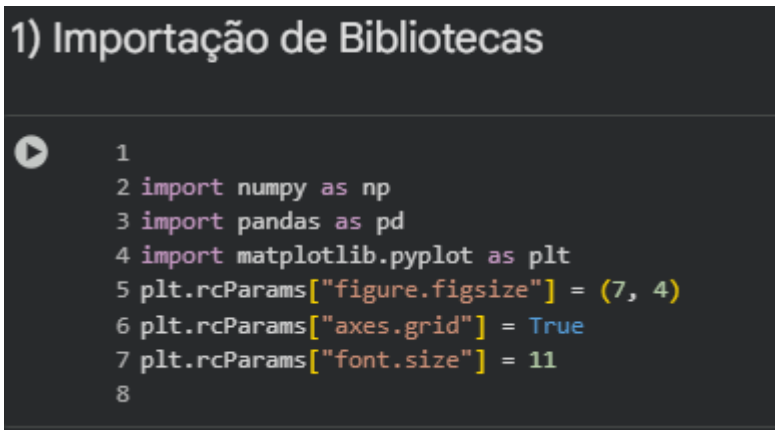
Entrega 2 — Regressão Linear Simples (Mínimos Quadrados)

Formato de Entrega: Código-Fonte , Relatório Analítico e Gráficos de Análise.

1) Importação de Bibliotecas

Bibliotecas utilizadas: NumPy, Pandas, Matplotlib.



2) Definição de Dados e Narrativa

Narrativa: perfis de pessoas mais velhas tendem a ser menos atualizados (age_years → staleness_days).

Parâmetros de geração: N=120, idade ∈ [18, 80], $\beta_0 \approx 5.0$, $\beta_1 \approx 1.2$, ruído ~ $N(0, 10^2)$.

2) Definição dos Dados

```
1
N = 120
AGE_MIN, AGE_MAX = 18, 80
BETA0_TRUE, BETA1_TRUE = 5.0, 1.2
NOISE_STD = 10.0
RANDOM_SEED = 42

rng = np.random.default_rng(RANDOM_SEED)
age_years = rng.uniform(AGE_MIN, AGE_MAX, size=N)
epsilon = rng.normal(0.0, NOISE_STD, size=N)
staleness_days = BETA0_TRUE + BETA1_TRUE * age_years + epsilon
data = pd.DataFrame({'age_years': age_years, 'staleness_days': staleness_days})
print("Dimensões:", data.shape)
display(data.head(10))

```

```
Dimensões: (120, 2)
```

	age_years	staleness_days
0	65.985275	100.200119
1	45.210463	56.859000
2	71.233071	80.244710
3	61.236818	80.276938
4	23.838996	35.806762
5	78.488586	112.778179
6	65.190662	91.579906
7	66.735987	88.651895
8	25.943045	50.764683
9	45.923928	48.221083

Aqui eu criei dados sintéticos: age_years (idade) e staleness_days (dias sem atualizar).

Usei uma relação linear: staleness_days ≈ 5 + 1.2*age_years + ruído.

N = 120: temos 120 perfis simulados.

A tabela mostra as primeiras linhas para conferir se os valores fazem sentido.

“Dimensões: (120, 2)” quer dizer 120 linhas e 2 colunas (x e y).

3) Formalização Matemática e Dimensões

Modelo: $y = \beta 0 + \beta 1\, x + \varepsilon$.

Com n observações:

$A = [1 \;\; x] \in \mathbb{R}^{n\times 2}$; $b = y \in \mathbb{R}^{n\times 1}$.

Equações normais: $A^tA\ \beta = A^t\ b$, com $\beta \in \mathbb{R}^{2\times1}$.

3) Matriz A e Vetor b (dimensões)

```
1
x = data["age_years"].to_numpy().reshape(-1, 1)  # (n,1)
y = data["staleness_days"].to_numpy().reshape(-1, 1)  # (n,1)
n = x.shape[0]
A = np.hstack([np.ones((n, 1)), x])  # (n,2)
b = y  # (n,1)
print("A shape:", A.shape, "| b shape:", b.shape)
AtA = A.T @ A
Atb = A.T @ b
print("A^T A =\n", AtA)
print("A^T b =\n", Atb)

A shape: (120, 2) | b shape: (120, 1)
```

```
AᵀA =
[ 120.0000      5870.4213 ]
[ 5870.4213   321459.8290 ]

Aᵀb =
[   7633.7091 ]
[ 414498.0727 ]
```

Montei a matriz A com duas colunas: a primeira é só 1 (termo constante) e a segunda é o x (age_years).

