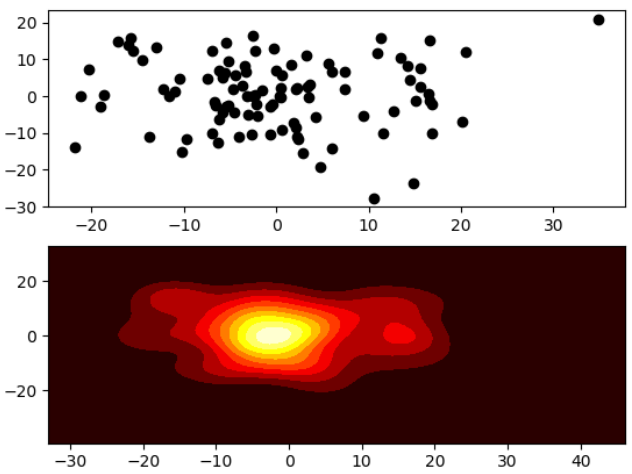
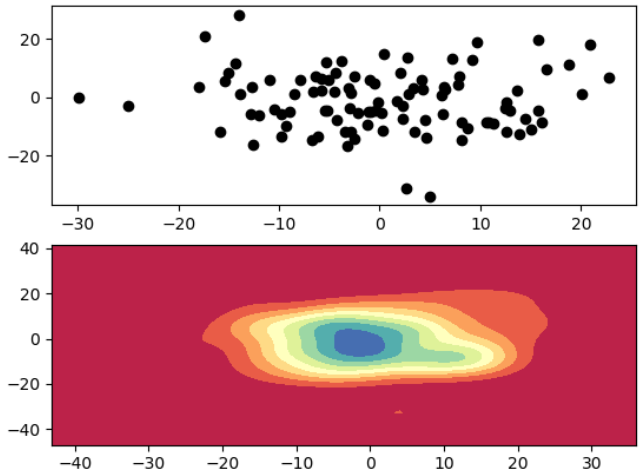
Programando em Python

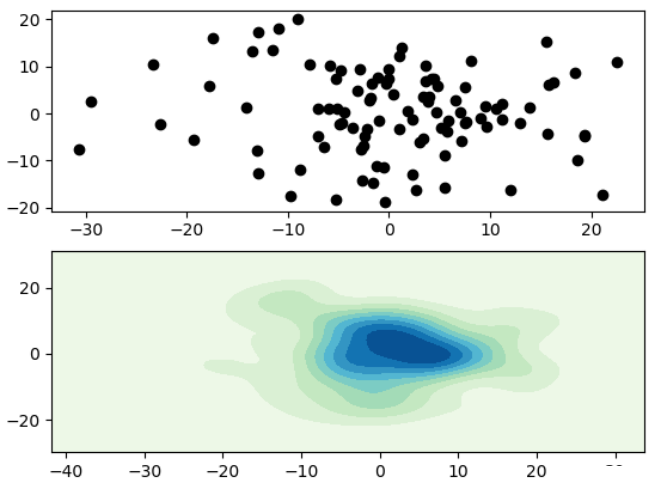
Biblioteca Seaborn

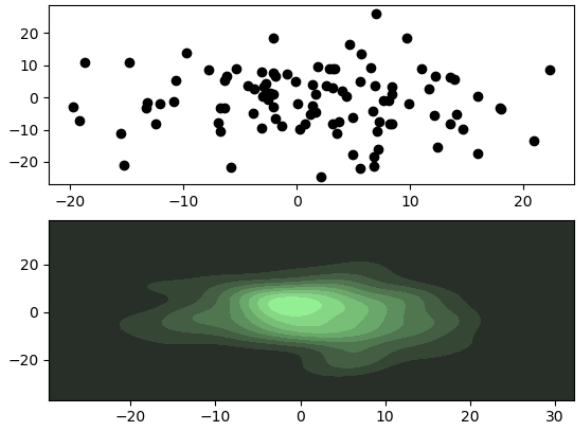
Prof. Dr. Marco Antonio Leonel Caetano

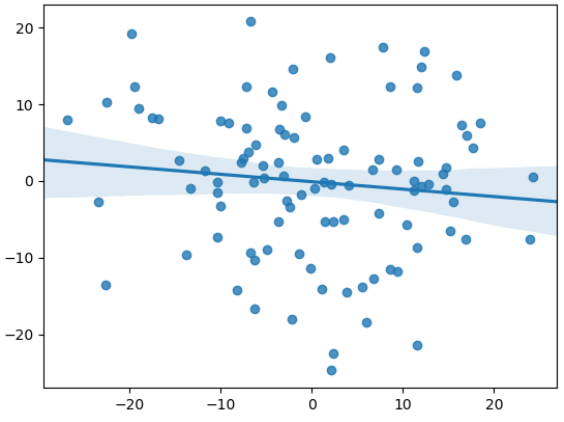
Seaborn

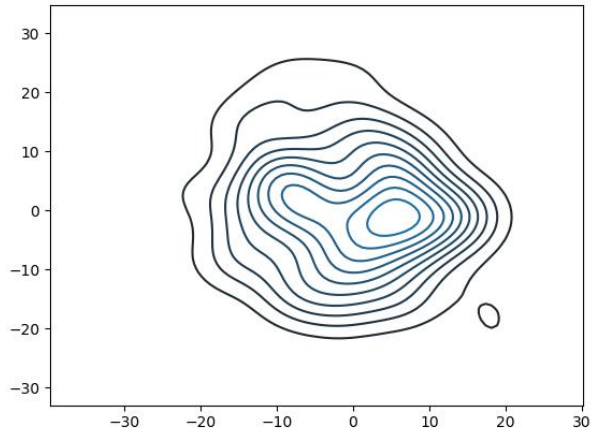


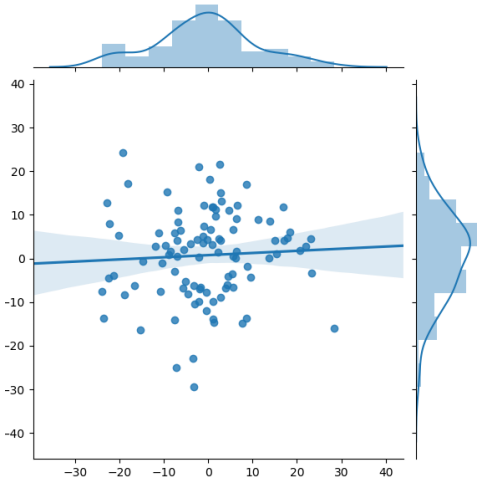






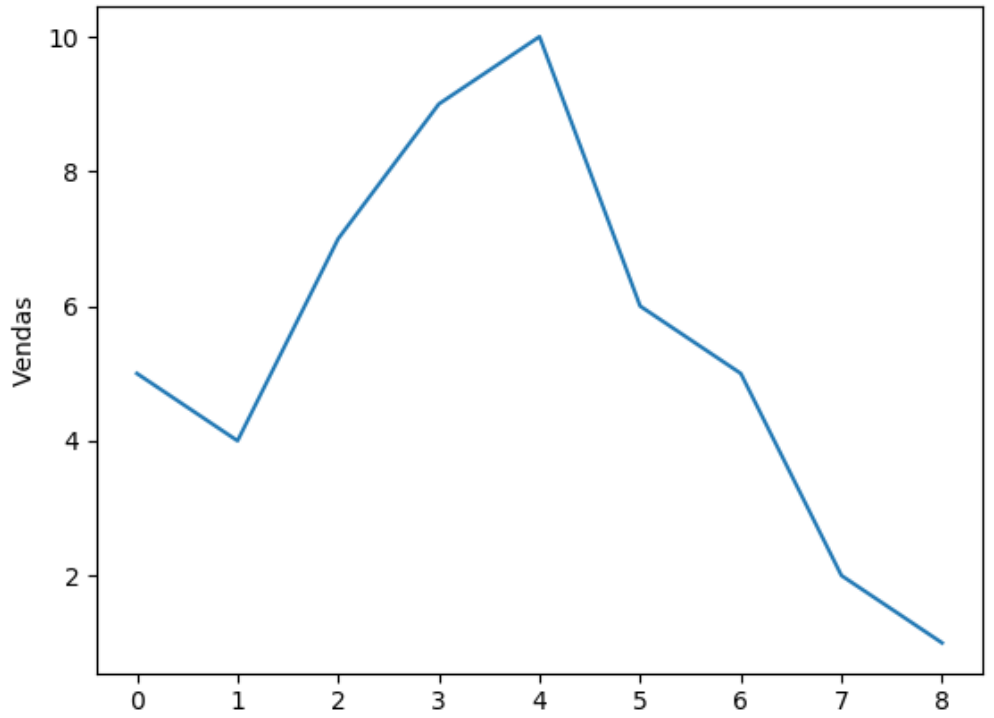






lineplot (gráfico de linhas)

