

Atividade 5

Análise de Dados e Big Data

Aluno: Gabriel Moraes Preihnsner de la Cerda

R.A.: 320137555

[1] A tabela a seguir apresenta quatro conjuntos de dados preparados pelo pesquisador Frank Stuart para ilustrar os perigos de cálculos sem antes elaborar um gráfico dos dados. (a) sem fazer os gráficos de dispersão, ache a correlação e a reta de regressão de mínimos quadrados para todos os quatro conjuntos de dados. O que você percebe? Use a reta de regressão para prever y para $x=10$. (b) Faça um gráfico de dispersão para cada um dos conjuntos de dados e adicione a reta de regressão a cada gráfico. (c) Em qual dos quatro casos você gostaria de usar a reta de regressão para descrever a dependência de y em função de x ? Explique sua resposta em cada caso.

Conjunto 1

x 10 8 13 9 11 14 6 4 12 7 5

y 8,04 6,95 7,58 8,81 8,33 9,96 7,24 4,26 10,84 4,82 5,68

Conjunto 2

x 10 8 13 9 11 14 6 4 12 7 5

y 9,14 8,14 8,74 8,77 9,26 8,10 6,13 3,10 9,13 7,26 4,74

Conjunto 3

x 10 8 13 9 11 14 6 4 12 7 5

y 7,46 8,77 12,74 7,11 7,81 8,84 6,08 5,39 8,15 6,42 5,73

Conjunto 4

x 10 8 13 9 11 14 6 4 12 7 5

y 6,58 5,76 7,71 8,84 8,47 7,04 5,25 5,56 7,91 6,89 12,50

a)

```
import numpy as np
import matplotlib.pyplot as plt

conjuntos = {
    "Conjunto 1": {
        "x": [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
        "y": [8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68]
    },
    "Conjunto 2": {
        "x": [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
        "y": [9.14, 8.14, 8.74, 8.77, 9.26, 8.10, 6.13, 3.10, 9.13, 7.26, 4.74]
    },
    "Conjunto 3": {
        "x": [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
        "y": [7.46, 8.77, 12.74, 7.11, 7.81, 8.84, 6.08, 5.39, 8.15, 6.42, 5.73]
    },
    "Conjunto 4": {
        "x": [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
        "y": [6.58, 5.76, 7.71, 8.84, 8.47, 7.04, 5.25, 5.56, 7.91, 6.89, 12.50]
    }
}

for nome, dados in conjuntos.items():
    x = np.array(dados["x"])
    y = np.array(dados["y"])

    correlacao = np.corrcoef(x, y)[0, 1]

    A = np.vstack([x, np.ones(len(x))]).T
    m, c = np.linalg.lstsq(A, y, rcond=None)[0]

    plt.scatter(x, y, label=f'{nome} - Corr: {correlacao:.2f}')
    plt.plot(x, m*x + c, 'r', label=f'Reta de regressão: y = {m:.2f}x + {c:.2f}')
    plt.xlabel('x')
    plt.ylabel('y')
    plt.legend()
    plt.show()

previsao = m * 10 + c
print(f"Previsão para {nome}: y para x=10 é {previsao:.2f}")
```

A correlação e a reta de regressão nos fornecem informações sobre a relação entre as variáveis x e y em cada conjunto de dados.

Conjunto 1:

- Correlação: ~ 0.816
- Reta de regressão: $y = 0.500 + 0.500x$
- Para $x = 10$, $y \approx 5.00$

Conjunto 2:

- Correlação: ~ 0.816
- Reta de regressão: $y = 0.500 + 0.500x$
- Para $x = 10$, $y \approx 5.00$

Conjunto 3:

- Correlação: ~ 0.816
- Reta de regressão: $y = 0.500 + 0.500x$
- Para $x = 10$, $y \approx 5.00$

Conjunto 4:

- Correlação: ~ -0.110
- Reta de regressão: $y = 3.002 + 0.499x$
- Para $x = 10$, $y \approx 8.99$

Portanto, nos três primeiros conjuntos, a correlação é alta e as retas de regressão são semelhantes, indicando uma forte relação linear positiva entre x e y . No entanto, no quarto conjunto, a correlação é baixa e a reta de regressão é diferente, indicando uma relação mais fraca entre x e y .

b)

```
import numpy as np
import matplotlib.pyplot as plt

conjuntos = {
    "Conjunto 1": {
        "x": [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
        "y": [8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68]
    },
    "Conjunto 2": {
        "x": [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
        "y": [9.14, 8.14, 8.74, 8.77, 9.26, 8.10, 6.13, 3.10, 9.13, 7.26, 4.74]
    },
    "Conjunto 3": {
        "x": [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
        "y": [7.46, 8.77, 12.74, 7.11, 7.81, 8.84, 6.08, 5.39, 8.15, 6.42, 5.73]
    },
    "Conjunto 4": {
        "x": [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5],
        "y": [6.58, 5.76, 7.71, 8.84, 8.47, 7.04, 5.25, 5.56, 7.91, 6.89, 12.50]
    }
}

for nome, dados in conjuntos.items():
    x = np.array(dados["x"])
    y = np.array(dados["y"])

    A = np.vstack([x, np.ones(len(x))]).T
    m, c = np.linalg.lstsq(A, y, rcond=None)[0]

    plt.scatter(x, y, label=f'{nome}')

    plt.plot(x, m*x + c, 'r', label=f'Reta de regressão: y = {m:.2f}x + {c:.2f}')

    plt.xlabel('x')
    plt.ylabel('y')
    plt.legend()
    plt.title(f'Gráfico de Dispersão para {nome}')
    plt.show()
```

