Using xtracks

Brian Wood

xtracks provides tools for the visualization and analysis of spatiotemporal data describing human movement patterns (real or simulated). This tool is built with social scientists and ecologists studying human movement using GPS data in mind. xtracks is in active development and new measures of individual and collective movement and social interaction will be added over time. This version (0.1.0) produces measures including daily distances traveled, sinuosity of travel paths, raster measures of landscape use, and rates of habitat exploration. This vignette explains these functions. A word of caution: xtracks imports several other R packages for geospatial analysis including `sp`, `sf`, `raster`, `mapview`, `geosphere`, and `maptools`. This is a hefty set of packages, and the first time you install this package you will no doubt be prompted to install or update several of these dependencies.

Installing xtracks

```
#run this next line if you don't have devtools installed
install.packages("devtools")
devtools::install_github("brianwood1/xtracks")
```

Load xtracks package

```
library(xtracks)
```

Load spatiotemporal data

Raw data taken from GPS devices takes many forms. I will leave the initial processing stage to you, blessed geospatial analyst. After you pluck, skin, and disinfect that raw data, you may import it into xtracks in the following form:

```
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 you feed into xtracks includes run-of-the-mill GPS data (lat, lon, elevation, time stamp, elevation) but also include a 'distance from camp' measure and 'in camp' binary variable. These later values are also your responsibility to construct, using criteria that are relevant to your study. They happen to be critical for functions xtracks uses. The meaning of these values will be explained more in a bit.

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. 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) referred 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 segmenting travel for the purposes of acquiring 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/

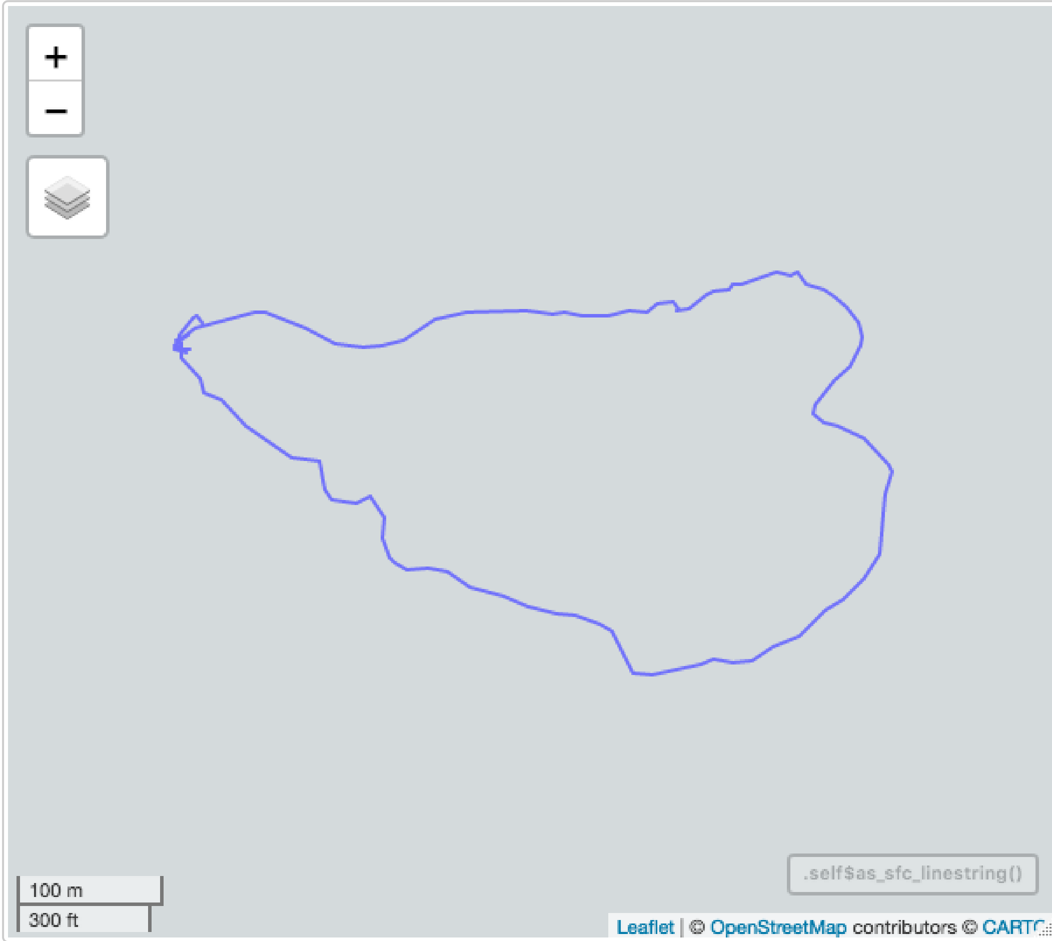
Assuming these initial data are in place, the code to construct an xtrack object is as follows:

```
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 interactive map of the xtrack can be created using mapview. More visualization options will be highlighted at the end of the vignette.

```
xt_2$as_mapview()
```



Analysis Functions

Get the total length of the xtrack in kilometers

```
xt_1$track_length_km
#> [1] 3.23
```

Get the total duration of the xtrack in hours

```r
xt_1$track_duration_hr
#> [1] 11.47083
```

Segmentation of travel into bouts of out of camp travel

Here I demonstrate segmentation of travel into *bouts* of out of camp travel. An out of camp bout is when an individual leaves camp, travels some distance, and then returns to camp.

```r
out_of_camp_bout_records_1 <- xt_1$get_out_of_camp_bout_records()
out_of_camp_bout_records_2 <- xt_2$get_out_of_camp_bout_records()
head(out_of_camp_bout_records_1)
#>   time_leaving_camp time_returning_to_camp total_time_of_bout_seconds
#> 1        1520748170             1520750345                       2175
#> 2        1520756215             1520756400                        185
#> 3        1520766480             1520766550                         70
#> 4        1520775495             1520775510                         15
#>   bout_identified trackpoint_id_max_dist_from_camp_during_bout
#> 1               1                                         1694
#> 2               1                                         3042
#> 3               1                                         5077
#> 4               1                                         6872
#>   max_dist_from_camp_during_bout_m mean_dist_from_camp_during_bout_m
#> 1                         438.7260                         278.92262
#> 2                          67.9154                          65.94240
#> 3                          95.3231                          82.81696
#> 4                          59.3988                          56.84257
#>   bout_number max_dist_during_bout_lon max_dist_during_bout_lat
#> 1           1                 35.12762                -3.576500
#> 2           2                 35.12346                -3.574771
#> 3           3                 35.12316                -3.574746
#> 4           4                 35.12354                -3.574803
#>   max_dist_unix_time total_length_of_bout_m
#> 1         1520749605            1230.985424
#> 2         1520756345              22.317448
#> 3         1520766520              54.935425
#> 4         1520775495               7.399626
```

