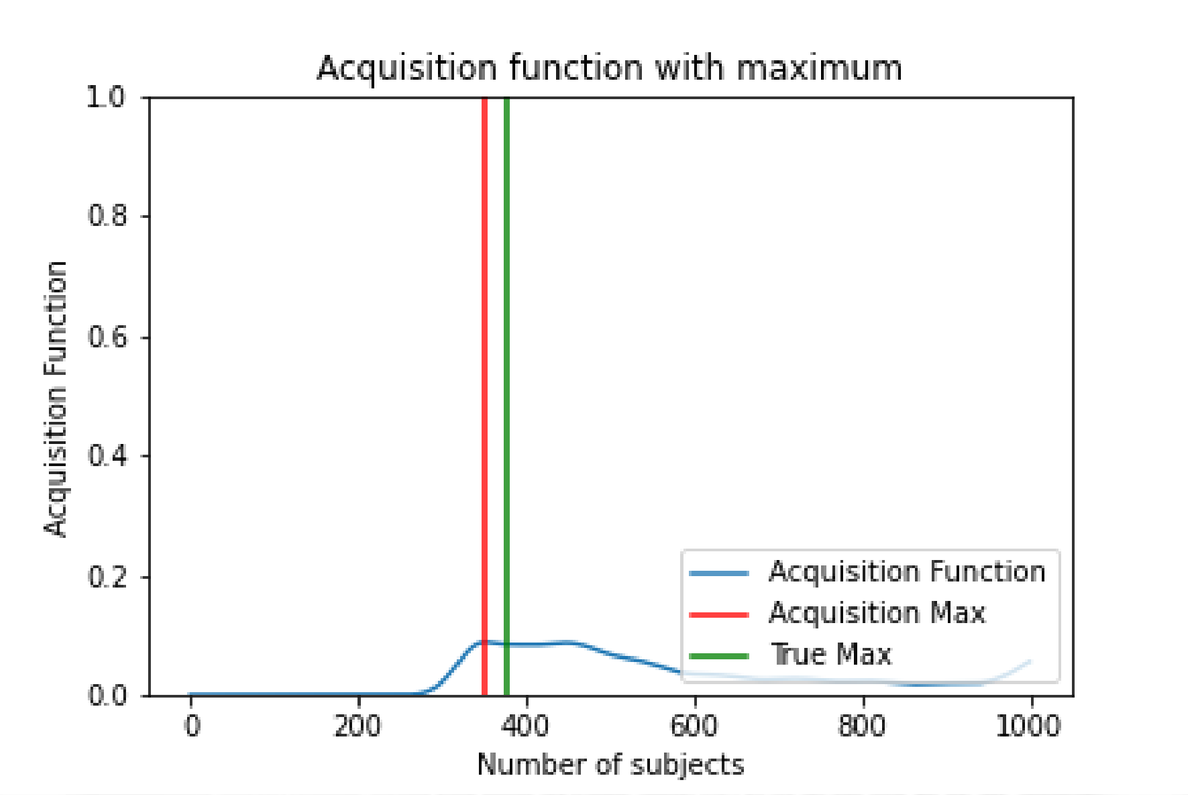
Medium smoothing nu=1.5, 100 sims per point, iteration 15





Medium smoothing nu=1.5, 100 sims per point, iteration 20





Summary

- Iterations more closely clustered around true max
- Good convergence
- Uncertainty looked reasonable
- Higher smoothing looked like it might over smooth (but we won’t rule it out for higher dimension optimization)
- Higher number of simulations looked unnecessary. At this point we haven’t simulated designs with long run time so no pressure to see if smaller number of sims is more efficient.

BERRY

Worked Example

BERRY

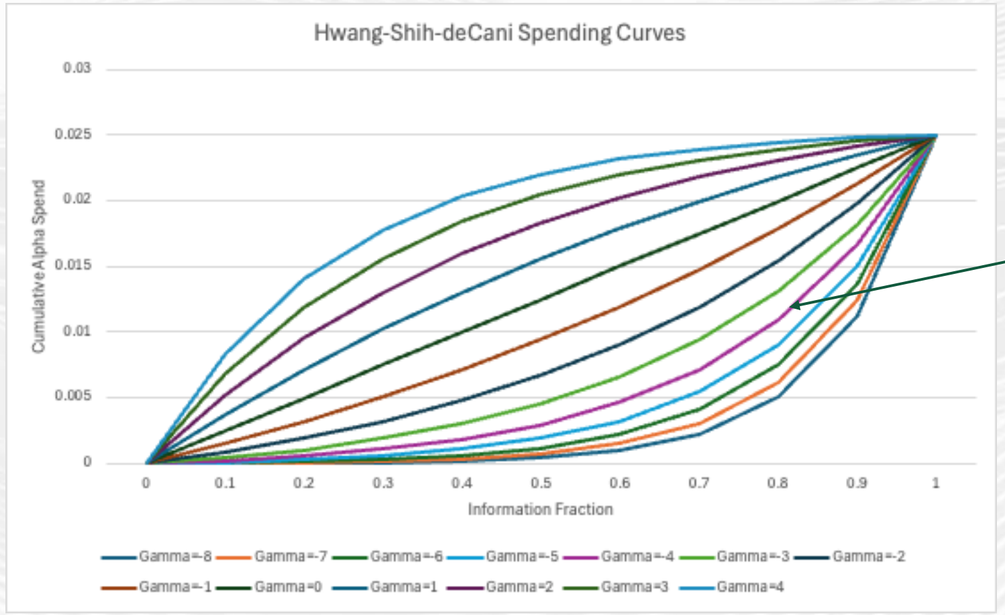
Example

- A phase 3 design, treatment vs control
- A continuous endpoint observed after 8 weeks
- Accrual 4 per week.
- Expected SD of endpoint: 3 points
- Target treatment difference 1 point.
- Required Type-1 error: 0.025, Power 90%
- Allocation 1:1
- Fixed Trial sample size: 380 (190 per arm).

The Group Sequential Design

- Allowed 2 interims
- Assume binding futility
- Timing of first interim can be explored, ranging from 0.1 of information to 0.8 of information
  - Information here is taken to be proportion of total subjects who have completed. So if overall max sample size is 400, an interim at 0.1 of information will be at 40 subjects complete.
- Second interim will always be halfway between first interim and fully complete.
- Boundaries will use Alpha and Beta spending and Hwang-Shih-deCani spending function
- The values for ɣ for the upper (success) and lower (futility) boundaries ranging from -8 to 4 will be the other two parameters we will explore

$$\alpha(\gamma, I) = \begin{cases} \alpha \frac{1-e^{-I\gamma}}{1-e^{-\gamma}} \; for \; \gamma \neq 0 \\ \alpha I \; for \; \gamma = 0 \end{cases}$$



Gamma=-4 is approximately equivalent to OBF

What is the target?