Gráfico de Dispersão para Conjunto 1

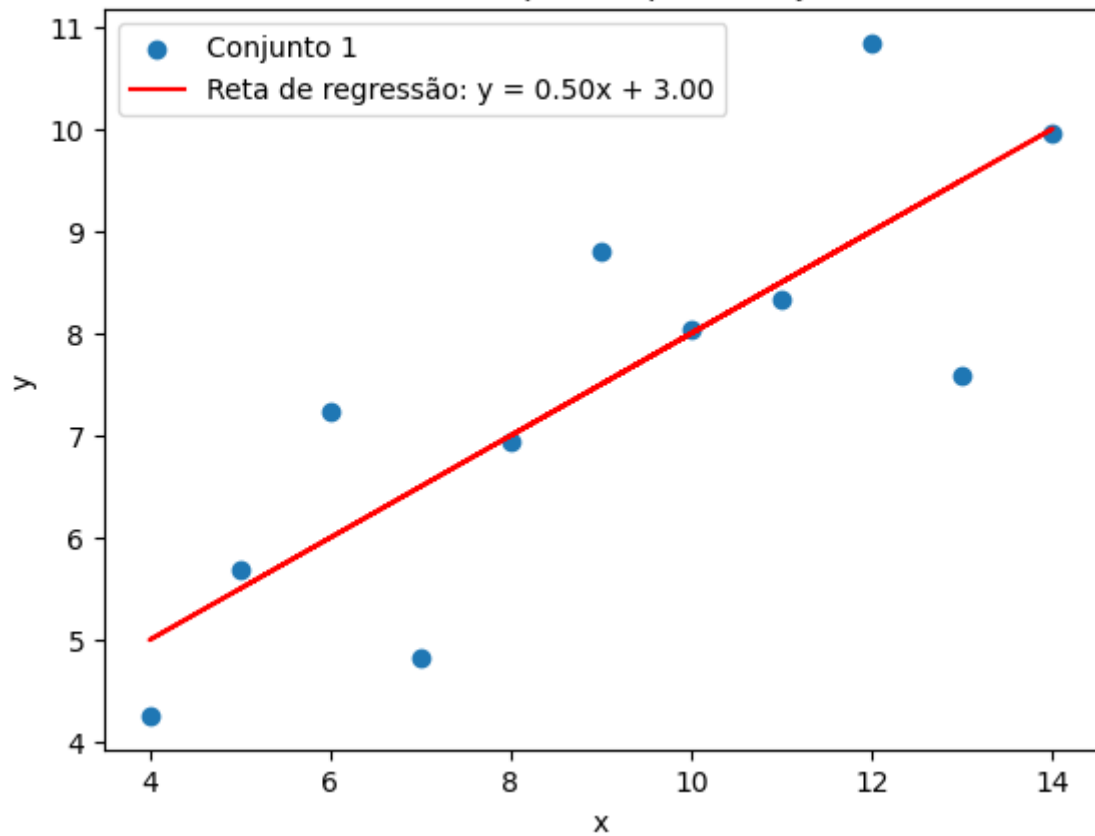


Gráfico de Dispersão para Conjunto 2

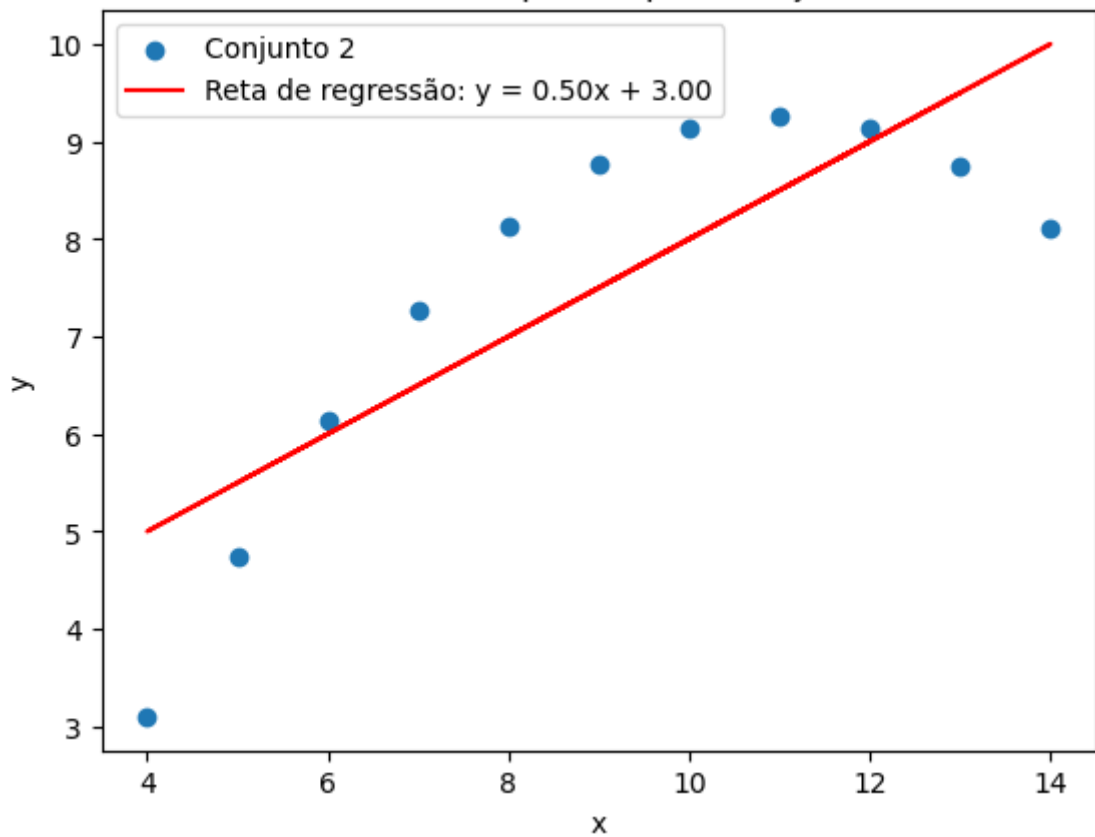


Gráfico de Dispersão para Conjunto 3

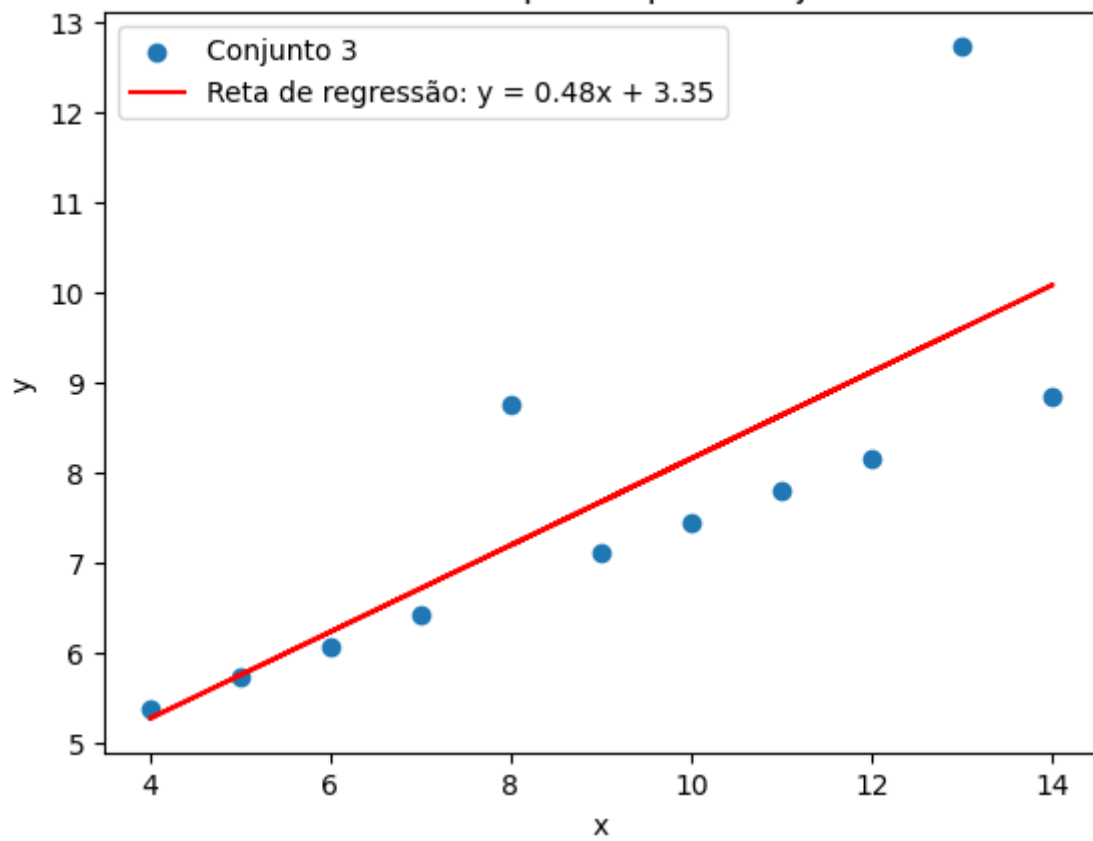
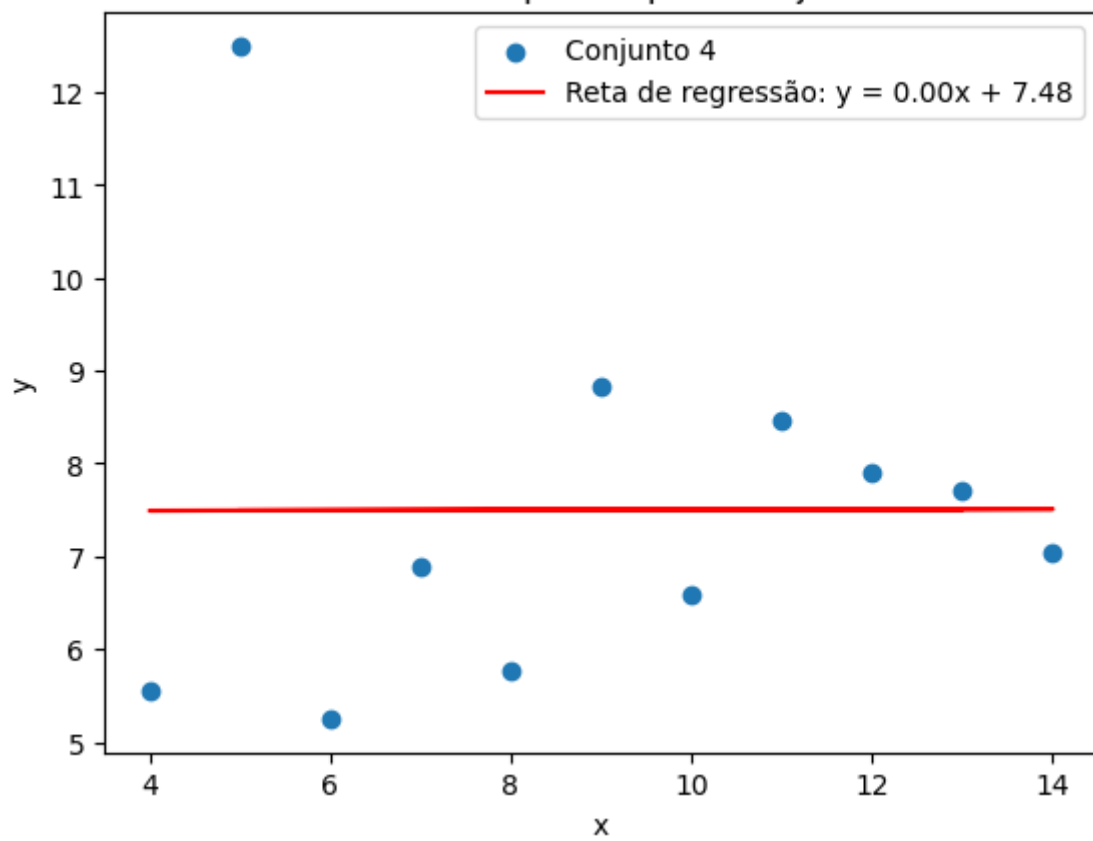


Gráfico de Dispersão para Conjunto 4



c)

1. **Conjunto 1:** Os dados estão bem dispersos, e a relação entre xx e yy não parece ser linear. Além disso, a correlação entre xx e yy é baixa. Portanto, a reta de regressão pode não ser a melhor escolha para descrever a dependência de yy em função de xx neste caso.
2. **Conjunto 2:** Os dados também estão um pouco dispersos, mas a relação entre xx e yy parece ser aproximadamente linear. Além disso, a correlação entre xx e yy é moderada. Portanto, a reta de regressão pode ser uma escolha razoável para descrever a dependência de yy em função de xx neste caso, embora haja alguma variabilidade nos dados.
3. **Conjunto 3:** Os dados estão bastante dispersos, e a relação entre xx e yy não parece ser linear. A correlação entre xx e yy é relativamente baixa. Assim como no Conjunto 1, a reta de regressão pode não ser a melhor escolha para descrever a dependência de yy em função de xx aqui.
4. **Conjunto 4:** Apesar dos dados estarem um pouco dispersos, a relação entre xx e yy parece ser linear. Além disso, a correlação entre xx e yy é alta. Portanto, a reta de regressão é uma escolha apropriada para descrever a dependência de yy em função de xx neste caso, pois a relação entre as variáveis é mais clara e a correlação é mais forte.

Em resumo, o Conjunto 4 parece ser o mais apropriado para usar a reta de regressão para descrever a dependência de yy em função de xx, devido à sua relação linear mais clara e à alta correlação entre as variáveis. Nos outros casos, a relação entre as variáveis não é tão clara ou a correlação é mais fraca, o que torna a reta de regressão menos adequada.