Get the trackpoints of the longest duration bout

```
trackpoints_of_longest_duration_bout <- xt_1$get_longest_bout_trackpoints()
head(trackpoints_of_longest_duration_bout)
#>            lat      lon elevation_m in_camp  unix_time distance_from_camp_m
#> 1407 -3.575064 35.12408        1078       0 1520748170              26.2907
#> 1408 -3.575084 35.12412        1078       0 1520748175              28.9424
#> 1409 -3.575107 35.12416        1080       0 1520748180              32.2295
#> 1410 -3.575126 35.12420        1078       0 1520748185              35.5535
#> 1411 -3.575141 35.12423        1078       0 1520748190              38.9693
#> 1412 -3.575155 35.12426        1081       0 1520748195              42.1207
#>      pk_trackpoint_id seconds_since_prior_trackpoint
#> 1407             1407                              5
#> 1408             1408                              5
#> 1409             1409                              5
#> 1410             1410                              5
#> 1411             1411                              5
#> 1412             1412                              5
#>      meters_from_prior_trackpoint speed_m_s_from_prior_trackpoint    utm_x
#> 1407                     5.185051                       1.0370102 735953.5
#> 1408                     5.080294                       1.0160588 735958.1
#> 1409                     5.398277                       1.0796555 735962.9
#> 1410                     4.677329                       0.9354657 735967.0
#> 1411                     4.229085                       0.8458170 735970.9
#> 1412                     3.794097                       0.7588194 735974.3
#>        utm_y bout_number
#> 1407 9604570           1
#> 1408 9604568           1
#> 1409 9604565           1
#> 1410 9604563           1
#> 1411 9604561           1
#> 1412 9604560           1
```

Test if the xtrack has data sufficient to enable sinuosity calculations of the manner carried out in Wood et al. (2021)

```r
xt_1$has_data_for_sinuosity_measures()
#> [1] FALSE
xt_2$has_data_for_sinuosity_measures()
#> [1] TRUE
```

Get inbound and outbound sinuosity following the methods of Wood et al. (2021)

```r
xt_2$get_inbound_sinuosity()
#> [1] 1.237696
xt_2$get_outbound_sinuosity()
#> [1] 1.478008
```

More sinuosity-related measures

More sinuosity-relevant measures are available. These include:

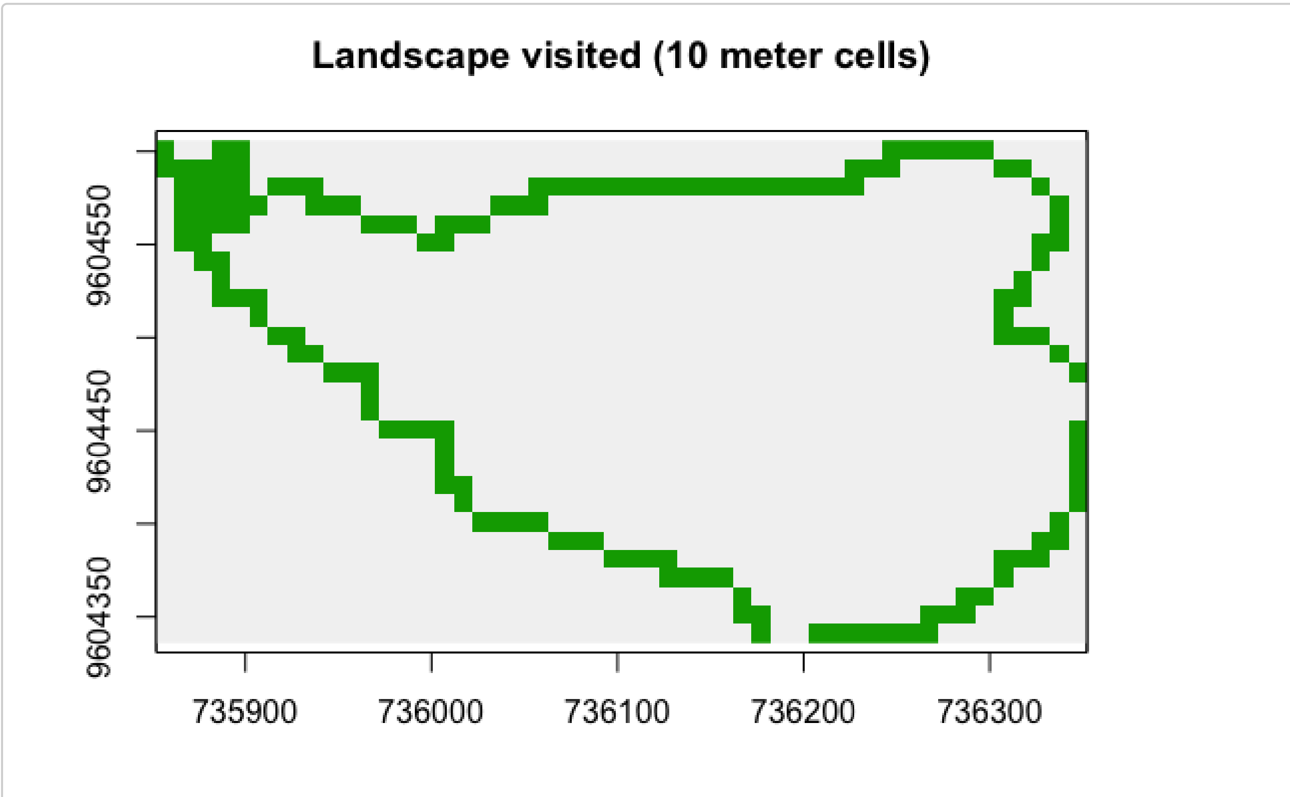
- The length (km) of the outbound and inbound segments as traveled
- The length (km) of the ‘as the crow flies’ distance from the point of leaving camp to the most distant point (sp_distance_outbound_km)
- The length of the ‘as the crow flies’ distance from the most distant point to the point of returning to camp (sp_distance_inbound_km)
- The mean sinuosity of the inbound and outbound segments
- The trackpoint_id of the most distant (from camp centroid) trackpoint

