

Aula de CDM - Ambiente Compartilhado

Boa dia pessoal !

Prof. Yoshi

Aula CDM - 01 - 20/03/2020 - Lista 01

Vamos começar a aula às 10h...

Pessoal, voces tem instalado o anaconda no PC de voces ?

Já volto...voltei

Tudo bem pessoal ?

Programa Exercicio 02 - Matplotlib - Fazer gráficos

#data: 27/03/2020

#Por: Yoshi

```
import numpy as np #biblioteca do matlab
import pandas as pd # biblioteca R
import matplotlib.pyplot as plt # biblioteca de graficos
```

```
tp = [24.2, 24.6, 23.6, 21.1, 18.9, 16.7, 16.5, 16.9, 17.9, 19.6, 21.3, 23.0]; # criar uma list com 12 várias variáveis
(temp média de floripa)
```

```
tp2 = np.array(tp); # converto minha list em um array (utilizo a biblioteca numpy para converter a list em um array)
```

```
mes=np.arange(1,13,1); # faz um array de 1 a 12 , com step de 1
```

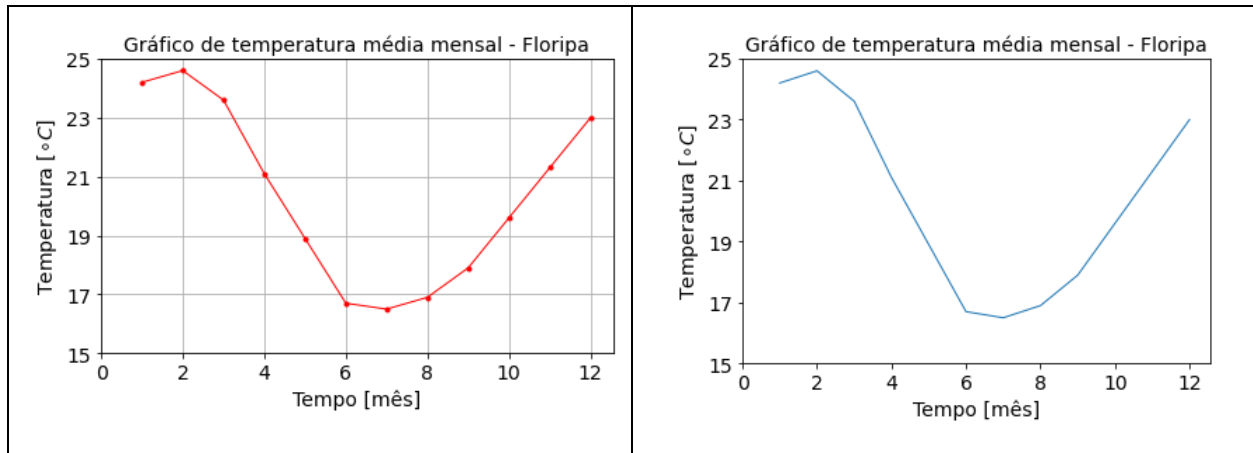
```
plt.plot(mes,tp2,linewidth=1.0, color='r',marker='.') # faço o gráfico da minha variável tp2
```

```
plt.title('Gráfico de temperatura média mensal - Floripa',fontsize=14) # insere título no gráfico com fonte de 14
```

```
plt.xlabel('Tempo [mês]',fontsize=14) # insere nome no eixo X com fonte de 14
```

```
plt.ylabel('Temperatura [°C]',fontsize=14) # insere o nome no eixo y com fonte de 14
```

```
plt.xticks(np.arange(0,13, step=2),fontsize=14) # defino a descrição da escala de 0 a 12 com step de 1
plt.yticks(np.arange(15,27, step=2),fontsize=14) # defino a descrição da escala de 15 a 27 com step de 2
plt.grid() # desenha as linhas de grade
plt.tight_layout() # ajusta para o tamanho da janela
```



Fim de aula

Aula CDM - 02 - 03/27/2020 - Lista 02

Próxima aula 27/03/20 - finalizar o gráfico e sua formatação, fazer gráfico com 2 variáveis

Exercício 02, Exemplo 02 - Criar um gráfico com 2 variáveis (segundo eixo)

#data: 27/03/2020

#Por: Yoshi

```
import numpy as np #biblioteca do matlab
import pandas as pd # biblioteca R
import matplotlib.pyplot as plt # biblioteca de graficos
```

_____variáveis_____

tp = [24.2, 24.6, 23.6, 21.1, 18.9, 16.7, 16.5, 16.9, 17.9, 19.6, 21.3, 23.0]; # criar uma list com 12 várias variáveis
(temp média de floripa)

tp2 = np.array(tp); # converto minha list em um array (utilizo a biblioteca numpy para converter a list em um array)

```
chuva = np.array([162.7, 196.9, 173.0, 92.8, 96.9, 89.5, 99.5, 95.3, 134.2, 109.8, 130.2, 137.0]); # cria um array dos valores de chuva mensal
```

```
mes=np.arange(1,13,1); # faz um array de 1 a 12 , com step de 1
```

```
# _____ Gráfico _____
```

```
fig, ax1 = plt.subplots() # defino o nome da minha figura (fig) e nome do meu eixo (ax1)
```

```
ax2 = ax1.twinx() # defino o nome do meu segundo eixo (ax2)
```

```
ax1.bar(mes,chuva, color='b',label = 'Precipitação') # faço o gráfico de precipitação no eixo (ax1), lado esquerdo
```

```
ax2.plot(mes,tp, color='r', linewidth=1,marker='o',label = 'Temperatura') # faço o gráfico de temperatura no eixo (ax2), lado direito
```

```
plt.title('Gráfico de Temp e Precipitação média mensal - Floripa',fontsize=14) # título do gráfico
```

```
ax1.set_xlabel('Tempo [mês]',fontsize=14) # eixo X é comum para as duas variáveis
```

```
ax1.set_ylabel('Precipitação [mm]',fontsize=14) # defino o título no eixo ax1
```

```
ax2.set_ylabel('Temperatura [°C]',fontsize=14) # defino o título no eixo ax2
```

```
# Ajuste a escala e a faixa da variável
```

```
ax1.xaxis.set(ticks=range(0,13,1),ticklabels=[' ','jan','fev','mar','abr','mai','jun','jul','ago','set','out','nov','dez',""])
```

```
ax1.yaxis.set(ticks=range(0,220,20))
```

```
ax2.yaxis.set(ticks=range(15,30,2))
```

```
fig.legend(loc=(0.6,0.75)) # mostra a legenda
```

```
# faixa toda da variável
```

