I'll illustrate how to use the package based on the following wrapped code snippet, which creates the leaflet and tmap maps and does the following:

- it loads the required files (**Alberta_Wolves.csv** and **Mountain_caribou.csv**)
- it iterates over the files
- inside the for-loop for each file (and animal type) separately
    - it keeps the required columns ('longitude', 'latitude', 'timestamp', 'individual_local_identifier', 'individual-taxon-canonical-name')
    - it builds a simple features object of the input data.tables
    - it creates a bounding box of the coordinate points
    - it extends the boundaries of the bounding box by 250 meters (so that points close to the boundaries are visible too)
    - it downloads and saves to a temporary directory the 30 meter elevation data for the Area of Interest (either for the 'Wolves' or for the 'Tarandus')
    - it creates a Virtual Raster (.VRT) mosaic file of the multiple downloaded Elevation .tif files
    - it crops the Digital Elevation Model (DEM) using the previously created bounding box (the downloaded DEM's cover a bigger area, because they consist of fixed grid tiles)
    - it saves the **tmap** of each processed input file and the **data.tables** which are required for the leaflet map

```
files = c(system.file('vignette_data/Alberta_Wolves.csv', package =
"CopernicusDEM"),
          system.file('vignette_data/Mountain_caribou.csv', package =
"CopernicusDEM"))


leafgl_data = tmap_data = list()


for (FILE in files) {

  cat(glue::glue("Processing of the '{basename(FILE)}' file ..."),
'\n')

  dtbl = data.table::fread(FILE, header = TRUE, stringsAsFactors =
FALSE)
  cols = c('location-long', 'location-lat', 'timestamp', 'individual-
local-identifier',
           'individual-taxon-canonical-name')

  dtbl_subs = dtbl[, ..cols]
  colnames(dtbl_subs) = c('longitude', 'latitude', 'timestamp',
'individual_local_identifier',
                          'individual-taxon-canonical-name')

  leafgl_data[[unique(dtbl_subs$`individual-taxon-canonical-name`)]] =
dtbl_subs
```

