

Chapter 3

Alan R. Vazquez

Exercises

```
import numpy as np
```

Exercises

1. In Exercise 4.3 you have studied the ECs of designs with 16 runs and six two-level factors. We cordially invite you to complete the characterization of the arrays in this series. Here are a few functions to help:
 - `oaf.ECIC` (already encountered) calculates the estimation capacity sequence or the information capacity sequence of a set of pure-level orthogonal arrays. The inputs are: (1) a tuple of `array_link` objects; (2) a scalar with the minimum number of interactions (default 1); a scalar with the maximum number of interactions (default 3) and (4) the kind of sequence required (either 'estimation' or 'information'; default 'estimation'). Output: a `numpy` array with EC or IC vectors for every orthogonal array. The first line of this array is a header showing the numbers of interactions in the models that were evaluated.
 - `oaf.PECPIC` calculates the projection estimation capacity sequence or the projection information capacity sequence of a set of pure-level orthogonal arrays. The inputs are: (1) a tuple of `array_link` objects; (2) a scalar with the minimum number of factors to project into (default 3); a scalar with the maximum number of factors to project into (default 4) and (4) the kind of sequence required (either 'estimation' or 'information'; default 'estimation'). Output: a `numpy` array with PEC or PIC vectors for every orthogonal array. The first line of this array is a header showing the dimension of the projections that were evaluated.
 - `oaf.GWLP`. Input: a tuple of `array_link` objects. There are two outputs: (1) a table with one line per design stating the design number, the rank of the interaction model matrix, including the intercept, A_3 , A_4 , F_3 , and F_4 and (2) a matrix with the same

information. The table has an informative header, while the matrix can be used for subsequent calculations.

- Load the designs and characterize them using all criteria discussed. Indicate which designs are regular.
- Sun, Wu, and Chen (1997) proposed the concept of *admissible designs* to reduce the number of designs that merit further consideration. A design is inadmissible if there is another design that is strictly better according to one criterion and at least as good according to the other criteria. Otherwise, it is admissible. Discuss what steps are needed to decide on admissibility of the $OA(16, 2^6, 2)$ according to the above criteria.

The function `oaf.Pareto2lev` determines admissible arrays from a tuple of `array_link` objects. By default, it bases admissibility on F_3 , F_4 , A_3 , A_4 , the rank of the interaction model matrix (including an intercept, all the main effects and all the interactions), and the PIC vector. You can switch off any of these criteria by setting the appropriate switches to 'no'. For example, the following line of code evaluates the tuple A ignoring F_4 and the rank of the interaction model matrix:

```
# Pareto2lev(A, f4='no', rank='no')
```

- Which of the designs are admissible? Check the criteria for these designs to see that none of the designs dominates the others.
2. The six-factor designs studied in the last few exercises may or may not be to your liking. It is always a good idea to consider design alternatives with other run sizes. For this purpose, we collected the 75 different $OA(20, 2^6, 2)$ in an ASCII file called `result-20.2-2-2-2-2-2.oa` and the 1350 different $OA(24, 2^6, 2)$ in an ASCII file `result-24.2-2-2-2-2-2.oa`.
 - Collect the top-3 designs in 20 runs according to G-aberration. For this purpose, sort the appropriate columns of the output matrix of `oaf.GWLP()`, `Gmat`, say, with the function `np.lexsort(col1, col2, ...)` (We import `numpy` as `np`). Characterize the designs with the tools introduced in Exercise 5.1. Does it pay off to have a few extra runs?
 - The 24-run series include two strength-3 designs. Observe that these designs do not permit fitting the full interaction model.
 - Restrict your attention to the 24-run designs that do permit fitting the interaction model. Equation 5.1 suggests that for the special case of models that can estimate all the two-factor interactions, $1/D \propto A_3 + A_4$. Check this relation for the 24-run cases. Also check the relation between D and A_3 . State a conclusion about aberration in designs that permit estimation of a full interaction model (all the main effects and all the two-factor interactions).

- Can there be reasons to prefer 16-run or 20-run designs to the 24-run designs that can fit the interaction model?
- 3. *Even* arrays have $A_{2q+1} = 0$ for any positive integer q . *Even/odd* arrays have $A_{2q+1} > 0$ for at least one positive integer value of q . The distinction is relevant for strength-3 arrays, as for strength-2 arrays $A_3 > 0$. In this exercise you are invited to explore the difference in statistical properties for these two types of array using the 353 OA(40, 2^{10} , 3) as examples; see `result-40.2-2-2-2-2-2-2-2-2-2.oa`.
- Consider a 20-run design that is folded over. So, in terms of the main-effect contrast vectors, every +1 in part 1 becomes a -1 in part 2 and every -1 in part 1 becomes a +1 in part 2. Prove that the resulting 40-run design is an even design. In general, every foldover design is an even design. Cheng, Mee, and Yee (2008) show that the reverse is also true: every even design must be a foldover design.
- Show that even designs have at most $N/2 - 1$ degrees of freedom available to estimate two-factor interactions.
- Use the foldover property to separate the 353 OA(40, 2^{10} , 3) into 351 even and 2 even-odd arrays. First, turn an array into a numpy array with, say, `npOA = np.array(OA)`. Then transfer the first 20 rows to some array, and do likewise with the reverse of the second 20 rows using

```
# h2 = np.flip(trdes[20:, :], axis=0)
```

In case the original array is a foldover array, each row of the first half is the mirror of the corresponding row in the reverse of the second half. Use this fact for your determination.

- Select the Pareto-optimal even designs; augment the set with the two even-odd designs and compare their properties. Tip: the function `oaf.GWLP()` has a switch called **strength**; specify that the strength equals 3.
- For OA(40, 2^{10} , 3), $J_5(s)$ for any five-factor set s can have values 0, 16 or 32. Check that the even designs indeed have $F_5 = 0$.
- How could you employ the available degrees of freedom in the even/odd arrays?
- Are there inadmissible designs in the set?
- Discuss how you could employ the fold-over feature of the even designs in the analysis of the data.

References

- Cheng, Ching-Shui, Robert W. Mee, and Oksoun Yee. 2008. "Second Order Saturated Orthogonal Arrays of Strength Three." *Statistica Sinica* 18: 105–19.
- Sun, D. X., C. F. J. Wu, and Y. Chen. 1997. "Optimal Blocking Schemes for 2^n and 2^{n-p} Designs." *Technometrics* 39: 298–307.