**02 - Estruturas de dados**

Chegamos à última seção do nosso treinamento, na qual, como prometido, conversaremos sobre outra biblioteca muito importante para a ciência de dados: o Pandas. No notebook do curso deixamos um pequeno texto introdutório sobre essa ferramenta:

Pandas é uma ferramenta de manipulação de dados de alto nível, construída com base no pacote Numpy. O pacote pandas possui estruturas de dados bastante interessantes para manipulação de dados e por isso é muito utilizado por cientistas de dados.

Dentre essas estruturas de dados, temos por exemplo as ***series***, arrays unidimensionais rotulados capazes de armazenar qualquer tipo de dado. Os rótulos das linhas de uma *series* são conhecidos como **index**, e temos abaixo a forma básica de criação dessa estrutura:

s = pd.Series(dados, index = index)COPIAR CÓDIGO

Aqui estamos chamando a biblioteca Pandas por meio de seu apelido pd seguido de Series(), para o qual passamos os dados e um index (quando necessário). Esse argumento dados pode ser um dicionário, uma lista, um array Numpy ou uma constante.

Também temos o ***Dataframe***, uma estrutura de dados tabular bidimensional com rótulos nas linha e colunas. Assim como as *series*, os dataframes são capazes de armazenar qualquer tipo de dado.

df = pd.DataFrame(dados, index = index, columns = columns)COPIAR CÓDIGO

Para a criação de um *dataframe*, passamos para a chamada de pd.DataFrame(), além dos nossos dados, os rótulos das linhas (index) e das colunas (columns). Nossos dados, mais uma vez, podem ser um dicionário, uma lista, um array Numpy, uma *series* ou mesmo outro dataframe.

[Documentação da versão 0.25.2 do Pandas](https://pandas.pydata.org/pandas-docs/version/0.25/)

Antes de começarmos a trabalhar com essas estruturas, assim como fizemos com o Numpy, precisamos importar a biblioteca Pandas, o que é feito com a instrução import pandas, que pode ser sucedida de as pd, definindo um apelido a ela. Como a maior parte da comunidade utiliza esse apelido, é bastante comum encontrar esse tipo de importação.

import pandas as pdCOPIAR CÓDIGO

A primeira forma de criarmos uma *series* é a partir de uma lista. Para esse teste, inicializaremos a lista carros:

carros = ['Jetta Variant', 'Passat', 'Crossfox']

carrosCOPIAR CÓDIGO

['Jetta Variant', 'Passat', 'Crossfox']

Como visto anteriormente, chamaremos pd.Series() passando a nossa lista como argumento.

pd.Series(carros)COPIAR CÓDIGO

Como resultado, teremos a nossa *series*, com o índice de cada linha (0, 1 e 2) e os nomes dos carros. Note que o tipo desses valores é "object", que é como o Pandas trabalha com strings.

0 Jetta Variant 1 Passat 2 Crossfox dtype: object

Prosseguiremos para a criação de um *dataframe*, algo que faremos a partir de uma lista de dicionários. Não chegamos a abordar essa construção, mas ficou evidente no curso anterior que as listas podem conter quaisquer tipos de dados, inclusive outras listas ou dicionários.

Para nosso exemplo, inicializaremos a variável dados como uma lista na qual cada item é um dicionário contendo as informações de um veículo.

dados = [

{'Nome': 'Jetta Variant', 'Motor': 'Motor 4.0 Turbo', 'Ano': 2003, 'Quilometragem': 44410.0, 'Zero\_km': False, 'Valor': 88078.64},

{'Nome': 'Passat', 'Motor': 'Motor Diesel', 'Ano': 1991, 'Quilometragem': 5712.0, 'Zero\_km': False, 'Valor': 106161.94},

{'Nome': 'Crossfox', 'Motor': 'Motor Diesel V8', 'Ano': 1990, 'Quilometragem': 37123.0, 'Zero\_km': False, 'Valor': 72832.16}

]COPIAR CÓDIGO

Criaremos então uma variável dataset que receberá a chamada de pd.DataFrame() passando como parâmetro nossa lista dados. Note que em DataFrame() ambas as consoantes "D" e "F" devem ser maiúsculas, do contrário receberemos um erro.

