GPS Preprocessing

This is the first step in creating a configuration file for an Agent Based Model. The goal of this notebook is to allow the ABM to be configured to accurately represent the GPS dataset used.

The first steps are to filter out GPS points that have a poor DOP, that could result in a poor location lock, and to seperate timeseries trajectories where the time interval is not regular. Several files are created based on the following conditions:

- Deer Season
- Male/Female animals

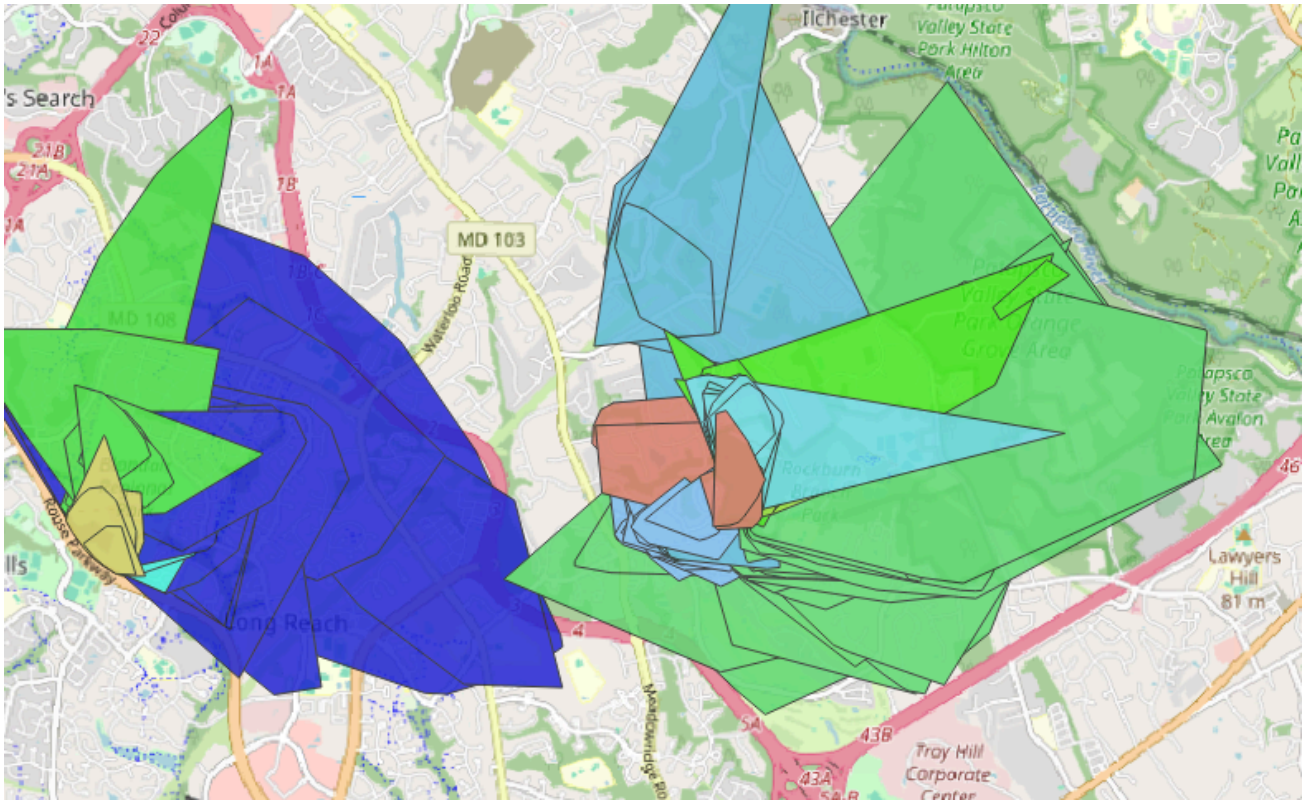
The output of this notebook is passed to another notebook, that uses an R kernel, to fit a movement model to the data.

```
/tmp/ipykernel_124625/1439837228.py:1: DtypeWarning:

Columns (11) have mixed types. Specify dtype option on import or set low_memory=Fals
e.
```

