

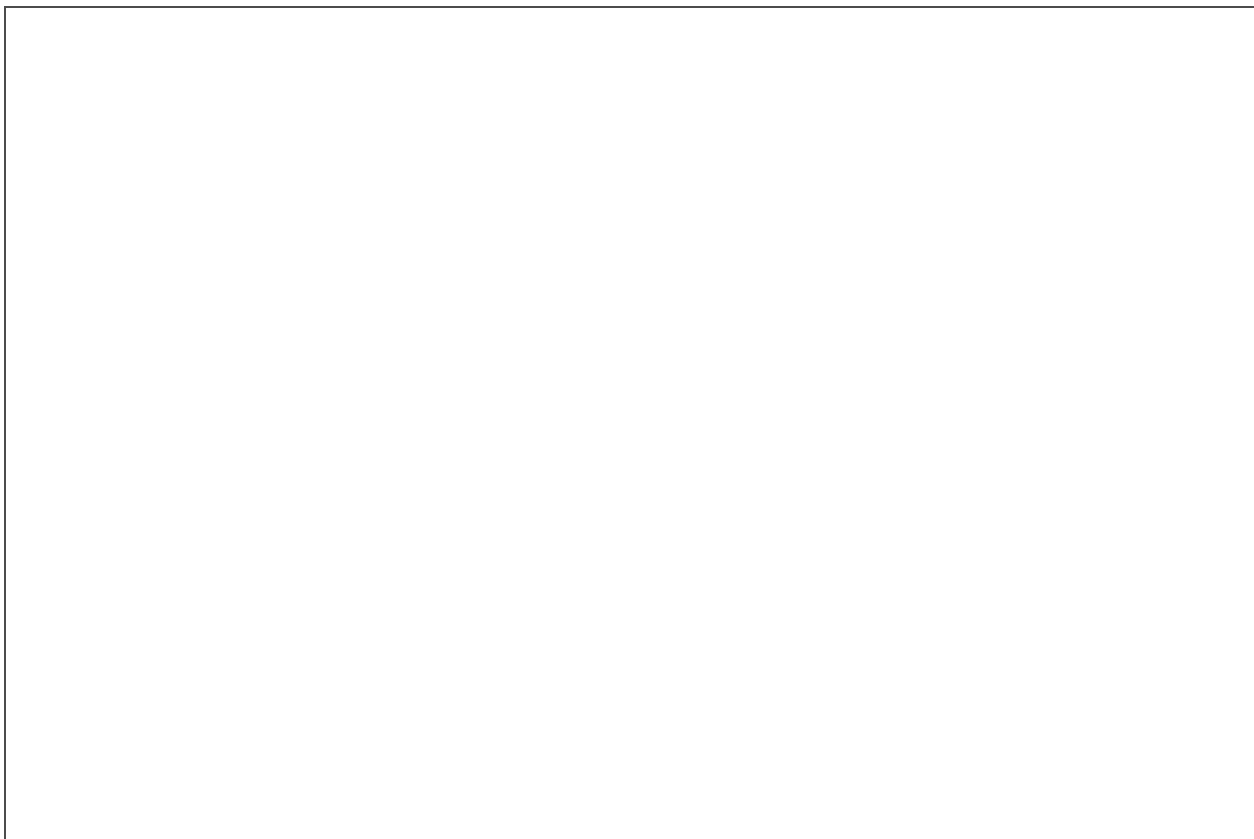
# Gráficos de Barras em Python



Rafael Fabri Chimidt · [Follow](#)

6 min read · May 30, 2020

Passo a passo de como fazer simples e belos gráficos de barras em Python utilizando a biblioteca Seaborn e Matplotlib.



**A**tualmente existem diversos softwares que nos dão a possibilidade de realizar inúmeras visualizações gráficas, como por exemplo o MatLab, OriginPro, Qtiplot, Excel e entre outros. No entanto, é possível também utilizar das bibliotecas do Python para a realização destes gráficos.

Cientistas de dados e programadores utilizam de maneira ampla essa linguagem para criar visualizações impactantes em seus projetos, porém, as vezes, para

peessoas que não têm tanta familiaridade com a programação, ou mesmo para programadores iniciantes, encontrar uma maneira de realizar tal tarefa pode demandar um certo tempo. Portanto, o objetivo deste artigo é mostrar o passo a passo de como fazer um gráfico de barras em Python e também algumas customizações.

