Introduction

Johannes Signer & Brian Smith

March 2025

Welcome!

- Welcome to this online course on animal movement.
- We are Brian and Johannes.
- Who are you? *Pair up and introduce your partner*
- Ask your partner:
  - What is your background?
  - Where do you come from?
  - Where do you study/work?
  - What is your study organism?
  - Why do you attend this course?

Outline of the course

Day 1:

- Introduction and exploratory data analysis for movement data (J)
- Data cleaning (B)

Day 2:

- Quantifying space use of animals with home ranges (B)
- Multiple instances (J)

Day 3:

- Introduction to habitat selection (B)
- Integrated step selection functions 1 (J)

Day 4:

- Integrated step selection functions 2 (B)
- Simulations from fitted iSSFs (J)

Day 5:

- Advanced (i)SSF topics (J)
- Validation of models for habitat selection (B)
- Time to discuss questions related to **your** projects.

Some logistics

- The course is scheduled from Monday (24th of March) to Friday (28th of March) from 2pm to 6pm Berlin time.
- We split these 4h block into two chunks, each roughly structured like this:
  - Lecture $\sim$ 45 min
  - R walkthrough $\sim$ 45 min
  - Introduction of exercises $\sim$ 5 min
- A 20 min break between the two chunks.
- Lectures will be held via zoom.
- During the whole workshop we have a slack channel where you ask questions (we will monitor the channel during the course, feel free to ask questions there also outside the course hours).

Analysis of movement data in R

- The statistical software package R has become a widely used tool for data analysis in ecology and evolution and also in movement ecology.
- Also visit R task view for tracking data: `https://cran.r-project.org/web/views/Tracking.html`

- A typical analysis usually undergoes a few steps (all of which can be performed in R), this was reviewed by Joo et al. 2020.

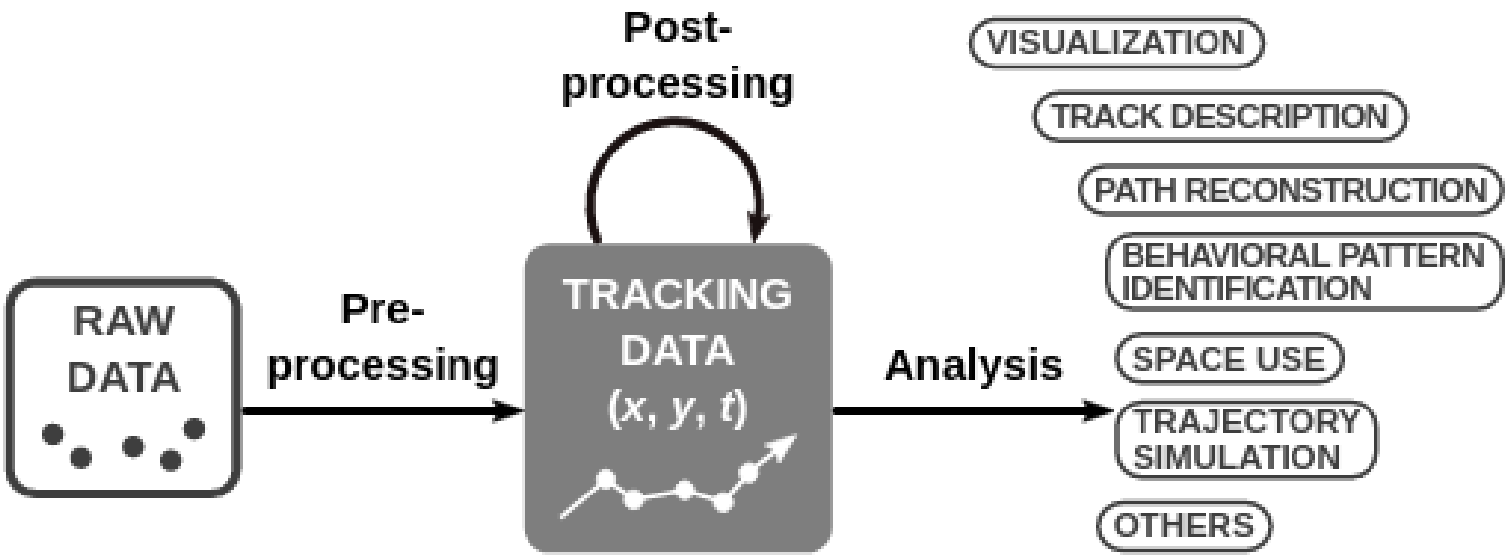


Figure 1: Figure from Joo et al 2020

Packages that we will use

- We will mainly use the `amt` package, but also occasionally other packages for movement analysis.
- See also `required_packages.R` for a list of all packages that we need and to get the latest version of all packages.

