



Inteligência Artificial e Big Data

Aula 02

Prof. Me Daniel Vieira

Agenda

- 1- Data Mining
- 2- Aplicações do Data Mining
- 3- Algoritmos de mineração de dados
- 4 - Processo KDD
- 5 -Onde encontrar bases de dados ?
- 6 - Biblioteca Matplotlib
- 7- Tipos de gráficos
- 9 - Exercícios

Mineração de Dados (Data Mining)

Mineração de dados surgiu nos anos de 1990 e se consolidou com o conceito de tecnologia da Informação.

Bruce Greenwald e Joseph Stiglitz.

A informação passou a ser utilizada como um diferencial para extrair insights para melhorar decisões.



Mineração de Dados (Data Mining)



Mineração de Dados (Data Mining)

Mineração de Dados (Data Mining) é o processo de transformar dados em informações úteis por meio da descoberta de padrões, associações e anomalias a partir dos dados coletados. O processo é realizado por algoritmos, muitas vezes de forma automatizada.



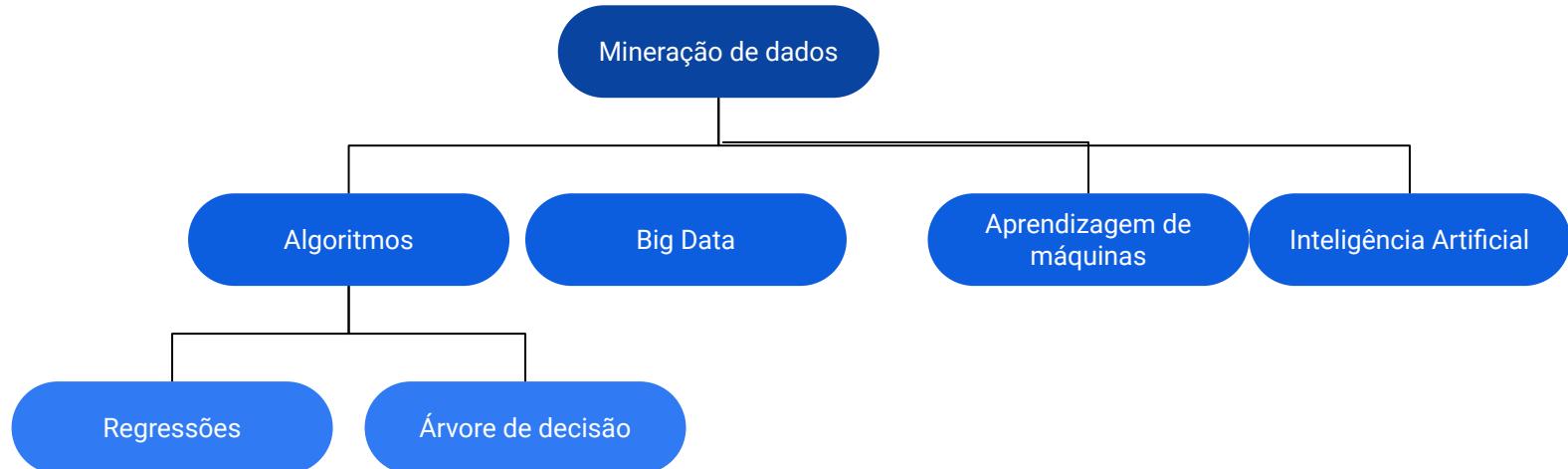
Mineração de Dados (Data Mining)

Análise exploratória

Análise estatística

Mineração de dados

Mineração de Dados (Data Mining)



Aplicações Mineração de Dados (Data Mining)



Exemplo 1



Família



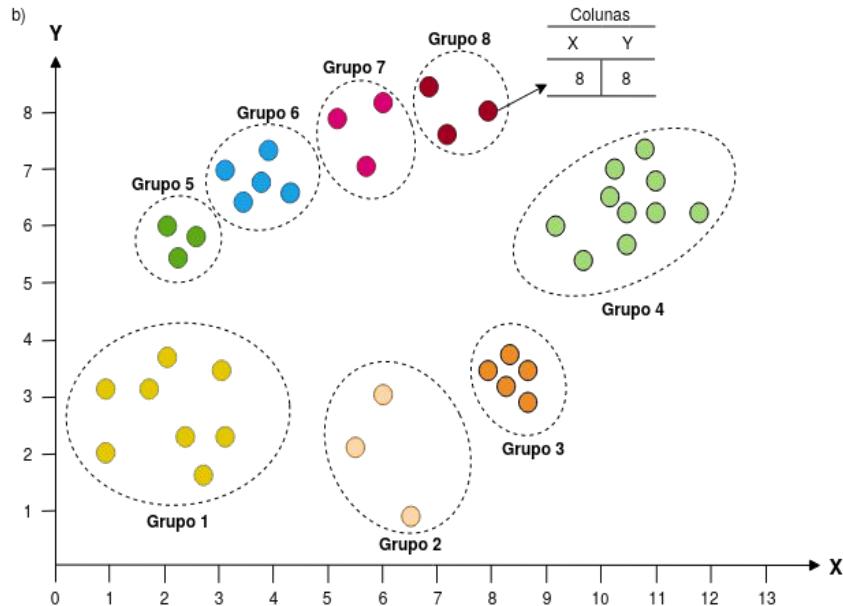
Escola



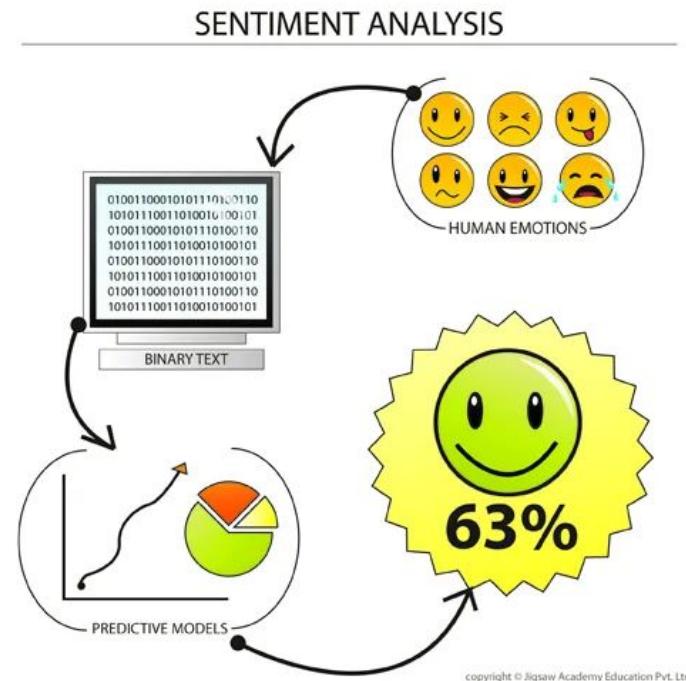
Mulheres



Homens



Aplicações Mineração de Dados (Data Mining)



Aplicações Mineração de Dados (Data Mining)



Adaptado de Rezende, 2003.

Aplicações Mineração de Dados (Data Mining)



BENEFÍCIOS DA **ANÁLISE PREDITIVA**

- 01** **Detecção de fraudes**
detecta e previne comportamentos criminosos ou suspeitos;
- 02** **Otimizar campanhas de marketing**
ajuda a atrair, reter e expandir clientes através da análise de dados;
- 03** **Melhorar operações**
prevê o estoque e gerencia os recursos com maior eficiência;
- 04** **Redução de riscos**
avalia padrões de compra dos consumidores, diminuindo riscos na empresa.

Aplicações Mineração de Dados (Data Mining)

Para atender às necessidades de sua empresa, é fundamental conhecer e diferenciar a Mineração de Dados de outras tecnologias aplicadas na análise de dados como o Business Intelligence (BI) e o Big Data.

O BI mostra o desempenho da empresa por meio de gráficos de visualização (*dashboards*) e indicadores que são gerados com base nos dados brutos de operação, sempre referentes ao passado da empresa. O **BI não possui inteligência aplicada**, por isso, necessita de pessoas para interpretar os dados.

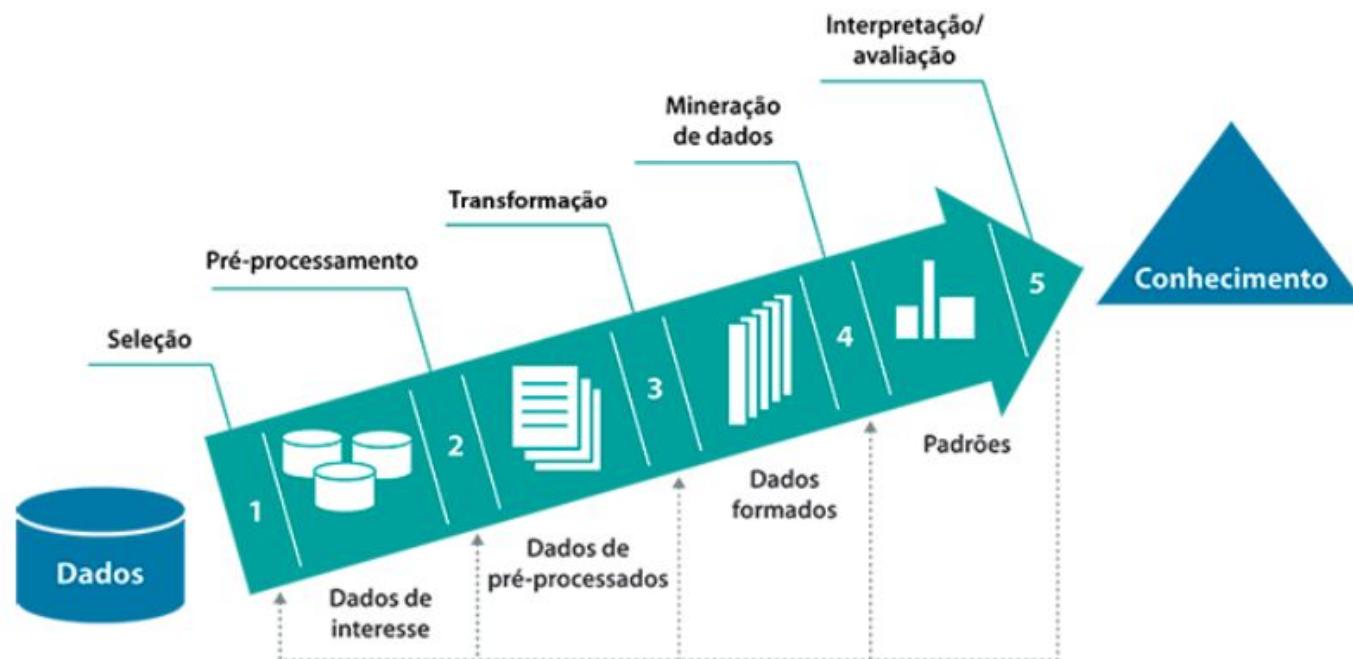
Aplicações Mineração de Dados (Data Mining)



