ANIMOVE

```
         P1.x=diag(c(0, 0.001, 0.001
         P1.y=diag(c(0, 0.001, 0.001))

splayPar(mov.model=~1, err.model=list(x=~errX, y=~errY), drift.mod
            data=nfsNew, fixPar=c(NA, 1, NA, 1, NA, NA, NA, NA))

t <- crwMLE(mov.model=~1, err.model=list(x=~errX, y=~errY), drift.mod
               data=nfsNew, coord=c("longitude", "latitude"), polar.coord
               Time.name="Time", initial.state=initial.drift,
               fixPar=c(NA, 1, NA, 1, NA, NA, NA, NA),
               control=list(maxit=2000,trace=1, REPORT=10),
```

MAX PLANCK INSTITUTE OF ANIMAL BEHAVIOR

CASUS CENTER FOR ADVANCED SYSTEMS UNDERSTANDING

DLR

INRAE

SENCKENBERG world of biodiversity

Smithsonian

UNI WÜ

June 2024

Trajectory Centered Analysis

Net Squared Displacement, First-Passage Time, Recurse, Variance Components of Movement

AniMove

Trajectory centered analysis

Information extracted from the geometry of tracks, without context.

To keep in mind before we start:

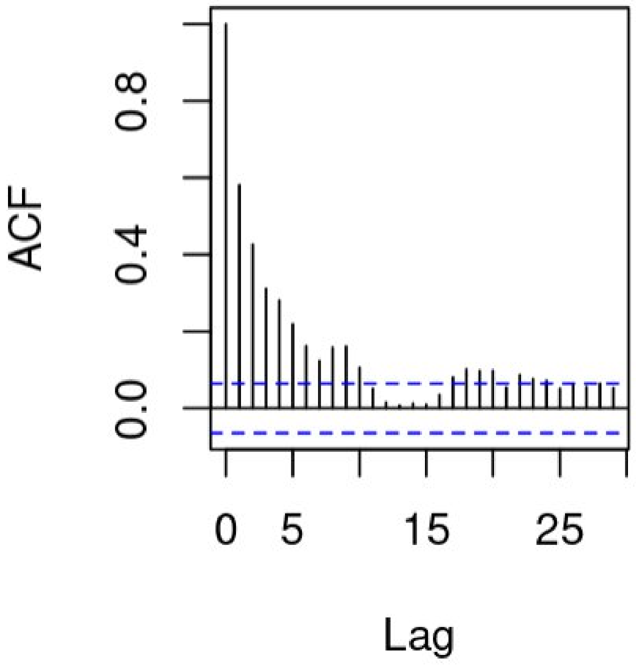
- Some methods are **scale sensitive and sensitive to sampling schedule.**
- For some methods data should be in an **equidistant projection** (distances are preserved).

Auto-correlation

Auto-correlation is the **degree of correlation of a variable with itself over time**. In almost all natural movements we can find some **degree of persistence in maintaining the direction and often also speed (correlation in geometric properties)**.

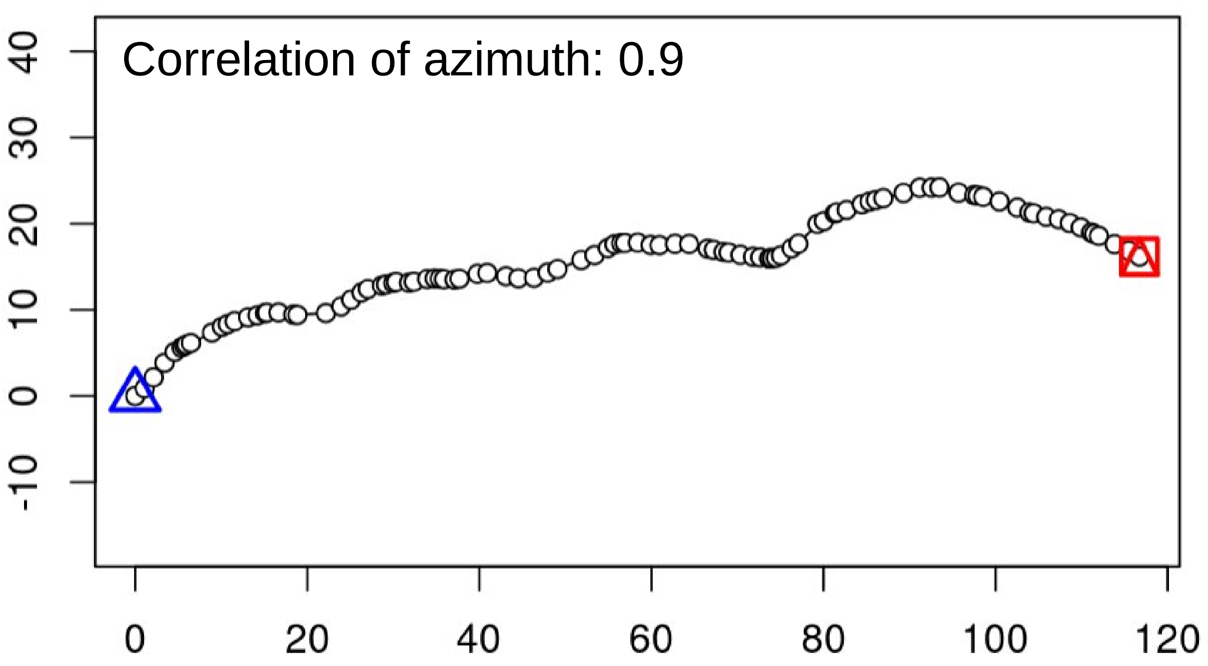
Auto-correlation is calculated and quantified based on correlation of values of a parameter (e.g. speed), dependent on the distance in the sequence of appearance (lag), between any two observations. Easy to explore in R using the function `acf(var)`.

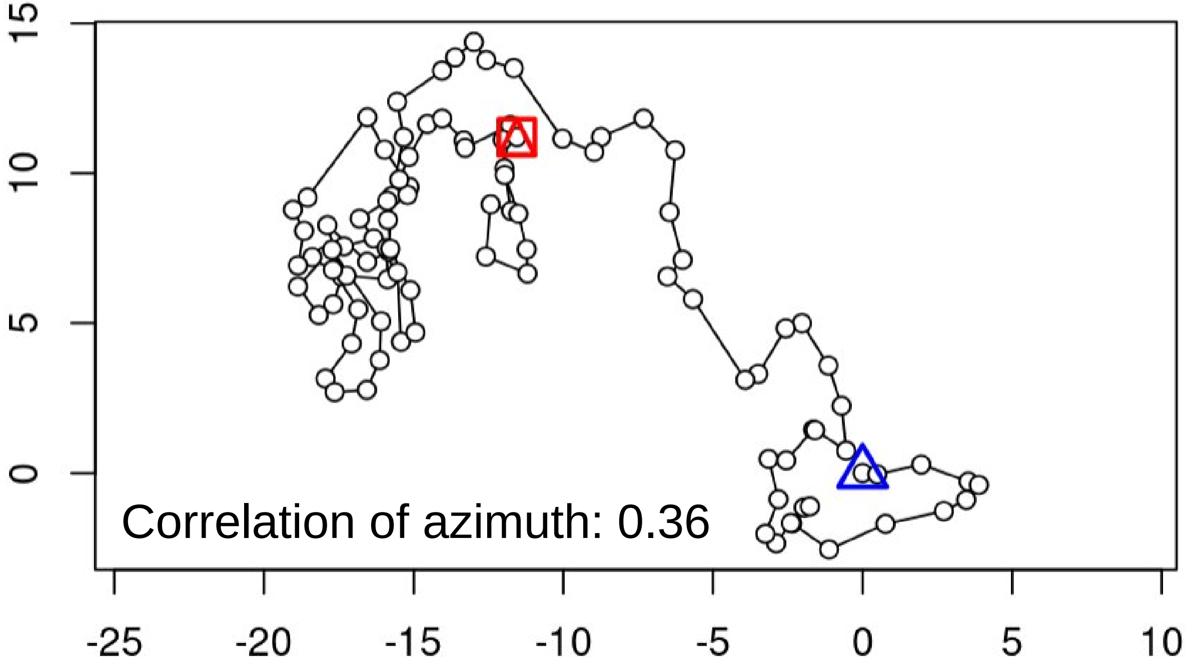
Auto-correlation of speed in a fisher



Auto-correlation and movement process

The correlation structure can **inform about the underlying movement processes and changes in these processes** (directional vs foraging movement, periodicity, etc..).





Quantify movement process based on track geometry

Net square displacement (NSD):

- Quantifies the net squared distance traveled over time compared to an point of reference (calculates distance from each location to a point of reference)

First passage time (FPT):

- Assumes *regular sampling and equidistant projection.*
- Calculates the time it takes to cross a circle of a given radius, i.e. time the animal spends in a given area at a certain spatial scale (”draws” circles around each point and calculates how long it takes to leave the circle)

Recursions:

- Assumes *regular sampling and equidistant projection.*
- Identifies frequently-used locations across one or multiple individuals (by pre-specify locations of interest or by examining all trajectory locations)

Variance components of movement:

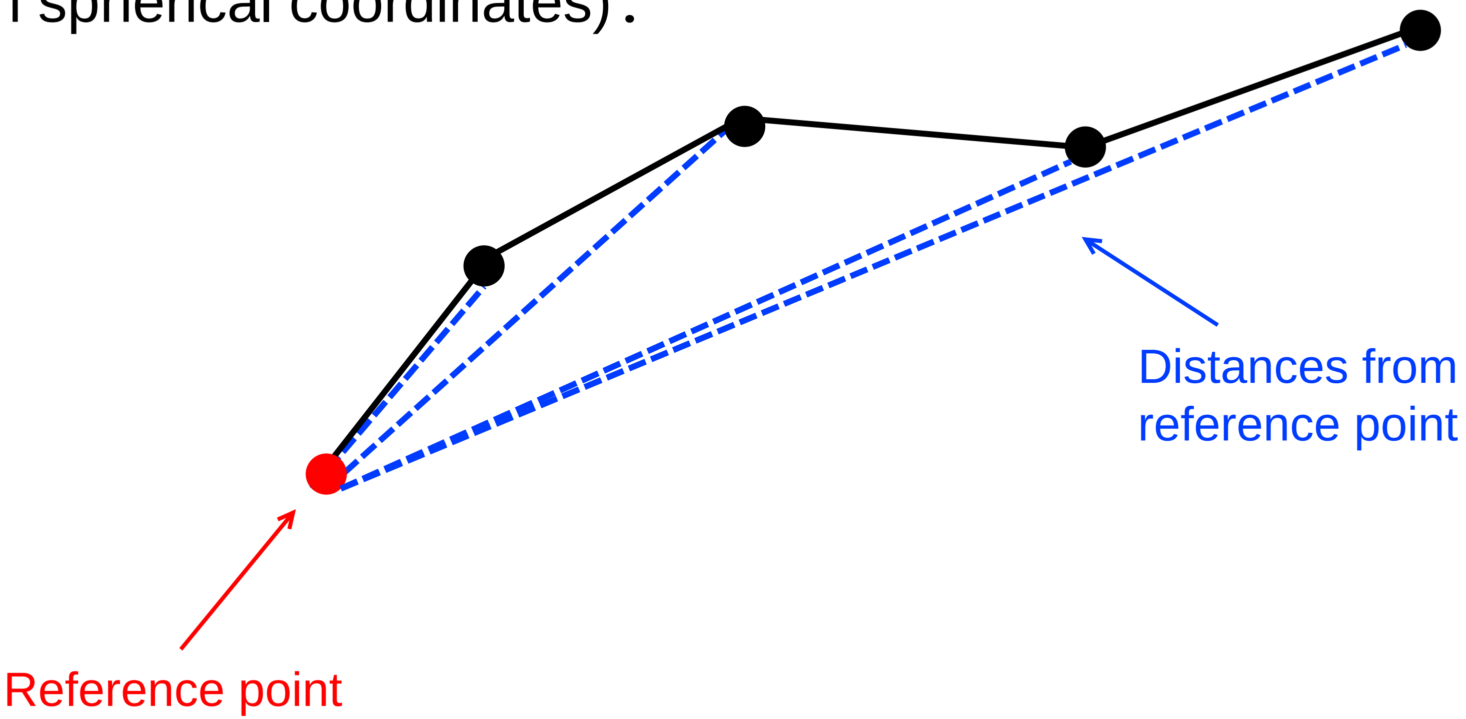
- Data can be irregular/gapy
- Estimate the probability of where the animal could have been when we did not observe it (between two known locations).

