

Upskill – Digital Skills and Jobs

Python for the Web Course

Introduction to Python Programming

Final Project

Airline companies' delays

Delivery until the 18th of December de 2020

The non-punctuality of commercial flights causes enormous losses every year. A flight delay can mean the loss of a call, an overnight stay at the airport waiting for the next call, and ultimately, a deal undone by being late or not attending an important meeting.

This work aims to analyze the phenomenon of delays in commercial flights, using data from the United States Department of Transportation. The [Bureau of Transportation Statistics](#) publishes all kinds of statistics on US commercial aviation. Based on this data we are interested in the following **top 10** indicators:

- The average delay by airline, expressed in minutes;
- The ratio of delayed flights, by airline;
- The average delay by destination airport, expressed in minutes;
- The ratio of flights delayed to arrival, by destination airport.

The result should be presented in the form of a figure with four bar graphs similar to Figura 1 on the next page.

The file with the data is related to a single month and must include the columns with year and month (same for all lines), so that it is possible to extract from one of the lines the information to be placed in the title of the figure (“February 2015”) in the example on the next page).

EM PARCERIA COM



COM O APOIO



XXII GOVERNO



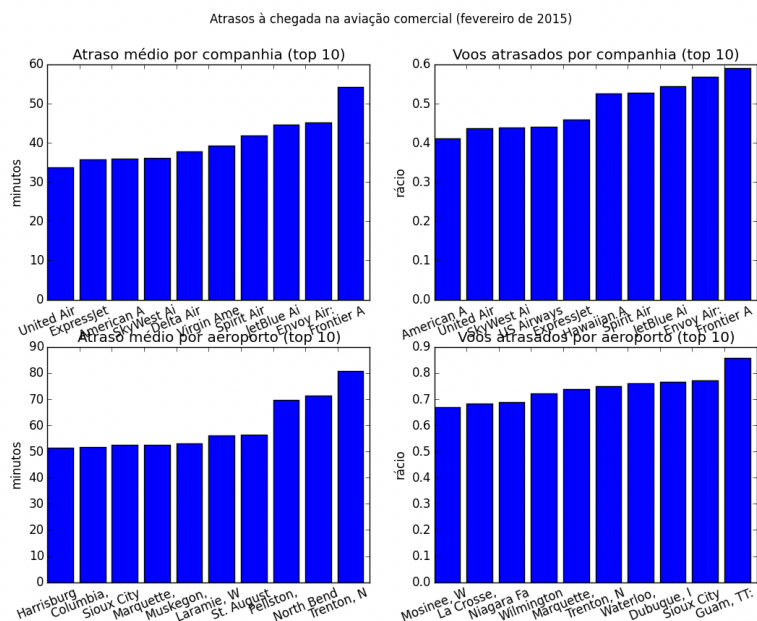


Figura 1. Project Results

Implement a class with the following methods:

1. An **atrasos** function that takes three file names as a parameter. Namely and in order:
 - The name of the CSV file with the relevant statistical data
 - The name of the CSV file with the correspondence between the identifiers and the names of the airlines
 - The name of the CSV file with the correspondence between the identifiers and the names destination airports.

This function should draw a figure with four graphs, as shown in the figure above.

Very important note: The function must read the CSV files *as they are*. Under no circumstances should you change the content of these files before importing them into your program. The first line of the downloaded file determines the names of the table keys. These are the keys that your programs should use. Under no circumstances should they alter or convert these keys (to obtain, for example, more readable key names or in Portuguese). This allows you to easily generate graphs for any geographical area and any time period. It also allows to test all solutions in a uniform way.

Suggestion: Use the bar plots from matplotlib.pyplot.

2. The method **atraso_medio_companhia(tabela_estatisticas, tabela_nomes_companhias)** receives a table with the relevant statistics (read directly from a CSV file downloaded from *TranStats* and unchanged, as described above) and another table that associates identifiers with airline names (also read from a CSV file and unchanged). Returns a pair of lists: abscissa and ordered. The first element of the pair contains the names of the airlines (first ten characters only), the second element of the pair contains the average delays on arrival, by airline. The graph should appear in increasing order of average delay times.

Note: Although the *atrasos* function builds graphs only for the top 10 companies, the graphs generated by this function and following ones must contain values for **all** companies (or airports).

3. The method **racios_atrasos_companhia** receives the same parameters as the **atraso_medio_companhia** delay method. Returns a pair of lists: ordinates and abscissas. The first element of the pair contains the names of the airlines (first ten characters only), the second element of the pair contains the ratio of flights that arrived late, by airline. The chart **should appear in increasing order** of ratio.
4. The method **atraso_medio_aeroporto (tabela_estatisticas, tabela_nomes_aeroportos)** receives a table with the relevant statistics (read directly from a CSV file) and another table that associates identifiers with airport names (also read from a CSV file and unchanged). Returns a pair of lists: abscissas and ordinates. The first element of the pair contains the names of the airports (first ten characters only), the second element of the pair contains the average delays *on arrival*, by airport. The graph should **appear in increasing order** of average delay times.
5. The method **racios_atrasos_aeroporto** receives the same parameters as **atraso_medio_aeroporto**. Returns a pair of lists: abscissa and ordered. The first element of the pair contains the names of the airports (first ten characters only), the second element of the pair contains the ratio of flights arriving late, by airport. The chart should appear **in increasing order** of ratio.

For each method you consider necessary (including the five methods listed above), include the following elements.

- A *docstring* describing the function, as learned in classes.
- A contract Params and Returns, also included in the *docstring*.

Note that this class comprises the minimum integration of object oriented in the solution of this problem. Groups that are able to distribute these methods over multiple classes (ex: class *CalculadorDeAtrasos* which is inherited by other classes that calculate each individual delay) will be beneficiated in the final grade

The module also has to have a *docstring*. Don't Forget to include your names: `__author__ = "Ana Silva"`.

During the development process, pay attention to the following points:

- The code is well structured, broken down into several functions?
- Each function has a simple and well-defined objective?
- There aren't too big? (i.e., they do too much)
- The *docstring* and contracts are clear?
- Is the code duplicate? (maybe I can write a function that avoids duplication?)
- Will the use of a list comprehension, or a high order function, or any other function from the python standard library Python, make my code more readable or more compact?

This Project is to be done in *groups of two*. The students must create a Bitucket group and a repository specific for the project and can push changes until 23h55 of the 18th of December 2020.

The projects will be analyzed by a plagiarism program. Remember: “Students detected in a situation of fraud or plagiarism, plagiarists and plagiarized, will not be approved in this project”.

EM PARCERIA COM



COM O APOIO



XXII GOVERNO