Processo KDD - Knowledge Discovery in Database

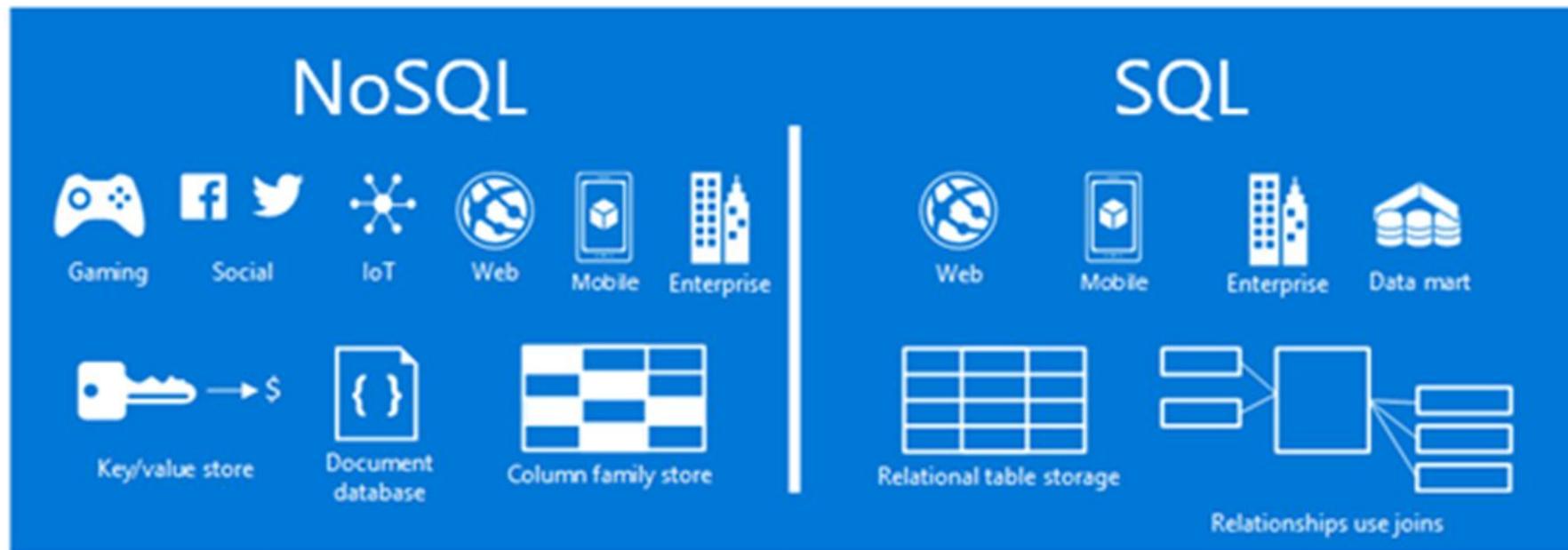


Processo KDD - Knowledge Discovery in Database



Processo KDD - Knowledge Discovery in Database

1. Seleção:



Processo KDD - Knowledge Discovery in Database

2. Pré processamento

Dados NAN

Dados inválidos

Dados não estruturados



Processo KDD - Knowledge Discovery in Database

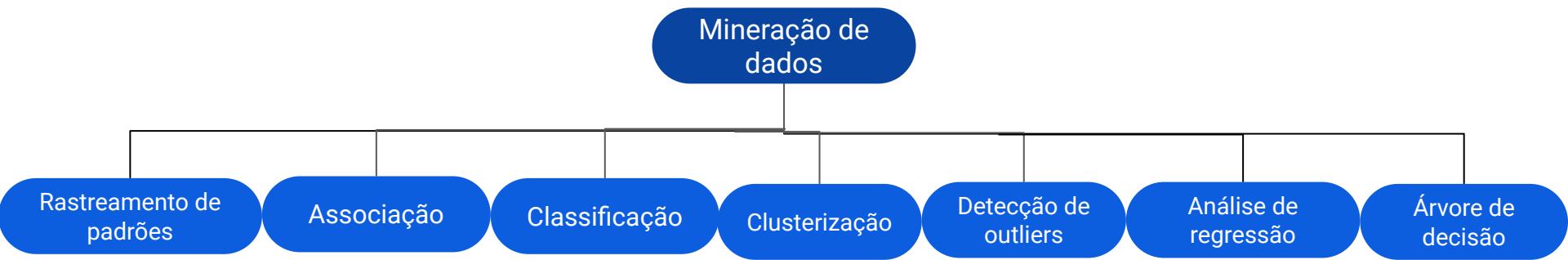
3-Transformação:

Os dados pré-processados são transformados de acordo com a fase posterior, pois alguns algoritmos de aprendizado podem não ser capazes de lidar com atributos em determinados formatos. Outro exemplo comum é a normalização dos dados, que proporciona melhores resultados para alguns algoritmos. Após transformados, os dados são reorganizados e armazenados para que possam ser interpretados na etapa seguinte.

Processo KDD - Knowledge Discovery in Database

Algoritmos utilizados

4. Mineração de Dados:



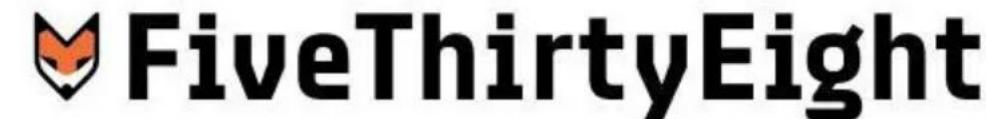
Processo KDD - Knowledge Discovery in Database

5. Interpretação e avaliação:

As informações serão interpretadas a fim de se verificar se são válidas ao problema proposto.

Serão avaliados os padrões descobertos e traduzidos em termos inteligíveis para os usuários, gerando conhecimento ao domínio. Também podem ser descobertos padrões irrelevantes ou redundantes que podem ser desconsiderados e removidos. O conhecimento descoberto deverá ser incorporado ao desempenho e partes interessadas do sistema gerando conhecimento ao domínio. Também podem ser descobertos padrões irrelevantes ou redundantes que podem ser desconsiderados e removidos. O conhecimento descoberto deverá ser incorporado ao desempenho e partes interessadas do sistema.

Onde encontrar bases de dados ?



Biblioteca Pandas - Como realizar a leitura de um arquivo ?

A biblioteca pandas permite a leitura de diversos tipos de arquivo: csv, xls, url e etc

Arquivo excel

```
df = pd.read_excel('/content/contador df.xlsx')
```

Arquivo csv

```
df_evento = pd.read_csv('DF06.10.csv', sep=';')
```

Biblioteca Pandas -

Como realizar a leitura de um arquivo ?

Url

```
url = 'https://raw.githubusercontent.com/pdpcosta/lab2learn/master/DATA/ebmocsna.csv'  
#endereço de onde estão os dados que serão utilizados  
df = pd.read_csv(url)
```

Biblioteca Matplotlib

Matplotlib é uma biblioteca de software para criar gráficos e visualizações de dados em geral, feita para Python e sua extensão matemática numpy

Podemos fazer diversos tipos de gráficos

Gráfico de linhas

Gráfico de Barras

Gráfico 3D



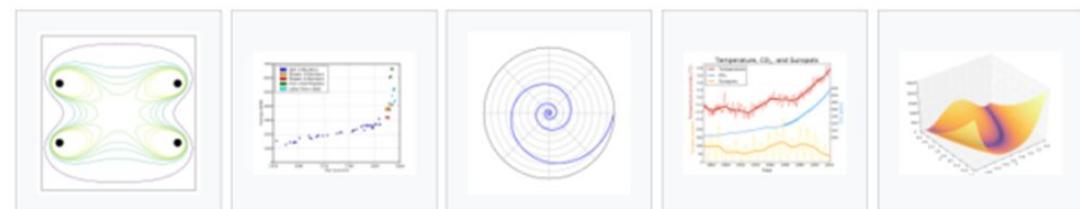
Line plot

Histogram

Scatter plot

3D plot

Image plot



Contour plot

Scatter plot

Polar plot

Line plot

3-D plot

Biblioteca Matplotlib

```
# Exemplo biblioteca matplotlib
import pandas as pd # biblioteca para preparar e modelar dados
import numpy as np # biblioteca para recursos matemáticos
import matplotlib.pyplot as plt # biblioteca para recursos gráficos
```

```
# Gerando numeros com numpy
Dados_x = np.linspace(0,10,10)
Dados_y = Dados_x * Dados_x
```

Biblioteca Matplotlib

```
plt.figure(figsize=(10,5),facecolor= 'gray') #figsize ajusta tamanho do grafico
```

```
#Adicionando eixos
```

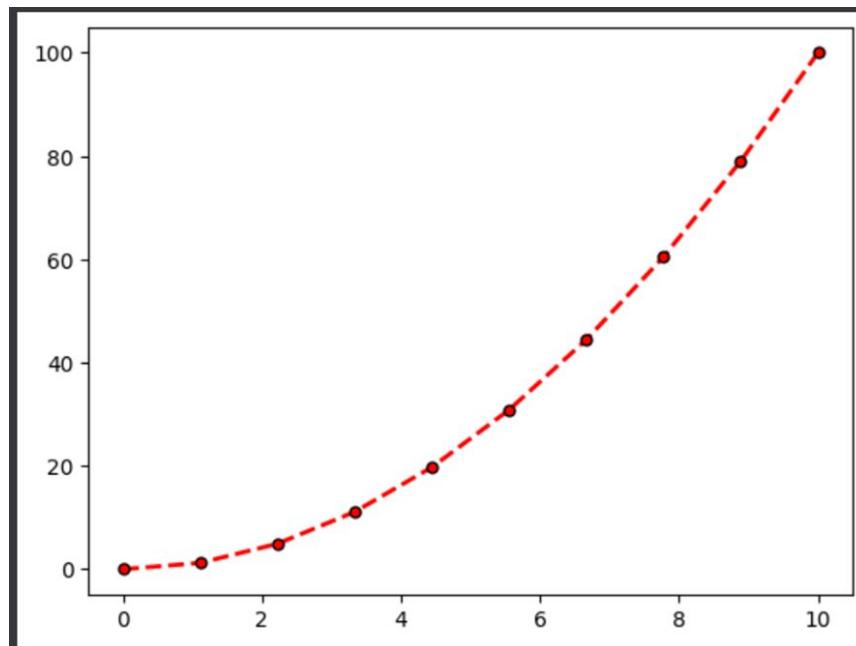
```
Ax = plt.axes()
```

```
Ax.set_facecolor('white')
```

```
#Criando grafico
```

```
plt.plot(Dados_x,Dados_y,# - Dados - do - grafico  
color = 'red', linewidth =2, # - Espessura - da - linha  
markersize=5, #tamanho - do - marcador  
marker = 'o', # - Tipo - de - marcador  
linestyle = 'dashed', # - tipo - da - linha  
markeredgecolor = 'black',  
)
```

Biblioteca Matplotlib



Biblioteca Matplotlib

```
#Definindo nome do label x
plt.xlabel('Eixo x', #nome do label
           fontweight='bold', #estilo
           fontsize='large', #Tamanho da fonte
           fontfamily='fantasy',#tipo da fonte
           color = 'blue',#Cor da fonte
           )
```

Biblioteca Matplotlib

```
#Definindo nome do label y
plt.ylabel('Eixo y', #nome do label
           fontweight='bold', #estilo
           fontsize='large', #Tamanho da fonte
           fontfamily='fantasy',#tipo da fonte
           color = 'blue',#Cor da fonte
           )
```

Biblioteca Matplotlib

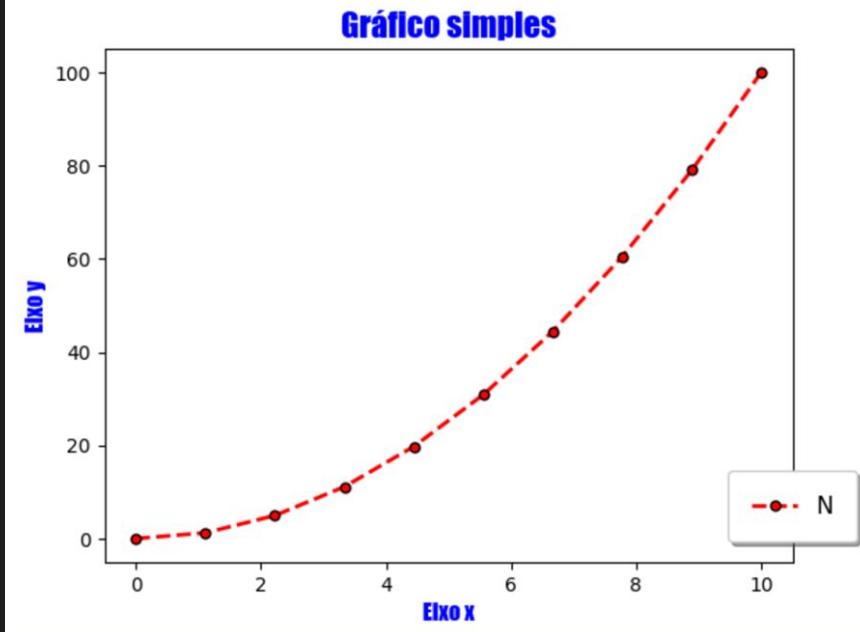
```
#Definindo um titulo
plt.title('Gráfico simples', #titulo do grafico
          fontweight='bold', #estilo
          fontsize='xx-large', #Tamanho da fonte
          fontfamily='fantasy',#tipo da fonte
          color = 'blue',#Cor da fonte
          )
```

Biblioteca Matplotlib

```
#Definindo um titulo
plt.title('Gráfico simples', #titulo do grafico
          fontweight='bold', # estilo
          fontsize='xx-large', #Tamanho da fonte
          fontfamily='fantasy',#tipo da fonte
          color = 'blue',#Cor da fonte
          )
plt.plot(Dados_x,Dados_y,# Dados do grafico
          color = 'red', linewidth =2, # Espessura da linha
          markersize=5, #tamanho do marcador
          marker = 'o', # Tipo de marcador
          linestyle = 'dashed', # tipo da linha
          markeredgecolor = 'black',
          )
#Definindo legendas
plt.legend('Numeros aleatórios', #Nome da legenda
           loc = 'upper center', #Poisção da legenda
           bbox_to_anchor=(1,0,0.0,0.2),
           fontsize='large', #tamanho do texto
           shadow = True, #sombra na legenda
           borderpad=1, #Borda da legenda
           )
```

Biblioteca Matplotlib

```
#Definindo nome do label x  
plt.xlabel('Eixo x', #nome do label  
fontweight='bold', # estilo  
fontsize='large', #Tamanho da fonte  
fontfamily='fantasy',#tipo da fonte  
color = 'blue',#Cor da fonte  
)  
  
#Definindo nome do label y  
plt.ylabel('Eixo y', #nome do label  
fontweight='bold', # estilo  
fontsize='large', #Tamanho da fonte  
fontfamily='fantasy',#tipo da fonte  
color = 'blue',#Cor da fonte  
)
```