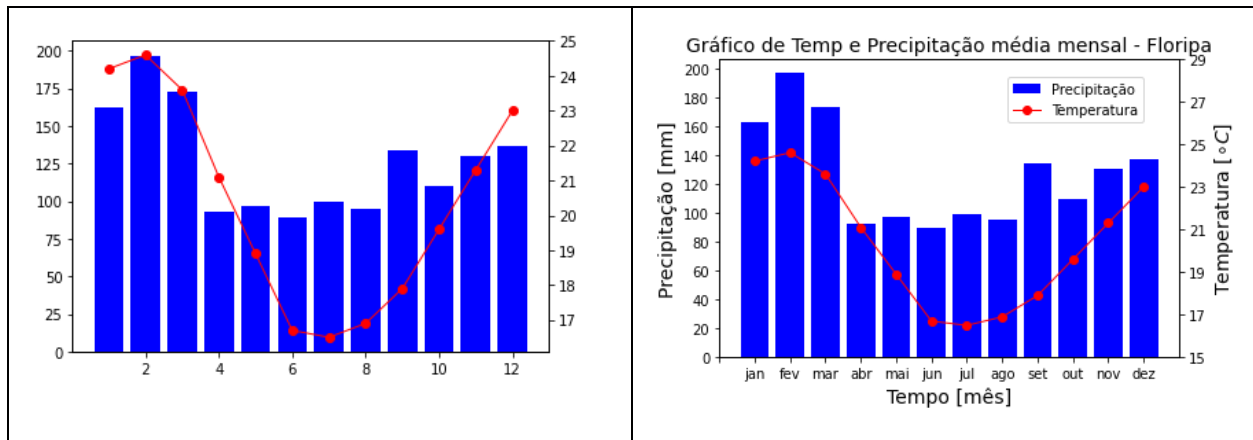
```
ax1.set_xlim((0,13))
```

```
ax2.set_ylim((15,29))
```

```
#ax1.grid()
```

```
plt.show()
```

```
plt.tight_layout()
```



Exercício 02, Exemplo 03 - Criar Múltiplos gráficos (subplot)

#data: 27/03/2020

#Por: Yoshi

```
import numpy as np #biblioteca do matlab
import pandas as pd # biblioteca R
import matplotlib.pyplot as plt # biblioteca de graficos
```

_____variáveis_____

tp = np.array([24.2, 24.6, 23.6, 21.1, 18.9, 16.7, 16.5, 16.9, 17.9, 19.6, 21.3, 23.0]); # cria um array dos valores de temperatura mensal

chuva = np.array([162.7, 196.9, 173.0, 92.8, 96.9, 89.5, 99.5, 95.3, 134.2, 109.8, 130.2, 137.0]); # cria um array dos valores de chuva mensal

pressao = np.array([1010.5, 1011.1, 1012.4, 1014.5, 1015.7, 1017.0, 1018.1, 1017.3, 1016.7, 1014.6, 1012.1, 1010.9]); # cria um array dos valores de pressao mensal

umidade = np.array([81.0, 82.0, 82.0, 82.0, 83.0, 83.0, 84.0, 83.0, 83.0, 81.0, 80.0, 80.0]);# cria um array dos valores de umidade mensal

