

Prevendo o peso cerebral

Exercício:

Parte 1: Desenvolva o código necessário para a fórmula básica da regressão linear simples, calculando os coeficientes.

Parte 2: Use o modelo para fazer previsões.

O dataset abaixo contém dados sobre medidas da cabeça de seres humanos e o peso do cérebro. Seu trabalho é criar um modelo de regressão linear simples que preveja o peso do cérebro conforme seu tamanho.

```
In [1]: # Importando as bibliotecas
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from statsmodels.formula.api import ols
```

```
In [2]: # Carregando os dados
data = pd.read_csv('pesos2.csv')
```

```
In [3]: # Verificando as 5 primeiras linhas
data.head()
```

```
Out[3]:
```

	Sexo	Grupo	Head Size	Brain Weight
0	1	1	4512	1530
1	1	1	3738	1297
2	1	1	4261	1335
3	1	1	3777	1282
4	1	1	4177	1590

```
In [4]: # Verificando as 15 primeiras linhas
data.head(15)
```

```
Out[4]:
```

	Sexo	Grupo	Head Size	Brain Weight
0	1	1	4512	1530
1	1	1	3738	1297
2	1	1	4261	1335
3	1	1	3777	1282
4	1	1	4177	1590
5	1	1	3585	1300
6	1	1	3785	1400
7	1	1	3559	1255
8	1	1	3613	1355
9	1	1	3982	1375
10	1	1	3443	1340
11	1	1	3993	1380
12	1	1	3640	1355
13	1	1	4208	1522
14	1	1	3832	1208

```
In [5]: # Verificando as 5 últimas linhas
data.tail()
```

```
Out[5]:
```

	Sexo	Grupo	Head Size	Brain Weight
232	2	2	3214	1110
233	2	2	3394	1215
234	2	2	3233	1104
235	2	2	3352	1170
236	2	2	3391	1120

```
In [6]: #Verificando o número de linhas e colunas
data.shape
```

```
Out[6]: (237, 4)
```

Out[6]: (237, 4)

```
In [20]: #Verificar se há valores NAN (Retirar se houver)
data.isnull().sum()
```

```
Out[20]: Sexo      0
         Grupo      0
         Head Size  0
         Brain Weight 0
         dtype: int64
```

```
In [27]: #Retirando a coluna Grupo
data.drop('Grupo', axis = 1, inplace=True)
```

```
In [28]: data.head()
```

```
Out[28]:
```

	Sexo	Head Size	Brain Weight
0	1	4512	1530
1	1	3738	1297
2	1	4261	1335
3	1	3777	1282
4	1	4177	1590

```
In [100]: #Mudando o nome das colunas para o português
data.rename(columns= {'Head Size':'tamanho da cabeça','Brain Weight':'peso do cérebro'}, inplace=True)
data.head()
```

```
Out[100]:
```

	Sexo	tamanho da cabeça	peso do cérebro
0	1	4512	1530
1	1	3738	1297
2	1	4261	1335
3	1	3777	1282
4	1	4177	1590

```
In [73]: #Utilizar o método describe para conhecer sua tabela
data.describe()
```

Out[73]:

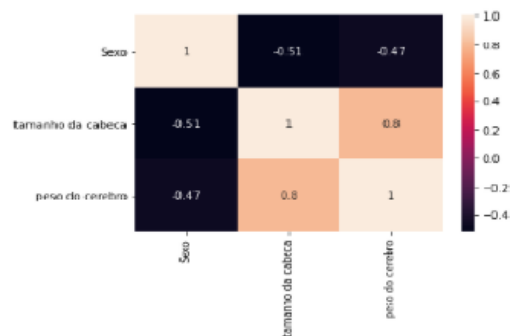
```
In [73]: #Utilizar o método describe para conhecer sua tabela
data.describe()
```

```
Out[73]:
```

	Sexo	tamanho da cabeça	peso do cérebro
count	237.000000	237.000000	237.000000
mean	1.434599	3633.691561	1282.873218
std	0.496753	365.261422	120.340446
min	1.000000	2720.000000	955.000000
25%	1.000000	3389.000000	1207.000000
50%	1.000000	3614.000000	1280.000000
75%	2.000000	3876.000000	1350.000000
max	2.000000	4747.000000	1635.000000

```
In [72]: #Fazer a correlação e seu gráfico de calor (heatmap)
corr= data.corr()
sns.heatmap(corr,annot = True)
```

```
Out[72]: <matplotlib.axes._subplots.AxesSubplot at 0x1298ca2fd38>
```

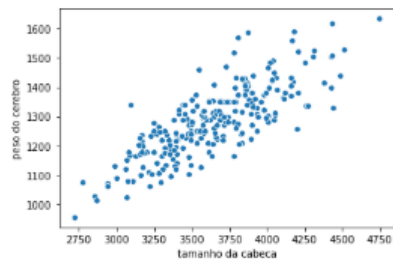


```
In [94]: #Fazer um gráfico de dispersão Peso x Tamanho do Cérebro
sns.scatterplot(x= 'tamanho da cabeça', y= 'peso do cérebro', data=data)
```

```
Out[94]: <matplotlib.axes._subplots.AxesSubplot at 0x1298cb2e848>
```

```
In [94]: #Fazer um gráfico de dispersão Peso x Tamanho do Cérebro
sns.scatterplot(x='tamanho da cabeça', y='peso do cérebro', data=data)
```

```
Out[94]: <matplotlib.axes._subplots.AxesSubplot at 0x1290cb2e040>
```



```
In [98]: #Fazer a regressão linear para prever o peso do cérebro, usando o tamanho dele
formula = 'tamanho da cabeça ~ peso do cérebro'
modelo_v1 = ols(formula, data = data).fit()
modelo_v1.summary()
```

Traceback (most recent call last):

```
File "C:\Users\leona\anaconda3\lib\site-packages\IPython\core\interactiveshell.py", line 3343, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)

File "<ipython-input-98-95508697cbff>", line 4, in <module>
    modelo_v1 = ols(formula, data = data).fit()

File "C:\Users\leona\anaconda3\lib\site-packages\statsmodels\base\model.py", line 168, in from_formula
    tmp = handle_formula_data(data, None, formula, depth=eval_env,

File "C:\Users\leona\anaconda3\lib\site-packages\statsmodels\formula\formulatools.py", line 64, in handle_formula_data
    result = dmatrices(formula, Y, depth, return_type='dataframe',

File "C:\Users\leona\anaconda3\lib\site-packages\patsy\highlevel.py", line 309, in dmatrices
    (lhs, rhs) = _do_highlevel_design(formula_like, data, eval_env,

File "C:\Users\leona\anaconda3\lib\site-packages\patsy\highlevel.py", line 164, in _do_highlevel_design
    design_infos = _try_incr_builders(formula_like, data_iter_maker, eval_env,

File "C:\Users\leona\anaconda3\lib\site-packages\patsy\highlevel.py", line 66, in _try_incr_builders
    return design_matrix_builders([formula_like.lhs_termlist,

File "C:\Users\leona\anaconda3\lib\site-packages\patsy\build.py", line 689, in design_matrix_builders
```