1 Creating random steps

Given your final bursts of at least three positions (this is because for calculating random steps the angle from the point before but the distance to the point afterwards is used) you need to check for correletaion between angle and distance. In case your species tends to move long distances by turning in small angles (e.g. Cougars while searching for prey or wandering around) you want to pick the distance and the angle as pairs dependent on each other. If no correlation is found you can pick both variables independently.

```
## Error in eval(expr, envir, enclos): could not find function "ld"
```

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

## Error in with(xmpl.cut.df, plot(dist, rel.angle)): object 'xmpl.cut.df' not found
```

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 profide with `x`. If your random steps shall be taken from a different data set you can do so by writting it in rand.dist = . Her you can also specify a distribution for estimating angle and distance to draw from.

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

The function `rdSteps` uses several default settings for calculating the random steps. You can 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 `dist.Max` to that value (per default all steps are taken). By setting reproduibilty = TRUE a seed is used to get reproducible random steps. If you want to exclude your current step to draw angle and distamce from than set `only.others = TRUE`.

```
head(xmpl.steps)

## Error in head(xmpl.steps): object 'xmpl.steps' not found
```

You see that the function `rdSteps` does give you the difference (column dx and dy) of your observed positions for each random step. To get new coordinates fo your random steps we simply add the x-coordinates to the dx and the same for the y-coordinates. Thereby, the first observed position will be overwritten as the first random step. That is necessary because there is no random step to compare the first observed position with (no angle to calculate for!). Also the last observed position will be lost for similar reasons (there is no distance to calculate the random step with!).

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

```
head(xmpl.steps)

## Error in head(xmpl.steps): object 'xmpl.steps' not found
```

Your final table includes a column for the differences of the observed positions to each random step (dx and dy) and now also a column for the actual coordinates of each random position. Depending on your analysis you might want to compare only the endpoints (observed positions) of your species or also the spatial attributes along the path.

strata ?? case ??