1 Creating random steps

Given your final bursts we now randomly draw angle and distance from your observed data to get random steps. The angle is taken from the observed position before your starting point of the random step while the distance is taken from the starting point to the next observed position. This means that each burst should at least contain of three observed positions. Before applying the function `rdSteps` you might want to check for correletaion between turning angle and distance. In case your individuals tend to move long distances by turning only in small angles (e.g. a species migrating) but stop for several days for feeding you want to pick the distance and the angle as pairs. If no correlation is found you can pick both variables independently. To check for this we use the `plot` function but first have to convert our ltraj object back to a data frame by using `ld`. AUTOCORRELATION ???

```
## Error in eval(expr, envir, enclos): konnte Funktion "ld" nicht finden
```

```
with(xmpl.cut.df, plot(dist, rel.angle))

## Error in with(xmpl.cut.df, plot(dist, rel.angle)): Objekt 'xmpl.cut.df' nicht
gefunden
```

The plot shows a correleation of step length and turning angle and therefore the random steps should be taken as pairs (`simult = T`). Per default the angle and distance for each random step is drawn from the observed values you provide with `x`. If your random steps shall be taken from a different dataset you can do so by writting it in `rand.dist = YourDataSet`. Hereby, you can also specify a distribution for estimating angle and distance. In our case we stick to the same dataset and apply `rdSteps`.

```
xmpl.steps <- rdSteps(x = xmpl.cut, nrs = 10, simult = T, rand.dis = NULL,
                      distMax = Inf, reproducible = TRUE, only.others = FALSE)
          # use simult = FALSE if your data is not correlated
```

The function `rdSteps` uses some default settings which offer you options to modify your random steps. You can for example, easily change the number of steps taken from the observed data by defining `nrs` (default is 10) or if you only need steps shorter than a certain value specify `distMax` to that value (per default all steps are taken). By setting reproducible = TRUE a seed is used to get reproducible random steps. If you want to exclude your current individual to draw angle and distance from than set `only.others = TRUE`. All in all, `rdSteps` is very straight forward and computes a lot of useful things for you:

```
head(xmpl.steps)

## Error in head(xmpl.steps): Objekt 'xmpl.steps' nicht gefunden
```

The table still includes your cat id, burst id, the rel.angle and dist of your observed positions. Furthermore, the "case" is provided as categories of 0 and 1 for available and used. The "strata" defines all 10 random steps and the one observed location, so you can later tell your function what to compare. Depending on your analysis you might want to compare only your observed positions with the endpoints of your random steps or you want to investigate in the selection of spatial attributes along the path. To do so you find the rel.angle and distance for each random point. Only new coordinates for your random points are missing. Instead two columns provide the differences of your x- and y- coordinates for each random step ("dx" and "dy"). To get new coordinates for your random steps we simply add these differences to your initial coordinates and create two new columns.

```
xmpl.steps$new_x <- xmpl.steps$x + xmpl.steps$dx
xmpl.steps$new_y <- xmpl.steps$y + xmpl.steps$dy
```

Thereby, the ???rst observed position will be overwritten as the ???rst random step. That is necessary because there is no random step to compare the ???rst observed position with (angle could not be calculated!). Also the last observed position will be lost for similar reasons (distance could not be calculated!).

After running this chapter you get your final SpatialPointDataFrame to use with the selection function.

```
head(xmpl.steps)

## Error in head(xmpl.steps): Objekt 'xmpl.steps' nicht gefunden
```