mes=np.arange(1,13,1); # faz um array de 1 a 12 , com step de 1

fig , axes = plt.subplots(4,1); # defino a minha figura para 04 subplots (04 gráficos)

```

(ax1,ax2,ax3,ax4)=axes # definir o nome dos meus eixos em uma matriz de 4 x 1
#(ax1,ax2],[ax3,ax4])=axes # definir o nome dos meus eixos em uma matriz de 2 x 2

ax1.plot(mes,tp, color='r', linewidth=1,marker='o')
ax2.plot(mes,umidade, color='m',linewidth=1,marker='*')
ax3.plot(mes,pressao, color='g', linewidth=1,marker='+')
ax4.bar(mes,chuva, color='b')

ax1.set_title('Gráfico Climatologia - Floripa',fontsize=14) # Faz o Título do gráfico (somente no primeiro)
ax4.set_xlabel('Tempo [mês]',fontsize=10) # Faz a descrição nos eixos X

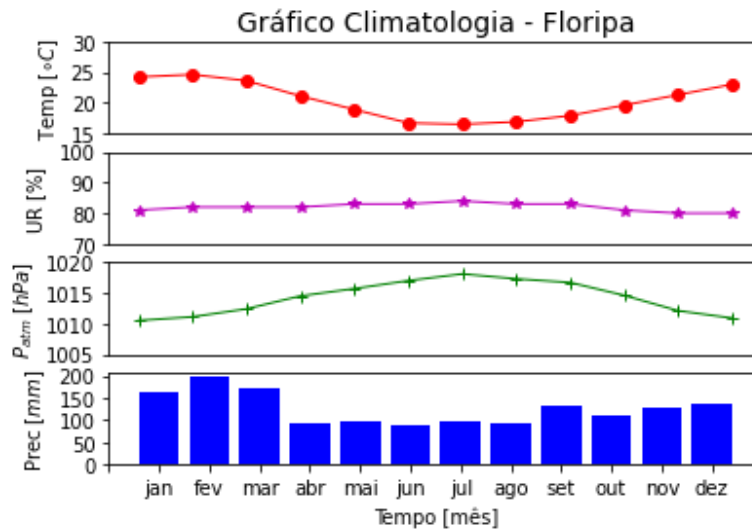
# Faz a descrição nos eixos Y
ax1.set_ylabel('Temp [ $^{\circ}$ C]',fontsize=10)
ax2.set_ylabel('UR [%]',fontsize=10)
ax3.set_ylabel(r'$P_{atm}$ [hPa]',fontsize=10)
ax4.set_ylabel('Prec [mm]',fontsize=10)

# Define a escala de cada variável
ax1.yaxis.set(ticks=range(15,35,5))
ax2.yaxis.set(ticks=range(70,110,10))
ax3.yaxis.set(ticks=range(1005,1025,5))
ax4.yaxis.set(ticks=range(0,220,50))
ax4.xaxis.set(ticks=range(0,13))
ax4.xaxis.set(ticklabels=[' ','jan','fev','mar','abr','mai','jun','jul','ago','set','out','nov','dez',""])

#ax1.legend(['Média Temp']);
ax1.xaxis.set_visible(False)
ax2.xaxis.set_visible(False)
ax3.xaxis.set_visible(False)
ax4.xaxis.set_visible(True)

plt.show()
plt.tight_layout()

```



Fim de aula !!

Aula CDM - 03 - 30/03/2020 - Lista 03

Bom dia !!

Estou no aguardo de vocês....

Fazer um gráfico de correlação entre duas variáveis.

Exercício 02, Exemplo 04 - Faz o gráfico de Correlação

#data: 30/03/2020

#Por: Yoshi

import numpy as np #biblioteca do matlab

import pandas as pd # biblioteca R

import matplotlib.pyplot as plt # biblioteca de graficos

#_____variáveis_____

```

tp = np.array([24.2, 24.6, 23.6, 21.1, 18.9, 16.7, 16.5, 16.9, 17.9, 19.6, 21.3, 23.0]); # cria um array dos valores de
temperatura mensal
chuva = np.array([162.7, 196.9, 173.0, 92.8, 96.9, 89.5, 99.5, 95.3, 134.2, 109.8, 130.2, 137.0]); # cria um array dos
valores de chuva mensal
pressao = np.array([1010.5, 1011.1, 1012.4, 1014.5, 1015.7, 1017.0, 1018.1, 1017.3, 1016.7, 1014.6, 1012.1,
1010.9]); # cria um array dos valores de pressao mensal
umidade = np.array([81.0, 82.0, 82.0, 82.0, 83.0, 83.0, 84.0, 83.0, 83.0, 81.0, 80.0, 80.0]);# cria um array dos valores
de umidade mensal

mes=np.arange(1,13,1); # faz um array de 1 a 12 , com step de 1

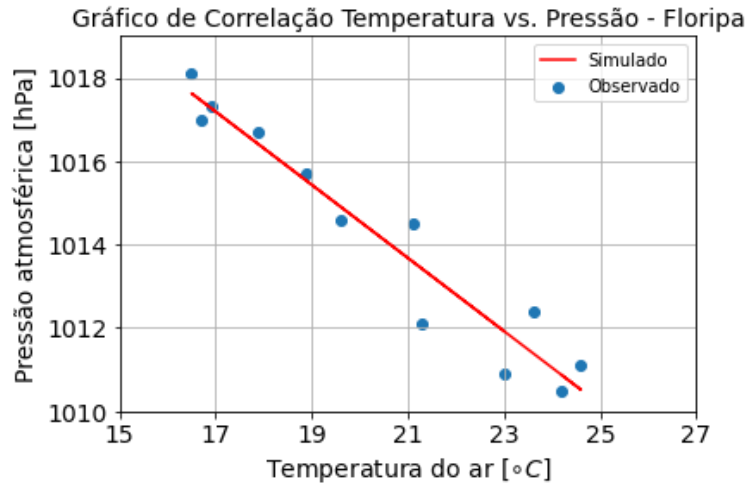
fig = plt.figure()

x=tp; y=pressao # defino temperatura no eixo X e pressão no eixo Y
plt.scatter(x,y) # faz gráfico de correlação entre temperatura e pressão
coef = np.polyfit(x,y,1) # meus coeficientes angular a=-0.88 e linear b=1.03 (eq. linear  $y(x) = ax + b$ )
f = np.poly1d(coef) # cria a função f(x), ou y(x)
plt.plot(x,f(x),'r') # faz o gráfico da curva ajustada

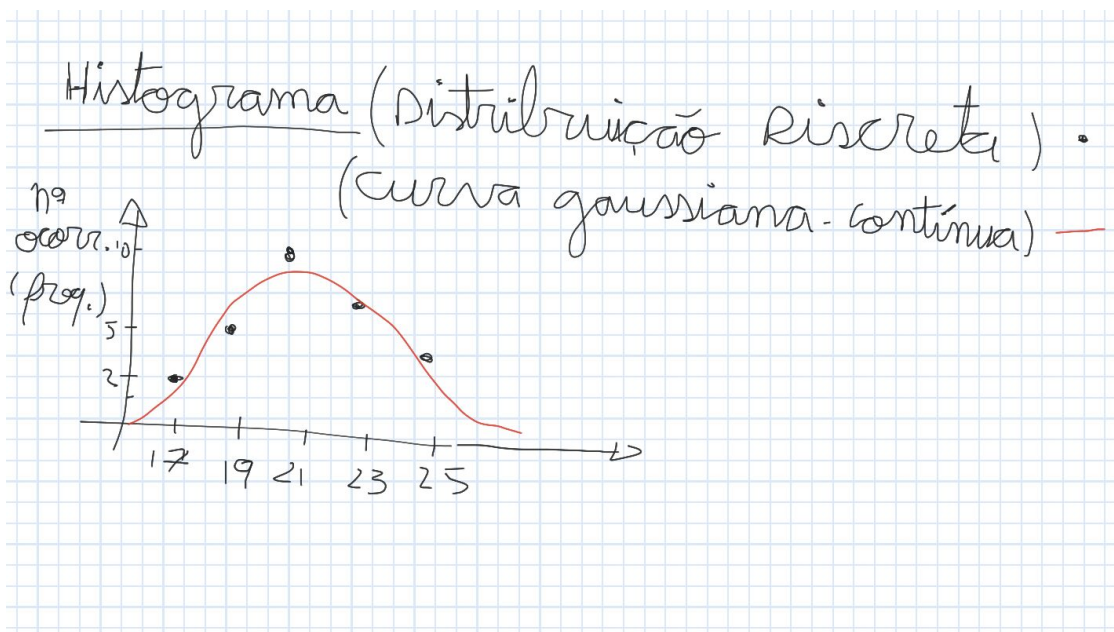
plt.title('Gráfico de Correlação Temperatura vs. Pressão - Floripa',fontsize=14)

plt.ylabel('Pressão atmosférica [hPa]',fontsize=14)
plt.xlabel('Temperatura do ar [ $^{\circ}$ C]',fontsize=14)
plt.yticks(np.arange(1010,1020, step=2),fontsize=14)
plt.xticks(np.arange(15,29, step=2),fontsize=14)
plt.legend(['Simulado','Observado']);
plt.xlim((15,27))
plt.ylim((1010,1019))
plt.grid()
plt.show() # mostra o gráfico na tela ( spyder, jupyter notebook), caso não esteja configurado para mostrar
automaticamente
plt.tight_layout() # ele ajusta o gráfico na janela da figura