Home Range Centroids

Let's assume that the animals spend most of their time within their home range. The DLD paper takes the average lon/lat position as the centroid to the home range which seems sensible. The data we have spans over years though, and it is likely that a home range will drift or change due to dispersement or population pressures. Let's calculate the monthly convex hull and use that centroid as the home range centroid:

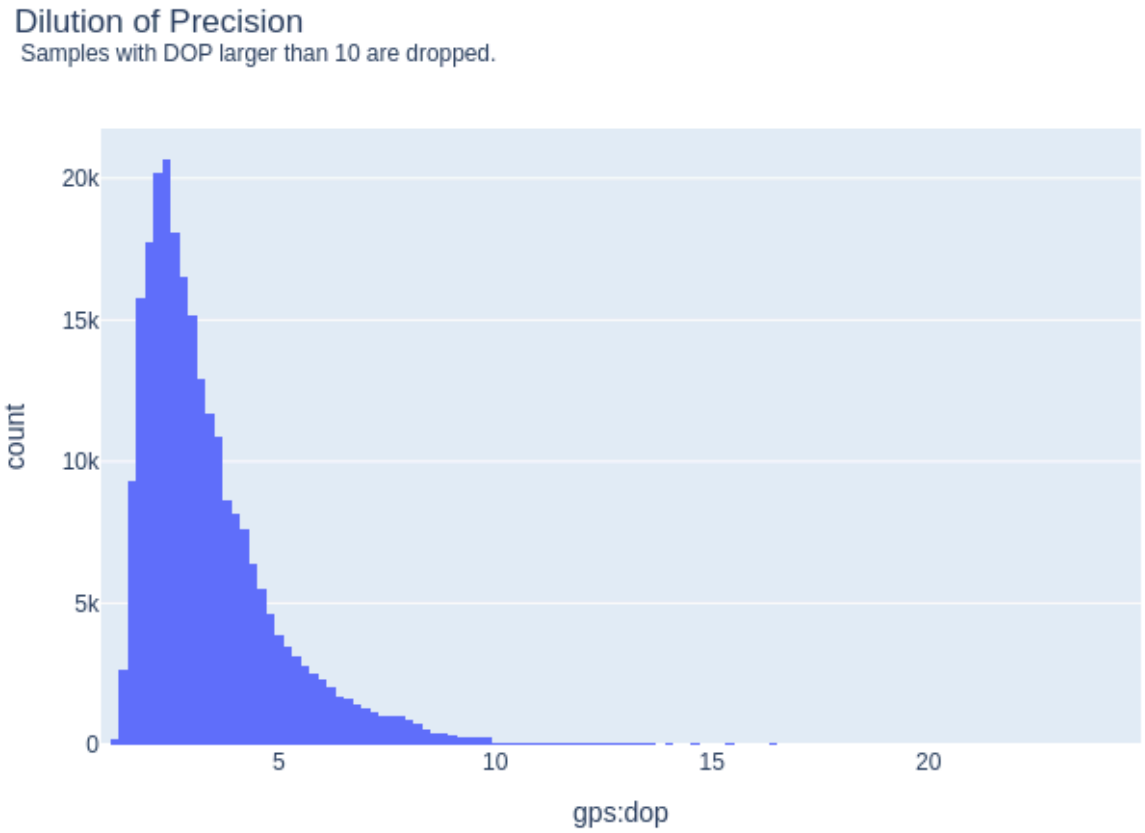


Monthly GPS home ranges coloured by deer ID

Filter GPS Data and create new IDs

Filter the GPS data to remove points with poor GPD DOP. The HMM and trajectory analysis assumes that the data is regularly sampled. We could either resample/interpolate the data to avoid gaps or create new ID's for trajectory segments that are seperated by gaps.