Some conventions in R

- %>% or |>: Pipes the output of one function to a next function. We will discuss this further later on.
- `::` to access a name space form a package.
- use of ‘a‘.
- `.` means this directory
- `..` refers to the parent directory
- `data.frame` or `tibble`?

- We often use `here::here("path to a file")`, when reading in a file.
- The first `here` calls the function `here()` from the package `here`.
- The function `here()` dynamically creates the absolute path to the project root.

```
here::here()
```

```
[1] "/Users/jsigner/git/movement_workshop"
```

This means, that we save all our data in the root directory data (even though my scripts are in different sub directories).

Brackets (`(`, `[`, `{`)

- round brackets or parentheses (`()`) usually indicate functions or are used in arithmetic calculations.

```
sqrt(3)
```

```
[1] 1.732051
```

or

```
2 * (3 + 1)
```

```
[1] 8
```

- square brackets (`[`, `[[`) are used to subset data structures.

```
letters[1:3]
```

```
[1] "a" "b" "c"
```

or

```
head(iris[["Species"]])
```

```
[1] setosa setosa setosa setosa setosa setosa
Levels: setosa versicolor virginica
```

- curly brackets or braces ({) are used to form code blocks (e.g., inside a function or control structure).

```
for (i in 1:10) {
  i^2
}
```

Functions

- Functions *do* something. For example function `sqrt()` takes the square root for a number.
- It is easy to recognize functions, because they usually have a name (e.g., `sqrt`) followed by round brackets `()`.
- Within these round brackets arguments are passed to a function. This arguments can be named or unnamed (as long as they are in the correct order).

Recommended setup

- We would recommend to download the whole repository from GitHub (`https://github.com/jmsigner/movement_workshop_spring2025`)[1].
- Then use the RStudio project (together with RStudio).
- Following these guides, you should have all paths correct.

[1] If you are familiar with git, feel free to clone the repository

Geographic data in brief

- Movement data is inherently spatial.
- Thus we will have to deal with tools to work with spatial data (R has a rich set of tools to deal with spatial data; e.g. https://geocompr.robinlovelace.net/).
- We will work frequently with raster data (spatial covariates) and possibly with vector data (i.e., home ranges).
- One of the challenges is to ensure that both – tracking data and covariates – have a matching coordinate reference system (CRS).

- The CRS defines the reference system that is being used to explicitly reference a feature in space.
- There are two classes of CRS: **geographic** (e.g., WGS84) and **projected** (e.g., UTM) CRS.
- Projected CRS flatten the three dimensional data to the a two-dimensional plane (and introduce some distortion).

[2] EPSG stands for European Petrol Survey Group, who came up with the system.

- The CRS defines the reference system that is being used to explicitly reference a feature in space.
- There are two classes of CRS: **geographic** (e.g., WGS84) and **projected** (e.g., UTM) CRS.
- Projected CRS flatten the three dimensional data to the a two-dimensional plane (and introduce some distortion).
- CRS are often referred to with their EPSG[2] code.
- EPSG codes are four to five digits number that refer to different CRS. See for example www.epsg.io.

[2] EPSG stands for European Petrol Survey Group, who came up with the system.

- The CRS defines the reference system that is being used to explicitly reference a feature in space.
- There are two classes of CRS: **geographic** (e.g., WGS84) and **projected** (e.g., UTM) CRS.
- Projected CRS flatten the three dimensional data to the a two-dimensional plane (and introduce some distortion).
- CRS are often referred to with their EPSG[2] code.
- EPSG codes are four to five digits number that refer to different CRS. See for example www.epsg.io.
- Which CRS is best to use? It depends on the range of the study species. I usually prefer projected CRS, because their units are meters and not degrees.

[2] EPSG stands for European Petrol Survey Group, who came up with the system.