```



Exercício 02, Exemplo 05 - Faz o gráfico de Histograma (Distribuição)



https://jamboard.google.com/d/17jdNhcb-09tRweJau7Op9Csrt_ueccTm5azxgIg-SOQ/viewer

Exercício 02, Exemplo 05 - Faz o gráfico de Histograma

#data: 30/03/2020

#Por: Yoshi

```
import numpy as np #biblioteca do matlab
```

```
import pandas as pd # biblioteca R
```

```
import matplotlib.pyplot as plt # biblioteca de graficos
```

```
from numpy import genfromtxt # Faz a leitura de cada linha e converte a string no formato de array (números -float)
```

```
uri="https://raw.githubusercontent.com/sakagamiyoshiaki/CDM/master/IFSC-032011.txt" # endereço onde estão os  
dados histórico (amostra de março de 2011 da estação do IFSC)
```

```
# ano, mes, dia, hora, min, temp, umid, press, chuva, vel, dir
```

```
data = genfromtxt(uri, delimiter=',')
```

```
fig =plt.figure()
```

```
y=data[:,5] # temperatura
```

```
bins2=np.arange(1,30,0.5) # são minhas "classes"
```

```
plt.hist(y,bins2); # Faz o gráfico de distribuição discreta (histograma)
```

```
plt.ylabel('Número de ocorrências [-]');
```

```
F=data[:,5]>15
```

```
y1=data[F,5]
```

```
#plt.hist(y1,bins2); # Faz o gráfico de distribuição discreta (histograma) - com filtro temperatura <15 , excluído
```

```
#count, bins, ignored = plt.hist(y1, 30, density=True)
```

```
#count*8033 # tenho o total de ocorrências em cada bin
```

Fim de aula

Aula CDM - 04 - 03/04/2020 - Lista 03

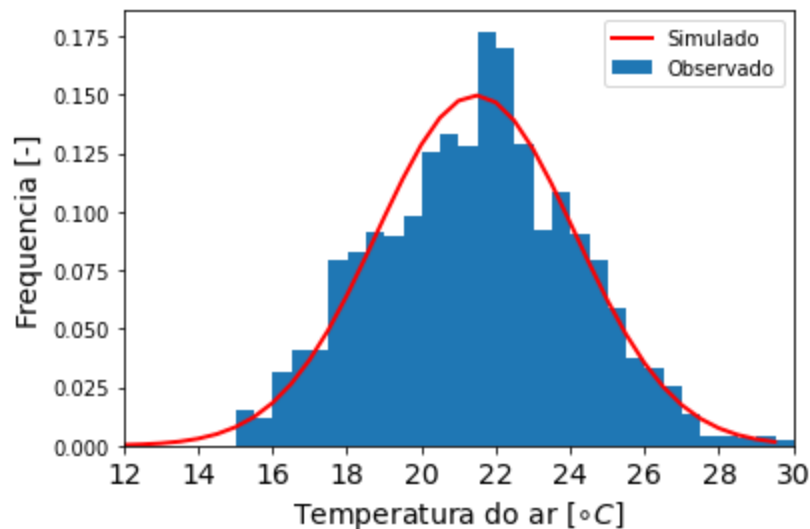
Próxima aula sexta 03/04/2020 (continuação dos gráficos - histograma - distribuição gaussiana - normal)

