July 2

**Desk**

In GEE, work on validating Timeseries Analysis v7 in GEE using Matopiba survey (Timeseries Validation w Matopiba)

**Meeting**

For validating plant/harvest dates with Matopiba, maybe try a bunch of different plant/harvest date estimates to come up with an estimated range, then compare the overlap of the estimated range with the reported range

For qual, propose the following: (1) plant/harvest date estimation, (2) analyzing plant/harvest date estimation as proxy for farmer behavior over history, (3) something with the statistical crop model.

July 3

**Desk**

In Timeseries Analysis v7, added Jake’s new land cover map to replace lc\_morgen. Also, need to “optimize” the best cutoff date for separating single vs double cropping. Do this by calculating area before and after I filter for “quality” (i.e. quarter periods). First export the images from Timeseries Analysis v7, then test it in Timeseries Validation of Quality mask in GEE.

In Timeseries Validation w Matopiba, add functionality to check if the estimated peak date is between the reported planting and harvest date. For 2017, do this for both the first and second crop peaks. This uses the output of Timeseries Analysis v7

In Timeseries Produce Final Asset in GEE, produce final asset to share out of the individual result images from Timeseries Analysis v7

July 4

**Desk**

Work on Timeseries Produce Final Asset in GEE – use it to produce single and double crop image results, share. Include with the shared assets a readme describing method and validation steps.

Varied cutoff time and produced % pixels with bad quality estimates using Timeseries Validation of Quality mask. Results are tabulated in Optimizing Timeseries Analysis Cutoff Date.xlsx.

Attempted to create binary rasters for whether a single/double cropped pixel has good or bad quality estimates; but the second peak in double cropping is sometimes null, which is ok for the % pixels with bad quality estimate stuff because the nulled pixels are badly classified, but not ok for producing a binary raster because we want those to be 0, rather than masked out.

Began Timeseries Analysis v8 to work out the nulled second peak pixels.

July 5

**Desk**

Continue using Timeseries Validation of Quality mask to “tune” cutoff dates; results in Timeseries Analysis Cutoff Date.xlsx

Update Timeseries Produce Final Asset in GEE to create binary quality images for all years, single and double cropped – produce and share quality masks to go with time estimate images.

July 6

**Desk**

Begin summary stats of crop timing map results and performance in GEE file “Crop Timing Statistics”

July 9

**Desk**

Summary stats of crop timing map results and performance over years and municipalities in GEE file “Crop Timing Statistics v2” – add in functionality to display multiple years at the same time.

July 10

**Desk**

Started “Aggregate Crop Points” in GEE to take a bunch of soy points and output a table of averaged properties over each municipality-year.

July 11

**Desk**

Re-run Timeseries Analysis v8 with Jake’s updated land cover map. Add subscript ‘\_2’ to the end of each timing and binaryQuality map in case. Also make an unmasked version in case land cover needs to be change again.

In Aggregate Crop Points, to export asset, probably need to add in geometry.

**Meeting**

If the points that Jake produces (with soy points associated with weather, etc) have onset, then I can use them for my own causal inference stuff.

For comparing Matopiba plant/harvest data to my crop timing map, take CAR polys of fields, estimate plant/harvest for each pixel in the field to get a range of plant/harvest dates in the field polygon, and then compare the Matopiba range to the crop timing range.

July 12-13

**Desk**

Finish visualization of crop timing (Crop Timing Statistics v2)

July 16

**Desk**

Work on transitioning readmes in crop\_timing on GitHub to markdown files

Look at Jake’s R code called temperature\_sample.R, try to run it. R versions probably need to be changed to comply with packages and with my OS version.

Begin looking at putting R code into Jupyter VM. Look at JupyterLab and Jupyter documentation and tutorials.

July 17

**Desk**

Learned about Jupyter notebooks; set up R on my Jupyter notebook

Used conda and VM terminal to install packages needed for Jake’s R code temperature\_sample.R

Figured out how to export csv as output from script

Changed Jake’s original script to no longer need a package that can’t be installed with conda

July 18

**Desk**

Try running temperature\_sample.ipynb in Jupyter Lab

Re-start work on plant/harvest date estimation with GEE script Timeseries Validation w Matopiba v2 (the first version calculated the timeseries from scratch but this makes code super long to read and probably to run)

July 19

**Desk**

Submitted qual committee form on CalCentral

In GEE, “timeseries validation w matopiba v2” to try plant/harvest date estimate methods and see if they fit in the reported matopiba range – and summarize, per year and method, the percent of methods that give estimated date within the reported range

July 20

**Desk**

Worked on GEE “timeseries validation w matopiba v2” to summarize plant/harvest date estimate errors and develop ui to chart each point’s EVI timeseries overlaid with actual plant/harvest range and estimated plant/harvest dates

July 23

**Desk**

Used GEE “Timeseries Validation w Matopiba v2” to look at specific badly estimated points, look at why they’re badly estimated, and adjust estimation method for specific points. Note, this version doesn’t compare all pixels in a CAR poly to the reported range; only a single point.

July 24-25

**Desk**

Used “GEE timeseries Validation w Matopiba v3” and v4 to compare estimated range to actual range in plant/harvest estimates. Explore two measures to compare ranges and look at both the full range and the interquartile range

July 30

**Desk**

In “GEE timeseries Validation w Matopiba v4” to look at estimate performance if we mask out pixels with bad quarter period – but here, only one polygon was taken out per year because only a polygon with absolutely no okay pixels is completely taken out.

July 31

**Desk**

In GEE “timeseries validation w Matopiba v5”, added a chart to overlay percentiles of pixel estimates within each polygon and the plant/harvest date reported – can look at spread and goodness of estimates within each polygon. Had to go with percentiles instead of individual values because otherwise there are too many points to plot.