What did I do today?

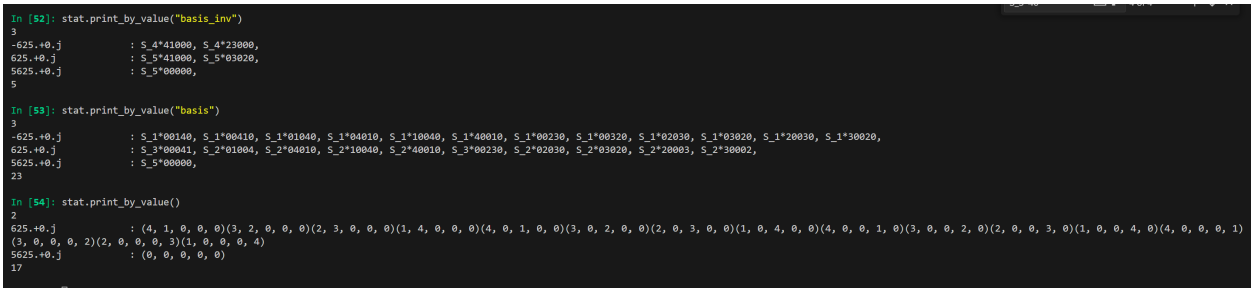
Recently I found a way of iterating through permutations in python, so I wanted to use that to create an implementation of the nice basis mentioned in 6/12 and 6/14 write up.

Now I have code implemented to find the nice basis and can print outputs for the decomposition of any of the statistics that I have implemented.

This decomposition compresses the information a lot more. For the statistics in the same family as lucky, each non-zero coefficient seems to appear only once, and the number of coefficients is close to the number of distinct coefficients for the standard module decomposition as a direct product of cyclic groups (Ex: for $n = m = 5$ and lucky, decomposed using a direct product of cyclic groups, there are 625 non-zero coefficients, but only 94 distinct coefficients. Using the modified basis, there are 99 coefficients with 99 distinct values).

This basis also captures the way that maximum displacement seems like it should be a higher frequency function. (for $n = m = 5$ for the modified basis there are 375 nonzero coefficients with 305 distinct values while for the standard C_n^m decomposition there are 625 nonzero with 375 distinct values)

There are some statistics which mesh better with an inverted version of the nice basis that we looked at (like leading elements). For example, leading elements $n = m = 5$, I ran the code for 3 different bases. One is the standard C_n^m basis, one modified basis where rearrangements matches up well with the one required for lucky, and one modified basis where the rearrangement starts on the other side.



There also seem to be some statistics that compress a fair amount with use of the modified basis that were not originally compressible. This could be explored for statistics having to do with passed. In particular, one of the passed statistics should correspond to defect, which would be interesting to look at. passed_i[n-1] I think

Lots of things that are easy to poke around and look at now

Goals for tomorrow?

- properly read Ian's reply to my random question
- fill in the hole: why is there exactly one prime parking function per coset of the diagonal subgroup of C_{n-1}^m?
- work on writing things down in random sampling paper/outline

- work on an oeis entry outline/formatted thing
- If $n \neq m+1$, is there a way to extend the connection between S_m module decomposition of parking functions, and the number of distinct non-zero Fourier coefficients for a statistic on elements of C_n^m that is constant on cosets of $S_m D$