Net-squared displacement (NSD)

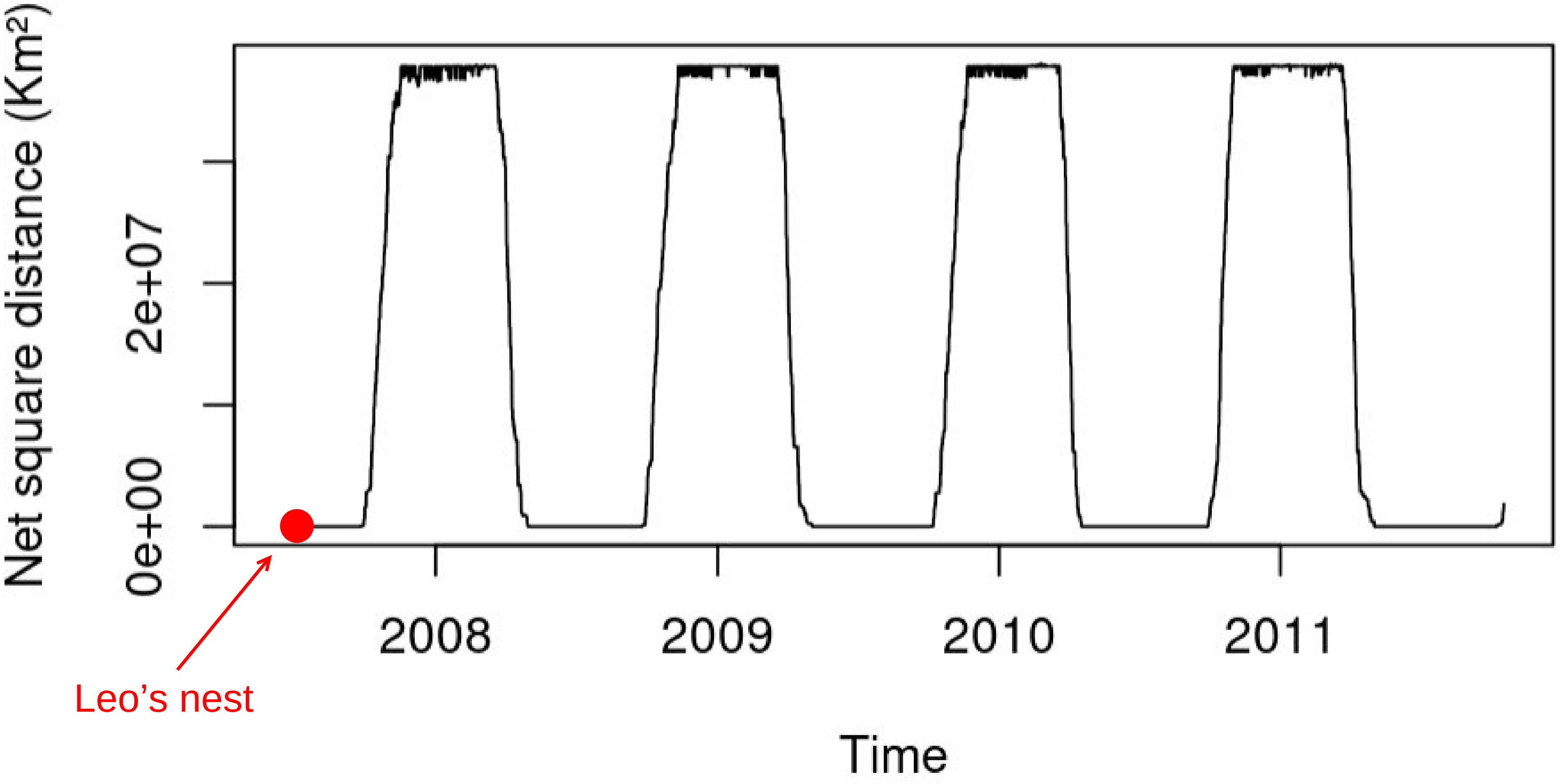
NSD quantifies the net squared distance traveled over time compared to a point of reference (calculates **distance from each location to the point of reference**).

A point of reference (nest, den, colony) is needed. If there is no point of reference, or animal is nomadic, use e.g. FPT (see later).

Implementation in any package calculating distances (e.g. `sf::st_distance`, which can also deal with spherical coordinates).

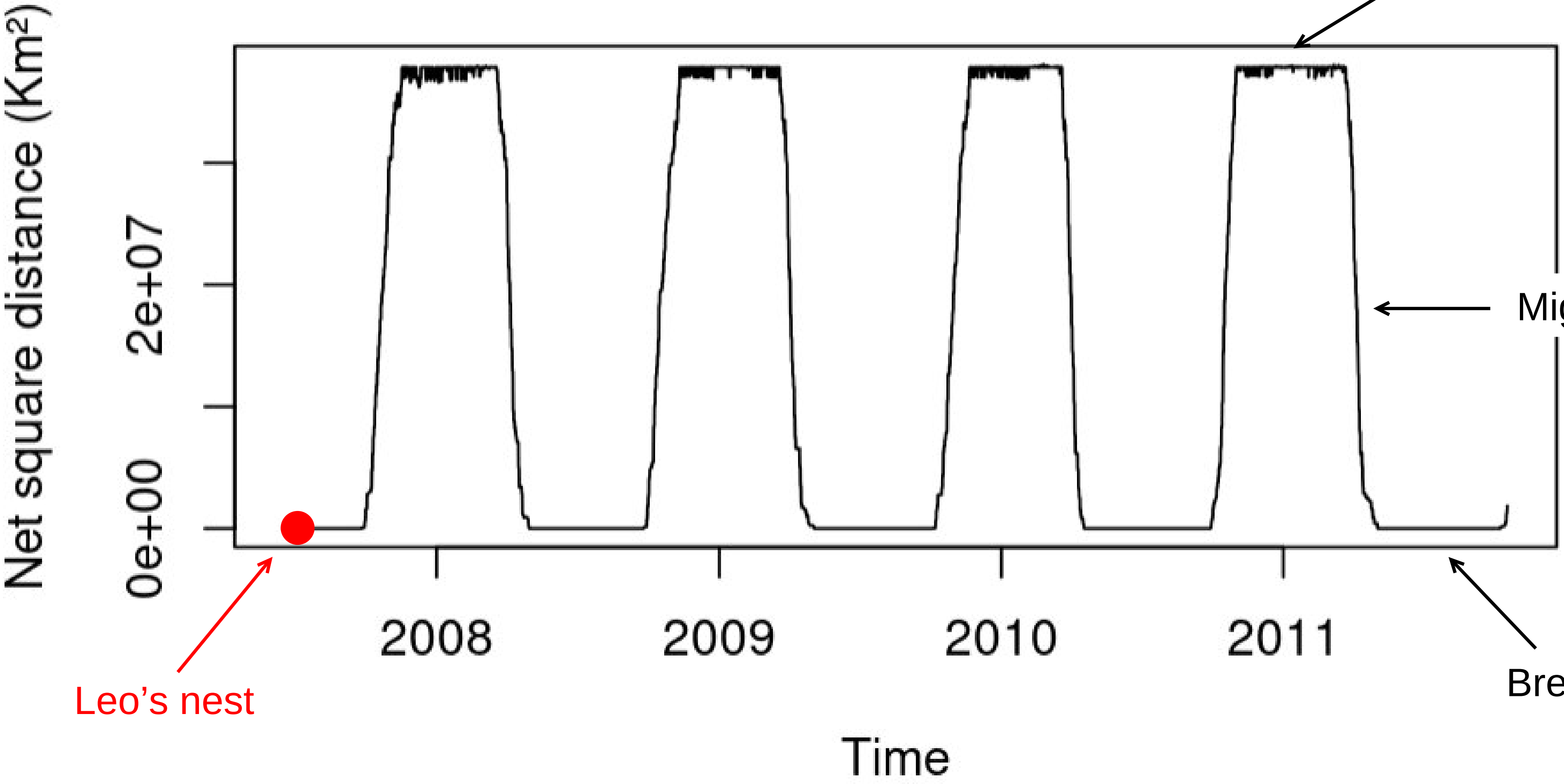


Net-squared displacement (NSD)



Leo’s nest

Net-squared displacement (NSD)



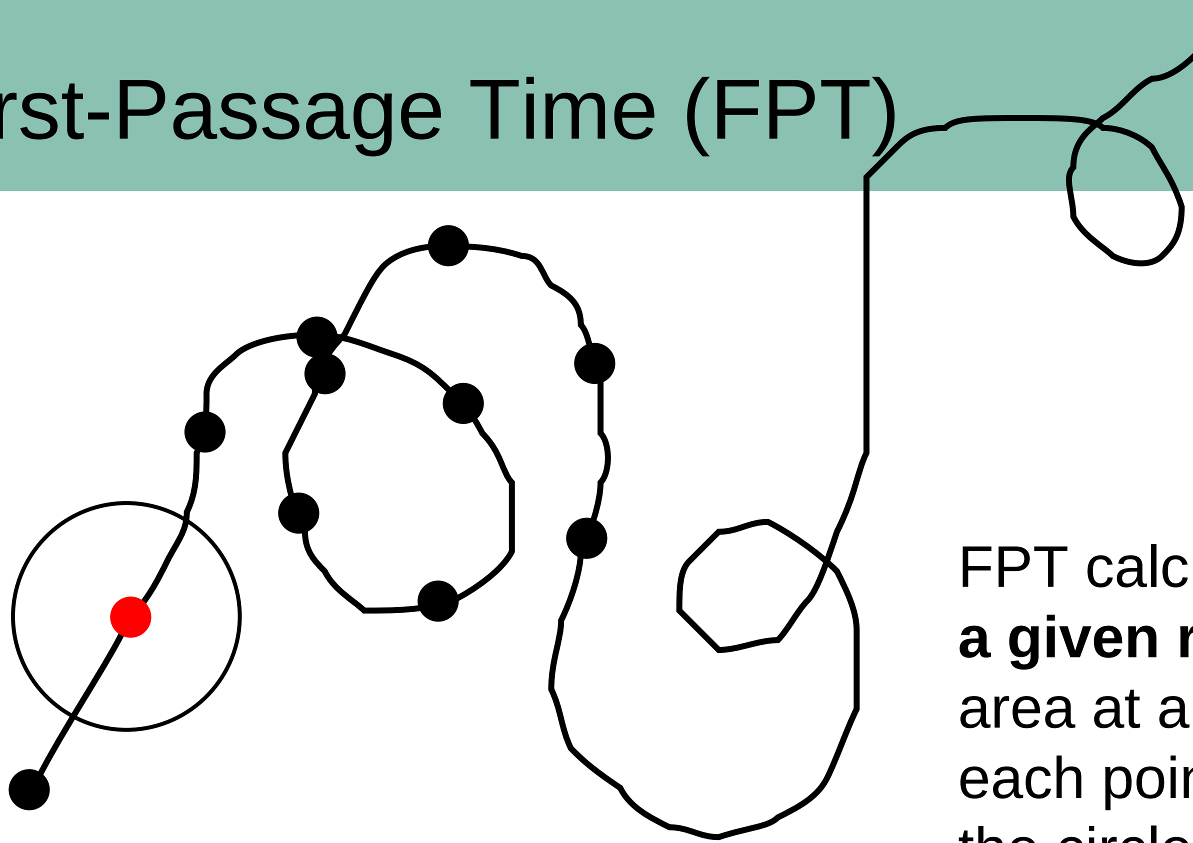
Wintering

Migration

Breeding

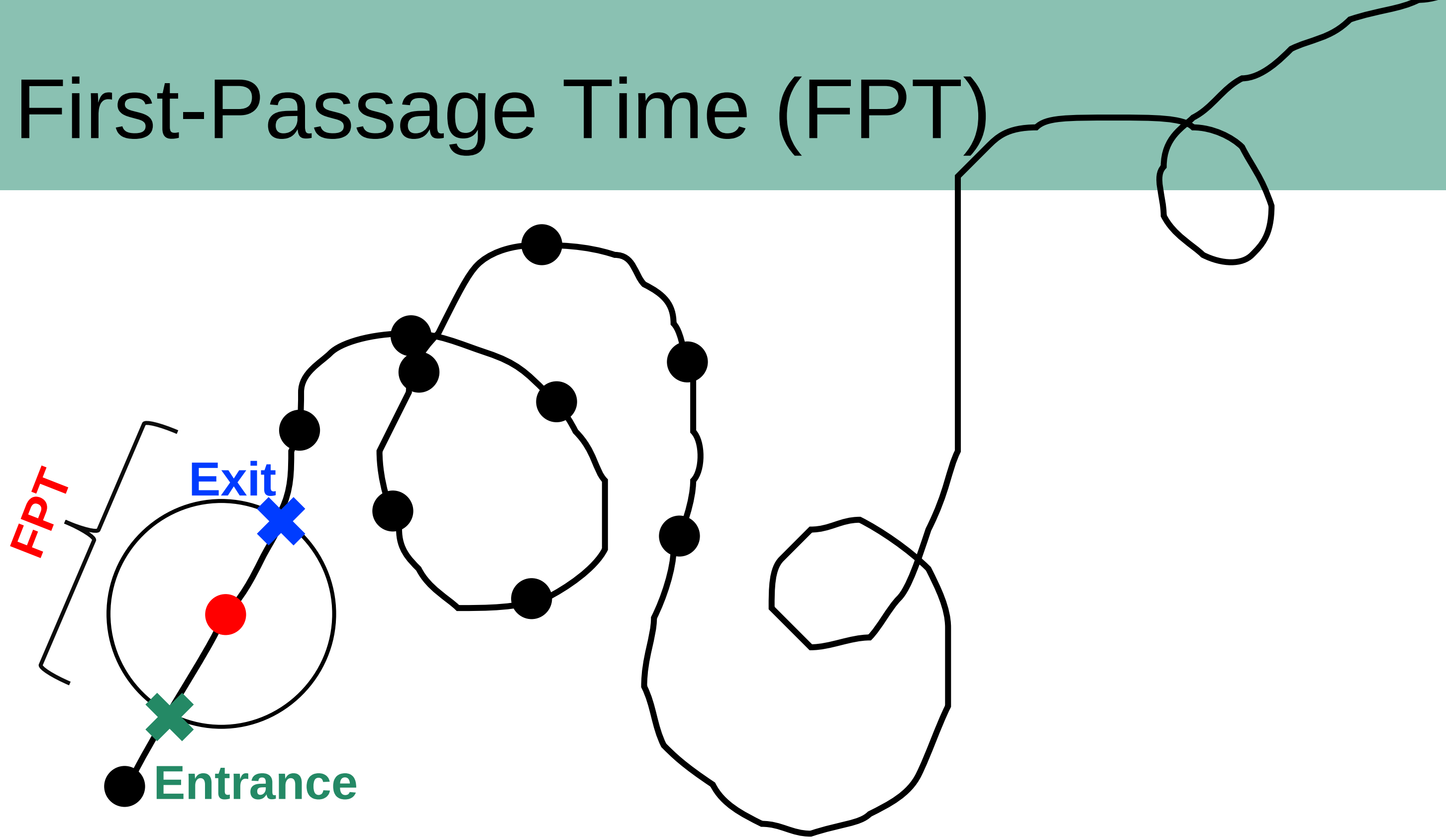
Leo’s nest

First-Passage Time (FPT)

