PTED / Exercise 2

Gioele Pinana

2022-04-29

Task 1: Import your Data

Load the necessary libraries

```
library(readr)        # to import tabular data (e.g. csv)
library(dplyr)        # to manipulate (tabular) data
library(ggplot2)      # to visualize data
library(sf)           # to handle spatial vector data
library(terra)        # To handle raster data
library(lubridate)    # To handle dates and times
```

Import the downloaded csv

```
wildschwein_BE <- read_delim("wildschwein_BE_2056.csv",",") # adjust path
```

```
## Rows: 51246 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): TierID, TierName
## dbl  (3): CollarID, E, N
## dttm (1): DatetimeUTC
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
wildschwein_BE <- st_as_sf(wildschwein_BE, coords = c("E", "N"), crs = 2056, remove = FALSE)
```

Task 2: Getting an Overview

time difference between subsequent rows. I used the function difftime() in combination with the function lead(). difftime calculates the time interval between to consecutive points and returns an object of the Class difftime, with as.integer I define that I want the output returned into an integer. Lead () return a vector of the same length as the input.

```
wildschwein_BE$timelag  <- as.integer(difftime(lead(wildschwein_BE$DatetimeUTC), wildschwein_BE$Datetim
wildschwein_BE
```

```
## Simple feature collection with 51246 features and 7 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 2568153 ymin: 1202306 xmax: 2575154 ymax: 1207609
## Projected CRS: CH1903+ / LV95
## # A tibble: 51,246 x 8
##    TierID TierName CollarID DatetimeUTC                E        N
##  * <chr>  <chr>       <dbl> <dttm>                 <dbl>    <dbl>
##  1 002A   Sabi        12275 2014-08-22 21:00:12 2570409. 1204752.
##  2 002A   Sabi        12275 2014-08-22 21:15:16 2570402. 1204863.
##  3 002A   Sabi        12275 2014-08-22 21:30:43 2570394. 1204826.
##  4 002A   Sabi        12275 2014-08-22 21:46:07 2570379. 1204817.
##  5 002A   Sabi        12275 2014-08-22 22:00:22 2570390. 1204818.
##  6 002A   Sabi        12275 2014-08-22 22:15:10 2570390. 1204825.
##  7 002A   Sabi        12275 2014-08-22 22:30:13 2570387. 1204831.
##  8 002A   Sabi        12275 2014-08-22 22:45:11 2570381. 1204840.
##  9 002A   Sabi        12275 2014-08-22 23:00:27 2570316. 1204935.
## 10 002A   Sabi        12275 2014-08-22 23:15:41 2570393. 1204815.
## # ... with 51,236 more rows, and 2 more variables: geometry <POINT [m]>,
## #   timelag <int>
```

- How many individuals were tracked? the function count () lets you count the unic value of a variable

```
wildschwein_BE %>% count(TierName)
```

```
## Simple feature collection with 3 features and 2 fields
## Geometry type: MULTIPOINT
## Dimension:     XY
## Bounding box:  xmin: 2568153 ymin: 1202306 xmax: 2575154 ymax: 1207609
## Projected CRS: CH1903+ / LV95
## # A tibble: 3 x 3
##   TierName     n                                                  geometry
## * <chr>    <int>                                          <MULTIPOINT [m]>
## 1 Rosa     14364 ((2569231 1205823), (2569245 1205925), (2569247 1206027), (256~
## 2 Ruth     14136 ((2568153 1205611), (2568155 1205613), (2568161 1205624), (256~
## 3 Sabi     22746 ((2568903 1206200), (2568925 1206207), (2568980 1206197), (256~
```

3 individuals were tracked: Rosa, Ruth and Sabi

- For how long were the individual tracked? Are there gaps?

```
summary(wildschwein_BE$DatetimeUTC)
```

```
##                  Min.               1st Qu.                Median
## "2014-08-22 21:00:12" "2015-02-20 12:18:57" "2015-04-08 20:00:10"
##                  Mean               3rd Qu.                  Max.
## "2015-03-24 23:09:22" "2015-06-01 05:45:13" "2015-07-27 11:00:14"
```

