

Lista de Exercícios - 10 parte 2

June 18, 2021

Universidade Federal de Mato Grosso do Sul

Campus de Campo Grande Estatística – Prof. Cássio Pinho dos Reis

10ª LISTA DE EXERCÍCIOS - parte 2

Turma: Engenharia de Software RGA: 2021.1906.069-7 Aluno: Maycon Felipe da Silva Mota

Importar bibliotecas que usaremos para esse exercício

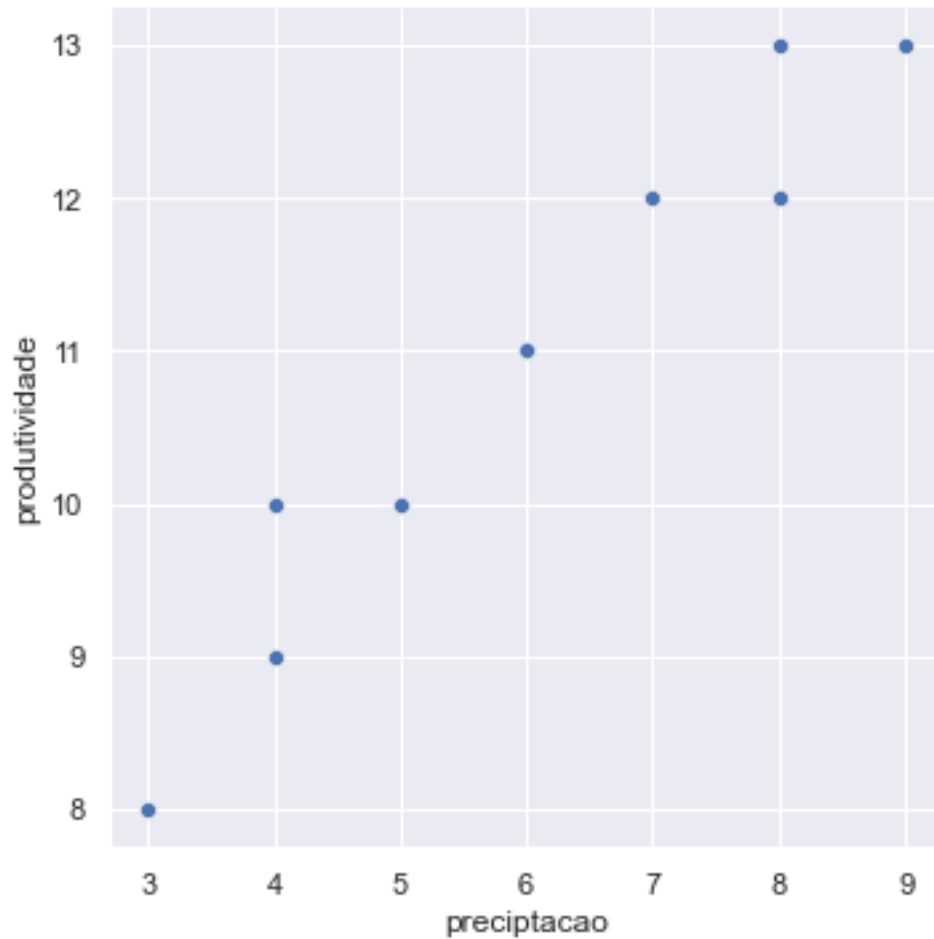
```
[1]: import pandas as pd
import numpy as np
import seaborn as sns
sns.set_theme(style="darkgrid")
```

- 1ª Questão – Um produtor deseja verificar se precipitação de chuva tem influencia significativa na produtividade diária da variedade de arroz que produz. A tabela abaixo apresenta os valores da produtividade de arroz em Kg/100m2 e da precipitação de chuva. Há essa influência? Se sim, construa um modelo de regressão linear simples, e estime qual a produtividade esperada de arroz, com uma precipitação de 6,5 de chuva.

```
[2]: produtividade = [8,9,10,10,11,11,12,12,13,13]
precipitacao = [3,4,4,5,6,6,7,8,8,9]

df_final = pd.DataFrame({'precipitacao': precipitacao, 'produtividade':
    ↳ produtividade})
sns.relplot(x="precipitacao", y="produtividade", data=df_final)
```

```
[2]: <seaborn.axisgrid.FacetGrid at 0x21f3c4138e0>
```