Data

Movement data

- Often times the data we receive is just a time series of coordinates (longitude, latitude and time stamp).
- Depending on the sensors we use, other (meta) information may also be stored (this could include temperature, coordinates in a different [projected] CRS, . . .).

```
No;CollarID;UTC_Date;UTC_Time;LMT_Date;LMT_Time;Origin;SCTS_Date;SCTS_Time;ECEF_X [m];ECEF_Y [m];ECEF_Z
13563;9977;30.03.2020;10:01:01;30.03.2020;10:01:01;Collar;11.05.2021;09:15:07;4040415;851181;4845535;49,
13562;9977;30.03.2020;09:01:06;30.03.2020;09:01:06;Collar;11.05.2021;09:15:07;4040398;851190;4845551;49,
13561;9977;30.03.2020;08:02:12;30.03.2020;08:02:12;Collar;11.05.2021;09:15:07;4040451;851194;4845498;49,
13560;9977;30.03.2020;07:01:18;30.03.2020;07:01:18;Collar;11.05.2021;09:15:07;4040383;851193;4845554;49,
13559;9977;30.03.2020;06:01:56;30.03.2020;06:01:56;Collar;11.05.2021;09:15:07;4040403;851166;4845550;49,
13558;9977;30.03.2020;05:01:04;30.03.2020;05:01:04;Collar;11.05.2021;09:15:07;4040452;851209;4845535;49,
13557;9977;30.03.2020;04:00:31;30.03.2020;04:00:31;Collar;11.05.2021;09:15:07;4040388;851194;4845495;49,
13556;9977;30.03.2020;03:01:09;30.03.2020;03:01:09;Collar;11.05.2021;09:15:07;4040401;851191;4845489;49,
13555;9977;30.03.2020;02:01:04;30.03.2020;02:01:04;Collar;11.05.2021;09:15:07;4040394;851198;4845524;49,
13554;9977;30.03.2020;01:01:07;30.03.2020;01:01:07;Collar;11.05.2021;09:15:07;4040417;851185;4845543;49,
13553;9977;30.03.2020;00:01:07;30.03.2020;00:01:07;Collar;11.05.2021;09:15:07;4040472;851176;4845521;49,
```

Environmental covariates

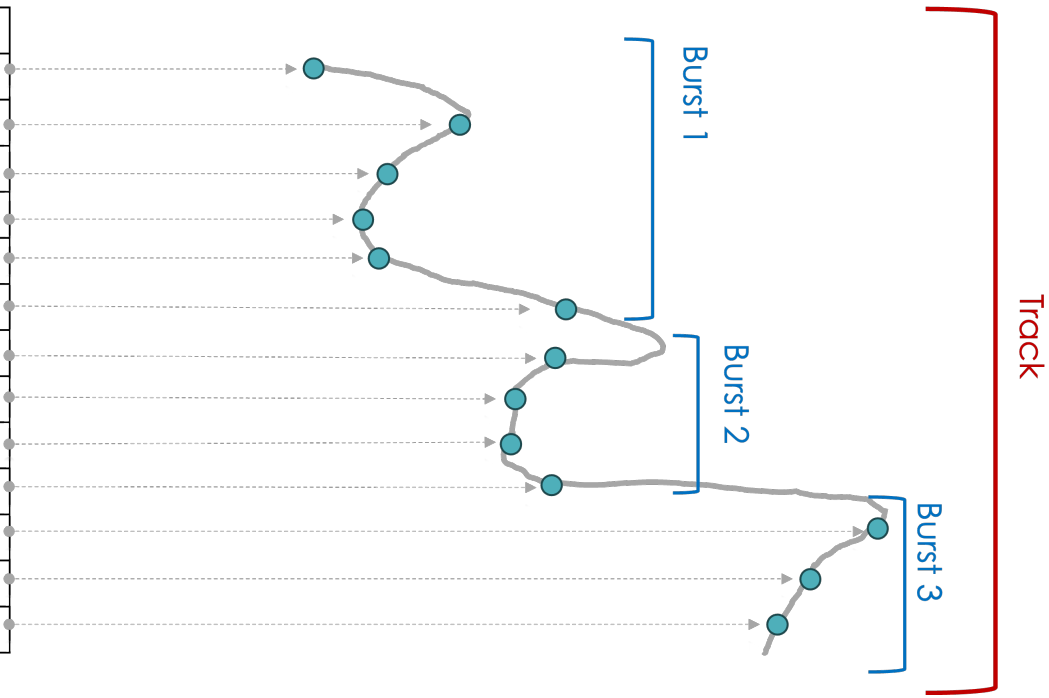
- Vector layers (e.g., road networks, rivers, protected areas)
- Raster layers (e.g., land use, remotely sensed data such as NDVI, climatic variables)

Tracks and bursts: the basic building block

- For movement data, we usually read a text file into R (`.csv`, `.txt`, ...) as a data frame and then create a analysis-specific object.
- When working with the `amt` package the function `make_track()` takes a sequence of coordinates (with or without timestamps) and creates a track. Note, at this point multiple individuals can be mixed and sampling rates can be heterogeneous.

- **Bursts** can be created from tracks.
- A burst is a sequence of (re)locations from the **same** individual at **equal** time intervals (with some tolerance).