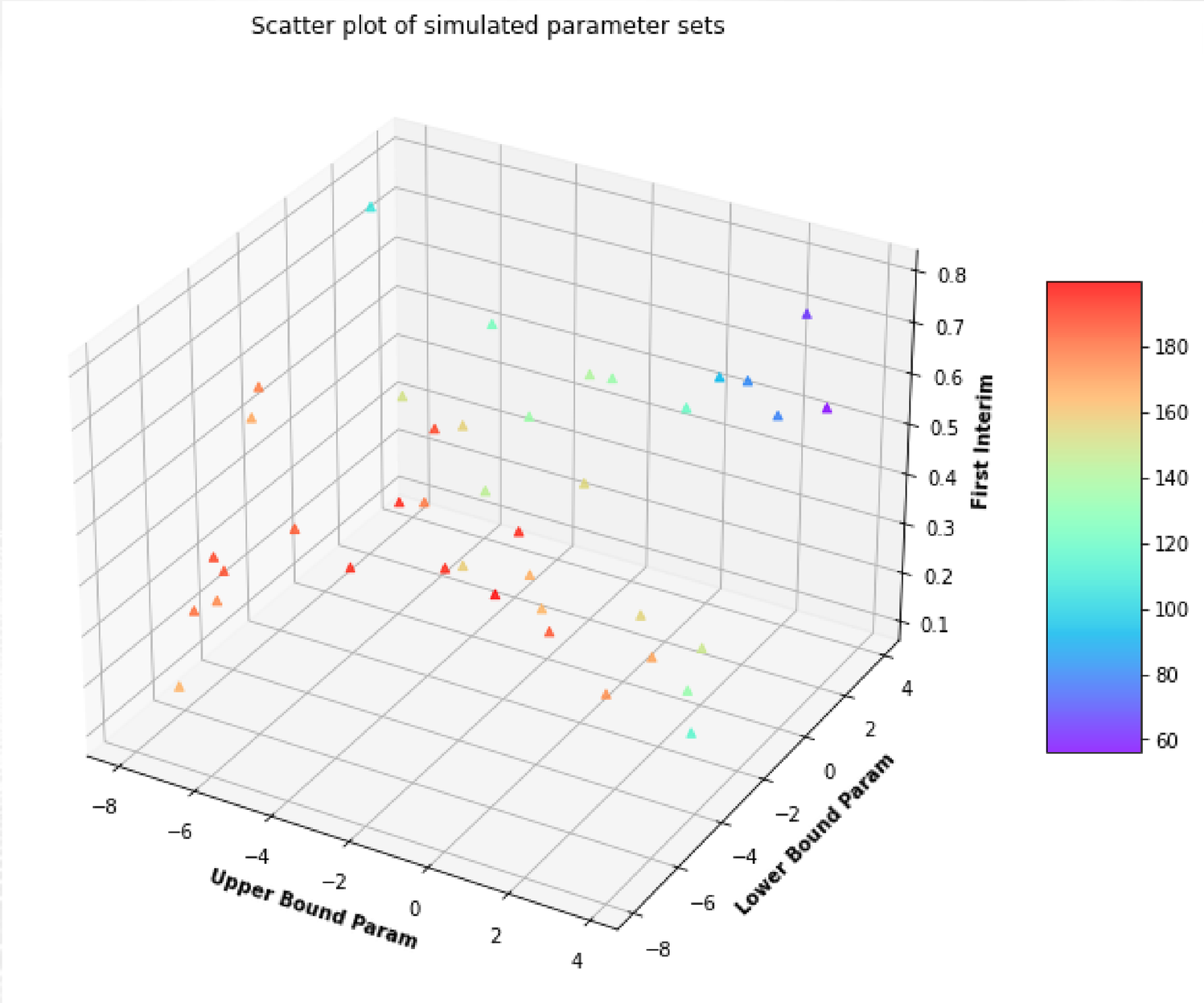
- Using Group Sequential design means that type-1 error and power are controlled. We can specify what we want them to be and given choices of number and timing of interims and boundaries compute the max sample size needed.
- The key operating characteristics by which the designs vary is sample size
  - Required max sample size
  - Expected sample size under different scenarios.
    - Null
    - 5 Alternate scenarios:
      - target efficacy (1) at expected SD of endpoint (3)
      - target efficacy +/- 20%, (0.8, 1.2) at expected SD of endpoint (3)
      - Target efficacy (1) at SD of endpoint (2.8, 3.5)
  - Choose to equally weight the scenarios, so Alternate scenarios (combined) have 5x of the weight of the Null, even though at the phase 3 stage the Null and the Alternative scenarios combined are more like 50:50 or 40:60 in likelihood (depending on indication). We felt that to a drug developer, saving time on a successful drug was a lot more important than saving money though early futility so used this 5:1 weighting. This could have easily been changed.

What is the target?

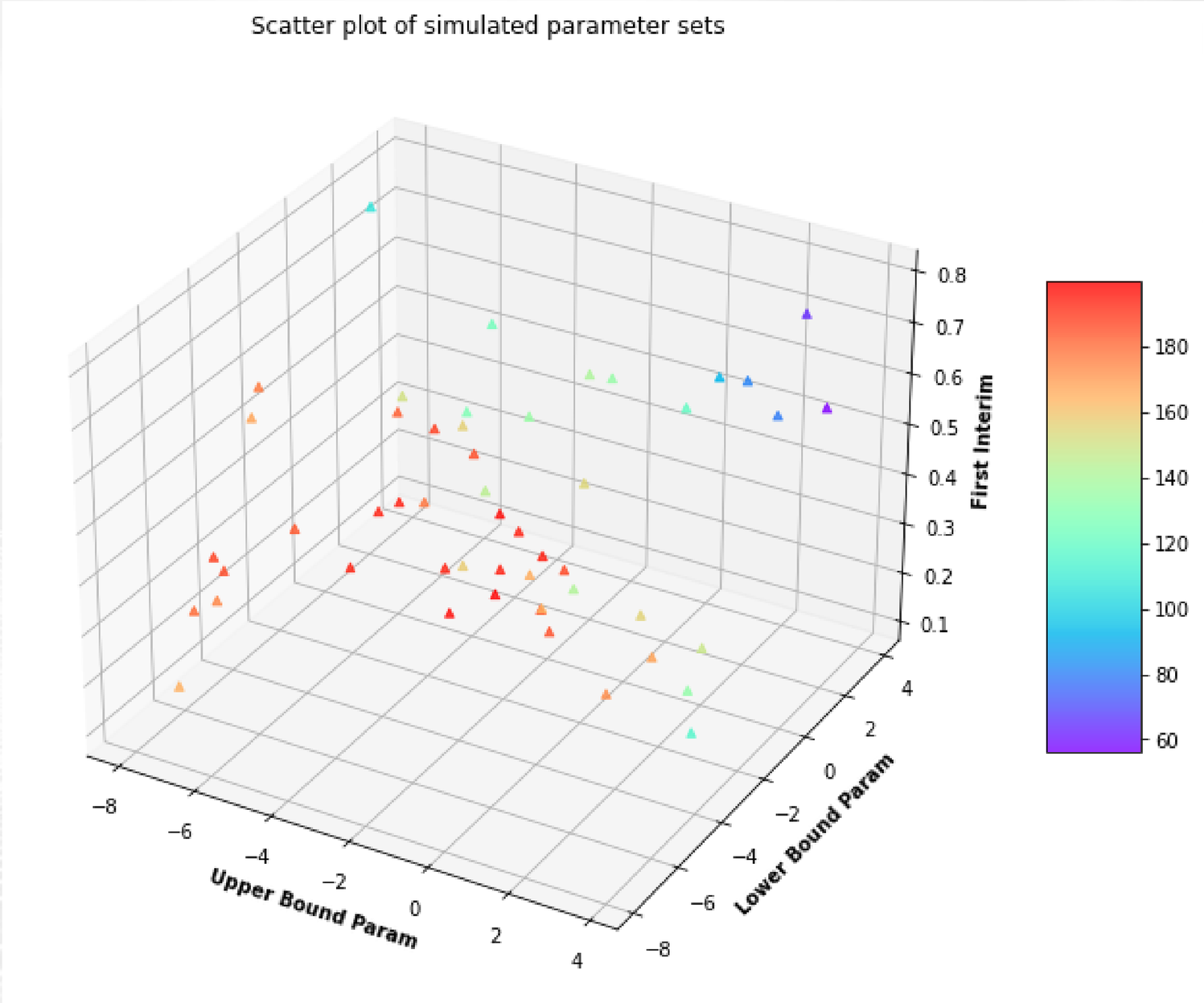
- We'd like to minimize both sample sizes. But in a GS design, as we explore different parameters typically as the Expected SS (ESS) comes down, it is as the Maximum SS (MSS) goes up.
- So we need to combine the two, and chose to include a weighting W.
- Target: minimize(ESS + W * MSS). Initially we set W to 0.5. This implies that a design change that reduces the ESS by 1 subject while increasing the MSS by less than2 subjects will be preferred. As will a change that reduces the MSS by 2 subjects while increasing the ESS by less than 1.
- Package needs +ve scores, so target is actually maximize 700 – (ESS + 0.5 * MSS)

- Before I show the results, you might like to consider what you think the best choice of parameters would be, and how certain you are of that.