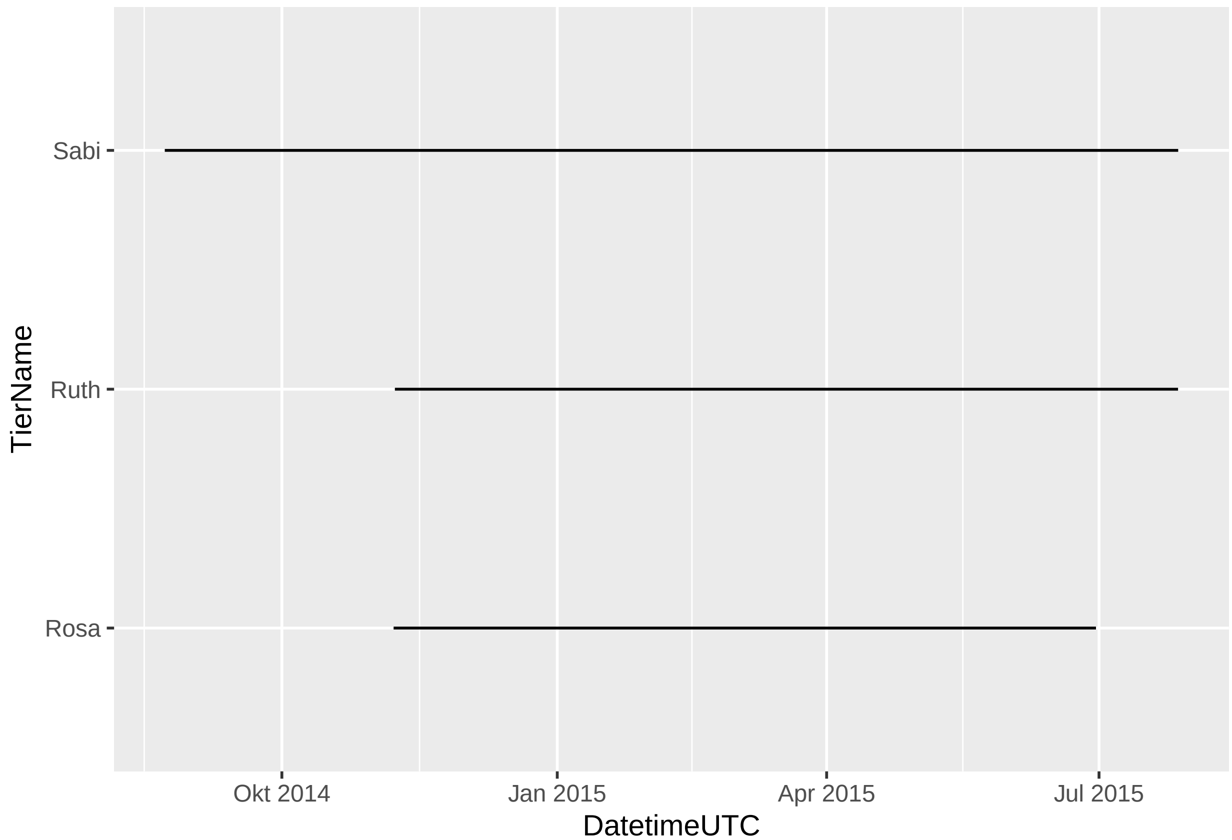
```
summary(wildschwein_BE$timelag)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
## -22648470       896       903       571       916     60367         1
```

From 22. August 2014 until 27. Juli 2015. There is one missing values but also some outliers.

- Were all individuals tracked concurrently or sequentially?

```
ggplot(wildschwein_BE, aes(DatetimeUTC, TierName))+
  geom_line()
```



Sabi from 08.2014 until 07.2015, Rosa from 11.2014 until 06.2015 and Ruth from 11.2014 until 07.2015.

- What is the temporal sampling interval between the locations? About 15 minutes.