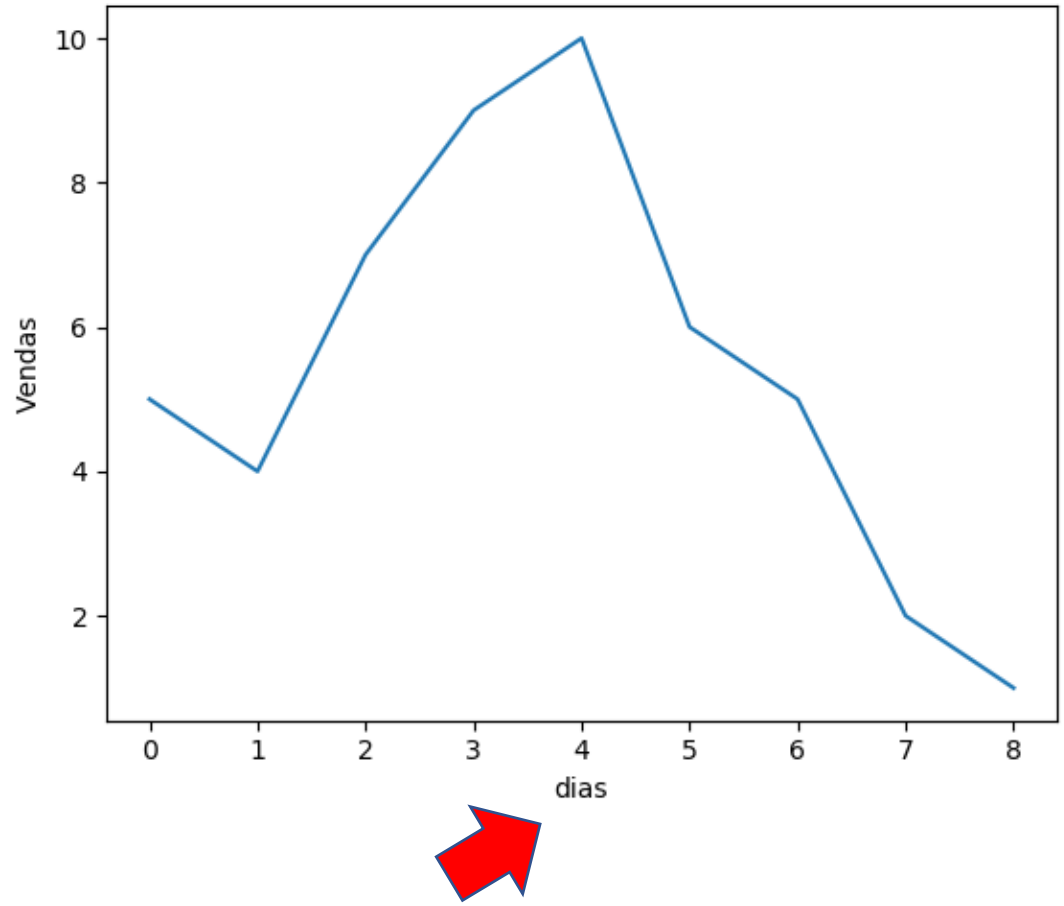
```
#biblioteca seaborn
import seaborn as sns
import pandas as pd

df=pd.DataFrame({'Vendas':[5,4,7,9,10,6,5,2,1],
                 'Lucro':[100,90,110,120,150,60,40,30,10]})

sns.lineplot(x=df.index,y=df['Vendas'])
```



lineplot (gráfico de linhas)

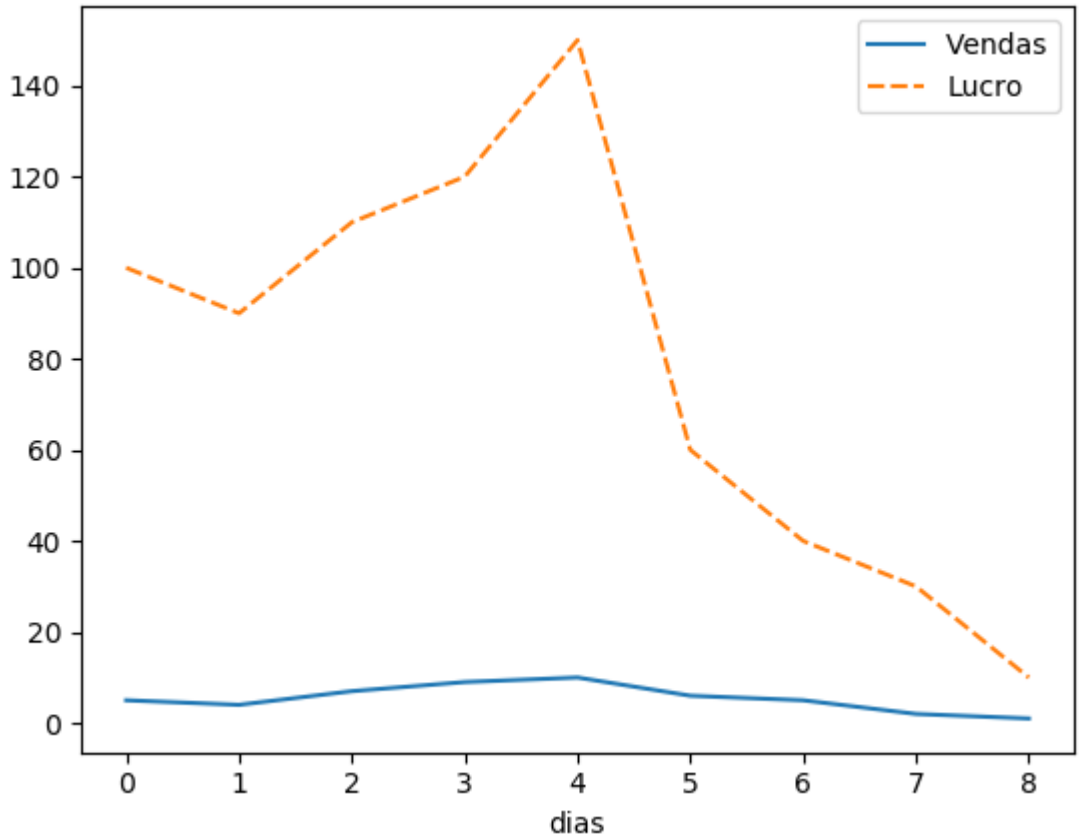
Pode-se utilizar a Matplotlib para formatar a figura com títulos, dos eixos, subplots, etc

```
#biblioteca seaborn
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as fig

df=pd.DataFrame({'Vendas':[5,4,7,9,10,6,5,2,1],
                 'Lucro':[100,90,110,120,150,60,40,30,10]})

sns.lineplot(x=df.index,y=df['Vendas'])
fig.xlabel('dias')
```



lineplot (Múltiplas linhas)

```python
#biblioteca seaborn
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as fig

df=pd.DataFrame({'Vendas':[5,4,7,9,10,6,5,2,1],
                 'Lucro':[100,90,110,120,150,60,40,30,10]})

sns.lineplot(data=df)
fig.xlabel('dias')
```



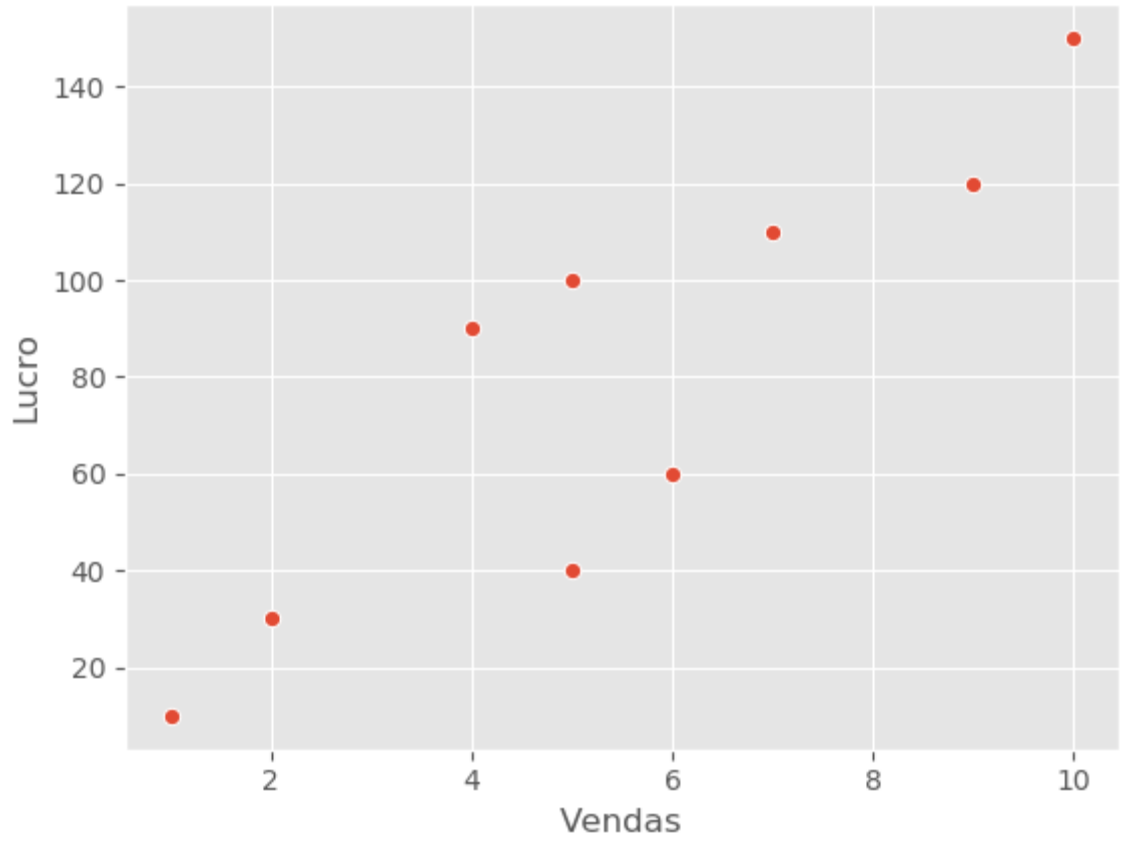
data=df

assume todo o dataframe
onde as colunas são diversas variáveis
a serem plotadas

scatterplot (var1 x var2)

```
#biblioteca seaborn
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as fig

df=pd.DataFrame({'Vendas':[5,4,7,9,10,6,5,2,1],
                 'Lucro':[100,90,110,120,150,60,40,30,10]})

fig.style.use('ggplot')
sns.scatterplot(data=df,x='Vendas',y='Lucro')
```