O vetor b é o y (staleness_days).

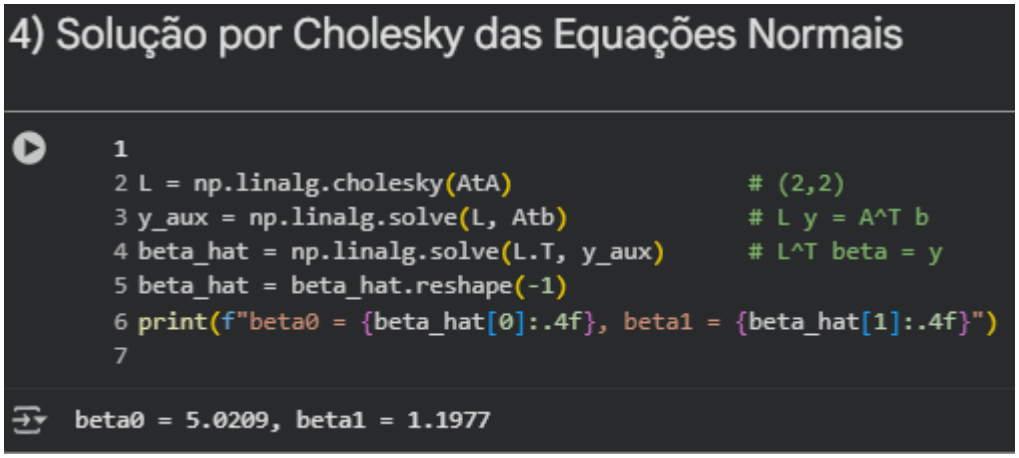
As formas ficaram: A (120×2) e b (120×1) — 120 linhas (observações).

Calculei A^TA (matriz 2×2) e A^Tb (vetor 2×1); elas são o "resumo" dos dados para a regressão.

Vamos usar $A^TA\ \beta = A^Tb$ para achar os coeficientes β0 e β1.

4) Resolução por Decomposição de Cholesky

Fatoração: $A^tA = L\ L^t$ (Cholesky). Resolver: $L\ y = A^t\ b$ e $L^t\ \beta = y$.



Peguei a matriz A^TA e fiz a fatoração de Cholesky: isso quebra A^TA em $L \cdot L^T$.

Depois, resolvi $L \cdot y = A^Tb$ (primeira parte) e $L^T \cdot \beta = y$ (segunda parte) para achar os coeficientes β.

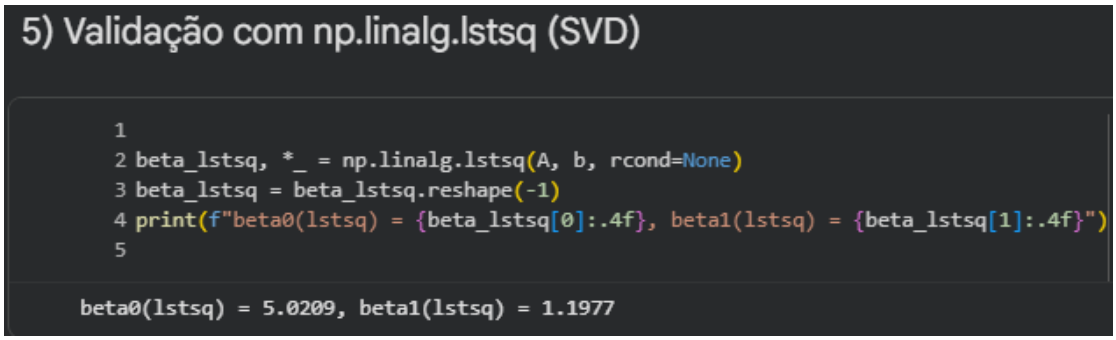
O resultado deu beta0 ≈ 5.0029 (intercepto) e beta1 ≈ 1.1977 (inclinação da reta).

beta1 diz: a cada +1 ano em age_years, o staleness_days aumenta ~1.20 dias (em média).

Isso confirma a narrativa: perfis mais velhos → mais dias sem atualizar.