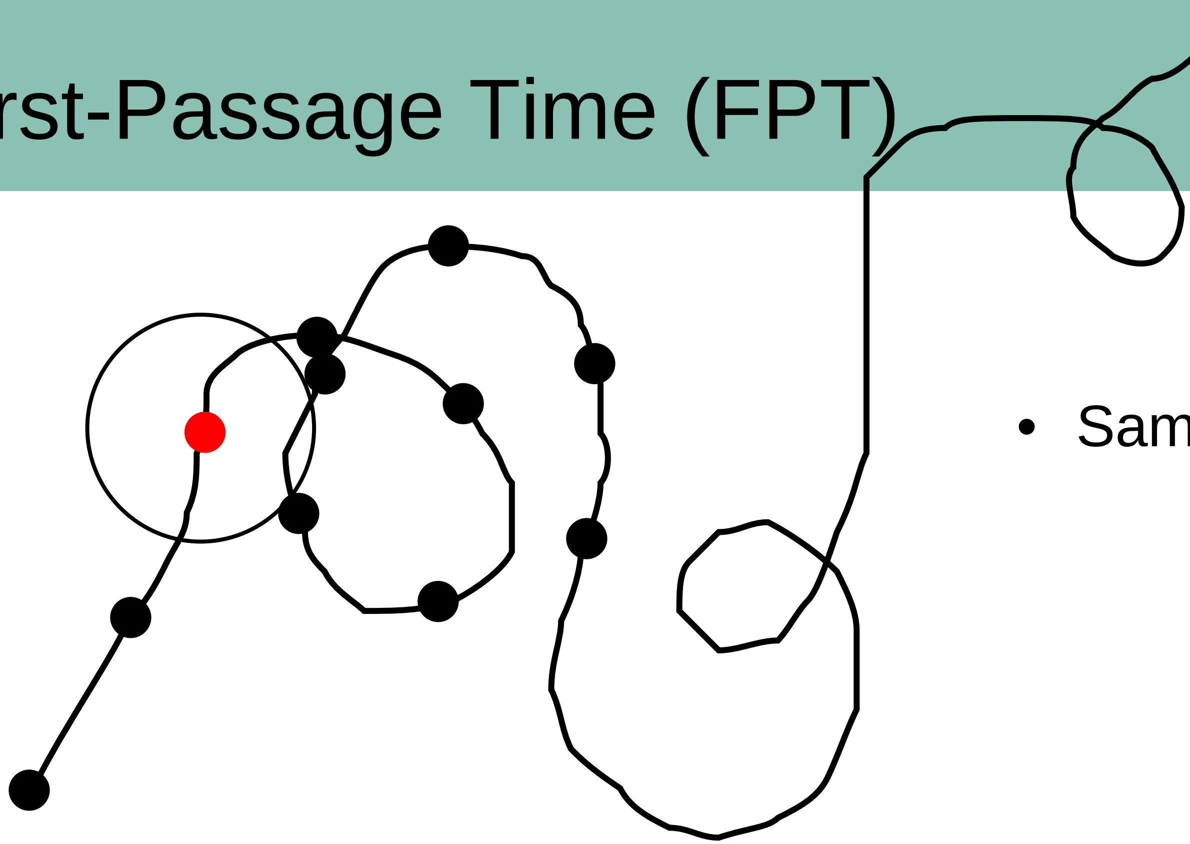


FPT calculates the **time it takes to cross a circle of a given radius**, i.e. time the animal spends in a given area at a certain spatial scale (”draws” circle around each point and calculates how long it takes to leave the circle).
Implementation in R package `adehabitatLT`.
Requires regular sampling and equidistant projection.

First-Passage Time (FPT)

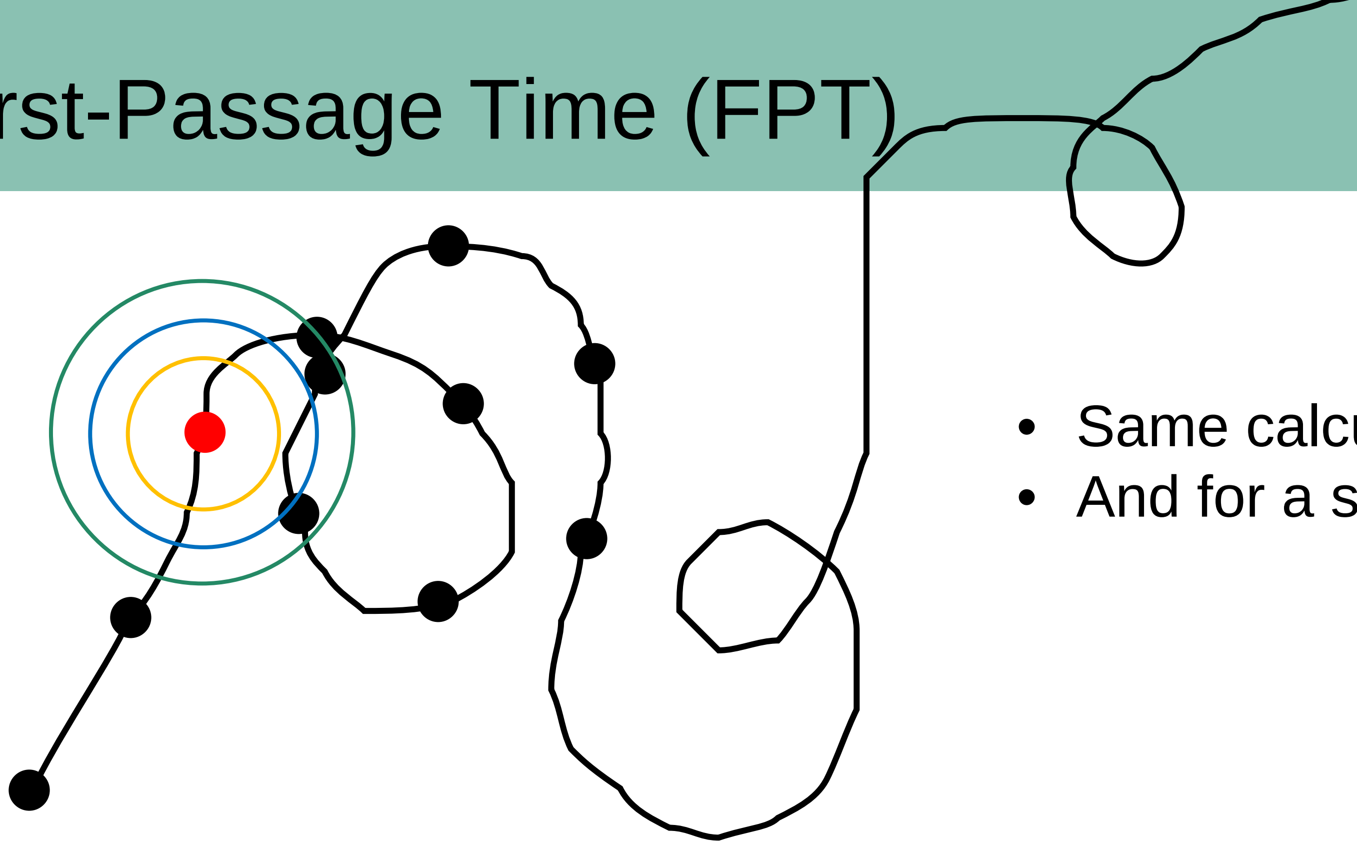


First-Passage Time (FPT)



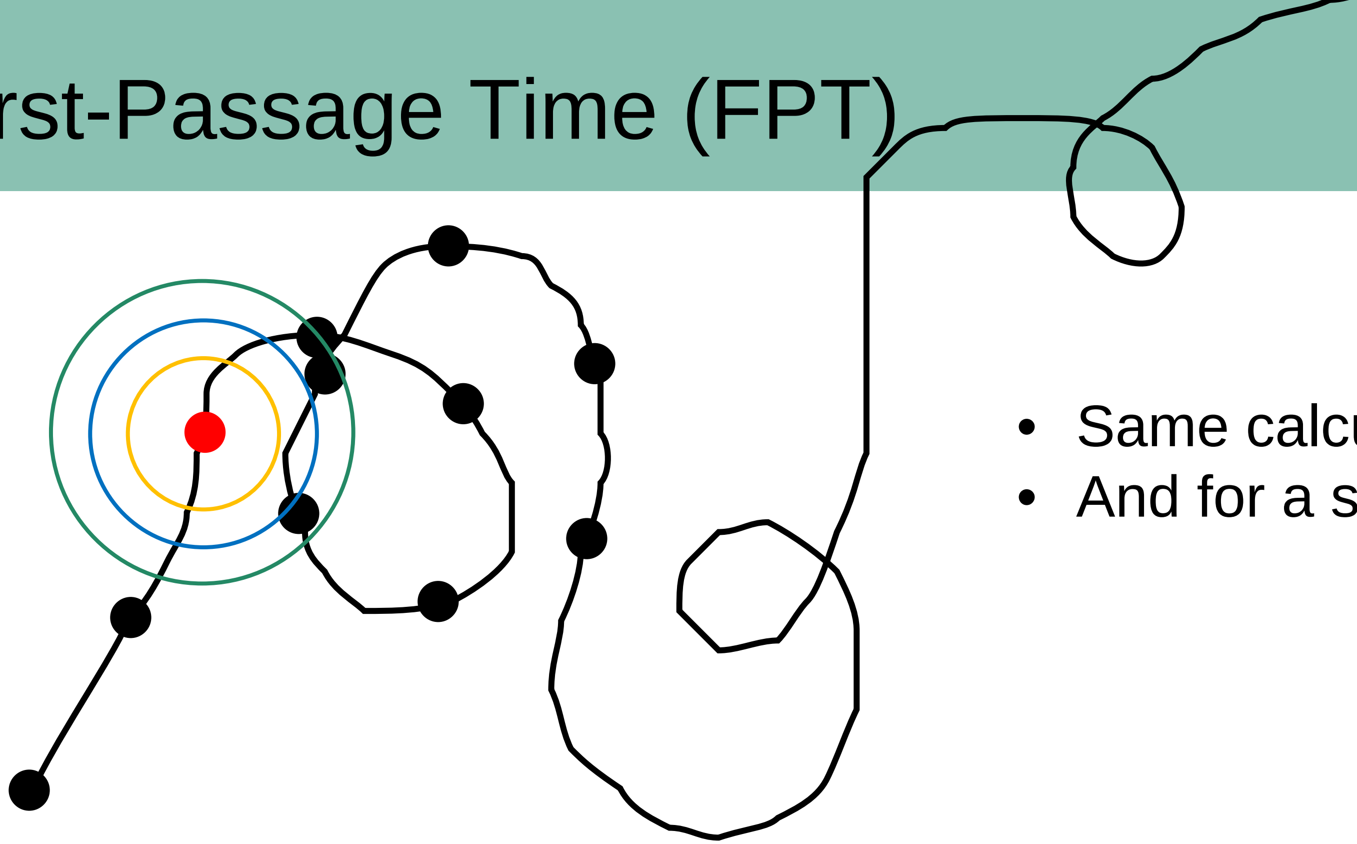
- Same calculation for each point in the trajectory

First-Passage Time (FPT)



- Same calculation for each point in the trajectory
- And for a sequence of different radii

First-Passage Time (FPT)

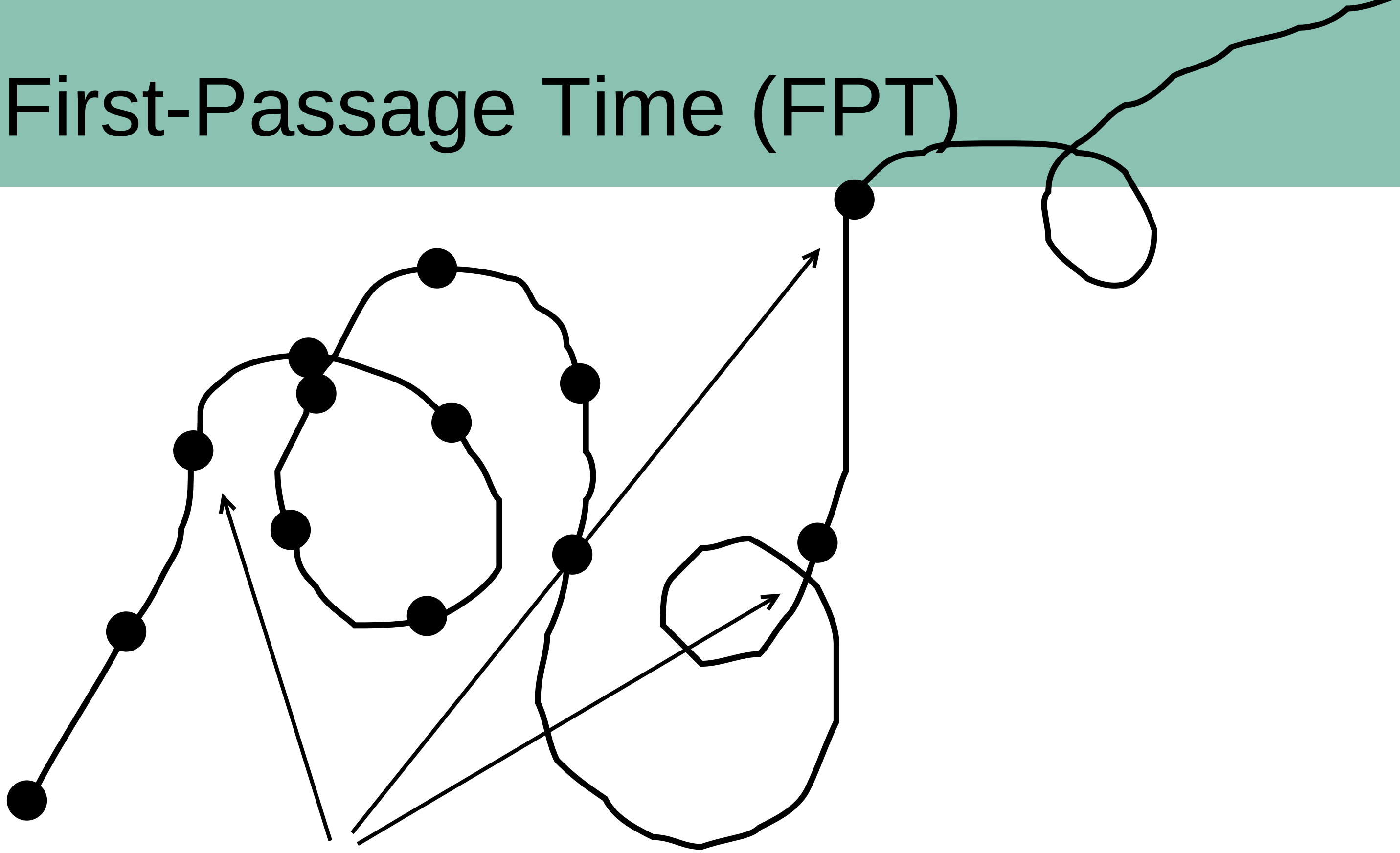


- Same calculation for each point in the trajectory
- And for a sequence of different radii

Will give us an understanding of the **movement processes that happen at different scales:**

