

## Relatório Atividade 1 – Regressão Linear

Nome: Wallesson Cavalcante da Silva

Curso: Ciência da Computação

Matrícula: 397670

### Considerações iniciais:

Tive um pouco de dificuldade no início por não conhecer a linguagem Python e das bibliotecas que poderia usar. Com isso justifico um pouco a deselegância do código. Para realizar a atividade recorri a pedir ajuda de alguns colegas que estavam mais habituados com a linguagem e com as bibliotecas usadas.

Decidi comentar também como elaborei o algoritmo.

### Implementação da Regressão Linear Simples:

Nesse trecho do código basicamente carreguei os dados do arquivo para a variável “base\_dados” e fiz a alteração dos valores que X iria receber para os 3 testes manualmente. (Como expliquei, foquei mais em resolver o problema e colher os resultados. Porém acredito que se tivesse pensado mais teria feito algo mais agradável).

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 base_dados = np.loadtxt('Advertising.csv', skiprows=1, delimiter=',')
5 X = base_dados[:,1]
6 Y = base_dados[:,4]
7
```

A seguir, criei a classe “Regressao\_Linear\_Simples”. Nessa classe tratei a implementação dos cálculos para encontrar os valores dos betas (B0 e B1) que são a chave principal para gerar a função linear que definiria a regressão.

Com isso, minha função “fit” recebe os dados coletados em “X” e “Y” (atualizados da base\_dados), onde realizo uma série de operações matemáticas simples para encontrar os valores estimados para os betas que se dá pela função geral:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

Primeiramente utilizei a função mean para armazenar em Media\_X o valor da média das amostras X e em Media\_Y o valor da média das observações Y.

Logo depois, tendo em vista resolver primeiramente a parte superior da formula supracitada, realizo a subtração das diferenças entre as amostras X com sua media e armazeno em Difer\_X, e das observações Y com sua media e armazeno em Difer\_Y.

Com esse resultado realizo uma multiplicação entre os valores armazenados em Difer\_X e Difer\_Y para obter o resultado do somatório superior da formula onde armazeno esse valor em Difer\_X\_vezes\_Y. E realizo uma multiplicação de Difer\_X por ele mesmo, para obter seu valor que deve ser ao quadrado seguindo a formula, armazenando ele em Difer\_X\_vezes\_X.

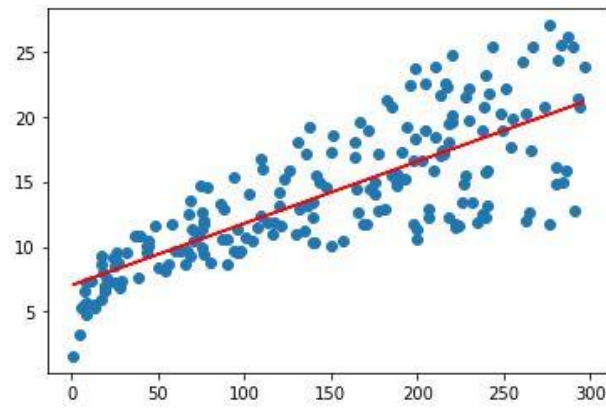
Para terminar, armazeno em "Superior" e "Inferior" o resultado obtido pela função sum aplicado na variável Difer\_X\_vezes\_Y e Difer\_X\_vezes\_X respectivamente.

Com isso, criei a função predict para montar equação linear. Onde B1 recebe Superior/Inferior e B0 recebe  $\text{Media\_Y} - (B1 * \text{Media\_X})$ .