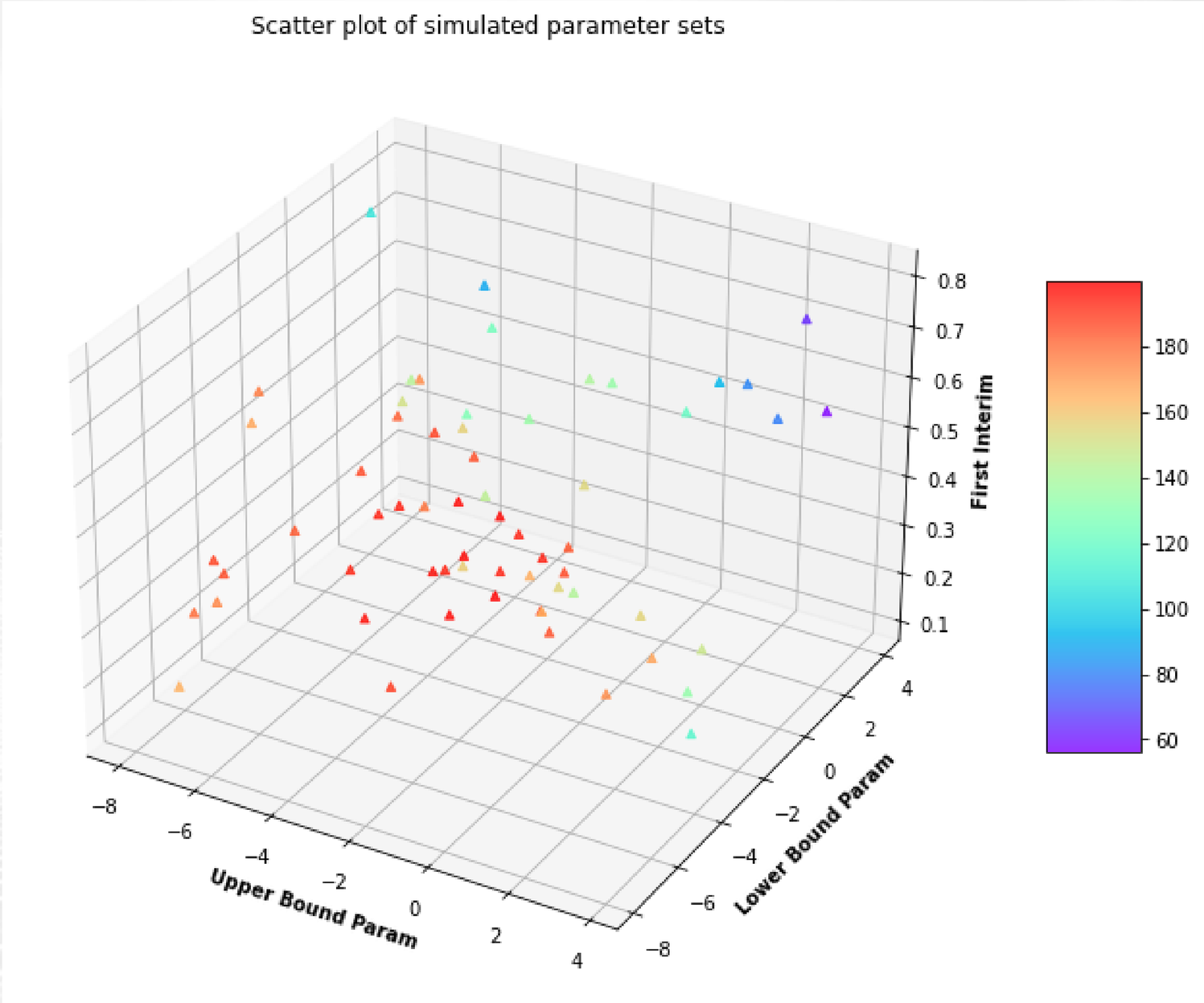
Scores of First 40 points



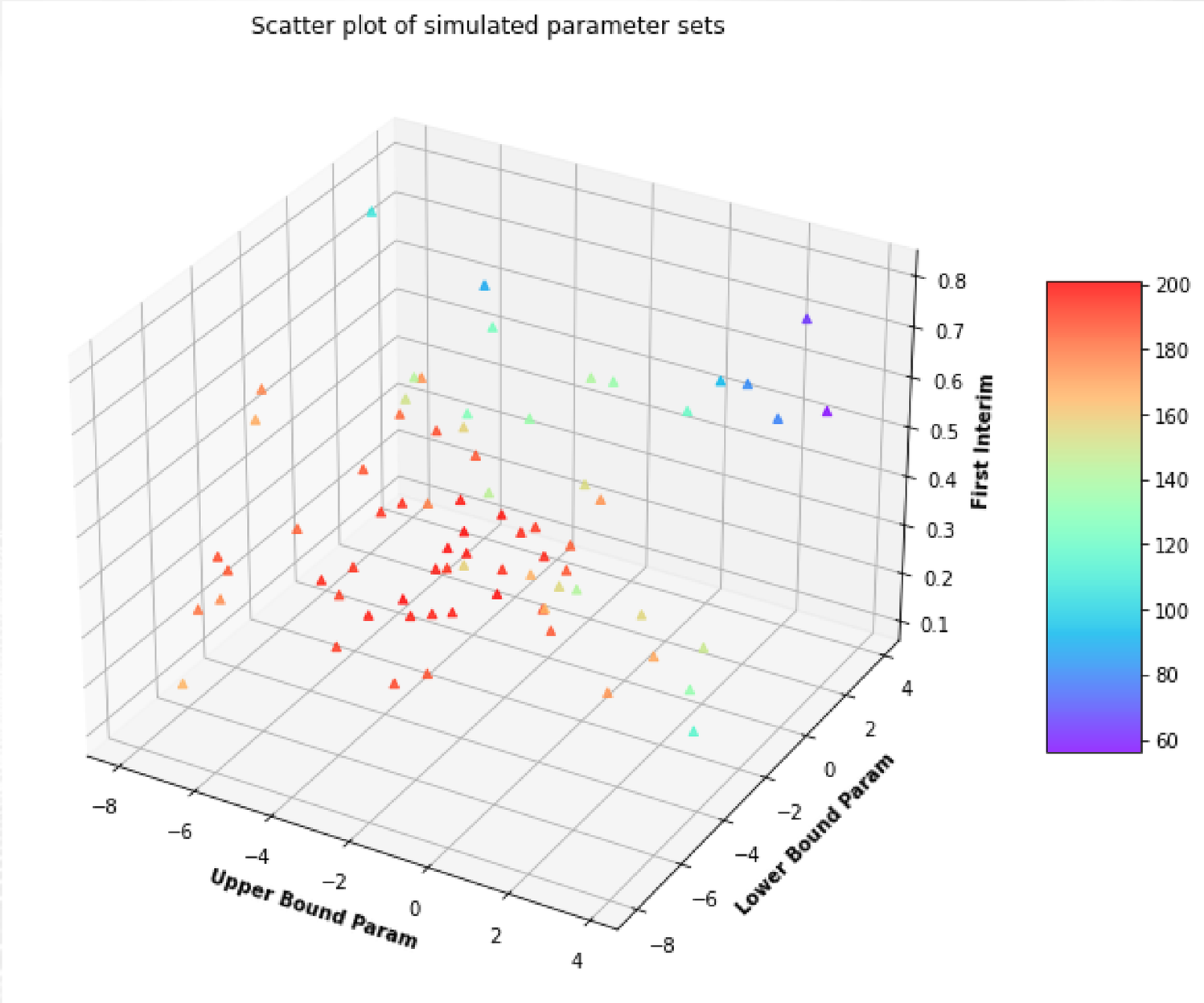
Iteration 1


Scatter plot of simulated parameter sets
Upper Bound Param
Lower Bound Param
First Interim
−8
−6
−4
−2
0
2
4
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
60
80
100
120
140
160
180

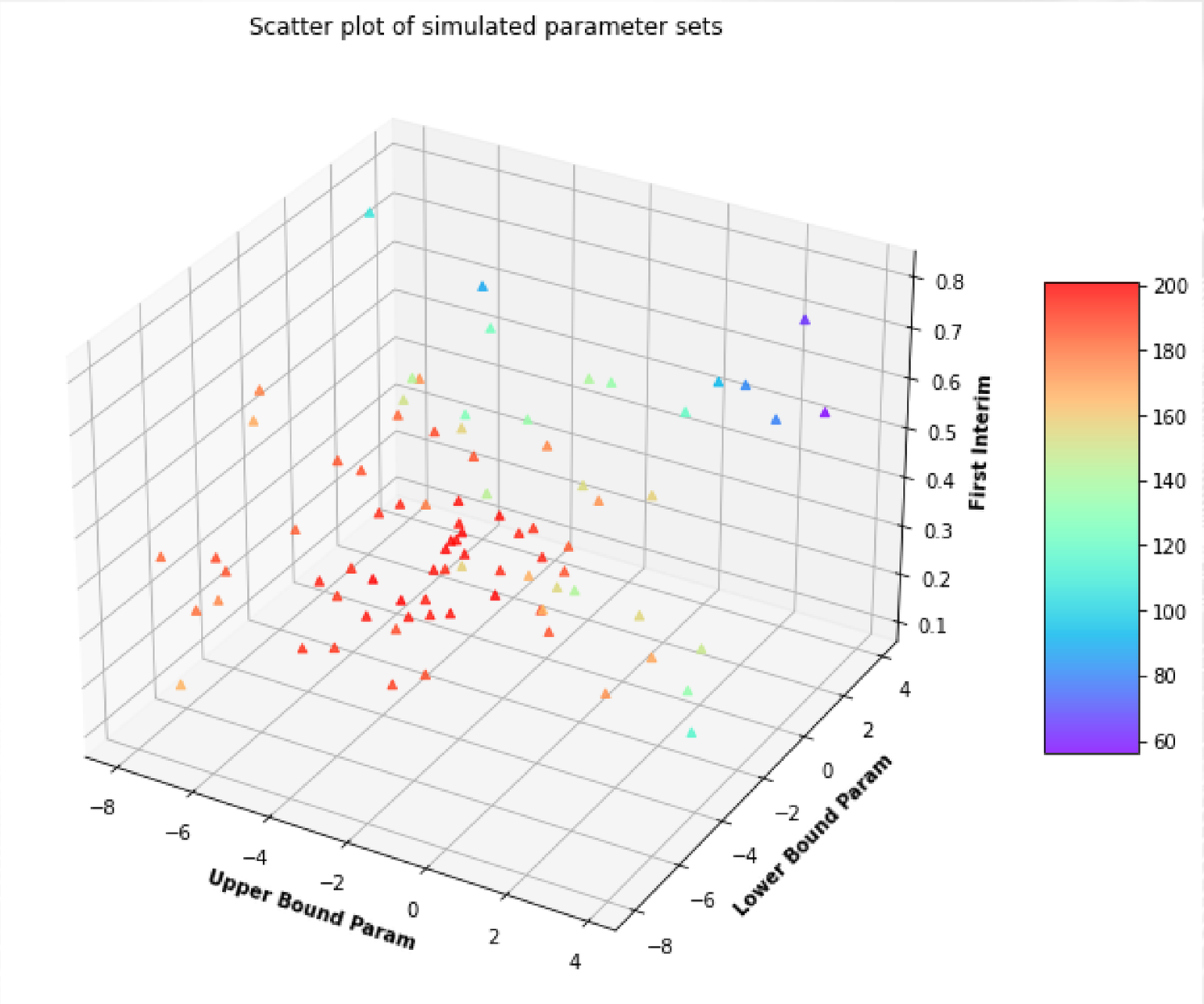
Iteration 2


Scatter plot of simulated parameter sets
Upper Bound Param
Lower Bound Param
First Interim
−8
−6
−4
−2
0
2
4
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
60
80
100
120
140
160
180

