**03 - Lendo nomes com Json**

Nos treinamentos anteriores, geralmente disponibilizávamos um arquivo .csv para realizarmos nossas análises. Porém, a biblioteca Pandas também nos permite importar e exportar diferentes tipos de arquivos, que é o que faremos nesse treinamento.

Começaremos analisando as informações de uma escola de programação que possui muitos alunos e alunas em diferentes cursos. Abriremos o [Google Colaboratory](https://colab.research.google.com/) e criaremos um novo notebook Python 3, que renomearemos para Pandas IO.ipynb.

Na atividade anterior, "Preparando o ambiente", temos um passo a passo do que é necessário para utilizar o Colaboratory, uma ferramenta disponibilidade pelo Google para programação em Python.

Repare que até o momento não temos nenhum arquivo para trabalhar, afinal não fizemos nenhum upload. De que outra forma podemos trazer as informações de nomes dos nossos alunos e alunas? Uma alternativa para criação desses nomes é procurarmos no Google uma API do IBGE contendo tais dados.

Nosso primeiro resultado será ["IBGE - API de serviço de dados"](https://servicodados.ibge.gov.br/api/docs/censos/nomes?versao=2), que nos levará direto a uma API com informações coletadas no censo de 2010, divididas em "Frequência por nome" e "Ranking por frequência". Acessaremos esse segundo conjunto clicando no link:

<https://servicodados.ibge.gov.br/api/v2/censos/nomes/ranking>

Receberemos no navegador uma lista de nomes (como "JOSE" e "MARIA") no formato JSON. Se aleterarmos o link para a versão 1 do conjunto (v1), teremos uma visualização mais facilitada.

<https://servicodados.ibge.gov.br/api/v1/censos/nomes/ranking>

Aqui identificaremos nomes como "ANA", "ANTONIO" e assim por diante. Também podemos filtrar mais as nossas informações. Após "ranking", adicionaremos uma interrogação (?) e o parâmetro qtd=20. Além disso, usaremos o caractere & (conhecido como "e comercial") para passarmos mais um parâmetro, sexo=f.

<https://servicodados.ibge.gov.br/api/v1/censos/nomes/ranking?qtd=20&sexo=f>

Nosso resultado será uma lista contendo 20 nomes, todos eles do sexo feminino. Se alterarmos o último parâmetro para sexo=m, conseguiremos os nomes de sexo masculino.

Voltando ao Colaboratory, importaremos o Pandas em nosso projeto com o conhecido comando import pandas as pd (utilizando o apelido convencional pd).

import pandas as pdCOPIAR CÓDIGO

Agora gostaríamos de ler a API contendo os nomes masculinos e femininos. Para isso, chamaremos a função pd.read\_json() - ou seja, uma função que lê arquivos JSON. Como parâmetro, entre aspas duplas, passaremos a URL dos nomes de sexo feminino.

pd.read\_json("https://servicodados.ibge.gov.br/api/v1/censos/nomes/ranking?qtd=20&sexo=f")COPIAR CÓDIGO

Ao executarmos, receberemos uma tabela contendo os 20 nomes da nossa consulta - ou seja, do índice 0 ao 19.

|  | **nome** | **regiao** | **freq** | **rank** | **sexo** |
| --- | --- | --- | --- | --- | --- |
| 0 | MARIA | 0 | 11694738 | 1 | F |
| 1 | ANA | 0 | 3079729 | 2 | F |
| 2 | FRANCISCA | 0 | 721637 | 3 | F |
| 3 | ANTONIA | 0 | 588783 | 4 | F |
| 4 | ADRIANA | 0 | 565621 | 5 | F |
| 5 | JULIANA | 0 | 562589 | 6 | F |
| 6 | MARCIA | 0 | 551855 | 7 | F |
| 7 | FERNANDA | 0 | 531607 | 8 | F |
| 8 | PATRICIA | 0 | 529446 | 9 | F |
| 9 | ALINE | 0 | 509869 | 10 | F |
| 10 | SANDRA | 0 | 479230 | 11 | F |
| 11 | CAMILA | 0 | 469851 | 12 | F |
| 12 | AMANDA | 0 | 464624 | 13 | F |
| 13 | BRUNA | 0 | 460770 | 14 | F |
| 14 | JESSICA | 0 | 456472 | 15 | F |
| 15 | LETICIA | 0 | 434056 | 16 | F |
| 16 | JULIA | 0 | 430067 | 17 | F |
| 17 | LUCIANA | 0 | 429769 | 18 | F |
| 18 | VANESSA | 0 | 417512 | 19 | F |
| 19 | MARIANA | 0 | 381778 | 20 | F |