Steps taken: - Calculate a datetime from the seperate columns - Calculate timedelta for consecutive GPS fields - Drop events that have a GPS:DOP greater than 10)

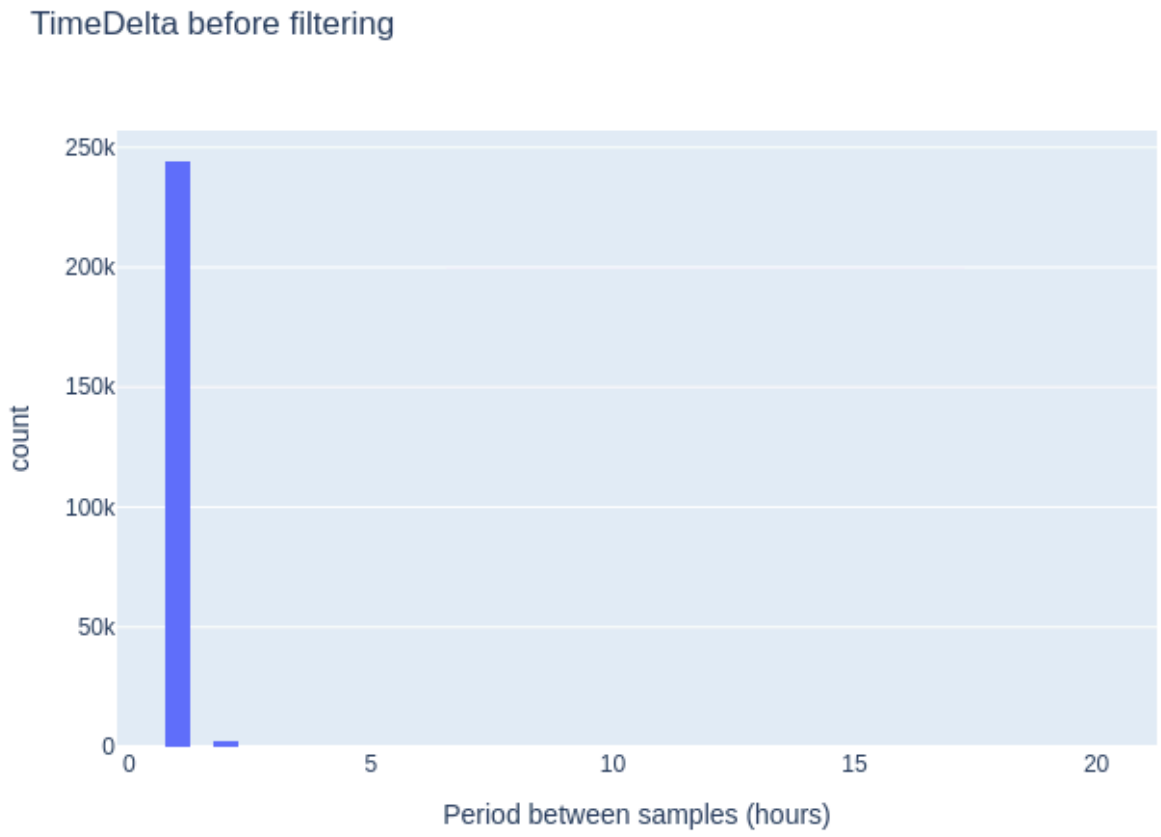



```
/opt/conda/lib/python3.11/site-packages/geopandas/geodataframe.py:1819: SettingWithCo
pyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
```

