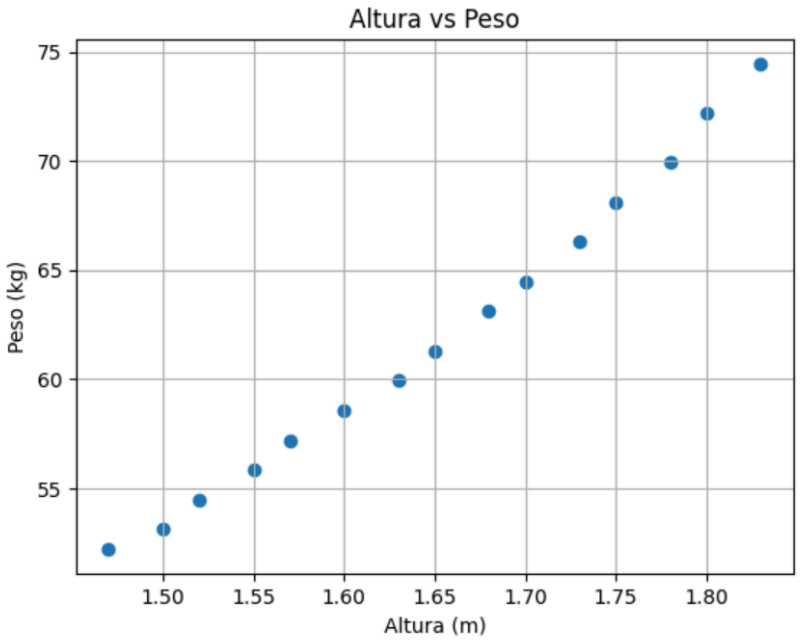
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

EXERCÍCIO
REGRESSÃO LINEAR

ALUNO: FELIPPE VELOSO MARINHO
MATRÍCULA: 2021072260
DISCIPLINA: APRENDIZADO DE MÁQUINA

Objetivo: Criar um modelo de regressão Linear capaz de prever o peso de uma pessoa através de sua altura.

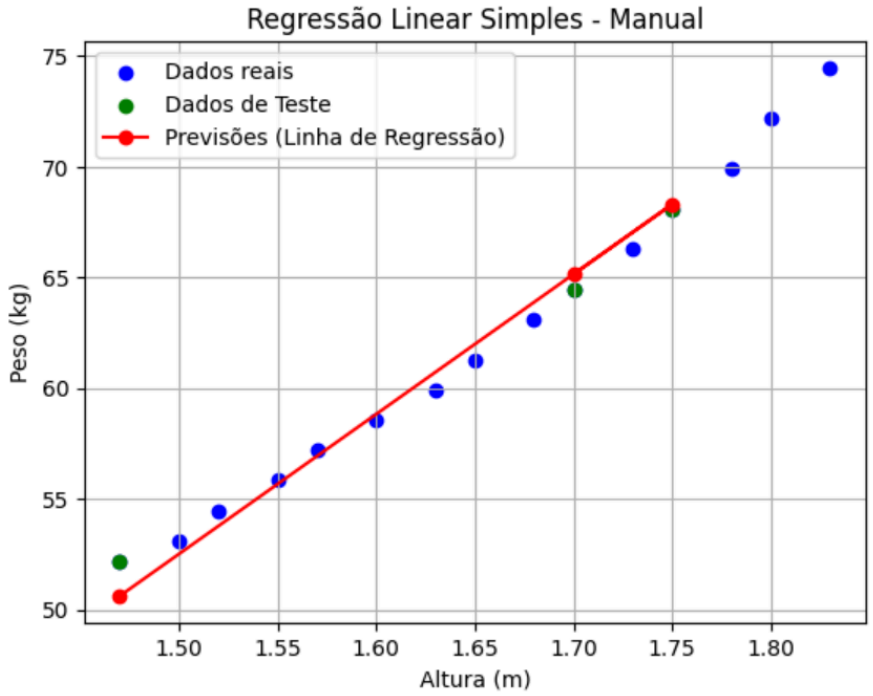
Após a criação dos dados fictícios plotados abaixo



