Prof. Dr. Jörg Müller
Arthur Fleig

AI VIII – Serious Games

UNIVERSITÄT BAYREUTH

Exercises for **Programming, Data Analysis, and Deep Learning in Python** (SoSe 2021)

Exercise Sheet no. 8, *Deadline*: Monday, June 14, 10:15

Notes

- Pay attention to the notes on the previous sheet.

Exercise 23 Planets (programming exercise) (9 points)

Download the associated csv file from E-Learning.

a) Import the csv file and display only the first five elements.

b) Plot a histogram that displays the number of planets discovered (“y axis”) depending on the year of discovery (“x axis”). One bin should represent one year.

c) How many different methods are listed in the data frame?

d) What is the largest orbital period?

e) Display the number of planets that have been discovered by each method.
(Hint: Group the dataFrame using the `groupby` command in Pandas.)

f) Between 1989 and 1992, were there any years with no new planets discovered and if yes, which years?

g) Create a scatter-plot that shows the distance of the planets with respect to the year of discovery. What is the general trend?

Exercise 24 Numpy High (programming exercise) (8 points)

Download the corresponding Python script from E-Learning and use it for the following tasks:

a) Load the associated npy file and plot it. It shows a fictional stock.

b) Find the largest profit that can be made with one transaction (buy once, sell once) if short selling is not allowed (you have to buy first, then sell, i.e., you cannot sell what you do not have).
(Hint: Calculating the cumulative minimum using `accumulate` helps a lot.)

c) Determine the time (index) when to buy and when to sell for highest profit if short selling is not allowed. Use the included function `plot_min_max` to display these points in time in the plot from a).

d) Create an array that consists of 50 random integers between 0 and 100. Ignoring the boundaries, find all numbers whose neighboring numbers (left and right) are smaller ("local maxima") and output them.

Exercise 25 Random Walks (programming exercise) (7 points)

A so-called random walk R at times $\Delta t, 2\Delta t, 3\Delta t, ...$ can be constructed via

$$R_{n\Delta t} = \sqrt{\Delta t} \sum_{i=1}^{n} Z_i, \tag{1}$$

where Z_i are independent standard normally distributed random numbers and $\Delta t > 0$ is a given step size. (For example, the value of the random walk at time $2\Delta t$ is given by $R_{2\Delta t} = \sqrt{\Delta t} \sum_{i=1}^{2} Z_i$.) Set $\Delta t = 0.05$ and set the random seed to 0.

a) Plot one random walk until the final time $365\Delta t$.

b) For 150000 random walks generated according to (1), calculate the respective final value $R_{365\Delta t}$. Plot these final values (150000 in total) in a histogram plot with 101 bins. Describe the plot. How does the plot change if the random walks are generated via

$$R_{n\Delta t} = 4 + \sqrt{0.75\Delta t} \sum_{i=1}^{n} Z_i \tag{2}$$

instead of (1)?

c) Create 6 random walks with $n = 20$ steps (final time $20\Delta t$) using formula (1). Make sure each random walk starts in 0. (You may end up with 21 steps.) Divide the 6 random walks in pairs of two and use these pairs to plot 3 random walks in 2D (in the (x, y) plane) by using one random walk for the x-coordinate and another for the y-coordinate. Make sure to plot the 2D random walks in different colors.