## Então vamos parar de enrolação e ir direto ao ponto!!!

Neste artigo realizaremos um gráfico de barras e algumas modificações, como por exemplo alterar cor, excluir linhas dos eixos (axis), inserir textos etc. Logo, se você gosta de gráficos e de Python esse artigo com certeza é para você.

Abaixo foi importada as bibliotecas utilizadas.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Foi criado um banco de dados bem simples, para assim executarmos os gráficos:

- Nos índices estão presentes os anos;
- Na coluna quantidade está presente alguns valores, esses foram escolhidos de maneira crescente a cada ano.

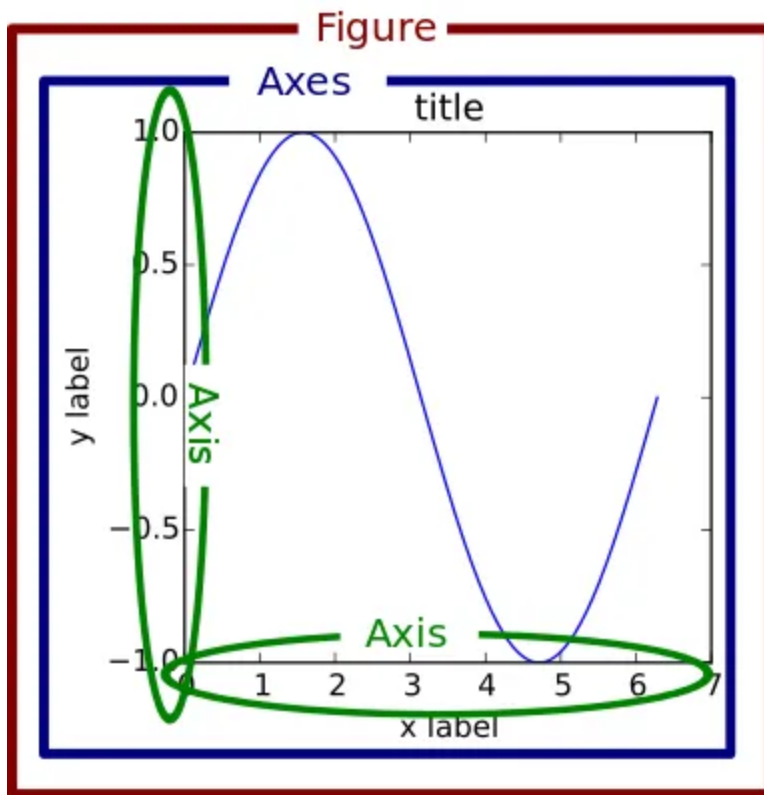
Abaixo encontra-se uma imagem referente ao *dataset*.

quantidade	
2000	1
2001	3
2002	5
2003	7
2004	9
2005	11
2006	13
2007	15
2008	17
2009	19

## Agora vamos plotar nosso gráfico!!!

- Para a realização do gráfico é necessário criarmos uma Figure e um Axes utilizando a função `plt.subplots()`.

Abaixo encontra-se uma imagem para um melhor entendimento da plotagem de um gráfico.

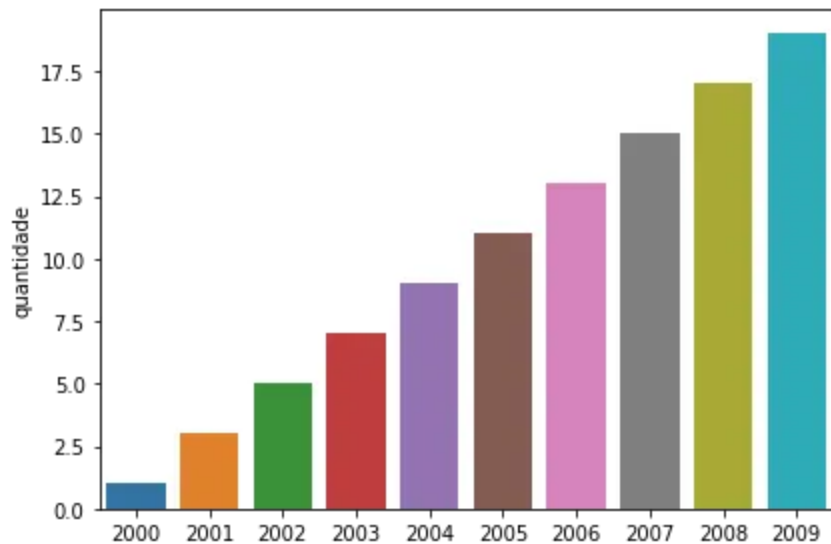


- Utilizaremos o `sns.barplot` para realizar o gráfico de barras.

```
#criando uma figure, axes
fig, ax = plt.subplots()

#criando o gráfico de barras
sns.barplot(x=df.index, y=df['quantidade'], ax=ax, data=df)

#otimizar espaço da figure
fig.tight_layout();
```



1º gráfico de barras

UAU, viu como que com simples 3 linhas foi possível plotar um gráfico de barras?!