Se alterarmos o último parâmetro para sexo=m, teremos os 20 nomes masculinos.

pd.read\_json("https://servicodados.ibge.gov.br/api/v1/censos/nomes/ranking?qtd=20&sexo=m")COPIAR CÓDIGO

|  | **nome** | **regiao** | **freq** | **rank** | **sexo** |
| --- | --- | --- | --- | --- | --- |
| 0 | JOSE | 0 | 5732508 | 1 | M |
| 1 | JOAO | 0 | 2971935 | 2 | M |
| 2 | ANTONIO | 0 | 2567494 | 3 | M |
| 3 | FRANCISCO | 0 | 1765197 | 4 | M |
| 4 | CARLOS | 0 | 1483121 | 5 | M |
| 5 | PAULO | 0 | 1417907 | 6 | M |
| 6 | PEDRO | 0 | 1213557 | 7 | M |
| 7 | LUCAS | 0 | 1116818 | 8 | M |
| 8 | LUIZ | 0 | 1102927 | 9 | M |
| 9 | MARCOS | 0 | 1101126 | 10 | M |
| 10 | LUIS | 0 | 931530 | 11 | M |
| 11 | GABRIEL | 0 | 922744 | 12 | M |
| 12 | RAFAEL | 0 | 814709 | 13 | M |
| 13 | DANIEL | 0 | 706527 | 14 | M |
| 14 | MARCELO | 0 | 690098 | 15 | M |
| 15 | BRUNO | 0 | 663271 | 16 | M |
| 16 | EDUARDO | 0 | 628539 | 17 | M |
| 17 | FELIPE | 0 | 615924 | 18 | M |
| 18 | RAIMUNDO | 0 | 611174 | 19 | M |
| 19 | RODRIGO | 0 | 598825 | 20 | M |

Claro, não queremos ter que rodar essas linhas de código toda vez que precisarmos visualizar nossos alunos e alunas. Sendo assim, criaremos as variáveis nomes\_m e nomes\_f armazenando os nomes masculinos e femininos, respectivamente.

nomes\_f = pd.read\_json("https://servicodados.ibge.gov.br/api/v1/censos/nomes/ranking?qtd=20&sexo=f")

nomes\_m = pd.read\_json("https://servicodados.ibge.gov.br/api/v1/censos/nomes/ranking?qtd=20&sexo=m")COPIAR CÓDIGO

Se quisermos, podemos visualizar nossos nomes a partir de cada variável.

nomes\_fCOPIAR CÓDIGO

|  | **nome** | **regiao** | **freq** | **rank** | **sexo** |
| --- | --- | --- | --- | --- | --- |
| 0 | MARIA | 0 | 11694738 | 1 | F |
| 1 | ANA | 0 | 3079729 | 2 | F |
| 2 | FRANCISCA | 0 | 721637 | 3 | F |
| 3 | ANTONIA | 0 | 588783 | 4 | F |
| 4 | ADRIANA | 0 | 565621 | 5 | F |
| 5 | JULIANA | 0 | 562589 | 6 | F |
| 6 | MARCIA | 0 | 551855 | 7 | F |
| 7 | FERNANDA | 0 | 531607 | 8 | F |
| 8 | PATRICIA | 0 | 529446 | 9 | F |
| 9 | ALINE | 0 | 509869 | 10 | F |
| 10 | SANDRA | 0 | 479230 | 11 | F |
| 11 | CAMILA | 0 | 469851 | 12 | F |
| 12 | AMANDA | 0 | 464624 | 13 | F |
| 13 | BRUNA | 0 | 460770 | 14 | F |
| 14 | JESSICA | 0 | 456472 | 15 | F |
| 15 | LETICIA | 0 | 434056 | 16 | F |
| 16 | JULIA | 0 | 430067 | 17 | F |
| 17 | LUCIANA | 0 | 429769 | 18 | F |
| 18 | VANESSA | 0 | 417512 | 19 | F |
| 19 | MARIANA | 0 | 381778 | 20 | F |

Considerando ambos os conjuntos, temos 40 nomes, o que é pouco pensando no escopo de uma grande escola de programação. Portanto, ao invés de apenas 20 nomes de cada sexo, passaremos a utilizar 200.