Time	Burst
2024-01-01 06:00	1
2024-01-01 07:00	1
2024-01-01 08:00	1
2024-01-01 09:00	1
2024-01-01 10:00	1
2024-01-01 11:00	1
2024-01-01 14:00	2
2024-01-01 15:00	2
2024-01-01 16:00	2
2024-01-01 17:00	2
2024-01-01 20:00	3
2024-01-01 21:00	3
2024-01-01 22:00	3

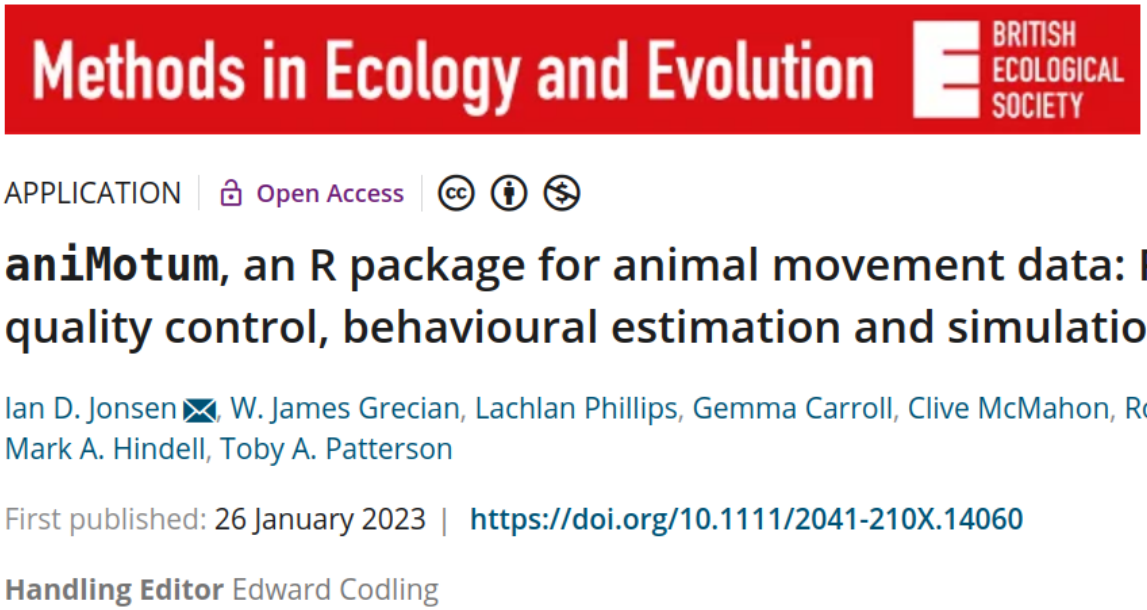


Sampling rates, and resampling

- The function `summarize_sampling_rate()` takes a track as input and gives a summary of the sampling rate.
- If there are multiple animals present, there is also the function `summarize_sampling_rate_many()`, which will do the same thing, but for many animals.

- Once a suitable sampling rate is determined, the function `track_resample()` can be used to take a relocation every predefined time interval (e.g., 30 minutes, 2 hours, ...) within a tolerance.
- The result of `track_resample()` is again a track with one additional column called `burst_`.

If you have gaps and/or different sampling rates, interpolation with continuous time movement models may be an option.



Methods in Ecology and Evolution

BRITISH ECOLOGICAL SOCIETY

APPLICATION | Open Access

aniMotum, an R package for animal movement data: Rapid quality control, behavioural estimation and simulation

Ian D. Jonsen, W. James Grecian, Lachlan Phillips, Gemma Carroll, Clive McMahon, Robert G. Harcourt, Mark A. Hindell, Toby A. Patterson

First published: 26 January 2023 | https://doi.org/10.1111/2041-210X.14060

Handling Editor Edward Codling

An more specifically for Step-Selection Functions

Methodology | Open access | Published: 09 May 2024

Methods for implementing integrated step-selection functions with incomplete data

David D. Hofmann ✉, Gabriele Cozzi & John Fieberg

Movement Ecology **12**, Article number: 37 (2024) | Cite this article

2688 Accesses | Metrics

Movement characteristics (`sl_`, `ta_`)

Tracks are still *just* the points as they were collected. If we want to get insights, we have can look at different characteristics of steps (i.e., two consecutive relocations).
This include:

- step length
- turn angle
- speed
- Net squared displacement

Note, unless you take care of different instances or bursts, they are ignored.

Net Squared displacement (NSD)

- The NSD is the squared distance between the first relocation of a track and the every relocation that follows.
- Bunnefeld et al. 2011 described different forms of the NSD that resemble different migratory behaviors.
- The different models can fit to the data (e.g., using nonlinear least square with the function `nls()` in R).

