42.0021 percent is the overall accuracy of subzone classification across all locations that were used as modeling plots in the most recent GNN modeling. It ended up taking a bit more logic than I had originally thought, but I'll try to be as explicit as possible to help facilitate writing this up in the methods section.

\* There are n=17,564 unique locations (LOC\_IDs) across Washington, Oregon,

 and California used in GNN modeling which represent n=26,719 plots (FCIDs).

 Plots are all based on the FIA annual design (4-point) and were measured

 between 2000 and 2016. At some point, we will probably want to break this

 down by LOC\_IDs per geographic region or by subzone assignment, but for now

 I'm treating it as a single pool. Because we may have used the same plot in

 multiple modeling regions (given a 10-km buffer), I am only calculating

 the predicted in the model region where the plot actually is.

\* Observed subzone values come from the lookup that Mike Simpson and the PNV

 mapping team created. The lookup dated 2021.02.22 was used. Locations have one and

 only one subzone assigned to them and all plots at that location go into

 the determination of subzone across the measured time period. Note that

 I believe Mike used plots other than FIA annual style plots (e.g. R6 CVS)

 to help guide this assignment. I verified that there was only one subzone

 call per location in my code.

\* To create the predicted subzone assignment for each location, I first had

 to retrieve the nearest neighbor assignments associated with the 3x3 30m

 pixel window at each location for all years, 1986-2017. The window is

 centered on the plot location which is first snapped to a NN pixel center.

 For each year, I extracted the nearest neighbors (FCIDs) for the first

 seven neighbors and for all nine pixels.

\* To preserve independence, a plot may not self-assign to any pixel in its

 footprint, nor may any other plot from the same location. I filtered the

 k=7 neighbor list for each pixel to remove any self-assignment and used

 the nearest remaining neighbor. Commonly, the first nearest

 neighbor is itself and the second neighbor is typically an independent

 assignment (this is why we sometimes refer to this as the second nearest

 neighbor accuracy assessment, but in truth it's the first \*independent\*

 neighbor that we use to represent the pixel).

\* At this point, we have a time-series of 32 (1986-2017) FCIDs for each pixel

 in a plot footprint. The first step is to crosswalk the FCIDs to subzone

 using the same lookup as above (although keyed off on FCID instead of LOC\_ID).

 To get to a single subzone call for the pixel, I implemented Mike's logic:

 \* Find the majority (mode) subzone across all 32 observations

 \* Find the number of distinct subzone assignments across all 32 observations

 \* Crosswalk each subzone assignment to a rank determined by your group

 and find the minimum rank across all 32 observations. Then crosswalk

 the rank back to its corresponding subzone.

 \* If the number of distinct subzone assignments is >= 4, use the majority

 subclass, otherwise use the subzone with minimum rank

\* Finally, I took the majority (mode) across all nine pixels in the location

 footprint to represent the predicted subzone.

\* Lastly, I merged the observed and predicted subzone assignments on LOC\_ID and

 created the error matrix (attached). Observed classes are rows and predicted

 classes are columns. I left them as subzone codes for now. I'm also sending

 along the raw file of observed and predicted subzone by location.

In the next step of the accuracy assessment, we will calculate fuzzy confusion matrices for subzone as well as strict and fuzzy confusion matrices for vegzone.  The final bit will be comparing area estimates of subzone/vegzone from plot data vs map data vs error-adjusted area estimates (Olofsson technique).  I committed to finishing this by the end of August [2021].

 --Matt Gregory

 Oregon State University