# Meeting : 14 May 2019

Kristyn Pantoja

5/3/2019

Last Time

Today

Fast Algorithm

One-at-a-Time Algorithm

Random Design

Space-Filling Design

D-Optimal Design

...

# Last Time

# Evaluating Model Ideas

Tried to evaluate the model based on:

1. Robustness
   - what happens when the power $k = 0 : 1$ in one-at-a-time algorithm
   - what happens when the number of designs $K$ gets large (how it affects $\gamma_k$) in fast algorithm
2. Objective Improvement
   - Total Potential Energy

# Today

# The Plan

1. Fix optimization thing in first step
2. Make a table!

- Methods:
  - One-at-a-Time Algorithm
  - Fast Algorithm
  - Random design
  - Space-Filling design
  - D-Optimal design (is there one for model selection? e.g. one for $H_0$ and one for $H_1$? criterion doesn't seem to incorporate that.)
  - etc.
- Evaluations:
  - Expected Posterior Probabilities of $H_0, H_1$ & Bayes Factors of data generated from each $H_0, H_1$ model
  - Regression variance on coefficient (slope here)
  - Total Potential Energy criterion
  - One-at-a-Time Algorithm criterion
  - Fast Algorithm criterion

# Regression Variance

- In the case where $y_i = x_i\beta + \varepsilon_i$ (with a fixed $\beta$) with $\varepsilon_i\ N(0, \sigma_\varepsilon^2)$, we have:

$$\hat{\beta} \sim N\left(\beta, \frac{\sigma_\varepsilon^2}{\sum_{i=1}^{N}(x_i - \bar{x})^2}\right)$$

Since $\hat{\beta} = \frac{\sum_i (y_i - \bar{y})(x_i - \bar{x})}{\sum_i (x_i - \bar{x})} = \sum_i w_i y_i$, where $w_i = \frac{(x_i - \bar{x})}{\sum_j (x_j - \bar{x})^2}$, and so $Var[\hat{\beta}] = Var[\sum_i w_i y_i] = Var[\sum_i w_i (x_i\beta + \varepsilon_i)] = \sum_i w_i^2 Var[\varepsilon_i] = \sigma_\varepsilon^2 \sum w_i^2$.

- It is similar when $\beta \sim N(\tilde{\beta}, \sigma_\beta^2)$:
- After marginalizing out $\beta$, $y_i \sim N(x_i\tilde{\beta}, \sigma_\varepsilon^2 + x_i^2\sigma_\beta^2)$. Hence:

$$Var[\hat{\beta}] = Var[\textstyle\sum_i w_i y_i] = \sum_i w_i^2 Var[y_i] = \sum_i w_i^2(\sigma_\varepsilon^2 + x_i^2\sigma_\beta^2)$$

# Criterions

1. The total potential energy, which both algorithms aim to minimize:

$$\sum_{i \neq j} \frac{q(\mathbf{x}_i)q(\mathbf{x}_j)}{d(\mathbf{x}_i, \mathbf{x}_j)}$$

2. One-at-a-Time Algorithm criterion tries to minimize:

$$\left\{ \sum_{i \neq j} \left( \frac{q(\mathbf{x}_i)q(\mathbf{x}_j)}{d(\mathbf{x}_i, \mathbf{x}_j)} \right)^k \right\}^{1/k}$$

3. Fast Algorithm tries to minimize:

$$\max_{i \neq j} \frac{q(\mathbf{x}_i)q(\mathbf{x}_j)}{d(\mathbf{x}_i, \mathbf{x}_j)}$$

# Parameters

- Parameters

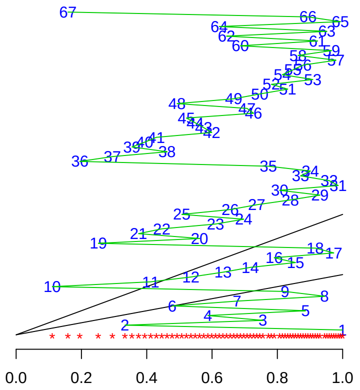
```
mean_beta0 = 1 # slope of null model
mean_beta1 = 1 / 2 # slope of alternative model
var_mean = 0.001 # variance on beta
var_e = 0.01 # variance on error
```