Deixa tom cinza no gráfico

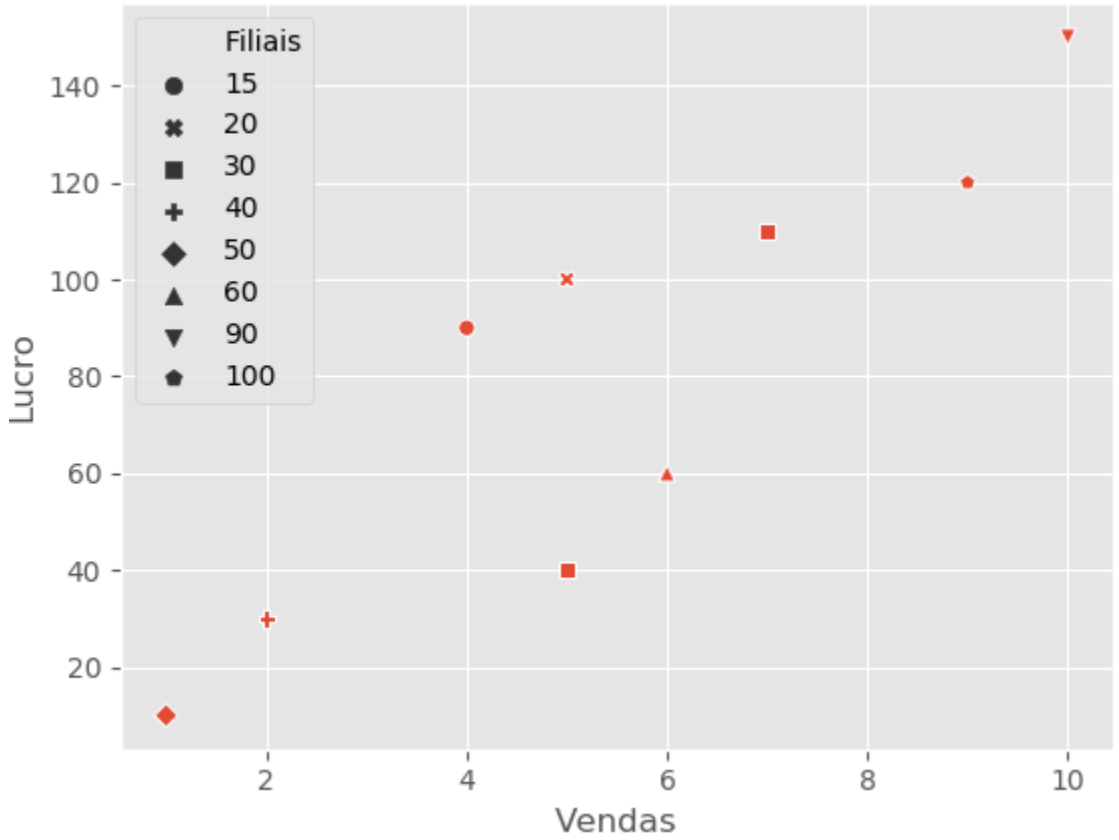


O scatter é um gráfico de pontos para relacionar Mudanças de uma variável em relação a outra

scatterplot (estilo dependendo de uma variável)

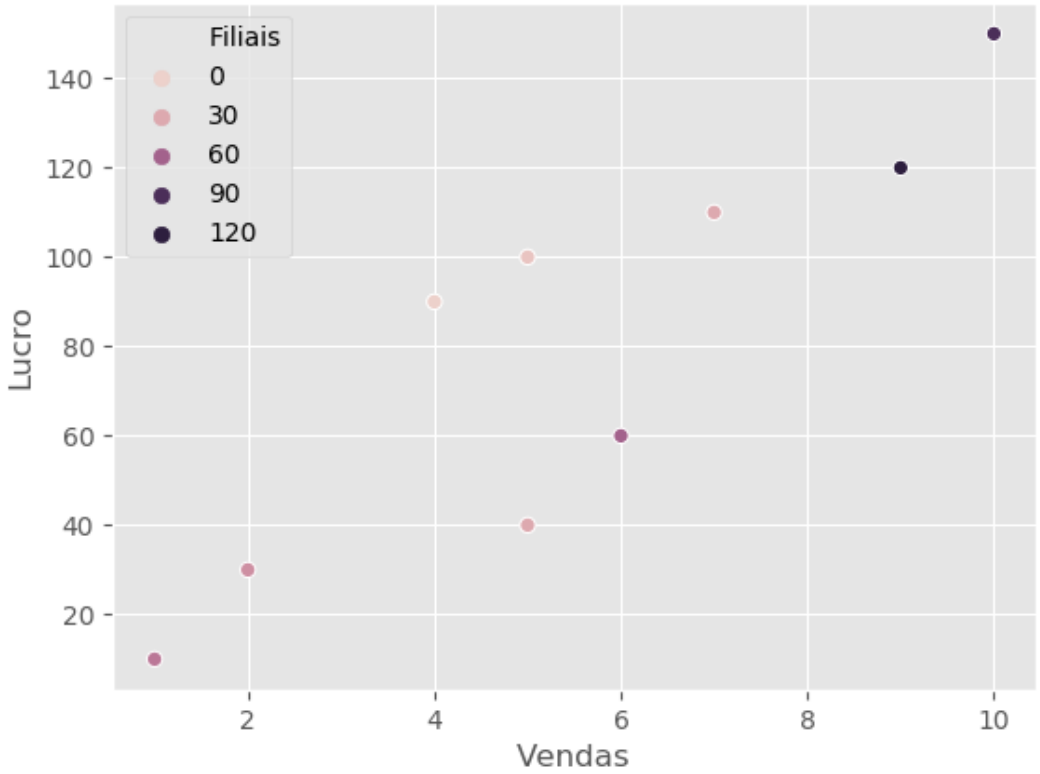
```
#biblioteca seaborn
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as fig

df=pd.DataFrame({'Vendas':[5,4,7,9,10,6,5,2,1],
                 'Lucro':[100,90,110,120,150,60,40,30,10],
                 'Filiais':[20,15,30,100,90,60,30,40,50]})

fig.style.use('ggplot')
sns.scatterplot(data=df,x='Vendas',y='Lucro',style='Filiais')
```

Variável não plotada no scatter, apenas usada para indicar style dos pontos



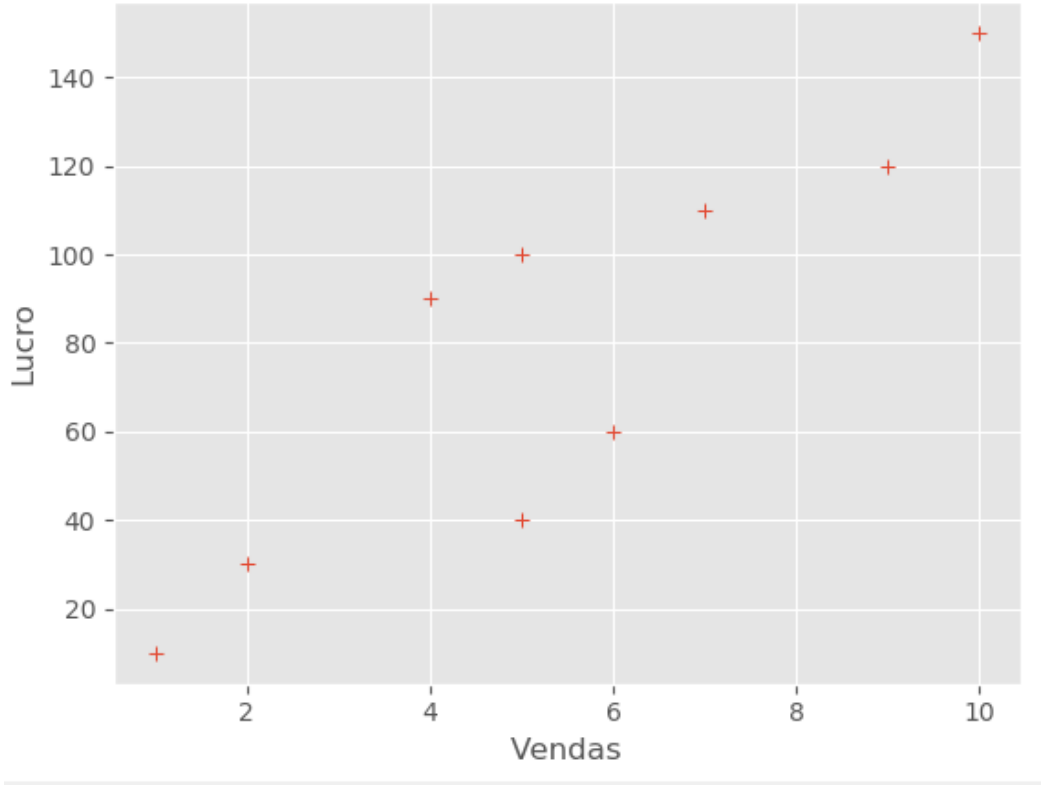
scatterplot (intensidade da cor dependendo de uma variável)

```
#biblioteca seaborn
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as fig

df=pd.DataFrame({'Vendas':[5,4,7,9,10,6,5,2,1],
                 'Lucro':[100,90,110,120,150,60,40,30,10],
                 'Filiais':[20,15,30,100,90,60,30,40,50]})

fig.style.use('ggplot')
sns.scatterplot(data=df,x='Vendas',y='Lucro',hue='Filiais')
```