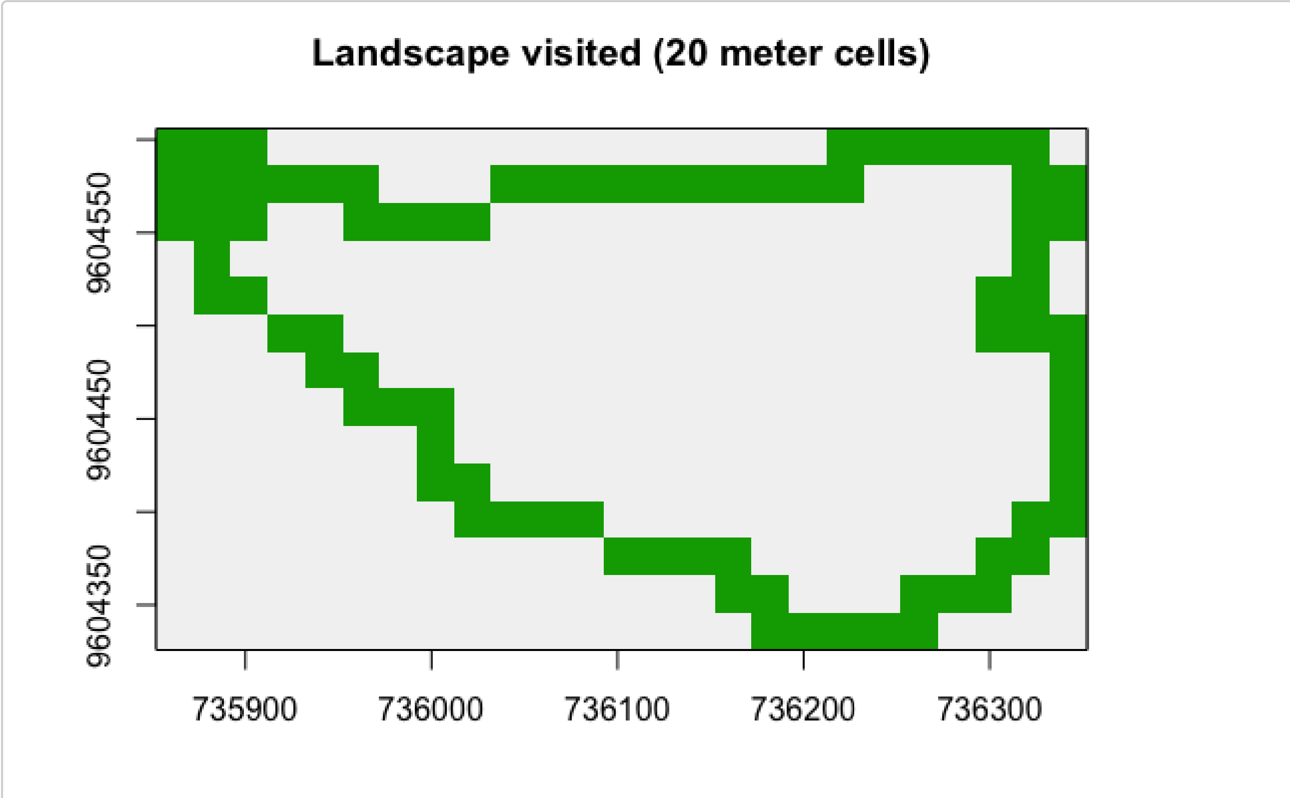
```r
xt_2$get_sinuosity_measures()
#>   length_outbound_section_km length_inbound_section_km sp_distance_outbound_km
#> 1                       1.82                      1.53                1.231387
#>   sp_distance_inbound_km outbound_sinuosity inbound_sinuosity mean_sinuosity
#> 1               1.236168           1.478008          1.237696       1.357852
#>   furthest_trackpoint_id
#> 1                   1586
```

Raster analysis identifying all places visited on the landscape

This produces a binary raster representation of the xtrack, where cells that are visited are given value 1, and those not visited given value 0. The length and width of the raster cells in meters is determined by the parameter `cell_size_m` and is by default 10. This function also accepts a parameter called `selected_trackpoints` which determines which of the trackpoints are rasterized. The acceptable values are ‘all’, ‘in_camp’, or ‘out_of_camp’. The default is ‘all’, as used in Wood et al. (2021).

```
bin_ras_1_10m <- xt_1$as_raster_of_habitat_visited_binary()
bin_ras_1_20m <- xt_1$as_raster_of_habitat_visited_binary(cell_size_m=20)
raster::plot(bin_ras_1_10m, main="Landscape visited (10 meter cells)", legend=FALSE, cex=0.5)
raster::plot(bin_ras_1_20m, main="Landscape visited (20 meter cells)", legend=FALSE, cex=0.5)
```





Raster analysis measuring visitation intensity / occupation counts of places on the landscape