Task 3: Deriving movement parameters I: Speed

The formula to calculate the velocity is v = euclidean distance / time. distance. I defined what E1, E2, N1 and N2 are and then I first calculated the distance between two consecutive point using the formula given in the task. Finally I then calculated the speed and put the results in a new column.

```
n=dim(wildschwein_BE)[1] #or
n=51247

E1=wildschwein_BE$E[1:(n-1)]
E2=wildschwein_BE$E[2:n]
N1=wildschwein_BE$N[1:(n-1)]
N2=wildschwein_BE$N[2:n]
```

```
dist=sqrt((E1-E2)^2+(N1-N2)^2)

wildschwein_BE$steplenght <- dist

wildschwein_BE$speed <- dist/wildschwein_BE$timelag
# speed is calculated in m/s
```

Task 4: Cross-scale movement analysis

To do the cross-scale movement analysis I first imported the new data set and I calculated timelag, steplength and speed using the same method I used in task 3 for the timespan of 1 minute.

```
caro <- read_delim("caro60.csv",",")
```

```
## Rows: 200 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): TierID, TierName
## dbl  (3): CollarID, E, N
## dttm (1): DatetimeUTC
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
caro <- st_as_sf(caro, coords = c("E", "N"), remove = FALSE)

caro$timelag  <- as.integer(difftime(lead(caro$DatetimeUTC), caro$DatetimeUTC, units = c("secs")))

n=201
E1=caro$E[1:(n-1)]
E2=caro$E[2:n]
N2=caro$N[2:n]
N1=caro$N[1:(n-1)]
dist=sqrt((E1-E2)^2+(N1-N2)^2)

caro$steplenght <- dist

caro$speed <- dist/caro$timelag
```

To Analyse animal movement across different scales I had to reduce the granularity of the data by sub-setting the data to every nth element, I did it every 3rd, 6th and 9th position using slice() and seq() where from = 1, to = the length of the dataset and by = n.

- slice(): lets you index rows by their (integer) locations. It allows you to select, remove, and duplicate rows.
- seq(): Generate regular sequences.

Using the same method as above I also calculated the timelag, steplength and speed for each data frame.