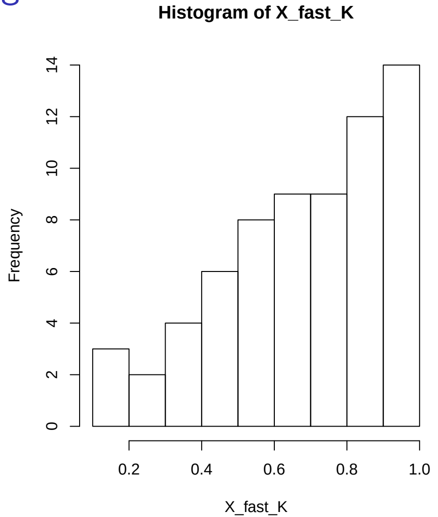
- Settings for Fast and One-at-a-Time algorithms

```
N = 67
# for fast algorithm:
K = 20 # ceiling(4* sqrt(p))
numParameters = 1 # number of parameters (just slope!)
p = numParameters * 2
# for one-at-a-time algorithm:
numCandidates = 10^5 # suggested 10^5
k = 4
```

# Fast Algorithm

# Design generated by Fast Algorithm




```r
mean(X_fast_K)
```

```
## [1] 0.6836593
```

```r
sd(X_fast_K)
```

```
## [1] 0.2296987
```

# Evaluations

```r
# Expected Post Probs and BF01 : Y | H0 Sims
c(expect_post_H0_YH0fast, expect_post_H1_YH0fast, BF01_YH0f
```

```
## [1] 0.874464 0.125536 6.965844
```

```r
# Expected Post Probs and BF01 : Y | H1 Sims
c(expect_post_H0_YH1fast, expect_post_H1_YH1fast, BF01_YH1f
```

```
## [1] 0.1232342 0.8767658 0.1405555
```

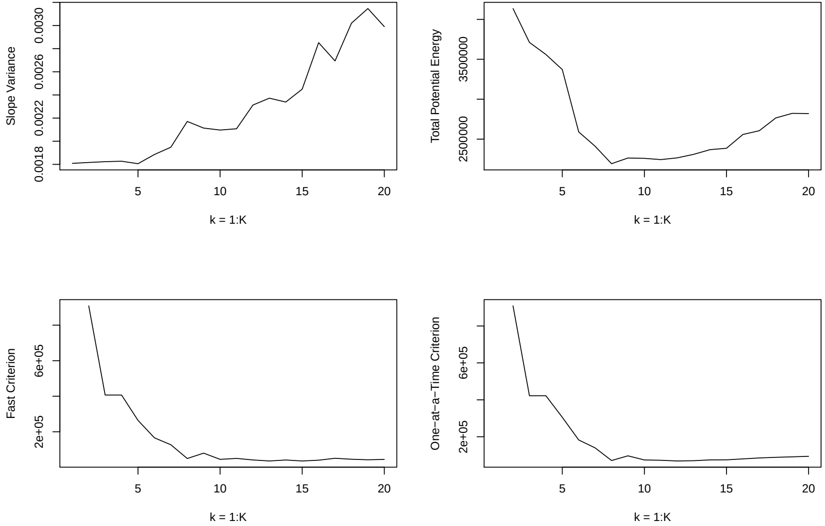
```r
# Slope Variance
v_fast
```

```
## [1] 0.002990201
```

```r
# Total PE, Fast Alg Crit, One-at-a-Time Alg Crit
c(TPE_fast, c1_fast, c2_fast)
```

```
## [1] 2820354.53   44460.74   94059.71
```

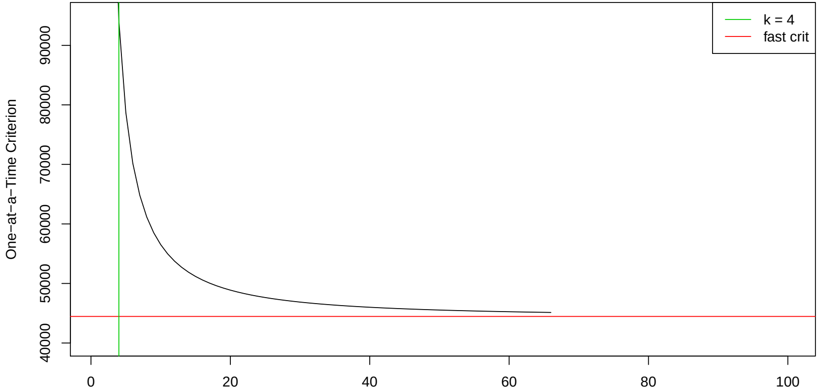
# Over the K Designs




- Total Potential Energy criterion starts to increase as the design approaches `K = 20`. It could be that `K = 20` is too large
- Variance on Slope ($\hat{\beta}$) also increases. However, this makes sense since it is mainly a function of the differences between design points and their mean.

