Zadanie: Analiza danych lotniczych

Rozwiązania: mdenkiewicz(at)student.uw.edu.pl z tematem [python2018]

Celem jest napisanie skryptów służących do analizy i wizualizacji danych dotyczących lotnictwa pasażerskiego.

Do wyboru jest:

A) Zaimplementowanie 2 z 3 poniższych funkcjonalności w formie notebooka jupyter/pliku *.py
   a) Każda funkcjonalność powinna być zrealizowane przez osobną funkcję, która może używać wstępnie przetworzonych danych, i która zwraca wyniki/rysuje wykres analizy dla podanego argumentu (określonego w funkcjonalności np. kraju, trasy, itp., przykład poniżej)

B) Zaimplementowanie 1 z 3 funkcjonalności jako skryptu wczytującego polecenia z linii komend (zachęcam! - uważam, że to dla was najbardziej rozwojowe)
   a) Stwórz plik py, który będzie wczytywał argumenty do pow. funkcji jako parametry linii komend i drukował/wypisywał wynik bądź zapisywał wykres do pliku.
      i) pierwszym argumentem powinna być nazwa komendy (podane w nawiasach kw. na początku podpunktu)
      ii) Nazwa pliku wynikowego powinna być podana jako argument opcjonalny.

Uwagi:

1. Można założyć, że dane znajdują się w tym samym katalogu co notebook/skrypt
2. **Proszę nie używać ścieżek bezwzględnych (“C:\Users\...”)**
3. **Zawsze trzeba wczytać wszystkie dane z “simple_avia_par” (nawet jeśli niektóre lata są niepotrzebne).** Podpowiedź - wczytaj dane do takiej postaci, jak w pliku all_data_example.csv - do ćwiczeń, na początku, można użyć tego pliku, jednak w **oceniane będzie wczytywanie danych z w.w. katalogu.**
4. Gwiazdka (*) oznacza zdanie trudniejsze, nieobowiązkowe.

Źródłem danych są:

1) pliki z katalogu “simple_avia_par” *[jeżeli ktoś użyje danych z avia_par, liczy się to jako gwiazdka*, z danych tych usunąłem jeden plik - dot. Czarnogóry - nie było tam odp. danych]*
   a) każdy plik to 1 rok
   b) code_dep i code_arr - odp. dla lotniska wylotu (departure) i przylotu (arrival): kody kraju, połączony z kodem ICAO lotniska np. “PL_EPWA” to Lotnisko Chopina w Polsce.
   c) passengers - liczba pasażerów na trasie w danym roku **w tysiącach**
   d) seats - liczba dostępnych miejsc na trasie w danym roku **w tysiącach**

e) *Źródło danych (orginalnych) to Eurostat (https://ec.europa.eu/eurostat/web/transport/data/database)*

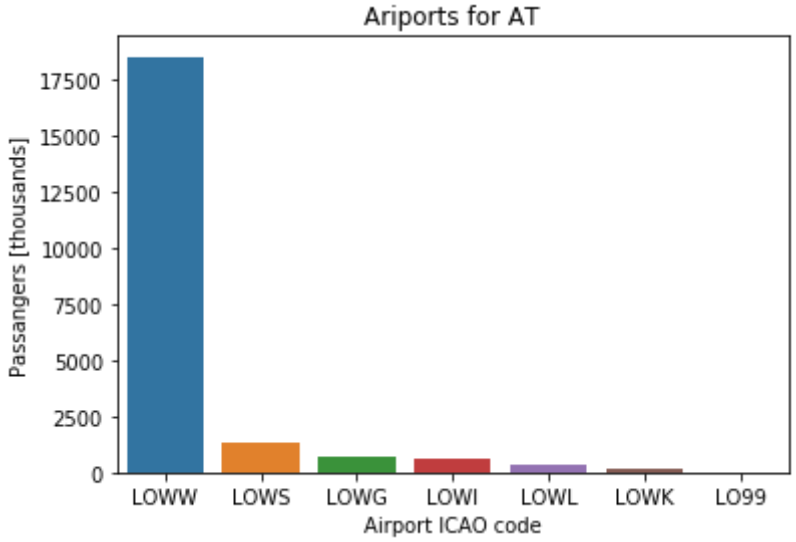
2) plik “airport-codes.csv”
   a) koordynaty (*latitude_deg* i *longitude_deg*) w stopniach (długość i szerokość geograficzna)
   b) pełne nazwy lotnisk (name)
   c) ident - identyfiaktor ICAO
   d) *Źródło - odnajdę na życzenie*
3) plik “country_codes.txt”
   a) nazwy krajów i ich kody
   b) Źródło - wikipedia