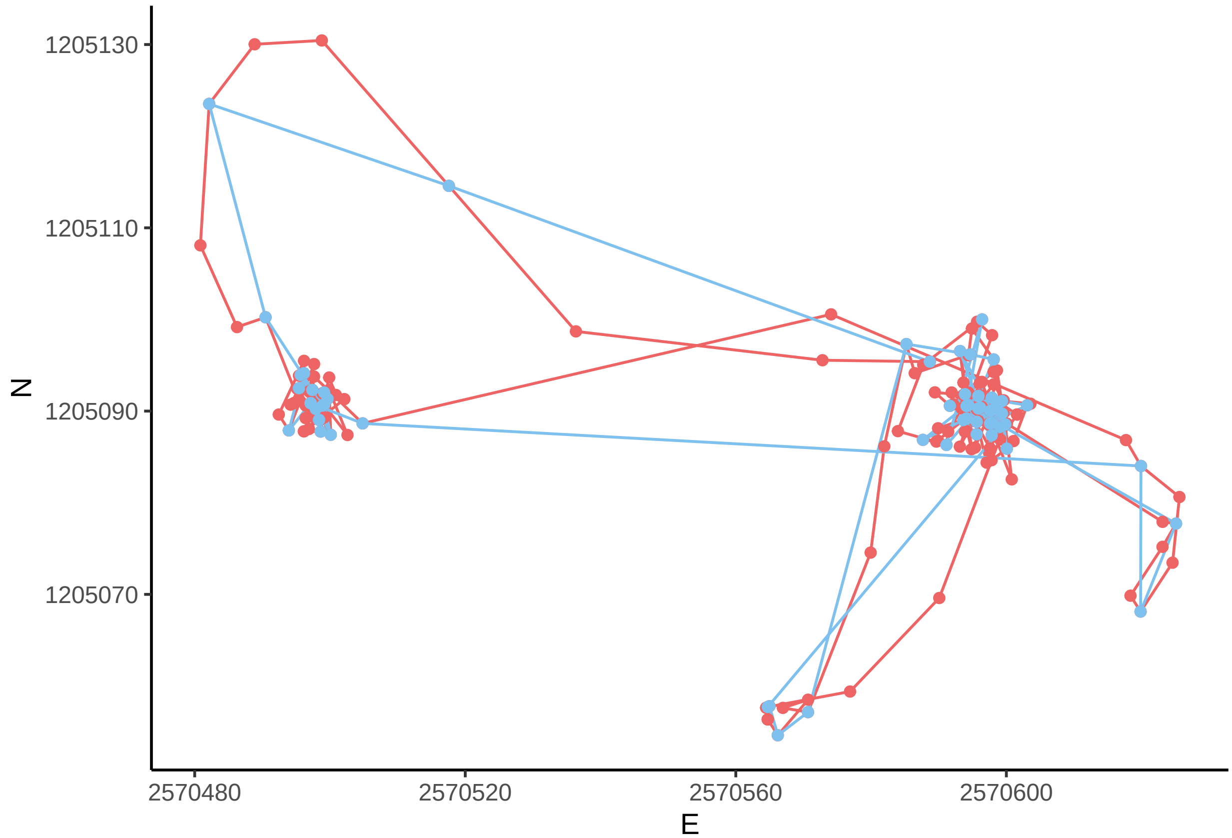
```
# 3
caro_3 <- caro %>%
  slice(seq(from = 1, to = 200, by = 3))

caro_3$timelag  <- as.integer(difftime(lead(caro_3$DatetimeUTC), caro_3$DatetimeUTC, units = c("secs"))

n=68
E1=caro_3$E[1:(n-1)]
E2=caro_3$E[2:n]
N2=caro_3$N[2:n]
N1=caro_3$N[1:(n-1)]
dist=sqrt((E1-E2)^2+(N1-N2)^2)

caro_3$steplenght <- dist

caro_3$speed <- dist/caro_3$timelag

# 6
caro_6 <- caro %>%
  slice(seq(from = 1, to = 200, by = 6))

caro_6$timelag  <- as.integer(difftime(lead(caro_6$DatetimeUTC), caro_6$DatetimeUTC, units = c("secs"))

n=35
E1=caro_6$E[1:(n-1)]
E2=caro_6$E[2:n]
N2=caro_6$N[2:n]
N1=caro_6$N[1:(n-1)]
dist=sqrt((E1-E2)^2+(N1-N2)^2)

caro_6$steplenght <- dist

caro_6$speed <- dist/caro_6$timelag

# 9
caro_9 <- caro %>%
  slice(seq(from = 1, to = 200, by = 9))

caro_9$timelag  <- as.integer(difftime(lead(caro_9$DatetimeUTC), caro_9$DatetimeUTC, units = c("secs"))

n=24
E1=caro_9$E[1:(n-1)]
E2=caro_9$E[2:n]
N2=caro_9$N[2:n]
N1=caro_9$N[1:(n-1)]
dist=sqrt((E1-E2)^2+(N1-N2)^2)

caro_9$steplenght <- dist

caro_9$speed <- dist/caro_9$timelag
```

Now it make sense to compare the original data with the re sampled data. To do this, I used the package ggplot2 and a different combination of plotting fuction:

- geom_path(): to plot the trajectories, it connets the observation in the order in which they appear in the data.
- geom_point(): to plot the locations.