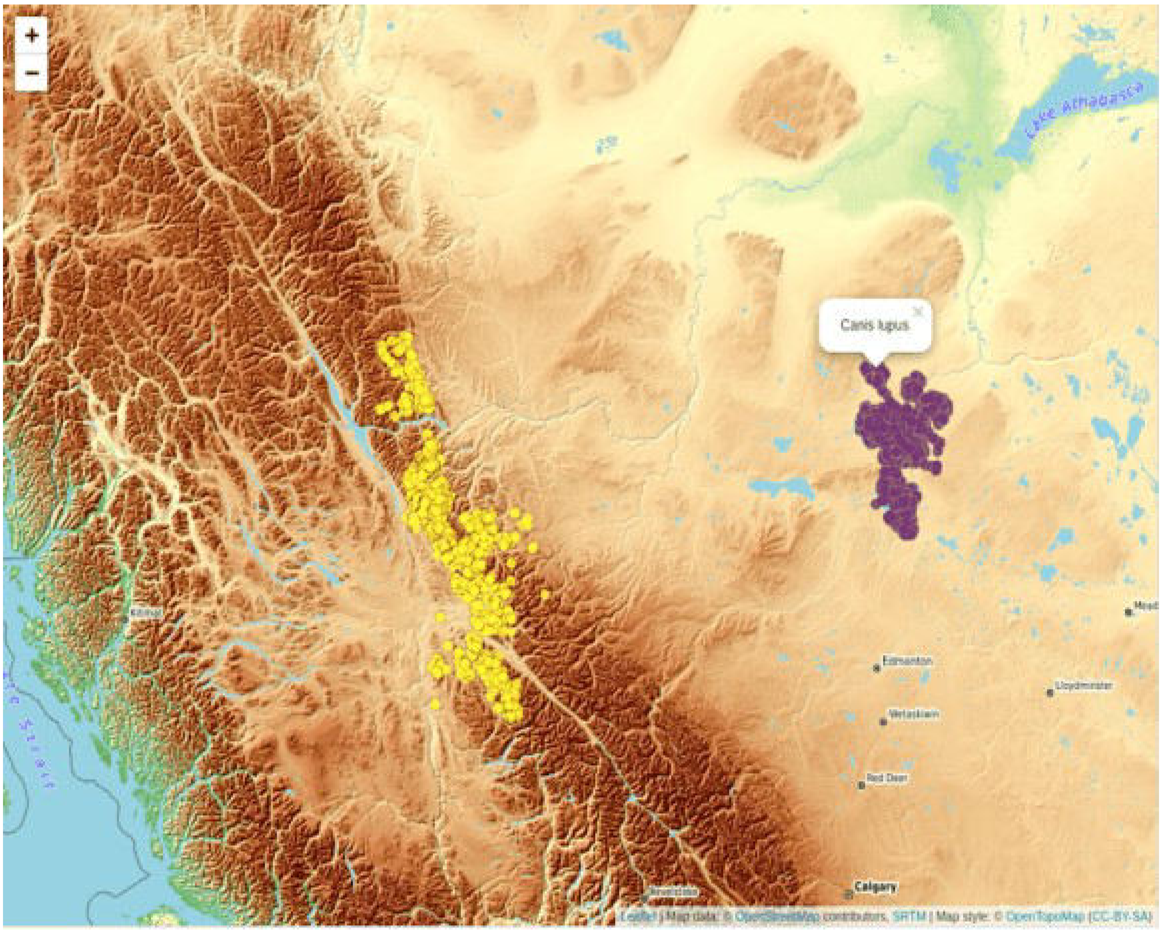
```
  dtbl_subs_sf = sf::st_as_sf(dtbl_subs, coords = c("longitude",
"latitude"), crs = 4326)

  sf_rst_ext = fitbitViz::extend_AOI_buffer(dat_gps_tcx = dtbl_subs_sf,
                                            buffer_in_meters = 250,
                                            CRS = 4326,
                                            verbose = TRUE)

  #................................................................
  # Download the Copernicus DEM 30m elevation data because it has
  # a better resolution, it takes a bit longer to download because
  # the .tif file size is bigger
  #...............................................................

  dem_dir = tempdir()

  dem30 = CopernicusDEM::aoi_geom_save_tif_matches(sf_or_file =
sf_rst_ext$sfc_obj,
                                                   dir_save_tifs =
dem_dir,
                                                   resolution = 30,
                                                   crs_value = 4326,
                                                   threads =
parallel::detectCores(),
                                                   verbose = TRUE)

  TIF = list.files(dem_dir, pattern = '.tif', full.names = TRUE)

  if (length(TIF) > 1) {

    #....................................................
    # create a .VRT file if I have more than 1 .tif files
    #....................................................

    file_out = file.path(dem_dir, 'VRT_mosaic_FILE.vrt')

    vrt_dem30 = CopernicusDEM::create_VRT_from_dir(dir_tifs = dem_dir,
                                                   output_path_VRT =
file_out,
                                                   verbose = TRUE)
  }

  if (length(TIF) == 1) {

    #..................................................
    # if I have a single .tif file keep the first index
    #..................................................

    file_out = TIF[1]
  }
```

```
  raysh_rst = fitbitViz::crop_DEM(tif_or_vrt_dem_file = file_out,
                                   sf_buffer_obj = sf_rst_ext$sfc_obj,
                                   CRS = 4326,
                                   digits = 6,
                                   verbose = TRUE)

  # convert to character to receive the correct labels in the 'tmap'
object
  dtbl_subs_sf$individual_local_identifier = as.character(dtbl_subs_sf$
individual_local_identifier)

  # open with interactive viewer
  tmap::tmap_mode("view")

  map_coords = tmap::tm_shape(shp = dtbl_subs_sf) +
    tmap::tm_dots(col = 'individual_local_identifier')

  map_coords = map_coords + tmap::tm_shape(shp = raysh_rst, is.master =
FALSE, name = 'Elevation') +
    tmap::tm_raster(alpha = 0.65, legend.reverse = TRUE)

  tmap_data[[unique(dtbl_subs$`individual-taxon-canonical-name`)]] =
map_coords
}
```

```
Processing of the 'Alberta_Wolves.csv' file ...
Convert the data.table to an 'sf' object ...
Transform the projection of the 'sf' object from  4326  to 7801 ...
Create a buffer of  250  meters using as input the initial sf object ...
Back-tranformation of the projection and computation of the bounding box ...
Use the bounding box to extract the raster extent ...
Compute the centroid of the sf-buffer object ...
Elapsed time: 0 hours and 0 minutes and 0 seconds.
Parallel download of the  6  .tif files using  8  threads starts ...
Elapsed time: 0 hours and 2 minutes and 45 seconds.
The VRT Mosaic will be built from  6   .tif files and will be saved in  '/tmp/RtmpSdQytm/VRT_mosaic_FILE.vrt' ...
Checking gdal_installation...
Scanning for GDAL installations...
Checking Sys.which...
GDAL version 2.2.2
GDAL command being used: "/usr/bin/gdalbuildvrt" "/tmp/RtmpSdQytm/VRT_mosaic_FILE.vrt" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N54_00_W113_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10
_N54_00_W114_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N55_00_W113_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N55_00_W114_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_1
0_N56_00_W113_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N56_00_W114_00_DEM.tif"
The raster will be red ...
The AOI will be extracted from the raster DEM ...
A data.table will be created from the x,y,z vectors ...
A raster will be created from the x,y,z data.table ...
tmap mode set to interactive viewing
Processing of the 'Mountain_caribou.csv' file ...
Convert the data.table to an 'sf' object ...
Transform the projection of the 'sf' object from  4326  to 7801 ...
Create a buffer of  250  meters using as input the initial sf object ...
Back-tranformation of the projection and computation of the bounding box ...
Use the bounding box to extract the raster extent ...
Compute the centroid of the sf-buffer object ...
Elapsed time: 0 hours and 0 minutes and 0 seconds.
Parallel download of the  16  .tif files using  8  threads starts ...
Elapsed time: 0 hours and 7 minutes and 43 seconds.
The VRT Mosaic will be built from  22   .tif files and will be saved in  '/tmp/RtmpSdQytm/VRT_mosaic_FILE.vrt' ...
Checking gdal_installation...
GDAL version 2.2.2
GDAL command being used: "/usr/bin/gdalbuildvrt" "/tmp/RtmpSdQytm/VRT_mosaic_FILE.vrt" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N53_00_W121_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10
_N53_00_W122_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N53_00_W123_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N53_00_W124_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_1
0_N54_00_W113_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N54_00_W114_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N54_00_W121_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_
10_N54_00_W122_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N54_00_W123_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N54_00_W124_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG
_10_N55_00_W113_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N55_00_W114_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N55_00_W121_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_CO
G_10_N55_00_W122_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N55_00_W123_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N55_00_W124_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_C
OG_10_N56_00_W113_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N56_00_W114_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N56_00_W121_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_
COG_10_N56_00_W122_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N56_00_W123_00_DEM.tif" "/tmp/RtmpSdQytm/Copernicus_DSM_COG_10_N56_00_W124_00_DEM.tif"
The raster will be red ...
The AOI will be extracted from the raster DEM ...
A data.table will be created from the x,y,z vectors ...
A raster will be created from the x,y,z data.table ...
tmap mode set to interactive viewing
```