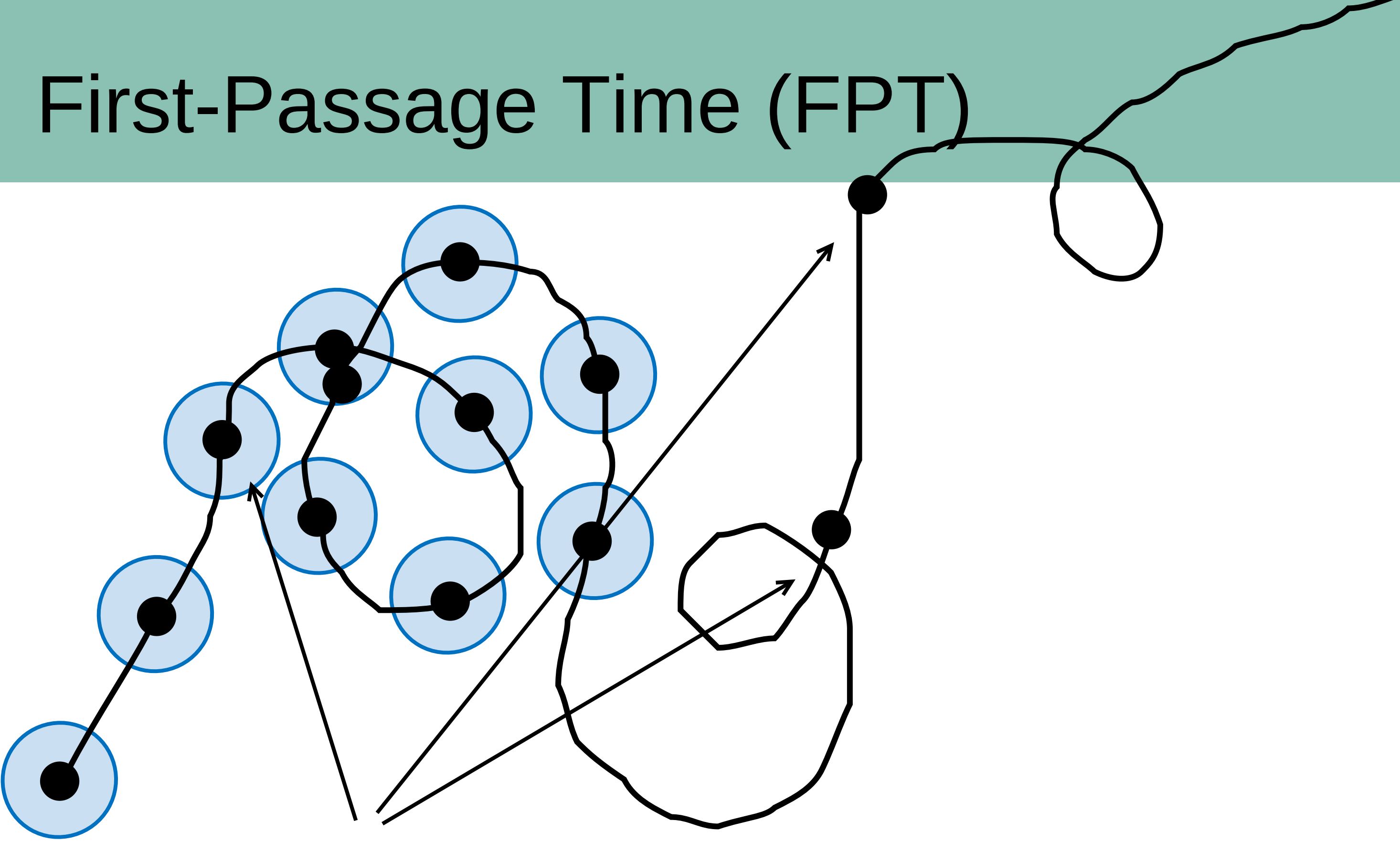
- the **variance of log(fpt) ~ log(radii)** can inform about the scales at which processes are likely to be changing
- the **slopes of the log(meanFPT) ~ log(radii)** can indicate the type of movement at each scale. Flatter slopes indicate more directional movement, steeper slopes more brownian movement.

First-Passage Time (FPT)



Likely switches of
movement process

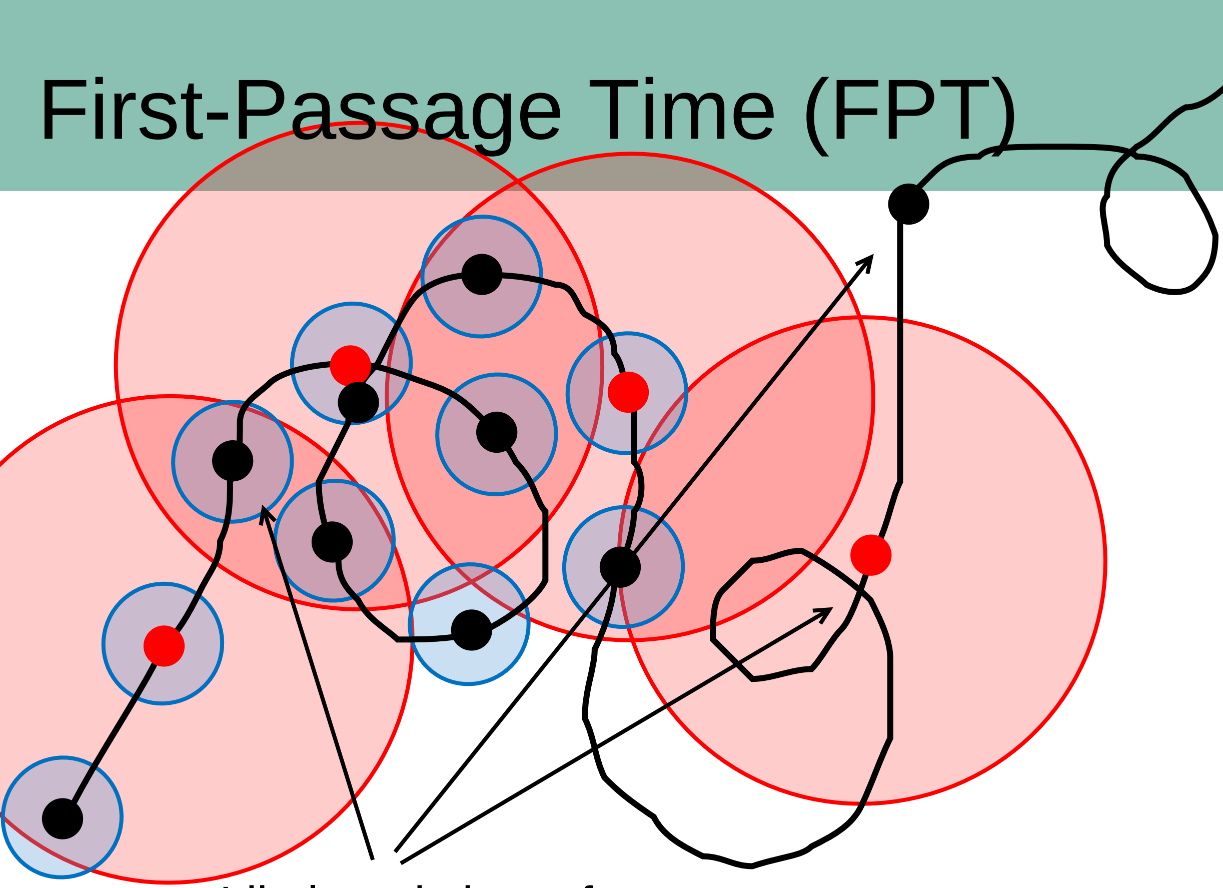
First-Passage Time (FPT)



Likely switches of movement process

○ Radii of low variance
(consistent movement process)

First-Passage Time (FPT)

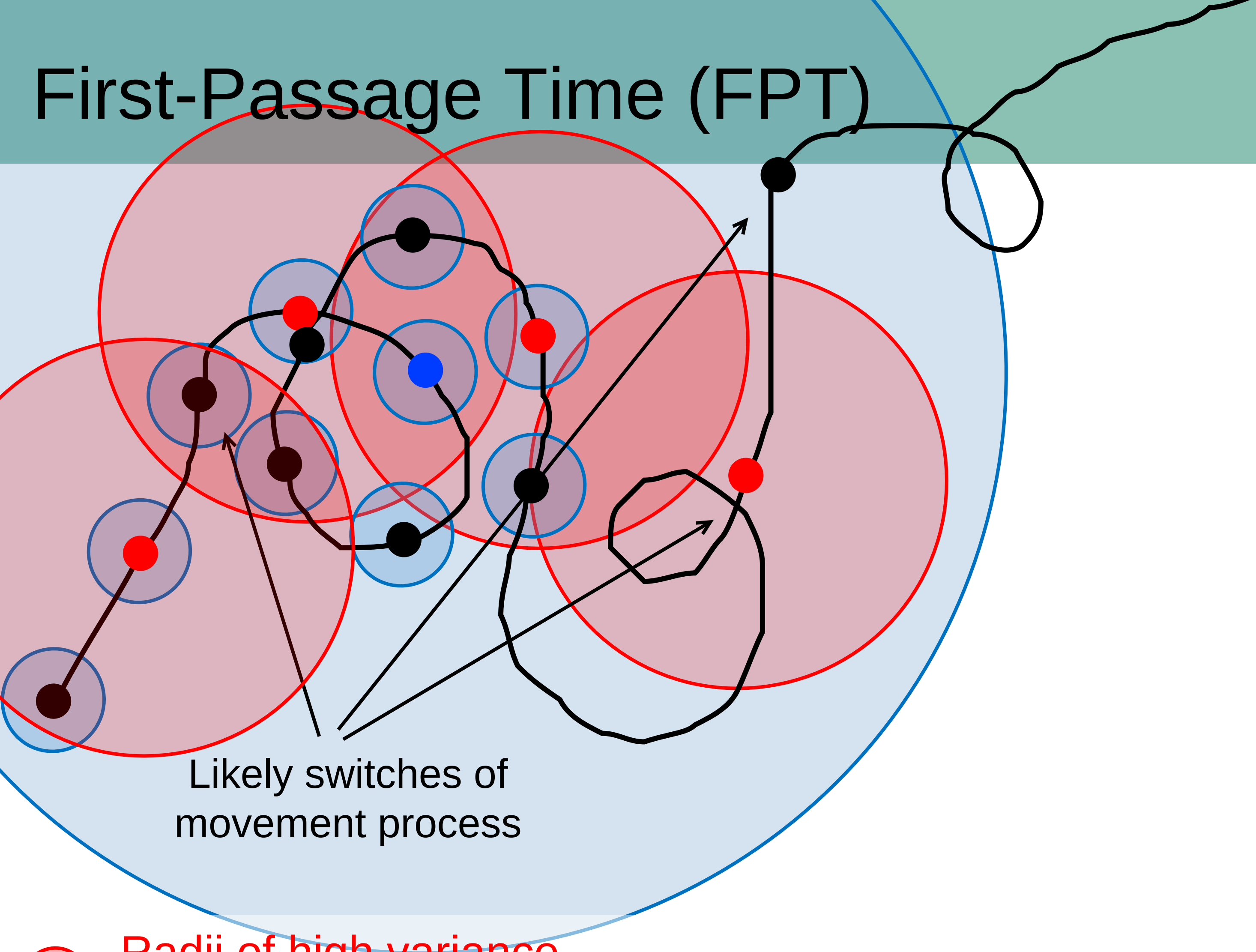


Likely switches of movement process

🔴 Radii of high variance
at the switch between behaviours

🔵 Radii of low variance
(consistent movement process)

First-Passage Time (FPT)

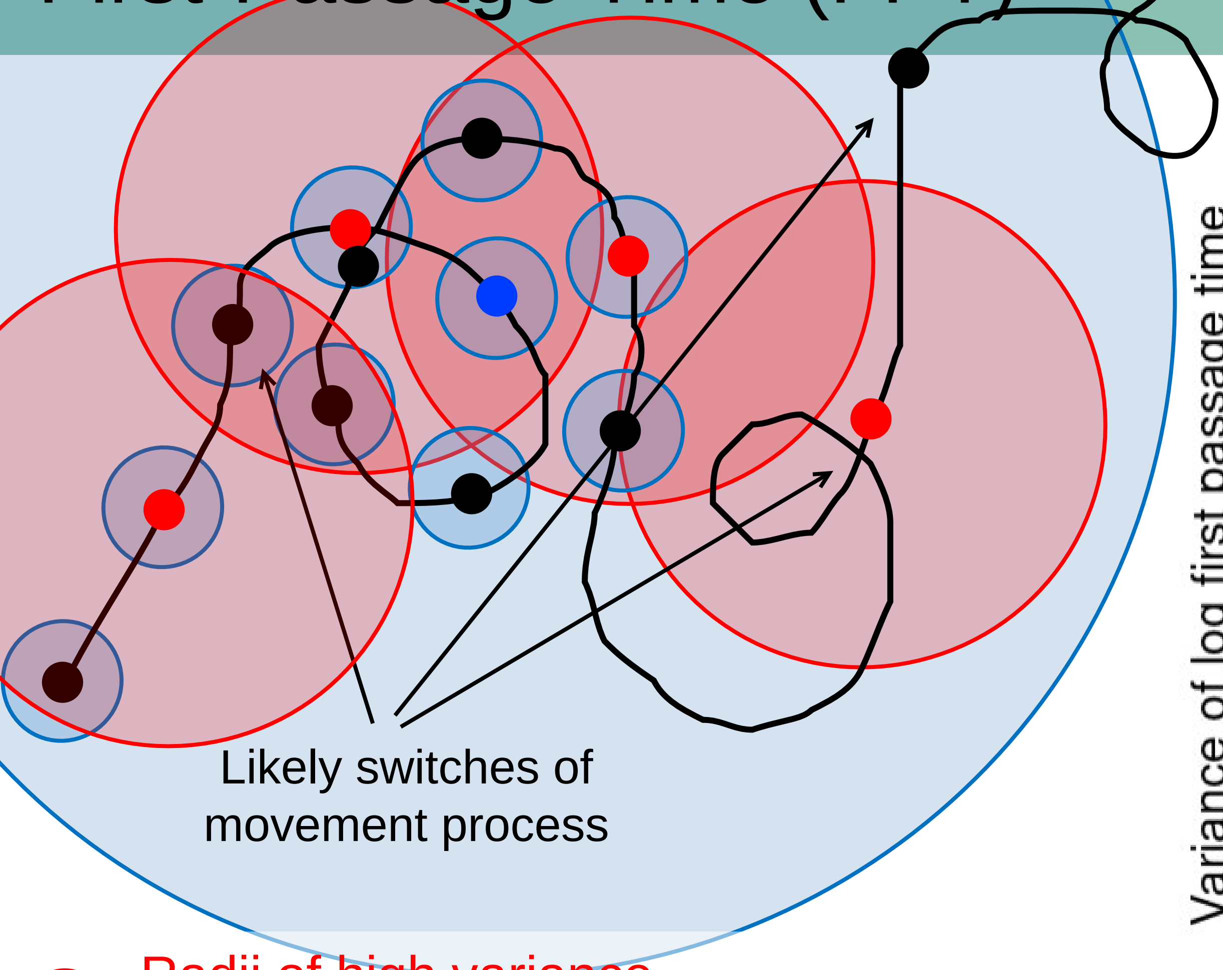


Likely switches of movement process

🔴 Radii of high variance
at the switch between behaviours

🔵 Radii of low variance
(consistent movement process)