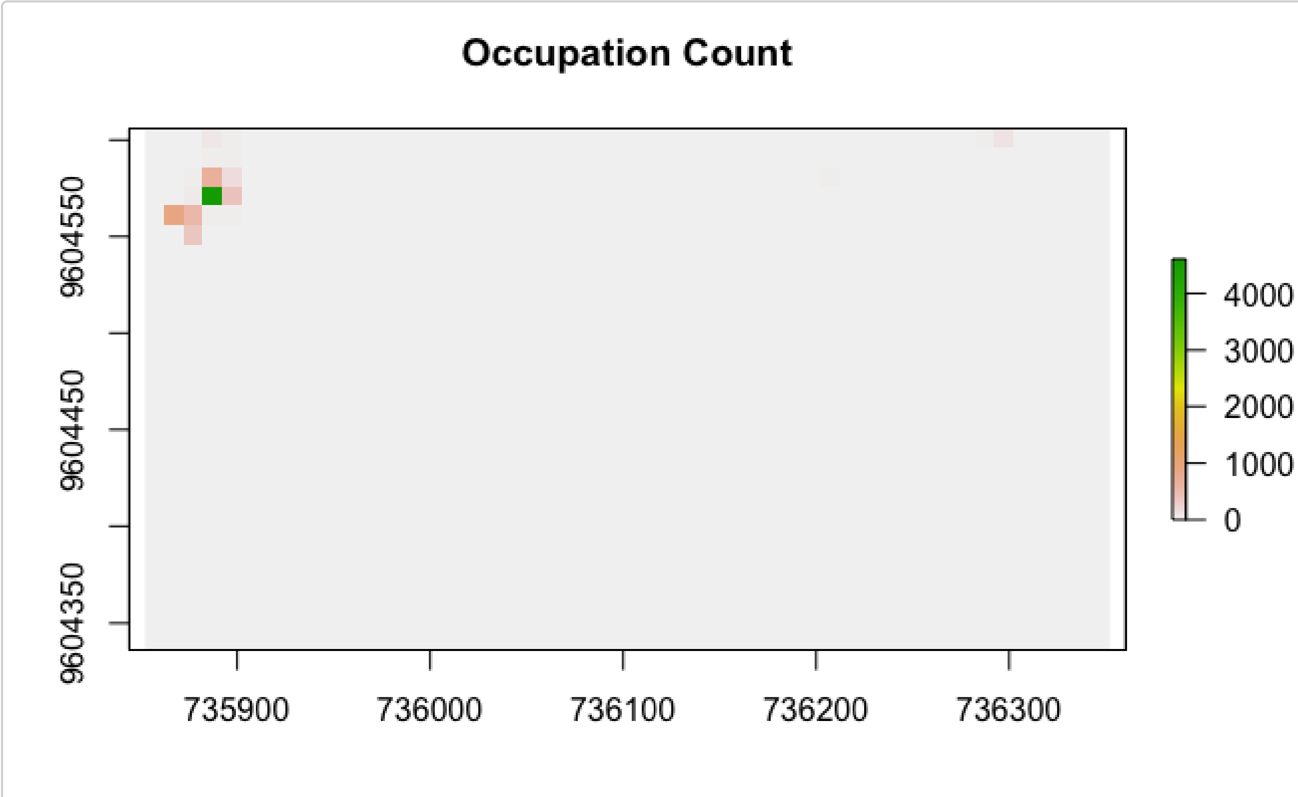
This produces a raster representation of the xtrack, where the count for each cell represents the number of trackpoints that fell within that cell's boundaries. Assuming that trackpoints are logged at regular time intervals, this measures the amount of time spent within each cell. Pretty cool! Un-visited cells are given value 0. As with the binary raster representation, The length and width of the raster cells in meters is determined by the parameter `cell_size_m` and is by default 10.

Laws of human nature / extremely skewed distributions of occupation counts

Law of human nature # 342: People are very choosy about where they go
Law of human nature # 716: People like to hang out in just a few places.

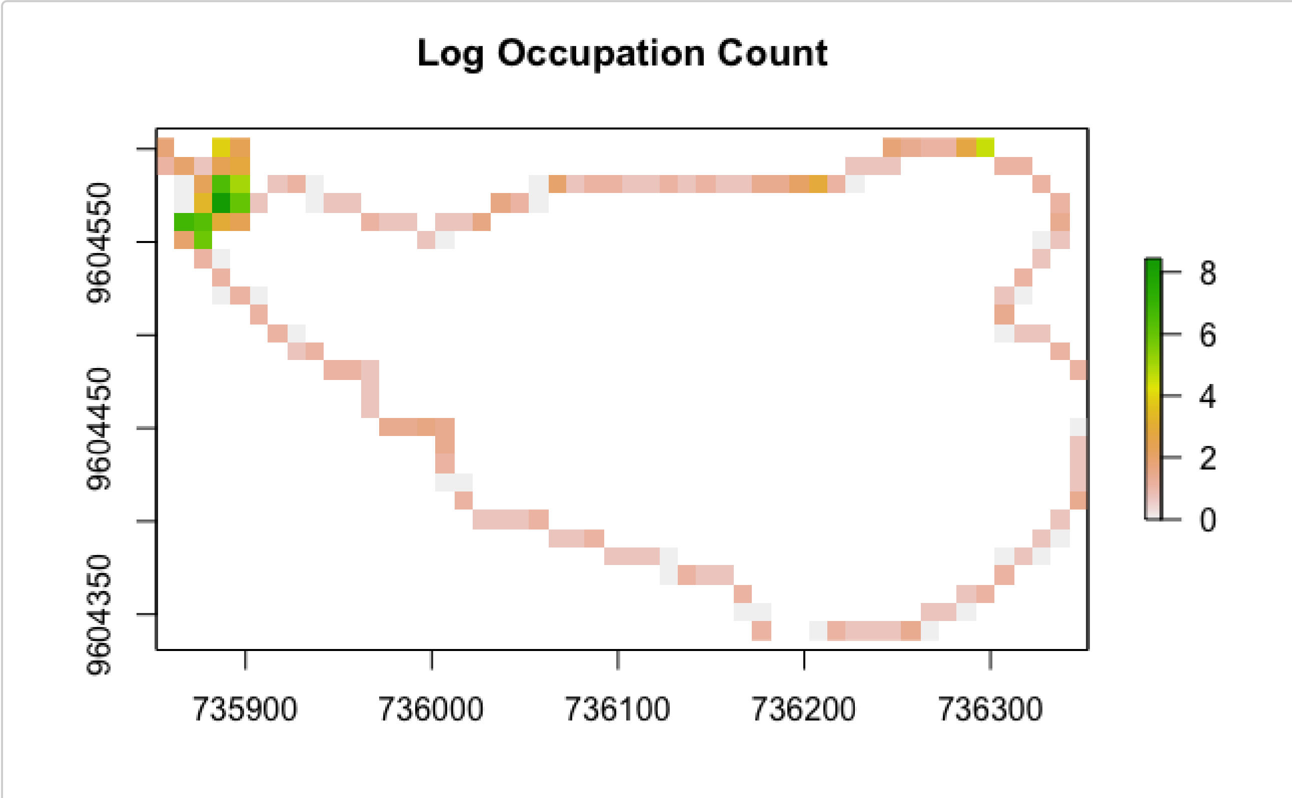
Because of these iron-clad laws, occupation counts tend to be very sparse across the landscape and occupation counts form *extremely* right-skewed distributions.

```
bin_ras_2 <- xt_1$as_raster_of_habitat_visited_counts()
raster::plot(bin_ras_2, main="Occupation Count")
```



Visualizing these distributions is challenging; a helpful approach can be to take the logarithm of visitation counts, which compresses the variability into a range that can be more comfortably viewed.

```
bin_ras_2 <- xt_1$as_raster_of_habitat_visited_counts()
log_scale_bin_ras_2 <- log(bin_ras_2)
raster::plot(log_scale_bin_ras_2, main="Log Occupation Count")
```



Analysis of rates of land exploration

Following Wood et al. (2021), this analysis computes the land visited / explored each day, the marginal ‘new’ land visited for each day, and the cumulative habitat explored across all days. In a research application modeled on our paper, this analysis should be done with a temporally-sorted list of xtracks, with each xtrack representing one day of travel of the same person. In the list, the first day of data should be in position 1. For demonstration purposes, I construct below some dummy data that does not actually represent 8 days of real travel, but instead, segments 1 day into 8 partially overlapping (temporally and spatially) segments. Imagine however that ‘xt_3’ through ‘xt_10’ represent 8 days of travel; the code works the same.