	ID	Deer ID	timestamp	sex	lat	lon	time_group	centroid	geome
0	399253139925	31.0	2017-02-01 00:00:10	f	39.024816	-76.926295	2017-02	POINT (1623729.656 1956854.626)	POI (1625039.7 1941318.6
1	399253139925	31.0	2017-02-01 01:01:00	f	39.025172	-76.926355	2017-02	POINT (1623729.656 1956854.626)	POI (1625026.7 1941356.5
2	399253139925	31.0	2017-02-01 02:00:54	f	39.025046	-76.926487	2017-02	POINT (1623729.656 1956854.626)	POI (1625018.4 1941340.
3	399253139925	31.0	2017-02-01 03:00:18	f	39.024953	-76.926290	2017-02	POINT (1623729.656 1956854.626)	POI (1625037.1 1941333.7
4	399253139925	31.0	2017-02-01 04:01:00	f	39.024991	-76.926614	2017-02	POINT (1623729.656 1956854.626)	POI (1625009.0 1941332.3



Check for time gaps that are too large

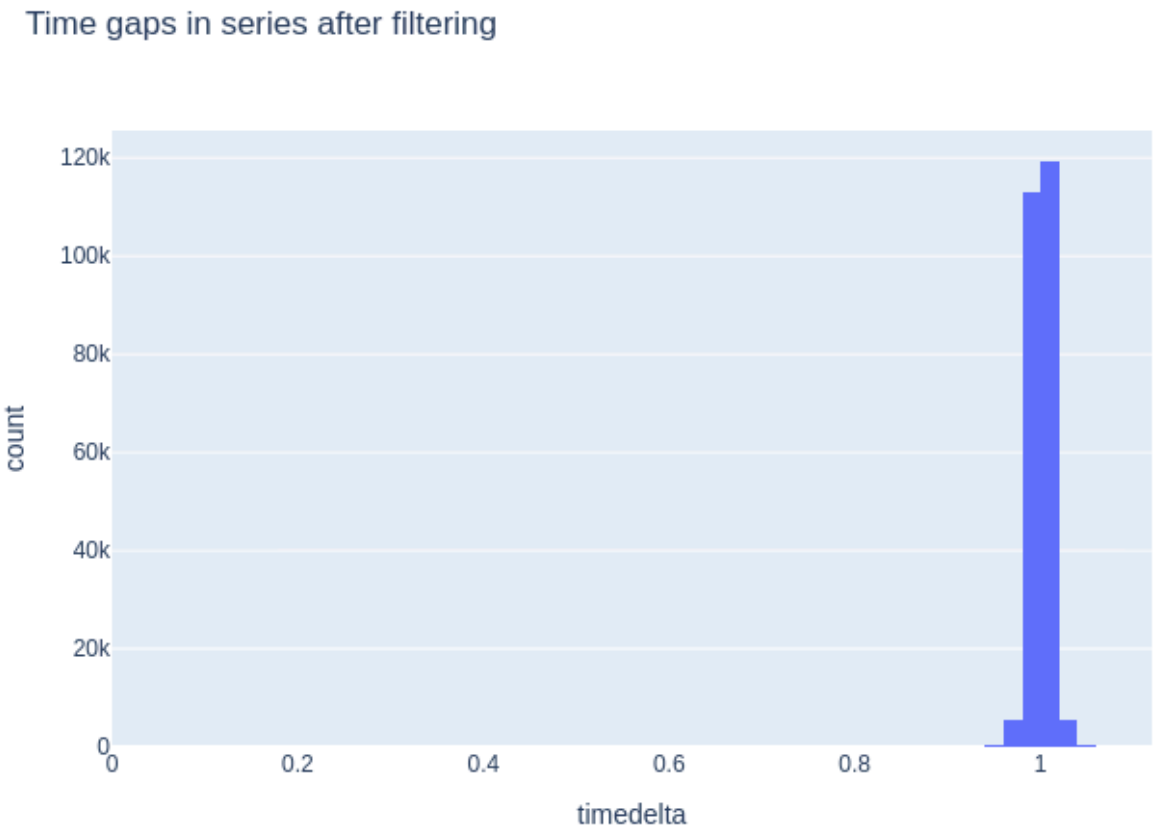
Hidden Markov Models assume that the timeseries is regularly sampled. If it isn't, well, that could cause some problems. This steps checks that the timestamp interval is close to 1 hour, and renames the ID where gaps are seen.