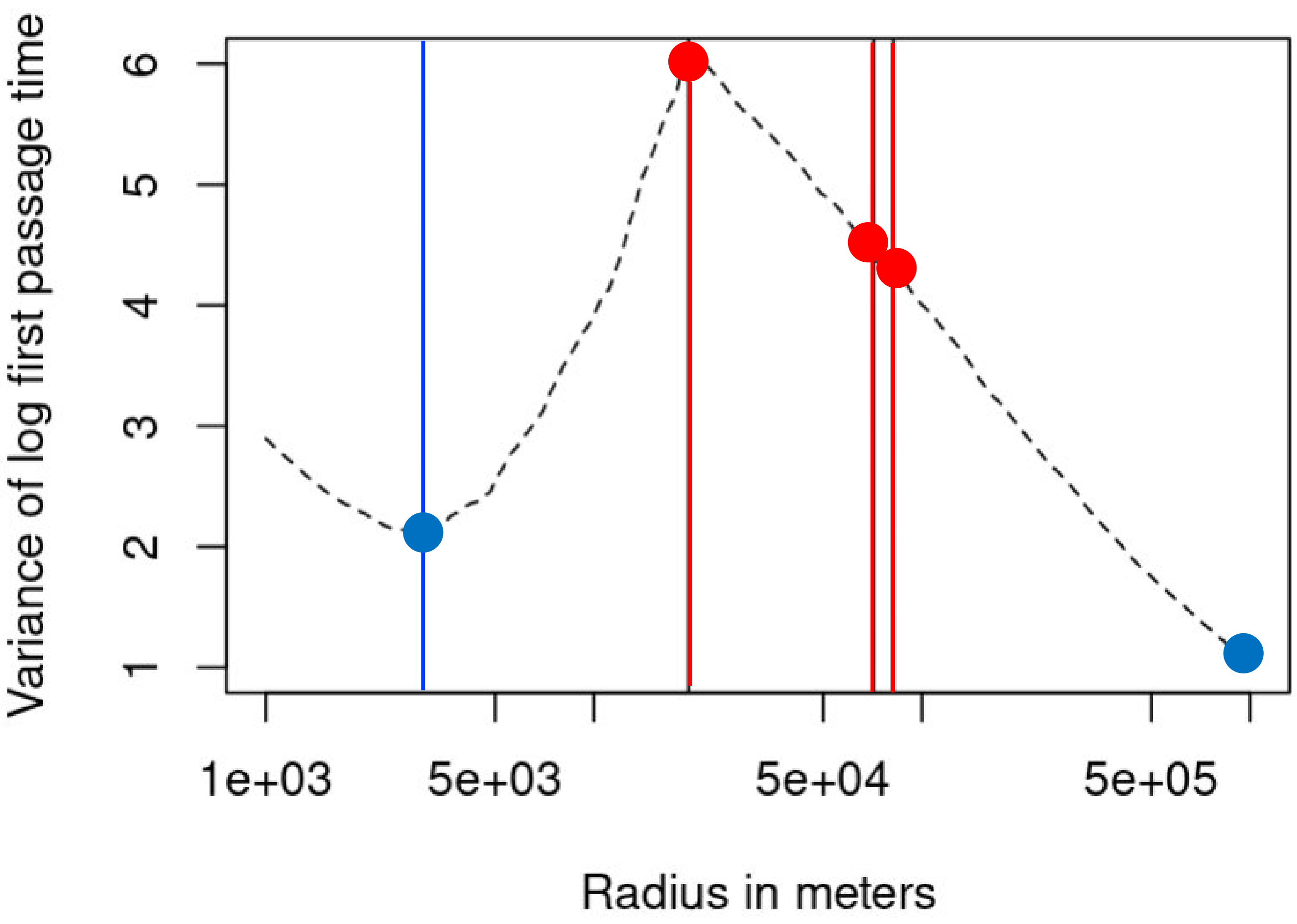
First-Passage Time (FPT)



Likely switches of movement process

● Radii of high variance at the switch between behaviours

● Radii of low variance (consistent movement process)



Variance of log(FPT) ~ log(radii)

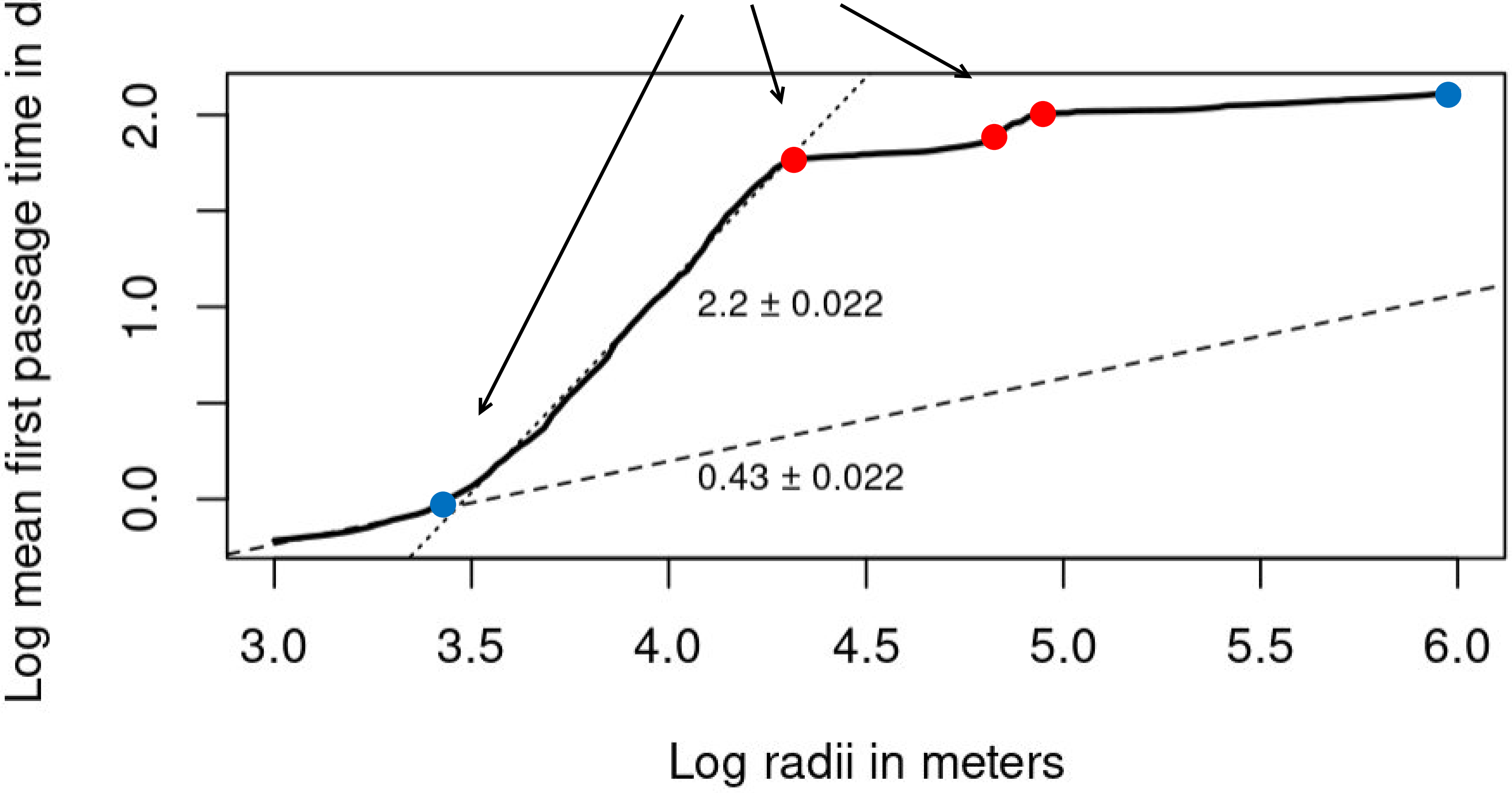
First-Passage Time (FPT)

- **flat slope** (< 2): **directional** movement (larger radius = small increase in time to cross it)

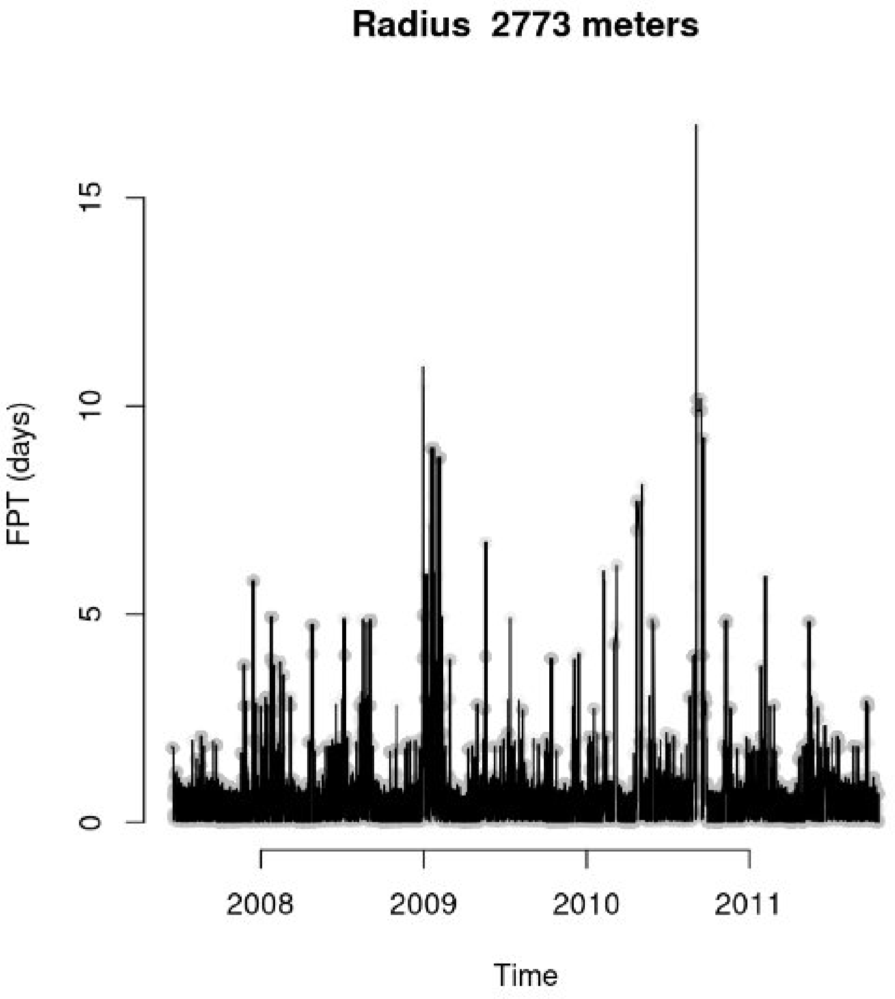
- **steep slope** (around 2): **brownian** movement (increase in radius = linear increase in time to cross it)

log(mean FPT) ~ log(radii)

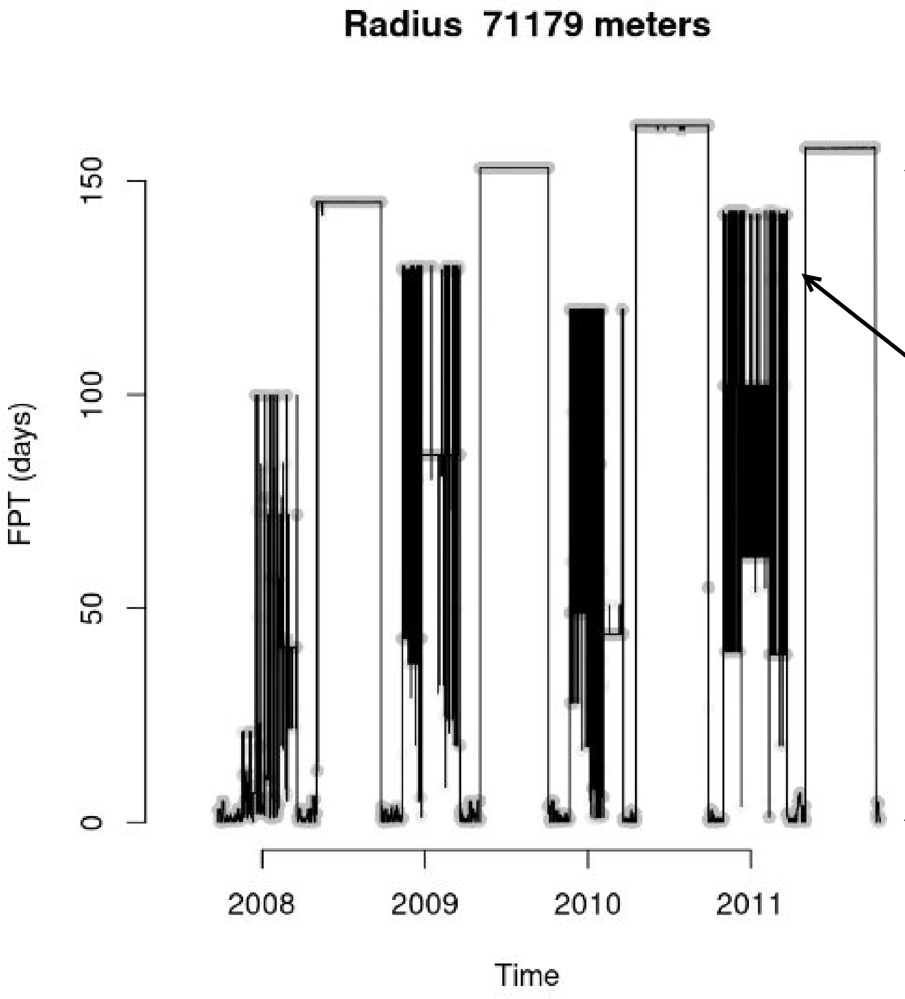
Min and Max in **variance of FPT**



First-Passage Time (FPT)



Radius ~3 km = probably daily ranges
(below 3 km Leo moves very directly)



Breeding

Wintering
(more movement)

Migration

Radius ~80 km = migration

Hands on

How to do this in R in the script:
“6A_TrajectoryAnalysis_NSD_FPT.R”

Recursions

Why do animals return?