No entanto, esse gráfico não está muito bonito ou agradável, não é mesmo? Então vamos fazer umas modificações.

Primeiramente mudaremos a cor das barras, para isso existe um argumento chamado *palette*, encontra-se dentro da função `barplot`, que tem o objetivo de mudar a cor das barras. Neste [link](#) existem várias cores de *palette*, a que utilizei foi 'RdPu', abaixo está a paleta de cores adicionada.



Palette 'RdPu'

- Adicionaremos um título ao gráfico, usando a função `ax.set_title()`;
- E também será utilizado o argumento *figsize*, na função `subplots`, para deixarmos maior o tamanho da figure;

- Além disso, aumentaremos o tamanho dos labels de cada ticks, função `ax.ticks_params()`;
- Algumas mudanças serão realizadas no tamanho dos *labels* do eixo y e eixo x através da função `ax.set_ylabel()` e `ax.set_xlabel()`.

```
#criando uma figure, axes, alterando tamanho
fig, ax = plt.subplots(figsize=(8,6))

#criando o gráfico de barras
sns.barplot(x=df.index, y=df['quantidade'], ax=ax, data=df,
            palette='RdPu')

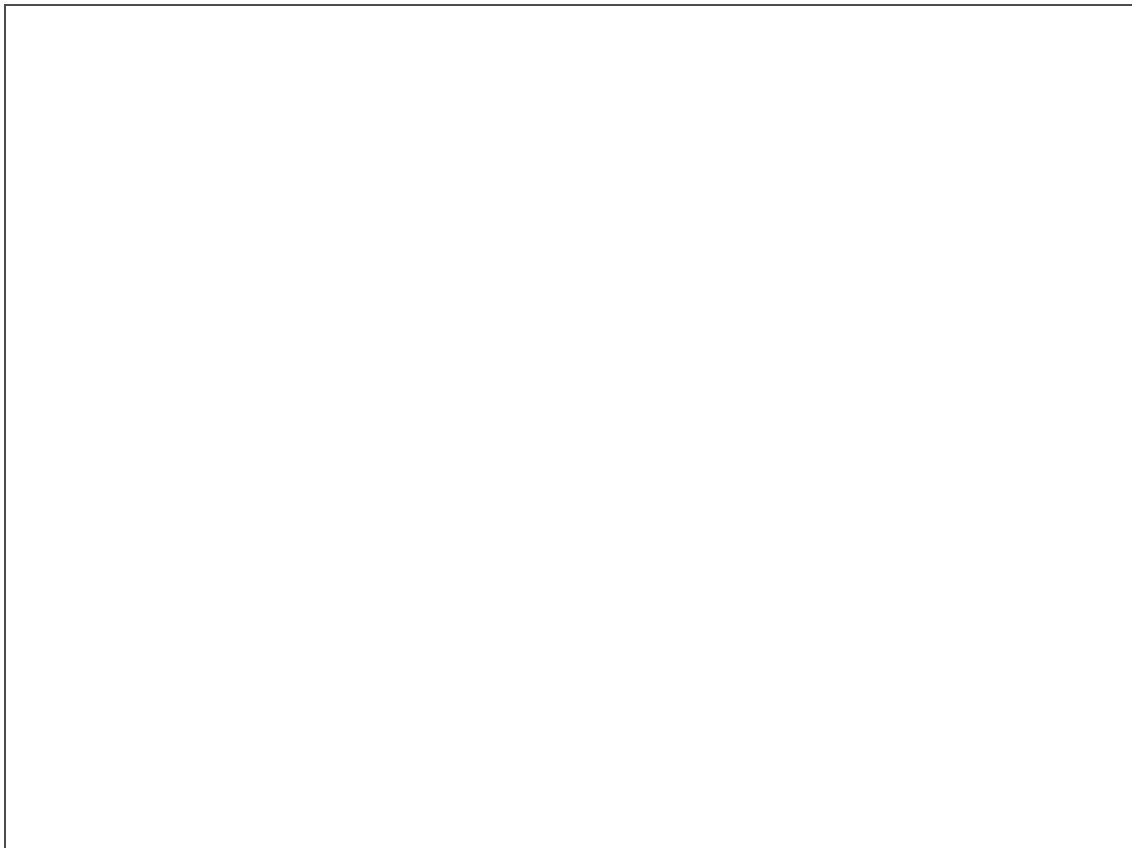
#adicionando título
ax.set_title("quantidade x ano", fontdict={'fontsize':15})

#mudando nome e tamanho do label x
ax.set_xlabel('Anos', fontdict={'fontsize':14})

#mudando nome e tamanho do label y
ax.set_ylabel('Quantidades', fontdict={'fontsize':14})

#mudando tamanho dos labels dos ticks
ax.tick_params(labelsize=14)

#otimizar espaço da figure
fig.tight_layout();
```



