Using xtracks

Brian Wood

Installing xtracks

```
install.packages("devtools")
devtools::install_github("brianwood1/xtracks")
```

Load xtracks package

```
library(xtracks)
```

Load some spatiotemporal data

First you need to load some spatiotemporal data into a dataframe. For demonstration purposes, the xtracks package comes with two dataframes named `d1` and `d2` that are appropriately formatted. The structure of the data can be seen below.

```
head(d1)
#>         lat      lon elevation_m in_camp  unix_time distance_from_camp_m
#> 1 -3.575118 35.12340        1056       1 1520741140              54.8557
#> 2 -3.575125 35.12339        1056       1 1520741145              56.2129
#> 3 -3.575149 35.12335        1056       1 1520741150              59.6770
#> 4 -3.575153 35.12334        1056       1 1520741155              61.1063
#> 5 -3.575150 35.12336        1059       1 1520741160              58.9206
#> 6 -3.575143 35.12338        1063       1 1520741165              56.5872
```

As you can see, the data xtracks objects are based on includes the sort that come from GPS devices (lat, lon, elevation, time stamp, elevation) but also include a 'distance from camp' measure for all trackpoints and an 'in camp' flag. These later values are critical for analyses performed by xtracks and should be computed as an initial data processing step.

Construct an xtracks object

To construct an xtrack object, one must specify:

- lat
- lon
- elevation
- in-camp status
- time
- distance from camp centroid

- whether each trackpoint is “in camp” or not
- the utm_epsg code.

lat and lon are expected to be in decimal-degree, WGS 84 format, which is the default in most GPS devices mobile devices. elevation is expected to be in meters above sea level. The “in_camp” parameter refers to whether each trackpoint is within or outside the boundaries of a residential area, which in Wood et al. 2021 refers to the spatial boundaries of a Hadza camp; but could more generally be considered the boundaries of a residential or habitation area, something like a village or a camp, as appropriate in a given field setting. This is useful for indicating travel for the purpose of aquiring resources – AKA foraging travel, and needed for sinuosity measures. Distance from camp centroid is expected to be the as-the-crow-flies distance from the center of a residential area / ‘camp’ in meters (also needed for sinuosity measures). The epsg code identifies the UTM zone of your study location. This is needed for projecting lat / lon coordinates into UTM space. To find the epsg code for your study location region of your track, check out https://spatialreference.org/ref/epsg/

Assuming these initial data are in place, the code to construct an xtrack object is as follows:

```
xt_1 <- xtrack(lat=d1$lat, lon=d1$lon, elevation_m=d1$elevation_m, in_camp=d1$in_camp,
         unix_time=d1$unix_time, distance_from_camp_m=d1$distance_from_camp_m, utm_epsg=32736)
xt_2 <- xtrack(lat=d2$lat, lon=d2$lon, elevation_m=d2$elevation_m, in_camp=d2$in_camp,
         unix_time=d2$unix_time, distance_from_camp_m=d2$distance_from_camp_m, utm_epsg=32736)
```

View the xtrack

A nice dynamic map of the xtrack can be created using mapview. More visualization options will be highlighted at the end of the vignette.

```
xt_1$as_mapview()
```


+
−
100 m
300 ft
.self$as_sfc_linestring()
Leaflet | © OpenStreetMap contributors © CARTO