Funkcjonalności:

- Poza F3 nie oczekuję pełnych nazw krajów ani lotnisk
- Oczekuję ładnych etykiet osi na wykresach.
- Nie oczekuję odtworzenia w 100% wyglądu wykresów przykładowych (ale upiększanie mile widziane)

F1. Dane dot. kraju (przykłady: Austria, AT):

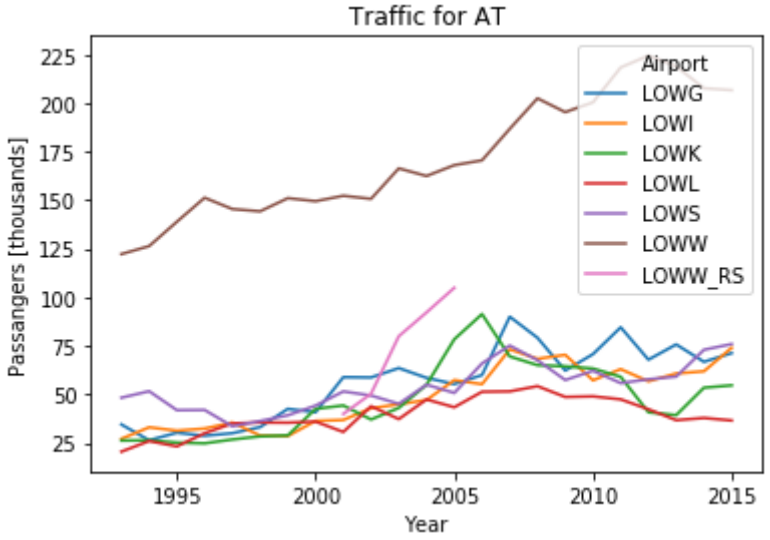
1. [airports] Barplot całkowitej liczby pasażerów (odlatujących i przylatujących) do każdego lotniska w kraju, posortowany malejąco **- dane z ostatniego dostępnego roku. Przykładowo: powinna istnieć funkcja, np. plot_airports, którą można wywołać tak plot_airports(‘AT’), albo która zwróci podany wykres:**



W wersji skryptowej powinno się wywoływać skrypt tak (nazwa pliku powinna zawierać inicjał imienia, nazwisko oraz nr. funkcjonalności):
python3 mdenkiewicz_F1.py airports AT

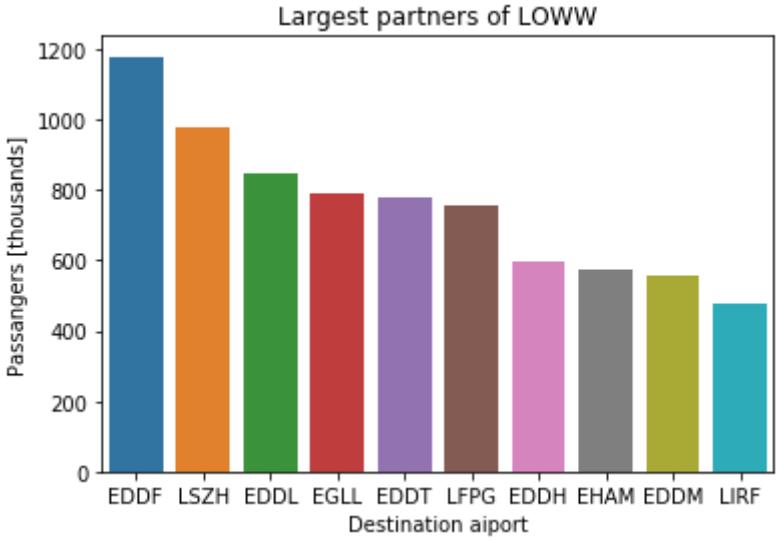
2. [traffic] Wykres liniowy liczby pasażerów odlatujących z każdego z lotnisk we wszystkich latach:
   a. podpowiedź: użyj np. seaborn.lineplot
   b. aby zmienić etykietę legendy w pow.:

i. ax = sns.lineplot(...
ii. ax.get_legend().texts[0].set_text('Airport')