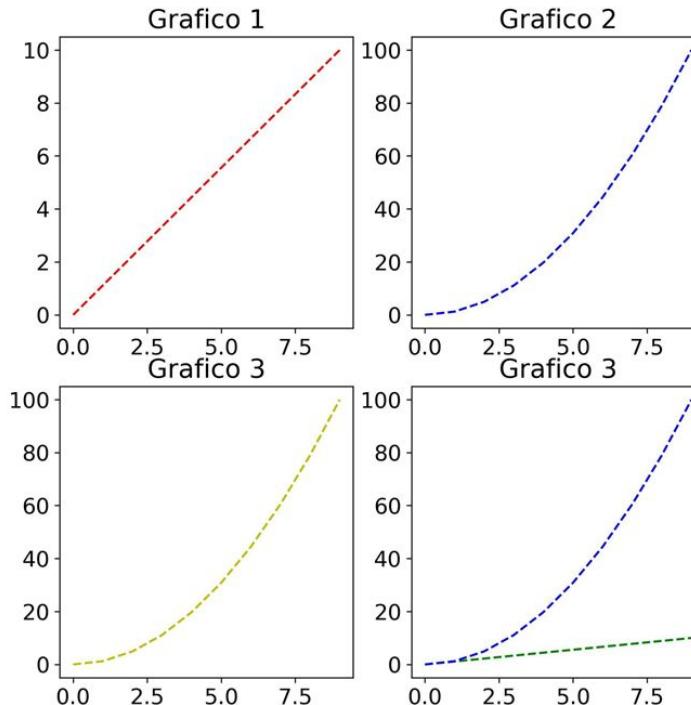


Biblioteca Matplotlib - Subplots

```
plt.rcParams.update({'font.size':15})
plt.figure(figsize= [10,10])
linhas =2
colunas =2
plt.subplot(linhas,colunas,1)
plt.plot(Dados_x,'r--')
plt.title('Grafico 1')
#Grafico 2
plt.subplot(linhas,colunas,2)
plt.plot(Dados_y,'b--')
plt.title('Grafico 2')
#Grafico 3
plt.subplot(linhas,colunas,3)
plt.plot(Dados_y,'y--')
plt.title('Grafico 3')
#Grafico 4
plt.subplot(linhas,colunas,4)
plt.plot(Dados_x,'g--')
plt.plot(Dados_y,'b--')
plt.title('Grafico 3')
```

```
%config InlineBackend.figure_formats=['svg']
plt.rcParams.update({'font.size':15})
plt.figure(figsize= (8,8))
linhas =2
colunas =2
plt.subplot(linhas,colunas,1)
plt.plot(Dados_x,'r--')
plt.title('Grafico 1')
#Grafico 2
plt.subplot(linhas,colunas,2)
plt.plot(Dados_y,'b--')
plt.title('Grafico 2')
#Grafico 3
plt.subplot(linhas,colunas,3)
plt.plot(Dados_y,'y--')
plt.title('Grafico 3')
#Grafico 4
plt.subplot(linhas,colunas,4)
plt.plot(Dados_x,'g--')
plt.plot(Dados_y,'b--')
plt.title('Grafico 3')
plt.savefig('Exemplo subplot .jpg', dpi=300, transparent=True, bbox_inches='tight')
```

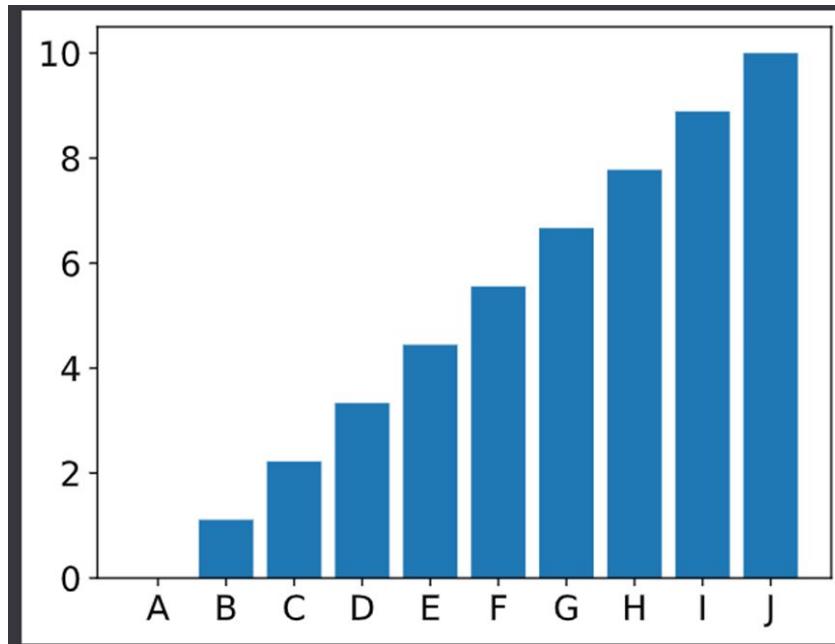
Biblioteca Matplotlib - Subplots



Biblioteca Matplotlib - Outros tipos de gráficos

```
Figura = plt.figure()
Grafico = Figura.add_axes([0,0,0.7,0.7]) #Tamanho do Grafico
Rotulos =[ 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J' ]
#Gerando gráfico
Grafico.bar(Rotulos,Dados_x)
```

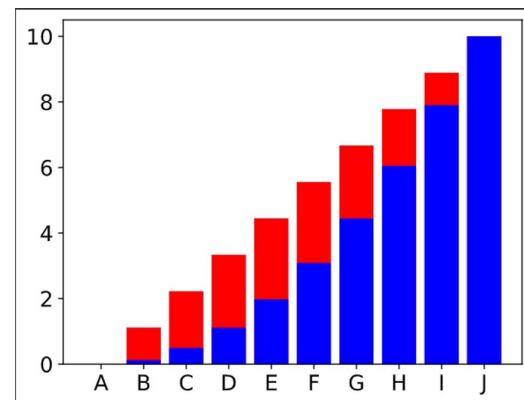
Biblioteca Matplotlib - Outros tipos de gráficos



Biblioteca Matplotlib - Outros tipos de gráficos

```
Grafico.bar(Rotulos,Dados_x, color ='r')  
Grafico.bar(Rotulos,Dados_y*0.1, color ='b')
```

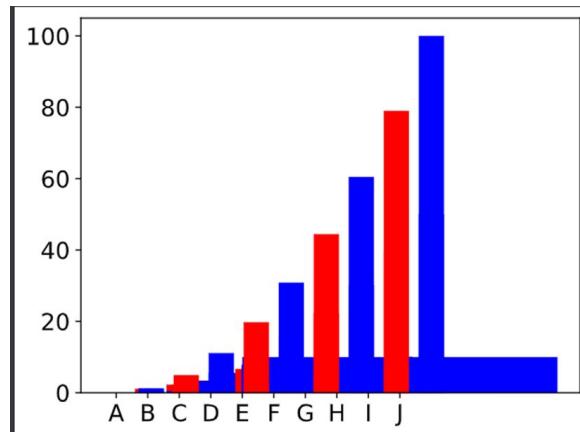
Figura



Biblioteca Matplotlib - Outros tipos de gráficos

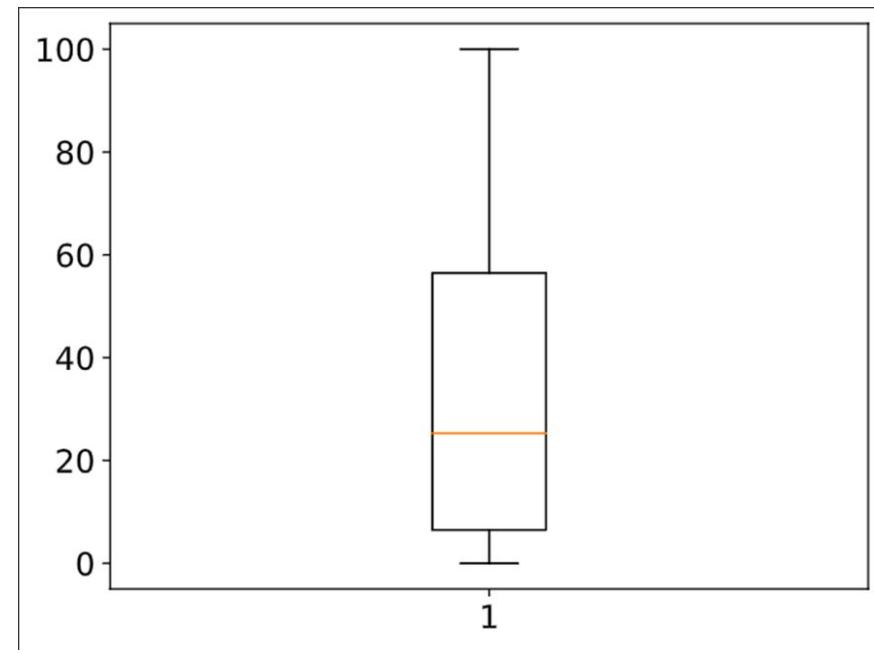
```
Grafico.bar(Dados_x,Dados_y/2, color =['r','b'])
```

Figura



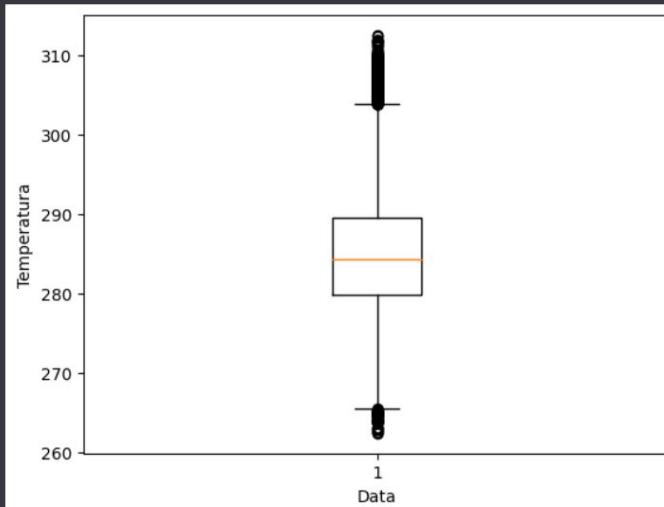
Biblioteca Matplotlib - Outros tipos de gráficos

```
plt.boxplot(Dados_y)
```



Biblioteca Matplotlib - Outros tipos de gráficos

```
plt.boxplot(df[ 'temperatura' ])  
  
plt.xlabel('Data')  
plt.ylabel('Temperatura')  
✓ 0.2s  
Text(0, 0.5, 'Temperatura')
```



Visualizações com o Matplotlib

```
import pandas as pd  
import matplotlib.pyplot as plt  
df=pd.read_csv('monitoramento_tempo.csv',sep=',')
```

| | temperatura | pressão | humidade | direção do vento | velocidade do vento | dia_da_semana | data |
|---|-------------|---------|----------|------------------|---------------------|---------------|---------------------|
| 0 | 282.080000 | 1024.0 | 81.0 | 0.0 | 0.0 | Domingo | 2012-10-01 12:00:00 |
| 1 | 282.080000 | 1024.0 | 81.0 | 0.0 | 0.0 | Domingo | 2012-10-01 13:00:00 |
| 2 | 282.083252 | 1024.0 | 80.0 | 4.0 | 0.0 | Domingo | 2012-10-01 14:00:00 |
| 3 | 282.091866 | 1024.0 | 80.0 | 18.0 | 0.0 | Domingo | 2012-10-01 15:00:00 |
| 4 | 282.100481 | 1024.0 | 80.0 | 31.0 | 0.0 | Domingo | 2012-10-01 16:00:00 |

Visualizações com o Matplotlib

df.info() - retorna o tipo dos dados

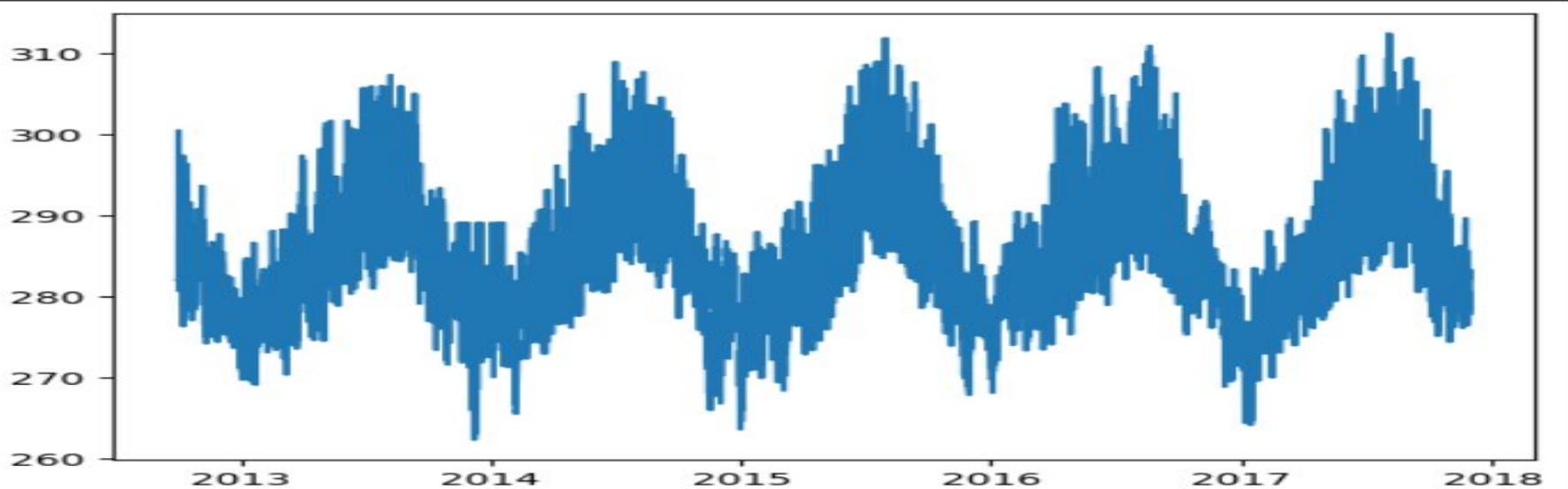
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45253 entries, 0 to 45252
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   temperatura     45253 non-null   float64
 1   pressão          45253 non-null   float64
 2   humidade         45253 non-null   float64
 3   direção do vento 45253 non-null   float64
 4   velocidade do vento 45253 non-null   float64
 5   dia_da_semana    45253 non-null   object 
 6   data              45253 non-null   object 
dtypes: float64(5), object(2)
memory usage: 2.4+ MB
```

Visualizações com o Matplotlib