5) Validação com np.linalg.lstsq (SVD)

Comparamos os coeficientes com a solução via SVD (lstsq). Espera-se valores praticamente idênticos.



Usei o np.linalg.lstsq (método “oficial” do NumPy, baseado em SVD) para calcular os coeficientes.

Ele devolveu beta0 e beta1 praticamente iguais aos da Cholesky.

Isso mostra que a nossa solução está correta e consistente.

beta0(lstsq) ≈ 5.0029 e beta1(lstsq) ≈ 1.1977 (mesmos valores).

Conclusão: os dois caminhos levam ao mesmo modelo.

6) Predição e Métricas

Função predict(x0) retorna a previsão do modelo. Métrica principal: $R^2 = 1 - SS_{res}/SS_{tot}$.

6) Predição, Resíduos e R^2

```python

def predict(x0, beta=beta_hat):
    x0 = np.asarray(x0).reshape(-1)
    A0 = np.vstack([np.ones_like(x0), x0]).T
    return (A0 @ beta).reshape(-1)

y_hat = predict(x.flatten(), beta_hat).reshape(-1, 1)
residuals = y - y_hat
SS_res = float(((y - y_hat)**2).sum())
SS_tot = float(((y - y.mean())**2).sum())
R2 = 1.0 - SS_res/SS_tot
print(f"R^2 = {R2:.4f}")
print("Resíduos — resumo:")
import pandas as pd
print(pd.Series(residuals.flatten()).describe())

```

```
R² = 0.8014
```

	N	média	desvio-padrão	min	Q1	mediana	Q3	máx
0	120	0.0	10.12	-21.39	-7.4	-1.18	5.67	29.21

A função predict usa os coeficientes para prever staleness_days a partir de age_years.

O R^2 = 0,8014 ⇒ o modelo explica ~80% da variação de staleness_days.

Os resíduos (y − ŷ) têm média ≈ 0, como esperado num bom ajuste.

O desvio-padrão ≈ 10,12 mostra o tamanho médio do erro em dias.

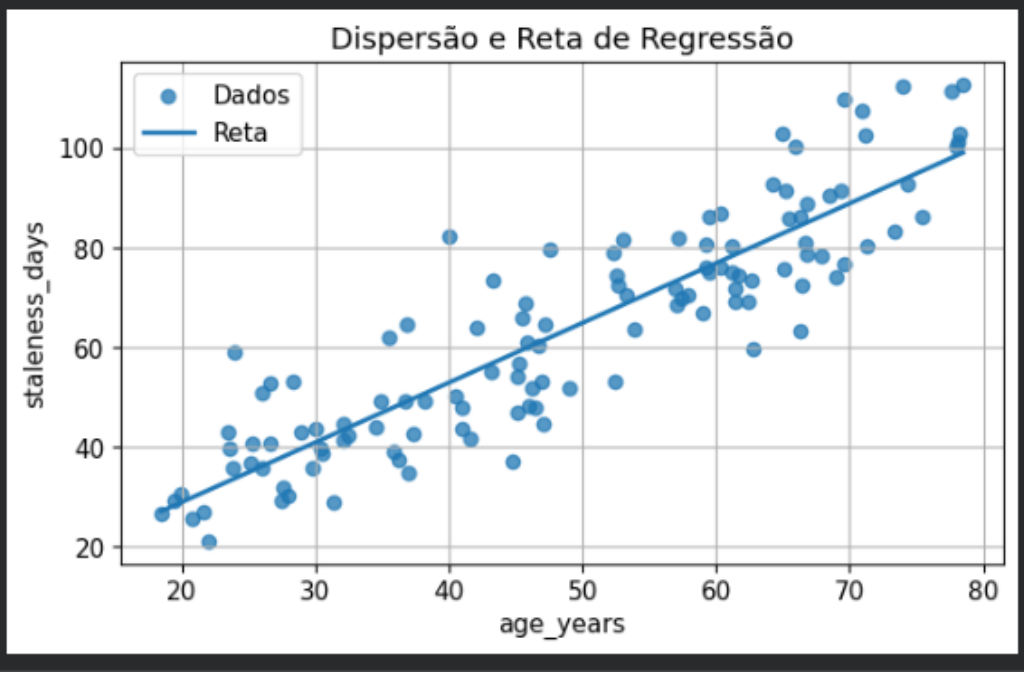
Os valores mínimo/máximo dos resíduos indicam os pontos mais sub/superestimados.