plot - Comparing original with 3 minutes re sampled data

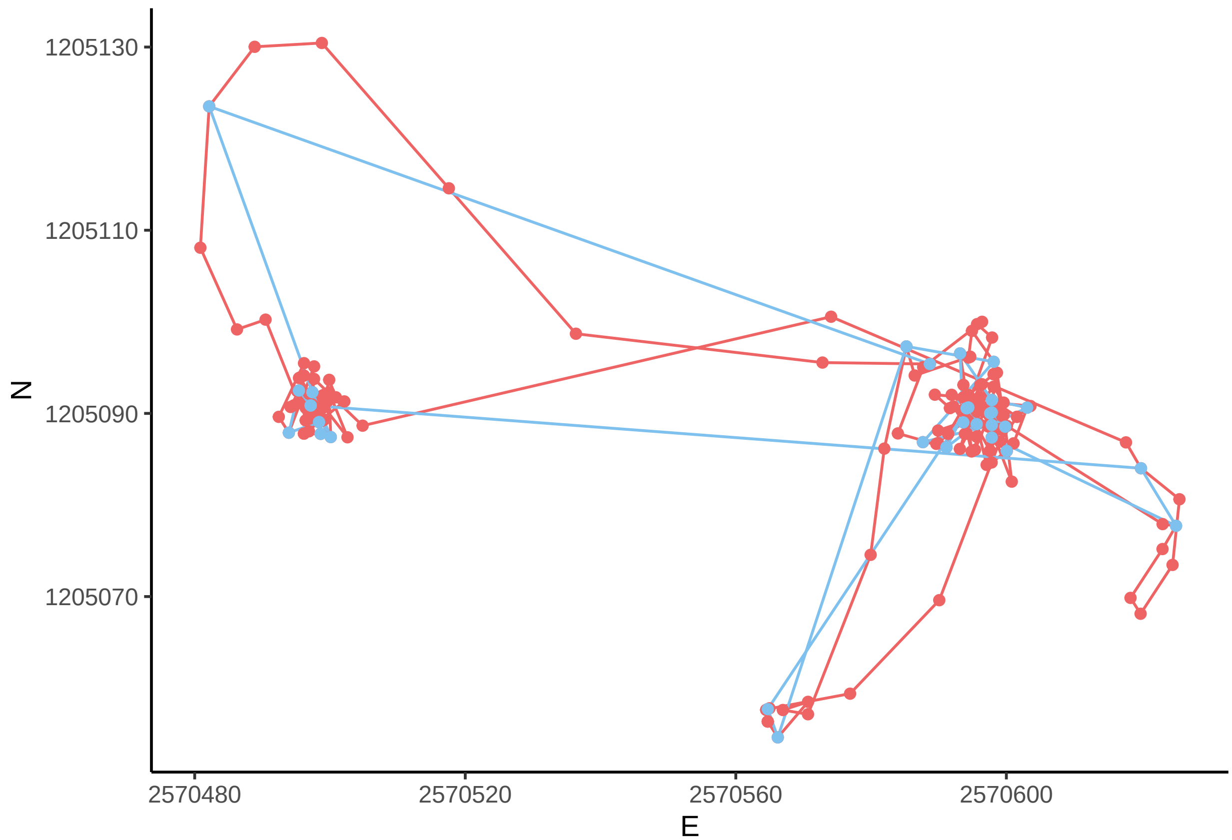
```
ggplot() +
  geom_path(data=caro, aes(x = E, y = N), color= "indianred2") +
  geom_path(data=caro_3, aes(x = E,y = N), color = "skyblue2") +
  geom_point(data=caro, aes(x = E, y = N), color= "indianred2") +
  geom_point(data=caro_3, aes(x = E,y = N), color = "skyblue2") +
  theme_classic() +
  scale_color_manual(name='Trajectory',
                     breaks=c('1 minute', '3 minutes'),
                     values=c('1 minute'='indianred2', '3 minutes'='skyblue2'))
```