```
import datetime
```

```
df['data']=pd.to_datetime(df['data']) – transforma a data em um objeto datetime
```

```
plt.plot(df['data'],df['temperatura'])
```



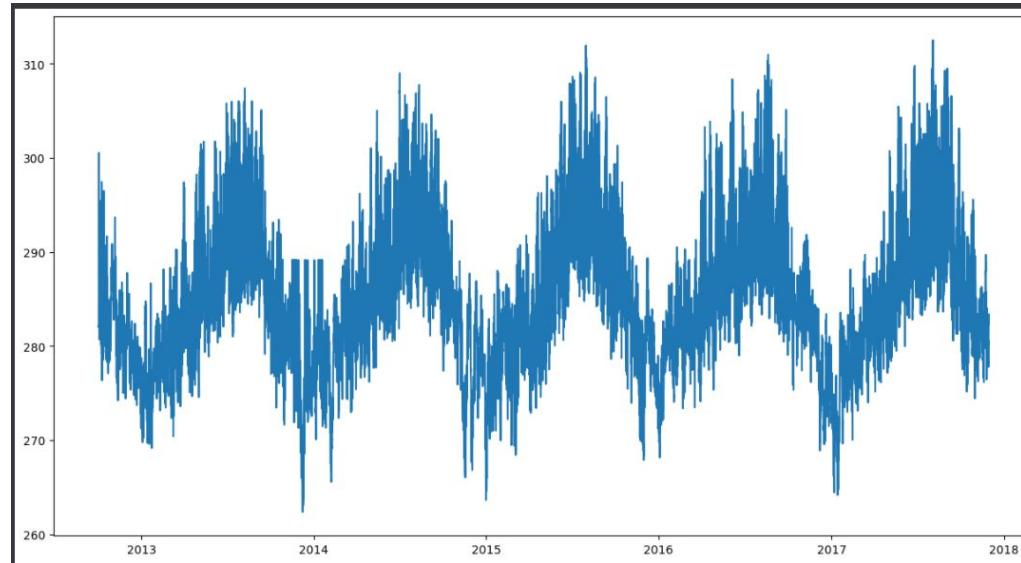
Visualizações com Matplotlib(Comando para converter tipo de dados com pandas)

```
df['temperatura']= df['temperatura'].astype('int')
```

No comando astype passa o tipo de dados que deseja converter a coluna do dataframe

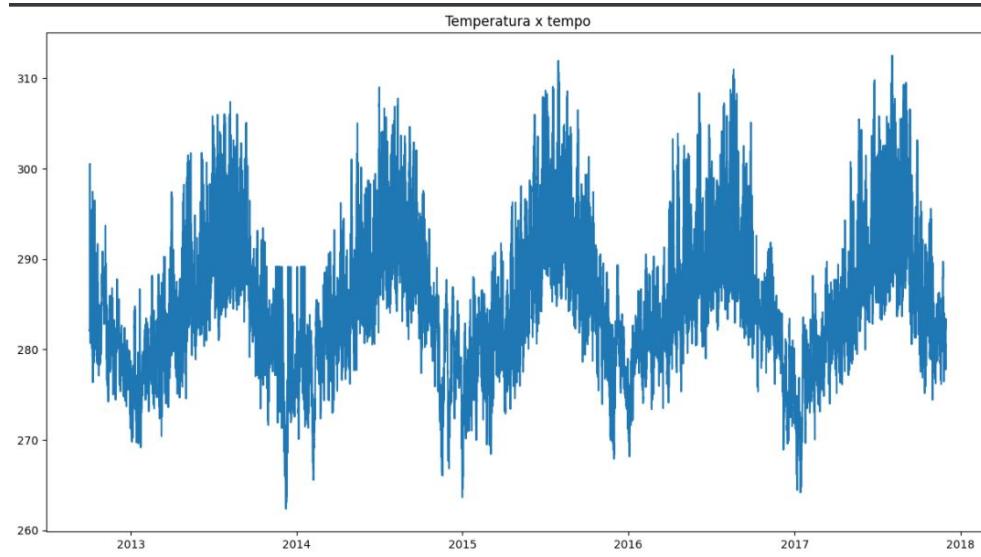
```
plt.figure(figsize=(15,8))
```

```
plt.plot(df['data'],df['temperatura'])
```



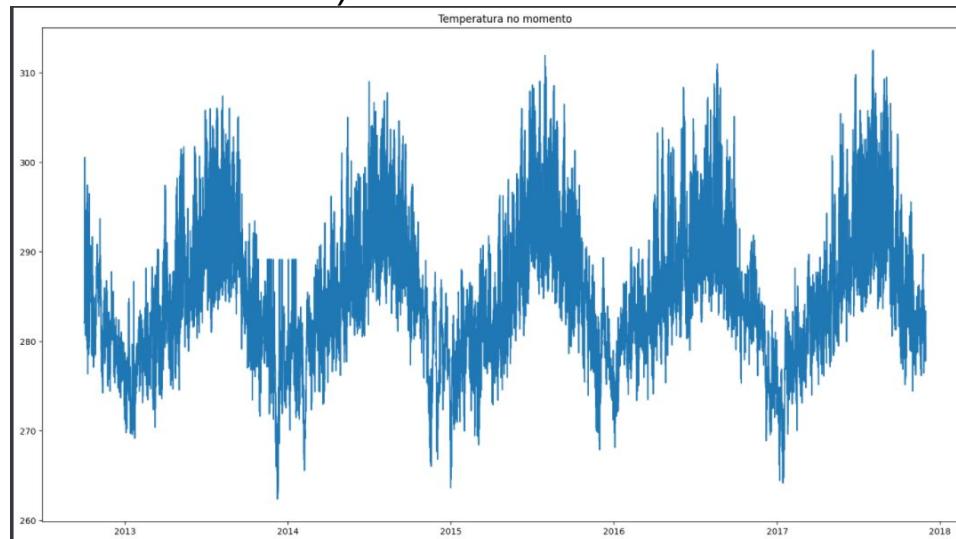
Visualizações com Matplotlib(Comando para converter tipo de dados com pandas)

```
plt.figure(figsize=(15,8))  
plt.title('Temperatura x tempo')  
plt.plot(df['data'],df['temperatura'])
```



Visualizações com Matplotlib(Comando para converter tipo de dados com pandas)

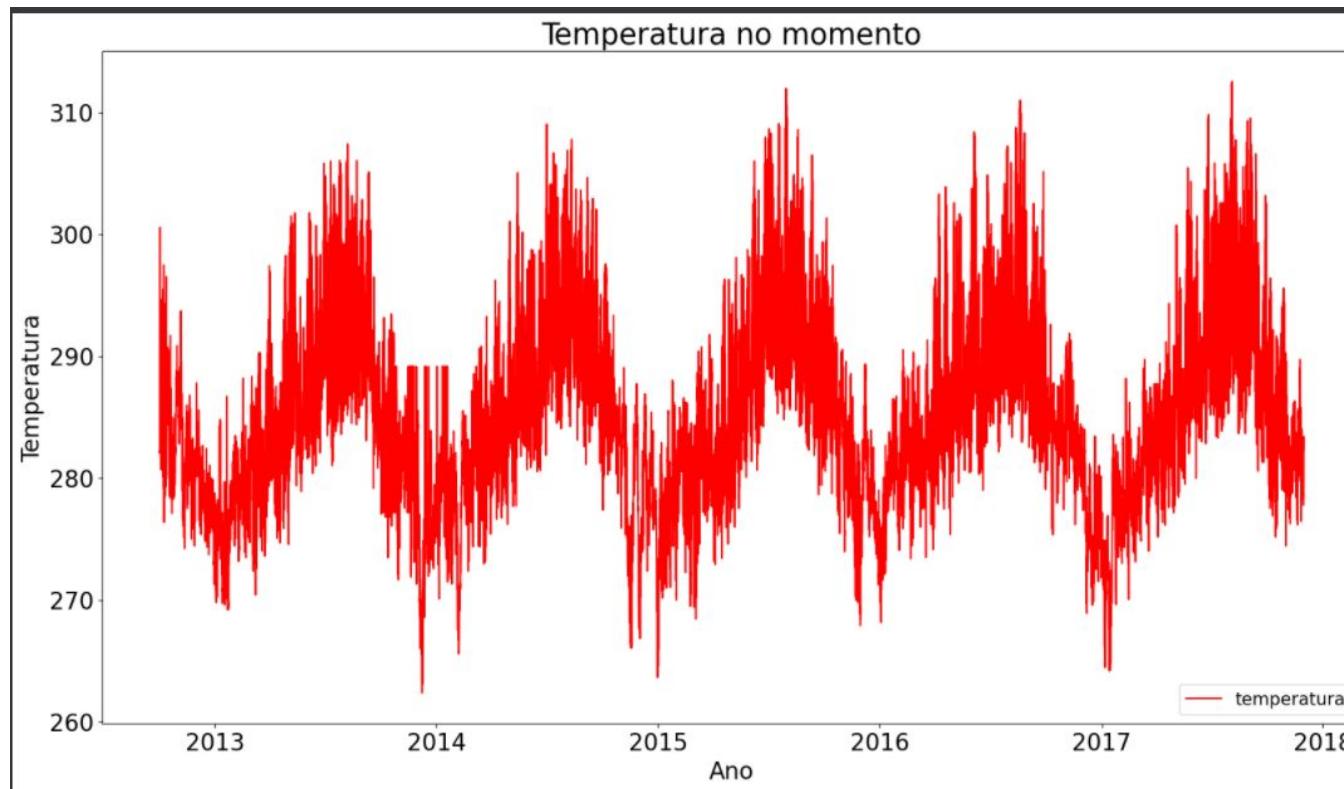
```
fig = plt.figure(figsize=(15,8))
eixo = fig.add_axes([0,0,1,1])
eixo.plot(df['data'],df['temperatura'])
eixo.set_title('Temperatura no momento')
```



Visualizações com Matplotlib(Alterando propriedades dos gráficos)

```
fig = plt.figure(figsize=(15,8))
eixo = fig.add_axes([0,0,1,1])
eixo.plot(df['data'],df['temperatura'])
eixo.set_title('Temperatura no momento',fontsize=25)
eixo.legend(['temperatura'],loc ='lower right',fontsize=15)
eixo.set_ylabel('Temperatura',fontsize=20)
eixo.set_xlabel('Ano',fontsize=20)
```

Visualizações com Matplotlib (Alterando propriedades dos gráficos)



Visualizações com Matplotlib(Afterando propriedades dos gráficos)

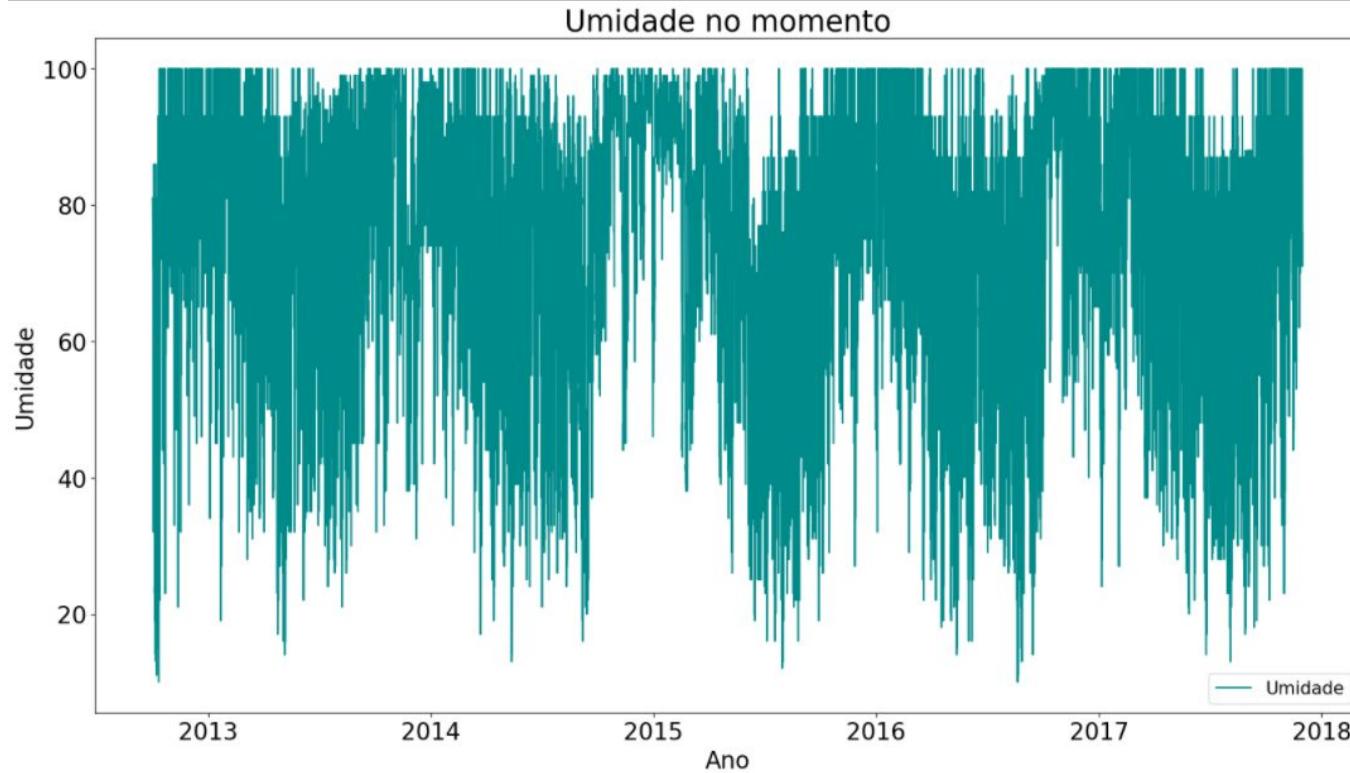


Exercícios