2º Gráfico de barras

Nossa, já melhorou bastante! Mas ainda podemos deixar melhor!!!

Agora realizaremos outras mudanças, como:

- Remover o label esquerdo;
- Remover axis direito, superior, e esquerda;
- Deixar a linha do axis inferior mais grossa.

Os códigos novos estarão em **negrito**.

```
#criando uma figure, axes, alterando tamanho  
fig, ax = plt.subplots(figsize=(8,6))
```

```
#criando o gráfico de barras
sns.barplot(x=df.index, y=df['quantidade'], ax=ax, data=df,
palette='RdPu')

#adicionando título
ax.set_title("quantidade x ano", fontdict={'fontsize':15})

#mudando o nome e tamanho do label x
ax.set_xlabel('Anos', fontdict={'fontsize':14})

#mudando tamanho do label eixo y
ax.set_ylabel('')

#mudando tamanho dos labels dos ticks
ax.tick_params(labelsize=14)

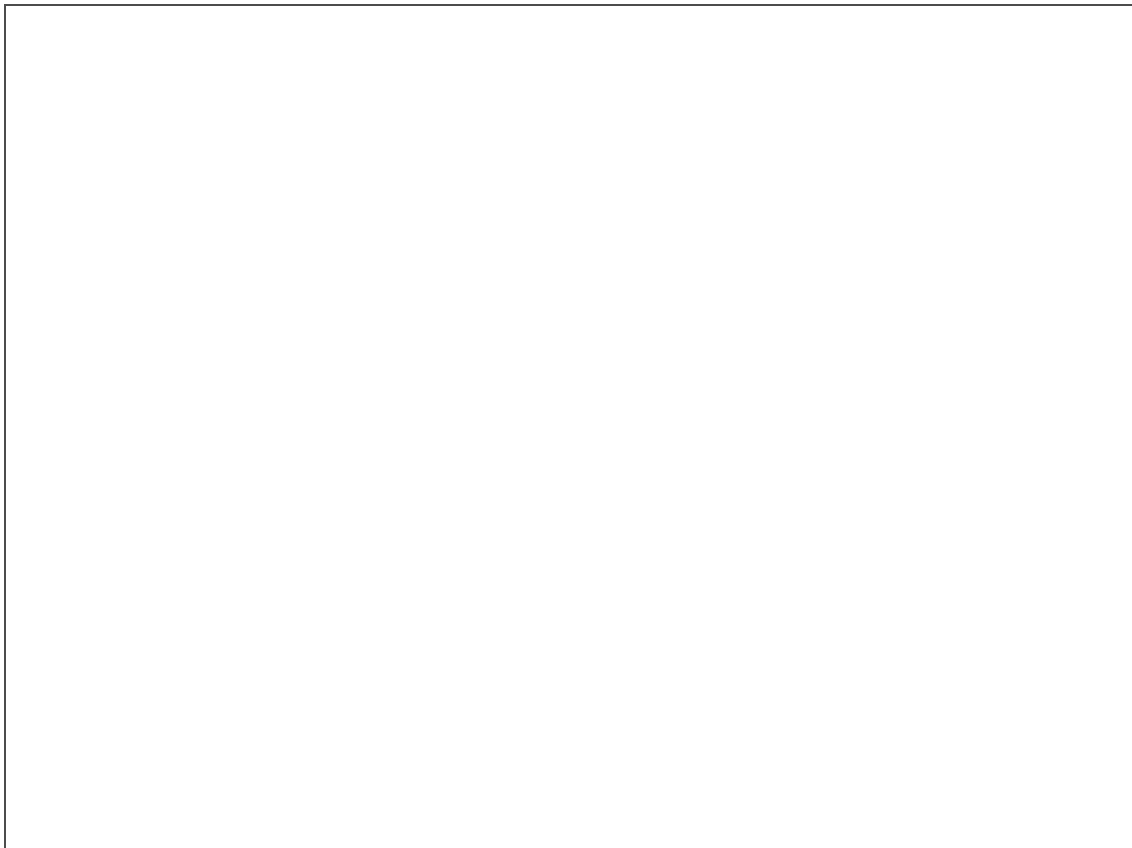
#aumentando espessura linha inferior
ax.spines['bottom'].set_linewidth(2.5)

#remoção dos outros três axis
for axis in ['top', 'right', 'left']:
    ax.spines[axis].set_color(None)

#remoção dos ticks
ax.tick_params(axis='y', labelleft=False, left=None)

#otimizar espaço da figure
fig.tight_layout();
```





