02

Selecionando os cursos

Transcrição

Todos os alunos cadastrados em nosso dataframe possuem `nome`, `id_aluno`, `email` e agora também uma quantidade de `matriculas`.

```
nomes.sample(5)
```

COPIAR CÓDIGO

	nome	id_aluno	dominio	email	matriculas
88	CAROLINA	96	@servicodoemail.com	carolina@servicodoemail.com	2
8	LUIZ	98	@servicodoemail.com	luiz@servicodoemail.com	2
127	TAINA	79	@servicodoemail.com	taina@servicodoemail.com	4
144	LORENA	91	@servicodoemail.com	lorena@servicodoemail.com	5
78	SILVIA	115	@servicodoemail.com	silvia@servicodoemail.com	1

Nessa amostra, por exemplo a "Silvia" está inscrita em `1` curso; a "Lorena" em `5`; "Taina" em `4`; e "Luiz" e "Carolina" em `2`. Porém, não temos uma tabela/dataframe mostrando quais cursos são esses. Nosso objetivo será justamente criar um dataframe que informe, por exemplo, que "Carolina" está inscrita nos cursos de Java e Python, que "Luiz" está inscrito nos cursos de C# e .NET, e assim por diante.

Claro, seria bastante cansativo fazermos isso manualmente, afinal precisaríamos escolher dois cursos para "Carolina", mais dois para "Luiz", mais 4 para "Tainá" e assim por diante contemplando todos os 400 alunos de nosso dataframe atual. A ideia, então, será criarmos o novo dataframe de forma aleatória.

Começaremos essa nova etapa do projeto criando a seção "Selecionando cursos". Para criarmos o novo dataframe, será necessário varrermos cada linha do conjunto `nomes` de modo a verificarmos quantas `matriculas` cada `nome` possui e, com essa informação, escolher a quantidade correta de `cursos`. Ou seja, teremos que criar um

loop que verificará, por exemplo, que a "Terezinha" possui `1` matrícula, e atribuirá um curso aleatório a ela; se "Marco" possui `2` matrículas, serão atribuídos dois cursos; e assim por diante.

Para fazermos essa distribuição, criaremos três variáveis. A primeira delas é `total_matriculas`, que será inicializada como um array vazio. Em seguida, criaremos uma variável `x` que receberá a chamada de `np.random.rand(20`, uma maneira de calcularmos randomicamente os 20 cursos que temos no dataframe. Por fim, teremos uma variável `prob` (de probabilidade) que receberá a divisão de `x` por `sum(x)` (a soma de `x`).

```
todas_matriculas = []
x = np.random.rand(20)
prob = x / sum(x)
```

COPIAR CÓDIGO

Depois de inicializarmos essas variáveis, a ideia é buscarmos em cada linha do dataframe o `nome`, o `id_aluno` e a quantidade de `matriculas` para então atribuirmos aos alunos a quantidade correta de cursos escolhidos aleatoriamente.

Passaremos para a criação de um iterador `for` que buscará o `index` e a linha que iremos utilizar, a qual chamaremos de `row`. Esse iterador percorrerá o dataframe `nomes` com o auxílio da função `iterrows()`, que nos devolve cada linha do conjunto.

```
for index, row in nomes.iterrows()
```

COPIAR CÓDIGO

A cada elemento encontrado, armazenaremos o `id` do aluno, conseguido com `row.id_aluno`, e a quantidade de `matriculas`, conseguida com `row.matriculas`.

```
for index, row in nomes.iterrows()
    id = row.id_aluno
    matriculas = row.matriculas
```

COPIAR CÓDIGO

De posse dessas informações, precisamos decidir quantos cursos terão que ser atribuídos a cada aluno com base na sua quantidade de matrículas. Para isso,

teremos um novo iterador `for` que buscará cada elemento (`i`) no intervalo `range(matriculas)` (de `0` até o valor de `matriculas`). Teremos então uma matrícula `mat` que receberá o `id` do aluno e o `id` do curso, que conseguiremos aleatoriamente utilizando a função `np.random.choice()` . Esta, por sua vez, receberá como parâmetros o índice de cursos (`cursos.index` e a variável de probabilidade que inicializamos anteriormente (`p = prob`).

```
for index, row in nomes.iterrows():
  id = row.id_aluno
  matriculas = row.matriculas
  for i in range(matriculas):
    mat = [id, np.random.choice(cursos.index, p = prob)]
```

COPIAR CÓDIGO

Prosseguiremos atribuindo ao array `todas_matriculas` , usando a função `append()` , as matrículas `mat` que estipulamos para o aluno.

```
for index, row in nomes.iterrows():
  id = row.id_aluno
  matriculas = row.matriculas
  for i in range(matriculas):
    mat = [id, np.random.choice(cursos.index, p = prob)]
    todas_matriculas.append(mat)
```

COPIAR CÓDIGO

Agora que rodamos todos os nomes selecionando os IDs de curso para cada matrícula, poderemos criar o dataframe `matriculas` usando a função `DataFrame()` do Pandas. Ela receberá o conteúdo de `todas_matriculas` , que distribuiremos nas colunas `id_aluno` e `id_curso` . Por fim, chamaremos `matriculas.head(5)` para verificarmos as cinco primeiras entradas deste dataframe.

```
for index, row in nomes.iterrows():
  id = row.id_aluno
  matriculas = row.matriculas
  for i in range(matriculas):
```

```
        mat = [id, np.random.choice(cursos.index, p = prob)]
        todas_matriculas.append(mat)

matriculas = pd.DataFrame(todas_matriculas, columns = ['id_aluno',
matriculas.head(5)
```

