

Zertifikatsklausur 30.01.2021

- Klausurtyp: Open-Book-Exam
- Bearbeitungszeit: 45 Minuten (+ 15 Minuten für Download & Upload)
- Erreichbare Punkte: 45
- Bestehensgrenze: mit 25 Punkten auf jeden Fall bestanden.

Abgabe bis spätestens 11:00 Uhr per E-Mail an kurse@stads.de.

Der Code muss in Python 3.8.5 mit numpy 1.19.2, pandas 1.1.3, matplotlib 3.3.2, seaborn 0.11.0 bzw. plotly 4.12.0 lauffähig sein. Falls weitere Pakete oder andere Versionen verwendet werden, muss die jeweilige Version angegeben werden.

Wir schreiben in dieser Klausur mit x^y die y-ste Potenz von x. Zum Beispiel schreiben wir 3^4 für 3 hoch 4 also für $3*3*3*3=81$.

Aufgabe 1: Grundlagen (18 Punkte)

- Erstellen Sie eine Python-Datei mit dem Namen `<Nachname>_<Vorname>_exam.py` (z.B. `kern_moritz_exam.py`) und bearbeiten in dieser Datei die Aufgabe.

Aufgabe 1a: Get Started (2 Punkte)

- Wir betrachten ein rechtwinkliges Dreieck mit den Seitenlängen a, b und c. Definieren Sie die die Variablen `b` und `c` als 4 bzw. 5. (1 Punkt)
- Geben Sie die fehlende Kathete des Dreieckes, d.h. $(c^2 - b^2)^{(1/2)}$, an. (Tipp: `**`, `print`) (1 Punkt)

Aufgabe 1b: Einfache Funktion (6 Punkte)

- Definieren Sie eine Funktion `pythagoras`, die zwei Variablen `b` und `c` als Input hat. (1 Punkt)
- Die Funktion soll prüfen, ob c^2 kleiner b^2 ist;
 - falls dies der Fall ist: Printe `c^2 muss groesser gleich b^2 sein.` und gebe 0 zurück. (2 Punkte)
 - andernfalls, gebe das Ergebnis von $(c^2 - b^2)^{(1/2)}$ zurück. (2 Punkte)
- Werten Sie die Funktion für die Inputkombination (b=12, c=13) und (b=24, c=7) aus. (1 Punkt)

Aufgabe 1c: Datentypen (6 Punkte)

- Erstellen Sie eine Variable `z` mit dem Wert 1. Konvertieren Sie `z` explizit zu einem Boolean (True/False) und speichern Sie das Ergebnis als `a` (1 Punkt)
- Definieren Sie die Variable `z_is_is_bool` als `TRUE`, falls `z` vom Typ `bool` ist und andernfalls als `FALSE`. Tipp: Verwenden Sie dazu die Funktion `isinstance`. (1 Punkt)
- Erstellen Sie ein `dictionary` mit dem Namen `coronaverbreitung` mit folgendem Mapping auf jeweils einen String. (3 Punkte)
 - `--` -> "Beide sind geimpft."
 - `-+` -> "Nur der Infizierende ist geimpft."
 - `+-` -> "Nur der Infizierte ist geimpft."

- "++" -> "Keiner der Beiden ist geimpft."
- Lassen Sie sich das Element mit dem Key ++ ausgeben. (1 Punkt)

Aufgabe 1d: Schleifen (4 Punkte)

- Erstellen Sie mit einer for-Schleife die folgende Ausgabe (... ausgeschrieben). (4 Punkte)

```
1
22
4444
88888888
16161616161616161616161616161616
32323232...32
64646464646464...64
```

Aufgabe 2: Wichtige Pakete (27 Punkte)

- Erstellen Sie ein IPython-Notebook mit dem Namen <Nachname>_<Vorname>_exam.ipynb (z.B. kern_moritz_exam.ipynb) und bearbeiten in dieser Datei die Aufgabe.