nomes\_f = pd.read\_json("https://servicodados.ibge.gov.br/api/v1/censos/nomes/ranking?qtd=200&sexo=f")

nomes\_m = pd.read\_json("https://servicodados.ibge.gov.br/api/v1/censos/nomes/ranking?qtd=200&sexo=m")COPIAR CÓDIGO

nomes\_mCOPIAR CÓDIGO

Com isso, o índice das nossas tabelas passará a registrar de 0 a 199, totalizando 200 nomes masculinos e 200 femininos.

|  | **nome** | **regiao** | **freq** | **rank** | **sexo** |
| --- | --- | --- | --- | --- | --- |
| 0 | JOSE | 0 | 5732508 | 1 | M |
| 1 | JOAO | 0 | 2971935 | 2 | M |
| 2 | ANTONIO | 0 | 2567494 | 3 | M |
| 3 | FRANCISCO | 0 | 1765197 | 4 | M |
| 4 | CARLOS | 0 | 1483121 | 5 | M |
| ... | ... | ... | ... | ... | ... |
| 195 | WALTER | 0 | 67207 | 196 | M |
| 196 | ARLINDO | 0 | 66669 | 197 | M |
| 197 | MICHAEL | 0 | 65736 | 198 | M |
| 198 | ALVARO | 0 | 65444 | 199 | M |
| 199 | GEOVANE | 0 | 65435 | 200 | M |

Podemos identificar a quantidade de nomes em nossos conjuntos imprimindo (print()) o tamanho de cada um deles. Para isso, concatenaremos a string "Quantidade de nomes: " com a soma len(nomes\_f) + len(nomes\_m)).

print("Quantidade de nomes: " + len(nomes\_f) + len(nomes\_m))COPIAR CÓDIGO

A execução dessa instrução resultará em um erro, afinal o retorno da soma é um número inteiro, e não uma string que possa ser concatenada com nosso texto. Corrigiremos isso utilizando a função conversora str().

print("Quantidade de nomes: " + str(len(nomes\_f) + len(nomes\_m)))COPIAR CÓDIGO

Quantidade de nomes: 400

Agora queremos juntar esses dois conjuntos em um único dataframe contendo os nomes de todos os alunos e alunas da escola. Para isso, criaremos uma variável frames que receberá uma lista contendo nossas informações.

frames = [nomes\_f, nomes\_m]COPIAR CÓDIGO

Em seguida, poderíamos usar a função pd.concat() do Pandas para concatenarmos os dados em um único dataframe. Entretanto, se observamos nossa lista frames, veremos que existem diversas colunas desnecessárias para o projeto, como freq (a frequência do nome no país), nem o rank (de *ranking*), a região ou o sexo.

Na função pd.concat(), é possível especificar quais colunas queremos concatenar, nesse caso apenas a nome. Ao final, usaremos a função to\_frame() para transformarmos o conjunto resultante em um dataframe.

pd.concat(frames)['nome'].to\_frame()COPIAR CÓDIGO

Como resultado, teremos:

|  | **nome** |
| --- | --- |
| 0 | MARIA |
| 1 | ANA |
| 2 | FRANCISCA |
| 3 | ANTONIA |
| 4 | ADRIANA |
| ... | ... |
| 195 | WALTER |
| 196 | ARLINDO |
| 197 | MICHAEL |
| 198 | ALVARO |
| 199 | GEOVANE |

Armazenaremos esse retorno em uma variável nomes:

nomes = pd.concat(frames)['nome'].to\_frame()COPIAR CÓDIGO

Com a função sample(), podemos conseguir um determinado número de amostras aleatórias dentro deste dataframe. Nesse caso, pediremos cinco:

nomes.sample(5)COPIAR CÓDIGO

|  | **nome** |
| --- | --- |
| 30 | ANDERSON |
| 5 | PAULO |
| 31 | RICARDO |
| 85 | FABRICIO |
| 17 | FELIPE |

Executando novamente, teremos outro resultado.

|  | **nome** |
| --- | --- |
| 156 | VALDEMAR |
| 178 | DENILSON |
| 13 | BRUNA |
| 187 | NAIARA |
| 82 | SOLANGE |

Com o comando pd.read\_json(), criamos um dataframe com 400 nomes, entre eles 200 femininos e 200 masculinos, que foram reunidos utilizando a função concat().