Programmierkurs Julia
Moritz Fromm & Johannes Nägele
Lehrstuhl Prof. Dr. Schlather
FSS 2023

Übungsblatt 4

Aufgabe 1

DataFrames und Plotting (verwendet die Pakete `DataFrames`, `Pipe`, `Dates`, `RDatasets` und `Statistics`).

(a) Ladet euch den Datensatz `airquality` aus dem Paket `RDatasets` mithilfe von `dataset("datasets", "airquality")` ein. Der Datensatz beinhaltet:

Daily readings of the following air quality values for May 1, 1973 (a Tuesday) to September 30, 1973.

- `:Ozone`: Mean ozone in parts per billion from 1300 to 1500 hours at Roosevelt Island
- `:Solar.R`: Solar radiation in Langleys in the frequency band 4000–7700 Angstroms from 0800 to 1200 hours at Central Park
- `:Wind`: Average wind speed in miles per hour at 0700 and 1000 hours at LaGuardia Airport
- `:Temp`: Maximum daily temperature in degrees Fahrenheit at La Guardia Airport.

Gebt euch eine Zusammenfassung des Datensatzes aus (`describe`). Welche Spalte hat die meisten missings?

(b) Gebt alle Spalten bis auf die letzten beiden aus.

(c) `sort`: Ordnet die Beobachtungen nach ansteigender Windgeschwindigkeit, dann nach absteigender.

(d) `transform!`: Wir hätten gerne die Temperatur in Grad Celsius anstelle von Grad Fahrenheit. Wandle also die Datenpunkte der Spalte `:Temp` mithilfe der Umrechnungsformel $°C = (°F - 32) \cdot 5/9$ um (Tipp: verwende `ByRow`).

(e) `dropmissing`, `describe`, `subset`: Betrachtet hier alle Beobachtungen, bei denen der Ozongehalt verfügbar ist. Berechnet und gebt euch *nur* die Gesamtzahl an Observationen sowie den mean vom Ozon aus (Tipp: mithilfe des Operators $\circ$ kann man Funktionen einfach miteinander verknüpfen).

(f) `transform`, `groupby`, `combine`: Fügt die Spalte `TempCat` zu `airquality` hinzu. Sie soll die Observationen in die Klassen heiß/kalt unterteilen, abhängig davon, ob die Temperatur höher oder niedriger als 25 °C war. Gruppiert `airquality` nach `TempCat` und zählt erneut alle Observationen.

(g) `transform`: Fügt ein Merkmal hinzu, das die absolute Abweichung des Ozonwerts von seinem Mittelwert darstellt. Ergänzt `airquality` ebenso um eine Spalte mit dem Namen `OS`, welche das Produkt von Ozongehalt und Sonneneinstrahlung abbildet.

(h) Erstellt einen Linienplot sowie ein Histogramm der Temperatur. Fügt jeweils sinnvolle Achsenbeschriftungen hinzu und entferne jeweils die Legende.

(i) `subset`/`filter`: Wählt alle Observationen aus, bei denen der Monat dem August entspricht, oder die durchschnittliche Windgeschwindigkeit weniger als 7 Einheiten beträgt. Entfernt danach alle Zeilen, bei denen der Eintrag zum Ozongehalt fehlt. Füge die gefilterten Temperaturdaten deinem Histogramm (`histogram!`) hinzu.

(j) `transform!`, `Date`: Wir stellen fest, dass bei unserem Linienplot die x-Achse nur von 1 bis 153 geht, stattdessen sollte dort ja aber eine Zeitangabe bzw. ein Datum stehen. Erstellt euch deshalb aus `:Month` und `:Day` eine neue Spalte `:Date` im Datumsformat.

(k) Wenn wir jetzt nochmal unseren Linienplot erstellen, dann bemerken wir, dass dort nun ein ordentliches Datumsformat auf der x-Achse zu sehen ist. Passt dieses nun so an, dass dort nur die entsprechenden Monate stehen (Tipp: Das entspricht dem Format `"mm"`; verwendet `xticks=(tick_years, date_tick)`, wobei `tick_years` eine range ist und `date_tick` mittels `Dates.format` erstellt wird).

Aufgabe 2 (Zusatzaufgabe)

Überlege dir, wie man wie man pipes auch ohne das Makro `@pipe` auf Funktionen, die mehrere Argumente haben, anwenden kann (Tipp: anonymous functions).

Viel Erfolg!