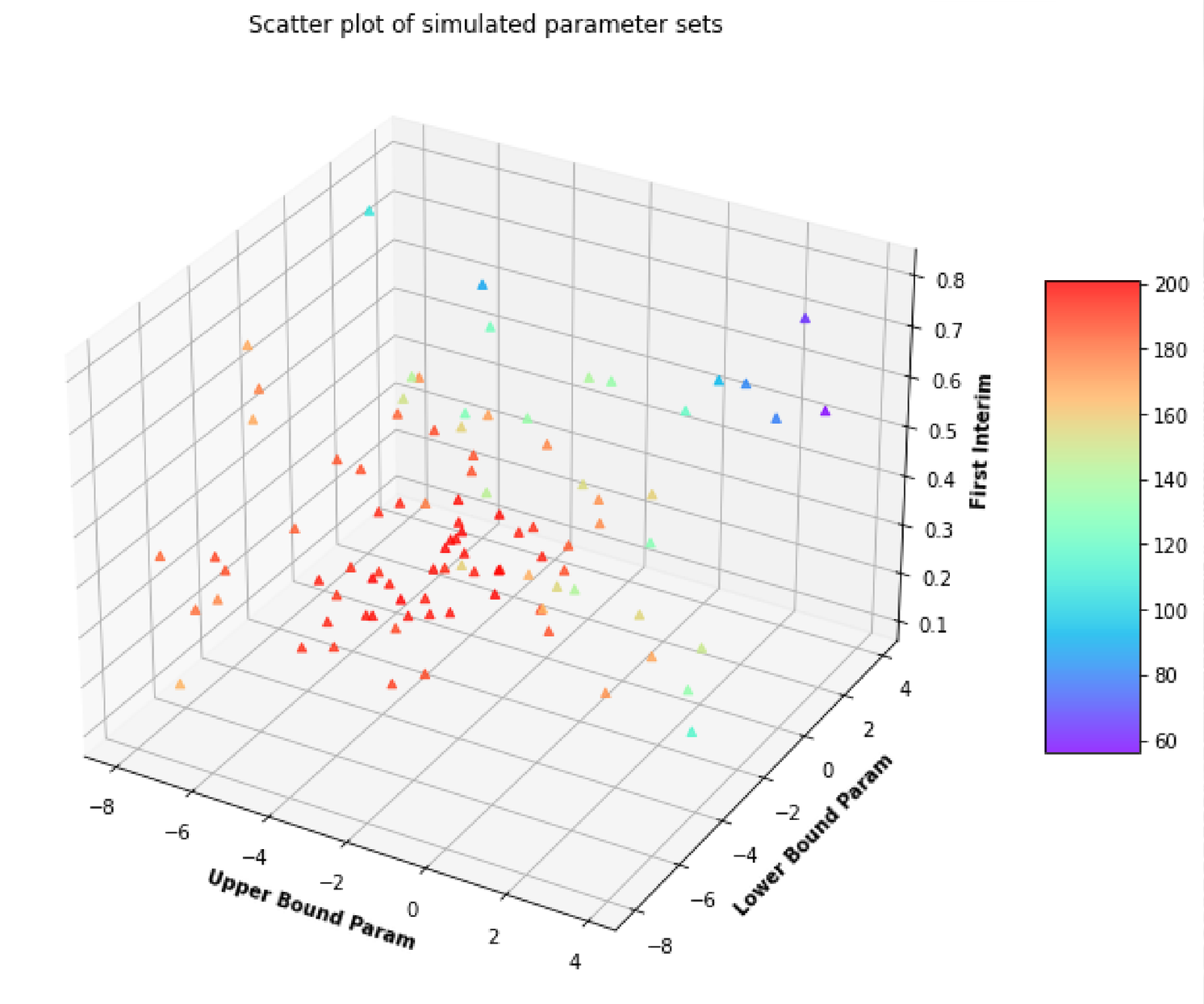
Iteration 3


Scatter plot of simulated parameter sets
Upper Bound Param
Lower Bound Param
First Interim
−8
−6
−4
−2
0
2
4
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
60
80
100
120
140
160
180
200

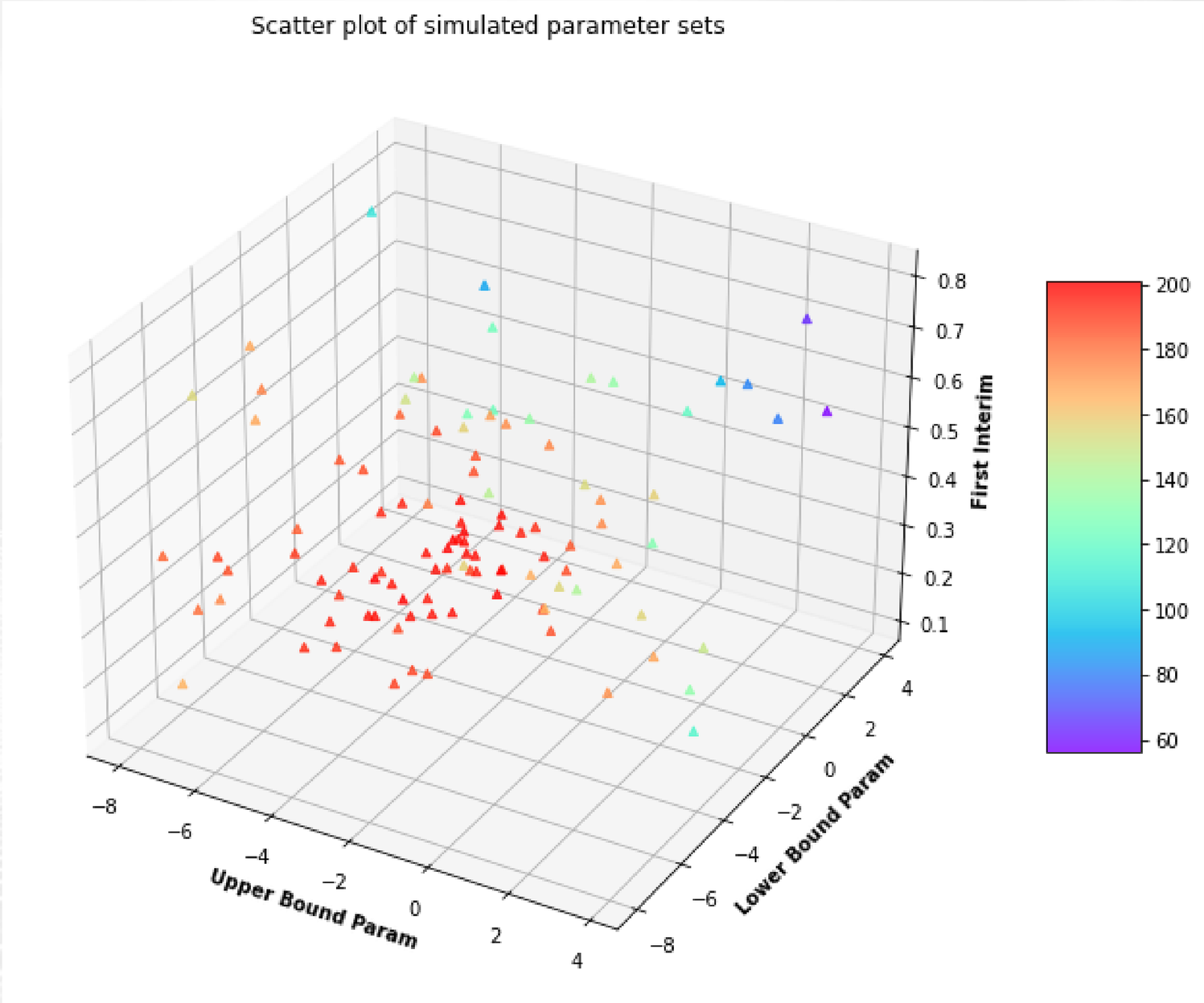
Iteration 4


Scatter plot of simulated parameter sets
Upper Bound Param
Lower Bound Param
First Interim
−8
−6
−4
−2
0
2
4
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
60
80
100
120
140
160
180
200