#Continuação do script acima - Programa para fazer a curva normal (gaussiana)

```
count, bins, ignored = plt.hist(y1, 30, density=True)
mu=np.mean(y1); #media de y1 - temperatura filtrada
sigma=np.std(y1); #desvio padrão de y1 (temperatura filtrada acima de 15C)
YY=1/(sigma * np.sqrt(2*np.pi))*np.exp( - (bins2 - mu)**2/(2*sigma**2) ) # curva normal (gaussiana)

plt.plot(bins2,YY,linewidth=2, color='r')

plt.ylabel('Frequencia [-]',fontsize=14)
plt.xlabel('Temperatura do ar [°C]',fontsize=14)
#plt.yticks(np.arange(0,1, step=0.05),fontsize=14)
plt.xticks(np.arange(12,31, step=2),fontsize=14)
plt.legend(['Simulado','Observado']);
plt.xlim((12,30))
#plt.ylim((0,1))
#plt.grid()
plt.show()
plt.tight_layout()
```



#Programa para fazer a rosa dos ventos

`#!pip install windrose`

`from windrose import WindroseAxes`

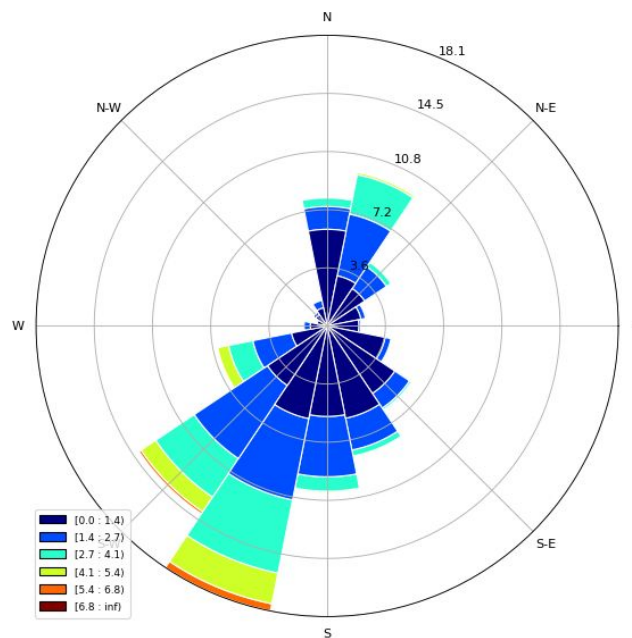
`ws=data[:,9]`

`wd=data[:,10]`

`ax = WindroseAxes.from_ax()`

`ax.bar(wd, ws, normed=True, opening=1, edgecolor='white')`

`ax.set_legend()`



Aula CDM - 05 - 06/04/2020 - Lista 04

#Lista Exercício 04 - Numpy Basic

#Cheat Sheet - Python Numpy

#Por: Yoshi

#Data: Jan/2020

```
#Principais bibliotecas
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from numpy import genfromtxt
!pip install windrose
import meteorologia as mt #(tem que fazer um upload da arquivo no Colab)
```

```
#Criei um array 1D
```

```
x0=np.arange(-3,10,1);
x1=np.arange(0,10,0.01);
x2=np.arange(-3,3,0.1);
```

```
y0=x0      # função linear
y1=np.log(x1); # função ln ( log natural)
y2=np.exp(x2); # função e^x (exponencial)
```

```
plt.plot(x0,y0)
plt.plot(x1,y1)
plt.plot(x2,y2)
```

```
a=np.ones(10,dtype=np.int16)*np.nan # exemplo de criar um array nulo
```

```
### Exemplo 02 - Criar uma função ( converter temperatura em kelvin)
```

```
tp=np.arange(15,30,1);
#tp=np.ones((25,25),dtype=np.int16)*25
```

```
#Definir uma função no python
```

```
tk=tp+273. # Estrutura de função (1)
```

```
def ConverterTK(x): # Estrutura de função (2)
    y=x+273.15
    return y
```

```
TCK=lambda x:x+273.15 # Estrutura de função (3)
```

`import meteorologia as mt` # Estrutura de função (4)

`mt.TK(10)`

Aula de CDM
06/04/2020

①

função: $f(x) = x + 273,15$

↓
variável
de saída

↓
variável
de entrada

(termo dependente) (termo independente)

Estrutura
(Função)

def converterTK(x):
y = x + 273,15
return y

②

Continuação funções...

Função
usando "lambda"

TK = lambda x: x + 273,15

↓
variável
de saída

↓
variável
de entrada

Equação

③

④
Função
usando uma biblioteca
arquivo externo: meteorologia.py
tem a função TK

import meteorologia as mt
mt.TK(x)

↳ biblioteca ↳ função

Fim de aula !

Aula CDM - 06 - 13/04/2020 - Lista 04

```
import numpy as np #biblioteca do matlab
import pandas as pd # biblioteca R
import matplotlib.pyplot as plt # biblioteca de graficos
```

```
tp=25; po=1013; p=920;
TK2= lambda x: x+273
```

```
alt=- (287*TK2(tp)/9.81)*np.log(p/po)
np.round(alt)
```

```
def deltaH(a,b,c):
    d=- (287*TK2(a)/9.81)*np.log(b/c)
    return d
```

Exercício 05 - Calcular a altitude

```
TK2= lambda x: x+273.15
```

Definido a funcao para calcular a espessura da camada

```
def deltaH3(x1,x2,x3,x4):
    yy=-((287*TK2(x1))/x4)*np.log(x2/x3)
    return yy
```

```
altitude=data[:,0] # altutude
```

```
temp=data[:,1] # tempe
```

```
g=data[:,2]
```

```
pressao=data[:,3]*100 #
```

```
alt2=deltaH3(temp,pressao,1013,g) # calculo da altitude pela Hipsométrica
```

```
plt.scatter(pressao,altitude) # gráfico da tabela da atm padrão
```

```
plt.plot(pressao,alt2) # gráfico calculado
```