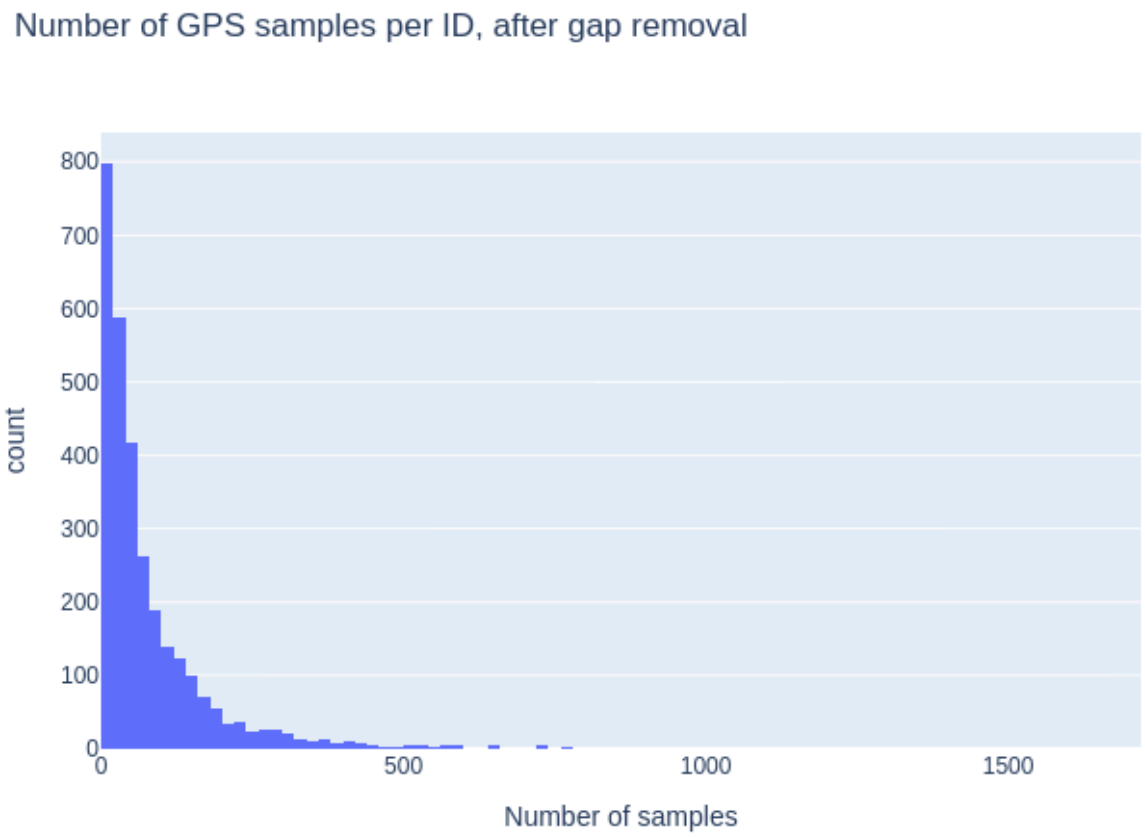
https://stackoverflow.com/questions/40118037/how-can-i-detect-gaps-and-consecutive-periods-in-a-time-series-in-pandas

```
/tmp/ipykernel_124625/1182845987.py:34: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated,
and in a future version of pandas the grouping columns will be excluded from the oper
ation. Either pass `include_groups=False` to exclude the groupings or explicitly sele
ct the grouping columns after groupby to silence this warning.
```