COPIAR CÓDIGO

	id_aluno	id_curso
0	235	15
1	235	4
2	43	6
3	43	4
4	43	10

Como resultado, temos um dataframe com o qual conseguimos saber em quais cursos cada aluno se inscreveu. Por exemplo, sabemos que o aluno `235` se inscreveu nos cursos `15` e `4`, e que o aluno `43` se inscreveu nos cursos `6`, `4` e `10`. Agrupando os dataframes `cursosatriculas`, conseguiremos inclusive saber quantos alunos estão matriculados em cada curso.

A partir do dataframe `matriculas`, chamaremos a função `groupby()` passando como parâmetro a coluna `id_curso`. Em seguida, chamaremos a função `count()` para contarmos as matrículas. Com a função `join()`, uniremos essa informação à tabela `cursos`, tomando como base a coluna `nome_do_curso`.

```
matriculas.groupby('id_curso').count().join(cursos['nome_do_curso']
```

COPIAR CÓDIGO

Como resultado, teremos um dataframe que nos mostra, além do ID e nome do curso, a quantidade de alunos em cada curso.

id_curso	id_aluno	nome_do_curso
1	14	Lógica de programação
2	2	Java para Web
3	47	C# para Web
4	34	Ruby on Rails
5	1	Cursos de Python
6	81	PHP com MySql
7	28	.NET para web
8	57	Novas integrações com Java
9	24	TDD com Java
10	57	Código limpo com C#
11	65	Preparatório para certificação Java
12	19	Hardware básico
13	65	Persistência com .NET
14	54	Desenvolvendo jogos
15	103	Análise de dados
16	34	Estatística básica
17	42	Internet das coisas
18	40	Programação funcional
19	26	Boas práticas em Java
20	20	Orientação objetos com Java

Entretanto, a coluna que se refere a essa informação está nomeada incorretamente como `id_aluno`. Corrigiremos isso usando `rename(columns={'id_aluno':'quantidade_de_alunos'}`.

```
matriculas.groupby('id_curso').count().join(cursos['nome_do_curso']
```

COPIAR CÓDIGO

id_curso	quantidade_de_alunos	nome_do_curso
1	14	Lógica de programação
2	2	Java para Web
3	47	C# para Web
4	34	Ruby on Rails
5	1	Cursos de Python
6	81	PHP com MySql
7	28	.NET para web
8	57	Novas integrações com Java
9	24	TDD com Java
10	57	Código limpo com C#
11	65	Preparatório para certificação Java
12	19	Hardware básico
13	65	Persistência com .NET
14	54	Desenvolvendo jogos
15	103	Análise de dados
16	34	Estatística básica
17	42	Internet das coisas
18	40	Programação funcional
19	26	Boas práticas em Java
20	20	Orientação objetos com Java

Vemos que existem `14` alunos matriculados em "Lógica de programação", `2` alunos em "Java para Web", `47` em "C# para Web" e assim por diante. Todos esses valores foram gerados de forma aleatória, poupando nosso tempo. Também devemos nos lembrar que um aluno pode se inscrever em mais de curso.

Até o momento, criamos 3 dataframes, começando pelo `nomes`.

```
nomes.sample(3)
```

COPIAR CÓDIGO

	nome	id_aluno	dominio	email	matriculas
140	NEUZA	23	@servicodoemail.com	neuza@servicodoemail.com	2
198	ISADORA	38	@dominioemail.com.br	isadora@dominioemail.com.br	2
190	NATANAEL	58	@servicodoemail.com	natanael@servicodoemail.com	2

Criamos também o dataframe `cursos`.

```
cursos.head()
```

COPIAR CÓDIGO

id	nome_do_curso
1	Lógica de programação
2	Java para Web
3	C# para Web
4	Ruby on Rails
5	Cursos de Python

E a tabela de `matriculas`, na qual é possível identificar o ID dos alunos e o ID dos cursos em que estão matriculados.

```
matriculas.head()
```

COPIAR CÓDIGO

	id_aluno	id_curso
0	235	15
1	235	4
2	43	6
3	43	4
4	43	10

Ao final, fizemos um agrupamento, que podemos chamar de `matriculas_por_curso`.

```
matriculas_por_curso = matriculas.groupby('id_curso').count().join(
```

COPIAR CÓDIGO

Ao executarmos `matriculas_por_curso` , conseguiremos visualizar a quantidade de alunos inscritos em cada curso.

```
matriculas_por_curso.head()
```

COPIAR CÓDIGO

id_curso	quantidade_de_alunos	nome_do_curso
1	14	Lógica de programação
2	2	Java para Web
3	47	C# para Web
4	34	Ruby on Rails
5	1	Cursos de Python

Lembrando que tudo que fizemos até o momento não foi simplesmente ler informações a partir de um JSON ou HTML, nós também trabalhamos os dados recebidos de modo a torná-los mais interessantes e úteis para nosso projeto.