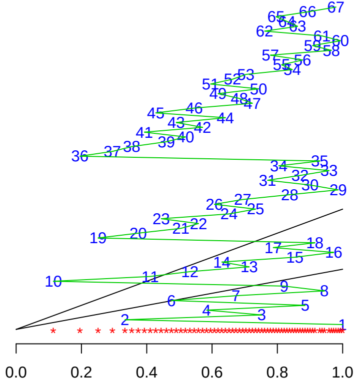
# One-at-a-Time Criterion, Different k Powers

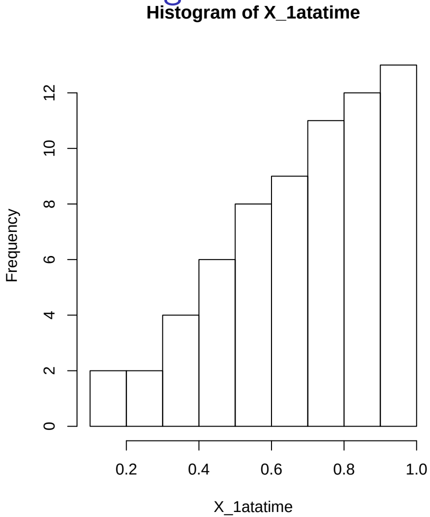
- To see if it decreases as $k \to \infty$, since this algorithm is supposed to be the asymptotic result of the one-at-a-time algorithm, here are the results for $k = 1 : 100$.
- For the fast algorithm's design, the one-at-a-time algorithm's criterion approaches the fast algorithm's criterion as k gets large (until it becomes `infty`. Compuational issue, probably.)



# One-at-a-Time Algorithm

# Design generated by One-at-a-Time Algorithm




```r
mean(X_1atatime)
```

```
## [1] 0.6886419
```

```r
sd(X_1atatime)
```

```
## [1] 0.2187448
```

# Evaluations

```
# Expected Post Probs and BF01 : Y | H0 Sims
c(expect_post_H0_YH0oneattime, expect_post_H1_YH0oneattime,
```

```
## [1] 0.8834295 0.1165705 7.5785023
```

```
# Expected Post Probs and BF01 : Y | H1 Sims
c(expect_post_H0_YH1oneattime, expect_post_H1_YH1oneattime,
```

```
## [1] 0.1148289 0.8851711 0.1297251
```

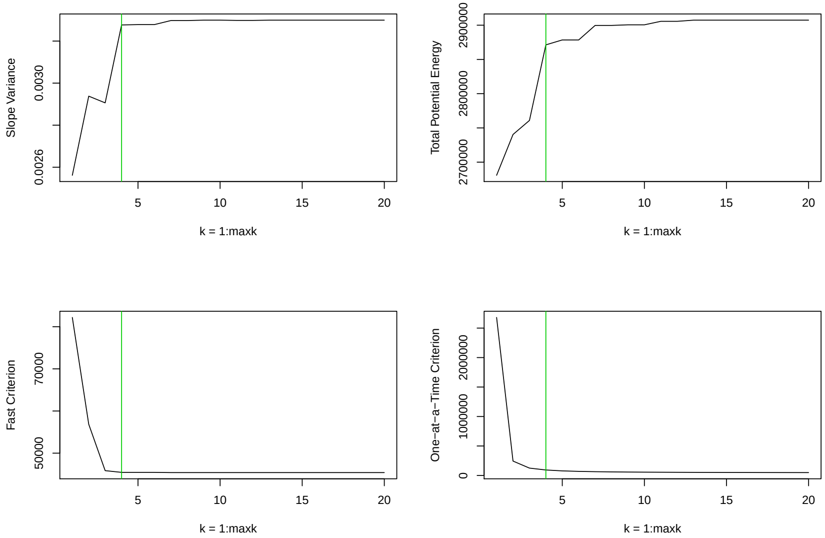
```
# Slope Variance
v_oneattime
```

```
## [1] 0.003300413
```