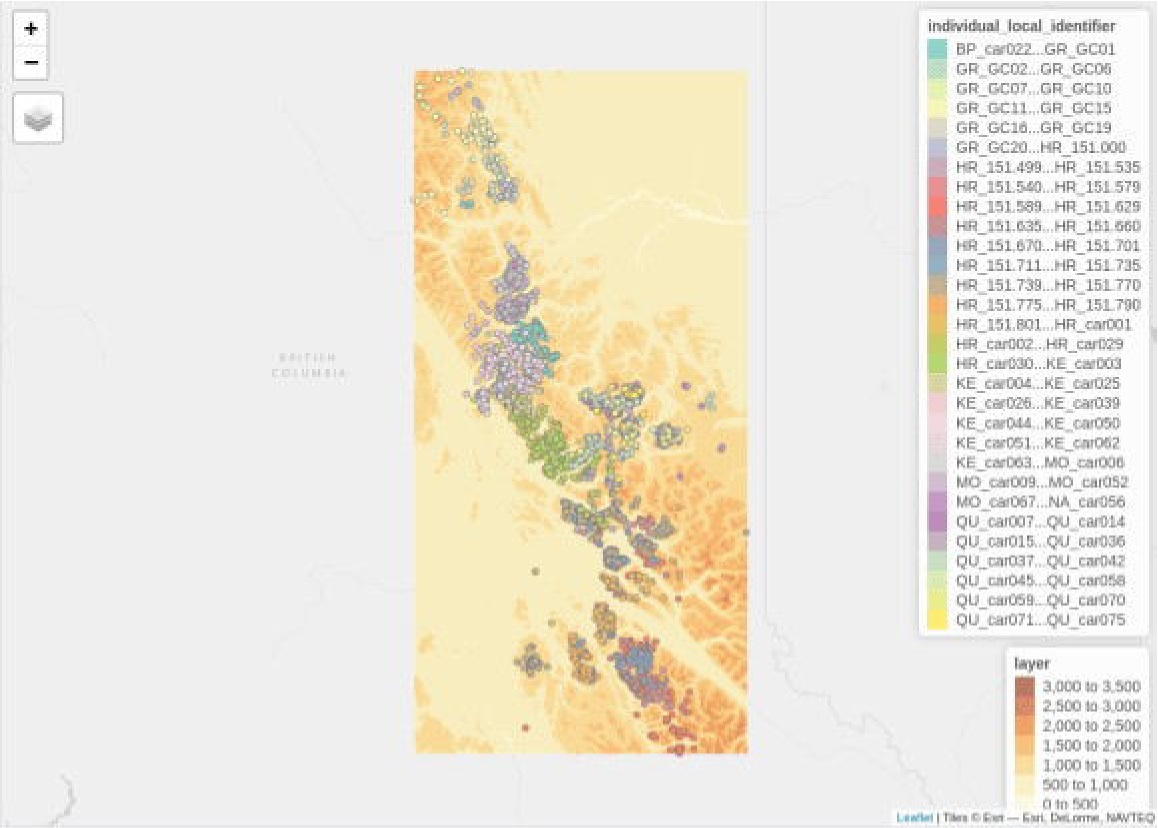
Now, based on the saved data.tables we can create first the **leaflet map** to view the data of both animal species in the same map,

```
#....................................
# create the 'leafGl' of both datasets
#....................................

dtbl_all = rbind(leafgl_data$`Canis lupus`, leafgl_data$`Rangifer
tarandus`)

# see the number of observations for each animal
table(dtbl_all$`individual-taxon-canonical-name`)

# create an 'sf' object of both data.tables
dat_gps_tcx = sf::st_as_sf(dtbl_all, coords = c("longitude",
"latitude"), crs = 4326)

lft = leaflet::leaflet()
lft = leaflet::addProviderTiles(map = lft, provider =
leaflet::providers$OpenTopoMap)

lft = leafgl::addGlPoints(map = lft,
                          data = dat_gps_tcx,
                          opacity = 1.0,
                          fillColor = 'individual-taxon-canonical-
name',
                          popup = 'individual-taxon-canonical-name')
lft
```



