Notes from conversation with Prof O

- Recall that you already have proved why functions that are constant on cosets of the diagonal subgroup have only n^{m-1} nonzero coefficients. See the section "Decomposing Circular Parking functions" of write up on 6/12
- Think about what would happen if you look at sine distribution on the first element and uniform on all other elements
  - Fourier transform of sine just involves $\chi_0, \chi_1, \chi_{n-1}$.
  - When you chronicer these with the trivial representation for the other coordinates, you only get 3 non-zero coordinates
  - The only one that matters convolution with statistics that are constant on diagonal coset is the trivial representation. This tells you that no matter which shifted version of the sine you use, you get the same thing.

Thinking through things in detail

Why does the Fourier transform of a sine wave just involve $\chi_0, \chi_1, \chi_{n-1}$?

I messed around with a few computations to convince myself of this.

Let us consider decomposing a function on C_n into irreducible characters. Fix some value for ϕ. Given the function

$$f_{n,k}(x) = \frac{1}{n}\left(1 + \cos\left(\frac{2k\pi}{n} + \phi\right)\right)$$

The only nonzero coefficients in front of characters are $\chi_0, \chi_{k(\text{mod}n)}, \chi_{n-k(\text{mod}n)}$. Note this describes some of the weird sampling interference behaviors for very high frequency signals sampled at a lower frequency, and also some of the "resonance" like aspects.

By messing around with computations, I suspect that the explicit expressions for the fourier coefficients are the following (or off by a conjugate)

$$\begin{aligned}
\text{For } \chi_0 :& 1 \\
\text{For } \chi_k :& \frac{1}{2}e^{-i\phi} \\
\text{For } \chi_{n-k} :& \frac{1}{2}e^{i\phi}
\end{aligned}$$

If 2 of the coefficients overlap and have the same value (for example when $k = n$, just add together the coefficient values given above)

I could likely prove these things with inner products, but that's probably already been done a lot since this is the standard discrete fourier transform, so meh. I have a feel for why this is true now though

So say I want to know the expected value of a statistic s on a probability distribution f. Then I am interested in the element wise product of these two elements in the group, which translates to the convolution of the characters in the context of abelian groups.

Say f is a sine wave distribution, and S is a compressable statistic (like total displacement, lucky, max displacement, other quesetions about the distribution of displacements, repeats, leading elements, number of break points, longest prime parking function component)

Then in fourier space there are 3 nonzero coefficients (indexed by $0, 1, n-1$)for a sin wave distribution and $k \leq n^{m-1}$ nonzero coefficients for the compressible statistic. So to understand the point wise product of the two, you just need to understand the $3k$ nonzero coefficients after convolution. There will be exactly $3k$ nonzero coefficients since k characters with nonzero coefficients will be indexed by group elements summing to 0, another set of k nonzero coefficients will be indexed by group elements summing to 1, and the last set of k nonzero coefficients will be indexed by group elements summing to $n-1$

testing both my code and understanding gives the following:

```
In [128]:test.set_stat(stats.total_disp)
In [129]: sin = CnmStat(7,7,[1,.5,0,0,0,0,.5] + [0]*(7**7-7), "fourier")
In [130]: res = sin.times(test)
In [131]: c = 0
     ...: for idx, x in np.ndenumerate(res.fourier):
     ...:     if not(np.isclose(x, 0)):
     ...:         c+=1
In [132]: c == 7**6*3
Out[132]: True
```

On the other hand, if you start with a preference list drawn uniformly at random from a compressible distribution (like total displacement 1 preference lists or unit interval preference lists or the set of permutations), and do a random walk from these preference lists, then you would convolve the two distributions, which is point-wise multiplication in fourier space. This means that the only nonzero coefficient left is the trivial character.

Perhaps this shouldn't be too surprising. Since using the sine distribution kroneckered with uniform distributions is randomizing a lot of things for a random walk

I worked on refining my code to be able to calculate convolutions and pointwise products easily

To Do next

Updating code

- update the Demos file with some demos of fourier stuff
- add more statistics to the IterateStats class, update names in Park class to actually match things of interest

- add sampling from parking functions, sampling from prime parking functions, and sampling from defect 1 preference lists to sampling
- have code to make plots from random walks

Random walks

- Think on how the localness of statistics that we have could help. Sample from one distribution d, do a random walk with another w, evaluate at some particular statistic s. The thing you care about is $(d * w^{*t}) \cdot (s)$ where $*$ is convolution and $\cdot$ is point wise multiplication (reversed in fourier transform world).