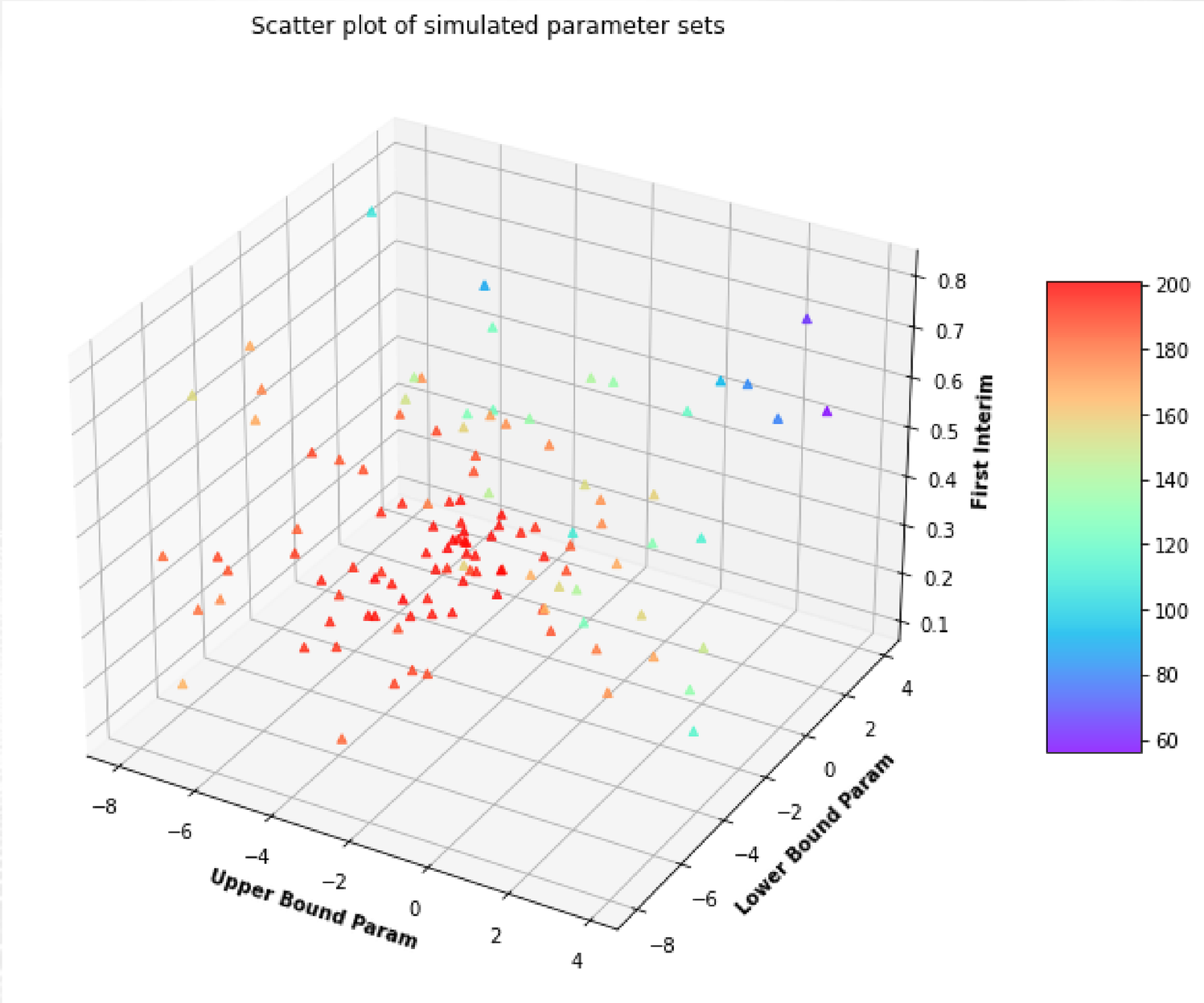
Iteration 5


Scatter plot of simulated parameter sets
Upper Bound Param
Lower Bound Param
First Interim
−8
−6
−4
−2
0
2
4
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
60
80
100
120
140
160
180
200

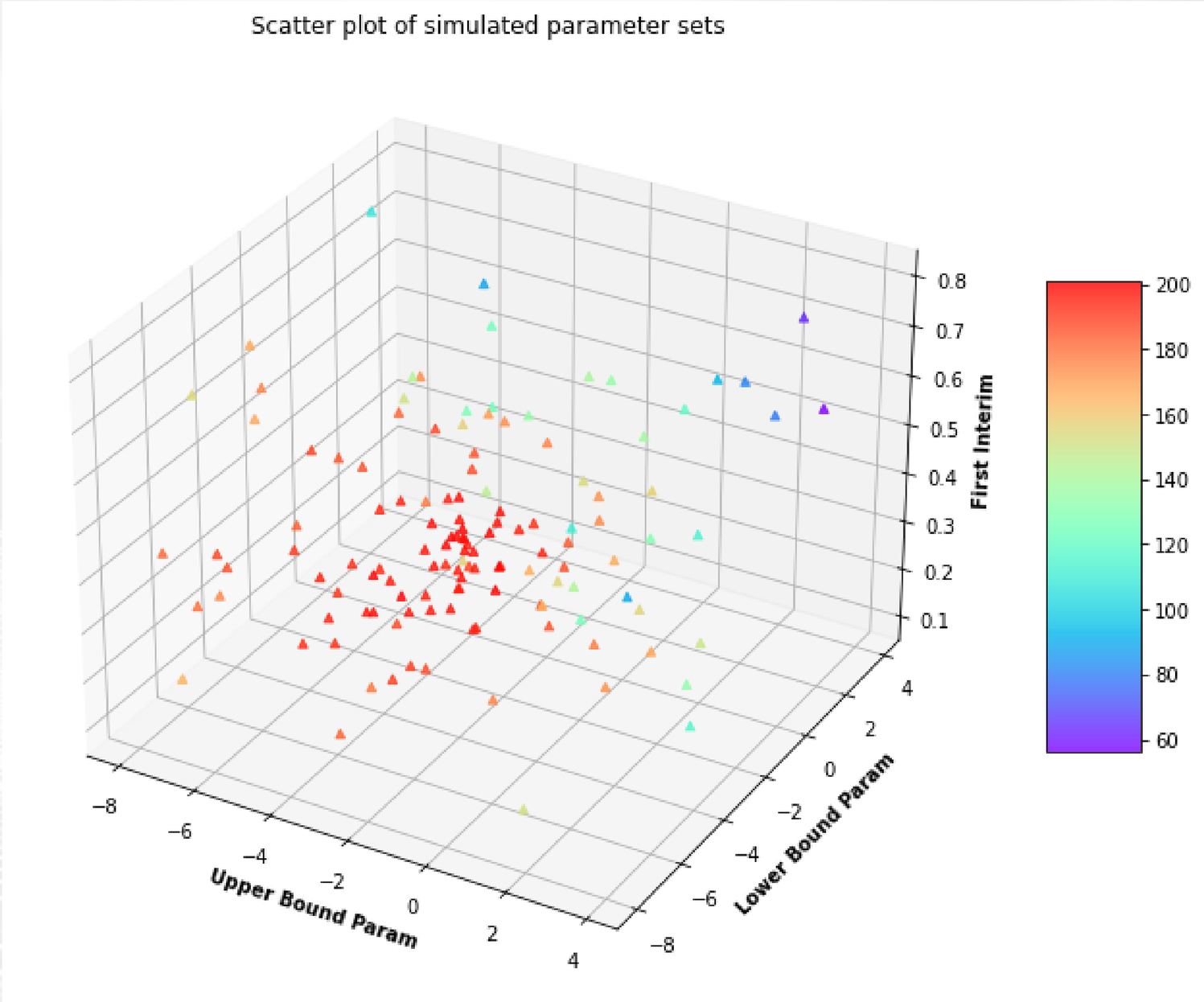
Iteration 6


Scatter plot of simulated parameter sets
Upper Bound Param
Lower Bound Param
First Interim
−8
−6
−4
−2
0
2
4
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
60
80
100
120
140
160
180
200