plot - Comparing original with 6 minutes resampled data

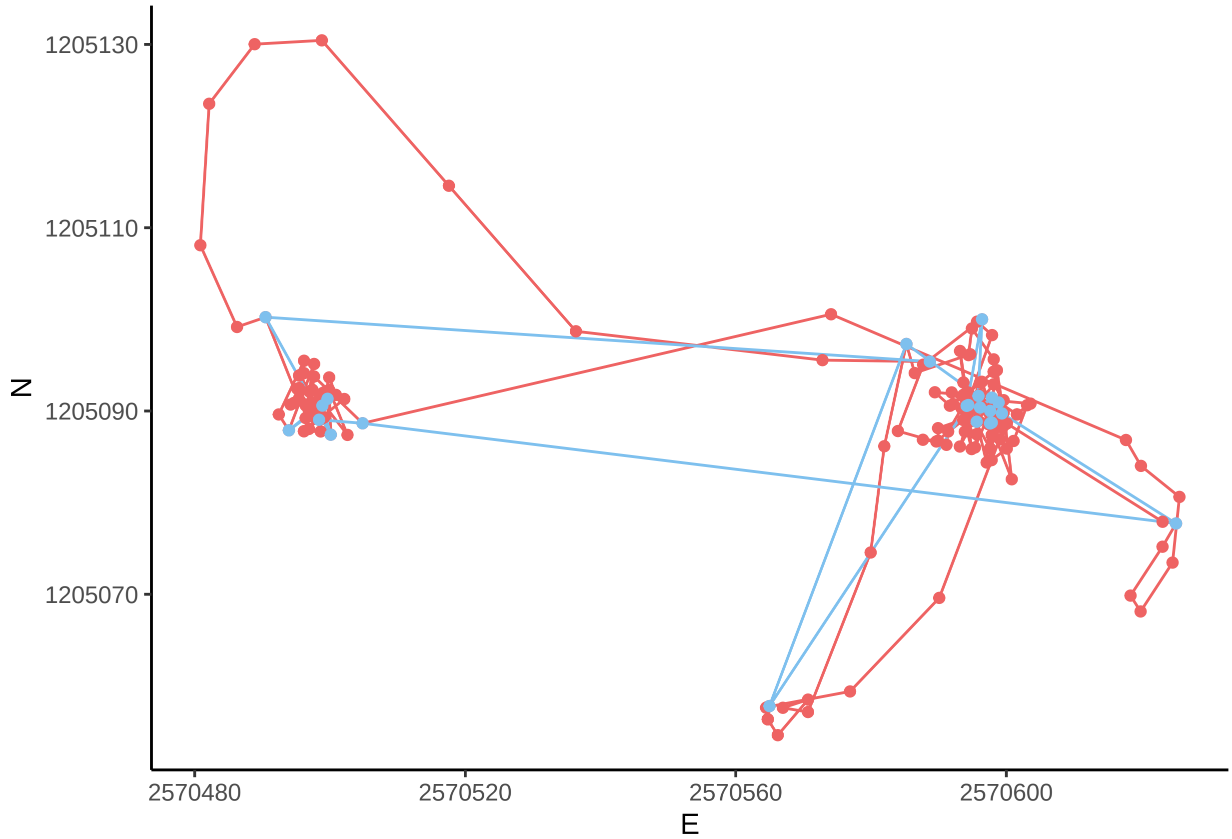
```
ggplot() +
  geom_path(data=caro, aes(x = E, y = N), color= "indianred2") +
  geom_path(data=caro_6, aes(x = E,y = N), color = "skyblue2") +
  geom_point(data=caro, aes(x = E, y = N), color= "indianred2") +
  geom_point(data=caro_6, aes(x = E,y = N), color = "skyblue2") +
  theme_classic() +
  scale_color_manual(name='Trajectory',
```

```
                     breaks=c('1 minute', '6 minutes'),
                     values=c('1 minute'='indianred2', '6 minutes'='skyblue2'))
```