dataset = pd.DataFrame(dados)

datasetCOPIAR CÓDIGO

Ao exibirmos o conteúdo de dataset, receberemos:

|  | **Nome** | **Motor** | **Ano** | **Quilometragem** | **Zero\_km** | **Valor** |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | Jetta Variant | Motor 4.0 Turbo | 2003 | 44410.0 | False | 88078.64 |
| 1 | Passat | Motor Diesel | 1991 | 5712.0 | False | 106161.94 |
| 2 | Crossfox | Motor Diesel V8 | 1990 | 37123.0 | False | 72832.16 |

Esse é o nosso dataframe. Note que não passamos os argumentos index ou columns; o index foi atribuído automaticamente com o padrão numérico, e as chaves de cada valor foram atribuídas aos rótulos das colunas.

Se quisermos modificar a ordem das colunas desse dataframe, basta passarmos o seu nome (dataset) seguido da abertura de dois pares de colchetes ([[]]) dentro dos quais passaremos nossos rótulos na ordem desejada.

dataset[['Nome', 'Motor', 'Ano', 'Quilometragem', 'Zero\_km', 'Valor']]COPIAR CÓDIGO

|  | **Nome** | **Motor** | **Ano** | **Quilometragem** | **Zero\_km** | **Valor** |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | Jetta Variant | Motor 4.0 Turbo | 2003 | 44410.0 | False | 88078.64 |
| 1 | Passat | Motor Diesel | 1991 | 5712.0 | False | 106161.94 |
| 2 | Crossfox | Motor Diesel V8 | 1990 | 37123.0 | False | 72832.16 |

Existem outras formas de criarmos *dataframes* a partir de dicionários. Para esse exemplo, inicializaremos o dicionário dados abaixo:

dados = {

'Nome': ['Jetta Variant', 'Passat', 'Crossfox'],

'Motor': ['Motor 4.0 Turbo', 'Motor Diesel', 'Motor Diesel V8'],

'Ano': [2003, 1991, 1990],

'Quilometragem': [44410.0, 5712.0, 37123.0],

'Zero\_km': [False, False, False],

'Valor': [88078.64, 106161.94, 72832.16]

}COPIAR CÓDIGO

Agora estamos trabalhando com um dicionário no qual as chaves são os rótulos dos dados e os valores são uma lista com todas as informações, como nomes, motores, anos e assim por diante, todos seguindo a mesma ordem. Executando novamente pd.DataFrame(dados), criaremos nosso novo dataset.

dataset = pd.DataFrame(dados)COPIAR CÓDIGO

|  | **Nome** | **Motor** | **Ano** | **Quilometragem** | **Zero\_km** | **Valor** |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | Jetta Variant | Motor 4.0 Turbo | 2003 | 44410.0 | False | 88078.64 |
| 1 | Passat | Motor Diesel | 1991 | 5712.0 | False | 106161.94 |
| 2 | Crossfox | Motor Diesel V8 | 1990 | 37123.0 | False | 72832.16 |

No [início do curso](https://cursos.alura.com.br/course/python-funcoes-pacotes-pandas/task/66364) você deve ter feito o download do projeto inicial, incluindo o arquivo db.csv, que subiremos para o notebook usando a aba "Upload". Usaremos esse arquivo para criarmos um dataframe a partir de dados externos. Na realidade, essa é a forma mais comum de criar dataframes.

Feito o *upload*, usaremos a função pd.read\_csv() para lermos o arquivo. Lembre-se que devemos passar como argumento, além do nome do arquivo (db.csv), um parâmetro sep que define o separador utilizado nesse conjunto - no nosso caso, ;.

Se você abrir o arquivo db.csv em um editor de texto, poderá verificar que os dados realmente estão separados por ponto e vírgula.

