



## Final Project: Project 3 – 2017-05-15

### NUMA01: Computational Programming with Python

Claus Führer, Alexandros Sopasakis

---

This assignment has 7 tasks.

---

The goal of the final project is to experience programming in a group. Proceed in the following way

- discuss the goal of the project in the group first until you fully understand the problem and get an idea for a practical approach
- divide the problem in subproblems
- discuss how the different parts should be tested
- prepare a max 15 min presentation which should present the background, the organization of your work, your solutions and maybe alternative attempts, which you decided to reject.

Bring your Laptop or an USB stick for the presentation.

---

## Background

In this project we process a bigger amount of data. In an ongoing experiment, which started in mid January 2015 the in- and out-movements of a bird<sup>1</sup> at a nesting box are observed and counted. The data is stored in a file contains lines of the type

```
2015-03-01 14:22:05.911302      2072
```

with the date, the time in UTC and the total number of registered movement so far. The nistbox reports the observations every two minutes 24/24h.

We would like to answer the following questions:

- How does the total number of registered movements per day change over the year? Can we observe the start of the breeding period? Can we observe the start and end of the feeding period?

---

<sup>1</sup>great tit (engl), talgoxe (sv)

- Are there motions when it is dark?
- Are the first and last motion per day somehow related to sunrise (or dawn) and sunset?
- We would like to have also an analysis on an hourly basis for a given day or dates.

Data should be represented graphically to answer these questions, cf. Fig. 1.

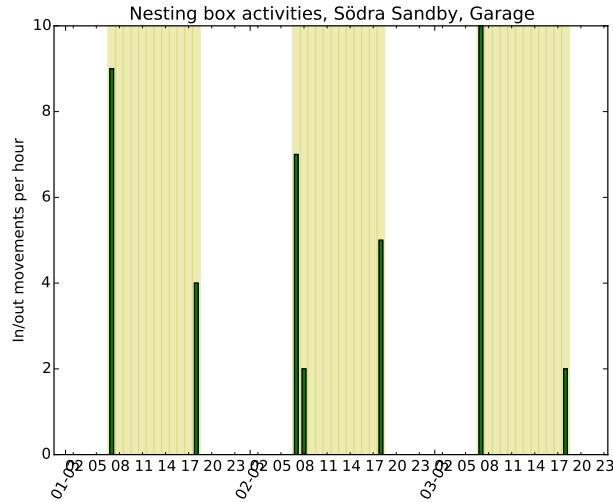


Figure 1: Bird movements over a three day period 2015-03-01 to 2015-03-03

Be aware that some of the data might be corrupted due to transmission problems from the processor at the nesting box (Arduino UNO) via an RF 433MHz signal to a receiving Raspberry PI computer.

Data corruption occurs in two ways:

- The counts are reported incompletely like in the second line:

```
2015-03-01 14:20:06.911302    2072
2015-03-01 14:22:05.911302      72
2015-03-01 14:24:08.911302    2072
```

- There some entire lines missing, that is there might be not always be measurements every two minutes.
- There might be even biological reasons for "wrong" data as the bird might not directly enter or leave. It might flutter at the entrance and so create multiple counts.

These three kind of data corruption require a preprocessing step. The action in the first case is quit obvious. In the second case missing data is just filled in a way which you might consider fair. And in the third case we limit the motions to maximal 4 per minute.

You get a datafile at the day of the project start.

You are allowed and even encouraged to use all appropriate Python modules you find, especially we recommend `datetime` and `astral`. The later has to be installed on your computer by the command

```
conda install --channel https://conda.binstar.org/clausfse astral
```

## Tasks

The following tasks are guidelines to what you need to do. Feel free to do more, or modify them, as long as the spirit of the project remains the same.

### Task 1

---

Read the file into a Python script and convert the dates and times to `datetime` objects.

### Task 2

---

Convert everything to our local timezone (for timezones there are special modules in Python).

### Task 3

---

Write functions or methods to do the preprocessing steps above.

### Task 4

---

Compute the data for the plots. You might design a user dialog for the start date, the number of days and also the intervals for your plot (total per hour, total per day, per week etc.)

### Task 5

---

Make your plots with the dates on the x-axis and if the plot is more detailed even with the hours as ticks on the x-axis.

## Task 6

---

Visualize the daylight and night phases.

## Task 7

---

Add own task if you have time and ideas.



Good Luck!