Sim, pois quanto mais aumenta a precipitação, aumenta a produtividade. Conforme o diagrama.

```
[3]: from sklearn.datasets import load_boston # para carregar os dados from sklearn.  
      ↪ model_selection import train_test_split  
      from sklearn.linear_model import LinearRegression # importa o modelo  
  
X = np.array([3,4,4,5,6,6,7,8,8,9]).reshape((-1, 1))  
y = [8,9,10,10,11,11,12,12,13,13]  
  
modelo = LinearRegression() # cria o modelo  
modelo.fit(X, y) # ajusta o modelo aos dados  
  
r_sq = modelo.score(X, y) # verifica o valor de R²  
print('Coeficiente de pearson:', r_sq)  
print('Valor de Alfa:', modelo.intercept_) # valor de b0  
print('Valor de Inclinação da reta:', modelo.coef_) # valor de b1
```

```

produtividade_esperada = modelo.predict(X) # verificar os valores previstos com
↳ realizado com base nos dados
pd.DataFrame({'pred': produtividade_esperada, 'produtividade_real':
↳ produtividade, 'precipitacao_real': precipitacao}) # gerar um quadro

```

Coeficiente de pearson: 0.9381972333779562

Valor de Alfa: 6.066666666666668

Valor de Inclinação da reta: [0.80555556]

```

[3]:      pred  produtividade_real  precipitacao_real
0    8.483333           8           3
1    9.288889           9           4
2    9.288889          10           4
3   10.094444          10           5
4   10.900000          11           6
5   10.900000          11           6
6   11.705556          12           7
7   12.511111          12           8
8   12.511111          13           8
9   13.316667          13           9

```

```

[4]: modelo.predict(np.array([6.5]).reshape((-1, 1)))[0] # prever os valores com 6.
↳ 5mm de precipitação

```

```

[4]: 11.302777777777777

```

2 2ª Questão – Considere os dados referentes à produção de matéria seca de uma cultura e a quantidade de radiação fotossintética ativa:

2.1 Dessas duas variáveis, qual você consideraria ser a variável resposta e qual a variável preditora? Por quê? Construa um diagrama de dispersão, faça a equação de regressão e interprete.

```

[5]: producao = [10,60,110,160,220,280,340,400,460,520]
radiacao = [18,55,190,300,410,460,570,770,815,965]

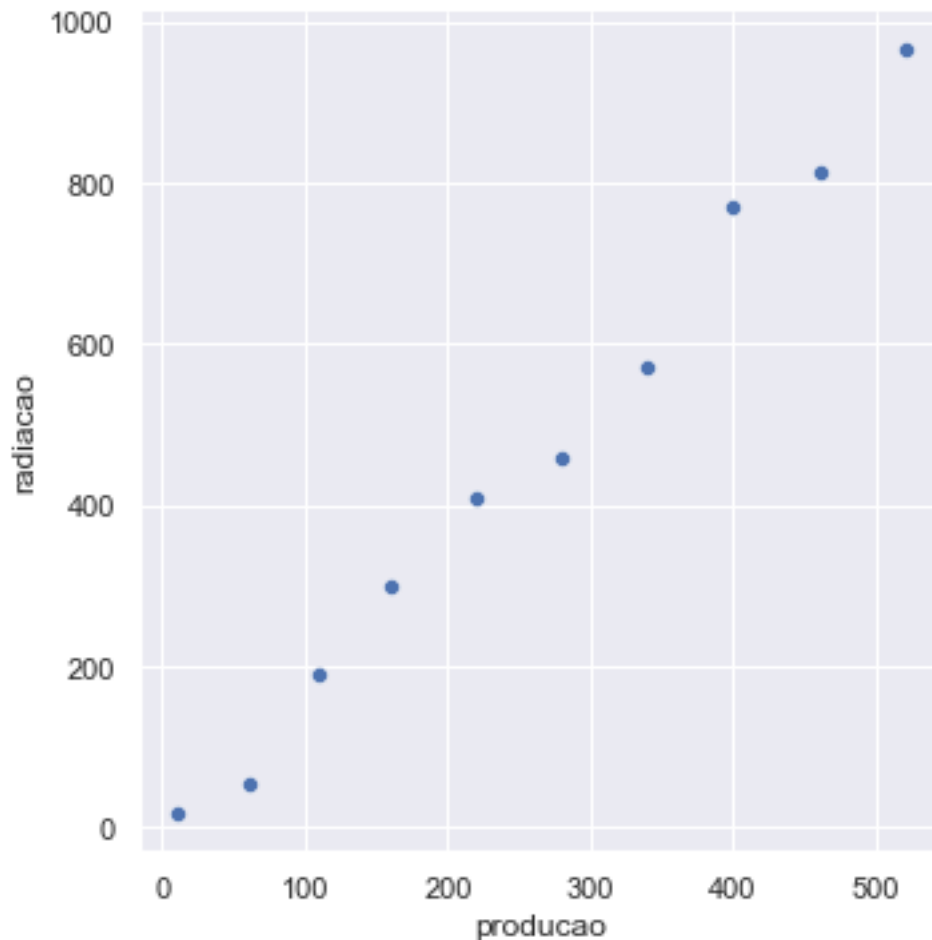
df_final = pd.DataFrame({'producao': producao, 'radiacao': radiacao})
sns.relplot(x="producao", y="radiacao", data=df_final)