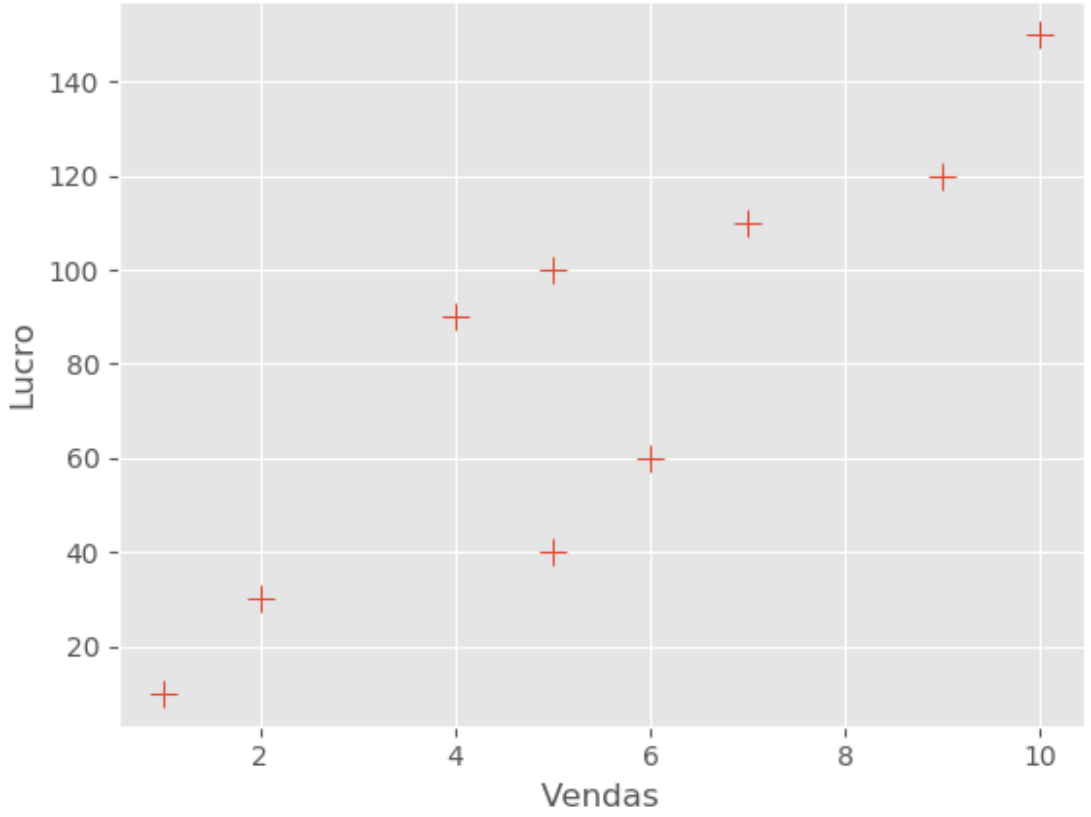
scatterplot (escolhendo o tipo do marcador)

```
1 #biblioteca seaborn
2 import seaborn as sns
3 import pandas as pd
4 import matplotlib.pyplot as fig
5
6 df=pd.DataFrame({'Vendas':[5,4,7,9,10,6,5,2,1],
7                  'Lucro':[100,90,110,120,150,60,40,30,10],
8                  'Filiais':[20,15,30,100,90,60,30,40,50]})
9
10 fig.style.use('ggplot')
11 sns.scatterplot(data=df,x='Vendas',y='Lucro',marker='+')
```



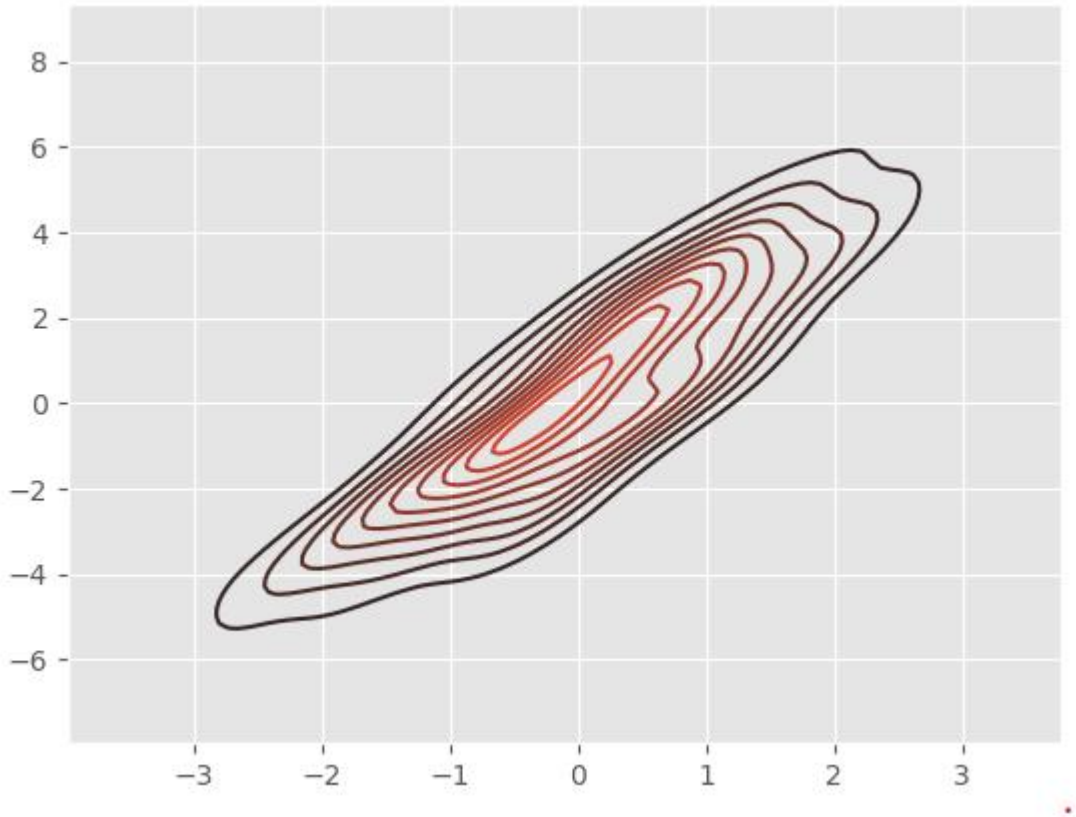
scatterplot (tamanho do marcador s=)

```python
#biblioteca seaborn
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as fig

df=pd.DataFrame({'Vendas':[5,4,7,9,10,6,5,2,1],
                 'Lucro':[100,90,110,120,150,60,40,30,10],
                 'Filiais':[20,15,30,100,90,60,30,40,50]})

fig.style.use('ggplot')
sns.scatterplot(data=df,x='Vendas',y='Lucro',marker='+',s=100)
```



Kde-plot (Kernel Density Estimate)

```
1 #biblioteca seaborn
2 import seaborn as sns
3 import numpy as np
4
5 x=np.random.randn(100)
6 y=np.random.randn(100)+2*x
7
8 sns.kdeplot(x,y)
```

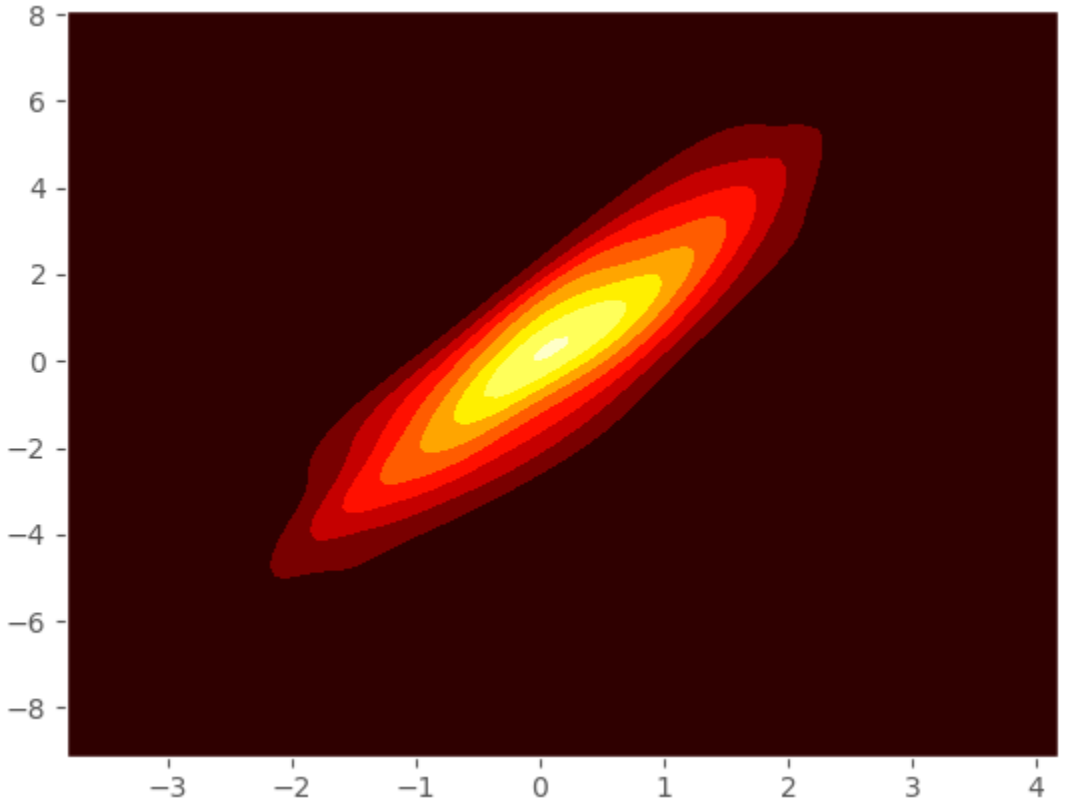


Kde-plot (Kernel Density Estimate, mapa de cores cmap)

```
1 #biblioteca seaborn
2 import seaborn as sns
3 import numpy as np
4
5 x=np.random.randn(100)
6 y=np.random.randn(100)+2*x
7
8 sns.kdeplot(x,y,cmap='hot',shade=True)
```