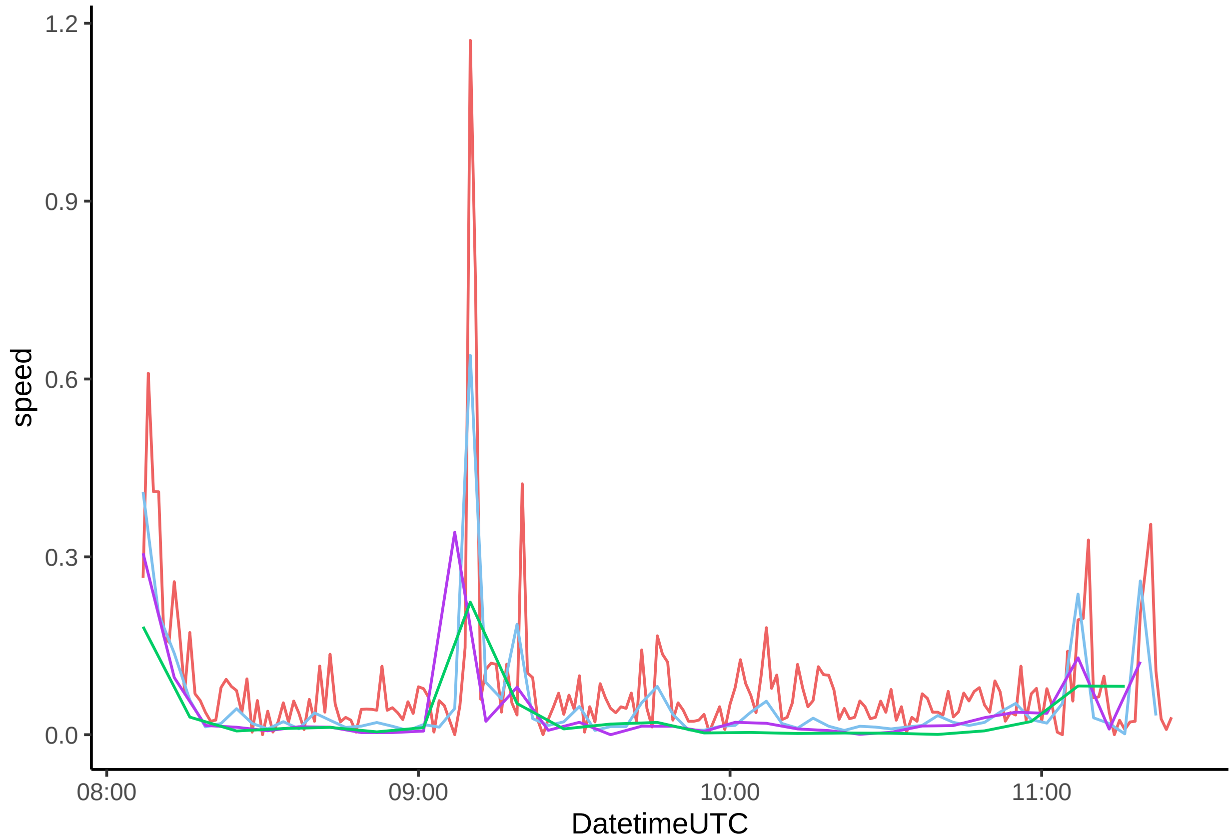
plot - Comparing original with 9 minutes resampled data

```
ggplot() +
  geom_path(data=caro, aes(x = E, y = N), color= "indianred2") +
  geom_path(data=caro_9, aes(x = E,y = N), color = "skyblue2") +
    geom_point(data=caro, aes(x = E, y = N), color= "indianred2") +
  geom_point(data=caro_9, aes(x = E,y = N), color = "skyblue2") +
  theme_classic() +
  scale_color_manual(name='Trajectory',
                     breaks=c('1 minute', '9 minutes'),
                     values=c('1 minute'='indianred2', '9 minutes'='skyblue2'))
```



Plot - comparing derived speed at different sampling interval

```
ggplot() +
  geom_path(data=caro, aes(x = DatetimeUTC, y = speed), color= "indianred2") +
  geom_path(data=caro_3, aes(x = DatetimeUTC, y = speed), color= "skyblue2") +
  geom_path(data=caro_6, aes(x = DatetimeUTC, y = speed), color= "darkorchid2") +
  geom_path(data=caro_9, aes(x = DatetimeUTC, y = speed), color= "springgreen3")+
    scale_color_manual(name='Colour',
                     breaks=c('1 minute', '3 minutes','6 minute', '9 minutes'),
                     values=c('indianred2', 'skyblue2','darkorchid2', 'springgreen3')) +
  theme_classic()
```



In these Graphs we see the influence of temporal scale on movement parameters such as speed, sinuosity, or turning angle. Exploring the degree of variation will be possible through the use of box whisker plots. In addition, the influence of uncertainty must be considered in order to draw more appropriate conclusions (Laube and Purves, 2011). The last graph shows a general decrease in speed as the sampling interval increases (red= original data, blue= 3 minutes re sampled data, violet = 6 minutes re sampled data and green= 9 minutes re sampled data).

Task 5: Deriving movement parameters II: Rolling window functions

Different approach using a moving window function (wanders over and always determines in a neighborhood the value).

Loading the package zoo

```
library(zoo)
```

roll_mean() to smooth the calculated speed, using different k, which is the integer width of the rolling window.

Some example with dummy data:

```
example <- rnorm(10)
rollmean(example,k = 3,fill = NA,align = "left")
```

```
##  [1] -0.8804999 -0.9512601  0.3192971  1.4795636  0.7429030  0.5081411
##  [7] -0.7165593 -0.4495363         NA         NA
```

```r
rollmean(example,k = 4,fill = NA,align = "left")
```