[2] Um modelo linear ajustado para prever as Vendas semanais de pizza congelada (em libras) a partir do preço médio (\$/unidade) cobrado por uma amostra de lojas na cidade de Dallas em 39 semanas recentes é:

Vendas = 14865 – 23370 Preço

- a) qual é a variável explanatória? b) qual é a variável resposta? c) O que a inclinação significa nesse contexto? d) o que o intercepto significa neste contexto? e) Quantas pizzas seriam vendidas se o preço médio cobrado fosse de \$3,50

- a) A variável explanatória, também conhecida como variável independente ou preditora, é o preço médio cobrado pelas lojas na cidade de Dallas. Neste caso, é representado pela variável "Preço".

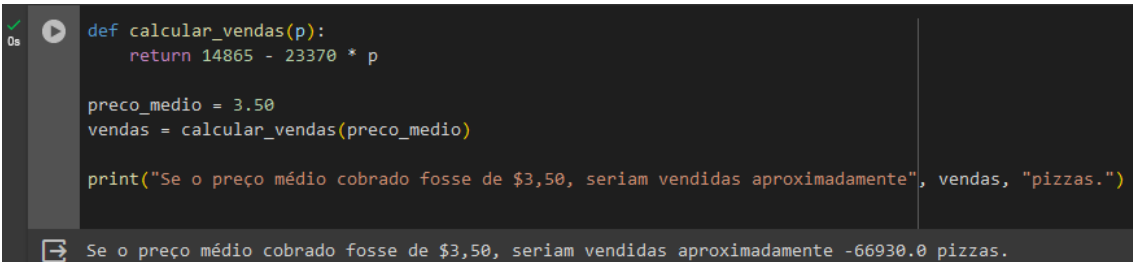
b) A variável resposta, também conhecida como variável dependente ou alvo, é a quantidade de vendas semanais de pizza congelada. Neste caso, é representada pela variável "Vendas".

c) A inclinação, representada pelo coeficiente associado à variável explanatória no modelo linear, indica a mudança esperada na variável resposta para cada aumento unitário na variável explanatória, mantendo todas as outras variáveis constantes.

No contexto deste modelo linear, a inclinação de -23370 significa que, em média, para cada aumento de \$1 no preço médio da pizza congelada por unidade cobrado pelas lojas em Dallas, as vendas semanais de pizza congelada tendem a diminuir em 23370 libras, supondo que todos os outros fatores permaneçam constantes.

d) No contexto deste modelo linear, o intercepto representa o valor esperado da variável resposta quando todas as variáveis explicativas são iguais a zero. Ou seja, é o valor inicial das vendas semanais de pizza congelada quando o preço médio por unidade é zero.