	ID	Deer ID	timestamp	sex	lat	lon	time_group	centroid	geome
0	399253139925	31.0	2017-02-01 00:00:10	f	39.024816	-76.926295	2017-02	POINT (1623729.656 1956854.626)	POI (1625039.7 1941318.6
1	399253139925	31.0	2017-02-01 01:01:00	f	39.025172	-76.926355	2017-02	POINT (1623729.656 1956854.626)	POI (1625026.7 1941356.5
2	399253139925	31.0	2017-02-01 02:00:54	f	39.025046	-76.926487	2017-02	POINT (1623729.656 1956854.626)	POI (1625018.4 1941340.
3	399253139925	31.0	2017-02-01 03:00:18	f	39.024953	-76.926290	2017-02	POINT (1623729.656 1956854.626)	POI (1625037.1 1941333.7
4	399253139925	31.0	2017-02-01 04:01:00	f	39.024991	-76.926614	2017-02	POINT (1623729.656 1956854.626)	POI (1625009.0 1941332.3


Time gaps in series after filtering
120k
100k
80k
60k
40k
20k
0
count
0
0.2
0.4
0.6
0.8
1
timedelta

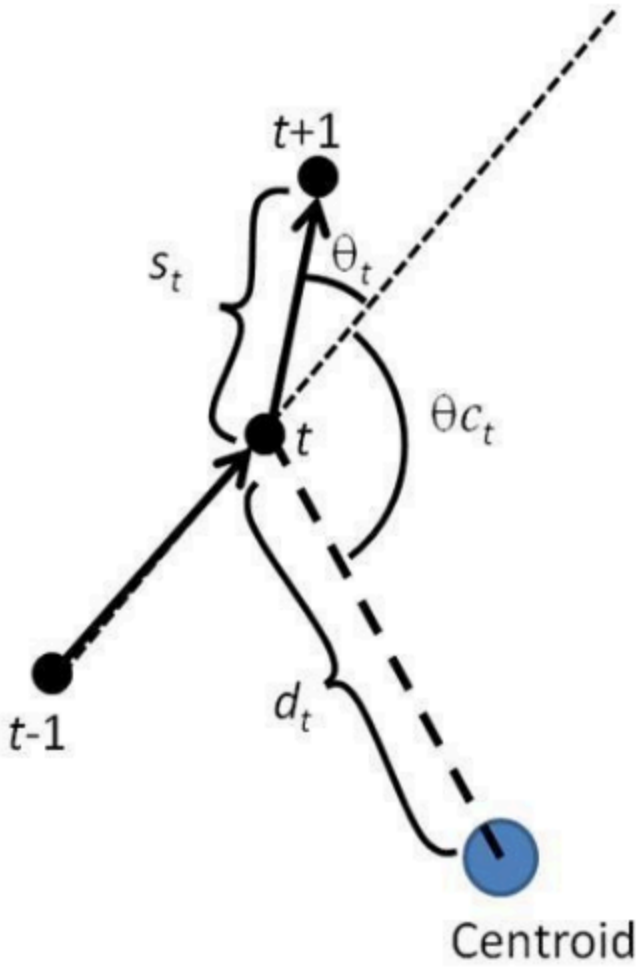


It looks like there are very many new IDs that have fewer than 20 samples... that might not be great.

Steps and Turns

Time to calculate the turning angle and step distance for various different points:

- Distance between current point and previous point
- Distance between current point and various centroids
- Turn angle between previous point, current point, and next point.
- Turn angle between previous point, current point, and centroids.