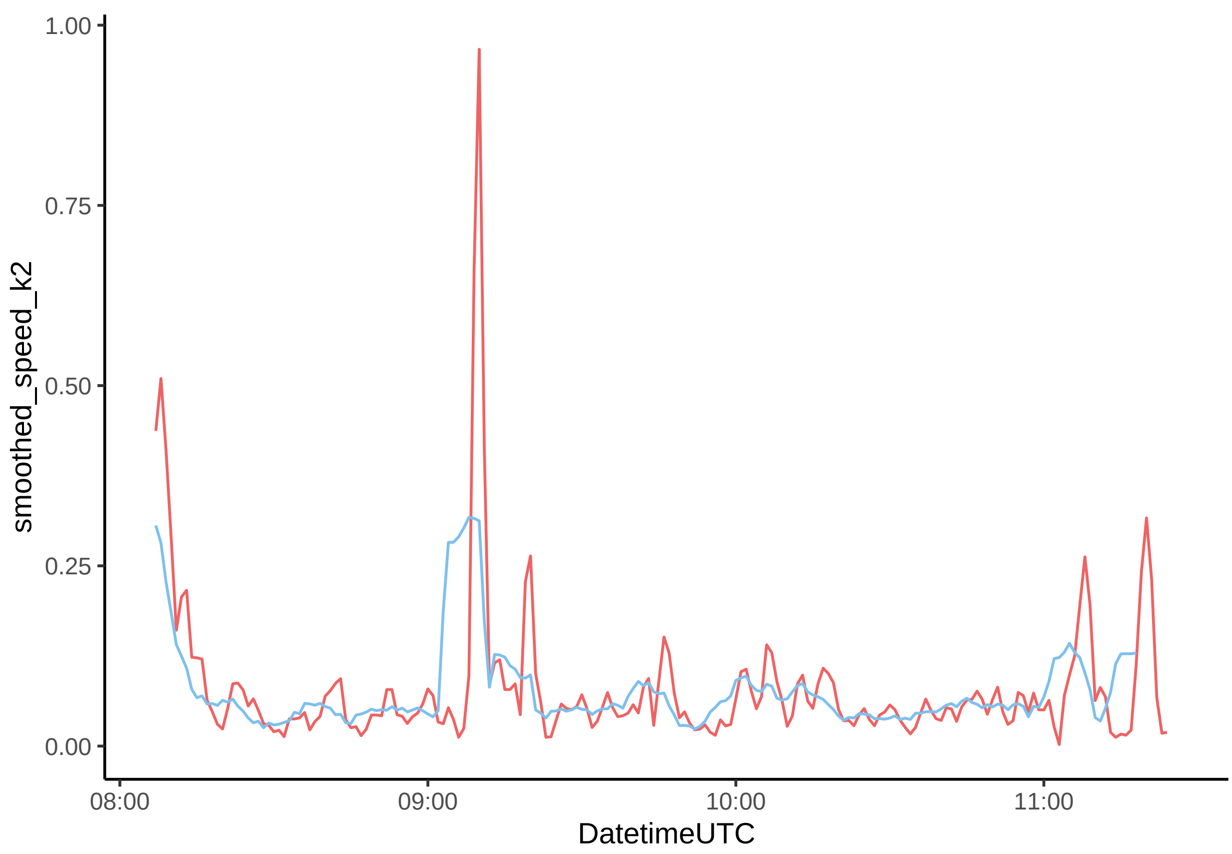
```
##  [1] -0.33758914 -0.48718973  0.80010443  0.87996307  0.60736116  0.02321212
##  [7] -0.56686186          NA          NA          NA
```

roll_mean() on speed, using the data set caro:

```r
caro$smoothed_speed_k2 <- rollmean(caro$speed,k = 2,fill = NA,align = "left")

caro$smoothed_speed_k3 <- rollmean(caro$speed,k = 3,fill = NA,align = "left")

caro$smoothed_speed_k4 <- rollmean(caro$speed,k = 4,fill = NA,align = "left")

caro$smoothed_speed_k8 <- rollmean(caro$speed,k = 8,fill = NA,align = "left")
```

Visualisation k3 and k8

```r
ggplot() +
  geom_path(data=caro, aes(x = DatetimeUTC, y = smoothed_speed_k2), color= "indianred2")+
  geom_path(data=caro, aes(x = DatetimeUTC, y = smoothed_speed_k8), color= "skyblue2")+
  theme_classic()
```



References

Laube P, Ross S. P. (2011) How fast is a cow? Cross-Scale Analysis of Movement Data. Transaction in gis. 15(3): 401–418