Aula CDM: Numpy
13/04/2020 Ex 05 (lista 4)

$$\Delta z = -\frac{R_d T}{g} \ln\left(\frac{P}{P_0}\right)$$

$$\text{ex } -\frac{\Delta z \cdot g}{R_d T} = \ln\left(\frac{P}{P_0}\right)$$

$$\frac{P}{P_0} = \exp\left[-\frac{\Delta z \cdot g}{R_d T}\right]$$

$$P_0 = P \times \exp\left[\frac{g \Delta z}{R_d T}\right]$$

densidade do ar

Eq. de estado

$$P = \rho \cdot R_d T$$

$$\rho = \frac{P}{R_d T}$$

Exemplo 05 - calcule a densidade do ar

```
altitude=data[:,0];  
pressao=data[:,3]*10000;  
tp=data[:,1];  
rho=pressao/(287*(tp+273.15))  
plt.scatter(data[:,4],data[:,0])  
plt.plot(rho,altitude)
```

continuação

$R_d, g \rightarrow \text{constantes}$

geopotencial

$T, P, P_0 \rightarrow \text{Variáveis de entrada}$

$$\Delta Z = - \frac{R_d T}{g} \ln\left(\frac{P}{P_0}\right)$$

↓
Variável de Saída

Equação

$R_d = \text{const ar seco}$
 $R_d = 287 \text{ J/kgK}$

$g = 9,81 \text{ m/s}^2$

```
plt.scatter(tp, altitude)
plt.scatter(g, altitude)
```

Aula CDM - 07 - 17/04/2020 - Lista 05

Uso de condicional (IF statement)

```
#Lista Exercício 05 - Numpy Basic
#Cheat Sheet - Python Numpy
#Por: Yoshi
#Data: Jan/2020
```

```
#Principais bibliotecas
```

```
import pandas as pd
```

```
import numpy as np
```

```
from matplotlib import pyplot as plt
```

```
from numpy import genfromtxt
```

```
# endereço do arquivo de dados
```

```
uri='https://raw.githubusercontent.com/sakagamiyoshiaki/CDM/master/dados-vento.csv'
```



```
data = genfromtxt(uri, delimiter=';') # abrir e ler o arquivo
```

```
vel=data[:,0]; # seleciona os dados de velocidade
```

```
teta=data[:,1]; # seleciona os dados de direcao
```

```
#Decomposição do vento - anemometro de copo e biruta
```

```
u=-vel*np.sin(teta*np.pi/180)
```

```
v=-vel*np.cos(teta*np.pi/180)
```

```
plt.plot(u)
```

```
plt.plot(v)
```

```
# Composição do vento do anemometro Sonico
```

```
uu=data[:,2]; # componente u ( direcao leste-oeste)
```

```
vv=data[:,3]; # componente v ( direção norte-sul)
```

```
wind=np.sqrt(uu**2+vv**2)
```

```
plt.plot(wind)
```

```
plt.plot(vel)
```

```
#Composição da direção do vento - Utilizando a estrutura condicional (IF)
```

```
uu[uu==0]=np.nan # excluo valores de uu=0 => uu=NaN (dado invalido)
```

```
vv[vv==0]=np.nan # excluo valores de vv=0 => vv=NaN (dado invalido)
```

```
def comp(x1,x2):
```

```
    y0=np.arctan(x1/x2)*180/np.pi;
```

```
    if (x1>0 and x2>0):          # Q1 - primeiro quadrante
```

```
        y=y0+180
```

```
    return y
```

```
    elif (x1<0 and x2>0):        # Q2 - segundo quadrante
```

```
        y=y0+180
```

```
    return y
```

```
    elif (x1<0 and x2<0):        # Q3 - terceiro quadrante
```

```
        y=y0
```

```
    return y
```

```
elif (x1>0 and x2<0):      # Q4 - quarto quadrante
    y=y0+360
    return y
```

```
composicao=np.vectorize(comp); # vetorizar (calcular celula a celula), senão o IF considera toda a
coluna
```

```
direcao=composicao(uu,vv)
```

```
plt.plot(direcao)
plt.plot(teta)
```

Aula CDM - 08 - 20/04/2020 - Lista 06

Uso de Looping (For , while)

```
# Simbolos de condicional:
# == igual
# != diferente
# > maior
# < menor
# >= maior e igual
# <= menor e igual
# ( x>5 and x<10 ) - intersecção
# ( x<5 or x<10 ) - uniao
```

```
a=np.array([1,4,2,6,3,4,5,6,7,8,8,10,12,5])
b=a.copy()
```

```
for i in range(0,14): # loop
    if (a[i]>5):      # condicional
        b[i]=0
    else:            # senao
        b[i]=1
```

```
print(b)
```

```
vel=data[:,0]; # valores de velocidade
```

```
v2=vel.copy() # fiz uma copia
```

```
for i in range(0,1440): # looping
    if (vel[i]<1 or vel[i]>3 ):      # se velocidade maior que 3
        v2[i]=vel[i]      # entao v2 vai ser nula
    else:      # senao v2 é a vel
        v2[i]=np.nan
```

```
x = np.ones(10)*np.nan
```

```
for i in range(0,10): # looping - faz uma iteracao de 0 a 9
    x[i]=i      # constroi uma sequencia de 0 a 9
```

```
# converter m/s em km/h elemento por elemento
x = np.ones(1440)*np.nan
for i in range(0,1440):
    x[i]=vel[i]*3.6
plt.plot(x)
```

```
x = np.ones(120)*np.nan
i=0
while i < 10:
    print(i)
    i += 1      # contador ( ele soma +1 a cada iteracao)
```

```
x = np.ones(1440)*np.nan
i=0
while i < 1440:
    x[i]=vel[i]*3.6
    i += 1      # contador ( ele soma +1 a cada iteracao)
plt.plot(x)
```