```
fig = plt.figure(figsize=(15,8))
plt.rcParams['font.size']=20
eixo = fig.add_axes([0,0,1,1])
eixo.plot(df['data'],df['humidade'],color='darkcyan')
eixo.set_title('Umidade no momento',fontsize=25)
eixo.legend(['Umidade'],loc ='lower right',fontsize=15)

eixo.set_ylabel('Umidade',fontsize=20)
eixo.set_xlabel('Ano',fontsize=20)
```

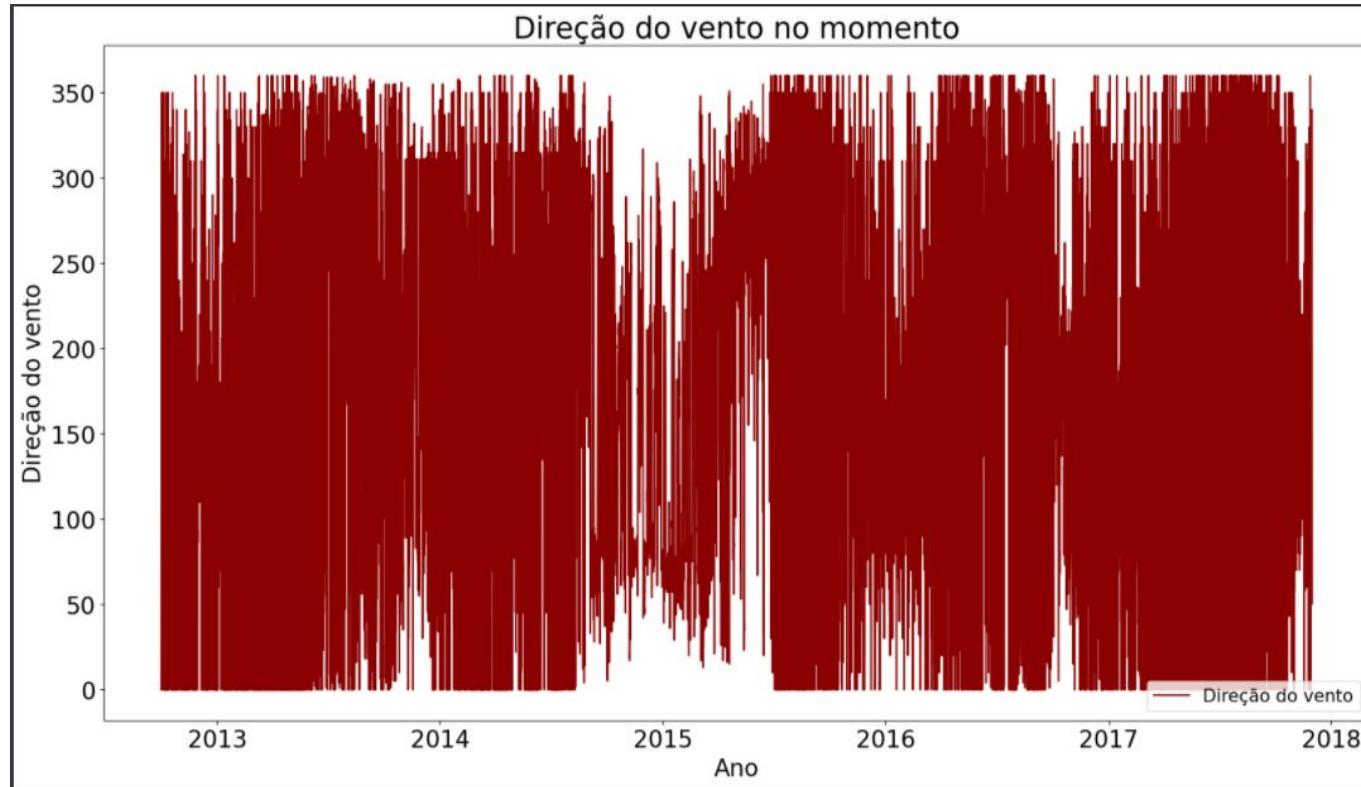
Exercícios



Exercícios

```
fig = plt.figure(figsize=(15,8))
plt.rcParams['font.size']=20
eixo = fig.add_axes([0,0,1,1])
eixo.plot(df['data'],df['direção do vento'],color='darkred')
eixo.set_title('Direção do vento no momento',fontsize=25)
eixo.legend(['Direção do vento'],loc ='lower right',fontsize=15)
eixo.set_ylabel('Direção do vento',fontsize=20)
eixo.set_xlabel('Ano',fontsize=20)
```

Exercícios

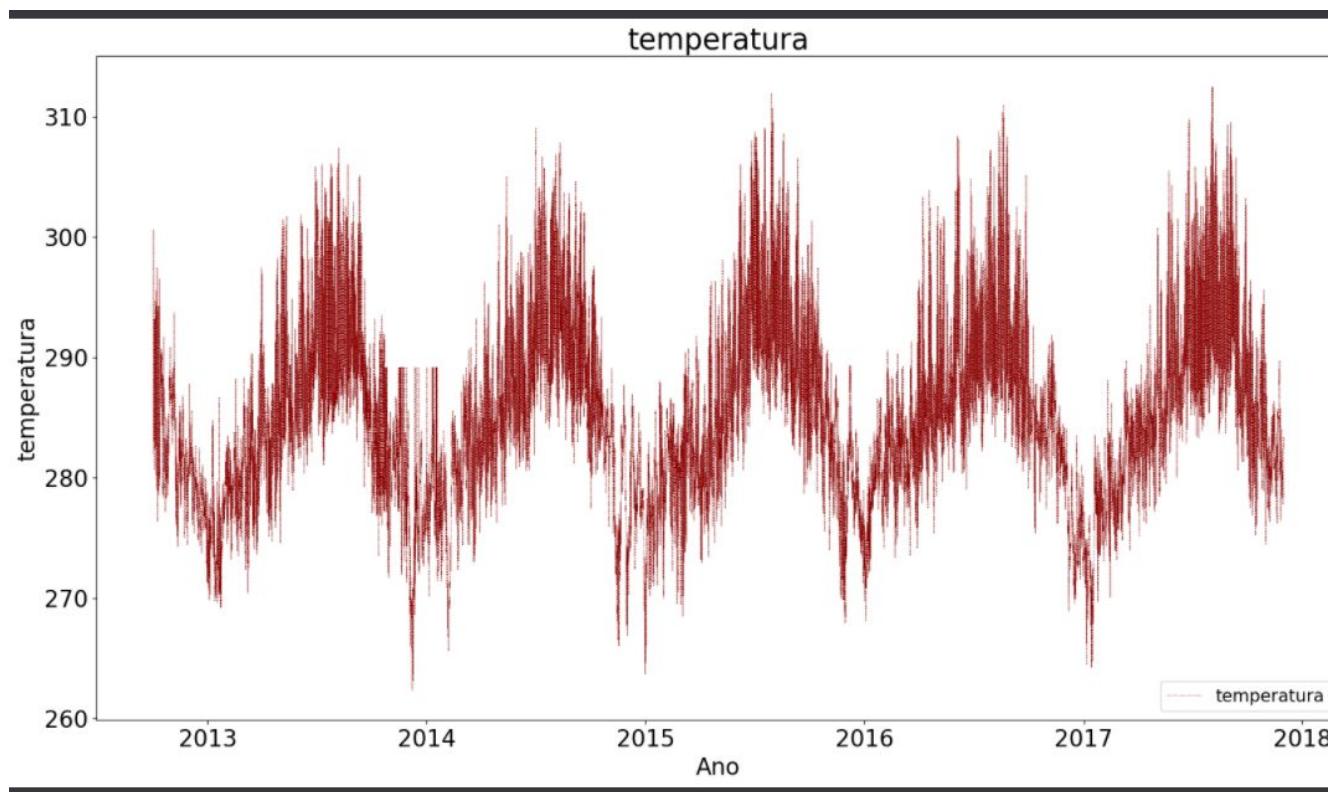


Matplotlib - Alterando parâmetros

```
fig = plt.figure(figsize=(15,8))
plt.rcParams['font.size']=20'
eixo = fig.add_axes([0,0,1,1])
eixo.plot(df['data'],df['temperatura'],color='darkred',lw=0.7,linestyle = 'dotted')
eixo.set_title('temperatura',fontsize=25)
eixo.legend(['temperatura'],loc ='lower right',fontsize=15)

eixo.set_ylabel('temperatura',fontsize=20)
eixo.set_xlabel('Ano',fontsize=20)
```

Matplotlib - Alterando parâmetros

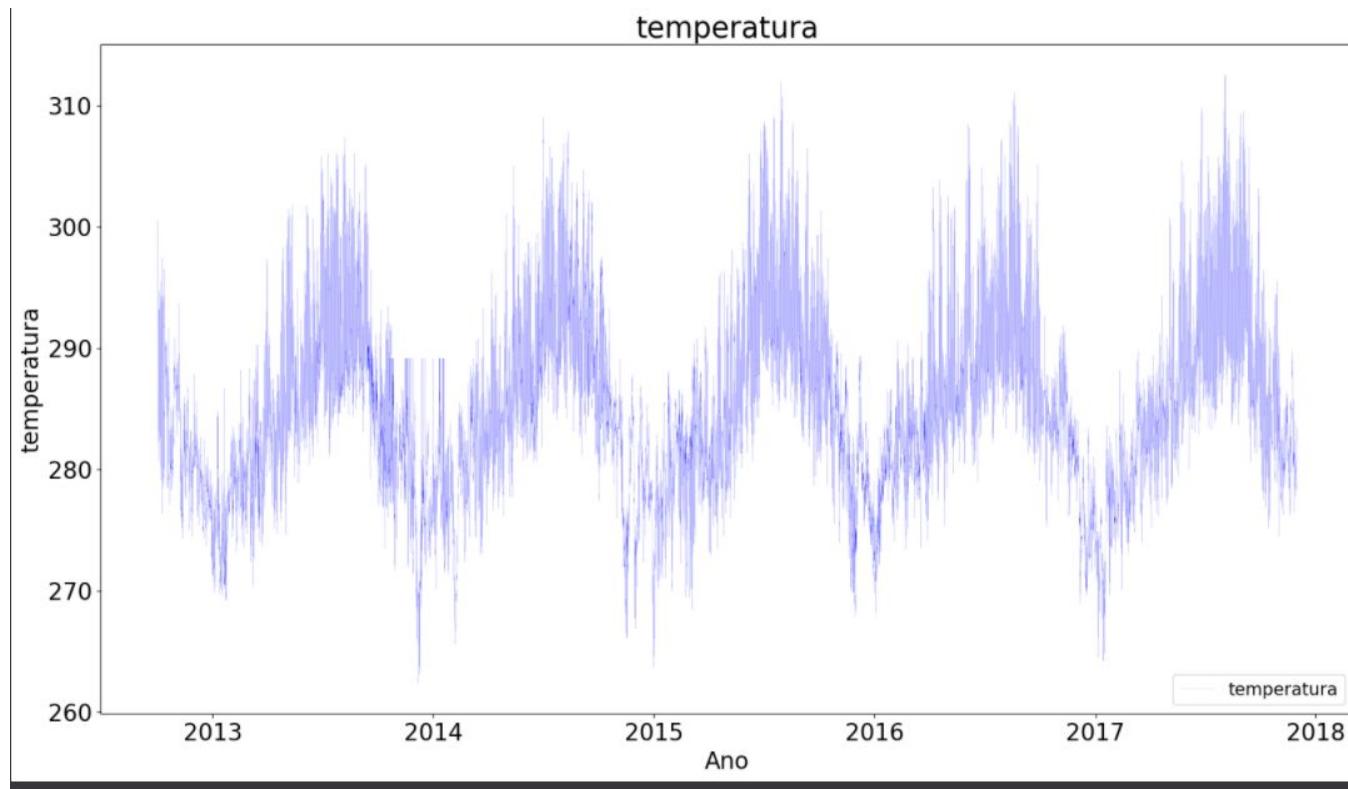


Matplotlib - Alterando parâmetros