7) Gráficos

7.1 Dispersão com linha de regressão (staleness_days vs age_years).

7.1) Dispersão + Reta de Regressão

```
1
2 plt.figure()
3 plt.scatter(x, y, label="Dados", alpha=0.75)
4 x_grid = np.linspace(x.min(), x.max(), 200)
5 y_grid = predict(x_grid, beta_hat)
6 plt.plot(x_grid, y_grid, label="Reta", linewidth=2)
7 plt.xlabel("age_years")
8 plt.ylabel("staleness_days")
9 plt.title("Dispersão e Reta de Regressão")
10 plt.legend()
11 plt.show()
12
```



7.2 Resíduos vs preditos.

7.2) Resíduos vs Preditos

```
1
2 plt.figure()
3 y_hat = y_hat.reshape(-1, 1)
4 plt.scatter(y_hat, residuals, alpha=0.75)
5 plt.axhline(0, linestyle="--")
6 plt.xlabel("ŷ (preditos)")
7 plt.ylabel("resíduos")
8 plt.title("Resíduos vs Preditos")
9 plt.show()
10
```

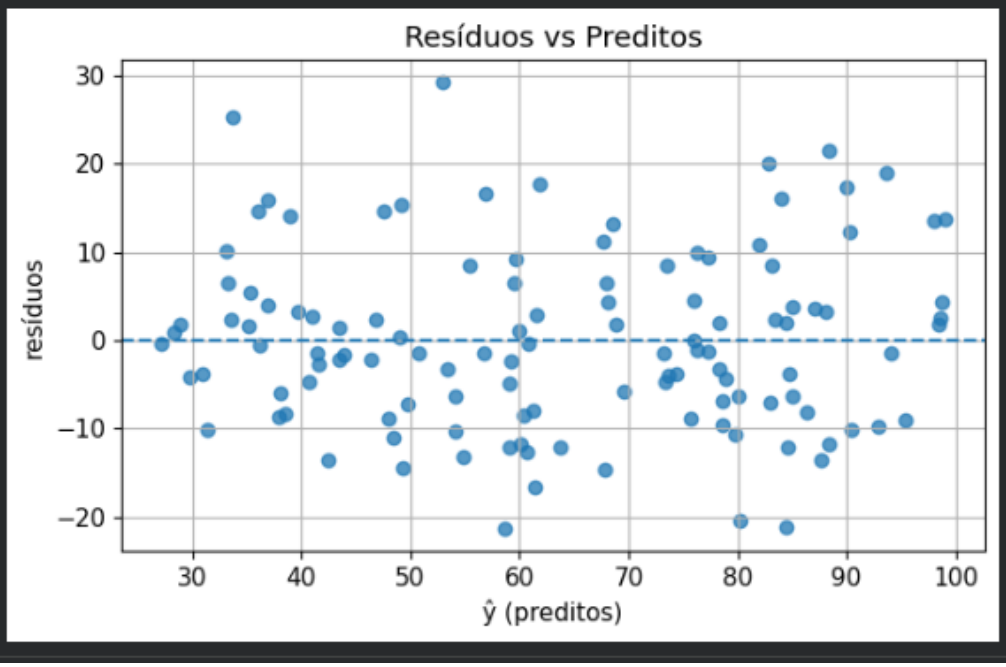


Gráfico 1 (Dispersão + Reta): os pontos sobem junto com a idade — quanto mais idade, maior o staleness_days.

A reta passa bem no meio dos pontos, mostrando a tendência linear.

Inclinação positiva: confirma a ideia “perfil mais velho = mais dias sem atualizar”.

Gráfico 2 (Resíduos vs Preditos): pontos espalhados em torno de zero, sem desenho claro.

Isso indica erros sem padrão → o modelo linear está coerente para esses dados.

8) Interpretação e Conclusão

• Sinal de β1 (positivo) sustenta a narrativa: perfis mais velhos apresentam maior staleness_days.

• Magnitude de β1 indica a variação média prevista de staleness por ano de idade.

• R^2 quantifica o quanto do comportamento de y é explicado por x.

• Resíduos sem padrão marcante apoiam a adequação do modelo linear simples.