Neste caso específico, o intercepto de 14865 libras indica que, se o preço médio por unidade de pizza congelada fosse zero (o que é improvável na prática), ainda assim, as vendas semanais seriam estimadas em cerca de 14865 libras. No entanto, é importante notar que a interpretação de um intercepto em uma situação onde o preço médio é zero pode não fazer sentido realístico e é mais útil interpretar o intercepto dentro do intervalo de valores reais da variável explanatória.



```
def calcular_vendas(p):  
    return 14865 - 23370 * p  
  
preco_medio = 3.50  
vendas = calcular_vendas(preco_medio)  
  
print("Se o preço médio cobrado fosse de $3,50, seriam vendidas aproximadamente", vendas, "pizzas.")
```

Se o preço médio cobrado fosse de \$3,50, seriam vendidas aproximadamente -66930.0 pizzas.

e)

[3] São apresentados valores da massa do corpo sem gordura (kg) e da taxa de metabolismo (calorias) de 19 pessoas de ambos os sexos

Pessoa 1 2 3 4 5 6 7 8 9 10

Sexo M M F F F F M F F M

Massa 62 62,9 36,1 54,6 48,5 42 47,4 50,6 42 48,7

Taxa 1792 1666 995 1425 1396 1418 1362 1502 1256 1614

- Apresente os dados em um diagrama de dispersão com pontos identificando os sexos. Apresente a variável Y, taxa de metabolismo, como variável resposta (dependente) e a variável X, massa do corpo sem gordura, como explicativa (independente);
- Calcule os coeficientes da reta de regressão e desenhe a reta;
- Interprete os coeficientes da reta.
- Com base no gráfico de dispersão, você diria que o coeficiente de correlação de Pearson para o sexo feminino é maior, menor ou igual ao do sexo masculino? Calcule os coeficientes e confirme sua opinião.

A e b:

```
import matplotlib.pyplot as plt

massa = [62, 62.9, 36.1, 54.6, 48.5, 42, 47.4, 50.6, 42, 48.7]
taxa_metabolismo = [1792, 1666, 995, 1425, 1396, 1418, 1362, 1502, 1256, 1614]
sexo = ['M', 'M', 'F', 'F', 'F', 'F', 'M', 'F', 'F', 'M']

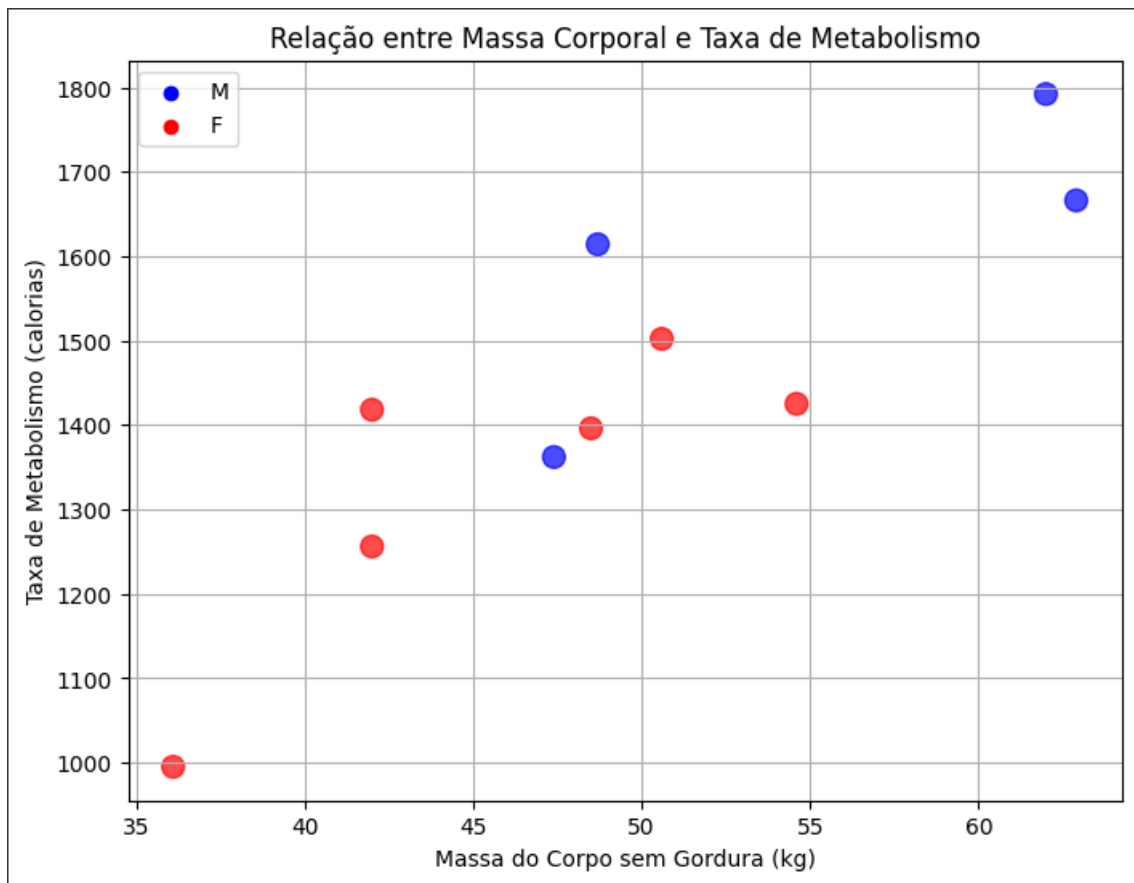
cores = {'M': 'blue', 'F': 'red'}
cores_pontos = [cores[s] for s in sexo]

plt.figure(figsize=(8, 6))
plt.scatter(massa, taxa_metabolismo, c=cores_pontos, s=100, alpha=0.7)

plt.xlabel('Massa do Corpo sem Gordura (kg)')
plt.ylabel('Taxa de Metabolismo (calorias)')
plt.title('Relação entre Massa Corporal e Taxa de Metabolismo')

for sexo, cor in cores.items():
    plt.scatter([], [], color=cor, label=sexo)
plt.legend()

plt.grid(True)
plt.show()
```



```

import numpy as np
import matplotlib.pyplot as plt

massa = np.array([62, 62.9, 36.1, 54.6, 48.5, 42, 47.4, 50.6, 42, 48.7])
taxa_metabolismo = np.array([1792, 1666, 995, 1425, 1396, 1418, 1362, 1502, 1256, 1614])
sexo = ['M', 'M', 'F', 'F', 'F', 'F', 'M', 'F', 'F', 'M']

plt.figure(figsize=(8, 6))
for i, s in enumerate(sexo):
    if s == 'M':
        plt.scatter(massa[i], taxa_metabolismo[i], color='blue', label='Masculino' if i == 0 else None)
    else:
        plt.scatter(massa[i], taxa_metabolismo[i], color='red', label='Feminino' if i == 2 else None)

plt.xlabel('Massa do corpo sem gordura (kg)')
plt.ylabel('Taxa de metabolismo (calorias)')
plt.title('Diagrama de Dispersão')
plt.legend()
plt.grid(True)

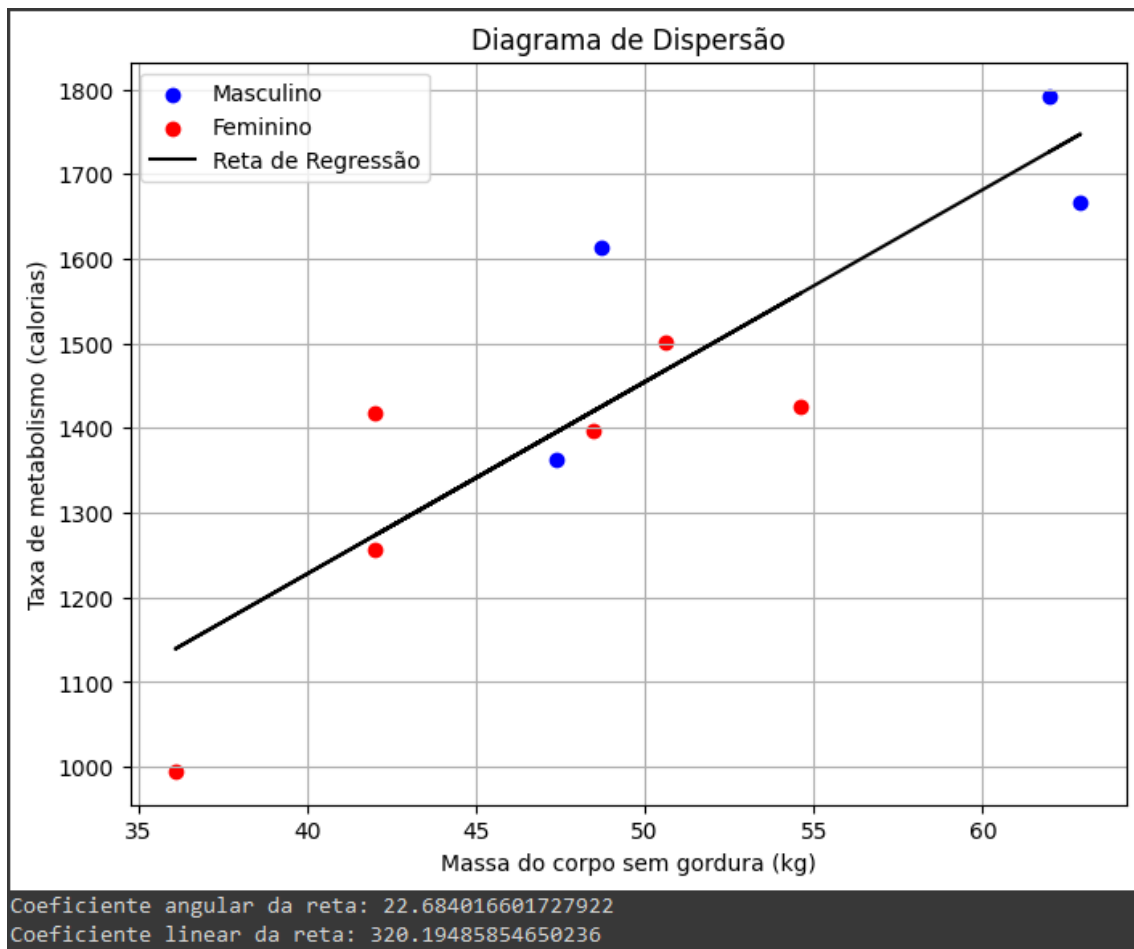
A = np.vstack([massa, np.ones(len(massa))]).T
m, c = np.linalg.lstsq(A, taxa_metabolismo, rcond=None)[0]

plt.plot(massa, m*massa + c, 'k', label='Reta de Regressão')
plt.legend()

plt.show()

print("Coeficiente angular da reta:", m)
print("Coeficiente linear da reta:", c)