Wikipedia

Alain Houle

Chicago Zoologiacl Society

Recursions

Applications of recursion analysis:

- Identify frequently-used locations across one or multiple individuals (by pre-specify locations of interest or by examining all trajectory locations)
- Visit-level analysis of **time of day, duration, time since last visit**, etc. and combine with other information (e.g, behavioral segmentation)
- **Intervisit** interval consistency at and across locations
- Visits to **user-specified polygon** (protected area, foraging ground)
- Residence time during user-specified intervals (**seasonal**, etc.)

Analysis including environmental context:

- Spatially/temporally coincident visit **covariates** (e.g., NDVI, temperature, snow etc.)

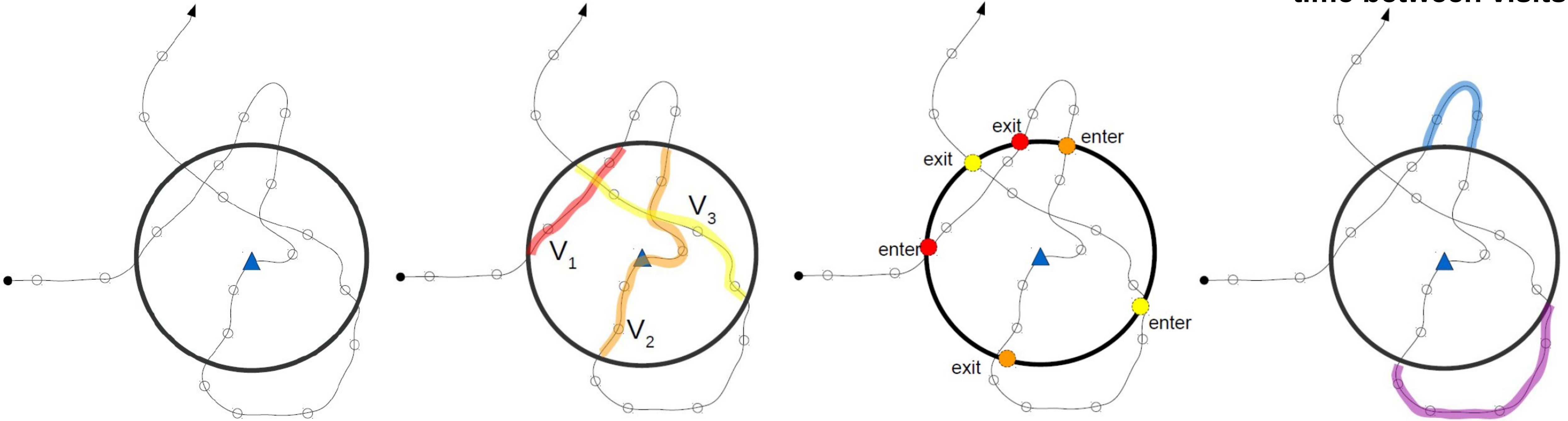
Requires regular sampling.

Implementation in R package `recurse` (CRAN vignette by Chloe Bracis https://cran.r-project.org/web/packages/recurse/vignettes/recurse.html).

Recursions

2. Select **segments inside** radius

4. Calculates **time between visits**



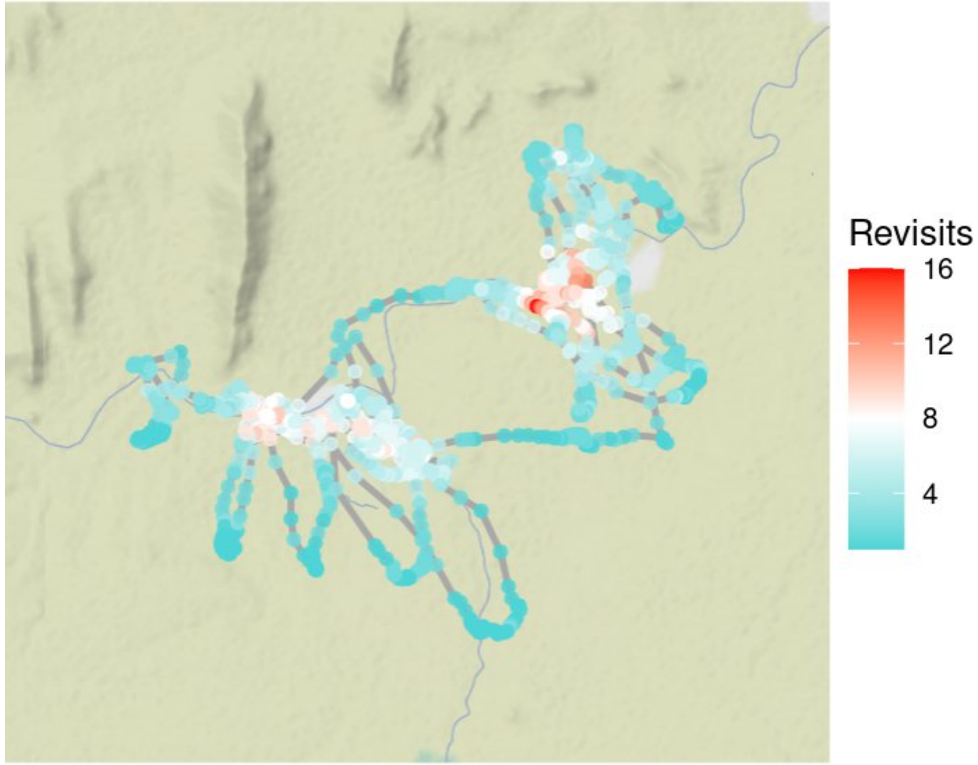
1. Draw a **radius** around each point (or define a **polygon** of interest)

3. Interpolate **entrances/exits**

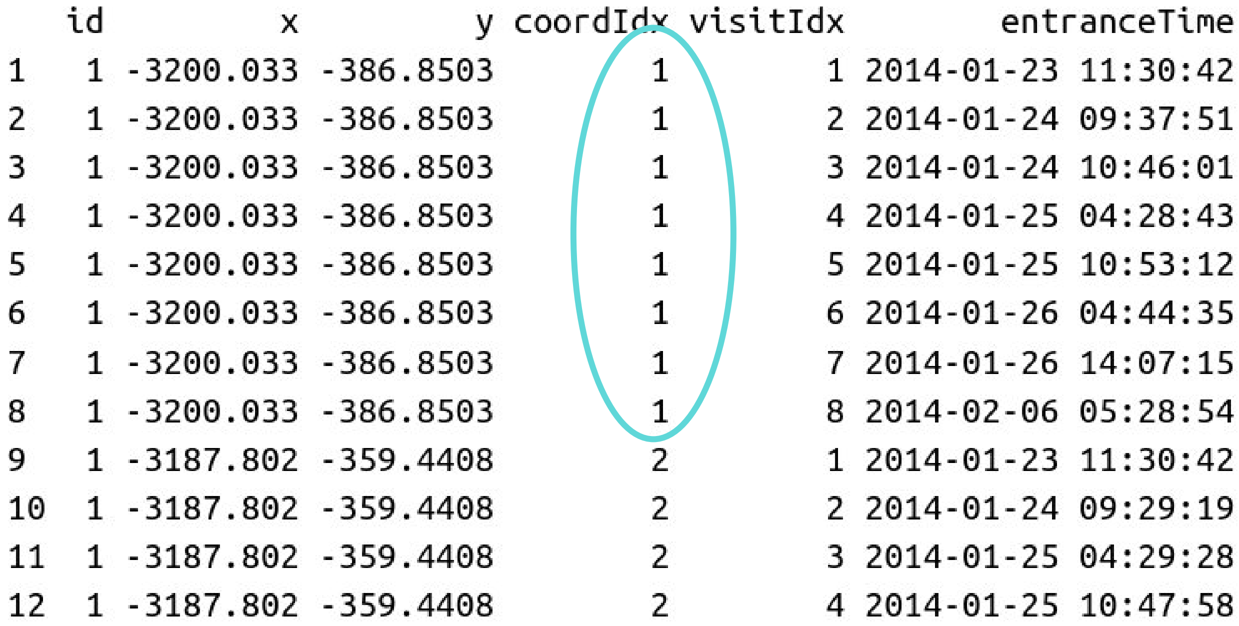
First time crossing a radius =
First Passage Time!

Recursions

Number of revisits on a 500 m radius around each location



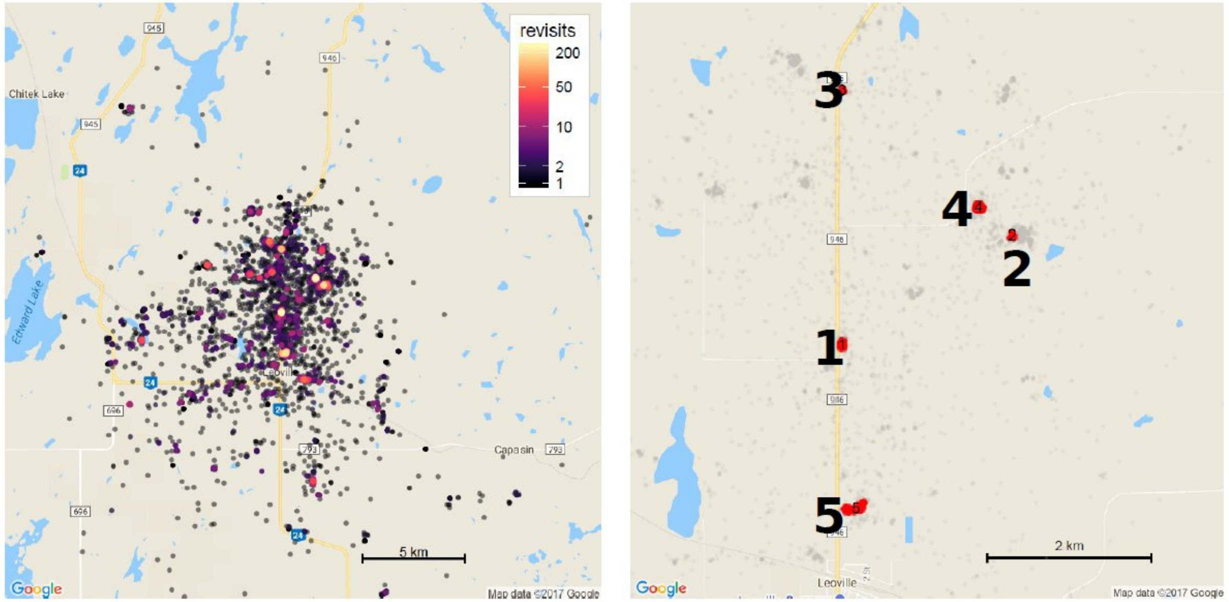
```
> habibaVisits <- getRecursions(Habiba,
                                radius = 500,
                                timeunits = "hours")
> head(habibaVisits, 12)
```



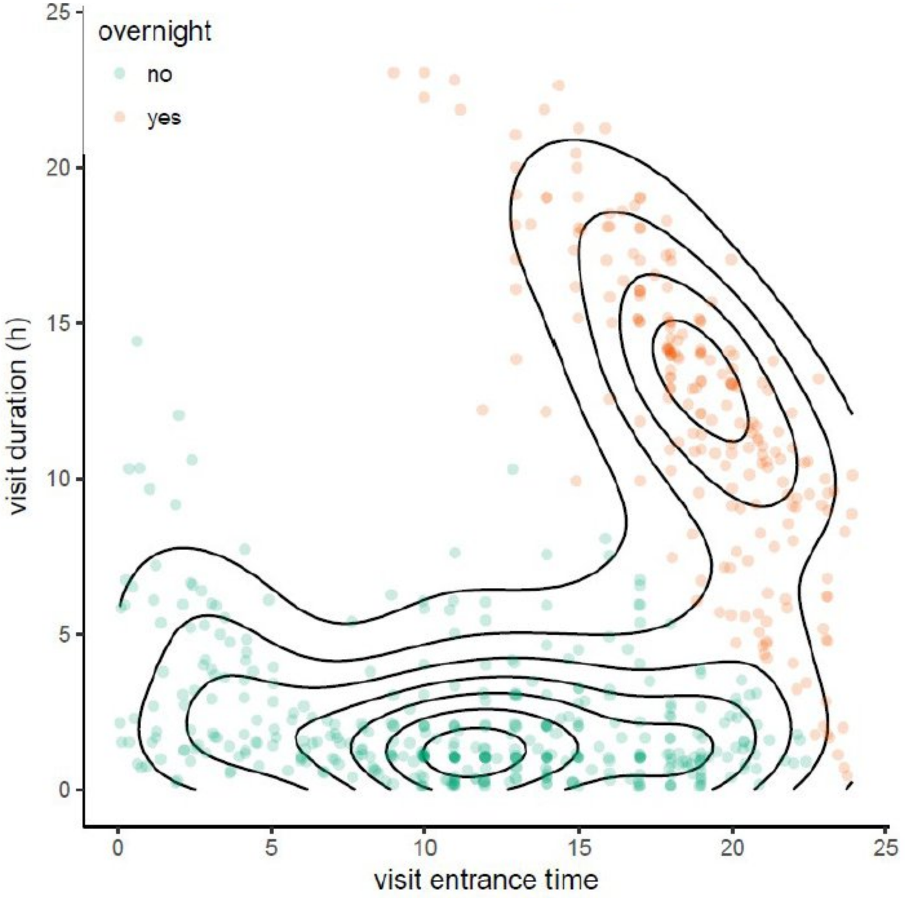
```
   id         x         y coordIdx visitIdx         entranceTime
1   1 -3200.033 -386.8503        1        1 2014-01-23 11:30:42
2   1 -3200.033 -386.8503        1        2 2014-01-24 09:37:51
3   1 -3200.033 -386.8503        1        3 2014-01-24 10:46:01
4   1 -3200.033 -386.8503        1        4 2014-01-25 04:28:43
5   1 -3200.033 -386.8503        1        5 2014-01-25 10:53:12
6   1 -3200.033 -386.8503        1        6 2014-01-26 04:44:35
7   1 -3200.033 -386.8503        1        7 2014-01-26 14:07:15
8   1 -3200.033 -386.8503        1        8 2014-02-06 05:28:54
9   1 -3187.802 -359.4408        2        1 2014-01-23 11:30:42
10  1 -3187.802 -359.4408        2        2 2014-01-24 09:29:19
11  1 -3187.802 -359.4408        2        3 2014-01-25 04:29:28
12  1 -3187.802 -359.4408        2        4 2014-01-25 10:47:58
```

