June 3, 2021 Meeting Agenda

June 2, 2021

1 Joint estimation of μ_p and μ_t

I finished implementing the ad hoc algorithm. I want to confirm that the exclusion criteria that we are currently using is the one we want. Here are the exclusion criteria:

Table 1: Plea Capacity Estimation Exclusion Criteria

Condition	**Old Clean Day**	**New Clean Day**
No inconsistencies between sentencing data and calendar	Y	Y
Judge has at least 10 'clean' days	Y	Y
Judge has at least one sentencing event that day	Y	
Judge has no trials in this county in entire dataset	Y	
Judge is only assigned to one county that day		Y
Judge only sentences in one county that day		Y
Judge never has more than x sentencing events in this county		Y
Judge calendar assignment is of type "GS"		Y

Table 2: Trial Capacity Estimation Exclusion Criteria

Condition	**Old Clean Day**	**New Clean Day**
Judge never has more than x sentencing events in this county	Y	Y
Judge has at least 1 trial in this county	Y	Y

1.1 Problems

- Step 4 of the ad-hoc algorithm (the part where we maximize log likelihood) is inexact. The gradient is usually not exactly equal to zero, but it does get smaller over iterations. In the last iteration it is at 0.002.
- I'm not actually using a stopping criteria for Step 4, I just run it for 100 iterations and keep the best value of μ_p. I played around with the number of iterations, and the objective doesn't seem to improve after 100.

1.2 Next Steps

- Finalize exclusion criteria.
- Do Poisson tail calculations.
- Add information about optimizer to document.

1.3 Pseudocode

Algorithm 1: Ad-hoc Algorithm

```
input  : InitialMuT, InitialMuP, tolerance
output: muP,muT
sdf = LoadSentencingData();
pleas = GetCleanDayPleas();
judgeDays = GetJudgeDays();
prevMuT = 0;
prevMuP = 0;
muT = InitialMuT;
muP = InitialMuP;
while (|muT − prevMuT| > tolerance) or (|muP − prevMuP| > tolerance) do
    prevMuP = muP;
    prevMuT = muT;
    muP = OptimizeMuP(prevMuT,prevMuP,sdf,judgeDays,pleas);
    muT = EstimateMuT(muP);
end
```