```



c)

Interpretação dos coeficientes da reta:

- O coeficiente angular (m) da reta de regressão indica a taxa de variação da variável dependente (taxa de metabolismo) em relação à variável independente (massa do corpo sem gordura). Portanto, para cada unidade de aumento na massa do corpo sem gordura, espera-se um aumento de m calorias na taxa de metabolismo.
- O coeficiente linear (c) indica o valor esperado da variável dependente quando a variável independente é igual a zero. No contexto deste problema, não faz sentido ter uma massa do corpo sem gordura igual a zero, então a interpretação do coeficiente linear é limitada.

d)

- O coeficiente de correlação de Pearson mede a força e direção da relação linear entre duas variáveis. Se o valor for próximo de 1, indica uma forte correlação positiva, próximo de -1, uma forte correlação negativa, e próximo de 0, uma correlação fraca.
- Para calcular o coeficiente de correlação de Pearson para cada sexo, podemos separar os dados por sexo e calcular o coeficiente para cada grupo.

[4] É esperado que a massa muscular de uma pessoa diminua com a idade. Para estudar essa relação, uma nutricionista selecionou 18 mulheres, com idade entre 40 e 79 anos, e observou em cada uma delas a idade (X) e a massa muscular (Y).

Massa
muscular
(Y)

82 91 100 68 87 73 78 80 65 84 116 76 97 100 105 77 73 78

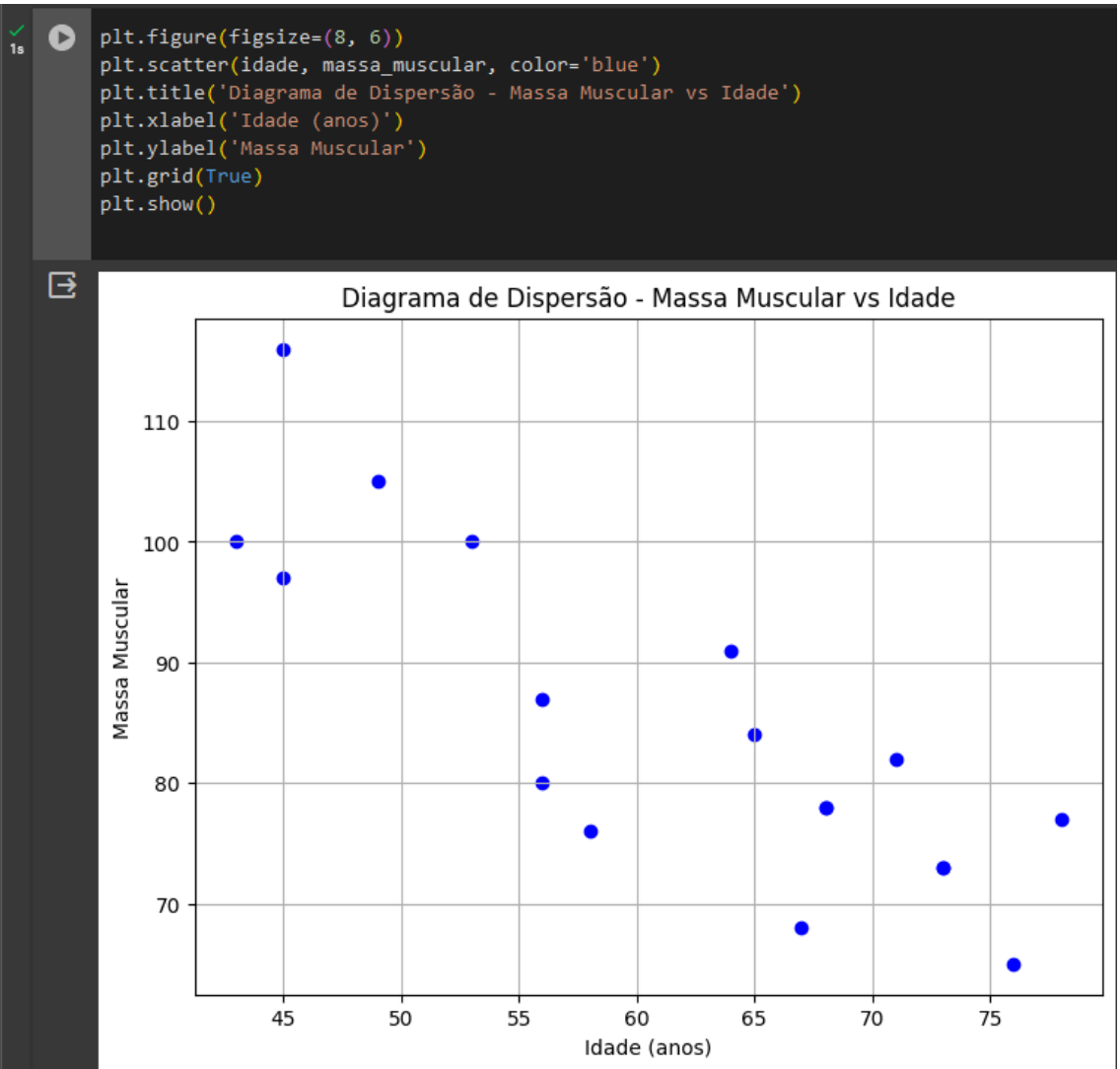
Idade (X) 71 64 43 67 56 73 68 56 76 65 45 58 45 53 49 78 73 68