Aula CDM - 09 - 24/04/2020 - Lista 06

```
#Lista Exercício 06 - Looping
#Cheat Sheet - Python Numpy
#Por: Yoshi
#Data: Jan/2020

#Principais bibliotecas
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from numpy import genfromtxt
from matplotlib import cm

uri='https://raw.githubusercontent.com/sakagamiyoshiaki/CDM/master/chuva.dat'

A = genfromtxt(uri, delimiter=',')
#A = genfromtxt('chuva.dat', delimiter=',')

ano=A[:,0]
mes=A[:,1]
dia=A[:,2]
chuva=A[:,3]

chuva1980=chuva.copy()

for i in range(0,11294):
    if ano[i]==1980:
        chuva1980[i]=chuva[i]
    else:
        chuva1980[i]=0

plt.plot(chuva1980)
```

```

chuvaAno=np.ones(31)*np.nan
year=np.arange(1980,2011)

for i in range(0,31):
    chuvaAno[i]=np.nansum(chuva[ano==1980+i])

plt.bar(year,chuvaAno)

chuvaMes=np.ones(12)*np.nan
month=np.arange(1,13)

for i in range(0,12):
    chuvaMes[i]=np.nansum(chuva[mes==i+1])/30

plt.bar(month,chuvaMes);
plt.xticks(np.arange(0,13, step=1));

# anomalia anual

for i in range(0,31):
    chuvaAno[i]=np.nansum(chuva[ano==1980+i])

chuvaAno[chuvaAno==0]=np.nan

chuvaTm=np.nanmean(chuvaAno) # media de 30 de chuva
chuvaTm
anomalia = chuvaAno-chuvaTm

plt.bar(year,100*(anomalia/chuvaTm))

# total de chuva de cada mes e cada ano
chuvaT = np.ones((12,31))

for i in range(0,31):
    for j in range(0,12):
        chuvaT[j][i]=np.nansum(chuva[(ano==1980+i) & (mes==j+1)])/30

```

```
chuvaT[chuvaT==0]=np.nan
```

```
plt.contourf(chuvaT);
```

Aula CDM - 10 - 27/04/2020 - Lista 06 e Revisão/Avaliação

```
#Avaliacao 01 - exercicio 03
```

```
uri='https://origin.cpc.ncep.noaa.gov/products/analysis_monitoring/ensostu  
ff/detrend.nino34.ascii.txt'
```

```
A = genfromtxt(uri)
```

```
dd=np.arange('1950-01', '2020-04', dtype='datetime64[M]')
```

```
plt.plot(dd,A[1:,4])
```

```
plt.grid()
```

Aula CDM - 11 - 04/05/2020 - Revisão da Avaliação

Bom Dia pessoal !

Apesar da suspensão do calendário, tem-se a possibilidade de continuar as atividades remotamente entre professor e aluno caso ambos queiram.

Da minha parte, estou à disposição para a continuidade das atividades online como a gente tem feito. Isso poderá ser validado depois.

Vou ficar até as 8h por aqui. Caso não tenha presença de nenhum aluno, estarei finalizando a transmissão e ficarei à disposição por email.

Att,
Prof. Yoshi

Aula CDM - 11 - 08/05/2020 - Lista 07

Bom Dia, vamos começar a aula às 9:45h

```
#Principais bibliotecas
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from numpy import genfromtxt
from matplotlib import cm

#A1 = np.loadtxt('CR1000_saída1min.dat',
delimiter=',',skiprows=1,dtype=float)
#A = genfromtxt(uri)

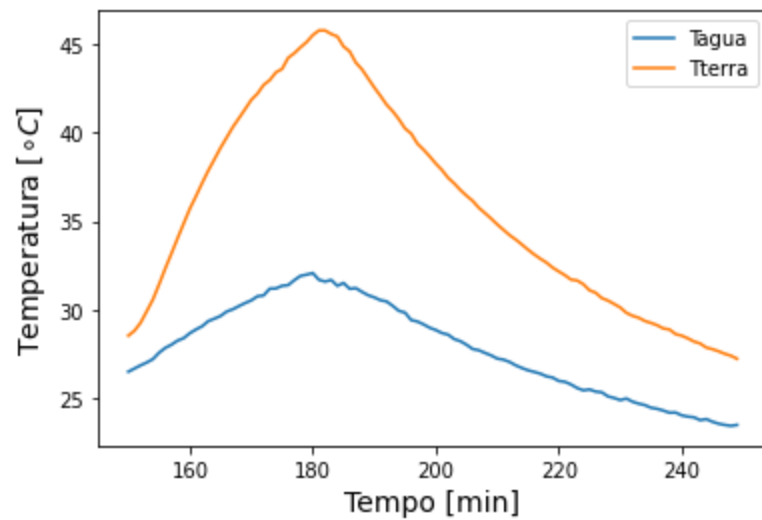
# 'df' é meu dataframe

df = pd.read_csv('CR1000_saída1min.dat',header=1,skiprows=[2,3]);
df[df=='NAN']=np.nan
df

plt.plot(np.arange(0,100, step=1),df['Tagua_Avg'][150:250].astype(float));
plt.plot(np.arange(0,100,
step=1),df['Tterra_Avg'][150:250].astype(float));
plt.legend(['Tagua', 'Tterra']);
plt.title('Gráfico de Experimento - Terra-agua',fontsize=14);
plt.xlabel('Tempo [min]',fontsize=14)
plt.ylabel('Temperatura [ $^{\circ}$  C]',fontsize=14)
plt.xticks(np.arange(0,110, step=10),fontsize=14);
plt.yticks(np.arange(20,50, step=5),fontsize=14);

plt.grid()
```