```
fig = plt.figure(figsize=(15,8))
plt.rcParams['font.size']=20
eixo = fig.add_axes([0,0,1,1])
eixo.plot(df['data'],df['temperatura'],color='b',lw=0.1,linestyle = '--',marker="")
eixo.set_title('temperatura',fontsize=25)
eixo.legend(['temperatura'],loc ='lower right',fontsize=15)

eixo.set_ylabel('temperatura',fontsize=20)
eixo.set_xlabel('Ano',fontsize=20)
```

Matplotlib - Alterando parâmetros

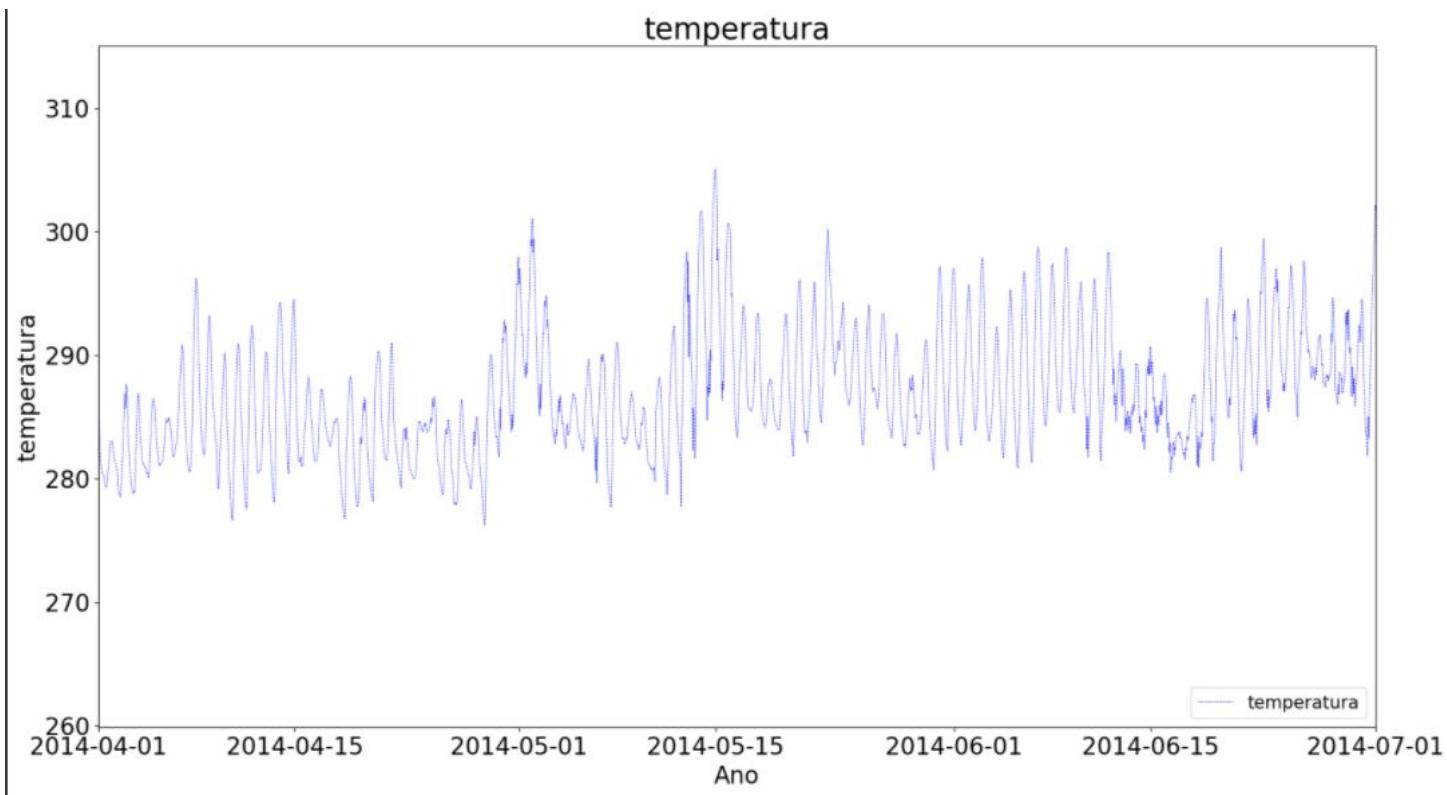


Matplotlib - Alterando parâmetros

```
fig = plt.figure(figsize=(15,8))
plt.rcParams['font.size']=20
eixo = fig.add_axes([0,0,1,1])
eixo.plot(df['data'],df['temperatura'],color='b',lw=0.5,linestyle = '--',marker="")
eixo.set_xlim(datetime.datetime(2014,4,1),datetime.datetime(2014,7,1))
eixo.set_title('temperatura',fontsize=25)
eixo.legend(['temperatura'],loc ='lower right',fontsize=15)

eixo.set_ylabel('temperatura',fontsize=20)
eixo.set_xlabel('Ano',fontsize=20)
```

Matplotlib - Alterando parâmetros

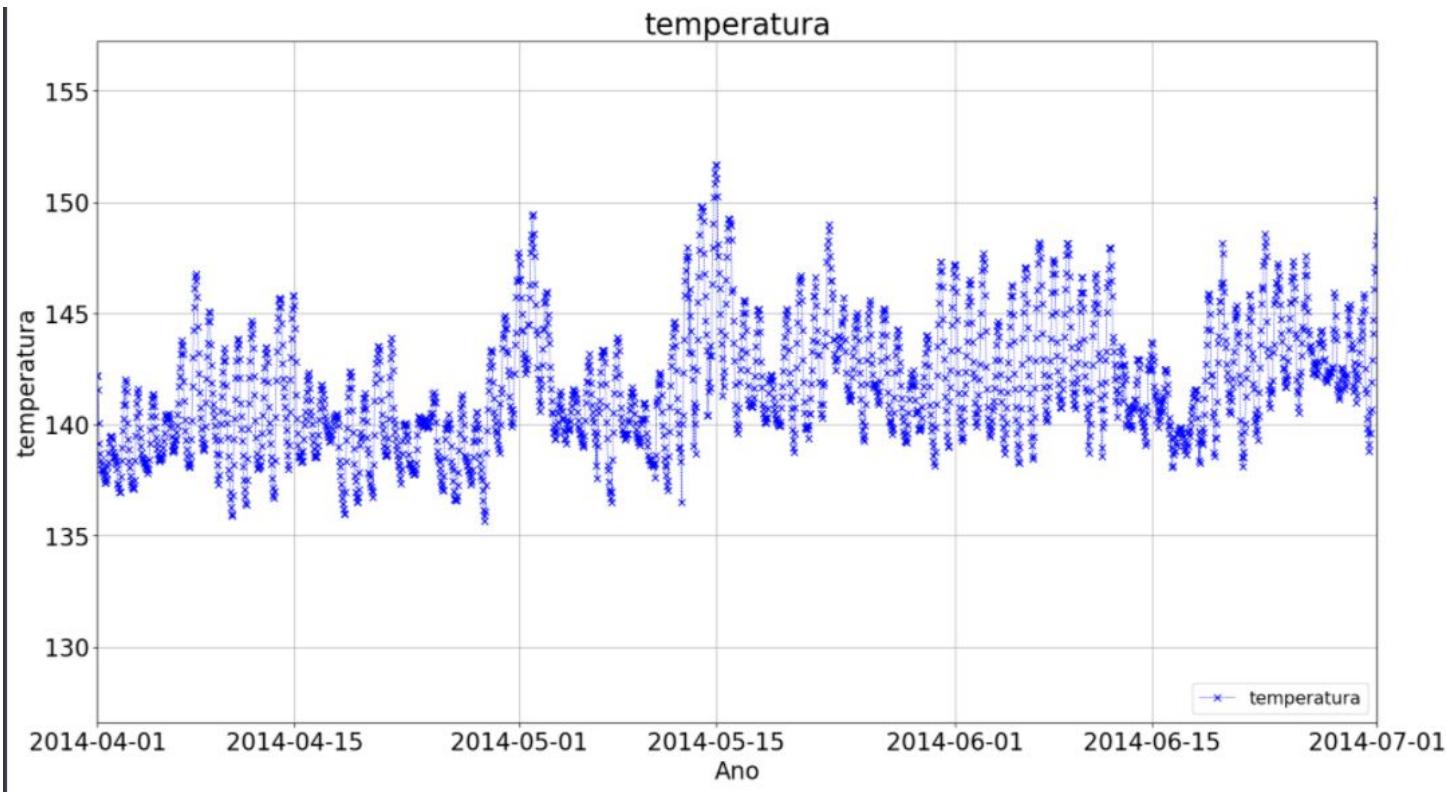


Matplotlib - Realizando cálculos em um dataset

```
df['temperatura']=(df['temperatura']-32) *(5/9)
fig = plt.figure(figsize=(15,8))
plt.rcParams['font.size']=20
eixo = fig.add_axes([0,0,1,1])
eixo.plot(df['data'],df['temperatura'],color='b',lw=0.5,linestyle = '--',marker='x')
eixo.set_xlim(datetime.datetime(2014,4,1),datetime.datetime(2014,7,1))
eixo.set_title('temperatura',fontsize=25)
eixo.legend(['temperatura'],loc ='lower right',fontsize=15)

eixo.set_ylabel('temperatura',fontsize=20)
eixo.set_xlabel('Ano',fontsize=20)
eixo.grid(True)
```

Matplotlib - Realizando cálculos em um dataset



Matplotlib - Plotando uma figura dentro da outra

```
fig = plt.figure(figsize=(15,5))
#plt.rcParams['font.size']=12
eixo = fig.add_axes([0,0,1,1])
eixo2 = fig.add_axes([0.35,0.60,0.5,0.3])
eixo.plot(df['data'],df['temperatura'],color = 'b',lw=0.5, linestyle='--', marker= "")
eixo.set_xlim(datetime.datetime(2016,1,1),datetime.datetime(2016,12,1))
eixo.tick_params(axis='x',labelsize=12) # Altera a fonte do eixo x do gráfico 2
eixo.set_title('Temperatura',fontsize =25)
eixo.set_ylim(200,400)
eixo.legend(['Temperatura'],loc = 'lower right', fontsize =15)
eixo.set_xlabel('Ano',fontsize=8)

eixo.set_ylabel('Temperatura',fontsize=20)
eixo.grid(True)
```

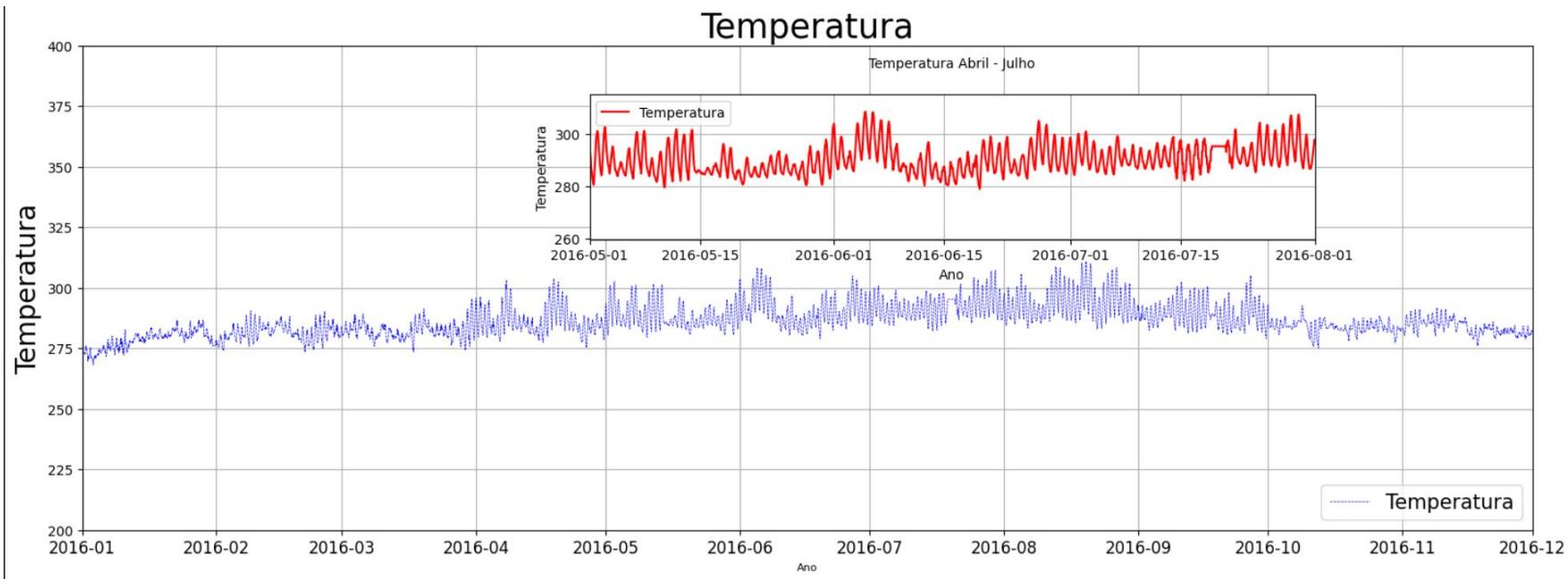
Matplotlib - Plotando uma figura dentro da outra