Aufgabe 2a: Numpy (6 Punkte)

- Erstellen Sie geschickt einen Vektor v , der aus 8 äquidistanten Stützstellen auf dem abgeschlossenen Intervall $[0,1]$ besteht, d.h. folgendermaßen aussieht. (1 Punkt)

```
[0, 0.1428, 0.2857, 0.4285, 0.5714, 0.7142, 0.8571, 1]
```

- Initialisieren Sie einen Zufallszahlengenerator. (1 Punkt)
- Verwenden Sie den Zufallszahlengenerator, um zwei Vektoren U und V mit jeweils 1000 unabhängig auf dem Intervall $[0, 1]$ gleichverteilten Zufallszahlen zu simulieren. (1 Punkt)
- Berechnen Sie aus den beiden Vektoren elementweise den Vektor Z als $(-2 \log(U))^{(-1/2)} * \cos(2 * \pi * V)$. (Tipp: `np.log`, `np.cos`, `np.pi`, `np.sqrt`) (2 Punkte)
- Geben Sie den Mittelwert und die Varianz von Z an. (1 Punkte)

Aufgabe 2b: Pandas Basics (9 Punkte)

Für diese Aufgabe benötigen Sie den Datensatz [election.csv](#).

- Importieren Sie den Datensatz `election.csv` und speichern Sie diesen als `df`. (2 Punkte)
- Lassen Sie sich die ersten 10 Zeilen ausgeben. (1 Punkt)
- Aus wie vielen Zeilen und Spalten besteht der Datensatz? (1 Punkt)
- Erstellen Sie eine neue Spalte `relative_votes`, die sich als Quotient der Spalten `candidatevotes` und `totalvotes` ergibt. (1 Punkt)
- Geben Sie an, wie viele Stimmen Donald Trump und Joe Biden jeweils bei der Wahl 2020 in Texas erhalten haben. (2 Punkte)

- Geben Sie an, welcher Kandidat die meisten `relative_votes` bei einer Wahl in einem Bundesstaat erhalten hat. Geben Sie zusätzlich den Bundesstaat und das Jahr an. (2 Punkte)

Aufgabe 2c: Grafiken (6 Punkte)

- Filtern Sie den Datensatz `df` auf die Wahlergebnisse der Demokraten in Delaware. (1 Punkte)
- Erstellen Sie einen Scatterplot mit dem Wahljahr auf der x-Achse und den `relative_votes` auf der y-Achse. Sie erhalten dann eine Darstellung der prozentualen Wahlergebnisse der Demokraten in Delaware über die Jahre hinweg. Sie dürfen dabei ein Paket Ihrer Wahl verwenden (z.B. Pandas, Matplotlib, Seaborn, Plotly). (5 Punkte)

Aufgabe 2d: Pandas Advanced (7 Punkte)

- Gruppieren Sie den Datensatz nach der Spalte `year`. Geben Sie an, in welchem Jahr die meisten Stimmen insgesamt abgegeben wurden. (2 Punkte)
- Gruppieren Sie den Datensatz nach der Spalte `year` und `party_simplified` und aggregieren Sie die Spalten, indem Sie die Zeilen aufsummieren. Speichern Sie das Ergebnis als `df_agg`. (2 Punkte)
- Erstellen Sie eine neue Spalte `relative_votes_agg`, die sich als Quotient der aufsummierten Spalten `candidatevotes` und `totalvotes` in `df_agg` ergibt. (1 Punkt)
- Geben Sie die Partei an, die in den vorhandenen Jahren die meisten prozentualen Stimmen erhalten hat. Geben Sie zusätzlich das zugehörige Wahljahr und den prozentualen Stimmenanteil an. (2 Punkte)

Abgabe

Senden Sie die *beiden* von Ihnen erstellten Dateien bis Abgabeschluss (11 Uhr) per E-Mail an kurse@stads.de.