

Hidden Markov Models - Practical Session 1

Exercise 1 - Muskox data

- a) Read in the data set "muskox.txt" that is provided in the *Lernraum*.
- b) Get an overview of the data and its variables, for example by answering the following questions:
 - How many individual muskoxen are present in the dataset?
 - How many observations/ GPS fixes does the dataset comprise?
 - What is the mean and standard deviation of the step length?
 - How many step length values are missing?
- c) Calculate the first step length using the GPS data. Use the Euclidean distance

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

between the first two UTM coordinates. Compare the value to the step length given in the dataset.

- d) Plot a time series of step lengths of the muskox using `type = "h"` and find arguments for or against the use of HMMs to analyse this data.
- e) Plot histograms of the step lengths and turning angles of the muskox.
- f) Plot the movement track (the x- and y-coordinates) of the muskox and add green points to densely-vegetated locations.
Hint: Use the `type = "l"` argument in the `plot()` function to connect points with lines.
Hint: You can use the `which()` function to find out which data rows correspond to densely vegetated locations.
- g)* An excursion to the **tidyverse**: Understand what happens in the following code. What does the output mean?
Hint: Note that there is no date (day/month/year) given in the dataset.

```
library(tidyverse)
library(lubridate)
strptime(data$time, "%H:%M:%S") %>% hour() %>% diff() %>% table()
```

Exercise 2 - Practicing R commands

- a) The following function `mysum()` returns the sum of two numeric values a and b :

```
mysum <- function(a, b){  
  c <- a + b  
  return(c)  
}
```

In this case, a and b are the *arguments* of the function. To return the sum given some inputs, e.g. $a = 3, b = 5$, type `mysum(3,5)`. Using your results from Exercise 1c), write a function `calc.step()` that returns the step length between UTM coordinates. Use the four necessary UTM coordinate-values (x_1, y_1, x_2, y_2) as arguments for the function.

- b) A for-loop can be used to iteratively apply some operation on a given object, e.g. for summing all values between 1 and 100:

```
result <- 0  
for(i in 1:100){  
  result <- result + i  
}
```

or for filling a vector with the squares of all values between 1 to 10:

```
v <- numeric(10)  
for(i in 1:10){  
  v[i] <- i^2  
}
```

Use your function `calc.step` and write a for-loop to calculate the hourly step length of the muskox for the first 100 observations. Compare your results to the step length values given in the dataset.

- c) Have a look at the help page `?dnorm` and guess what the output of the following commands will be. What do they mean?

```
dnorm(0.5)  
pnorm(0.5)  
qnorm(0.5)  
rnorm(1)  
set.seed(123)  
rnorm(1)  
set.seed(123)  
rnorm(1)
```

Hint: It might help you to have a look at the density of the standard normal distribution which you can plot with

```
x <- seq(-4, 4, 0.01)
plot(x, dnorm(x), type = "l")
```