**10 - Iterando com DataFrames**

Nesse vídeo aprenderemos uma forma de fazer iterações em dataframes. A forma que aprenderemos não tem um desempenho muito bom em dataframes muito extensos, e na realidade existem formas mais sofisticadas, entretanto, como estamos em um curso introdutório, preferimos abordar um conteúdo mais básico.

Continuaremos trabalhando com o dataset que importamos nos vídeos anteriores.

dataset.head()COPIAR CÓDIGO

| **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 |

Uma maneira simples de iterarmos sobre esses dados é utilizando a estrutura do for que já conhecemos.

for item in dataset:

print(item)COPIAR CÓDIGO

A devolução dessa iteração consistirá nos rótulos das nossas colunas.

Motor Ano Quilometragem Zero\_km Acessórios Valor

Porém, isso não tem muita utilidade para nós. O Pandas nos disponibiliza uma ferramenta interessante chamada iterrows(), que nos ajuda a criar um iterador. No exemplo abaixo, criaremos uma lista a partir desse iterador.

list(dataset.iterrows())COPIAR CÓDIGO

[('Jetta Variant', Motor Motor 4.0 Turbo Ano 2003 Quilometragem 44410 Zero\_km False Acessórios ['Rodas de liga', 'Travas elétricas', 'Piloto ... Valor 88078.6 Name: Jetta Variant, dtype: object), ('Passat', Motor Motor Diesel Ano 1991 Quilometragem 5712 Zero\_km False Acessórios ['Central multimídia', 'Teto panorâmico', 'Fre... Valor 106162 Name: Passat, dtype: object), ...

O resultado disso será uma lista contendo diversas tuplas, cada uma composta pelo índice do nosso dataframe (os nomes dos veículos) e uma *series* com as informações da linha correspondente a esse índice. Isso nos permitirá acessar tais informações.

Como nosso dataset.iterrows() é um iterador, podemos utilizá-lo em conjunto com o for para desempacotarmos tais tuplas em dados específicos, por exemplo index e row, ou seja, os índices e informações das linhas.

for index, row in dataset.iterrows():COPIAR CÓDIGO

No nosso primeiro treinamento, fizemos o cálculo da quilometragem média dos veículos do conjunto db.csv a partir da quilometragem total e o ano de fabricação desses veículos. Faremos isso novamente, dessa vez criando uma nova coluna no nosso dataframe para armazenar esses valores.

Como não queremos receber um erro de divisão por 0, algo que pode acontecer quando nos deparamos com um carro que foi fabricado em 2019 (ano de gravação deste curso), precisaremos tratar as informações que utilizamos nesse cálculo. Para isso, incluiremos no nosso for uma condicional if que verifica se 2019 - row['Ano'] (a subtração de 2019 pelo ano de fabricação do veículo) é diferente de 0.

Em caso positivo, utilizaremos o .loc para criarmos uma nova coluna Km\_media em nosso dataset. A essa coluna atribuiremos o valor da quilometragem média, que conseguiremos com o cálculo row['Quilometragem'] / 2019 - row['Ano'] (a quilometragem total do veículo dividida pela subtração de 2019 pelo ano de fabricação deste veículo).

Já em caso negativo, simplesmente atribuiremos 0 como valor da linha na nova coluna.

for index, row in dataset.iterrows():

if(2019 - row['Ano'] != 0):

dataset.loc[index, 'Km\_media'] = row['Quilometragem'] / (2019 - row['Ano'])

else:

dataset.loc[index, 'Km\_media'] = 0

datasetCOPIAR CÓDIGO

Nosso dataframe será então atualizado com as novas informações. Como o conjunto é relativamente pequeno, essa operação se dará rapidamente.

| **Nome** | **Motor** | **Ano** | **Quilometragem** | **Zero\_km** | **Acessórios** | **Valor** | **Km\_media** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Jetta Variant | Motor 4.0 Turbo | 2003 | 44410.0 | False | ['Rodas de liga', 'Travas elétricas', 'Piloto ... | 88078.64 | 2775.625000 |
| Passat | Motor Diesel | 1991 | 5712.0 | False | ['Central multimídia', 'Teto panorâmico', 'Fre... | 106161.94 | 204.000000 |
| Crossfox | Motor Diesel V8 | 1990 | 37123.0 | False | ['Piloto automático', 'Controle de estabilidad... | 72832.16 | 1280.103448 |
| DS5 | Motor 2.4 Turbo | 2019 | NaN | True | ['Travas elétricas', '4 X 4', 'Vidros elétrico... | 124549.07 | 0.000000 |
| Aston Martin DB4 | Motor 2.4 Turbo | 2006 | 25757.0 | False | ['Rodas de liga', '4 X 4', 'Central multimídia... | 92612.10 | 1981.307692 |

Note que carros fabricados em 2019 figuram com 0 como valor de Km\_media. Assim conseguimos obter facilmente os quilômetros rodados de cada veículo do nosso conjunto, e para isso utilizamos as iterações em dataframes. Como informamos anteriormente, existem formas mais sofisticadas de fazer isso, e que seriam úteis principalmente se estivéssemos lidando com dataframes maiores. Você pode buscar informações sobre elas em cursos mais avançados de ciência de dados e/ou Pandas.

No próximo vídeo aprenderemos a fazer alguns tratamentos nos nossos dados.