3º Gráfico de barras

Agora que removemos o eixo y, vamos colocar o valor de cada quantidade em cima de sua respectiva barra.

Exemplificando, nota-se que para o ano 2000 temos uma quantidade 1 und, para 2001 temos 3 und, assim 1 ficará acima da primeira barra e 3 ficará em cima da segunda barra e em cada próxima barra iremos fazer a mesma coisa, isso é realizado para facilitarmos a leitura do gráfico.

quantidade	
2000	1
2001	3

Para a execução desta tarefa, utilizaremos a função `ax.text()`, que tem a finalidade de atribuir textos em qualquer lugar do gráfico, nesta função existe 4 parâmetros importantes:

- `(x, y)`: nesses argumentos são disponibilizados as localizações (`x`:horizontal e `y`:vertical) que devem ser colocadas no texto.
- `x`: são as barras, e como são 10 barras, essas localizações irão variar de (0 até 9).
- `y`: são as alturas de cada barra, logo é o valor do eixo y, que para primeira barra é 1 und, para segunda barra 3 und etc.
- `s`: esse argumento é a respectiva quantidade em cada barra (1,3,5,7...), este é o texto que irá representar a quantidade da barra, o valor `s` é o mesmo da variável altura (`y`). Deve-se ter atenção, pois o argumento `s` deve ser uma variável do tipo string.
- `fontsize`: define o tamanho do texto.

Uma imagem pode valer mais de mil palavras!!! kkk

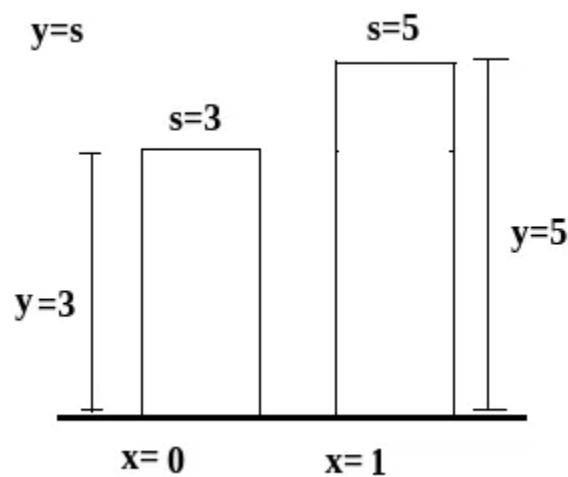


Imagem representativa

**`ax.text(x, y, s)`**

Portanto, abaixo foi criado as duas variáveis necessárias para atribuição dos valores no gráfico, primeiro foi criado a variável chamada *posição*, que será utilizada para cada posição das barras no eixo x.

Após foi criado a variável chamada altura, essa diz respeito a altura de cada barra no eixo y, e também será utilizada para representar o texto da barra.

- x=posicao
- y=altura
- s=altura

```
altura = []
for i in df['quantidade']:
    altura.append(i)

posicao = []
for i in range(0,10,1):
    posicao.append(i)
```

```
altura
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
```

```
posicao
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Agora com as variáveis prontas executaremos novamente o gráfico.