Construct the example data

```
d_temp <- d1
d3 <- d_temp[1:1000,]
d4 <- d_temp[500:2000,]
```

```
d5 <- d_temp[1500:3000,]
d6 <- d_temp[2500:4000,]
d7 <- d_temp[3500:5000,]
d8 <- d_temp[4500:6000,]
d9 <- d_temp[5500:7000,]
d10 <- d_temp[6500:8260,]

xt_3 <- xtrack(lat=d3$lat, lon=d3$lon, elevation_m=d3$elevation_m, in_camp=d3$in_camp,
         unix_time=d3$unix_time, distance_from_camp_m=d3$distance_from_camp_m, utm_epsg=32736)
xt_4 <- xtrack(lat=d4$lat, lon=d4$lon, elevation_m=d4$elevation_m, in_camp=d4$in_camp,
         unix_time=d4$unix_time, distance_from_camp_m=d4$distance_from_camp_m, utm_epsg=32736)
xt_5 <- xtrack(lat=d5$lat, lon=d5$lon, elevation_m=d5$elevation_m, in_camp=d5$in_camp,
         unix_time=d5$unix_time, distance_from_camp_m=d5$distance_from_camp_m, utm_epsg=32736)
xt_6 <- xtrack(lat=d6$lat, lon=d6$lon, elevation_m=d6$elevation_m, in_camp=d6$in_camp,
         unix_time=d6$unix_time, distance_from_camp_m=d6$distance_from_camp_m, utm_epsg=32736)
xt_7 <- xtrack(lat=d7$lat, lon=d7$lon, elevation_m=d7$elevation_m, in_camp=d7$in_camp,
         unix_time=d7$unix_time, distance_from_camp_m=d7$distance_from_camp_m, utm_epsg=32736)
xt_8 <- xtrack(lat=d8$lat, lon=d8$lon, elevation_m=d8$elevation_m, in_camp=d8$in_camp,
         unix_time=d8$unix_time, distance_from_camp_m=d8$distance_from_camp_m, utm_epsg=32736)
xt_9 <- xtrack(lat=d9$lat, lon=d9$lon, elevation_m=d9$elevation_m, in_camp=d9$in_camp,
         unix_time=d9$unix_time, distance_from_camp_m=d9$distance_from_camp_m, utm_epsg=32736)
xt_10 <- xtrack(lat=d10$lat, lon=d10$lon, elevation_m=d10$elevation_m, in_camp=d10$in_camp,
         unix_time=d10$unix_time, distance_from_camp_m=d10$distance_from_camp_m, utm_epsg=32736)
```

Create a list of xtracks

```
list_of_xtracks <- list(xt_3, xt_4, xt_5, xt_6, xt_7, xt_8, xt_9, xt_10)
```

Execute the land exploration analysis

```
hab_exp_results <- get_hab_exp_across_days(list_of_xtracks, cell_size_m = 10)
```

Results of this analysis are handed back in a hopefully easy to understand data frame format.

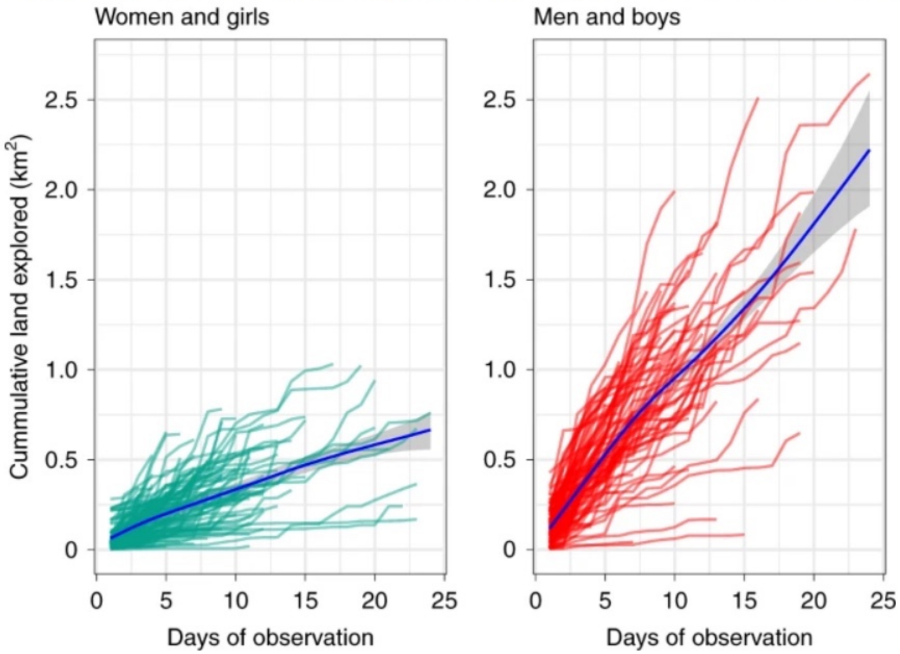
```
hab_exp_results
#>   day sum_cells_visited_this_day cum_sum_cells_visited_across_days
#> 1   1                          5                                 5
#> 2   2                        145                               146
#> 3   3                        111                               146
#> 4   4                         11                               151
```

```
#> 5   5                        10                              153
#> 6   6                        19                              159
#> 7   7                         8                              159
#> 8   8                        12                              159
#>   n_new_cells_visited_this_day square_meters_per_cell
#> 1                           NA                    100
#> 2                          141                    100
#> 3                            0                    100
#> 4                            5                    100
#> 5                            2                    100
#> 6                            6                    100
#> 7                            0                    100
#> 8                            0                    100
#>   sum_square_meters_visited_this_day cum_sum_square_meters_visited_across_days
#> 1                                500                                       500
#> 2                              14500                                     14600
#> 3                              11100                                     14600
#> 4                               1100                                     15100
#> 5                               1000                                     15300
#> 6                               1900                                     15900
#> 7                                800                                     15900
#> 8                               1200                                     15900
#>   square_meters_new_habitat_visited_this_day
#> 1                                         NA
#> 2                                      14100
#> 3                                          0
#> 4                                        500
#> 5                                        200
#> 6                                        600
#> 7                                          0
#> 8                                          0
```

