

Übung 04

Datenanalyse in Python

14.11.2018

1. Der erste Datensatz ist dem aus der letzten Übung sehr ähnlich: Die Gesamtdaten sind eine Matrix, die erste Reihe (nach dem Header) ist eine Wellenlängenchse. Der erste Eintrag in jeder Spalte ist der Loop, der zweite Eintrag entspricht dem Zeitdelay, ist also ein Pikosekunden Wert. Der dritte Eintrag ist ebenfalls ein Zeitwert, aber kann ignoriert werden (ist nicht relevant). Insgesamt ist ein Spektrum mit zwei Peaks zu sehen, d.h. nicht die gesamte Wellenlängenchse enthält ein tatsächliches Signal. Macht euch vertraut mit dem Plotten verschiedener Ausschnitte aus den Daten und den verschiedenen Indizes.
 - Darstellen des gesamten Spektrums als Bild (z.B. mit den Matplotlibbefehlen `imshow` oder `pcolormesh`),
 - Ein Schnitt zu einem bestimmten Zeitpunkt, also einzelne Spektren
 - Einzelne Transiente Kurven (also Schnitte an einzelnen Wellenlängen über alle Zeiten). Achtet darauf, welche Wellenlängen Informationen enthalten und welche nur Rauschen beinhalten.
 - Vergleicht den Durchschnitt aller Loops mit den Kurven einzelner Loops. Entwickelt einen Algorithmus um einzelne Loops aus dem Durchschnitt zu entfernen, um die Datenqualität zu verbessern
2. Der zweite Datensatz ist ein öffentlicher Datensatz von dem Deutschen Wetterdienst. Er beinhaltet Daten über Weinanbau in den Jahren 1970 bis 2017. In der zweiten Tabelle könnt ihr die Beschreibung der STATION IDs finden, die im ersten Datensatz erwähnt werden.
 - Stellt eine Verbindung zwischen den beiden Tabellen her (wie ihr möchtet), um die örtlichen Informationen mit den Weininformationen zu verbinden.
 - Beschäftigt euch mit bestimmten Korrelationen: gibt es einen Zusammenhang zwischen Qualität des Weines und Breitengrad? Ist der Wein bestimmter Stationen über die Jahre besser geworden? Wieviele Hangrichtungen gibt es typischerweise pro Station? Seid kreativ und spielt etwas mit den Daten :)

(Hinweis: die Umlaute in dem Dokument erschweren das Einlesen mit Pandas alleine. Eine mögliche Lösung wäre die Kombination aus `pd.read_table(open("file", 'r'), ..)`)