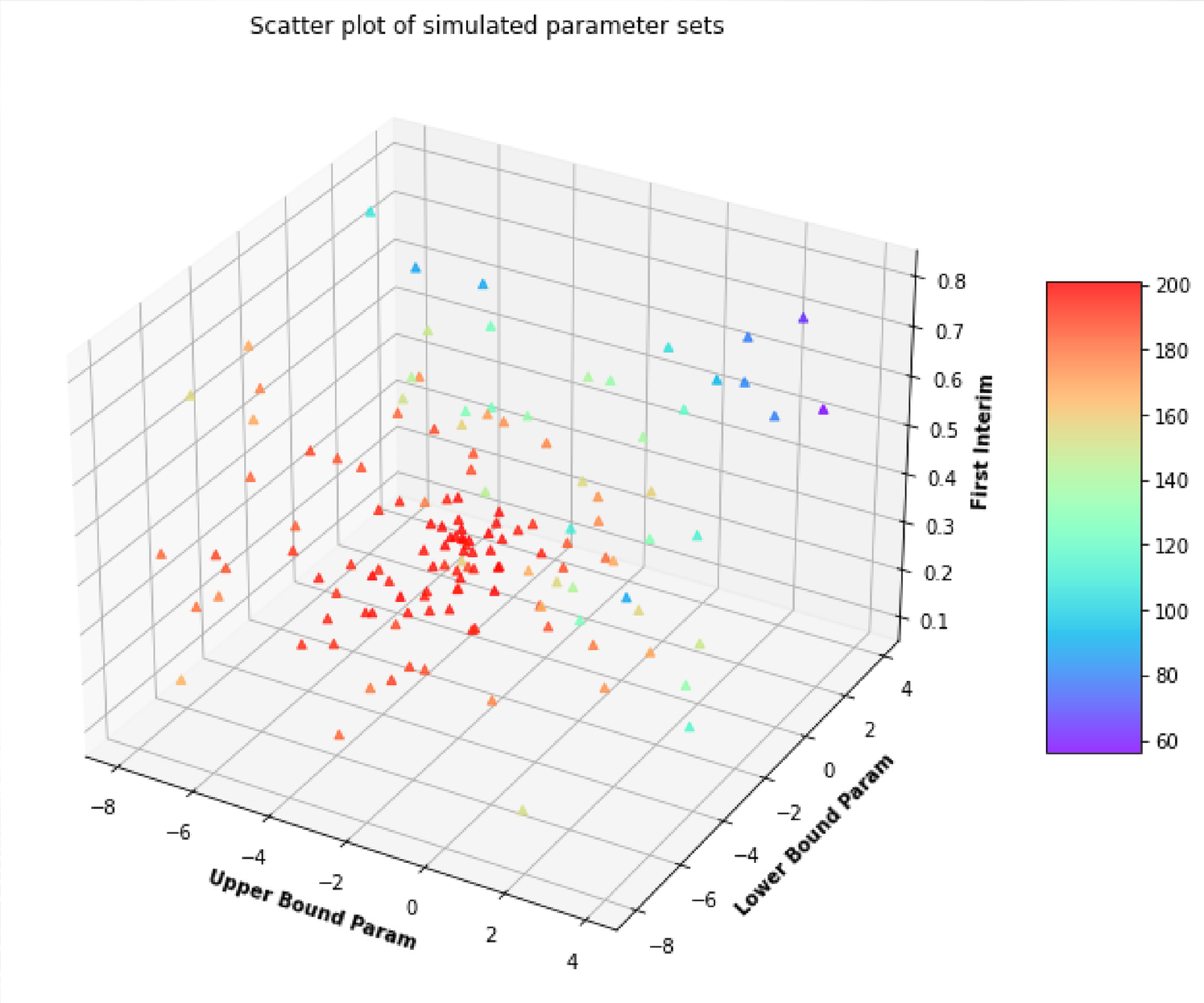
Iteration 7


Scatter plot of simulated parameter sets
Upper Bound Param
Lower Bound Param
First Interim
−8
−6
−4
−2
0
2
4
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
60
80
100
120
140
160
180
200

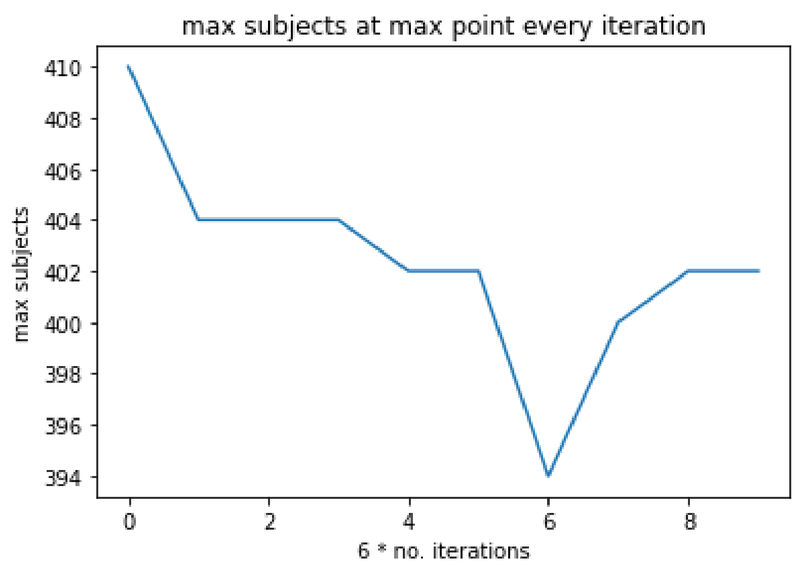
Iteration 8

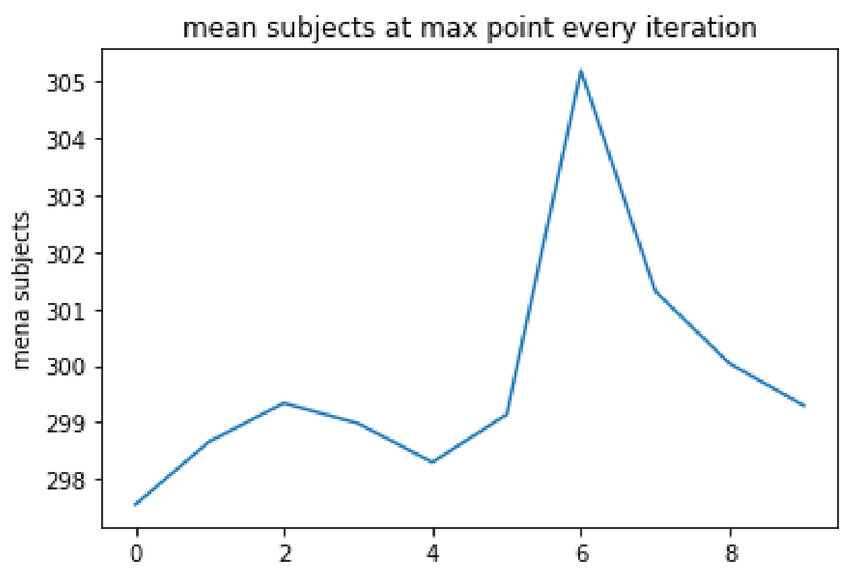

Scatter plot of simulated parameter sets
Upper Bound Param
Lower Bound Param
First Interim
−8
−6
−4
−2
0
2
4
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
60
80
100
120
140
160
180
200

Iteration 9

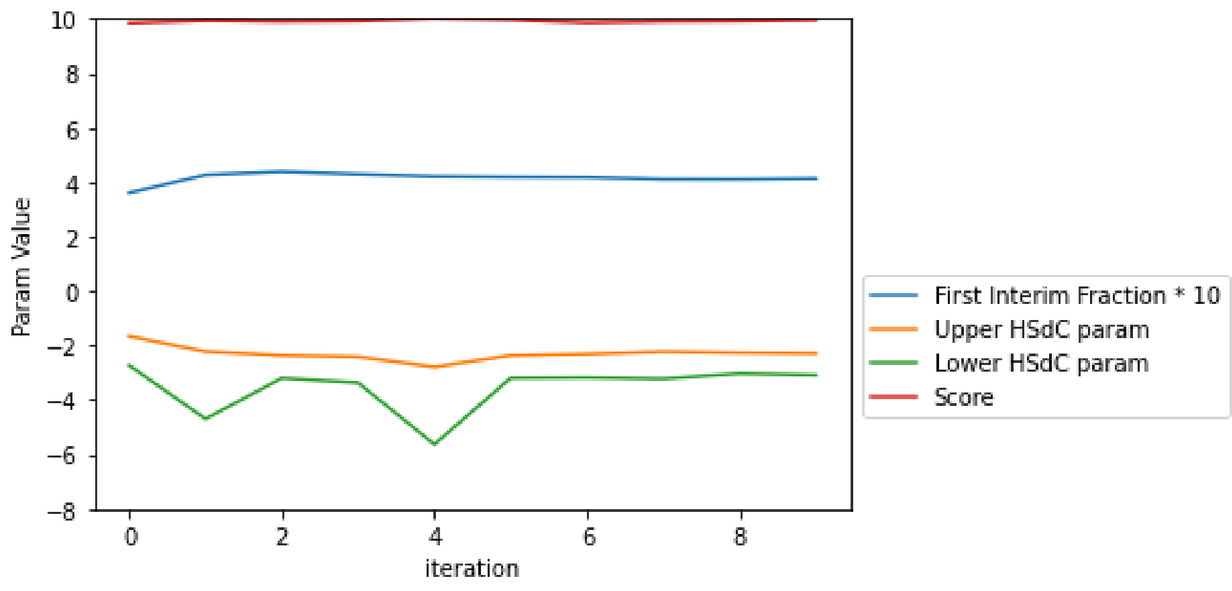

Scatter plot of simulated parameter sets
Upper Bound Param
Lower Bound Param
First Interim
−8
−6
−4
−2
0
2
4
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
60
80
100
120
140
160
180
200