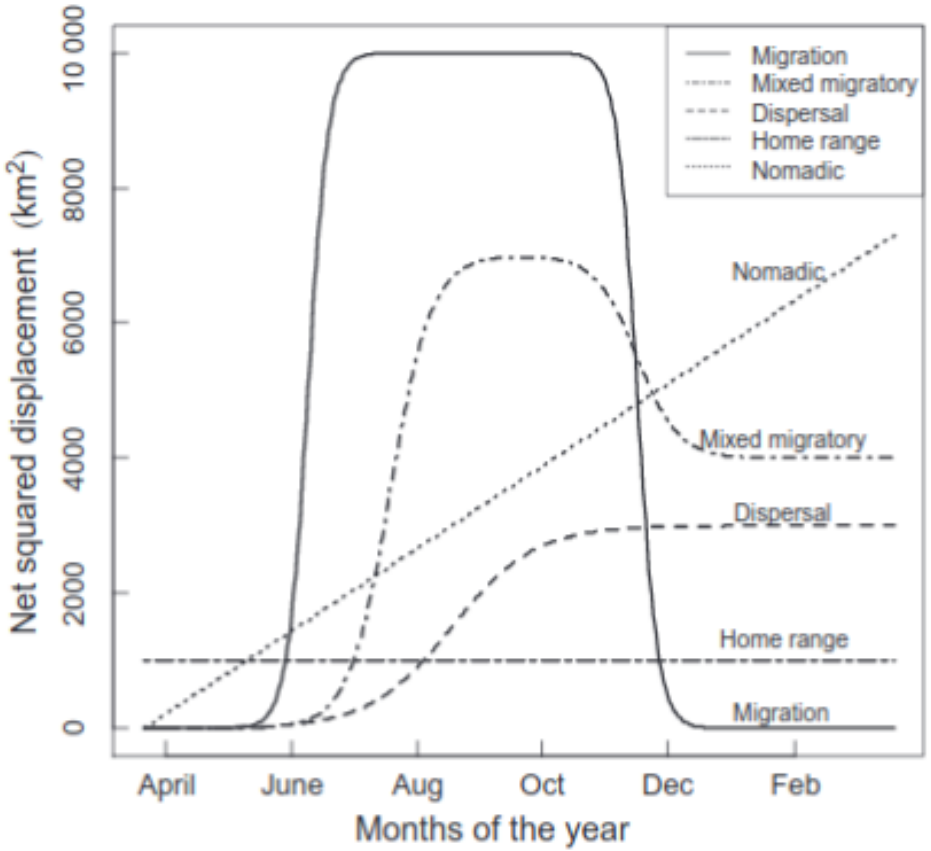


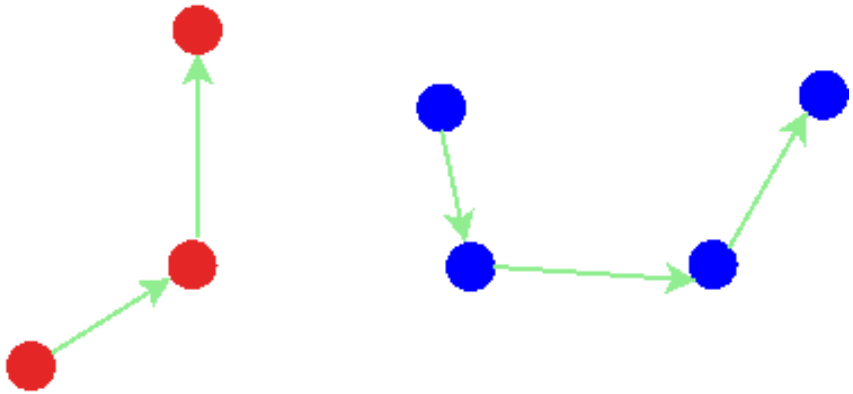
Figure 2: Figure taken from Bunnefeld et al. 2011