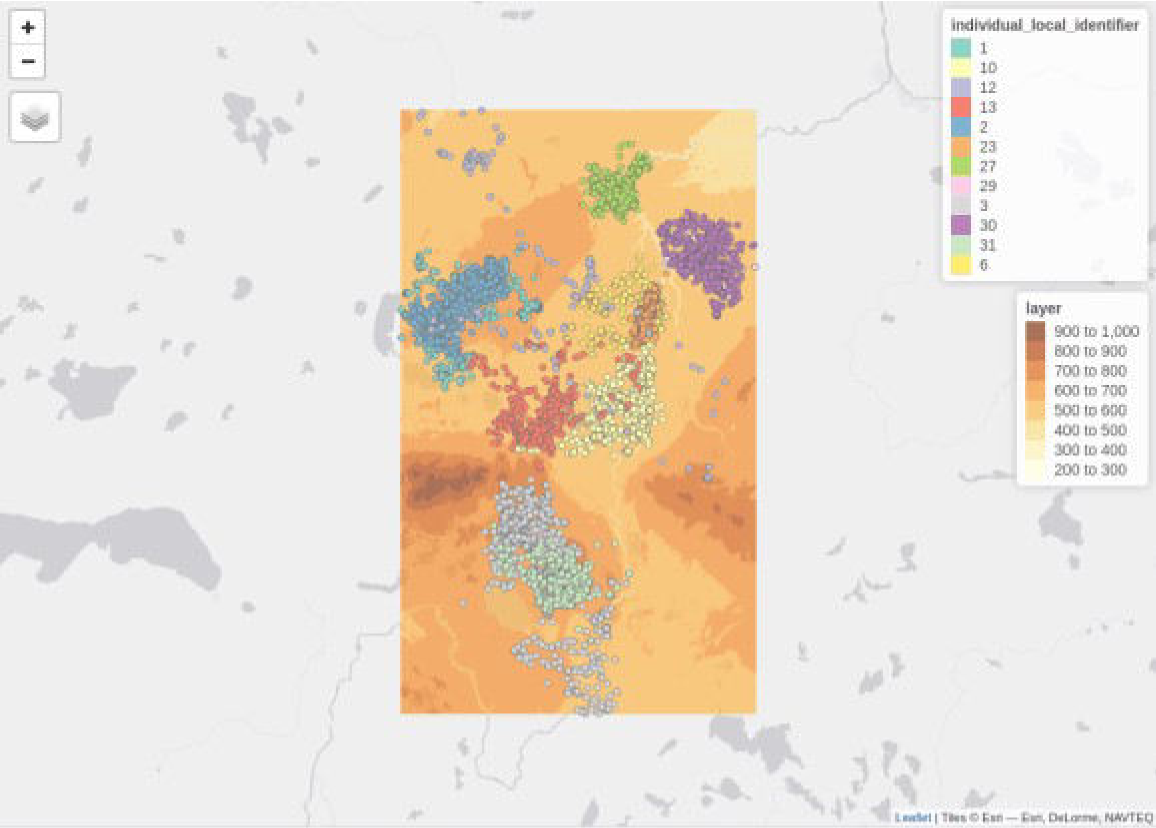
The tracking data of the *Caribou* are on a higher elevation compared to the data of the *Wolves*. This is verified by the next *tmap* which includes the Elevation legend (**layer**). The additional legend shows the **individual identifier** of the animal – for the **Tarandus** there are **138** unique id's whereas

```
tmap_data$`Rangifer tarandus`                                   # caribou
```




```
tmap_data$`Canis lupus`                                  # wolves
```

