**01 - Matriculando alunas e alunos**

Como resultado das aulas anteriores, temos uma lista de cursos da nossa escola e outra de nomes dos alunos e alunas matriculados. Geralmente, em escolas desse tipo, temos alguns alunos que fazem um curso, menos que fazem dois cursos, menos ainda que fazem 3 ou mais, e assim por diante. Ou seja, poucos alunos fazem muitos cursos, algo que queremos refletir em nosso projeto.

A ideia é criarmos em nosso dataframe nomes a quantidade de matrículas que cada aluno tem. Mantendo a organização, teremos uma nova seção "Matriculando os alunos nos cursos".

Criaremos uma nova coluna nomes['matriculas'] que receberá a chamada de np.random.exponential(), uma função exponencial da biblioteca Numpy. Passaremos para ela o parâmetro size=total\_alunos. Em seguida, usarmeos astype(int) de modo a definirmos o tipo dessa coluna como inteiro, e pediremos uma amostra do novo dataframe com nomes.sample(5).

nomes['matriculas'] = np.random.exponential(size=total\_alunos).astype(int)

nomes.sample(5)COPIAR CÓDIGO

|  | **nome** | **id\_aluno** | **domínio** | **email** | **matriculas** |
| --- | --- | --- | --- | --- | --- |
| 15 | LETICIA | 347 | @domoniodoemail.com.br | leticia@domoniodoemail.com.br | 0 |
| 157 | JAIME | 214 | @servicodoemail.com | jaime@servicodoemail.com | 0 |
| 36 | ROBERTO | 364 | @domoniodoemail.com.br | roberto@domoniodoemail.com.br | 0 |
| 91 | CELIA | 367 | @domoniodoemail.com.br | celia@domoniodoemail.com.br | 1 |
| 183 | MARLON | 74 | @servicodoemail.com | marlon@servicodoemail.com | 0 |

Entretanto, tivemos matrículas com o valor mínimo 0, algo que não queremos no nosso projeto. Para ajustarmos isso, encobriremos a chamada de np.random.exponential() com a chamada da função np.ceil(), que retorna arredonda os valores para cima.

nomes['matriculas'] = np.ceil(np.random.exponential(size=total\_alunos)).astype(int)

nomes.sample(5)COPIAR CÓDIGO

|  | **nome** | **id\_aluno** | **dominio** | **email** | **matriculas** |
| --- | --- | --- | --- | --- | --- |
| 0 | JOSE | 331 | @servicodoemail.com | jose@servicodoemail.com | 1 |
| 145 | FRANCIELE | 316 | @dominiodoemmail.com.br | franciele@dominiodoemmail.com.br | 1 |
| 186 | INES | 319 | @servicodoemail.com | ines@servicodoemail.com | 1 |
| 66 | DANILO | 348 | @servicodoemail.com | danilo@servicodoemail.com | 2 |
| 163 | REGIANE | 308 | @servicodoemail.com | regiane@servicodoemail.com | 1 |

Também gostaríamos de aumentar o número de cursos nos quais os alunos e alunas estão matriculados. Pensando nisso, multiplicaremos o resultado do gerador de números randômicos por 1.5.

nomes['matriculas'] = np.ceil(np.random.exponential(size=total\_alunos) \* 1.5).astype(int)

nomes.sample(5)COPIAR CÓDIGO

|  | **nome** | **id\_aluno** | **dominio** | **email** | **matriculas** |
| --- | --- | --- | --- | --- | --- |
| 98 | GILSON | 134 | @servicodoemail.com | gilson@servicodoemail.com | 1 |
| 17 | FELIPE | 113 | @dominiodoemmail.com.br | felipe@servicodoemail.com | 5 |
| 129 | EDVALDO | 313 | @servicodoemail.com | edvaldo@servicodoemail.com | 3 |
| 137 | NAIR | 129 | @servicodoemail.com | nair@servicodoemail.com | 1 |
| 24 | CLAUDIA | 16 | @servicodoemail.com | claudia@servicodoemail.com | 4 |

Assim temos nossos alunos inscritos em mais cursos do que anteriormente. Nosso objetivo agora é descrevermos como ficou essa distribuição, algo que conseguiremos com nomes.matriculas.describe().

nomes.matriculas.describe()COPIAR CÓDIGO

A função describe() nos trará algumas informações, como a quantidade de matrículas (400), a média (2 cursos por pessoa) e um desvião padrão de 1.

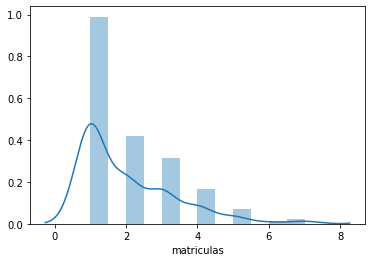
count 400.000000 mean 2.032000 std 1.389702 min 1.000000 25% 1.000000 50% 2.000000 75% 3.000000 max 11.000000 Name: matriculas, dtype: float64

Além disso, todos os alunos estão inscritos em pelo menos 1 curso, e o máximo de cursos em que um aluno está inscrito é 11. Para visualizarmos as informações em um gráfico, importaremos a biblioteca seaborn com o apelido convencional sns.

import seaborn as snsCOPIAR CÓDIGO

Em segiuda, usaremos a função sns.distplot() para exibirmos um histograma com a distribuição de nomes.matriculas.

sns.distplot(nomes.matriculas)COPIAR CÓDIGO



Como esperávamos, temos muitos alunos inscritos em um curso, menos inscritos em 2, menos ainda em 3 e assim por diante. Passaremos para uma visualização do número exato de alunos com cada quantidade de matrículas, que conseguiremos a partir da função value\_counts().

nomes.matriculas.value\_counts()

COPIAR CÓDIGO

1 191 2 102 3 57 4 24 5 15 6 7 7 2 11 1 9 1 Name: matriculas, dtype: int64

Dentro de nossa base contendo 400 nomes, 191 estão inscritos em apenas 1 curso; 102 em 2; 57 em 3, e assim por diante, incluindo 1 aluno(a) inscrito em 11 cursos (mais da metade da na nossa tabela). Temos aqui um cenário bem interessante e parecido com o que acontece na realidade!