

## Visualizando um Histograma

Nesta aula, entendemos melhor as propriedades de um histograma, inclusive o que está programado no código abaixo:

```
fig = plt.figure(figsize=(5,4))
eixo = fig.add_axes([0,0,1,1])

df_is = df[df['espécie'] == 'Iris-setosa']

mu, sigma = df_is['comprimento_pétala'].mean(), df_is['comprimento_pétala'].std()

eixo.hist(df_is['comprimento_pétala'], bins = 10)
eixo.grid(True)
eixo.set_title('Histograma Iris Setosa', fontsize = 15, pad = 10)
eixo.set_xlabel('Comprimento da pétala', fontsize = 15)

eixo.annotate('$\mu = {0:.2f}$\n$\sigma={1:.2f}$'.format(mu, sigma),
              xy=(1.7, 7), fontsize=20)

eixo.annotate('média', xy=(df_is['comprimento_pétala'].mean()-0.2, 13.5), fontsize=20)
eixo.axvline(df_is['comprimento_pétala'].mean(), color='k', linestyle='--')

eixo.annotate('mediana', xy=(df_is['comprimento_pétala'].median()+0.05, 12), fontsize=20)
eixo.axvline(df_is['comprimento_pétala'].median(), color='g', linestyle='--')

fig.savefig('histograma_is.png', bbox_inches='tight')
```

Sobre o código acima, seguem algumas afirmações:

- **I:** Por meio do método `axvline()`, podemos adicionar tanto linhas verticais quanto horizontais;
- **II:** A linha `df_is = df[df['espécie'] == 'Iris-setosa']` faz um filtro no conjunto de dados, selecionando apenas os registros cujas espécies são Íris Setosa;
- **III:** O argumento `bins` — do método `.hist()` — se refere à quantidade de registros que serão utilizados para montar o histograma;
- **IV:** O método `.savefig()` — do objeto `fig` — salva toda a figura, mesmo se ela possuir mais de um eixo.

Considerando estas afirmações, podemos dizer que:

Selecione uma alternativa

**A** Todas afirmações estão corretas.

**B** Apenas a afirmação I está correta.

**C**

Apenas as afirmações IV e II estão corretas.

**D**

Apenas a afirmação II e III estão corretas.

**E**

Apenas a afirmação III está correta.