

Übungsblatt 11: Python

VORBESPRECHUNG: 27.05.2022

ABGABE: 03.06.2022

Alle Aufgaben sind vollständig zu lösen.

Geben Sie die Übung bitte in Gruppen von 3 oder 4 Personen ab. Erstellen sie dazu ein Team auf ILIAS. Für die Abgabe der Übung selbst laden Sie bitte die Python-Dateien (*.py) hoch. Geben Sie die schriftlichen Aufgaben bitte in Form einer PDF- oder Textdatei (*.pdf oder *.txt) ab. Diese Abgabe wird dann für alle Gruppenmitglieder übernommen.

EXTERNE PAKETE UND PROJEKTE

Aufgabe 1

Ziel dieser Aufgabe ist es, eine Funktion `random_path` zu schreiben. Diese Funktion erhält als Parameter eine Zahl und generiert dann einen zufälligen Pfad dieser Länge. Die Funktion soll den Pfad dann mit Hilfe des Moduls `matplotlib` anzeigen. Der Pfad soll bei den Koordinaten (0, 0) beginnen und dann in jedem weiteren Schritt die x- und y-Koordinaten so verändern, dass der jeweilige Wert entweder um 1 verkleinert wird, gleich bleibt oder um 1 erhöht wird. Daraus ergeben sich dann 8 verschiedene Richtungen, die zufällig in jedem Schritt generiert werden.

Tipps:

1. Erstellen Sie zwei Listen. Eine enthält alle x-Koordinaten, die andere alle y-Koordinaten.
2. Verwenden Sie die Funktion `randint` des Moduls `random`.
3. Sie können den Pfad anzeigen, indem sie die Funktion `plot` des Moduls `matplotlib.pyplot` verwenden.
4. Die Funktion `plot` erhält als ersten Parameter alle x-Koordinaten und als zweiten Parameter alle y-Koordinaten in Form einer Liste.

Aufgabe 2

Auf Ilias finden Sie die Datei `covid-data.csv`¹. Lesen Sie diese mit Hilfe des Pakets `pandas` ein und erfüllen Sie folgende Aufgaben:

1. Machen Sie sich zunächst mit der Tabelle vertraut. Welche Daten werden abgebildet? Welche Spalten gibt es?

¹<https://github.com/owid/covid-19-data/tree/master/public/data/>

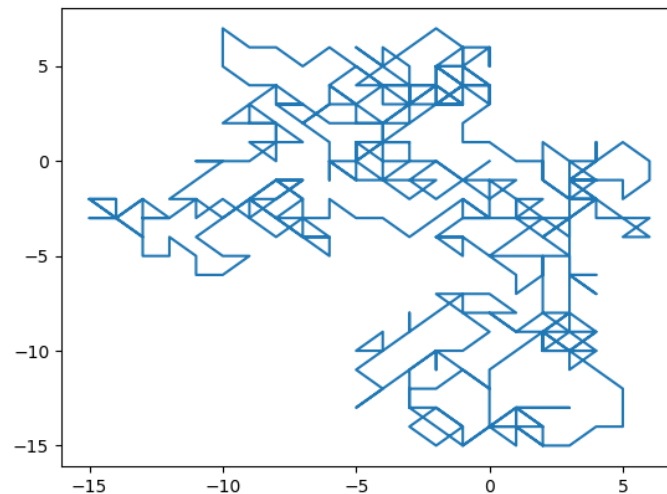


Abbildung 0.1: Ein zufälliger Pfad der Länge 500, generiert durch den Funktionsaufruf `random_path(500)`.

2. Geben Sie für jede Ortschaft (`location`) die totalen Fälle (`total_cases`) sowie die totalen Todesfälle (`total_deaths`) an.
3. Fügen Sie dem in Schritt 2 berechneten Dataframe eine neue Spalte (`death_rate`) hinzu. In dieser Spalte soll die Todesrate vom jeweiligen Land berechnet werden (dieser Wert muss selber berechnet werden).
Tipp: Um einem Dataframe `df` eine neue Spalte `address` hinzufügen zu können, kann `df["address"] = values` gebraucht werden, wobei `values` für die Werte in der Spalte stehen.
4. (Zusatz) Geben Sie die 10 Länder mit den höchsten Todesraten zurück.
Tipp: Brauchen Sie die pandas-Funktion `sort_values()`.

Aufgabe 3

In dieser Aufgabe beschäftigen wir uns mit der `scipy`-Funktion `curve_fit`. Diese Funktion nimmt als Parameter eine Funktion, und zwei Listen `xdata` und `ydata` und gibt ein 2-Tupel zurück. Gehen Sie Schritt für Schritt durch die Aufgabe:

1. Importieren Sie Bibliotheken `numpy`, `matplotlib.pyplot`, `scipy.optimize` und `random`.
2. Definieren Sie die folgende Funktion: $f(x, a, b, c) = a \cdot e^{-b \cdot x} + c$
3. Kreieren Sie eine Variabel `xdata` und weisen Sie ihr eine Liste mit Werten zwischen 0 und 4 in 0.1-Schritten zu.
Tipp: Die Funktion `numpy.arange(a,b,c)` gibt eine Liste zwischen `a` und `b` mit Einträgen in Schritt `c` zurück.
4. Kreieren Sie eine Variabel `ydata` und weisen sie ihr eine Liste zu. Das Element an Stelle `i` in dieser Liste soll folgenden Wert haben:
 $f(x_i, 2.5, 1.3, 0.5) + \text{random.uniform}(-0.1, 0.1)$, wobei x_i das i -te Element in der Liste `xdata` ist.
5. Rufen Sie die Funktion `curve_fit` auf mit den passenden Parametern und speichern Sie das Resultat in Variablen `popt` und `pcov` ab.
6. Rufen Sie die Funktion `plot` mit den Parametern `xdata` und `ydata` auf.

7. Rufen Sie die Funktion `plot` mit den Parametern `xdata` und `f(xdata, popt[0], popt[1], popt[2])` auf.