```
8 class Regressao_Linar_Simples(object):
9     def __init__(self):
10         self.B0 = None
11         self.B1 = None
12
13     def fit(self, X,Y):
14         self.X = np.array(X)
15         self.Y = np.array(Y)
16
17         Media_X = X.mean()
18         Media_Y = Y.mean()
19
20         Difer_X = X - Media_X
21         Difer_Y = Y - Media_Y
22
23         Difer_X_vezes_Y = Difer_X*Difer_Y
24         Difer_X_vezes_X = Difer_X*Difer_X
25
26         Superior = sum(Difer_X_vezes_Y)
27         Inferior = sum(Difer_X_vezes_X)
28
29         self.B1 = Superior/Inferior
30         self.B0 = Media_Y-(self.B1* Media_X)
31
32     def predict(self, X):
33         return self.B0+(self.B1*X)
34
35
```

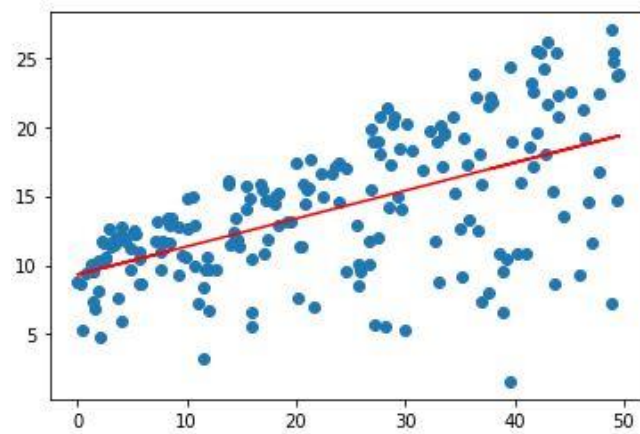
## 2) Resultados

A/B)

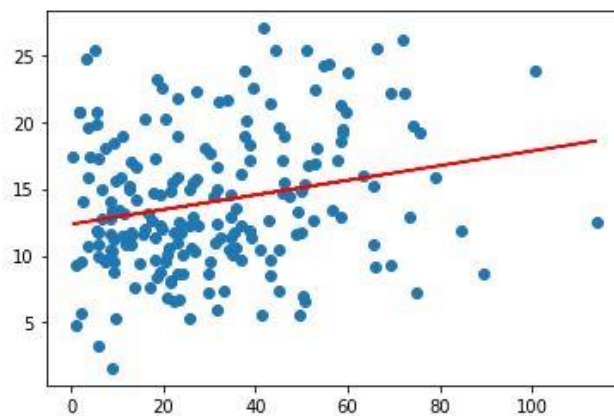


In [3]:

**Resultado obtido com as amostras de dados de TV**



**Resultado obtido com as amostras de dados de Radio**



### Resultado obtido com as amostras de dados de Jornal

```
In [31]: runfile('C:/Users/Wallesson/Desktop/ML/
Regressao_Linear_Simples.py', wdir='C:/Users/Wallesson/
Desktop/ML')
RSS_Simples: 2102.5305831313517
RSS_Simples: 3.258656368650463
RSS_Simples: 0.611875050850071
RSS_Simples: 2.5498060389274873
```

```
Beta_0: 7.032593549127704
Beta_1: 0.04753664043301969
```

### Funções aplicadas às amostras de TV

```
Desktop/ML')
RSS_Simples: 3618.479549025088
RSS_Simples: 4.27494435490106
RSS_Simples: 0.33203245544529525
RSS_Simples: 3.3202187976420663
```

```
Beta_0: 9.311638095158287
Beta_1: 0.20249578339243954
```

### Funções aplicadas às amostras de Radio

```
Desktop/ML')
RSS_Simples: 5134.804544111939
RSS_Simples: 5.092480366520192
RSS_Simples: 0.05212044544430516
RSS_Simples: 4.146559743838489
```

```
Beta_0: 12.351407069278162
Beta_1: 0.05469309847227332
```

### Funções aplicadas às amostras de Jornal

**OBS:** Na função RSS deu um valor bastante diferente da dos meus colegas. Acredito que errei na elaboração de alguma formula matemática, porém não encontrei o erro para corrigi-lo. O que estranhei mais foi o fato de as outras darem certo.

## C) Conclusões sobre as medidas de erros e os parâmetros

Não sei se minha análise está correta, porém observando os dados obtidos e fazendo uma comparação dos gráficos também, percebi que quando é utilizada a amostra de TV o gráfico tende a ter um comportamento mais parecido com a função linear.