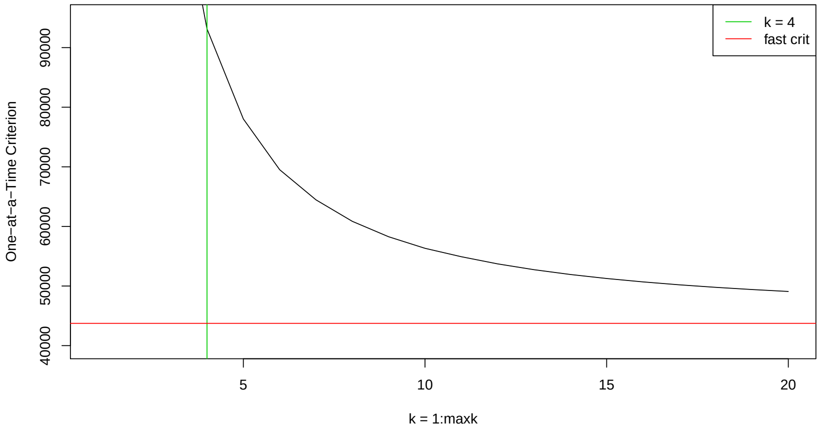
```
# Total PE, Fast Alg Crit, One-at-a-Time Alg Crit
c(TPE_oneattime, c1_oneattime, c2_oneattime)
```

```
## [1] 2867630.36   43737.03   92199.06
```

# Robustness Across k Power



# Comparing Evaluations for Criterion



# Random Design

# Random Designs

10 simulated random designs ($\mathbf{x} \sim U([0,1]^p)$, $\forall \mathbf{x} \in \mathbf{D}_{\text{random}}$).

```
# Mean Expected Post Probs and BF01 : Y | H0 Sims
c(expect_post_H0_YH0rand, expect_post_H1_YH0rand, BF01_YH0rand)
```

```
## [1] 0.7301598 0.2698402 2.7301864
```

```
# Mean Expected Post Probs and BF01 : Y | H1 Sims
c(expect_post_H0_YH1rand, expect_post_H1_YH1rand, BF01_YH1rand)
```

```
## [1] 0.2694990 0.7305010 0.3700613
```

```
# Mean Slope Variance
v_rand
```

```
## [1] 0.001841713
```

```
# Mean Total PE, Fast Alg Crit, One-at-a-Time Alg Crit
c(TPE_rand, crit1_rand, crit2_rand)
```

```
## [1] 955312216 767617960 770812994
```

```
# SD Slope Variance
v_rand_sd
```

```
## [1] 0.0001323766
```

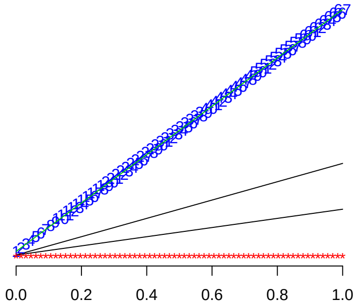
```
# SD Total PE, Fast Alg Crit, One-at-a-Time Alg Crit
c(TPE_rand_sd, crit1_rand_sd, crit2_rand_sd)
```

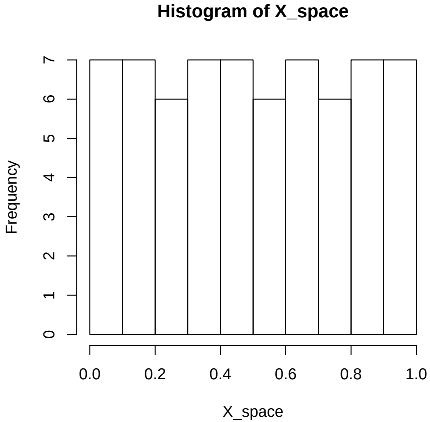
```
## [1] 1575304574 1496514486 1495025103
```

# Space-Filling Design

# The Design Points

Where the points are in no particular order.




```r
mean(X_space)
```

```
## [1] 0.5
```

```r
sd(X_space)
```

```
## [1] 0.2952278
```