Re-simulating the “max” at each iteration





The “Max” converged pretty quickly. Final values: 1st interim at 0.44, upper ɣ -2.4, lower ɣ -3.2



Score from running 10,000 at the Max and neighbouring points

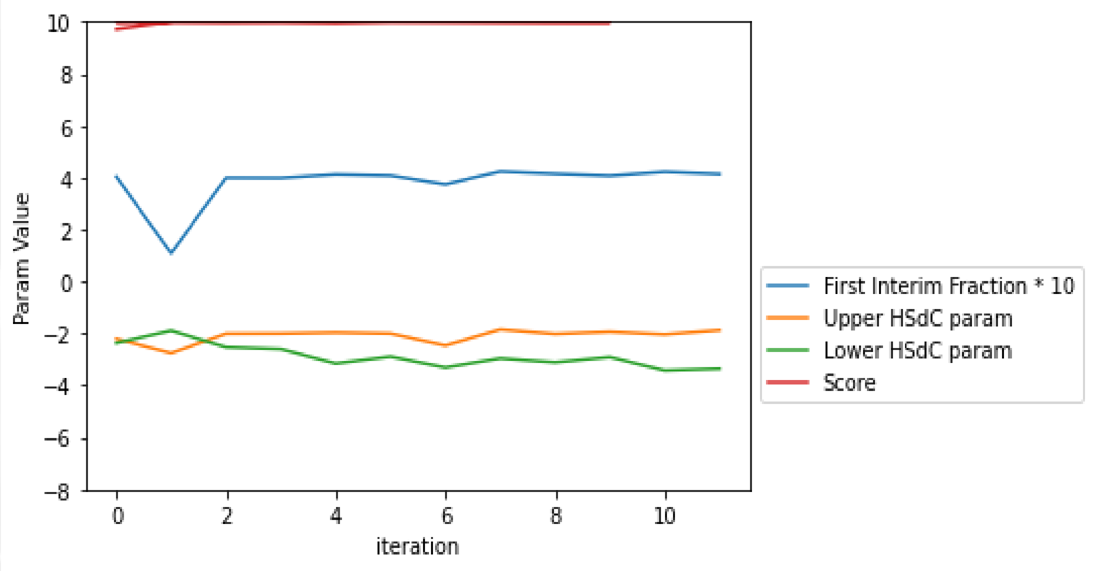
First Interim	Upper Bound γ	Lower Bound γ	Max SS	Expected SS	Weighted Sum
0.44	**-2.4**	**-3.2**	**402**	**299.7**	**500.7**
0.47	-2.4	-3.2	402	301.1	502.1
0.405	-2.4	-3.2	402	299.9	500.9
0.44	-1.8	-3.2	408	298.5	502.5
0.44	-2.9	-3.2	400	301.5	501.5
0.44	-2.4	-2.7	404	299.7	501.7
0.44	-2.4	-3.7	402	299.6	500.6

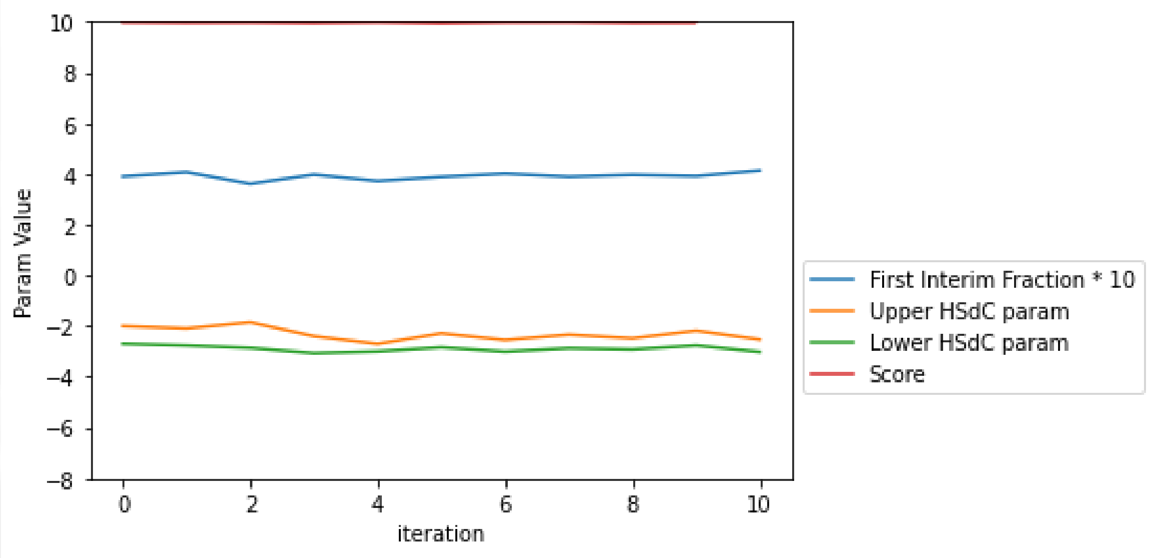
Top line show the results of simulating 10,000 times each of the scenarios using the selected maximum parameters. Is this actually the max? (Minimum of the weighted sum of the MSS and ESS)

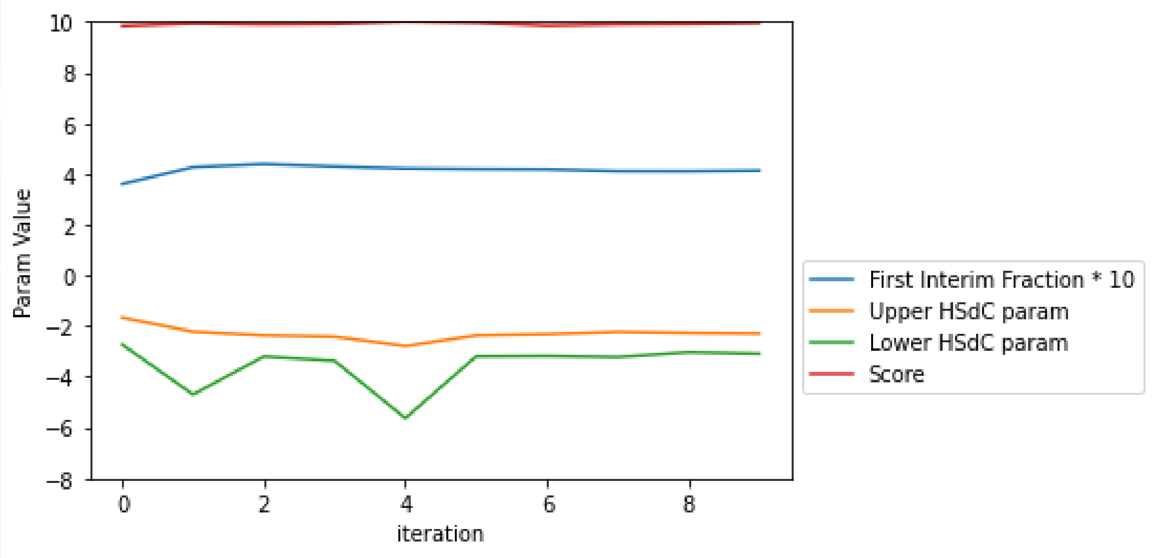
Ran 10,000 simulations of each scenarios at each parameter combination, varying each parameter up and down 10%. The area is fairly flat. The weighted sum of the sample sizes increases in 5 directions and is flat in one (lowering the lower bound)