```
#eixo 2
eixo2.set_xlim(datetime.datetime(2016,5,1),datetime.datetime(2016,8,1))
eixo2.tick_params(axis='x',labelsize=10)

eixo2.plot(df['data'],df['temperatura'],color ='r')
eixo2.grid(True)

eixo2.set_title("Temperatura Abril - Julho",fontsize=10, pad =20)
eixo2.set_xlabel("Ano",fontsize=10)
eixo2.set_ylabel("Temperatura",fontsize=10)
eixo2.legend(['Temperatura'],loc='best',fontsize=10)
fig.savefig('Exercicio.jpg')
```

Matplotlib - Plotando uma figura dentro da outra



Matplotlib - Plotando uma figura dentro da outra

```
fig = plt.figure(figsize=(12,5))
eixo = fig.add_axes([0,0,1,1])
eixo.set_ylim(200,500)
eixo2 = fig.add_axes([0.35,0.50,0.5,0.3])
eixo.plot(df['data'],df['temperatura'],color= 'b')
eixo.grid(True)
eixo.set_xlim(datetime.datetime(2014,5,1),datetime.datetime(2014,6,1))
eixo.set_title('Temperatura no momento',fontsize=25, pad =20)
eixo.set_ylabel('Temperatura', fontsize=20)
eixo.set_xlabel('Ano', fontsize=20)
eixo.legend(['temperatura'],loc='best')
```

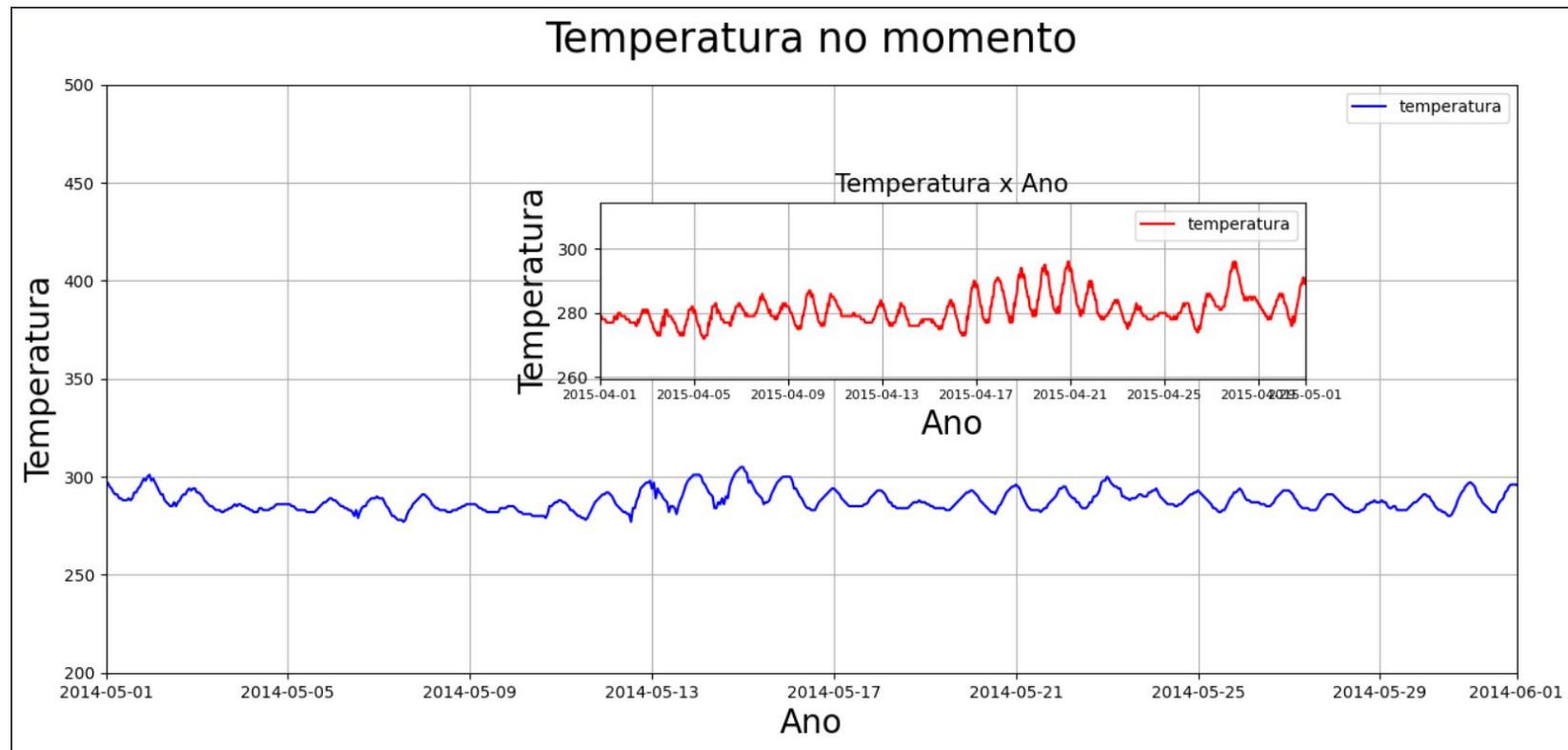
Matplotlib - Plotando uma figura dentro da outra

```
#eixo2
eixo2.plot(df['data'],df['temperatura'],color= 'r')

eixo2.set_xlim(datetime.datetime(2015,4,1),datetime.datetime(2015,5,1))
eixo2.tick_params(axis='x',labelsize=8)

eixo2.grid(True)
eixo2.set_title('Temperatura x Ano',fontsize=15,)
eixo2.set_ylabel('Temperatura', fontsize=20)
eixo2.set_xlabel('Ano', fontsize=20)
eixo2.legend(['temperatura'],loc='best',fontsize=10)
```

Matplotlib - Plotando uma figura dentro da outra



Matplotlib - Exercício

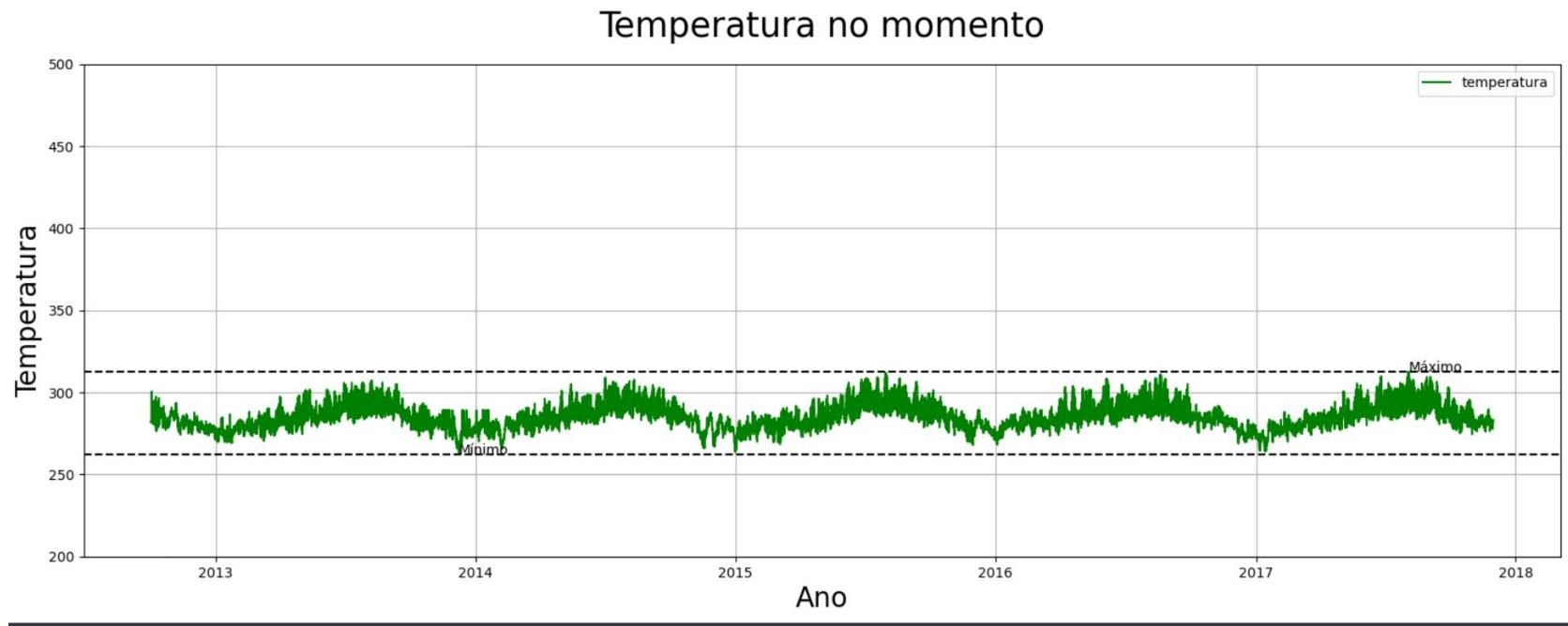
```
fig = plt.figure(figsize=(15,8))
eixo = fig.add_axes([0,0,1,1])
eixo.set_ylim(200,500)
eixo.plot(df['data'],df['temperatura'],color= 'g')
eixo.grid(True)
```

Matplotlib - Exercício

```
eixo.set_title('Temperatura no momento', fontsize=25, pad =20)
eixo.set_ylabel('Temperatura', fontsize=20)
eixo.set_xlabel('Ano', fontsize=20)
eixo.legend(['temperatura'], loc='best')

eixo.axhline(max(df['temperatura']), color='k', linestyle='--')
eixo.axhline(min(df['temperatura']), color='k', linestyle='--')
x1=df['data'][df['temperatura'].idxmax()]
y1 = max(df['temperatura'])
eixo.annotate('Máximo', xy=(x1,y1))
x1=df['data'][df['temperatura'].idxmin()]
y1 = min(df['temperatura'])
eixo.annotate('Mínimo', xy=(x1,y1))
```

Matplotlib - Exercício



Matplotlib - Exercício

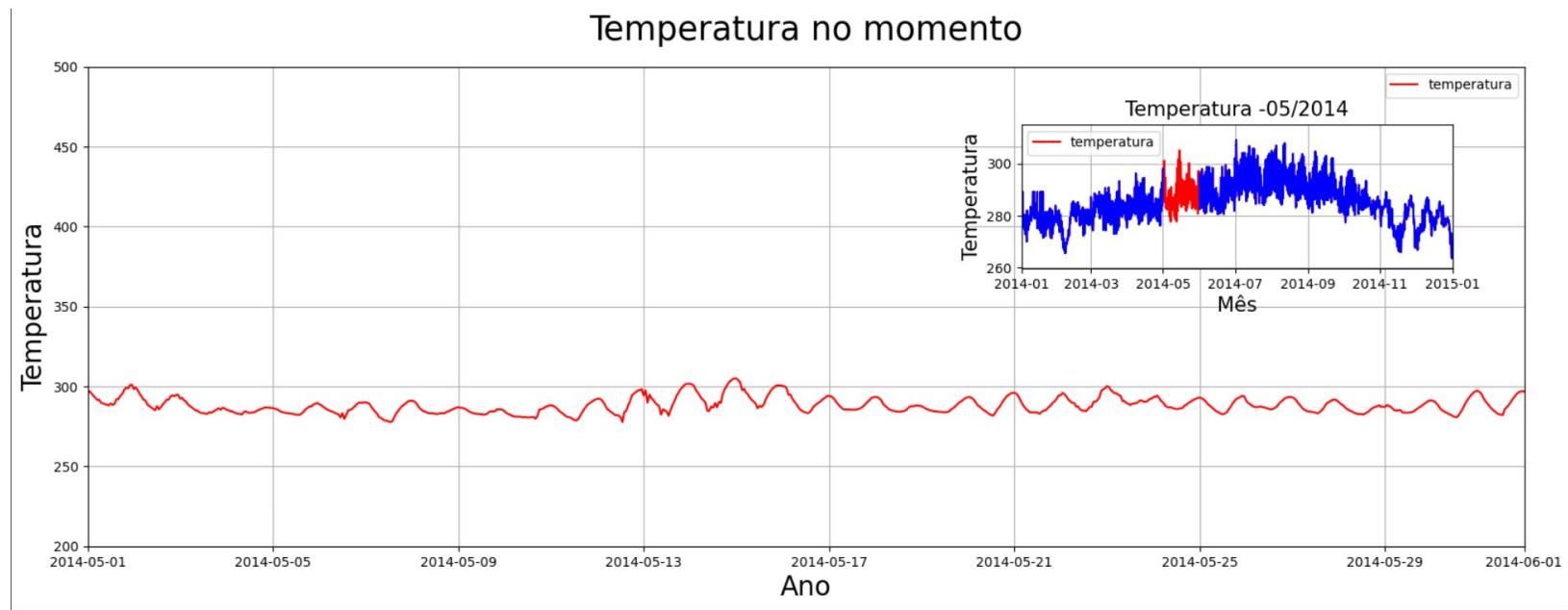
```
fig = plt.figure(figsize=(15,8))
plt.rcParams['font.size']=10'
eixo = fig.add_axes([0,0,1,1])
eixo.set_ylim(200,500)
eixo2 = fig.add_axes([0.65,0.58,0.3,0.3])
eixo.plot(df['data'],df['temperatura'],color= 'r')
eixo.grid(True)
eixo.set_xlim(datetime.datetime(2014,5,1),datetime.datetime(2014,6,1))
eixo.set_title('Temperatura no momento',fontsize=25, pad =20)
eixo.set_ylabel('Temperatura', fontsize=20)
eixo.set_xlabel('Ano', fontsize=20)
eixo.legend(['temperatura'],loc='best')
azul_esquerda = df['data'] <datetime.datetime(2014,5,1)
azul_direita = df['data'] >datetime.datetime(2014,6,1)
```

Matplotlib - Exercício