Plot the results of land exploration analysis

Here is figure 3 in Wood et al. 2021

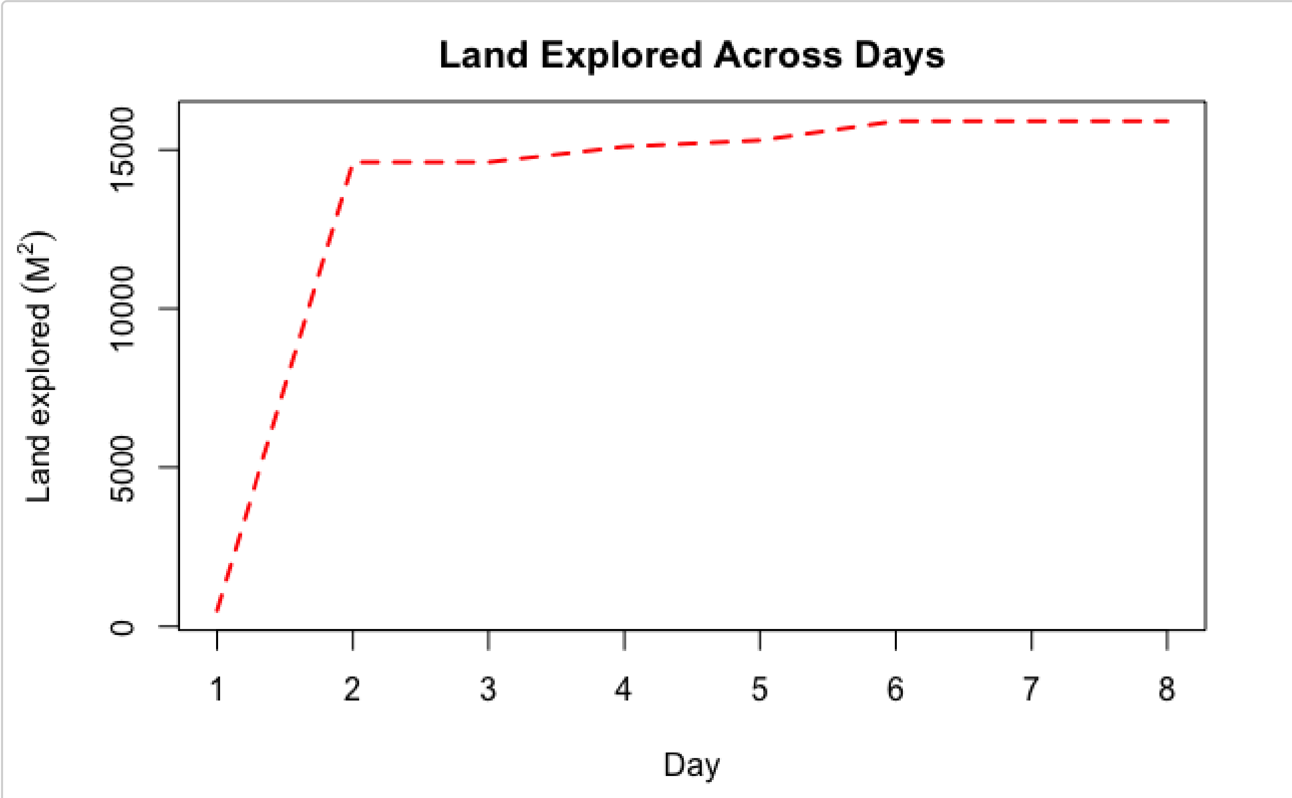
Fig. 3: Cumulative land explored by Hadza individuals across days of observation.



Each line represents one person in one camp. The blue trend line is the estimated mean, and the shaded area is its 95% CI from fit model 4. The data measure the land explored by 92 men and boys and 87 women and girls, representing 1,097 days of female travel and 981 days of male travel (Table 1). See "Land exploration" in Methods for more details of this analysis.

The plot below is modeled on that figure, though not as fancy. What is plotted is just one individual's travel across simulated days. The y-axis here is also in units of square meters, not square kilometers. The goal here is demonstrate the computing tools; turning this bare-bones plot into something nicer will be your design challenge :).

```
y_ax <- expression(Land~ explored ~ (M^"2"))
par(mar=c(4.5,4.5,2.5,2.5))
plot(x=hab_exp_results$day, y=hab_exp_results$cum_sum_square_meters_visited_across_days, type="l",
       xlab="Day", ylab=y_ax, col="red", main="Land Explored Across Days", cex=0.5, lwd=2, lty=2)
```



Export from xtracks to other formats

dataframe

It is easy to get all the trackpoints from an xtrack into dataframe format. Note that this representation has more more information and annotations that the 'raw' data used to construct an xtrack. This includes columns for the time between each trackpoint, meters traveled between trackpoints, speed of travel between trackpoints, and the utm coordinates of each trackpoint.

```
trackpoints_1 <- xt_1$get_trackpoints()
head(trackpoints_1)
#>         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
#>   pk_trackpoint_id seconds_since_prior_trackpoint meters_from_prior_trackpoint
#> 1                1                              0                     0.000000
#> 2                2                              5                     1.687472
#> 3                3                              5                     4.675570
#> 4                4                              5                     1.592730
#> 5                5                              5                     2.260842
#> 6                6                              5                     2.599409
#>   speed_m_s_from_prior_trackpoint    utm_x   utm_y bout_number
#> 1                       0.0000000 735878.7 9604564     in camp
#> 2                       0.3374944 735877.2 9604563     in camp
#> 3                       0.9351140 735873.3 9604561     in camp
#> 4                       0.3185461 735871.8 9604560     in camp
#> 5                       0.4521683 735874.0 9604561     in camp
#> 6                       0.5198819 735876.5 9604562     in camp
```

KML

KML files are used in Google Earth and elsewhere.

```
xt_1$write_kml_file(kml_file_name = "xt_1_c.kml")
```

GPX

```
xt_1$write_gpx_file(gpx_file_name="xt_1.gpx", gpx_track_name="XT1")
```

SpatialLinesDataFrame

This is an object type in the `sp` package, an important R package for spatial analysis

```
xt_1_sldf <- xt_1$as_spatial_lines_dataframe()
class(xt_1_sldf)
```