Time of the day

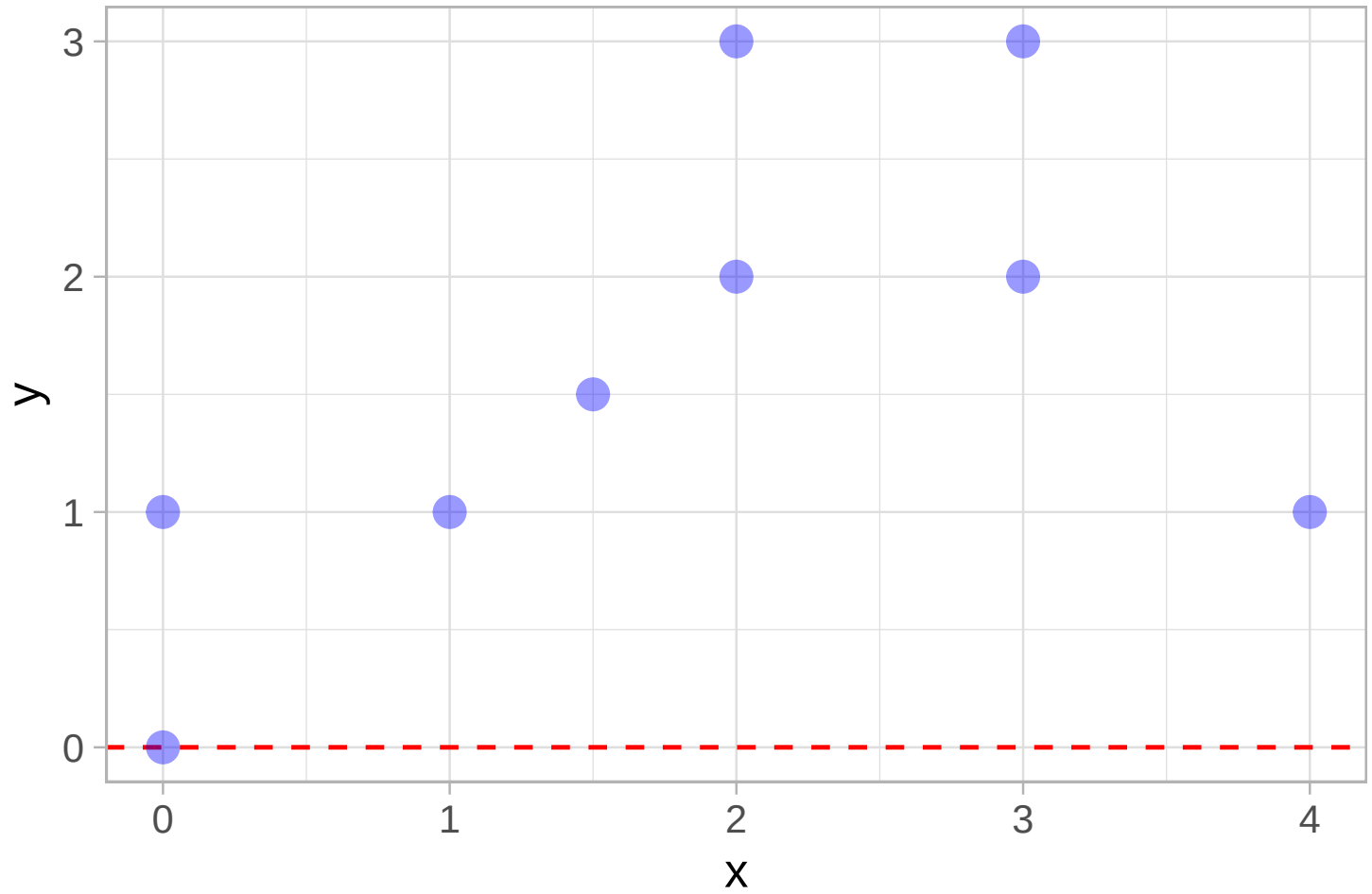
- Time of day can be annotated to steps with the function `amt::time_of_day()`. This will add an additional column to the data frame of steps `tod_end` or `tod_start` depending on the argument `when`.
- If the data is of sufficient temporal resolution, it is also possible to annotate twilight (dawn and dusk).