# Evaluations

```
# Expected Post Probs and BF01 : Y | H0 Sims
c(expect_post_H0_YH0space, expect_post_H1_YH0space, BF01_YH
```

```
## [1] 0.7370318 0.2629682 2.8027405
```

```
# Expected Post Probs and BF01 : Y | H1 Sims
c(expect_post_H0_YH1space, expect_post_H1_YH1space, BF01_YH
```

```
## [1] 0.2618302 0.7381698 0.3547019
```

```
# Slope Variance
v_space
```

```
## [1] 0.001808684
```

```
# Total PE, Fast Alg Crit, One-at-a-Time Alg Crit
c(TPE_space, c1_space, c2_space)
```

```
## [1] Inf Inf Inf
```

# D-Optimal Design

# D-Optimal Criterion

Seeks to minimize:

$$|(X^T X)^{-1}|$$

i.e. maximize:

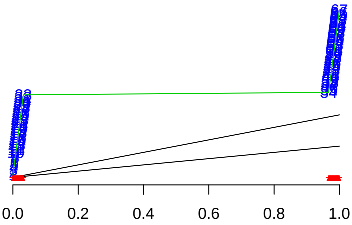
$$|X^T X|$$

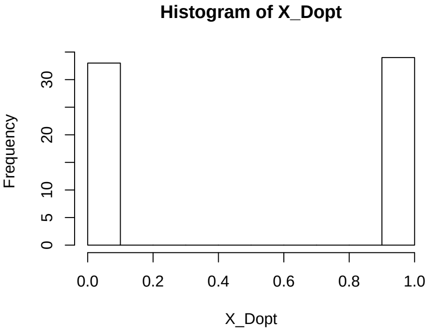
where $X$ is the design.

# Design generated by D-Optimal Criterion

Using `AlgDesign` package (using Federov's exchange algorithm),

where the points are in no particular order... assumed to be random order?




```
mean(X_Dopt)
```

```
## [1] 0.5069697
```

```
sd(X_Dopt)
```

```
## [1] 0.4874279
```

# Evaluations

```r
# Expected Post Probs and BF01 : Y | H0 Sims
c(expect_post_H0_YH0dopt, expect_post_H1_YH0dopt, BF01_YH0c
```

```
## [1] 0.6703381 0.3296619 2.0334113
```

```r
# Expected Post Probs and BF01 : Y | H1 Sims
c(expect_post_H0_YH1dopt, expect_post_H1_YH1dopt, BF01_YH1c
```

```
## [1] 0.3296886 0.6703114 0.4918439
```

```r
# Slope Variance
v_dopt
```

```
## [1] 0.0006681488
```

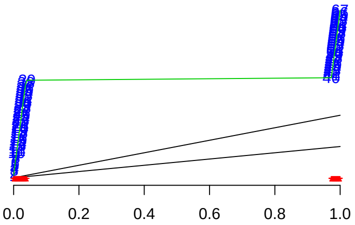
```r
# Total PE, Fast Alg Crit, One-at-a-Time Alg Crit
c(TPE_dopt, c1_dopt, c2_dopt)
```

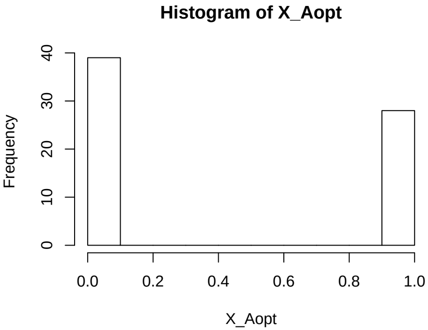
```
## [1] Inf Inf Inf
```

...

# Design generated by A-Optimal Criterion

Using `AlgDesign` package (using Federov's exchange algorithm),




```r
mean(X_Aopt)
```

```
## [1] 0.4233935
```

```r
sd(X_Aopt)
```

```
## [1] 0.4808294
```

# Evaluations

```
# Expected Post Probs and BF01 : Y | H0 Sims
c(expect_post_H0_YH0aopt, expect_post_H1_YH0aopt, BF01_YH0a
```

```
## [1] 0.6327927 0.3672073 1.7232576
```

```
# Expected Post Probs and BF01 : Y | H1 Sims
c(expect_post_H0_YH1aopt, expect_post_H1_YH1aopt, BF01_YH1a
```

```
## [1] 0.3670183 0.6329817 0.5798245
```

```
# Slope Variance
v_aopt
```