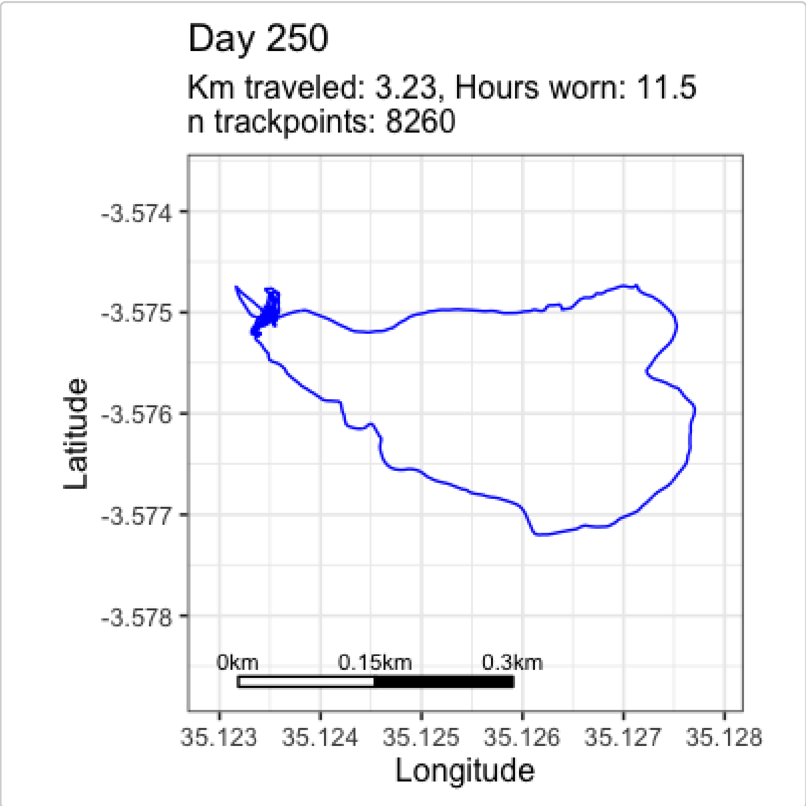
```
#> [1] "SpatialLinesDataFrame"
#> attr(,"package")
#> [1] "sp"
```

Plotting options

Nice map

A call to `plot_nice_map_of_track` creates a 'nice map' that harnesses `ggplot2` functions. It is a clean plot of the xtrack's travel path with a simple `ggplot2` black and white theme, a customized scale bar, and some metadata displayed in the subtitle area. This function accepts parameters for a title (the_title), and the color of the line representing the xtrack (line_color).

```
xt_1$plot_nice_map_of_track(the_title="Day 250", line_color="blue")
```

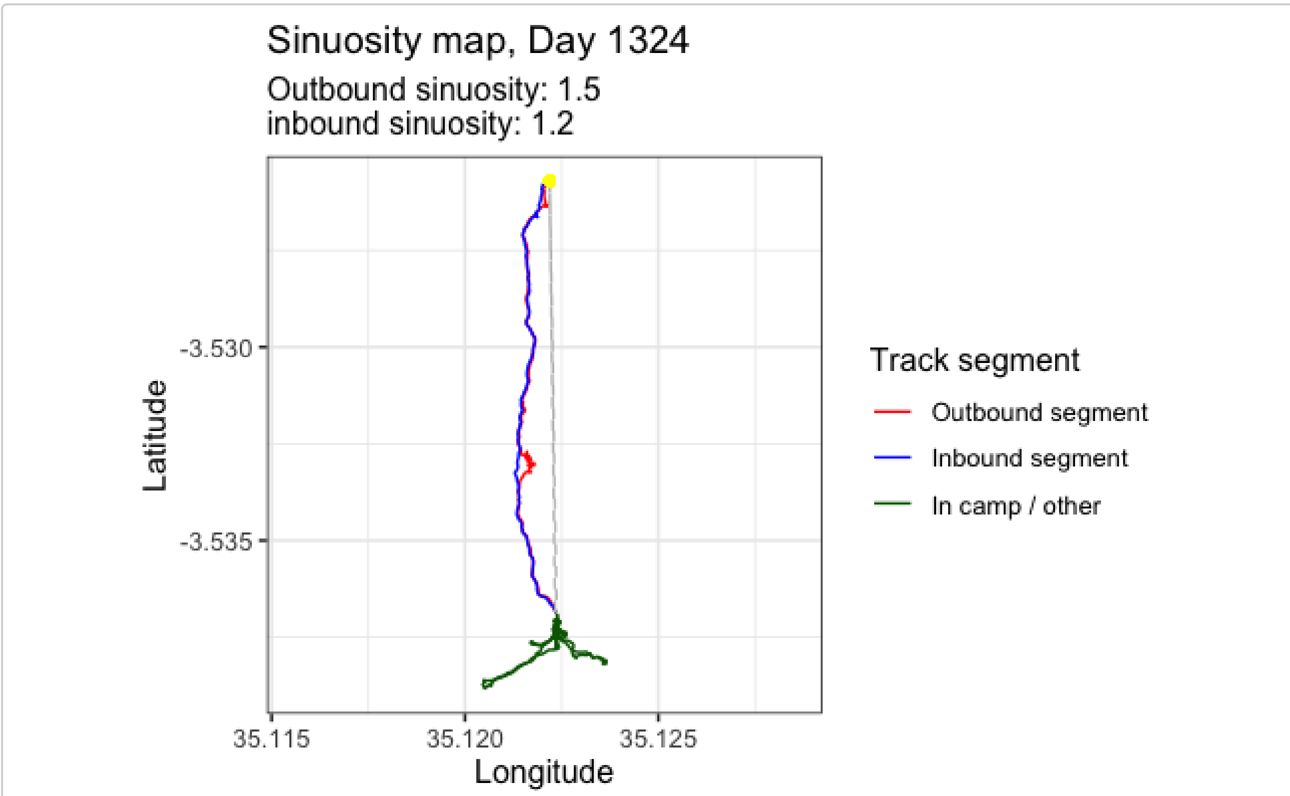


Sinuosity map

A call to `plot_sinuosity_map` creates a visual representation of the outbound travel segment (red), the inbound segment (blue), travel in camp or during shorter out of camp segments (green), the 'as the crow flies' shortest path segments used to calculate sinuosity measures (grey dashed line). The trackpoint that is maximally distant from the camp centroid is plotted in yellow. The sinuosity measures themselves are plotted in the subtitle and the plot accepts a parameter for the map title.

```
xt_2$plot_sinuosity_map(the_title="Sinuosity map, Day 1324")
```

```
knitr::include_graphics("sin_map.png")
```



