Programmierkurs Julia
Johannes Nägele
Lehrstuhl Prof. Dr. Schlather
FSS 2024

Übungsblatt 7

Aufgabe 1 (10 Punkte)

Ladet Euch für die folgenden Aufgaben den Datensatz `hflights.csv` herunter und lest ihn als `DataFrame` ein.

Ein kleiner Tipp zum Plotting: Falls bei euch das Plotting sehr lange dauert, dann liegt das eventuell an VSCode. Es kann helfen, die Option `Julia: Use Plot Pane` auszuschalten und den aktuellen Julia-Prozess neuzustarten. Dann öffnen sich Plots in einem neuen Fenster und (eventuell) schneller.

(a) Verschafft Euch einen ersten Überblick und schaut die Datentypen der Spalten an. Die Spalte `Month` soll dabei kategorisch und in aufsteigender Reihenfolge angeordnet sein.

(b) Stellt mit einem geeigneten Plot die Anzahl an Flügen pro Monat dar und ändert die verwendete Farbe zu Dunkelblau (Barplot oder Histogramm?).

(c) Filtert den Datensatz nach den Carriern `DL` und `EV`. Erstellt den Plot aus der vorherigen Teilaufgabe erneut und passt die Farbe der Balken an den Anteil an Flügen vom jeweiligen Carrier an (Tipp: `group=...`).

(d) Filtert nach Flügen mit weniger als 2 Stunden Verspätung bei der Landung. Verwendet ab jetzt diesen gefilterten Datensatz.

(e) Stellt die auftretenden Verspätungen in einem geeigneten Plot dar (Tipp: Welche Zeiteinheit haben wir wahrscheinlich im Datensatz? Barplot oder Histogramm?).

(f) Erweitert die vorherige Teilaufgabe mit verschiedenen Farben für die Startflughäfen (Tipp: Es macht Sinn zu normalisieren; hilfreich ist auch das Argument `alpha`).

(g) Erstellt nun einen neuen Plot für die Relation `AirTime` $\sim$ `Distance`. Stellt hier auch die Verspätung bei der Landung dar (Tipp: `marker_z`; ergänzt am besten eine colorbar und einen dazugehörigen title. Es kann hilfreich sein, an `markersize` und `markerstrokewidth` rumzuschrauben).

(h) Fügt dem Plot eine Ausgleichsgerade (Lineare Regression) hinzu.

(i) Unterteilt diesen Plot wiederum nach dem Startflughafen in mehrere Subplots (Tipp: `marker_z` sowie `group` in Kombination mit `npanels` als Anzahl der Gruppen; bei der Regression kann man über die Gruppen eines `GroupedDataFrame` loopen und zu diesen dann auch die Ergebnisse von `predict` wieder hinzufügen).

Aufgabe 2 (5 Punkte)

Wiederholung.

(a) Berechnet $a = (12 + 22 + ... + 152)$.

(b) Schreibt eine Funktion `print_triangle(n)` die Output dieser Form erzeugt:

```
  julia> print_triangle(5)
*
**
***
****
*****
```

(c) Bestimmt die Summe aller (natürlichen) Zahlen kleiner 200, die ein Vielfaches von 6 sind.

(d) Schreibt eine Funktion `rev(text)`, welche für einen eingegebenen String das gleiche macht wie die Funktion `reverse`. Es soll also die Reihenfolge der Buchstaben umgedreht und das Ergebnis zurückgegeben werden (Tipp: Iteriere mit einem `for`-loop über `text`.).

Aufgabe 3 (Zusatzaufgabe, 6 Punkte)

In dieser Aufgabe lernen wir implizit zwei Dinge:

1. Was macht einen `AbstractArray` aus?
2. Wie können wir inkompatible Interfaces miteinander verknüpfen?

Den hier verwendeten Trick nennt man auch *adapter pattern* oder *wrapper* (was ein *design pattern* ist, wird später erklärt). Nehmen wir mal an, für eine Linked List (vom letzten Blatt) `list` und einen Array `foo = [0.0, 0.0]` würden wir gerne `foo .= 2 .* list` rechnen, sprich broadcasting verwenden. Das funktioniert aber nicht, denn broadcasting geht nur für Typen, die Julia als Array auffassen kann:

```
julia> foo .= 2 .* list
ERROR: MethodError: no method matching length(::LinkedList{Float64})
```

Also müssen wir unsere List entweder konvertieren, oder aber via einen *wrapper* übergeben. Geht dazu wie folgt vor:

(a) Definiert einen (parametric) type `ListArray`, der ein Subtyp von `AbstractArray` sein soll sowie ein Datenfeld vom Typ `LinkedList` hat.

(b) Schreibt eine Funktion `Base.getindex(ar::ListArray, i::Int)`, die uns entsprechend den `i`-ten Wert unserer Linked List zurückgibt.

(c) Schreibt eine Funktion `Base.size(ar::ListArray)`, welche uns die Länge unserer Linked List im `ListArray` zurückgibt.

(d) Nun sollte unser Ausgangsproblem gelöst sein, wenn man `ListArray(list)` anstelle von `list` verwendet.

Viel Erfolg!