```

```

[5]: <seaborn.axisgrid.FacetGrid at 0x21f41d76f70>

```



```
[6]: X = np.array([10,60,110,160,220,280,340,400,460,520]).reshape((-1, 1))
y = [18,55,190,300,410,460,570,770,815,965]

modelo = LinearRegression() # cria o modelo
modelo.fit(X, y) # ajusta o modelo aos dados

r_sq = modelo.score(X, y) # verifica o valor de R²
print('Coeficiente de pearson:', r_sq)
print('Valor de Alfa:', modelo.intercept_) # valor de b0
print('Valor de Inclinação da reta:', modelo.coef_) # valor de b1

radiacao_esperada = modelo.predict(X) # verificar os valores previstos com
↳ realizado com base nos dados
pd.DataFrame({'radiacao_pred': radiacao_esperada, 'radiacao': radiacao,
↳ 'producao': producao}) # gerar um quadro
```

Coeficiente de pearson: 0.9905357857593803
Valor de Alfa: -22.084528915840792
Valor de Inclinação da reta: [1.86478332]

```
[6]: radiacao_pred radiacao producao
0      -3.436696        18        10
1      89.802470        55        60
2     183.041636       190       110
3     276.280802       300       160
4     388.167801       410       220
5     500.054800       460       280
6     611.941799       570       340
7     723.828798       770       400
8     835.715796       815       460
9     947.602795       965       520
```

Nessa equação eu usaria a variável de produtividade para saber quanto de radioatividade eu teria.
A equação da reta pode ser dada por $y = -22.08 + 1.8647 \text{ producao} + \text{producao}$

