02. Aufgaben Data Wrangling

0. Vorbereitung

- Richtet für die Aufgaben ein eigenes R Projekt ein.
- Legt die Daten und ein neues Skript mit einem informativen Namen im Projektordner ab
- Schreibt einen Header für das Skript
- Installiert mit dem `install.packages()` Befehl das Paket `tidyr`
- Ladet den Datensatz in das Skript und speichert ihn unter dem Namen `penguins_raw`

1. Inspiziert den Datensatz

- Wie viele Variablen? Wie heißen diese?
- Wie viele Beobachtungen?
- Wie viele unterschiedliche Pinguinarten im Datensatz?
- Wie viele „Adelie“ Pinguine?

2. Räumt den Datensatz auf

- Erstellt einen neuen Datensatz `penguins`, der nur die relevanten Variablen des gegebenen Datensatzes enthält: die Spezies, die Insel, die ID, die vier metrischen Variablen in mm und g sowie das Geschlecht“
- Benennt die Variablen so um, dass sie einem konsistenten Schema entsprechen (z.B. snake case) und keine Leerzeichen beinhalten (informativ sollten sie natürlich trotzdem sein ☺)
- Manche Variablen haben Werte, die Leerzeichen beinhalten oder komplett in Großbuchstaben kodiert sind. Ändert diese so, dass man damit einfach arbeiten kann.

3. Deskriptive Berechnungen

- Lasst euch eine Zusammenfassung aller Variablen im Datensatz ausgeben – bei welchen Variablen ergibt das keinen Sinn?
- Wie viele der unterschiedlichen Pinguinarten wurden auf den jeweiligen Inseln erhoben?
- Berechnet die Mittelwerte der gegebenen metrischen Variablen gruppiert nach Pinguinart und speichert sie jeweils in separaten Objekten.
- Fügt die Objekte zu einem Objekt zusammen – welcher Befehl eignet sich dafür?
- In welchem Format ist dieser zusammengefügte Datensatz aus den Mittelwerten (long vs. Wide)? Bringt ihn mit den `pivot_wider()` und `pivot_longer()` erst in das andere Format und wieder zurück.
- Wie schwer ist der leichteste Pinguin jeder Art?
- Wie schwer sind der schwerste männliche und weibliche Pinguin jeweils? Welcher Spezies gehören sie jeweils an?

4. Bonus & Wiederholung

- Wiederholung for-loop & if - else: Die `aggregate()` und `table()` Funktionen sind praktisch um mit einem einzigen Befehl Daten zusammenfassen. Die Aufgaben von oben lassen sich jedoch auch über einen For-Loop lösen.
  Findet das Gewicht der schwersten männlichen und weiblichen Pinguine mithilfe eines For-Loops, der durch alle Zeilen im Dataframe iteriert, raus. Vergleicht zur Überprüfung das Ergebnis mit dem aus Aufgabe 3.

- Aber welche der beiden Möglichkeiten sollte man denn jetzt benutzen? Dabei gibt es unterschiedliche Dinge zu beachten: Der `aggregate()`–Befehl ist z.B. kürzer und übersichtlicher als die Variante mit for-Loop. Ein weiterer wichtiger Punkt ist die Geschwindigkeit bzw. Effizienz: Besonders wenn man mit großen Datensätzen arbeitet ist es wichtig, effizienten code zu schreiben und zu beachten, wie lange einzelne Funktionen dauern.
  Testet, welche der beiden Varianten, die Minima zu berechnen, schneller ist. Schreibt dafür einen weiteren for-Loop, der den eben geschriebenen Code 10000 mal ausführt und stoppt die benötigte Zeit. Wiederholt dann das gleiche Prozedere mit dem `aggregate()` Befehl von eben. Zur Zeitmessung könnt ihr eine Stoppuhr auf dem Handy benutzen oder die Zeit direkt in R erfassen (https://stackoverflow.com/questions/6262203/measuring-function-execution-time-in-r).