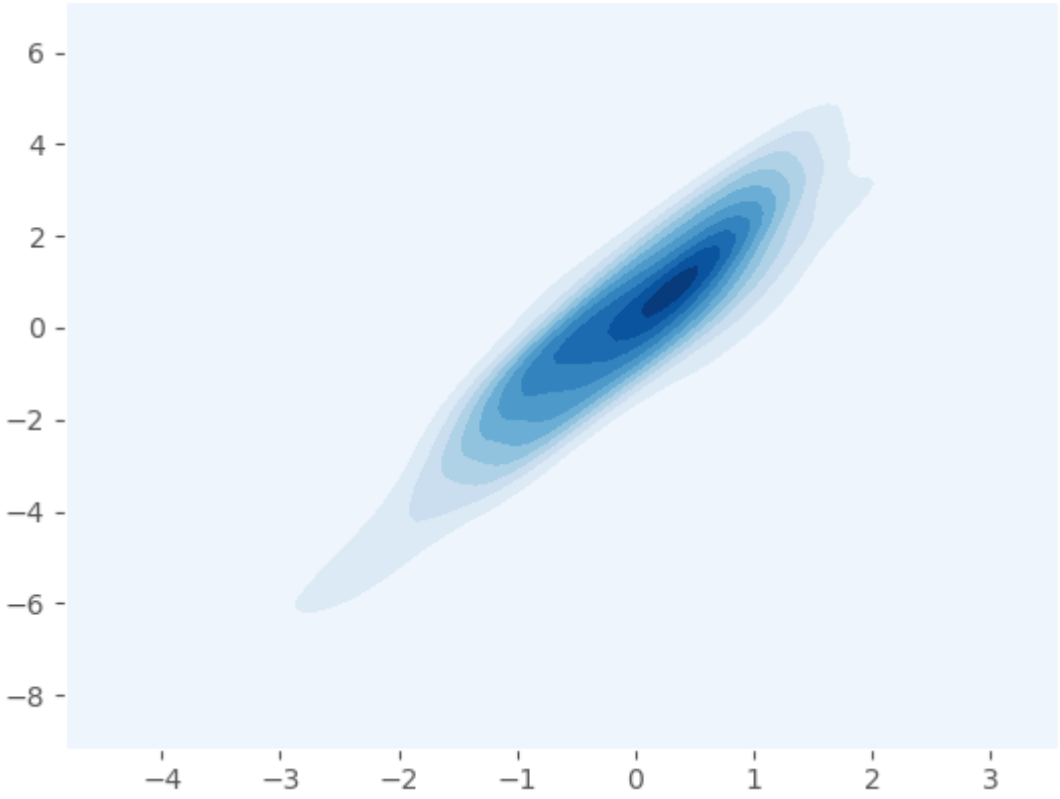
Mapa de cores

Fundo escuro



Kde-plot (Kernel Density Estimate, cores "frias")

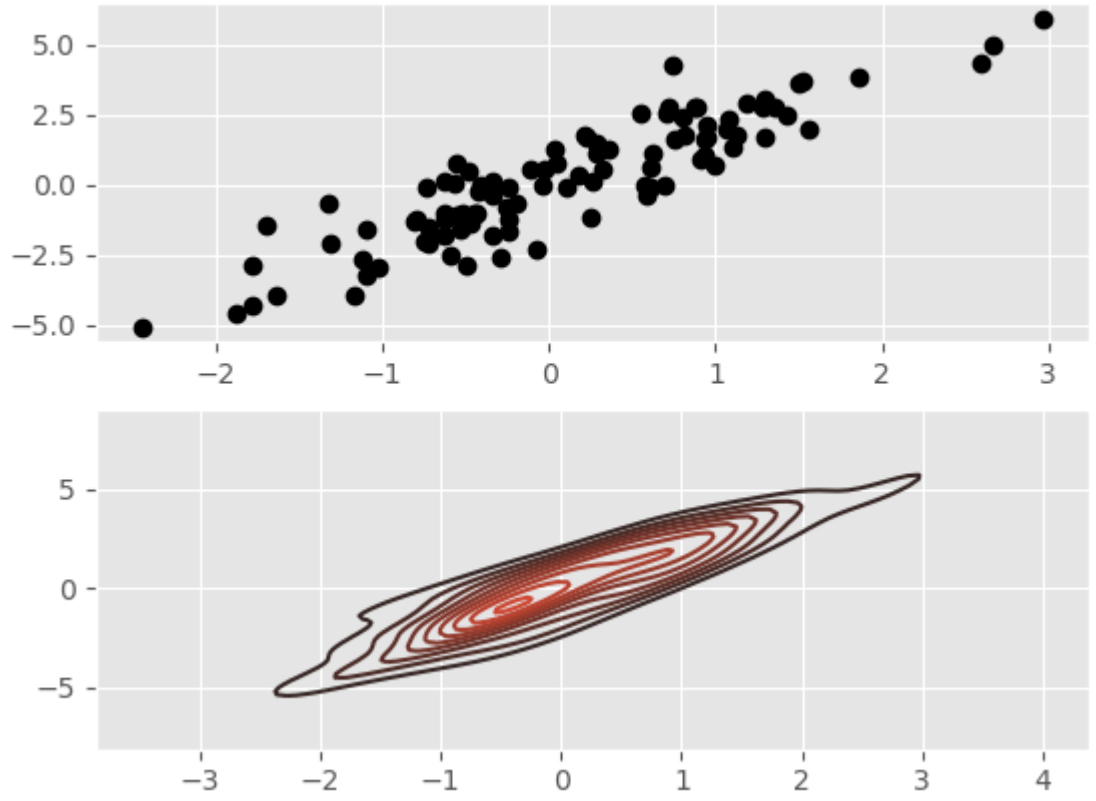
```
1 #biblioteca seaborn
2 import seaborn as sns
3 import numpy as np
4
5 x=np.random.randn(100)
6 y=np.random.randn(100)+2*x
7
8 sns.kdeplot(x,y,cmap='Blues',shade=True)
```



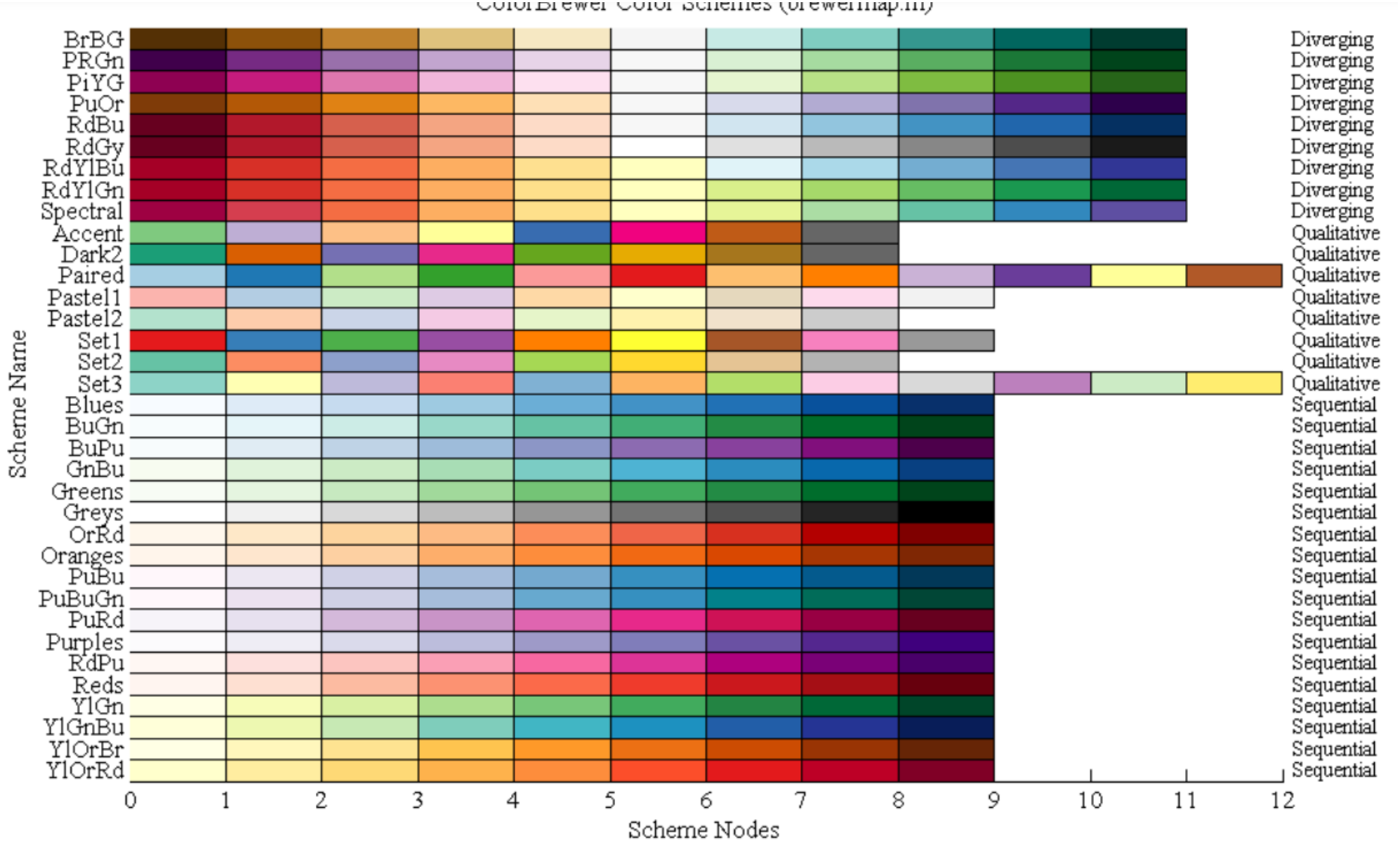
Kde-plot (Kernel Density Estimate com subplot)

```
1  #biblioteca seaborn
2  import matplotlib.pyplot as fig
3  import seaborn as sns
4  import numpy as np
5
6  x=np.random.randn(100)
7  y=np.random.randn(100)+2*x
8
9  fig.subplot(211)
10 fig.plot(x,y,'ok')
11
12 fig.subplot(212)
13 sns.kdeplot(x,y)
```