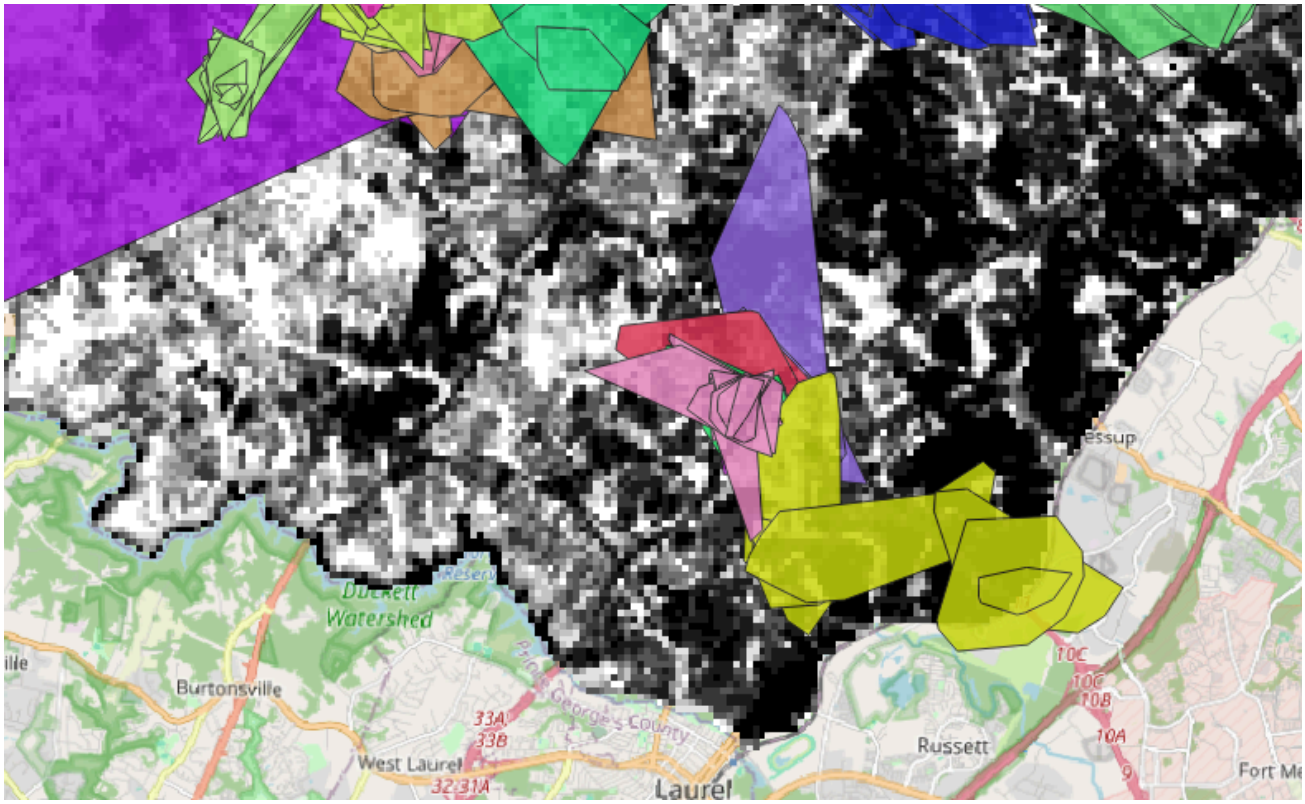


NOTE: For some reason the turn angles calculated here are **slightly** different to the turn angles calculated in the R package... Boils down to Great Circle arcs and was mentioned in git issue.

Other Data Fields

What about time of year, the canopy cover, some other values? Stick them here!

Note there are some Index Errors for the Suitability index raster, some of the GPS points are outside Howard County boundaries...



Final Output:

Below is the final processed GPS files that is used for the HMM fitting:

Write this all to pdf for sharing...

The R notebooks doesn't like this syntax, so creating the PDF here...

Let's look at some different model inputs...

There seems to be an issue with the way that the current HMM is using step and turns. It would be great to include the centroid in some way to reduce the simulated homerange size. We might be able to do that by using a turn model that mixes the turn-angle, return-home angle, and distance to centroid. First thing though, let's make a joy-plot to look at the distribution over distance to eyeball whether there is a change in turn angle vs distance to centroid...

The above plot looks very similar to all the other distance plots, a nice weibull distibution. Let's pick some obvious bins, like every 10 km, and then recalculate the turn angles per 10km bin from centroid. Hopefully we'll see that the turn distribution changes as we get further from home.