

# PRJ - 22 - Projeto Conceitual de Aeronave

## Lab 01

### Atividade em Grupo

Cap Eng **Ney** Rafael Secco  
ney@ita.br

1T Eng João A. **Dantas** de J. Ferreira  
dantas@ita.br

## Instruções

- Data de entrega: 15/08/2021, 23:59;
- Enviar para dantas@ita.br;
- Pena por atraso: 20% da nota final por dia de atraso (20% no dia 15, 40% no dia 16 e assim sucessivamente);
- Os grupos devem entregar um mini-relatório cada, com, pelo menos, os itens em vermelho. Deve constar no mini-relatório: nome da equipe, nome dos integrantes, gráficos e dados pedidos. **Não** precisa ter introdução, desenvolvimento teórico e conclusão!
- Códigos desenvolvidos devem ser enviados. Os códigos devem ter comentários; e
- Podem discutir soluções com o outro grupo, mas não podem compartilhar a planilha ou os códigos.

## 1 Introdução

O objetivo dessa atividade é fazer uma pesquisa em diversas fontes de informações técnicas para encontrarmos tendências nas características da aeronave proposta.

Com a pesquisa pronta, um pequeno código em Python3 deverá ser escrito para ajudar a verificar essas tendências.

## 2 Informações Mínimas Necessárias

Está sendo fornecida uma tabela com algumas informações mínimas a serem buscadas. Essa tabela é um bom ponto de partida, mas informações extras encontradas podem ser acrescentadas a ela. Essa atividade servirá como uma fonte de informações futuras, nos diversos Labs que vocês farão.

Lembrem-se que aeronaves podem ter múltiplas versões, variando pequenas coisas entre elas. Como exemplo, veja duas versões do A320.



(a) Lufthansa A320 - Foto por Julian Herzog



(b) Air India A320 - Foto por Jakkrit Prasertwit

Figura 1: Note o trem de pouso principal em cada aeronave

A versão da Air India possui quatro rodas de cada lado do trem de pouso principal (tipo *double-bogey*). Isso foi solicitado pela empresa à Airbus para minimizar problemas que possam ocorrer devido à baixa qualidade das pistas de pouso e decolagem da Índia, à época.

Cabe a cada um (e à equipe) decidir qual versão utilizar. Versões muito específicas (como esse A320 da Air India) não devem ser favorecidas em detrimento de outras, mais comuns, exceto se a especificidade for encontrada, também, nos requisitos do seu projeto.

### 3 Fontes de Informação

Existem muitas fontes de informação para essa pesquisa. Dentre as mais indicadas, podemos citar:

1. Wikipédia (útil para procurar aeronaves relacionadas, **atentar** para as fontes das informações)
2. Site do fabricante
3. [Plan 3-vues](#) - Útil para figuras razoavelmente confiáveis com 3 vistas de aeronaves
4. Livros diversos

É importante manter algum registro de qual fonte forneceu qual informação para cada aeronave, especialmente quando encontrarem informações potencialmente conflitantes.

**Deverá ser entregue uma planilha por grupo com as informações pedidas**

### 4 Código para regressão

Com a planilha preenchida, o grupo deverá desenvolver um código em Python3 para verificar as tendências das características das aeronaves.

No relatório deverão constar, no mínimo, plots das regressões para:  $\text{Peso vazio} \times \text{MTOW}$ ,  $\text{Número de passageiros} \times \text{MTOW}$  e  $\text{Número de passageiros} \times \text{Comprimento da fuselagem}$ . Plots adicionais que a equipe julgar úteis também devem ser colocados no relatório. A qualidade dos plots (**conteúdo** e **forma**) será levada em conta na avaliação.

Para isso, utilizaremos o *package pandas*. A seguir, um exemplo de utilização é fornecido.

```
1 '''
2 This script loads an Excel spreadsheet using the pandas package, uses the numpy
3 package to find trends and plots with the matplotlib package
4 '''
5
6 # Imports
7 import numpy as np
8 import pandas as pd
9 import matplotlib.pyplot as plt
10
11 # File read
12 df = pd.read_excel('notas.xlsx')
```

```

13
14 # Print the names of the columns
15 print("Column Headings: ")
16 print(df.columns)
17 print("-----")
18
19 # Print the names of the students (column 'Aluno')
20 print("Students: ")
21 print(df['Aluno'])
22 print("-----")
23
24 # Print the 5 top rows
25 print('Printing the 5 top rows')
26 print(df.head())
27 print("-----")
28
29 # Print the 6 top rows
30 print('Printing the 6 top rows')
31 print(df.head(6))
32 print("-----")
33
34 # Print the 3 last rows
35 print('Printing the 3 last rows')
36 print(df.tail(3))
37 print("-----")
38
39 # Find the averave grade for 'Prova' and print it
40 nota_media = df['Prova'].mean()
41 print('The average grade was ' + str(nota_media))
42 print("-----")
43
44 # Use "describe" to view some statistical analysis
45 print('Some statistics...')
46 print(df.describe())
47 print("-----")
48
49 # Save "Prova" to a list
50 prova = df['Prova'].values
51 print('Printing prova as a list')
52 print(prova)
53 print("-----")
54
55 # Plot histogram for "Trabalho"
56 plt.figure()
57 df['Faltas'].plot.hist(bins=5)
58 plt.show()
59
60
61 print("-----")
62 print("-----")
63 print("-----")
64 # Linear Regression
65
66 # Create X and Y based on the grades for the 'Prova' and 'Apresentacao'
67 X = df['Prova'].values
68 Y = df['Apresentacao'].values
69
70 # Create fit
71 Z = np.polyfit(X, Y, 3)
72 # Z is an array with the coefficients. Z[0] is the highest order coefficient
73 # and Z[-1] is the independent coefficient
74
75 print('These are the coefficients:')
76 print(Z)
77
78 # Create an object with the polynomial

```

```

79 p = np.poly1d(Z) # np.poly1d receives a list of coefficients
80
81 # Print the polynomial
82 print('\n\nAnd this is the polynomial:\n')
83 print(p)
84 print('\nNote the exponents on the polynomial!')
85
86 # Predict values based on the X values - Sorted
87 Y_pred = p(np.sort(X))
88
89 # Create figure
90 plt.figure()
91
92 # Scatter plot the original values
93 plt.scatter(X, Y)
94
95 # Add the predicted values as a line - Sorted X values
96 plt.plot(np.sort(X), Y_pred, color='red')
97
98 # Now let's create a different array for plotting the polynomial with more
99 # points.
100
101 X_new = np.linspace(7.5, 10, num=50)
102 Y_new = p(X_new)
103
104 # Add the new data points to the existing plot as a green dashed line
105 plt.plot(X_new, Y_new, color='green', linestyle='dashed')
106
107 # Show plot
108 plt.show()

```