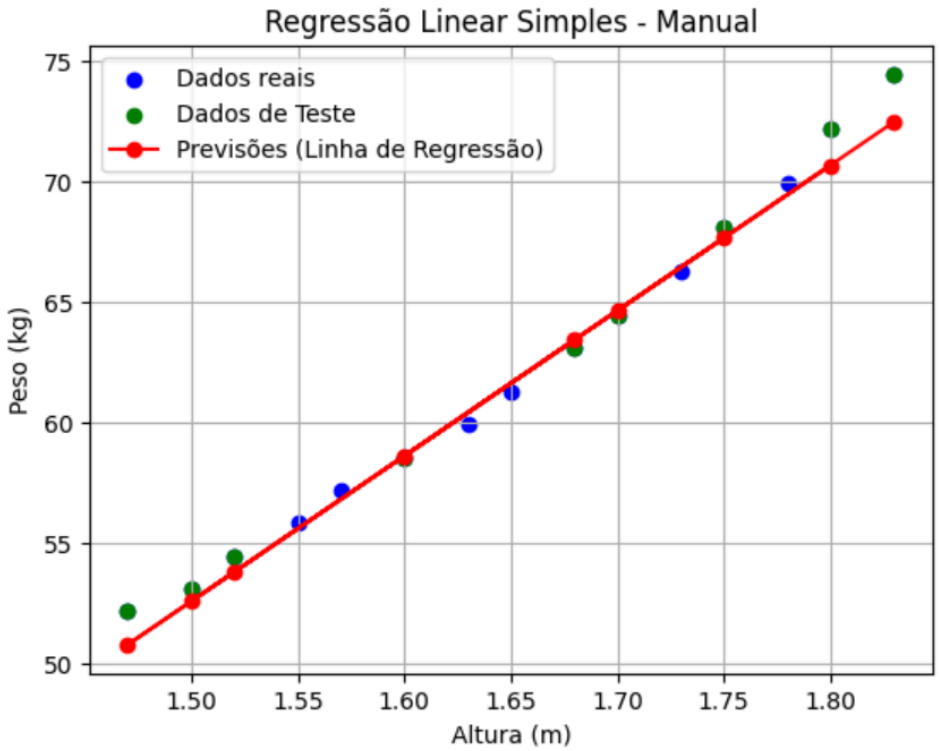
A criação do modelo pode ser feita de várias maneiras. Para a atividade, optei por criar uma classe simples com dois métodos além do construtor. Dentre eles a função fit, responsável por ajustar o modelo aos dados de treinamento e o predict, responsável em realizar ass previsões com base no modelo linear:

$$y = \beta_0 + \beta_1 x$$

De maneira a explorar mais a atividade, me inspirei no modelo pronto do sk.learn para separar em dados de teste e de treino. Devido a quantidade limitada de dados (15 variáveis para cada) a divisão não poderia ser muito adequada. Separando em 20% de dados para treino, não temos dados suficiente para estimar os mínimos e máximos do conjunto de dados. Sendo assim, nossa reta de previsão (Linha de regressão) acaba ficando limitada, não cobrindo todo os dados de teste. Aumentando a porcentagem de dados de treino para 60%, temos uma estimativa maior porém a quantidade de dados de teste fica muito limitada para nosso modelo.



Modelo de Regressão com 20% de dados para teste



Modelo de Regressão com 60% de dados para teste

Com o modelo pronto é possível estimar o peso de qualquer altura existente. Sendo assim, qual seria o peso do Cristo Redentor se ele do nada virasse uma pessoa e saísse andando que nem uma propaganda que ele saia jogando bola uma vez?

```
# Previsão para uma altura específica
height_input = 38 # Altura do crito redentor
predicted_weight = model.predict(np.array([height_input]))
print(f'A previsão de peso para o Cristo Redentor é de {predicted_weight[0]:.2f} kg.')
```

```
A previsão de peso para o Cristo Redentor é de 2356.83 kg.
```

Calculando RSE e R^2

O RSE mede a quantidade média que as previsões do modelo diferem dos valores reais. Quanto menor o RSE, melhor o modelo se ajusta aos dados. Ele é calculado da seguinte forma:

- n é o número de observações,
- yi são os valores reais (pesos observados),
- ^yi são os valores previstos pelo modelo.

$$RSE = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Em código, nós pegamos o peso predito, calculamos os resíduos e verificamos através da fórmula. Em sequência realizamos o cálculo do R^2:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Onde:

- y_i são os valores reais,
- $\hat{y}_i$ são os valores preditos pelo modelo,
- n é o número de observações.
- $\overline{y}$ é a média dos valores reais.