Mapview map

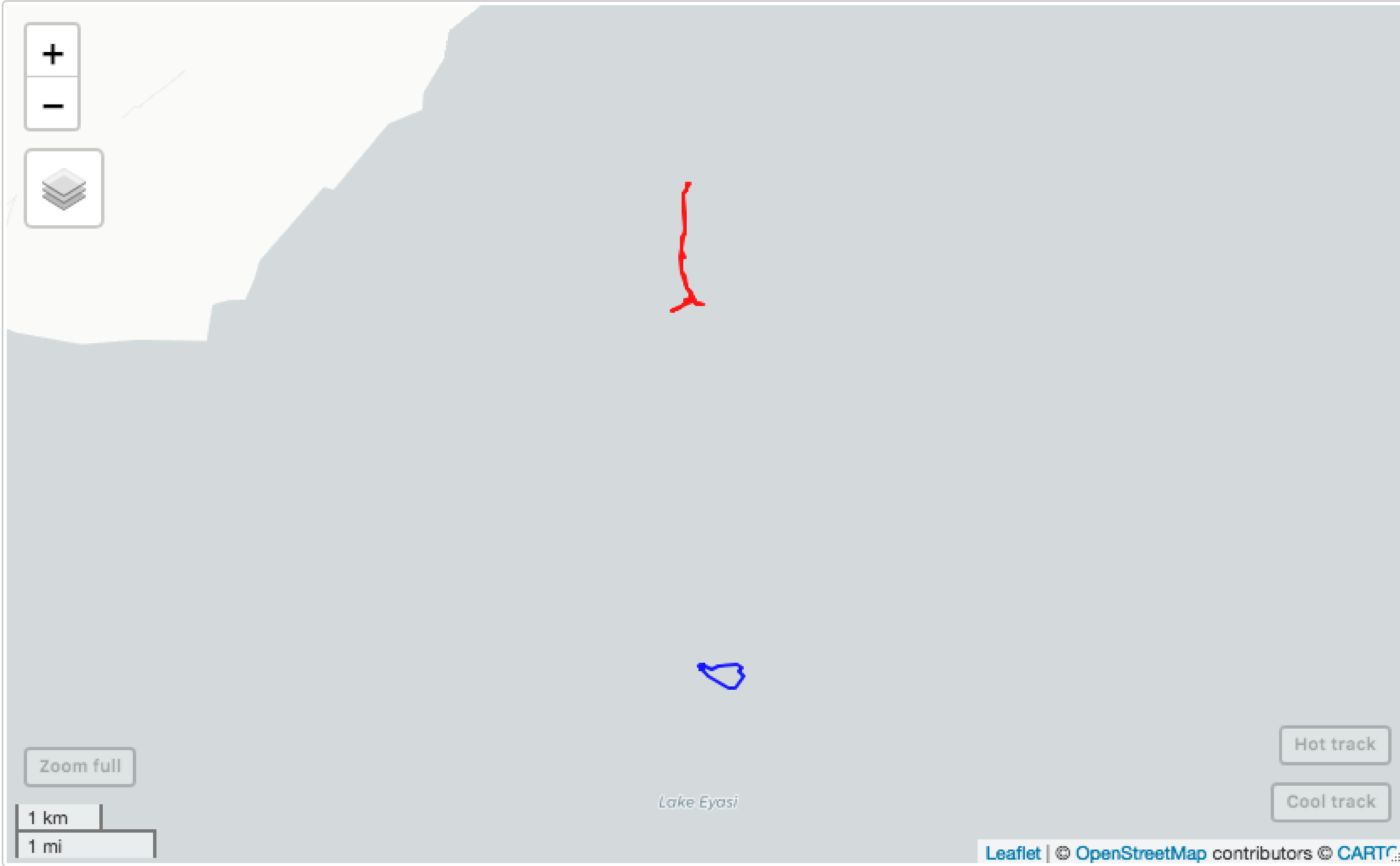
A call to `as_mapview()` is a thin interface to the package mapview, producing an interactive plot that can be viewed in the plot window or view in shiny applications. There is a known issue that makes it impossible currently to render a mapview map in an R markdown document, user beware. What are shown here are what you would see appearing in a plot window if you ran this code in an R-script; not what you would see if you ran this code in an R markdown document.

This function accepts all parameters that can be fed to the function mapview in the package mapview, such as layer.name, color, etc.

```
xt_1$as_mapview(format="line", layer.name="Hot track", color="red")
```

Using the power of mapview, we can also display multiple xtracks on a dynamic map background as follows:

```
a <- xt_1$as_mapview(format="line", layer.name="Cool track", color="blue")
b <- xt_2$as_mapview(format="line", layer.name="Hot track", color="red")
a+b
```



Other plotting options you can chose from might harness the power of ‘plot’ functions defined in other packages, such as `sp` or `raster`; since xtracks exports to those formats.

References

Appelhans, T., Detsch, F., Reudenbach, C., and Woellauer, S. (2020). mapview: Interactive Viewing of Spatial Data in R. R package version 2.9.0. https://CRAN.R-project.org/package=mapview

Bivand, R., and Lewin-Koh, N. (2021). maptools: Tools for Handling Spatial Objects. R package version 1.1-1. https://CRAN.R-project.org/package=maptools

Hijmans, R. (2020). raster: Geographic Data Analysis and Modeling. R package version 3.4-5. https://CRAN.R-project.org/package=raster

Hijmans, R. (2019). geosphere: Spherical Trigonometry. R package version 1.5-10. https://CRAN.R-project.org/package=geosphere

Pebesma, E.J., and Bivand, R. 2005. Classes and methods for spatial data in R. R News 5 (2), https://cran.r-project.org/doc/Rnews/.

Pebesma, E., 2018. Simple Features for R: Standardized Support for Spatial Vector Data. The R Journal 10 (1), 439-446, https://doi.org/10.32614/RJ-2018-009

Wood, B. M., Harris, J. A., Raichlen, D. A., Pontzer, H., Sayre, K., Sancilio, A., … & Jones, J. H. (2021). Gendered movement ecology and landscape use in Hadza hunter-gatherers. Nature Human Behaviour, 5, 436–446.