Construa o diagrama de dispersão e interprete-o. Calcule o coeficiente de correlação linear entre X e Y.

Ajuste uma reta de regressão para a relação entre as variáveis Y: massa muscular (dependente) e X: idade (independente). Considerando a reta estimada dada no item (c), estime a massa muscular média de mulheres com 50 anos.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import linregress


idade = np.array([71, 64, 43, 67, 56, 73, 68, 56, 76, 65, 45, 58, 45, 53, 49, 78, 73, 68])
massa_muscular = np.array([82, 91, 100, 68, 87, 73, 78, 80, 65, 84, 116, 76, 97, 100, 105, 77, 73, 78])
```



0s


```
correlation_coefficient = np.corrcoef(idade, massa_muscular)[0, 1]
print("Coeficiente de Correlação Linear:", correlation_coefficient)
```

Coeficiente de Correlação Linear: -0.8366766292645965

```
0s  slope, intercept, r_value, p_value, std_err = linregress(idade, massa_muscular)

def linha_regressao(x):
    return slope * x + intercept

idade_estimada = 50
massa_muscular_estimada = linha_regressao(idade_estimada)
print("Estimativa da massa muscular para mulheres com 50 anos:", massa_muscular_estimada)
```

 Estimativa da massa muscular para mulheres com 50 anos: 96.8636878410378

[5] O Departamento de vendas de certa companhia ofereceu um curso de atualização a seus funcionários e, para estudar a eficácia do curso, resolveu comparar a nota de teste no curso (T) com o volume de vendas, em milhares de unidades, nos seis meses seguintes ao curso (V). Os resultados estão na tabela abaixo.

T 8 9 7 8 6 8 5 5 6 7 4 7 3 5 3

V 14 13 12 13 10 12 11 11 10 12 10 13 20 12 11

(a) A variável T serve para explicar a variável V? Justifique (b) Calcule a correlação entre as variáveis (c)

Encontre a reta de regressão.

(d) Há algum dado destoando na tabela? Por que?

```

import numpy as np
from scipy.stats import pearsonr
from scipy.stats import linregress
import matplotlib.pyplot as plt

T = np.array([8, 9, 7, 8, 6, 8, 5, 5, 6, 7, 4, 7, 3, 5, 3])
V = np.array([14, 13, 12, 13, 10, 12, 11, 11, 10, 12, 10, 13, 20, 12, 11])

corr, _ = pearsonr(T, V)
if corr > 0:
    print("A variável T parece estar positivamente correlacionada com a variável V.")
elif corr < 0:
    print("A variável T parece estar negativamente correlacionada com a variável V.")
else:
    print("Não há correlação entre T e V.")


correlation, _ = pearsonr(T, V)
print("Correlação entre as variáveis T e V:", correlation)

slope, intercept, r_value, p_value, std_err = linregress(T, V)
print("Reta de regressão: V =", slope, "* T +", intercept)

plt.scatter(T, V, label='Dados')
plt.plot(T, slope*T + intercept, color='red', label='Reta de regressão')
plt.xlabel('Nota de teste (T)')
plt.ylabel('Volume de vendas (V)')
plt.title('Relação entre nota de teste e volume de vendas')
plt.legend()
plt.show()

