

Upskill – Digital Skills and Jobs

Curso de desenvolvimento Web com Python

Introdução à programação Python

Projeto Final

Os atrasos das companhias aéreas

Entrega a 18 de Dezembro de 2020

A não pontualidade dos voos comerciais causa todos os anos prejuízos avultados. Um atraso num voo pode significar a perda duma ligação, uma dormida no aeroporto à espera da ligação seguinte, e em última análise, um negócio desfeito por se chegar tarde ou não se comparecer numa reunião importante.

Este trabalho pretende analisar o fenómeno dos atrasos em voos comerciais, utilizando dados do Departamento de Transportes dos Estados Unidos da América. O [Bureau of Transportation Statistics](#) publica todo o tipo de estatísticas sobre a aviação comercial dos EUA. Baseado nestes dados estamos interessados nos seguintes indicadores **top 10**:

- O atraso médio à chegada por companhia aérea, expresso em minutos;
- O rácio de voos com atraso à chegada, por companhia aérea;
- O atraso médio à chegada por aeroporto destino, expresso em minutos;
- O rácio de voos com atraso à chegada, por aeroporto destino.

O resultado deverá ser apresentado na forma de **uma** figura com quatro *gráficos de barras* semelhante à da página seguinte.

EM PARCERIA COM



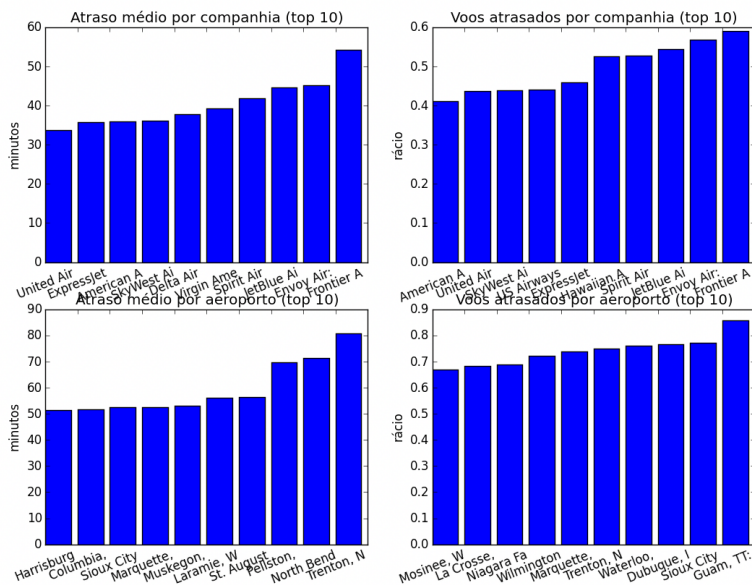
COM O APOIO



XXII GOVERNO



Atrasos à chegada na aviação comercial (fevereiro de 2015)



O ficheiro com os dados é sempre relativo a um único mês, e deve incluir as colunas com ano e mês (iguais para todas as linhas), de modo a que seja possível extrair de uma das linhas a informação a colocar no título da figura (“fevereiro de 2015” no exemplo da página seguinte).

Crie uma classe com os seguintes métodos:

1. Um método `atrasos` que recebe três nomes de ficheiro como parâmetro. A saber e por ordem: o nome do ficheiro CSV com os dados estatísticos relevantes, o nome do ficheiro CSV com a correspondência entre os identificadores e os nomes das companhias aéreas, e o nome do ficheiro CSV com a correspondência entre os identificadores e os nomes dos aeroportos de destino. Esta função deve traçar uma figura com quatro gráficos, tal como exemplificado na figura acima.

Nota muito importante: A função deve ler os ficheiros CSV *tal como estão*. Não deve em caso algum alterar o conteúdo destes ficheiros antes de os importar no seu programa. A primeira linha do ficheiro descarregado determina os nomes das chaves da tabela. São estas as chaves que os vossos programas devem utilizar. Não devem em caso algum alterar ou converter estas chaves (para obter, por exemplo, nomes de chaves mais legíveis ou em português). Tal permite facilmente gerar gráficos para qualquer área geográfica e qualquer período temporal. Permite também testar todas as soluções de um modo uniforme.

Sugestão: Utilize os gráficos de barras do [`matplotlib.pyplot`](https://matplotlib.org/).