Precisa da Matplotlib

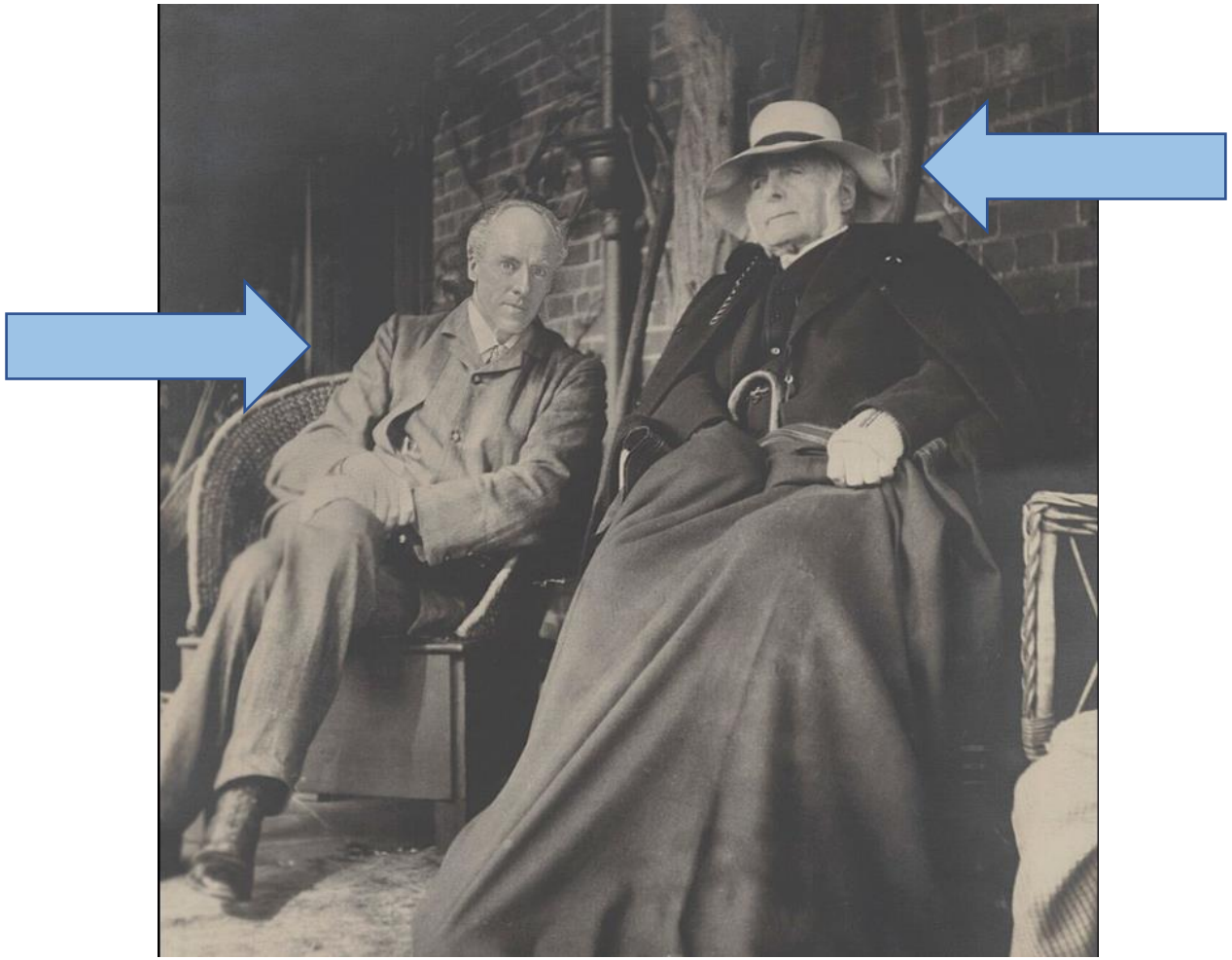


Palettes das cores



Regressão linear

Karl Pearson



Francis Galton

Regressão linear

Francis Galton

Primo de Charles Darwin

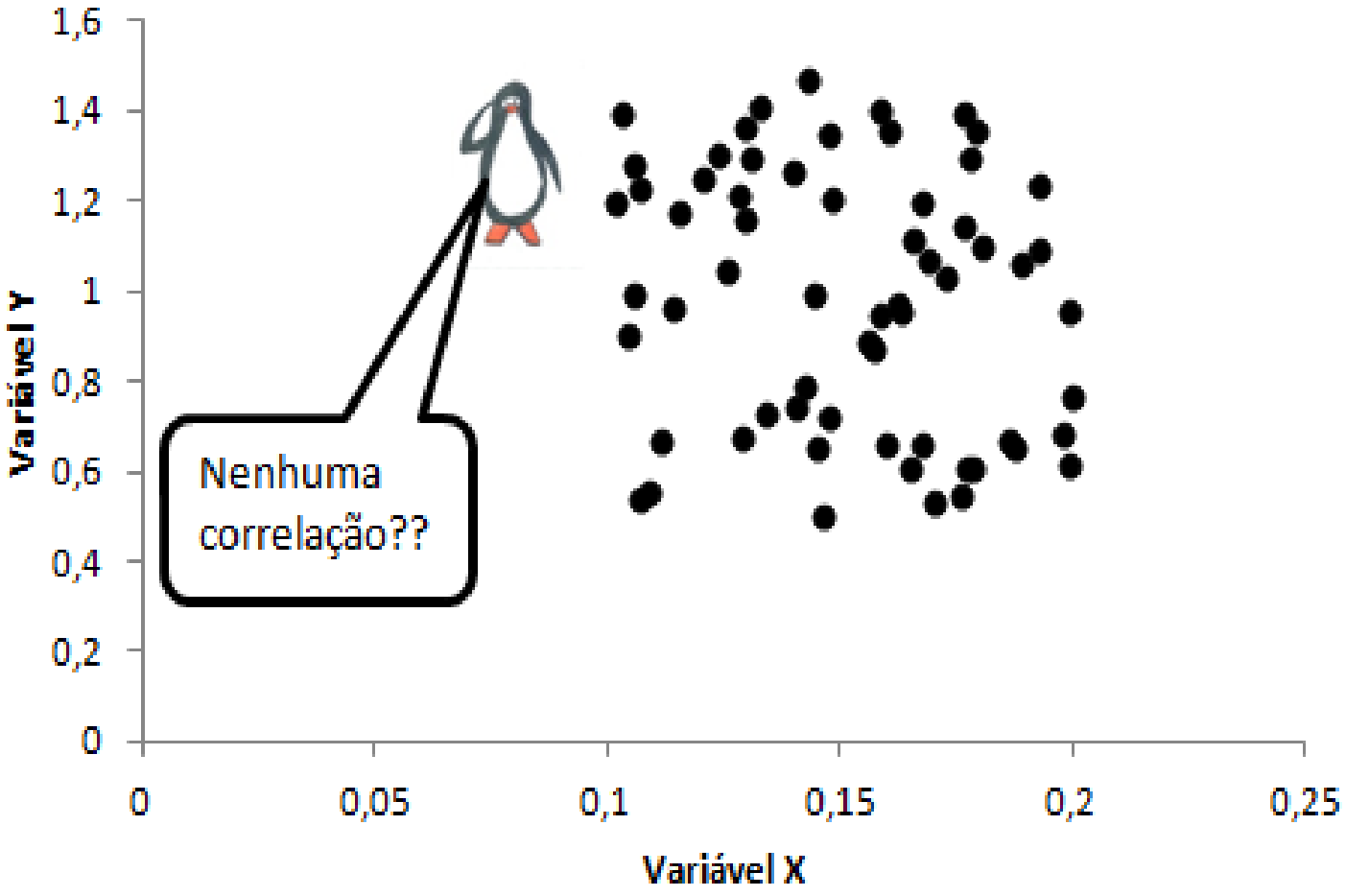
Regressão linear

Francis Galton

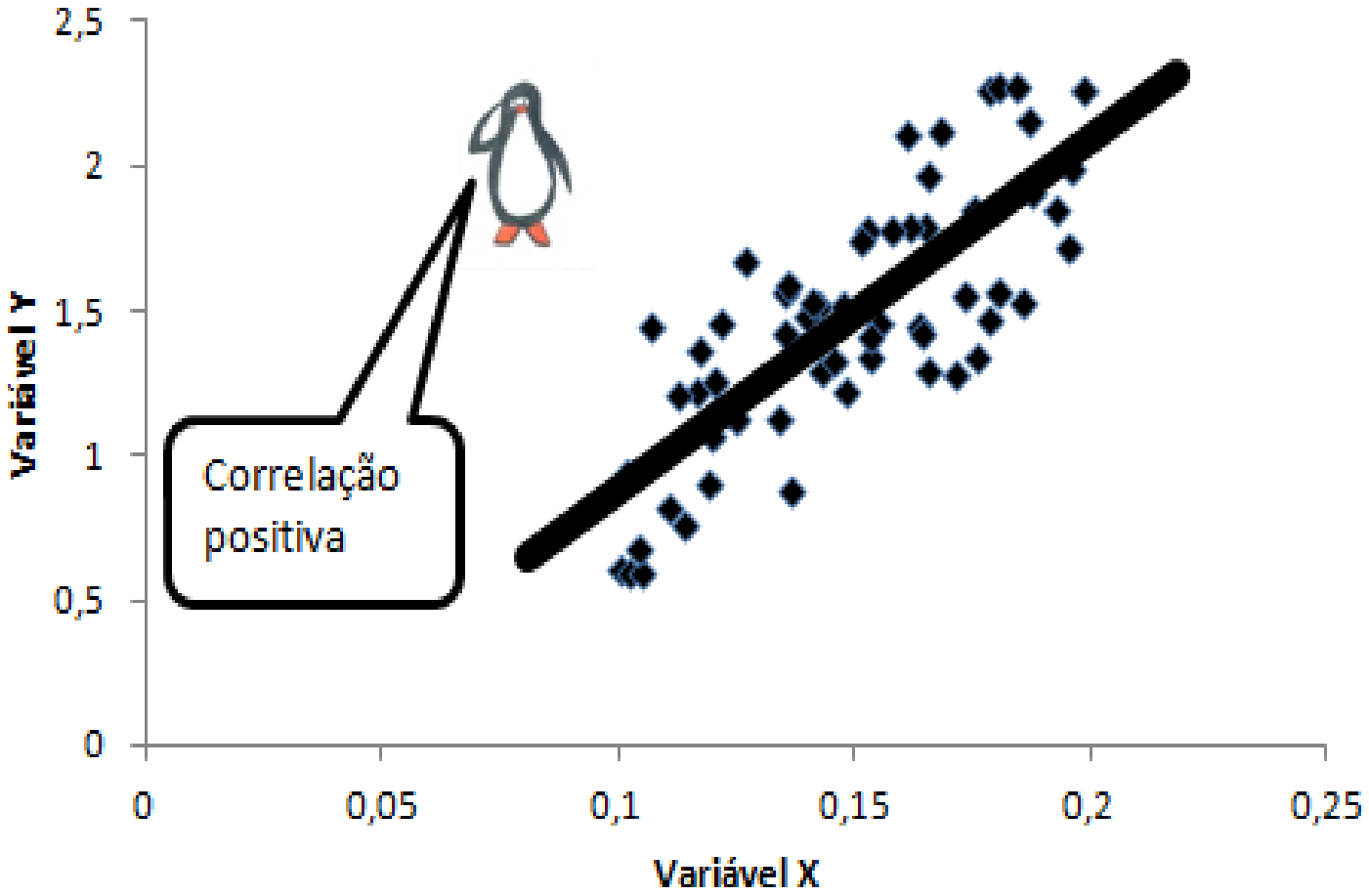
Primo de Charles Darwin

Pioneiro em 1870 em ajustar histogramas com a distribuição de probabilidade normal (gaussiana)