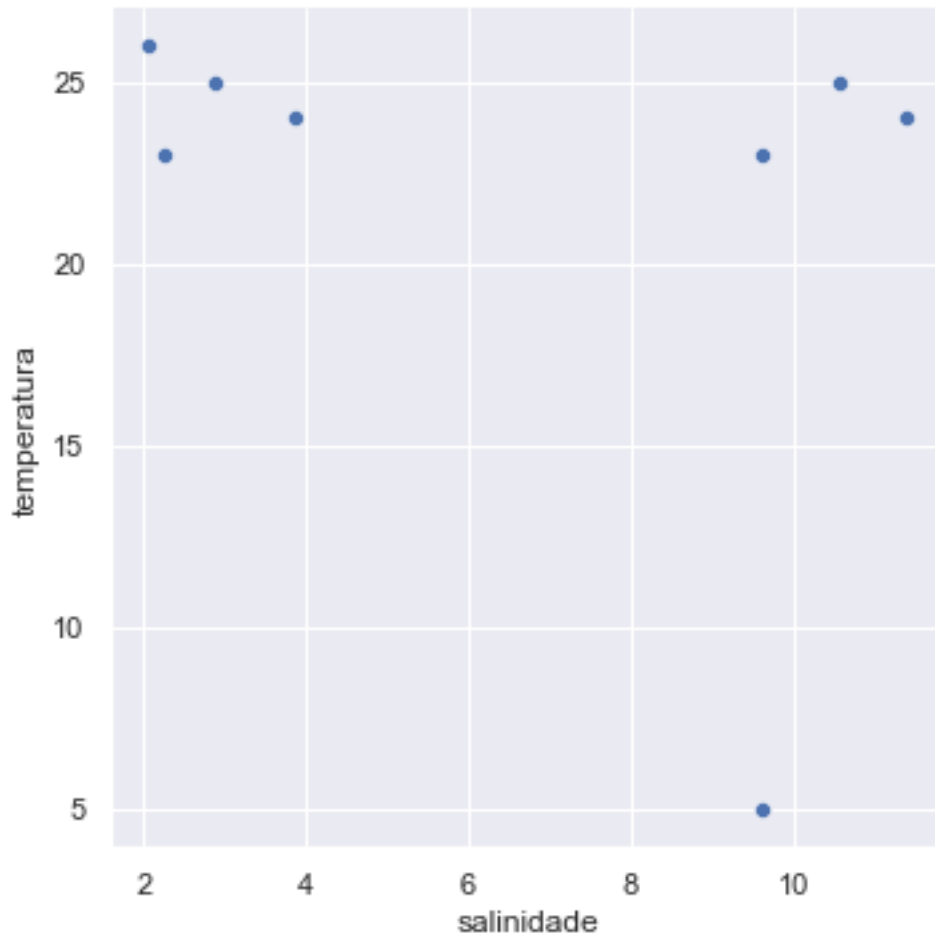
3 3ª Questão – Uma pesquisadora está interessada em verificar se existe correlação entre a salinidade (g/l) e a temperatura numa região de uma lagoa. Faça o diagrama de dispersão, calcule o índice de correlação de Pearson, classificando-o e estime qual a salinidade da lagoa em uma temperatura de 23,5°C.

```
[ ]:
```

```
[10]: temperatura = [24,23,23,26,25,5,25,24]
salinidade = [3.85,9.61,2.26,2.06,2.89,9.61,10.58,11.40]

df_final = pd.DataFrame({'temperatura': temperatura, 'salinidade': salinidade})
sns.relplot(x="salinidade", y="temperatura", data=df_final)
```

```
[10]: <seaborn.axisgrid.FacetGrid at 0x21f41e63ac0>
```



```
[14]: X = np.array([24,23,23,26,25,5,25,24]).reshape((-1, 1))
y = [3.85,9.61,2.26,2.06,2.89,9.61,10.58,11.40]

modelo = LinearRegression() # cria o modelo
modelo.fit(X, y) # ajusta o modelo aos dados

r_sq = modelo.score(X, y) # verifica o valor de R²
print('Coeficiente de pearson:', r_sq)
print('Valor de Alfa:', modelo.intercept_) # valor de b0
print('Valor de Inclinação da reta:', modelo.coef_) # valor de b1

salinidade_esperada = modelo.predict(X) # verificar os valores previstos com
↳ realizado com base nos dados
pd.DataFrame({'salinidade_esperada': salinidade_esperada, 'salinidade':
↳ salinidade, 'temperatura': temperatura}) # gerar um quadro
```

Coeficiente de pearson: 0.11053125851910783
Valor de Alfa: 10.857720615846791
Valor de Inclinação da reta: [-0.19772437]

```
[14]:      salinidade_esperada  salinidade  temperatura
0          6.112336         3.85         24
1          6.310060         9.61         23
2          6.310060         2.26         23
3          5.716887         2.06         26
4          5.914611         2.89         25
5          9.869099         9.61          5
6          5.914611        10.58         25
7          6.112336        11.40         24
```

Por apresentar um coeficiente de pearson < 0.5 , não é recomendável utilizar regressão linear simples para essa análise.