

Homework 3

As always, use docstrings (where appropriate) and good style, and don't forget to test your code! Write your solutions in a file named `yourname_homework3.py` and submit to `m.spacek@lmu.de` before class 6 (June 14).

You've collected some simultaneous data from two sensors, `x1` and `x2`, along with their timepoints `t`, in 3 lists:

```
import numpy as np

x1 = [ 0.00000000e+00,  2.93892626e+00,  4.75528258e+00,  4.75528258e+00,
       2.93892626e+00,  6.12323400e-16, -2.93892626e+00, -4.75528258e+00,
      -4.75528258e+00, -2.93892626e+00, -1.22464680e-15,  2.93892626e+00,
              np.nan,  4.75528258e+00,  2.93892626e+00,  1.83697020e-15,
      -2.93892626e+00, -4.75528258e+00, -4.75528258e+00,              np.nan]

x2 = [ 0,  2,  3, -1,  6,  7,  8,  9, 10, 10, 10, 10, -1,  9,  8,  7,  6,  5,  3,
       2]

t = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9,
     1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9] # seconds
```

1. How many timepoints are there in `t`? Do the number of readings in `x1` and `x2` match?
2. Write a `for` loop to calculate the product of `x1` and `x2` at each timepoint. Store the results in a list called `y`. When you're done, convert `y` to an array and print it out. What is the data type of `y`?
3. Now convert `x1`, `x2` and `t` to arrays. Redo the calculation of `y` in a single line using vector math.
4. Given the `dtype` of arrays `x1` and `x2`, what `dtype` do you expect array `y` to have? Given their lengths and `dtype`, calculate the expected number of bytes of each array. Check the `.nbytes` attribute of each array to ensure you got it right.
5. It turns out that both sensors have some invalid readings, represented by `np.nan` in `x1` and `-1` in `x2`. Use boolean array operations and boolean fancy indexing to filter out (remove) those invalid values from both sensors, along with their timepoints in `t`. Make sure your filtered `x1`, `x2` and `t` have the same length (i.e., are "aligned"). How many timepoints are left?
6. Recalculate `y` and print it out. Is it aligned with `t`?
7. Use `.argmax()` or `np.argmax()` to find the index `i` of the maximum value of your newly-filtered `x2`. What are the corresponding values of `x1`, `x2` and `t` at that index `i`?
8. Check programmatically if the maximum value of `x2` is unique. What does this tell you about the behaviour of `.argmax()`?