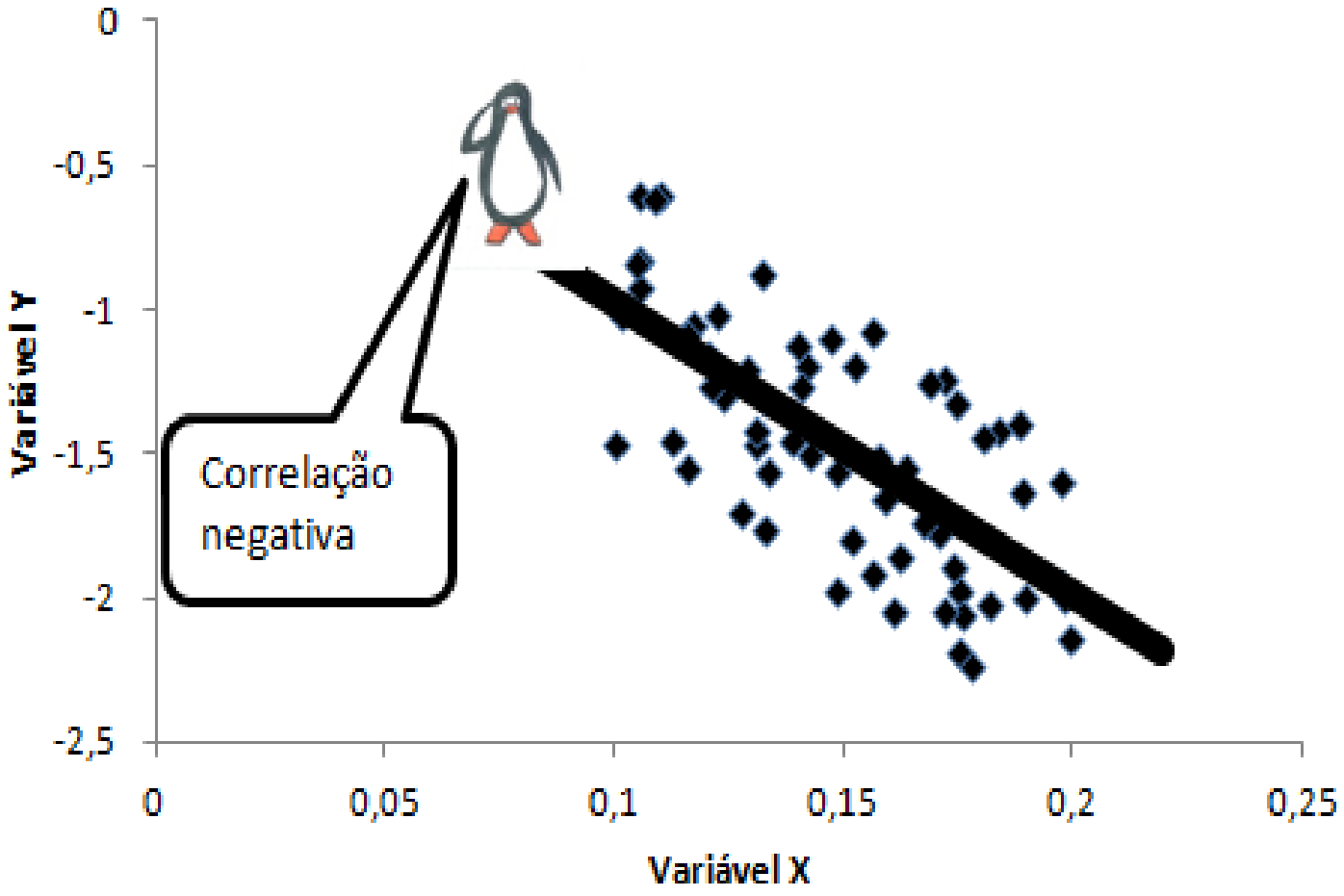
Correlação linear nula



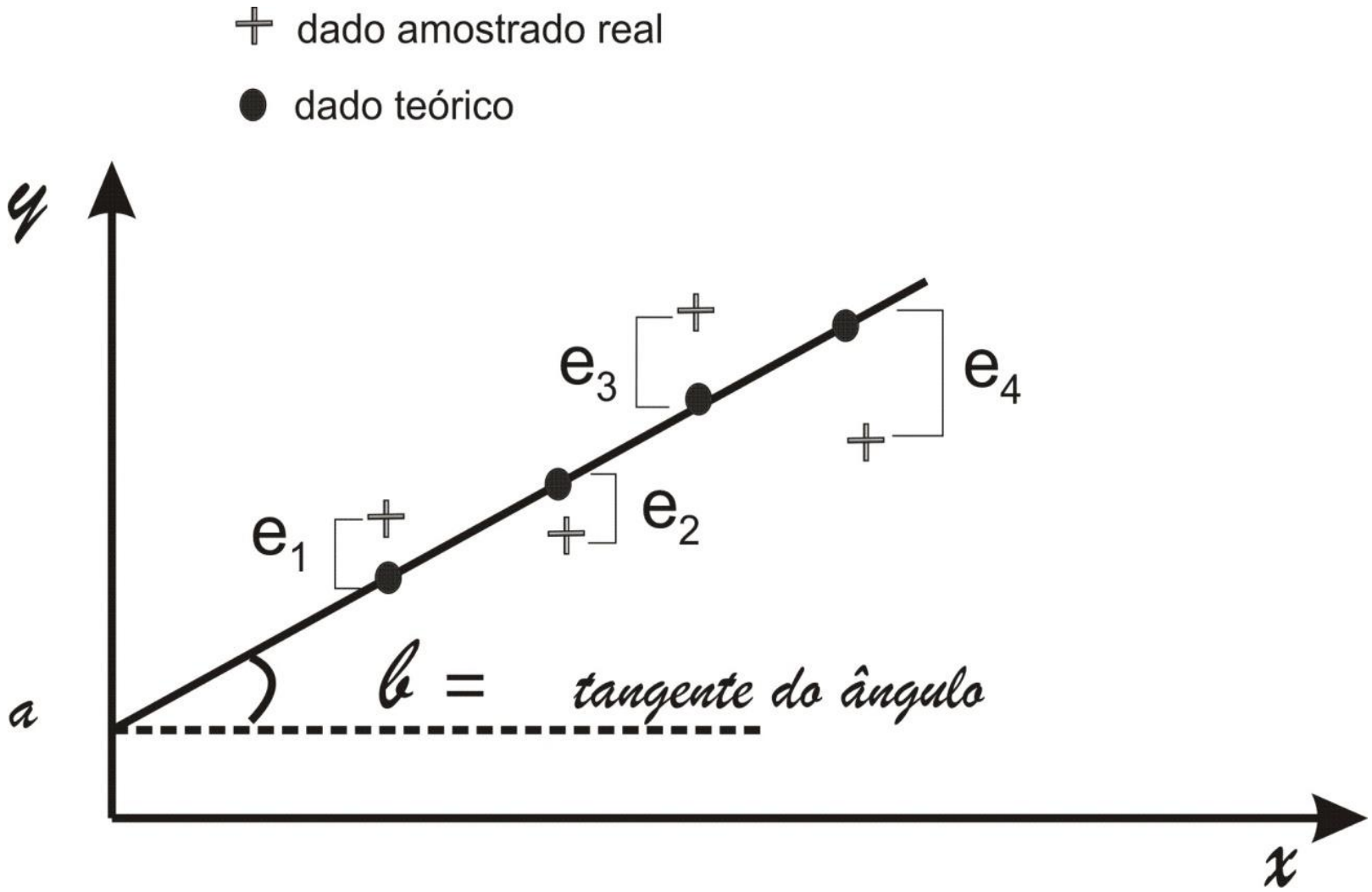
Correlação linear positiva



Correlação linear negativa



Regressão linear



Regressão linear

$$y = \hat{a} + \hat{b}x$$

Regressão linear

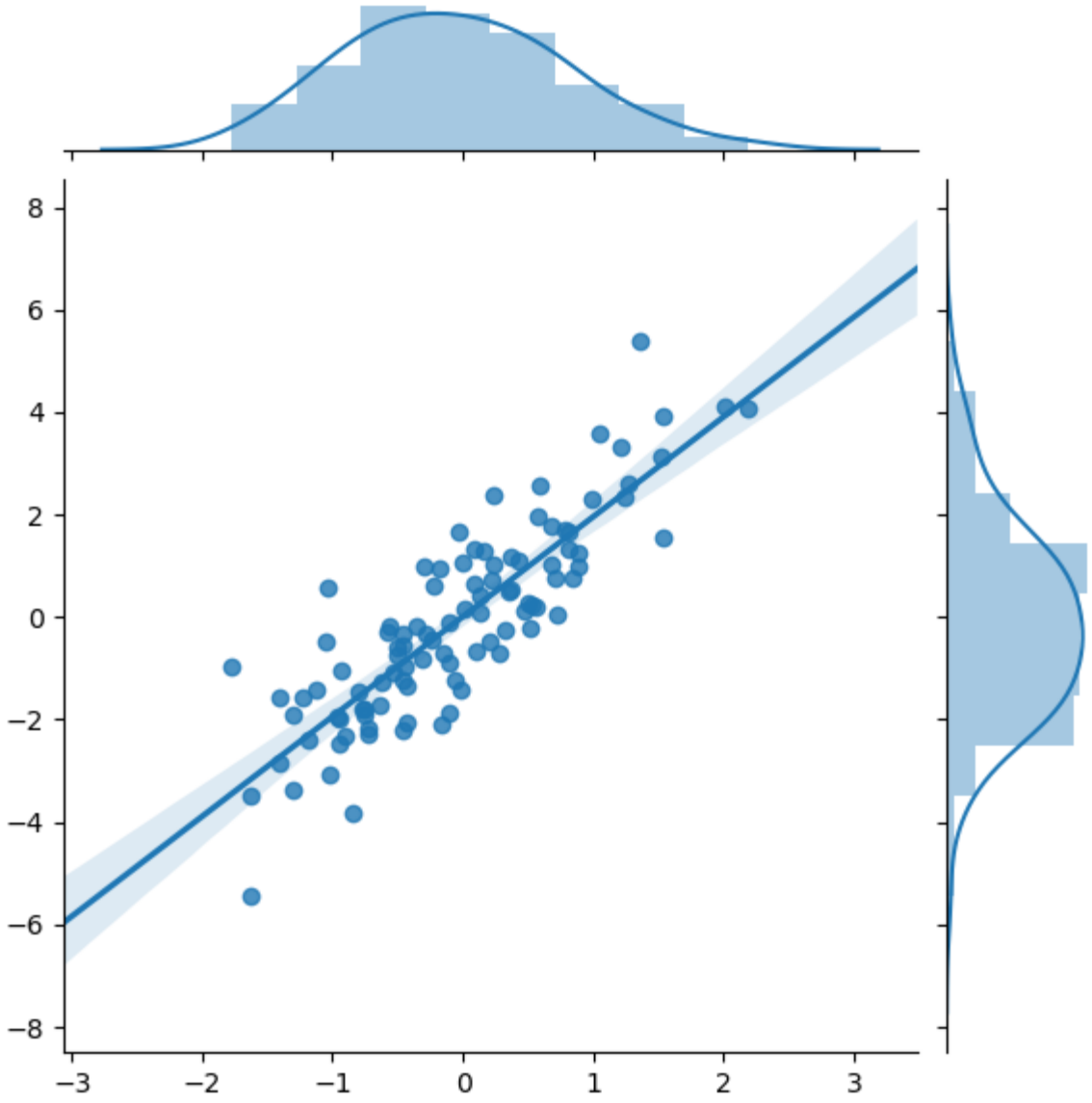
$$y = \hat{a} + \hat{b}x$$

$$\hat{a} = \frac{\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} x_i^2\right) - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} x_i y_i\right)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

$$\hat{b} = \frac{n\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

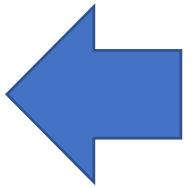
Jointplot - Regressão linear

```
1 #biblioteca seaborn
2 import seaborn as sns
3 import numpy as np
4
5 x=np.random.randn(100)
6 y=np.random.randn(100)+2*x
7
8 sns.jointplot(x,y,kind='reg')
```



Jointplot - Regressão linear – correlação linear

```
1 #biblioteca seaborn
2 import seaborn as sns
3 import numpy as np
4 import pandas as pd
5
6 x=np.random.randn(100)
7 y=np.random.randn(100)+2*x
8
9 ########### Transformando vetor em DataFrame ################
10 df=pd.DataFrame({'c1':x,'c2':y})
11 ##############################################################
```

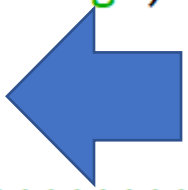


** PASSO IMPORTANTE**

Passa de vetor para DataFrame

Jointplot - Regressão linear – correlação linear