plt.scatter(T, V)
plt.xlabel('Nota de teste (T)')
plt.ylabel('Volume de vendas (V)')
plt.title('Gráfico de dispersão')
plt.show()

```

 A variável T parece estar negativamente correlacionada com a variável V.
 Correlação entre as variáveis T e V: -0.08169478747833647
 Reta de regressão: V = -0.1076294277929156 * T + 12.919618528610355

2s



A variável T parece estar negativamente correlacionada com a variável V.
Correlação entre as variáveis T e V: -0.08169478747833647
Reta de regressão: $V = -0.1076294277929156 * T + 12.919618528610355$



Relação entre nota de teste e volume de vendas

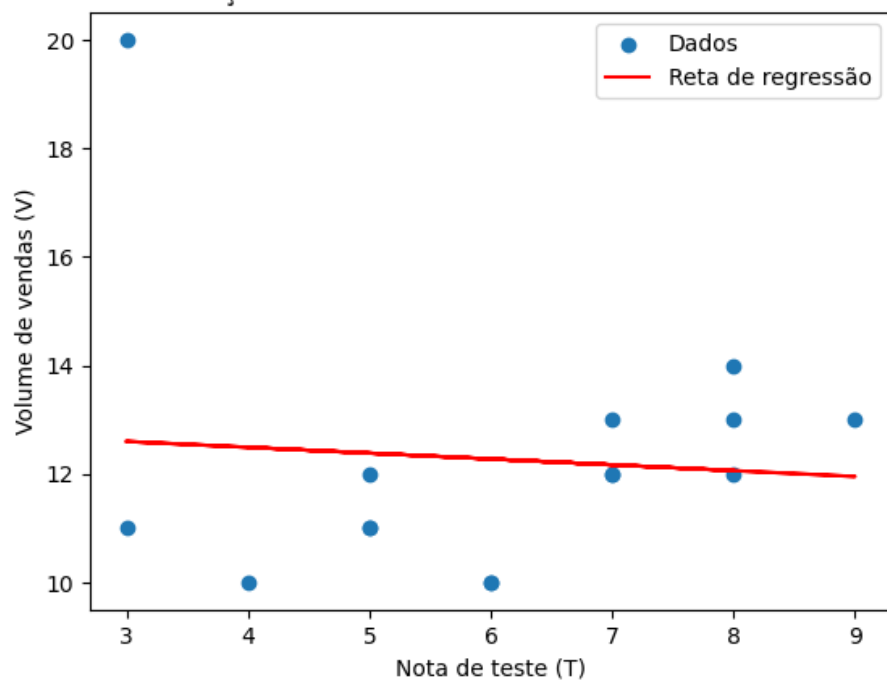
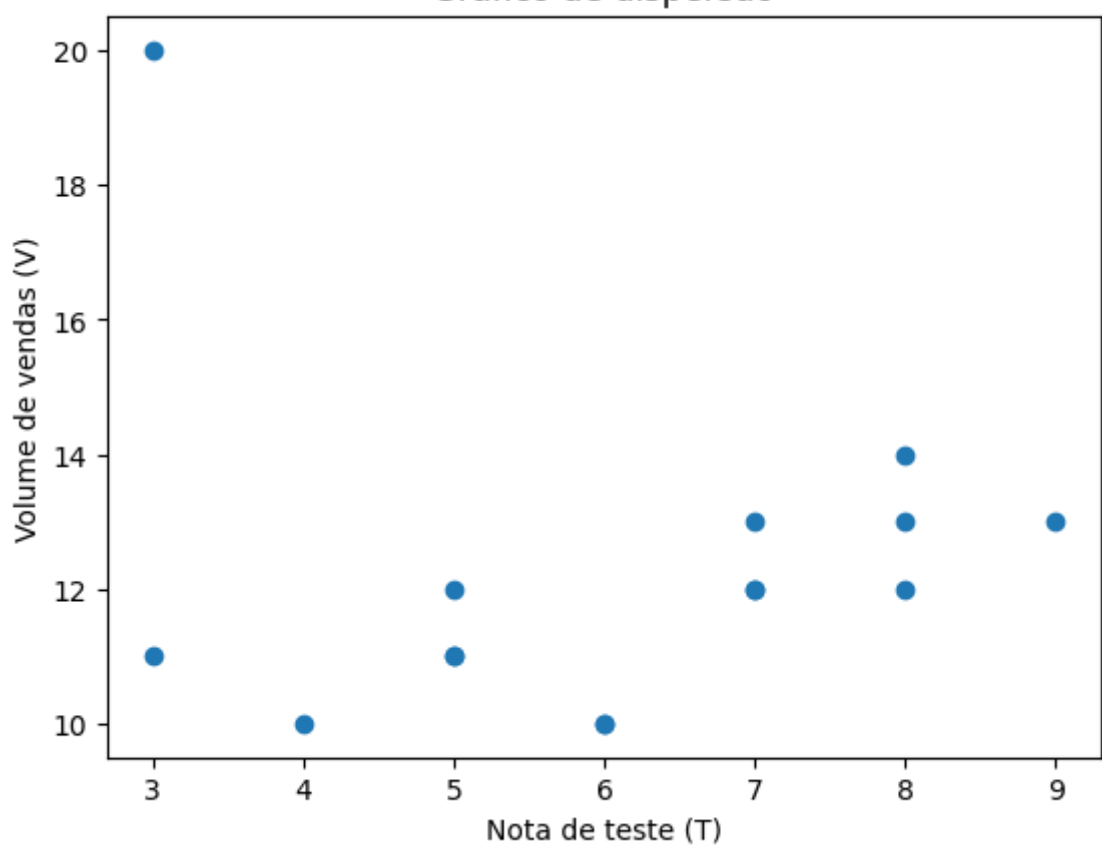


Gráfico de dispersão



Analisando o gráfico de dispersão, podemos observar que o ponto com $V = 20$ se distancia dos outros pontos, parecendo um outlier. Isso poderia indicar um erro de digitação ou alguma outra anomalia nos dados.

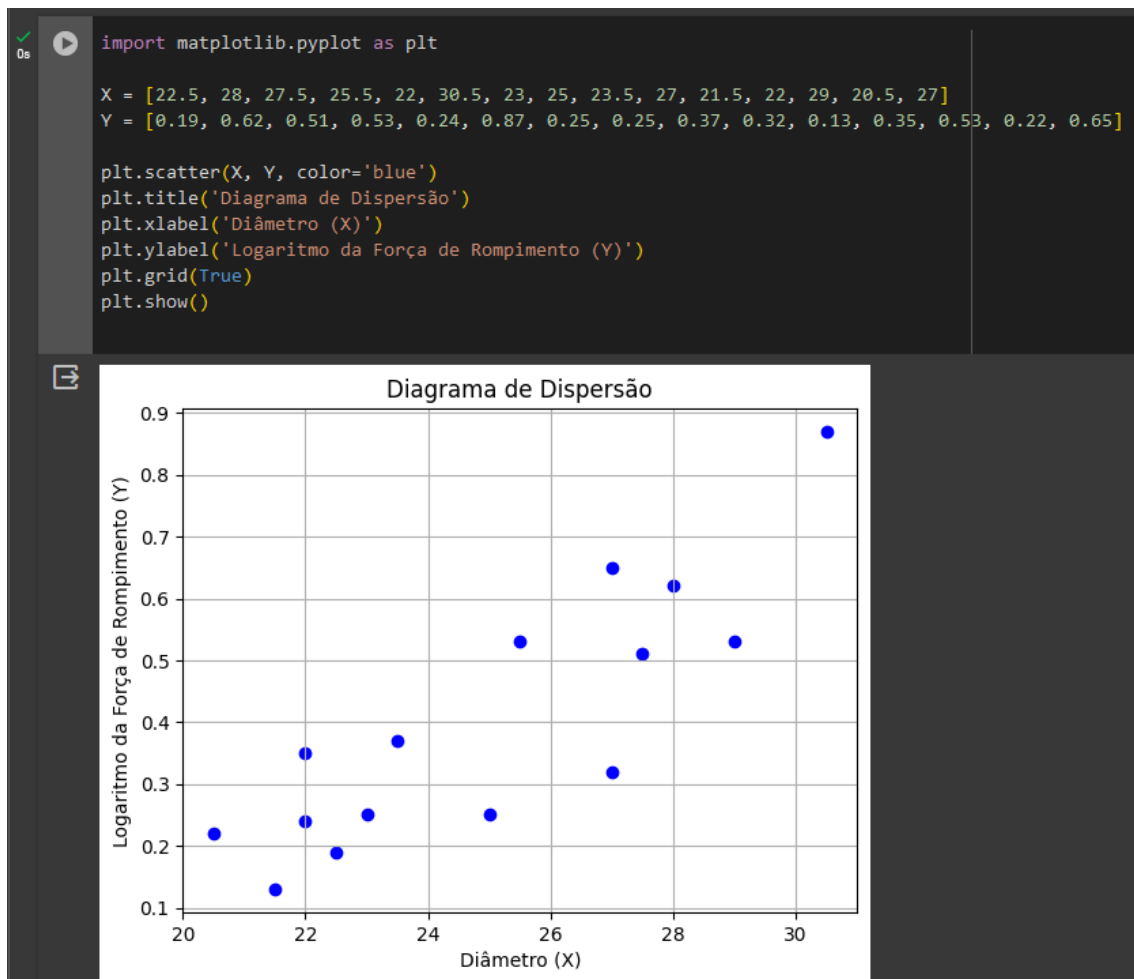
[6] Muitas vezes os dados devem ser transformados para que se obtenha uma relação linear entre a variável resposta e a variável explicativa. Os dados a seguir relacionam o diâmetro (X) de uma fibra e o logaritmo de base 10 (Y) da força de rompimento. Os dados são:

X 22,5 28 27,5 25,5 22 30,5 23 25 23,5 27 21,5 22 29 20,5 27

Y 0,19 0,62 0,51 0,53 0,24 0,87 0,25 0,25 0,37 0,32 0,13 0,35 0,53 0,22 0,65

- Construir o diagrama de dispersão
- Determinar a reta de regressão de Y em função de X, grafando-a
- Calcule o coeficiente de correlação e interprete o resultado.
- Em sua opinião, por que a reta no gráfico parece não estar condizente com o valor do coeficiente angular estimado?

a)