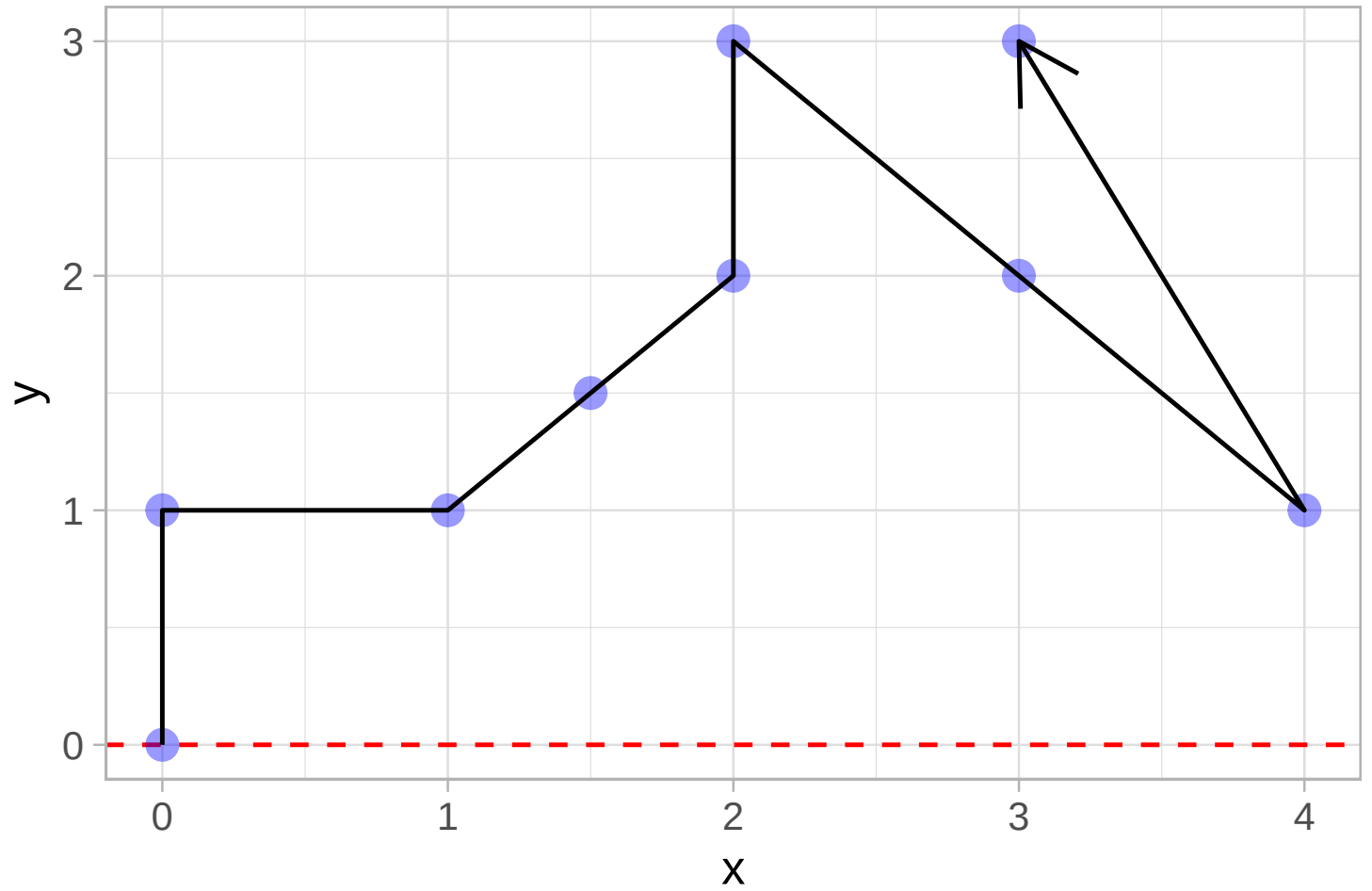
Steps

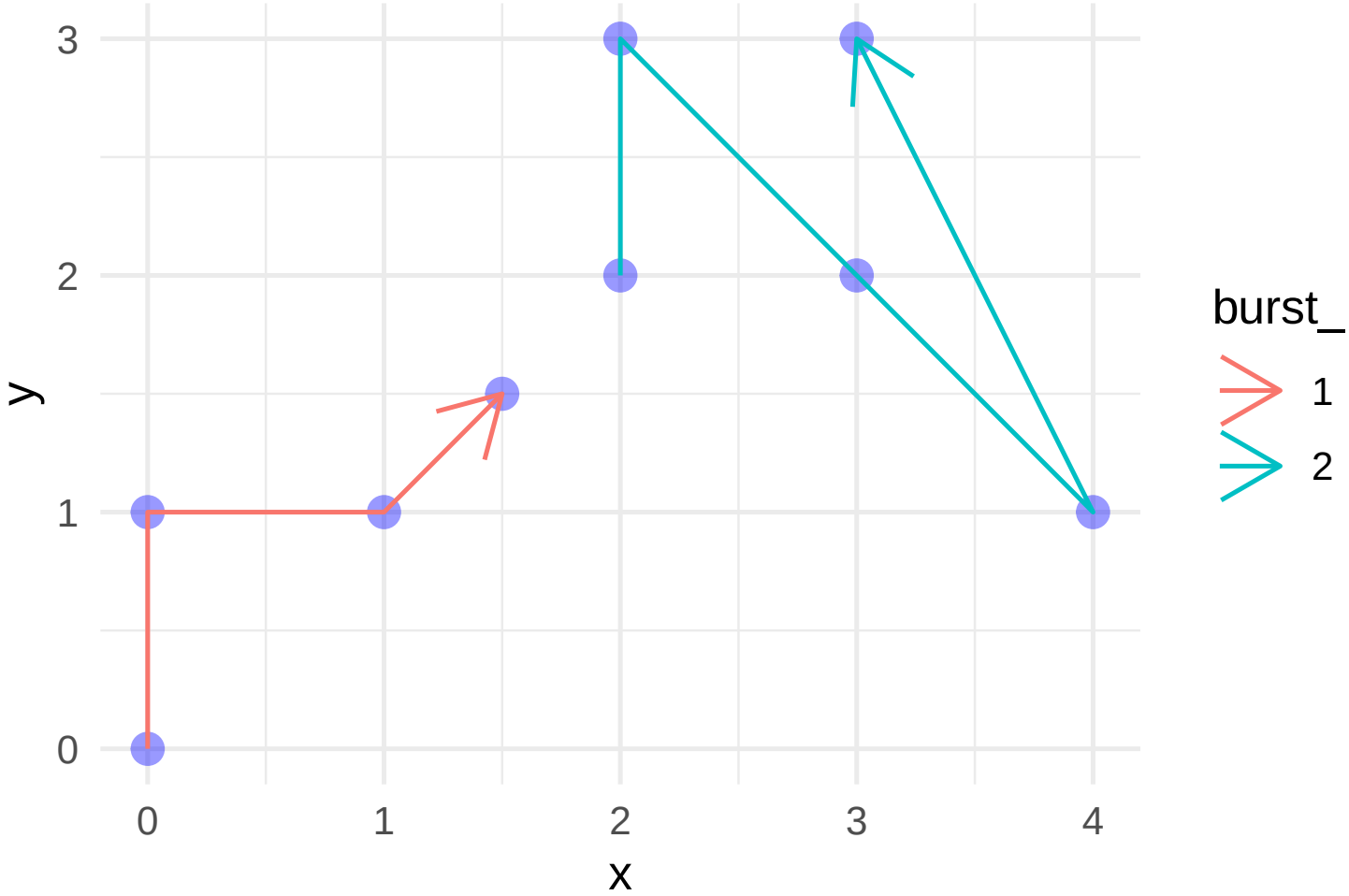
We can start to create a `steps`-representation.