```
#eixo2
eixo2.set_xlim(datetime.datetime(2014,1,1),datetime.datetime(2015,1,1))
eixo2.plot(df['data'],df['temperatura'],color= 'r')
eixo2.plot(df[azul_esquerda]['data'],df[azul_esquerda]['temperatura'],color='b')
eixo2.plot(df[azul_direita]['data'],df[azul_direita]['temperatura'],color='b')

eixo2.grid(True)
eixo2.set_title('Temperatura -05/2014',fontsize=15,)
eixo2.set_ylabel('Temperatura', fontsize=15)
eixo2.set_xlabel('Mês', fontsize=15)
eixo2.legend(['temperatura'],loc='best',fontsize=10)
```

Matplotlib - Exercício



Matplotlib - Exercício

```
fig = plt.figure(figsize=(15,5))
eixo = fig.add_axes([0,0,1,1])
#eixo.set_ylim(200,500)

eixo.plot(df['data'],df['temperatura'],color= 'g')
eixo.grid(True)

eixo.set_title('Temperatura no momento',fontsize=25, pad =20)
eixo.set_ylabel('Temperatura', fontsize=20)
eixo.set_xlabel('Ano', fontsize=20)
eixo.legend(['temperatura'],loc='best')
```

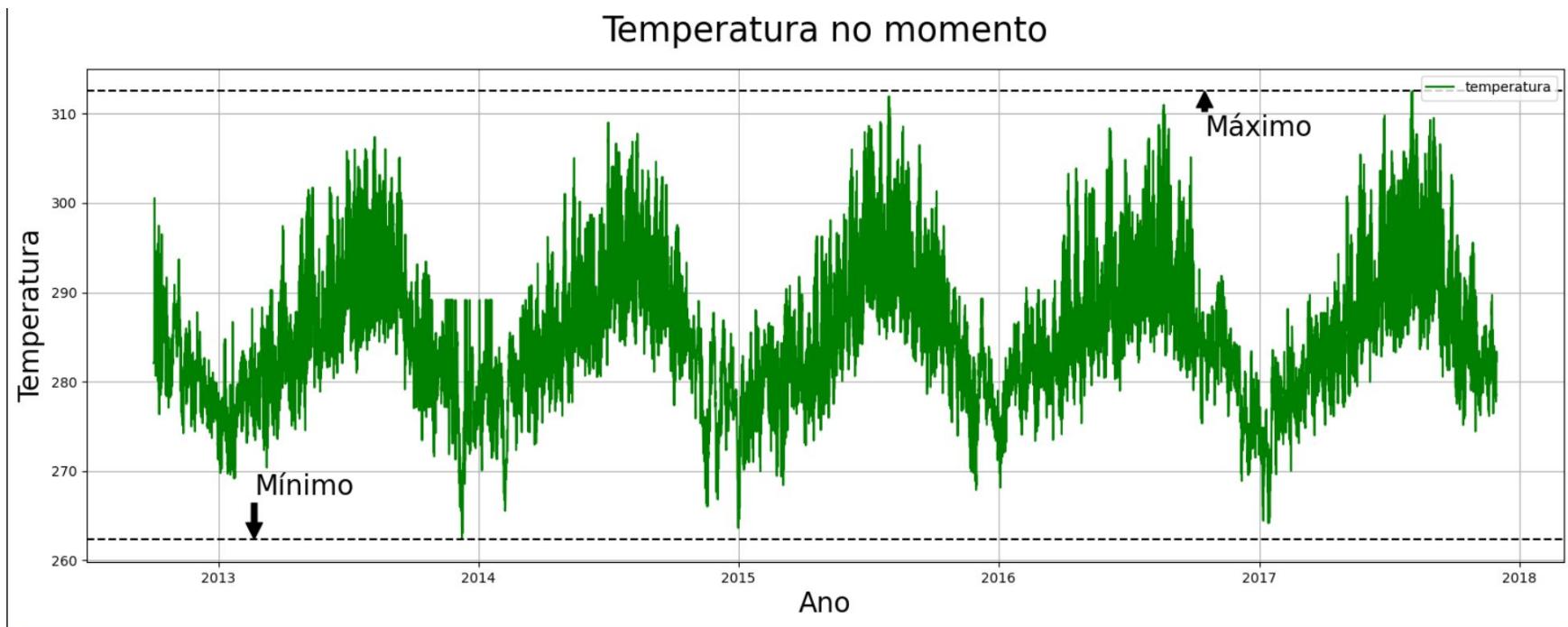
Matplotlib - Exercício

```
eixo.axhline(max(df['temperatura']),color='k',linestyle= '--')
eixo.axhline(min(df['temperatura']),color='k',linestyle= '--')
```

```
x1=df['data'][df['temperatura'].idxmax()]
y1 = max(df['temperatura'])
x2= x1=df['data'][df['temperatura'].idxmax()- 7000]
y2= max(df['temperatura']) -5
eixo.annotate('Máximo',xy=(x1,y1),fontsize=20,xytext=(x2,y2), arrowprops=dict(facecolor='k'))
```

```
x1=df['data'][df['temperatura'].idxmin()]
y1 = min(df['temperatura'])
x2= x1=df['data'][df['temperatura'].idxmin()- 7000]
y2= min(df['temperatura']) +5
eixo.annotate('Mínimo',xy=(x1,y1),fontsize=20,xytext=(x2,y2), arrowprops=dict(facecolor='k'))
```

Matplotlib - Exercício



Matplotlib -Plotando histograma

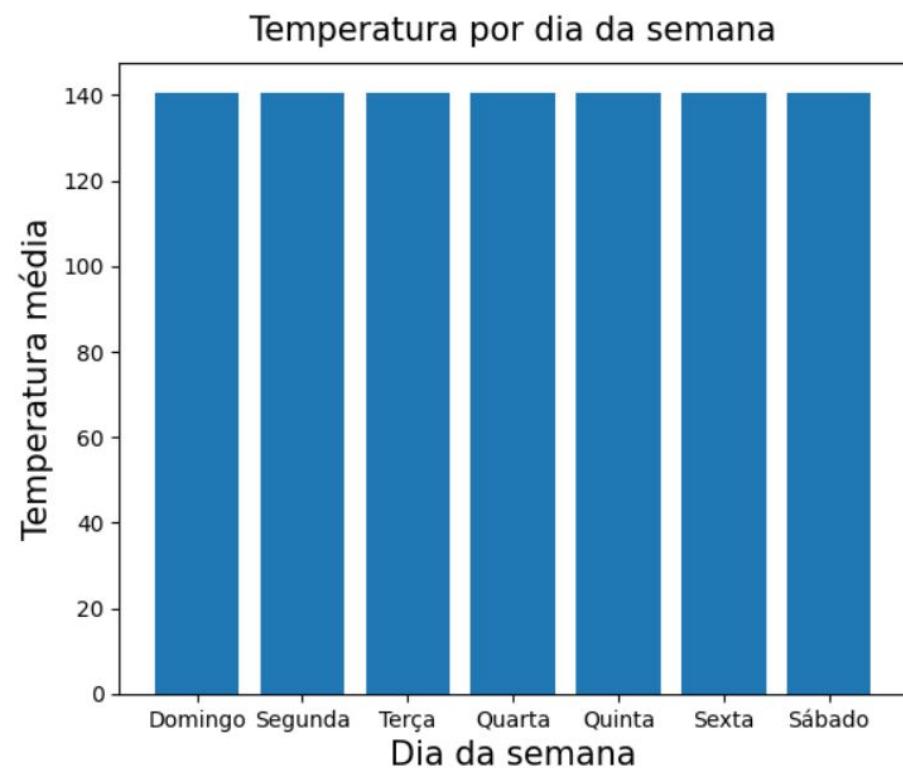
```
temperatura_dia_semana =  
df.groupby('dia_da_semana')[‘temperatura’].mean()  
temperatura_dia_semana  
nome_dia=[‘Domingo’,‘Segunda’,‘Terça’,‘Quarta’,‘Quinta’,‘Sexta’,‘Sábado’]  
temperatura_dia_semana = temperatura_dia_semana[nome_dia]
```

Matplotlib -Plotando histograma

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

eixo.bar(indice, temperatura_dia_semana)
eixo.set_title('Temperatura por dia da semana', fontsize=15, pad=10)
eixo.set_xlabel('Dia da semana', fontsize=15)
eixo.set_ylabel('Temperatura média', fontsize=15)
eixo.set_xticks(indice)
eixo.set_xticklabels(nome_dia)
```

Matplotlib -Plotando histograma



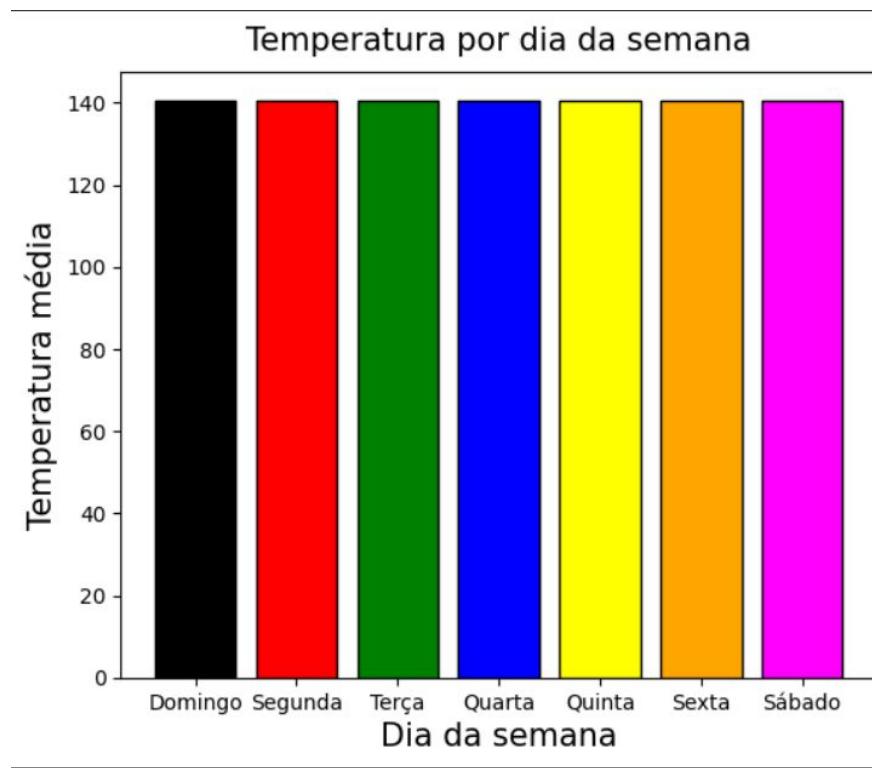
Matplotlib -Plotando histograma

```
fig = plt.figure(figsize=(5,4))
eixo = fig.add_axes([0,0,1,1])
indice = range(len(temperatura_dia_semana))
cores = ['black', 'r', 'g', 'b', 'yellow', 'orange', 'magenta']

eixo.bar(indice, temperatura_dia_semana,color=cores,edgecolor = 'black')

eixo.set_title('Temperatura por dia da semana', fontsize=15, pad=10)
eixo.set_xlabel('Dia da semana', fontsize=15)
eixo.set_ylabel('Temperatura média', fontsize=15)
eixo.set_xticks(indice)
eixo.set_xticklabels(nome_dia)
```

Matplotlib -Plotando histograma



Exercícios

- 1) Plote o gráfico umidade x ano, colocando título na imagem, título dos eixos, alterando o tamanho da imagem e da fonte
- 2) Plote o gráfico velocidade do vento x ano, colocando título na imagem, título dos eixos, alterando o tamanho da imagem e da fonte

Biblioteca Matplotlib - Outros tipos de gráficos

- 3) Realizar anotações no gráfico de umidade
- 4) Realizar anotações no gráfico de direção do vento
- 5) Realizar anotações no gráfico de pressão
- 6) Plotar boxplot para a umidade
- 7) Plotar boxplot para a direção do vento
- 8) Plotar boxplot para a pressão

Matplotlib - Exercício

- 9) Escolher o período de 05-2016 - 08-2016 e plotar o gráfico de temperatura no gráfico maior e no gráfico menor plotar o gráfico do ano de 2016 do mês 1 ao 12
- 10) Escolher o período de 05-2017 - 08-2017 e plotar o gráfico de temperatura no gráfico maior e no gráfico menor plotar o gráfico da umidade para o mesmo período

Obrigado!

Prof. Me Daniel Vieira

Email: danielvieira2006@gmail.com

Linkedin: Daniel Vieira

Instagram: Prof daniel.vieira95