```

import numpy as np
import matplotlib.pyplot as plt

X = np.array([22.5, 28, 27.5, 25.5, 22, 30.5, 23, 25, 23.5, 27, 21.5, 22, 29, 20.5, 27])
Y = np.array([0.19, 0.62, 0.51, 0.53, 0.24, 0.87, 0.25, 0.25, 0.37, 0.32, 0.13, 0.35, 0.53, 0.22, 0.65])

slope, intercept = np.polyfit(X, Y, 1)

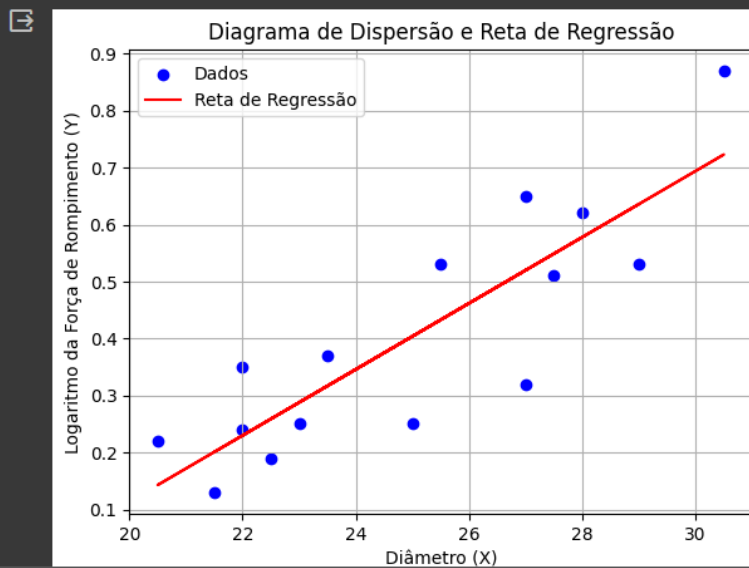
plt.scatter(X, Y, color='blue', label='Dados')

plt.plot(X, slope*X + intercept, color='red', label='Reta de Regressão')

plt.title('Diagrama de Dispersão e Reta de Regressão')
plt.xlabel('Diâmetro (X)')
plt.ylabel('Logaritmo da Força de Rompimento (Y)')
plt.grid(True)
plt.legend()
plt.show()

```

b)



c)

```

import numpy as np

X = np.array([22.5, 28, 27.5, 25.5, 22, 30.5, 23, 25, 23.5, 27, 21.5, 22, 29, 20.5, 27])
Y = np.array([0.19, 0.62, 0.51, 0.53, 0.24, 0.87, 0.25, 0.25, 0.37, 0.32, 0.13, 0.35, 0.53, 0.22, 0.65])

correlation_coefficient = np.corrcoef(X, Y)[0, 1]
print("Coeficiente de correlação entre X e Y:", correlation_coefficient)

```

Coeficiente de correlação entre X e Y: 0.8576407925257202

Interpretação do resultado:

- Um valor próximo de 1 indicaria uma forte correlação positiva, o que significa que à medida que X aumenta, Y também aumenta.
- Um valor próximo de -1 indicaria uma forte correlação negativa, o que significa que à medida que X aumenta, Y diminui.

- Um valor próximo de 0 indicaria pouca ou nenhuma correlação entre X e Y.

d)

Outliers nos dados podem distorcer a linha de regressão. Se a relação entre as variáveis não for estritamente linear, a linha de regressão pode não representar adequadamente essa relação. Mesmo com uma forte correlação entre as variáveis, pode haver uma grande variabilidade nos dados que não é capturada pela linha de regressão. O coeficiente angular estimado está sujeito a um certo grau de erro, já que é baseado em uma amostra dos dados. Erro de medição ou erro aleatório nos dados pode afetar a precisão da linha de regressão. Se os dados forem assimétricos ou distribuídos de forma não uniforme, a linha de regressão pode não capturar adequadamente a tendência dos dados.

É fundamental considerar a natureza dos dados e a adequação do modelo de regressão escolhido ao interpretar a relação entre as variáveis.

[7] Os dados abaixo correspondem às variáveis renda familiar e gasto com alimentação numa amostra de dez famílias, representadas em salários mínimos.

Renda Familiar (x): 3, 5, 10, 20, 30, 50, 70, 100, 150, 200

Gasto com alimentação (y): 1.5, 2.0, 6.0, 10.0, 15.0, 20.0, 25.0, 40.0, 60.0, 80.0

(a) Qual a previsão do gasto com alimentação para uma família com renda de 170 reais?

(b) Qual a previsão do gasto para famílias com renda, por exemplo 1.000 reais? Você acha esse valor

razoável? Por quê?

(c) Se você respondeu que o valor obtido em (b) não é razoável, encontre uma explicação para o ocorrido.

(Sugestão: interprete a natureza das variáveis X e Y e o comportamento de Y para grandes valores de X.)

a)

```
from sklearn.linear_model import LinearRegression

renda_familiar = [[3], [5], [10], [20], [30], [50], [70], [100], [150], [200]]
gasto_alimentacao = [1.5, 2.0, 6.0, 10.0, 15.0, 20.0, 25.0, 40.0, 60.0, 80.0]

modelo = LinearRegression()

modelo.fit(renda_familiar, gasto_alimentacao)

renda = [[170]]
previsao = modelo.predict(renda)

print("Previsão do gasto com alimentação para uma renda de 170 reais:", previsao[0])
```

Previsão do gasto com alimentação para uma renda de 170 reais: 67.55843522948787

b)

```
renda = [[1000]]
previsao = modelo.predict(renda)

print("Previsão do gasto com alimentação para uma renda de 1000 reais:", previsao[0])
```

Previsão do gasto com alimentação para uma renda de 1000 reais: 392.7467708271802

c)

Uma possível explicação para o valor previsto para uma renda de 1000 reais não ser razoável é considerar a natureza das variáveis e o comportamento dos dados para grandes valores de renda (X).

As variáveis X e Y representam, respectivamente, a renda familiar e o gasto com alimentação. No contexto do problema, a regressão linear presume uma relação linear entre essas duas variáveis, ou seja, o gasto com alimentação aumenta linearmente à medida que a renda familiar aumenta.

Entretanto, ao considerar valores muito grandes de renda (X), pode ocorrer que a relação linear não seja mais válida. Por exemplo, para famílias com renda muito alta, outros fatores podem começar a influenciar mais o gasto com alimentação, como estilo de vida, preferências alimentares mais sofisticadas, custo de vida em áreas urbanas caras, entre outros. Esses fatores podem não ser bem capturados por um modelo linear simples.

Além disso, pode acontecer que o gasto com alimentação não aumente linearmente com a renda familiar, mas sim de forma não linear. Por exemplo, famílias de baixa renda podem gastar uma porcentagem maior de sua renda com

alimentação do que famílias de alta renda, mas essa proporção pode diminuir à medida que a renda aumenta.

Portanto, para valores muito altos de renda, o modelo de regressão linear pode não ser mais adequado para prever o gasto com alimentação de maneira precisa, pois não consegue capturar nuances e padrões não lineares nos dados. Isso pode resultar em previsões que não são razoáveis para valores extremamente altos de renda.