dataset = pd.read\_csv('db.csv', sep = ';')

datasetCOPIAR CÓDIGO

|  | **Nome** | **Motor** | **Ano** | **Quilometragem** | **Zero\_km** | **Acessórios** | **Valor** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | Jetta Variant | Motor 4.0 Turbo | 2003 | 44410.0 | False | ['Rodas de liga', 'Travas elétricas', 'Piloto ... | 88078.64 |
| 1 | Passat | Motor Diesel | 1991 | 5712.0 | False | ['Central multimídia', 'Teto panorâmico', 'Fre... | 106161.94 |
| 2 | Crossfox | Motor Diesel V8 | 1990 | 37123.0 | False | ['Piloto automático', 'Controle de estabilidad... | 72832.16 |
| 3 | DS5 | Motor 2.4 Turbo | 2019 | NaN | True | ['Travas elétricas', '4 X 4', 'Vidros elétrico... | 124549.07 |
| 4 | Aston Martin DB4 | Motor 2.4 Turbo | 2006 | 25757.0 | False | ['Rodas de liga', '4 X 4', 'Central multimídia... | 92612.10 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 253 | Phantom 2013 | Motor V8 | 2014 | 27505.0 | False | ['Controle de estabilidade', 'Piloto automátic... | 51759.58 |
| 254 | Cadillac Ciel concept | Motor V8 | 1991 | 29981.0 | False | ['Bancos de couro', 'Painel digital', 'Sensor ... | 51667.06 |
| 255 | Classe GLK | Motor 5.0 V8 Bi-Turbo | 2002 | 52637.0 | False | ['Rodas de liga', 'Controle de tração', 'Câmbi... | 68934.03 |
| 256 | Aston Martin DB5 | Motor Diesel | 1996 | 7685.0 | False | ['Ar condicionado', '4 X 4', 'Câmbio automátic... | 122110.90 |
| 257 | Macan | Motor Diesel V6 | 1992 | 50188.0 | False | ['Central multimídia', 'Teto panorâmico', 'Vid... | 90381.47 |

Ao importarmos um arquivo com a função pd.read\_csv(), também podemos passar um parâmetro index\_col = 0, definindo a coluna que desejamos assumir como índice do dataframe.

dataset = pd.read\_csv('db.csv', sep = ';', index\_col = 0)

datasetCOPIAR CÓDIGO

Note que, ao invés do índice numérico, passaremos a assumir a variável Nome como índice.

| **Nome** | **Motor** | **Ano** | **Quilometragem** | **Zero\_km** | **Acessórios** | **Valor** |
| --- | --- | --- | --- | --- | --- | --- |
| Jetta Variant | Motor 4.0 Turbo | 2003 | 44410.0 | False | ['Rodas de liga', 'Travas elétricas', 'Piloto ... | 88078.64 |
| Passat | Motor Diesel | 1991 | 5712.0 | False | ['Central multimídia', 'Teto panorâmico', 'Fre... | 106161.94 |
| Crossfox | Motor Diesel V8 | 1990 | 37123.0 | False | ['Piloto automático', 'Controle de estabilidad... | 72832.16 |
| DS5 | Motor 2.4 Turbo | 2019 | NaN | True | ['Travas elétricas', '4 X 4', 'Vidros elétrico... | 124549.07 |
| Aston Martin DB4 | Motor 2.4 Turbo | 2006 | 25757.0 | False | ['Rodas de liga', '4 X 4', 'Central multimídia... | 92612.10 |
| ... | ... | ... | ... | ... | ... | ... |
| Phantom 2013 | Motor V8 | 2014 | 27505.0 | False | ['Controle de estabilidade', 'Piloto automátic... | 51759.58 |
| Cadillac Ciel concept | Motor V8 | 1991 | 29981.0 | False | ['Bancos de couro', 'Painel digital', 'Sensor ... | 51667.06 |
| Classe GLK | Motor 5.0 V8 Bi-Turbo | 2002 | 52637.0 | False | ['Rodas de liga', 'Controle de tração', 'Câmbi... | 68934.03 |
| Aston Martin DB5 | Motor Diesel | 1996 | 7685.0 | False | ['Ar condicionado', '4 X 4', 'Câmbio automátic... | 122110.90 |
| Macan | Motor Diesel V6 | 1992 | 50188.0 | False | ['Central multimídia', 'Teto panorâmico', 'Vid... | 90381.47 |

Podemos inclusive fazer acessos a partir desse índice, que é um rótulo no formato de texto, e aprenderemos a fazer isso nos próximos vídeos. O índice numérico continua existindo, mas está "embutido" e deixa de ser exibido.

Conhecemos então algumas estruturas de dados muito importantes do Pandas, e que servem de contêiner para outras bibliotecas de *data science*, como Scikit-learn e Statsmodels. No próximo vídeo começaremos a fazer seleções em um dataframe.