Com ambas informações conseguimos verificar o quanto o modelo se ajusta bem aos dados e sabemos a proporção da variabilidade das nossas variáveis.

```
# Cálculo do RSE
predicted_weight = model.predict(height)
residuals = weight - predicted_weight  # Erros
RSE = np.sqrt(np.sum(residuals**2) / (n - 2))

# Cálculo do R^2
SS_res = np.sum(residuals**2)  # Soma dos quadrados dos resíduos
SS_tot = np.sum((weight - np.mean(weight))**2)  # Soma total dos quadrados
R_squared = 1 - (SS_res / SS_tot)

# Resultados
print(f'Erro Padrão Residual (RSE): {RSE:.3f}')
print(f'Coeficiente de Determinação (R^2): {R_squared:.3f}')
```

```
Erro Padrão Residual (RSE): 0.792
Coeficiente de Determinação (R^2): 0.988
```

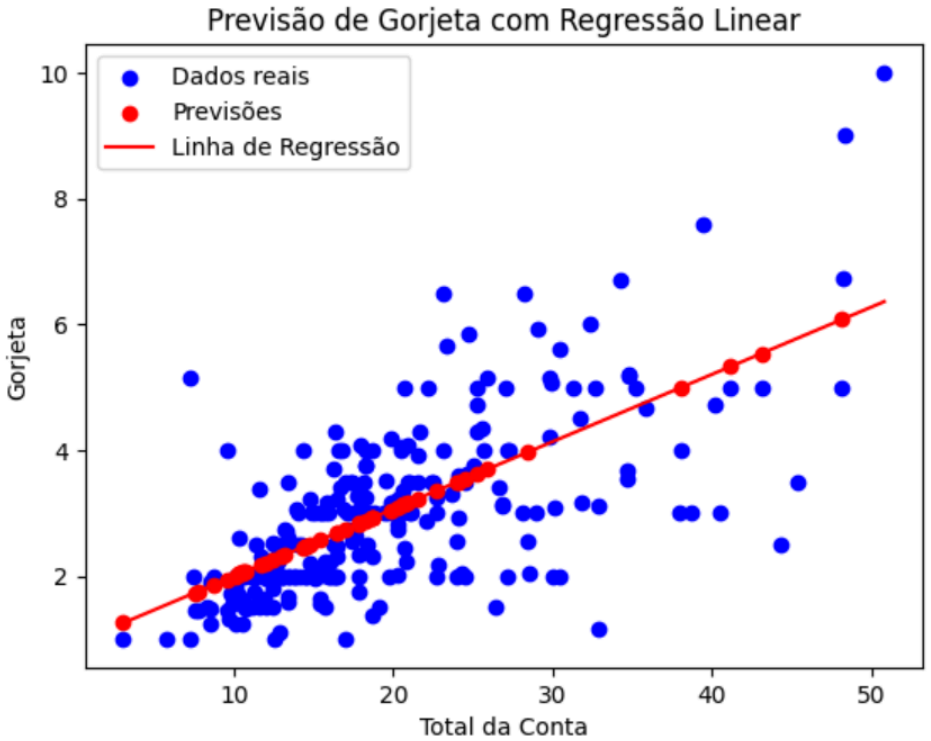
É possível verificar que nosso modelo tem um erro residual relativamente baixo, considerando que nossos dados variam de 52 a 75 kg, temos uma média consideravelmente boa.

Seguindo, sendo um coeficiente próximo a 1 um ótimo sinal de ajuste dos dados, temos um valor muito bom. Ou seja, o valor de 0.988 indica que uma grande parte da variabilidade nos dados de peso é explicada pela altura

Objetivo: Utilizando o dataset tips, que contém dados sobre gorjetas e pode ser usado para prever o valor da gorjeta com base em variáveis como total da conta.

Para essa atividade, foi feita uma pequena alteração no modelo criado com a adição de funções para conversão dos dados de DataFrame vindo da Sk.learn para arrays NumPy.

Aplicando nosso modelo chegamos em:



Repetindo os processos para calcular o Erro Padrão Residual (RSE) e o Coeficiente de Determinação (R^2):

```
Erro Padrão Residual (RSE): 0.770
Coeficiente de Determinação (R²): 0.545
```

Os resultados indicam que, embora haja alguma relação entre o total da conta e a gorjeta, o modelo pode ser melhorado. Um R^2 de 0.545 sugere que outras variáveis podem ser relevantes, e um RSE de 0.770 é aceitável, mas não ótimo.

Por fim, verificando quanto o garçon irá ganhar de gorjeta se o total da conta for de 80U$:

```
total_bill_input = 80 # Previsão para um total da conta de 80 U$
predicted_tip = model.predict(pd.DataFrame([[total_bill_input]]))

print(f'A previsão de gorjeta para uma conta de U${total_bill_input} é de U${predicted_tip[0]:.2f}.')
```

```
A previsão de gorjeta para uma conta de U$80 é de U$9.48.
```