Recursions

Look at sites with > 75 visits, i.e. nests or roosts: **5 sites**



Longer visits only happen at night and include overnight (roosting sites)



Hands on

How to do this in R in the script:
“6B_TrajectoryAnalysis_Recurse.R”

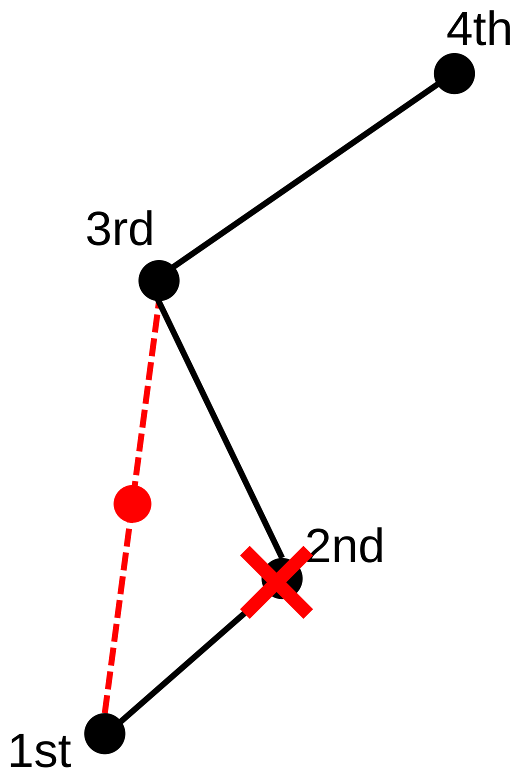
Variance components of movement

Variance estimation is based on a **leave-one-out method** where from 3 consecutive locations the 2nd one is omitted. Based on the timestamp of the 2nd location, a location is estimated on a straight line between location 1 and 3. **The distance between the actual and the estimated 2nd location represents the amount of variance**.

Can deal with irregular/gapy data.
Implementation in R package `move`.

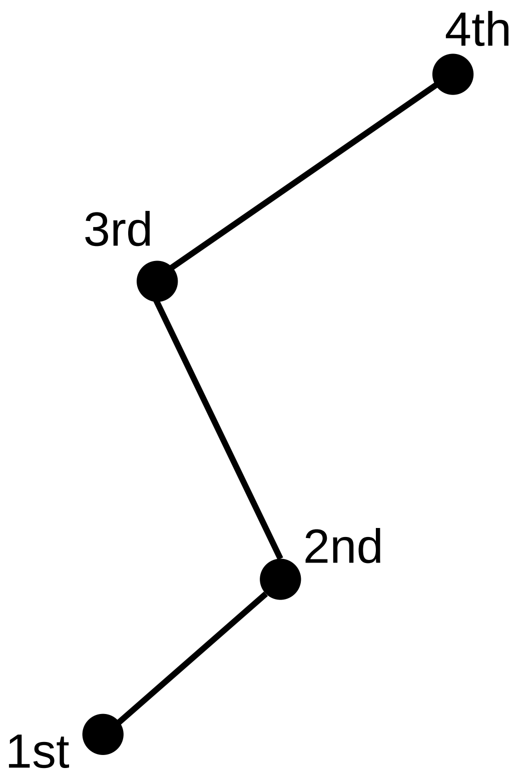
Estimated variances of the movement process are dynamic, **can change over a trajectory**.

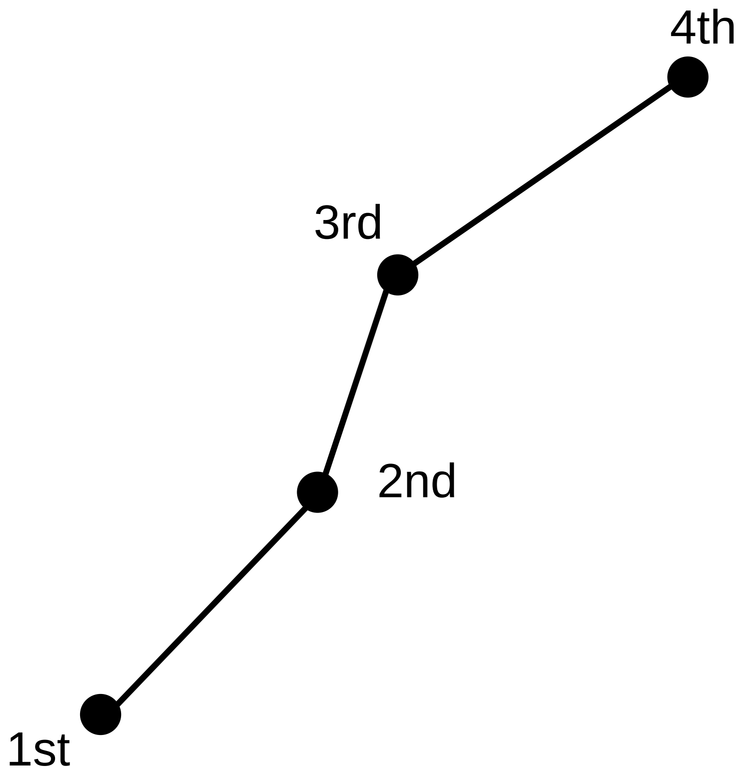
- Variance of **dBB**: changes in speed and/or deviation from straight line
- Variances of **dBGB**: parallel (changes in speed) and orthogonal (deviation from the straight line)



Variance of dBB (Dynamic Brownian Bridge Movement Model)

dBB allows to assigns a probability of **where the animal could have been when we did not observe it (between two known locations)**. It assumes brownian motion between consecutive points and returns the probability of being anywhere between any two points at any given time.

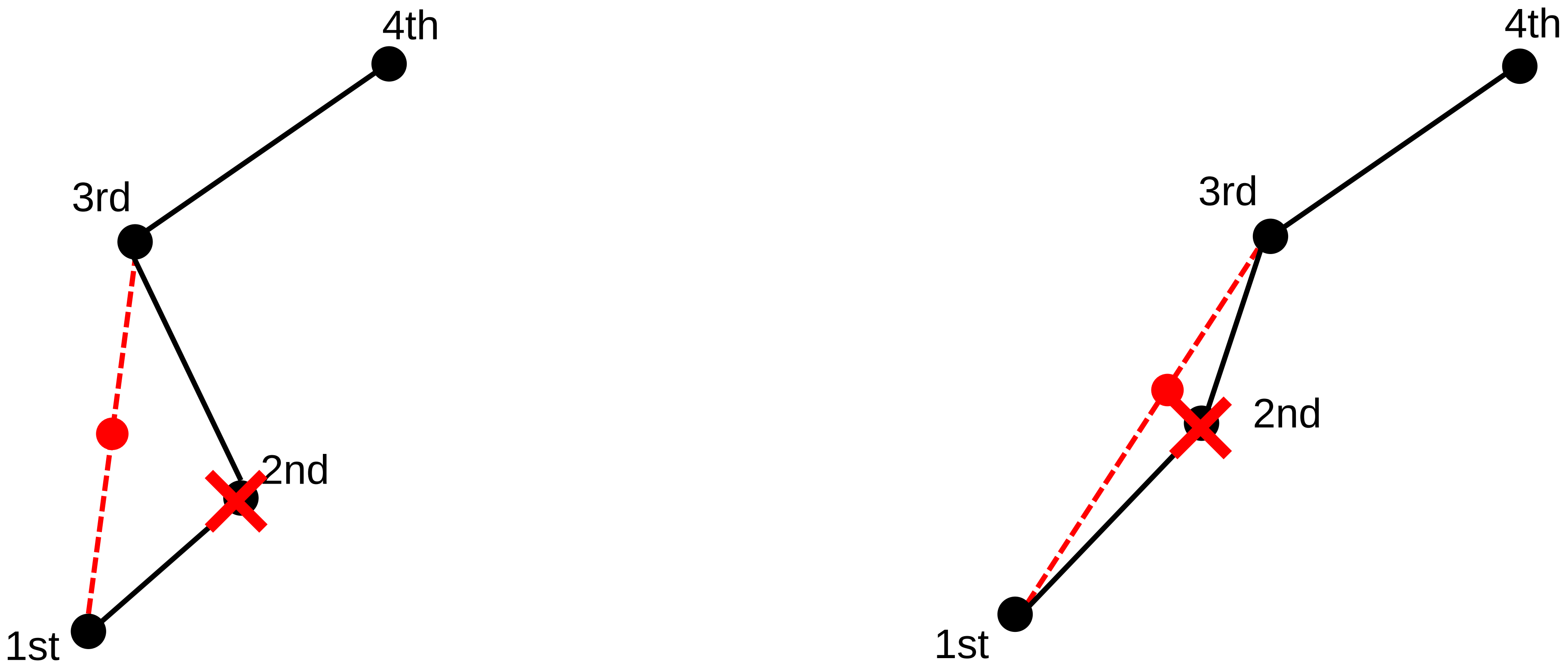




Variance of dBB (Dynamic Brownian Bridge Movement Model)

The estimated probability is based on the movement variance of the trajectory: **the faster and more direct, the less the variance** between the two points.

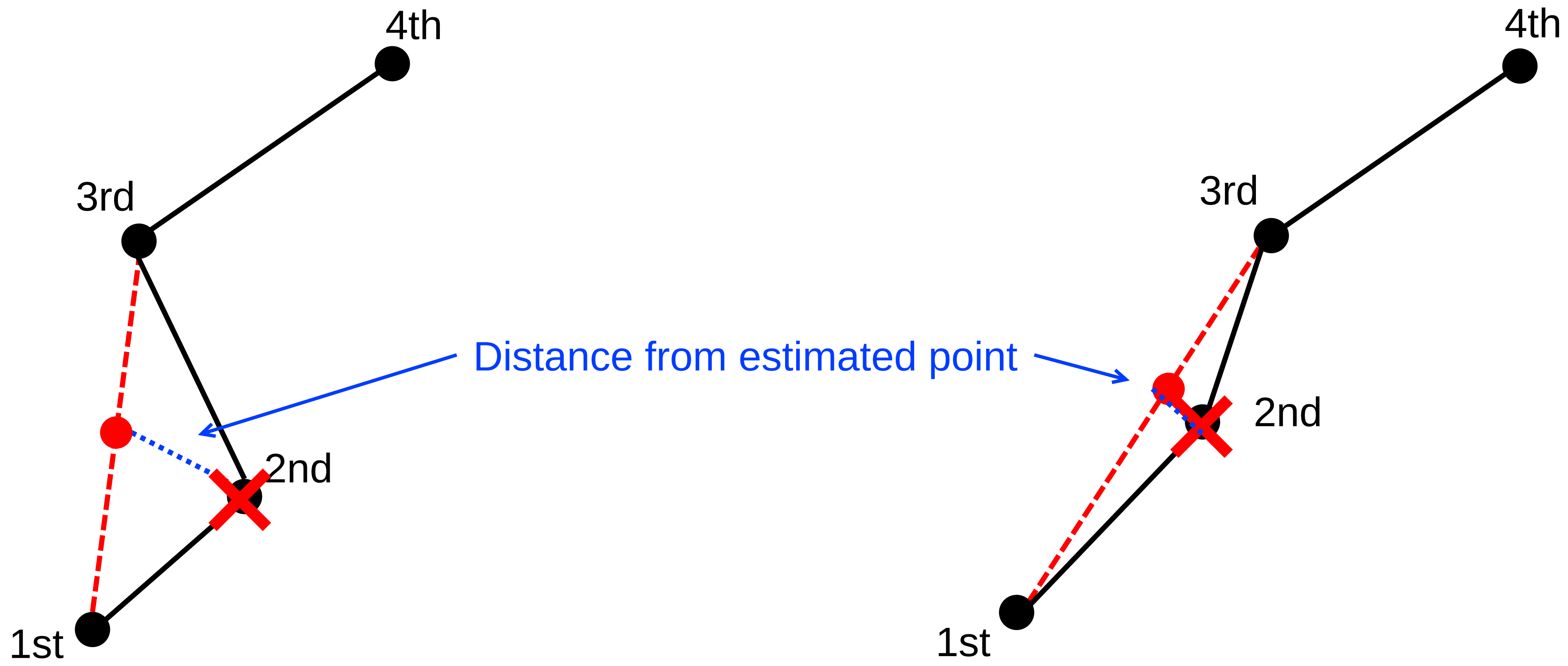
Variance is calculated leaving out the 2nd of three locations and calculating its **distance between the observed 2nd location and the estimated 2nd location.**



Variance of dBB (Dynamic Brownian Bridge Movement Model)

The estimated probability is based on the movement variance of the trajectory: **the faster and more direct, the less the variance** between the two points.

