

Information Visualization 2

Abgabe über die NextCloud bis 23:59 Uhr des o.g. Datums.

Aufgabe 1 Animated Bubble Chart

(10 Punkte)

In dieser Aufgabe soll analog zur Visualisierung auf [-www.gapminder.org/tools](https://www.gapminder.org/tools) ein animierter Scatterplot erstellt werden. In `task11_1.py` finden Sie ein Minimalbeispiel für eine solche Animation: Hier werden zwei Punkte durch lineare Interpolation von drei vorgegebenen Zeitschritten animiert. Dabei wird auch die Punktgröße und -farbe angepasst. Es liegen außerdem vier Tabellen im .csv-Format vor. Diese enthalten folgende Daten von 184 Ländern für jedes Jahr von 1900 bis 2019:¹

- Das GDP pro Einwohner in internationalem Dollar (inflationsbereinigt).
- Die durchschnittliche Lebenserwartung bei Geburt in Jahren.
- Die Anzahl der Einwohner.
- Die Kindersterblichkeit (< 5 Jahre) pro 1000 Geburten.

Die Tabellen sind bereits so gefiltert, dass die Zeilen und Spalten gleich sind, d.h. die Positionen [Land, Jahr] stimmen in allen Tabellen überein. Passen Sie das Beispielprogramm wie folgt an:

a) (3 Punkte)

Laden Sie die Daten. Erstellen Sie ein Numpy-Array, das die Plotkoordinaten für jedes Land an den bekannten Zeitschritten (Keyframes) vorgibt. Das GDP soll auf die x - und die Lebenserwartung auf die y -Achse gemappt werden. Achten Sie darauf, dass das Array direkt zur Nutzung in der Animation geeignet ist.

b) (3 Punkte)

Erstellen Sie zwei weitere Arrays für die Punktgröße (Bevölkerungsanzahl) und -farbe (Kindersterblichkeit). Skalieren Sie die Größe mit einem geeigneten Faktor, sodass die Datenpunkte im Scatterplot differenzierbar sind. Sie können linear skalieren und brauchen nicht Steven's scale (s. Übungsblatt 3) zu berechnen, da die Punktgröße hier nur zur groben Einschätzung der Bevölkerungsanzahl dienen soll.

c) (1 Punkt)

Plotten Sie den initialen Zeitschritt 1900, inklusive aller Mappings (Position, Größe, Farbe). Legen Sie die Skalierung der Colormap auf $[0, 200]$ fest und zeigen Sie die dazugehörige Colorbar.

d) (2 Punkte)

Passen Sie die Funktion `animate` an, sodass die Punkte mit ihren Attributen von 1900 bis 2019 animiert werden. Wählen Sie eine geeignete Auflösung `time_res` und Geschwindigkeit `time_speed`.

e) (1 Punkt)

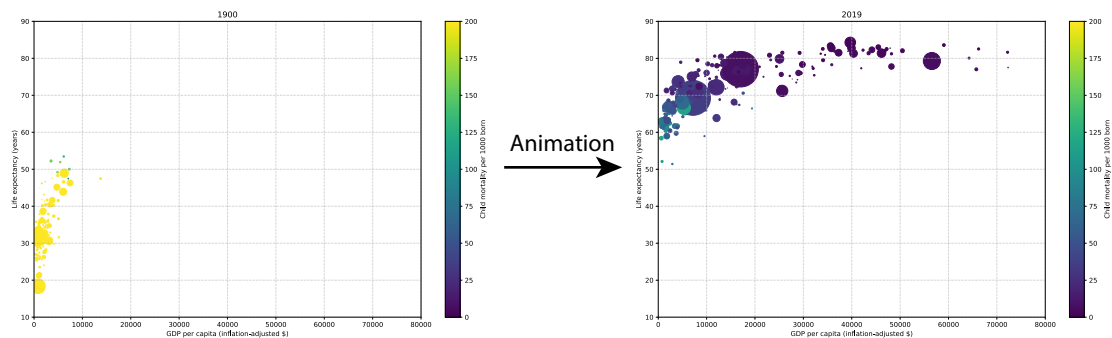
Aktualisieren Sie den Titel während der Animation, damit das korrekte Jahr angezeigt wird. Bei der Interpolation von zwei Keyframes können Sie das nächstgrößere Jahr anzeigen.

¹Reale Daten von [-www.gapminder.org](https://www.gapminder.org).

Hinweise:

- Sie müssen die festgesetzten Achsenlimits anpassen, damit die Daten sichtbar werden.
- Es kann notwendig sein, ein Numpy-Array `a` mittels `a = a.astype('float64')` zu einem Fließkomma-Array zu konvertieren, damit interpoliert werden kann. Dies sollte man nur machen wenn sicher ist, dass das Array ausschließlich Zahlen enthält.
- Die Arrays können ohne aufwändige Schleifen erzeugt werden (siehe Cheatsheet *Accessing and Dimensions*).

Das Ergebnis sieht so aus:



Aufgabe 2 Theorie

(5 Punkte)

Geben Sie die Antworten auf die Theorieaufgaben in der Multiple-Choice-Datei `MC11.txt` an. Es ist immer genau eine Auswahlmöglichkeit richtig. Bitte keine anderen Anmerkungen in diese Datei schreiben und den Dateinamen nicht verändern.

a) (1 Punkt)

Welchen *Frame of Reference* haben die Daten aus Aufgabe 1?

- (a) Abstract
- (b) Spatial

b) (1 Punkt)

Die finale Darstellung in Aufgabe 1 zeigt welche Art von Variablen?

- (a) Univariate Variablen pro Zeitschritt.
- (b) Multivariate Variablen pro Zeitschritt.

c) (1 Punkt)

Welches *time arrangement* haben die Daten aus Aufgabe 1?

- (a) Linear
- (b) Zyklisch

d) (1 Punkt)

Welche Art von *time primitive* haben die Daten, die der Visualisierung in Aufgabe 1 zugrunde liegen?

- (a) Instant
- (b) Interval

e) (1 Punkt)

Welche Art von Mapping wird in Aufgabe 1 erzeugt?

- (a) Static display of temporal relations.
- (b) Dynamic display of temporal relations.