This can be achieved with the function `amt::steps()`. If we resampled the data previously, we can even use `amt::steps_by_burst()`.


3
2
1
0
y
0
1
2
3
4
x


3
2
1
0
y
0
1
2
3
4
x


3
2
1
0
y
0
1
2
3
4
x
burst_
1
2

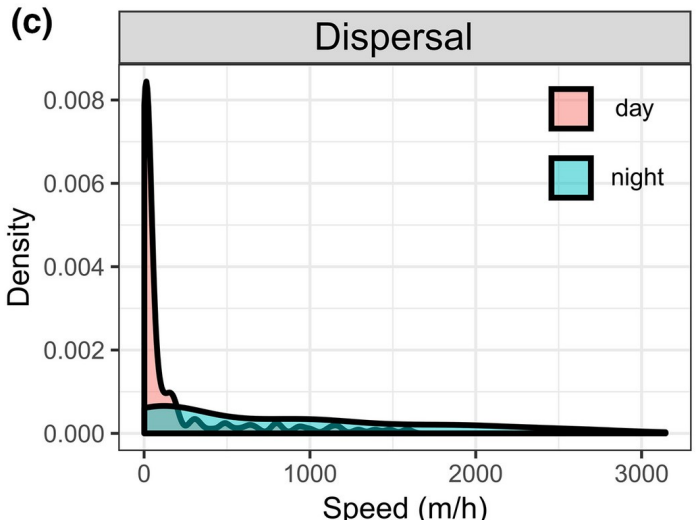
This automatically calculates several step attributes:

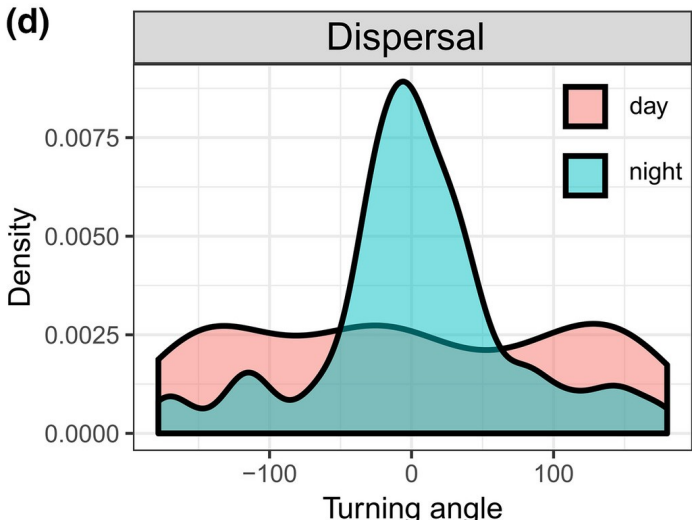
- Start and end point
- Step length
- Absolute and relative turn angles
- Duration

This allows already calculate some step characteristics. It becomes even more informative, if we pair this for example with the whether a step was in the night, day or twilight.

Example

Remington Moll observed a (rare) long distance dispersal for White Tail deer[3] and looked at the turn angle and step distribution for day and night.





[3] Moll et. al Ecology and Evolution; https://onlinelibrary.wiley.com/doi/full/10.1002/ece3.7354.

Data sets that we use

- We will use several data sets during this course including a data set on fishers, elephants and deer.
- Feel free to use your own data during the exercises, we are happy to help to get it into shape.
- For the R walkthrough we often simulate data. We believe if understand how data is generated it is much easier to understand how a specific method works.