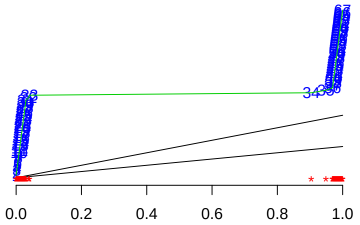
```
## [1] 0.000692498
```

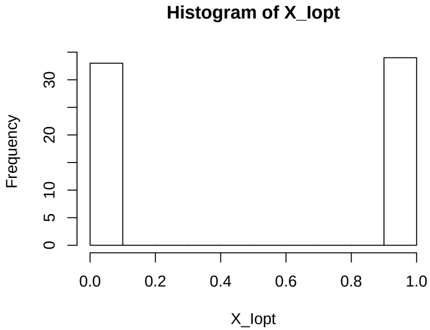
```
# Total PE, Fast Alg Crit, One-at-a-Time Alg Crit
c(TPE_aopt, c1_aopt, c2_aopt)
```

```
## [1] Inf Inf Inf
```

# Design generated by I-Optimal Criterion

Using `AlgDesign` package (using Federov's exchange algorithm),




```r
mean(X_Iopt)
```

```
## [1] 0.5059836
```

```r
sd(X_Iopt)
```

```
## [1] 0.4860835
```

# Evaluations

```
# Expected Post Probs and BF01 : Y | H0 Sims
c(expect_post_H0_YH0iopt, expect_post_H1_YH0iopt, BF01_YH0i
```

```
## [1] 0.6703644 0.3296356 2.0336530
```

```
# Expected Post Probs and BF01 : Y | H1 Sims
c(expect_post_H0_YH1iopt, expect_post_H1_YH1iopt, BF01_YH1i
```

```
## [1] 0.3296856 0.6703144 0.4918372
```

```
# Slope Variance
v_iopt
```

```
## [1] 0.0006717665
```

```
# Total PE, Fast Alg Crit, One-at-a-Time Alg Crit
c(TPE_iopt, c1_iopt, c2_iopt)
```

```
## [1] Inf Inf Inf
```

# The Table

# Results!

| | Fast | 1atTime | Random | Space | D-opt | A-opt | I-opt |
|---|---|---|---|---|---|---|---|
| H0 \| Y, H0 | 0.8745 | 0.8834 | 0.7302 | 0.737 | 0.6703 | 0.6328 | 0.6704 |
| H1 \| Y, H0 | 0.1255 | 0.1166 | 0.2698 | 0.263 | 0.3297 | 0.3672 | 0.3296 |
| BF01 \| Y, H0 | 6.966 | 7.579 | 2.73 | 2.803 | 2.033 | 1.723 | 2.034 |
| H0 \| Y, H1 | 0.1232 | 0.1148 | 0.2695 | 0.2618 | 0.3297 | 0.367 | 0.3297 |
| H1 \| Y, H1 | 0.8768 | 0.8852 | 0.7305 | 0.7382 | 0.6703 | 0.633 | 0.6703 |
| BF01 \| Y, H1 | 0.1406 | 0.1297 | 0.3701 | 0.3547 | 0.4918 | 0.5798 | 0.4918 |
| Var Slope | 0.00299 | 0.0033 | 0.001842 | 0.001809 | 0.0006681 | 0.0006925 | 0.0006718 |
| TPE | 2820000 | 2868000 | 955300000 | Inf | Inf | Inf | Inf |
| Fast Crit | 44460 | 43740 | 767600000 | Inf | Inf | Inf | Inf |
| 1atTime Crit | 94060 | 92200 | 770800000 | Inf | Inf | Inf | Inf |
| Mean(D) | 0.6837 | 0.6886 | NA | 0.5 | 0.507 | 0.4234 | 0.506 |
| sd(D) | 0.2297 | 0.2187 | NA | 0.2952 | 0.4874 | 0.4808 | 0.4861 |

- Fast & One-at-a-Time Algorithms have highest expected Bayes Factors (when the $H_0$ is true, lowest when $H_A$ is true), and hence are better for testing.
- They also have higher variance on $\hat{\beta}$, though, which means they are not as accurate in estimating $\beta$. As expected, the $D$-optimal design is best for estimation.
- Also noticed values of `Inf` for the space-filling and $D$-optimal designs in the evaluations of each of the 3 criteria. This is to be expected in the $D$-optimal designs, since they include 0 which has gives as Wasserstein distance of 0 (in the denominator). Underestimated these by using maximum evaluation for $q$ over $i, j$, instead. $D$-optimal design - though best for estimating, is worst in criteria evaluations, as a consequence.