for the **Wolves** only **12**,



Elevation data using the **CopernicusDEM** R package can be visualized also in **3-dimensional space**. For the corresponding use case have a look to the Vignette of the fitbitViz R package which uses internally the Rayshader package (especially the **last** image of the Vignette).

Movebank References:

- Latham Alberta Wolves
    - Latham ADM (2009) Wolf ecology and caribou-primary prey-wolf spatial relationships in low productivity peatland complexes in northeastern Alberta. Dissertation. ProQuest Dissertations Publishing, University of Alberta, Canada, NR55419, 197 p. url:https://www.proquest.com/docview/305051214
    - Latham ADM and Boutin S (2019) Data from: Wolf ecology and caribou-primary prey-wolf spatial relationships in low productivity peatland complexes in northeastern Alberta. Movebank Data Repository. \doi:10.5441/001/1.7vr1k987
- Mountain caribou in British Columbia-radio-transmitter
    - BC Ministry of Environment (2014) Science update for the South Peace Northern Caribou (Rangifer tarandus caribou pop. 15) in British Columbia. Victoria, BC. 43 p. https://www2.gov.bc.ca/assets/gov/environment/plants-animals-and-ecosystems/wildlife-wildlife-habitat/caribou/science_update_final_from_web_jan_2014.pdf url:https://www2.gov.bc.ca/assets/gov/environment/plants-animals-and-ecosystems/wildlife-wildlife-habitat/caribou/science_update_final_from_web_jan_2014.pdf
    - Seip DR and Price E (2019) Data from: Science update for the South Peace Northern Caribou (Rangifer tarandus caribou pop. 15) in British Columbia. Movebank Data Repository. \doi:10.5441/001/1.p5bn656k

Installation & Citation:

An updated version of the **CopernicusDEM** package can be found in my Github repository and to report bugs/issues please use the following link, https://github.com/mlampros/CopernicusDEM/issues.

If you use the **CopernicusDEM** R package in your paper or research please cite https://cran.r-project.org/web/packages/CopernicusDEM/citation.html:

```
@Manual{,
  title = {CopernicusDEM: Copernicus Digital Elevation Models},
  author = {Lampros Mouselimis},
  year = {2021},
  note = {R package version 1.0.1 produced using Copernicus
    WorldDEMTM-90 DLR e.V. 2010-2014 and Airbus Defence and Space
    GmbH 2014-2018 provided under COPERNICUS by the European Union
    and ESA; all rights reserved},
  url = {https://CRAN.R-project.org/package=CopernicusDEM},
}…
```