Variance is calculated leaving out the 2nd of three locations and calculating its **distance between the observed 2nd location and the estimated 2nd location.**



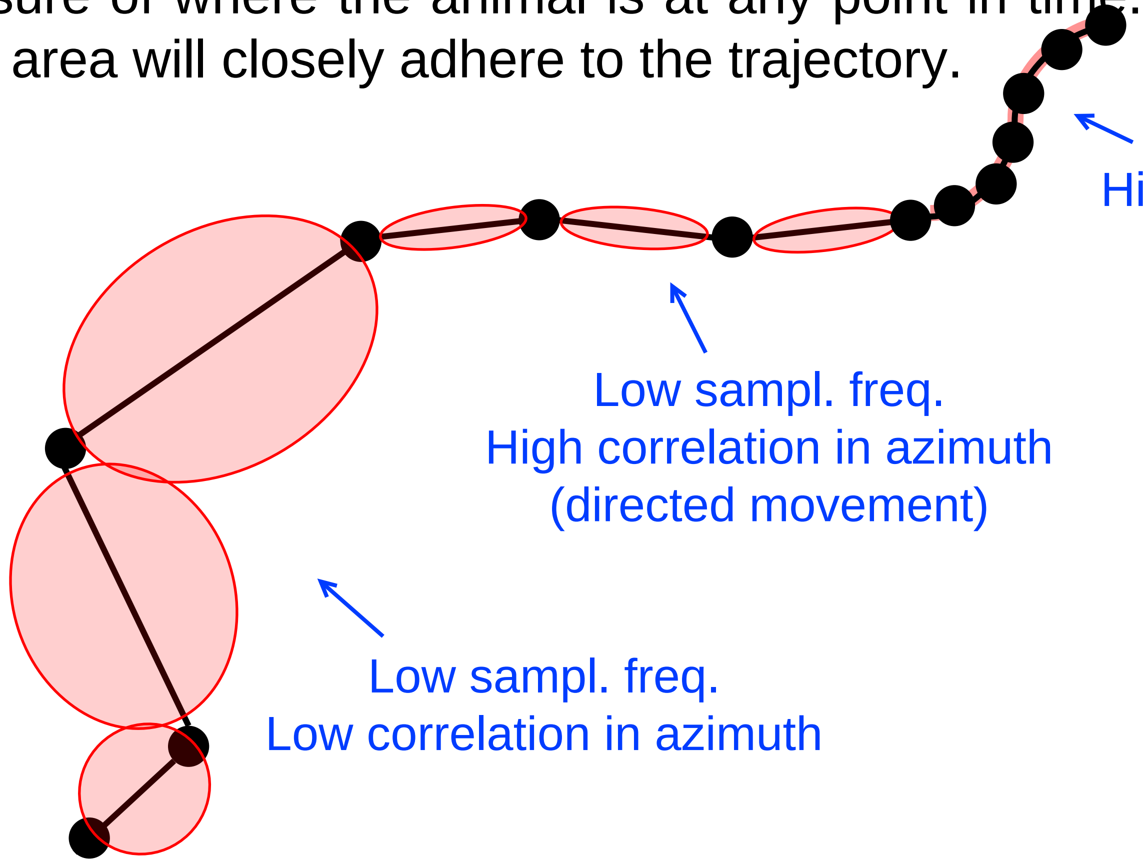
Variance of dBB (Dynamic Brownian Bridge Movement Model)

Based on the calculated **variance**, the dBB will be able to calculate, for each time gap between any two consecutive locations, **the probability of where the animal could have been**.

The smaller the time gap and the more consistent the movement process, the smaller the probability area between locations, which is allowed to vary along the trajectory.

With a 1 Hz sampling rate we are sure of where the animal is at any point in time. There the variance will be close to 0 and the probability area will closely adhere to the trajectory.

You will see dBB again tomorrow in relation to **space use**. The estimated probability can be used to calculate the area used by an animal during the time of tracking: **Utilization Distributions (occurrence ranges).**

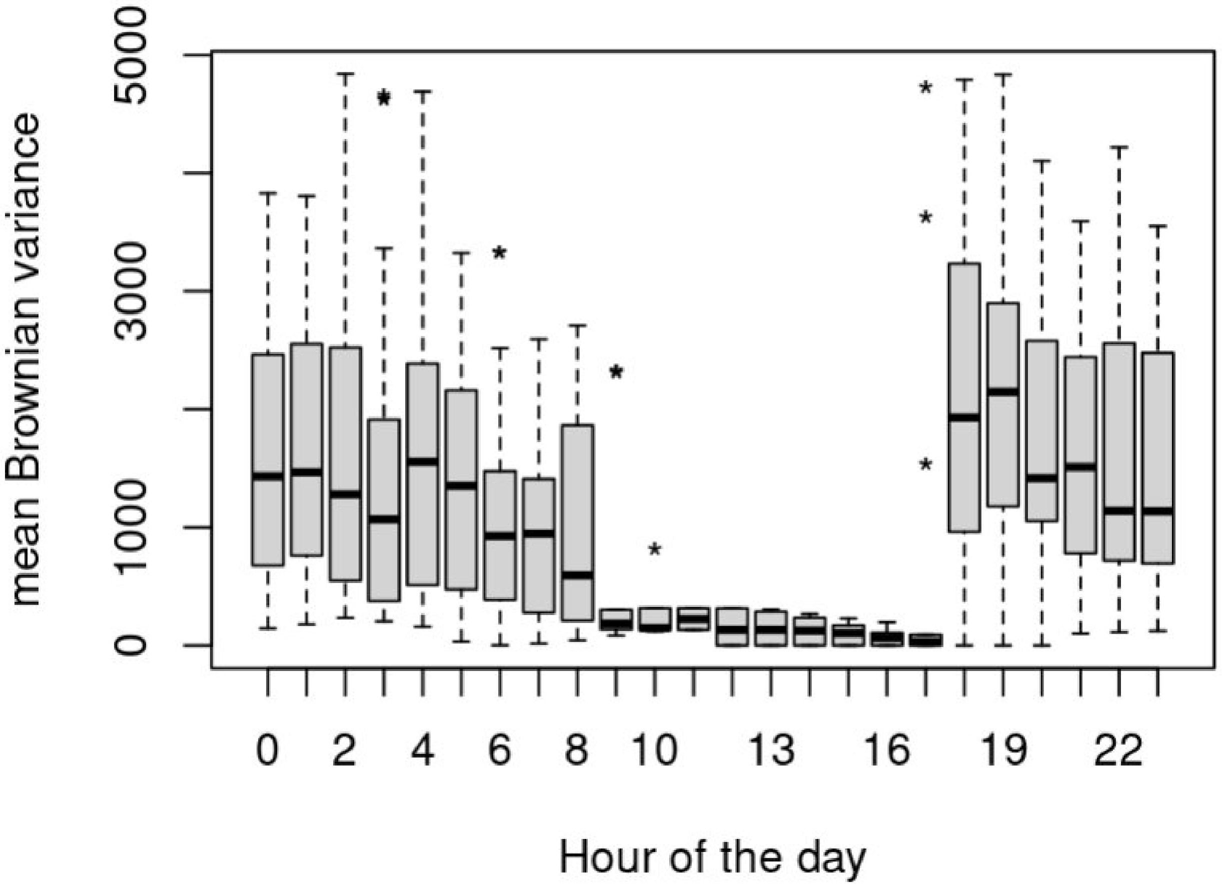


Very high sampl. freq.
High correlation in azimuth
as a consequence

Low sampl. freq.
High correlation in azimuth
(directed movement)

Low sampl. freq.
Low correlation in azimuth

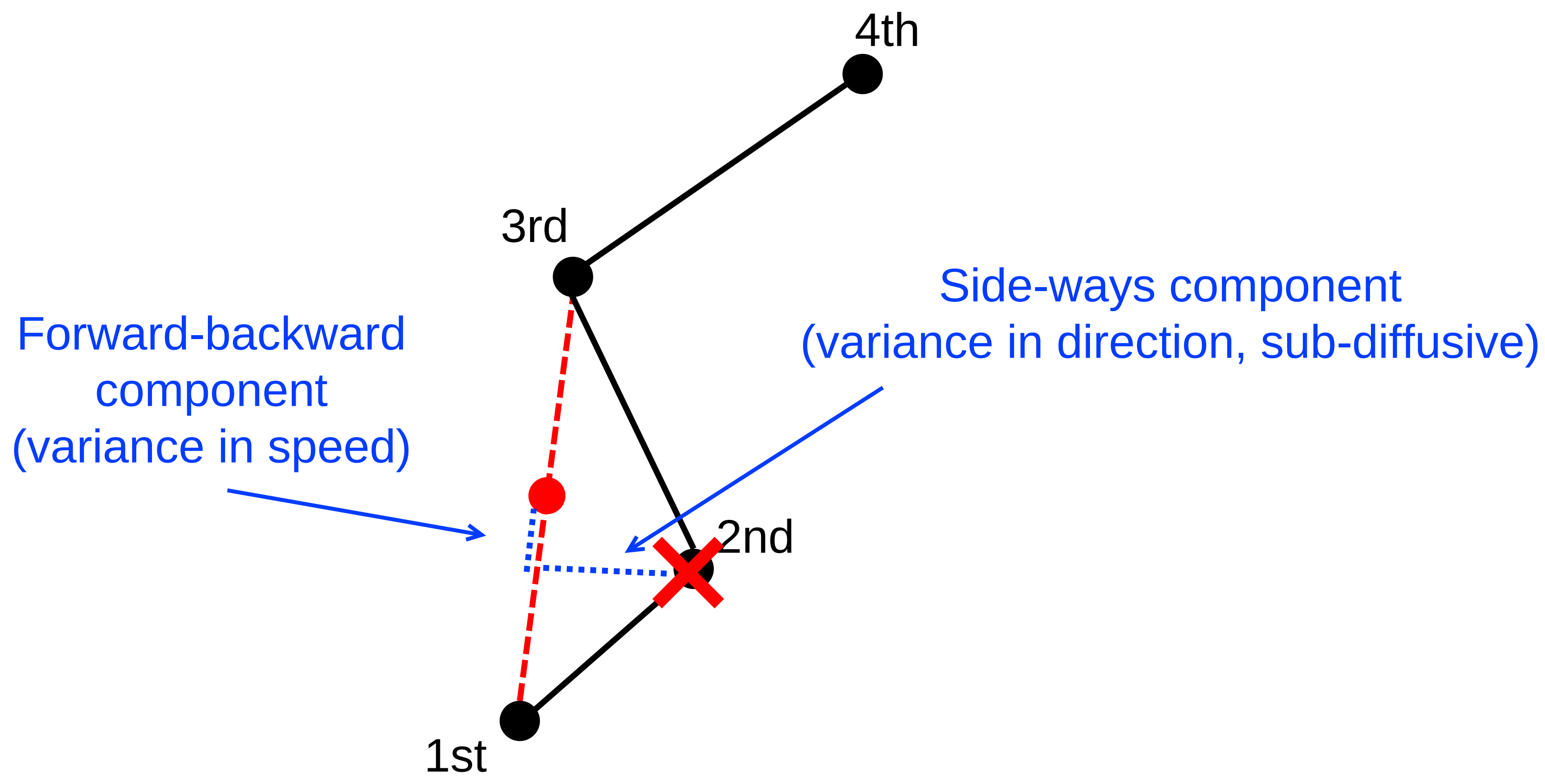
Variance of dBB (Dynamic Brownian Bridge Movement Model)



Leroy moves around during the night, and sleeps during the day. Low variance in the motion variance means **the location I left out is easy to predict (either stationary, or moving with constant speed).**

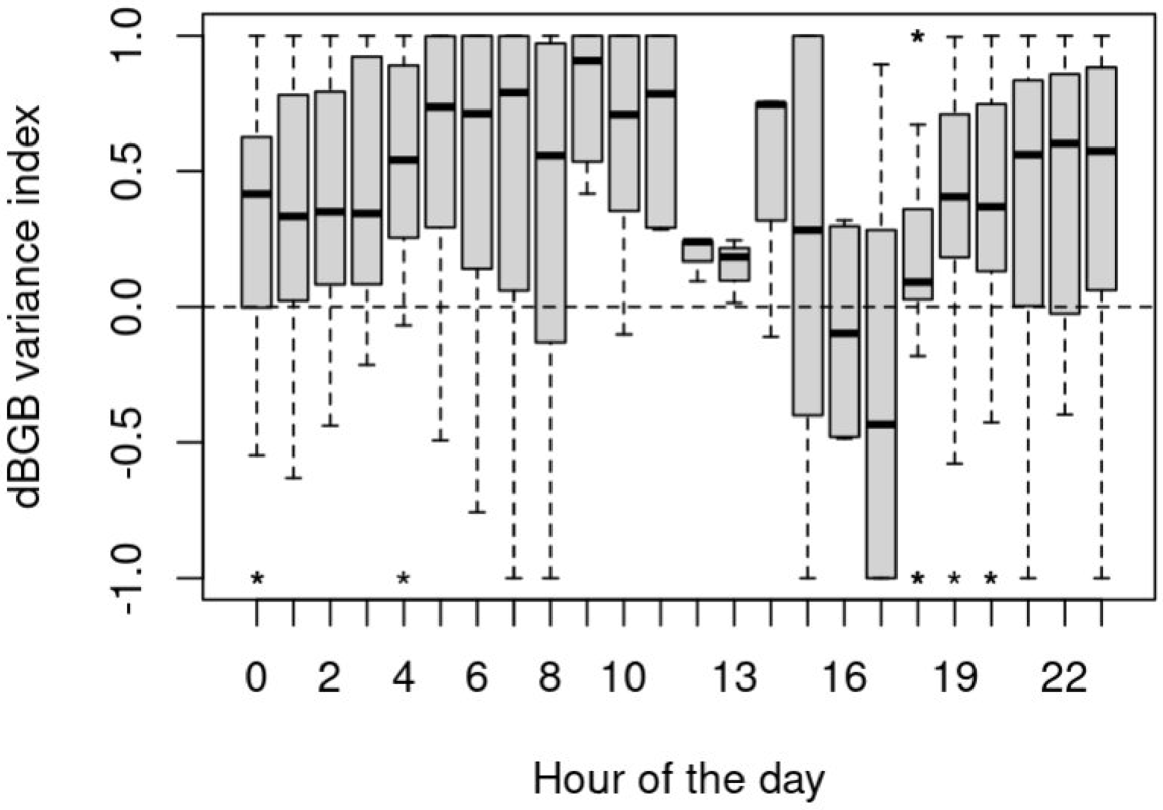
Variance of dBGB (Dynamic Bi-Gaussian Bridge Movement Model)

The variance in dBGB decomposes the variance in two:
Forward-backward (parallel to movement) and **side-ways component** (perpendicular to movement). It helps distinguishing stationary from movement (which resulted in similar dBB variance above).


4th
3rd
2nd
1st
Forward-backward component (variance in speed)
Side-ways component (variance in direction, sub-diffusive)

Variance of dBGB (Dynamic Bi-Gaussian Bridge Movement Model)

- **Values around zero**: true brownian motion, diffusive in all directions, as likely to go sideways than front or back. (Almost nothing in nature is brownian, but **GPS error** is, if data fall on 0 it could be error e.g. in the den, or the sampling frequency could be so low that autocorrelation is removed).
- **Positive values**: directional movement, more front or back than side, less changes in direction and more in velocity.
- **Negative values**: more likely to go sideways, there are more changes in direction and less in velocity, e.g. search behaviour.



Hands on

How to do this in R in the script:
"6C_TrajectoryAnalysis_dBBMM.R"