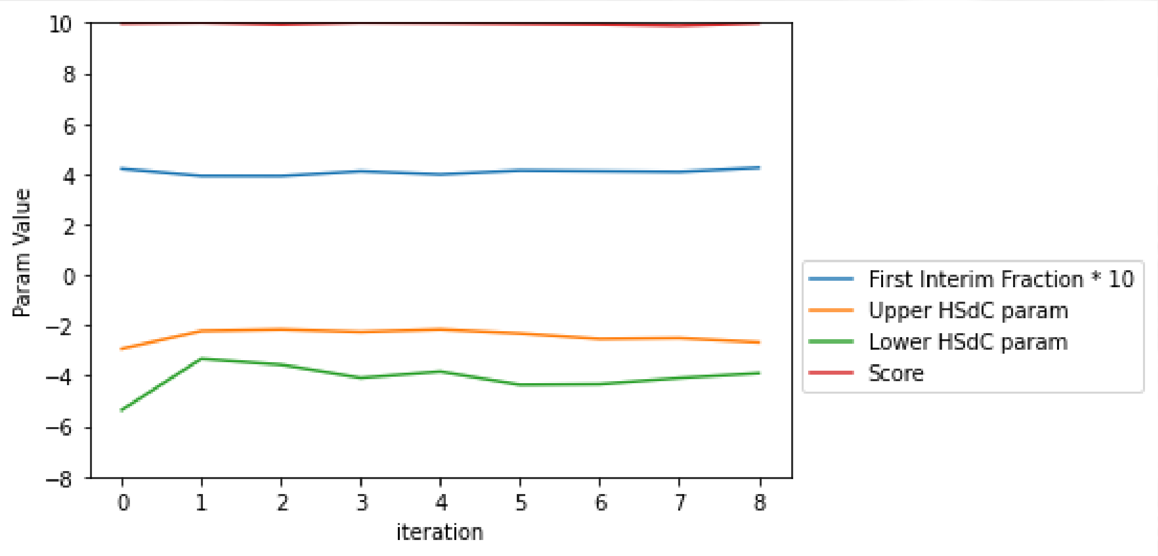
BERRY

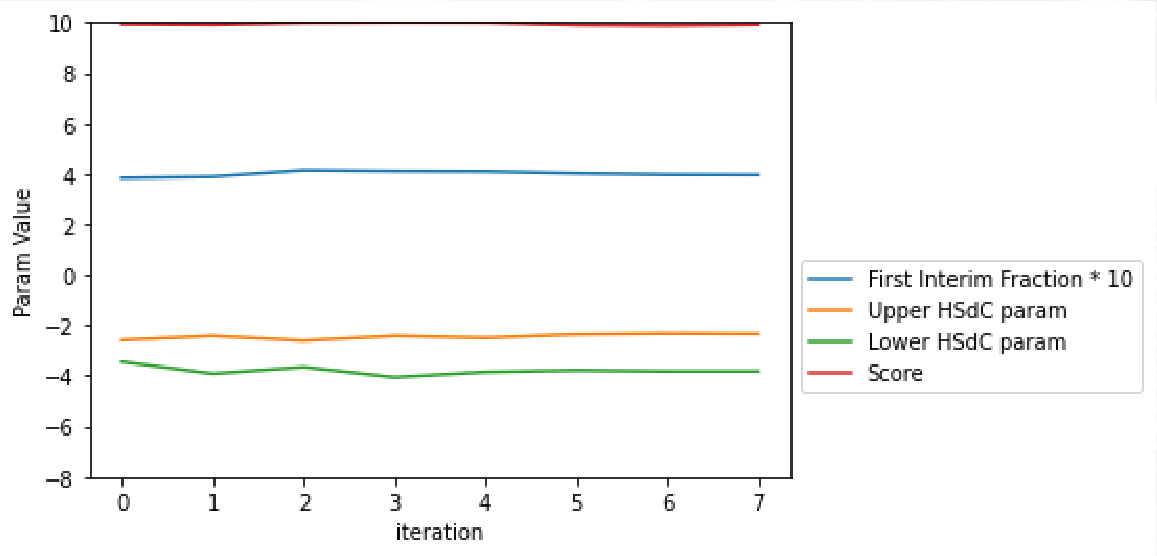
What if we change the weight?

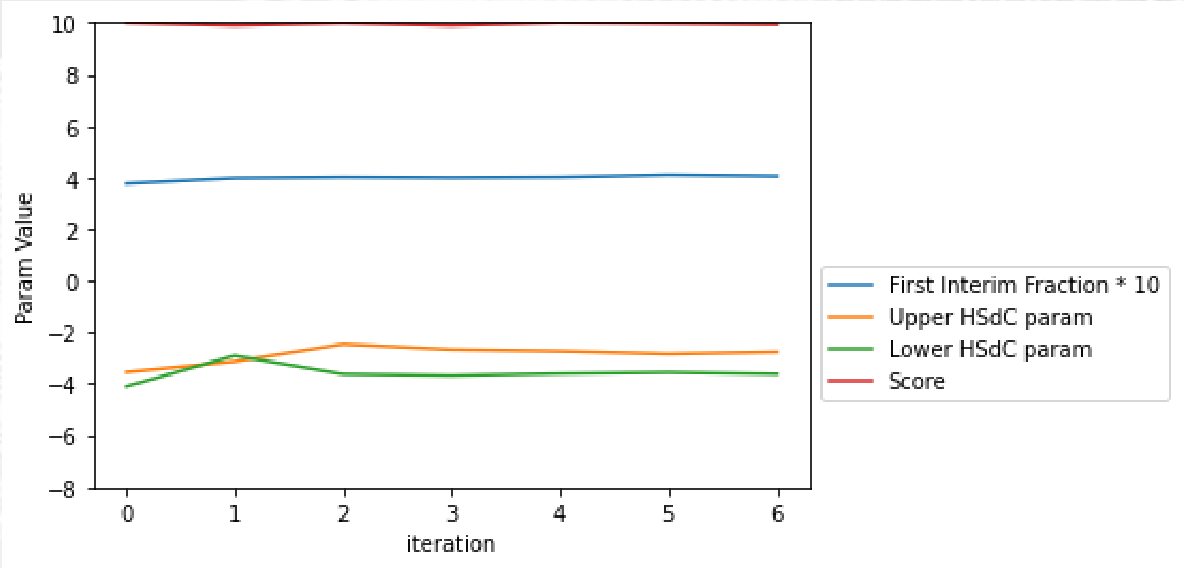


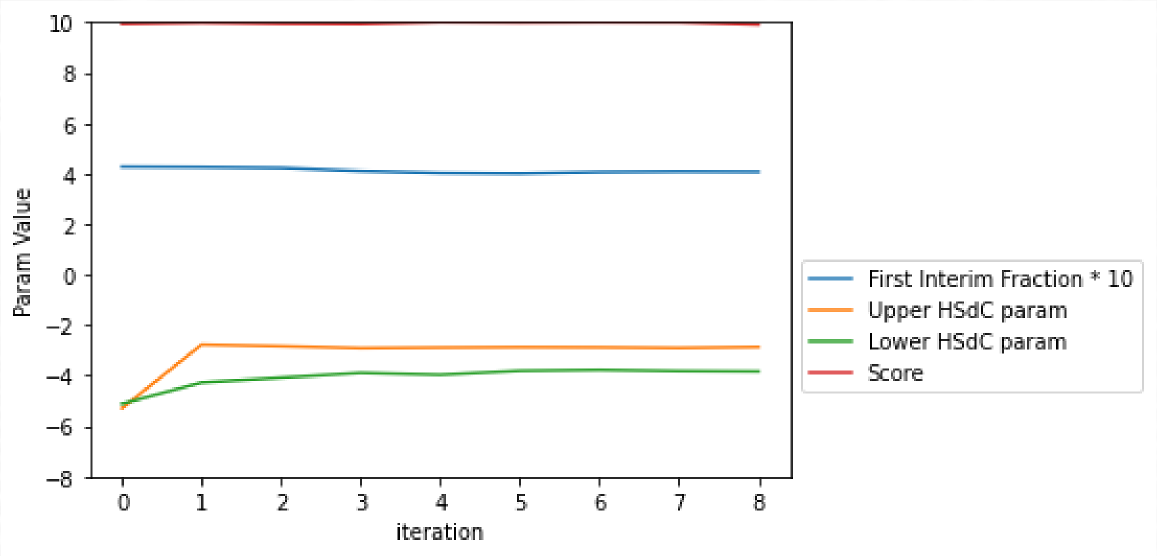












A plot of the parameters at the Max, for values of W in the sum of ESS + W * MSS from 0.3 to 0.9.

BERRY

Best parameters at the different weights

Weight	First Interim	Upper Bound γ	Lower Bound γ
0.3	0.42	-2.03	-3.42
0.4	0.41	-2.53	-3.02
0.5	0.44	-2.37	-3.20
0.6	0.42	-2.67	-3.90
0.7	0.41	-2.60	-3.65
0.8	0.43	-3.04	-4.99
0.9	0.43	-5.29	-5.11

1. The optimal first interim is at around 0.41-0.44 regardless of weight
2. Both boundaries broadly trend downwards (more conservative) the more we value limiting Max Sample Size
3. In the range of weight values 0.4-0.7 the values of the best parameter point are broadly the same.

BERRY

Conclusions

- This seems like a promising approach. Its only one example so conclusions are tentative but …

- With fast simulation times we can easily optimize the parameter values for a complex design.
- It will be interesting to see if it gets much harder / takes much longer as the number of parameters increases.
- The search converged very quickly
- We don’t need to sweat finding the ‘exact max’ it is likely to be in a pretty flat area.
- We may not even have to sweat the ‘exact utility’, a sensitivity analysis may show that the results for small changes in the utility function are pretty much the same.

Berry Consultants

Statistical Innovation

CONTACT INFORMATION

info@berryconsultants.com | (512) 213-6428

WEBSITE
berryconsultants.com

YOUTUBE
youtube.com/berryconsultants

TWITTER
@berryconsultant