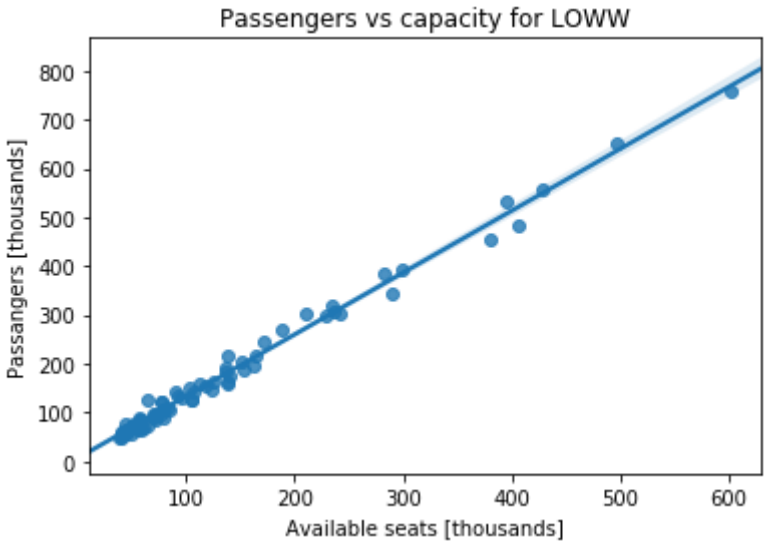


F2. Dane dot. lotniska (przykłady: Wiedeń, LOWW):

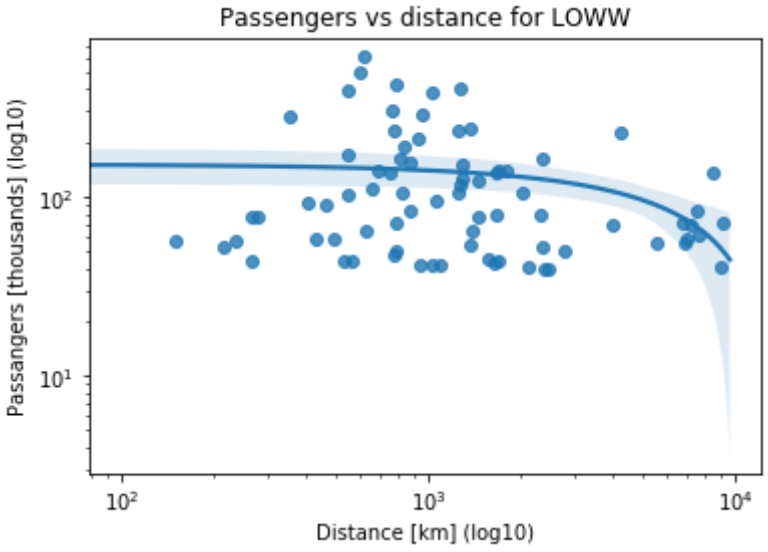
1. [partners] Barplot dla max. 10 partnerów (lotnisk) danego lotniska, największych pod względem ruchu (l.pasażerów - odlatujących i przylatujących sumarycznie) **- dane z ostatniego dostępnego roku**



2. [capacity] Wykres z linią regresji liczba pasażerów odlatujących vs liczba wolnych miejsc (też odloty). Jeden punkt to jedno lotnisko docelowe. **- dane z ostatniego dostępnego roku**



3. [distance]* Wykres regresji: liczba pasażerów odl. vs dystans do lotniska docelowego. Obie skale logarytmiczne.
    a. Liczenie dystansu - np. https://stackoverflow.com/questions/19412462/getting-distance-between-two-points-based-on-latitude-longitude



F3. Dane dot. trasy (podajemy rok, port początkowy i końcowy).

(nie trzeba tworzyć plików, wystarczy wypisać na wyjście)

1. [nazwa niepotrzebna] Jeżeli jest ruch na tej linii, chcemy dostać informację o:
    a. liczbie pasażerów
    b. liczbie dostępnych miejsc
    c. pełną informację o lotniskach (nazwa, w jakim kraju się znajduje)

d. długość trasy w km (Liczenie dystansu - np. https://stackoverflow.com/questions/19412462/getting-distance-between-two-points-based-on-latitude-longitude)

Wskazówka - nie trzeba wykonywać operacji merge/join, wystarczy dobrze operować indeksami i metodami loc/at itp...

Przykład:

print_route(2015, 'LOWG', 'EDDL')

Route information (2015):
Origin: Graz Airport (LOWG) airport in Austria (AT)
Destination: Düsseldorf International Airport (EDDL) airport in Germany (DE)
Distance: 791 km
Passengers: 26724
Available seats: 43855

2. ** Jeżeli nie ma ruchu na tej linii - wyszukaj przesiadkę (max 3) i podaj pełną listę etapów, wraz z w.w. danymi *(uwaga: tu jest sporo pracy)*