```
1 #biblioteca seaborn
2 import seaborn as sns
3 import numpy as np
4 import pandas as pd
5
6 x=np.random.randn(100)
7 y=np.random.randn(100)+2*x
8
9 ########### Transformando vetor em DataFrame #####
10 df=pd.DataFrame({'c1':x,'c2':y})
11 ##########################################
12 sns.jointplot(x,y,kind='reg')
13
14 correlacao=df.corr()
15
16 print('++++++++++++++++++++++++++++++++++++ ')
17 print('              correlação')
18 print(correlacao)
19 print('++++++++++++++++++++++++++++++++++++')
```

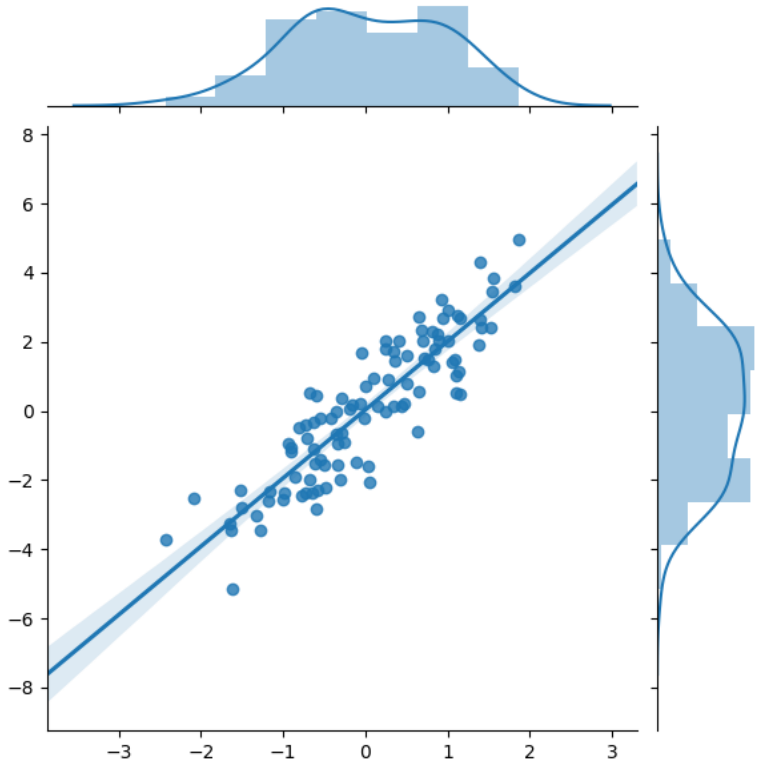


** PASSO IMPORTANTE**

Calcula a correlação linear entre x e y

Jointplot - Regressão linear – correlação linear

```python
#biblioteca seaborn
import seaborn as sns
import numpy as np
import pandas as pd

x=np.random.randn(100)
y=np.random.randn(100)+2*x

########### Transformando vetor em DataFrame #####
df=pd.DataFrame({'c1':x,'c2':y})
##################################################
sns.jointplot(x,y,kind='reg')

correlacao=df.corr()

print('++++++++++++++++++++++++++++++++++++')
print('            correlação')
print(correlacao)
print('++++++++++++++++++++++++++++++++++++')
```




```
++++++++++++++++++++++++++++
            correlação
          c1        c2
c1  1.000000  0.892079
c2  0.892079  1.000000
++++++++++++++++++++++++++++
```

89% de correlação entre x e y