2. Um método `atraso_medio_companhia(tabelaEstatisticas, tabelaNomesCompanhias)` recebe uma tabela com as estatísticas relevantes (lida diretamente de um ficheiro CSV descarregado do TranStats e inalterada, tal como descrito acima) e outra tabela que associa os identificadores aos nomes das companhias aéreas (também lida dum ficheiro CSV e inalterada). Devolve um par de listas: abcissas e ordenadas. O primeiro elemento do par contém os nomes das companhias aéreas (primeiros dez caracteres apenas), o

segundo elemento do par contém os atrasos médios *à chegada*, por companhia. O gráfico deve aparecer **ordenado por ordem crescente** de tempos de atraso médio.

Nota: Apesar da função atrasos traçar gráficos apenas para as companhias *top 10*, os gráficos gerados por esta função e seguintes devem conter valores para **todas** as companhias (ou aeroportos).

3. O método `racios_atrasos_companhia` recebe os mesmos parâmetros da função `atraso_medio_companhia`. Devolve um par de listas: abcissas e ordenadas. O primeiro elemento do par contém os nomes das companhias aéreas (primeiros dez caracteres apenas), o segundo elemento do par contém o rácio de voos que chegaram atrasados, por companhia. O gráfico deve aparecer **ordenado por ordem crescente** de rácio.
4. O método `atraso_medio_aeroporto` (`tabela_estatisticas`, `tabela_nomes_aeroportos`) recebe uma tabela com as estatísticas relevantes (lida diretamente de um ficheiro CSV) e outra tabela que associa os identificadores aos nomes dos aeroportos (também lida dum ficheiro CSV e inalterada). Devolve um par de listas: abcissas e ordenadas. O primeiro elemento do par contém os nomes dos aeroportos (primeiros dez caracteres apenas), o segundo elemento do par contém os atrasos médios *à chegada*, por aeroporto. O gráfico deve aparecer **ordenado por ordem crescente** de tempos de atraso médio.
5. O método `racios_atrasos_aeroporto` recebe os mesmos parâmetros da função `atrasoMedioAeroporto`. Devolve um par de listas: abcissas e ordenadas. O primeiro elemento do par contém os nomes dos aeroportos (primeiros dez caracteres apenas), o segundo elemento do par contém o rácio de voos que chegaram atrasados, por aeroporto. O gráfico deve aparecer **ordenado por ordem crescente** de rácio.

Para cada método que considerar necessário (incluindo os cinco acima enumerados), inclua os seguintes elementos.

- Uma descrição no formato *docstring*, tal como sugerido nas aulas.
- Um contrato usando cláusulas `Requires` e `Ensures`, também no formato *docstring*.

Repare que esta classe consiste na integração mínima de programação por objetos neste problema. Grupos que consigam separar estes métodos por mais do que uma classe (ex: uma classe `CalculadorDeAtrasos` que depois é estendida por outras classes cada uma calculando um atraso para um atraso em específico) serão beneficiados na nota final.

O módulo em si deve também estar equipado com uma descrição em formato *docstring*. Não se esqueça de incluir o seu nome: `__author__ = "Ana Silva"`.

Durante o processo de desenvolvimento, preste atenção aos seguintes pontos:

- O código está bem estruturado, decomposto em várias funções?
- Cada função tem um objectivo simples e bem definido?
- As funções não são grandes demais? (isto é, fazem demasiadas coisas)
- O *docstring* e os contratos estão claramente enunciados, em português correto?
- Há código duplicado? (ou será que posso escrever uma função que evita a duplicação?)
- Será que a utilização duma lista em compreensão, duma função de ordem superior, ou duma outra função da biblioteca da linguagem Python, tornará o código mais legível ou mais compacto?

Este é um trabalho de resolução em *grupos de dois alunos*. Os alunos devem criar um grupo no Bitucket e um repositório separado dos exercícios para o qual podem fazer push até às 23h55 do dia 18 de Dezembro de 2020.

Os trabalhos de todos os alunos serão comparados por uma aplicação computacional. Lembre-se: “Alunos detetados em situação de fraude ou plágio, plagiadores e plagiados, ficam reprovados no projeto.

EM PARCERIA COM



COM O APOIO



XXII GOVERNO