```
#criando uma figure, axes, alterando tamanho
fig, ax = plt.subplots(figsize=(8,6))

#criando o gráfico de barras
sns.barplot(x=df.index, y=df['quantidade'], ax=ax, data=df,
palette='RdPu')
```

```
#adicionando título
ax.set_title("quantidade x ano", fontdict={'fontsize':15})

#mudando e nome e tamanho do label x
ax.set_xlabel('Anos', fontdict={'fontsize':14})

#mudando tamanho do label eixo y
ax.set_ylabel('')

#mudando tamanho dos labels dos ticks
ax.tick_params(labelsize=14)

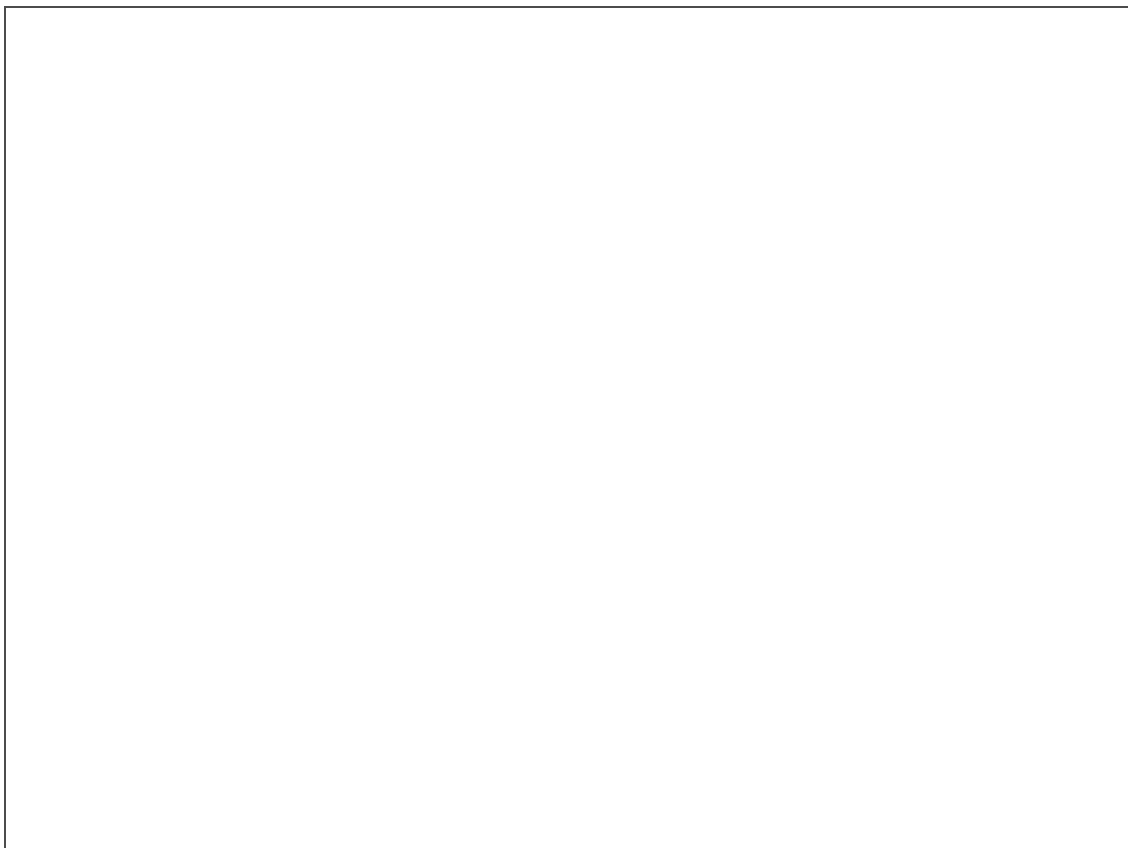
#aumentando espessura linha inferior
ax.spines['bottom'].set_linewidth(2.5)

#remoção dos outros três axis
for axis in ['top', 'right', 'left']:
    ax.spines[axis].set_color(None)

#remoção dos ticks
ax.tick_params(axis='y', labelleft=False, left=None)

#Colocando a quantidade em cada barra
for i in range(10):
    ax.text(x=posicao[i]-0.2, y=altura[i]+0.2, s=str(altura[i]),
        fontSize=15)

#otimizar espaço da figure
fig.tight_layout();
```



4º gráfico de barras

Nossa esse gráfico ficou bem melhor do que o primeiro, conseguimos realizar várias modificações utilizando as bibliotecas matplotlib e seaborn. Existe outra diversas possibilidades de modificações, acredito que nos próximos artigos irei mostrar melhor outras alterações.