Resolução Avaliação 01 - exercício 01



Aula CDM - 12 - 13/05/2020 - Lista 07

Bom Dia,

Hoje a aula será sobre como abrir arquivo de dados em diferentes formatos

```
#Lista Exercício 07 - Import data
```

```
#Principais bibliotecas
```

```
import pandas as pd
```

```
import numpy as np
```

```
from matplotlib import pyplot as plt
```

```
## Exemplo 01 - Importar arquivo texto sem cabeçalho
```

```
uri='https://raw.githubusercontent.com/sakagamiyoshiaki/CDM/master/saida2010.txt'
```

```
A1 = np.loadtxt(uri, delimiter=',', skiprows=1, dtype=float)
```

```
plt.plot(A1[:,6])
```

```
df1 = pd.read_csv(uri, header=0);
```



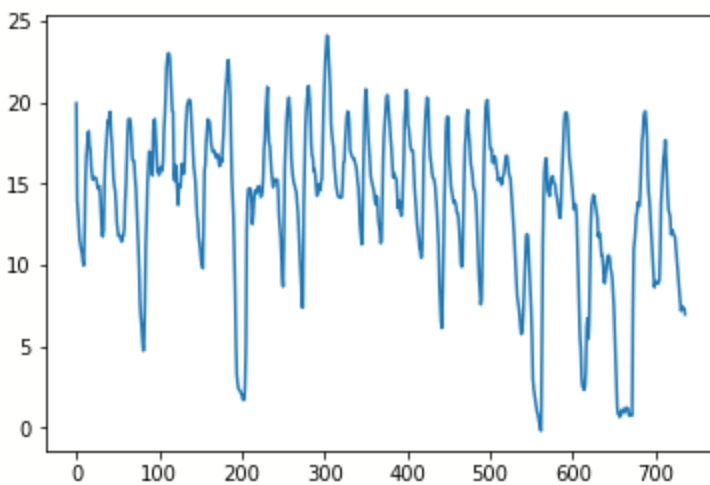
```
## Exemplo 02 Importar arquivo texto com cabeçalho e timestamp
```

```
uri='https://raw.githubusercontent.com/sakagamiyoshiaki/CDM/master/IFSC_IFSC60.dat'
```

```
A2 = np.loadtxt(uri, delimiter=',', skiprows=5, usecols=range(1,12))
```

```
df2 = pd.read_csv(uri, header=1, skiprows=[2,3]);
```

```
plt.plot(df2['Temp_ar_Max'])
```



```
## Exemplo 03 Importar dados Merra (netcdf)
```

```
#wget --load-cookies ~/.urs_cookies --save-cookies ~/.urs_cookies  
--auth-no-challenge=on --keep-session-cookies --content-disposition -i  
url.txt
```

```
#https://disc.gsfc.nasa.gov/datasets/M2T1NXSLV_5.12.4/summary?keywords=%22  
MERRA-2%22
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
#!pip install netcdf4
```

```
import xarray as xr
```

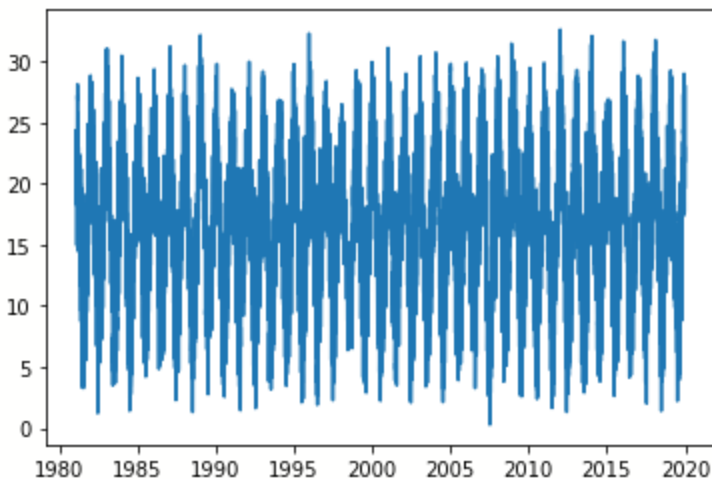
```

import netCDF4

#nc = xr.open_dataset('AirTEMP_BuenosAires.nc');
nc = netCDF4.Dataset('AirTEMP_BuenosAires.nc');
#nc =
netCDF4.Dataset('https://github.com/sakagamiyoshiaki/CDM/blob/master/AirTE
MP_BuenosAires.nc?raw=true');

h = nc.variables['T2M']
times = nc.variables['time']
jd = np.array(netCDF4.num2date(times[:],times.units),
dtype='datetime64[s]')
hs = pd.Series(h[:,0,0],index=jd)
plt.plot(hs)

```



```

#%% Exercio extra 07

uri='https://github.com/sakagamiyoshiaki/CDM/blob/master/Temperatura-Media
-Compensada_NCB_1961-1990.xls?raw=true'
#uri='Temperatura-Media-Compensada_NCB_1961-1990.xls'

dfx = pd.read_excel(uri,header=3);

```

Aula CDM - 13 - 15/05/2020 - Lista 08

Aula de introdução ao Pandas

Lista Exercício 08 - Introdução ao Pandas

#Cheat Sheet - Pandas Basics

#Por: Yoshi

#Data: Maio/2020

#Principais bibliotecas

```
import pandas as pd
```

```
import numpy as np
```

```
from matplotlib import pyplot as plt
```

```
#!pip install mpld3
```

```
#import mpld3
```

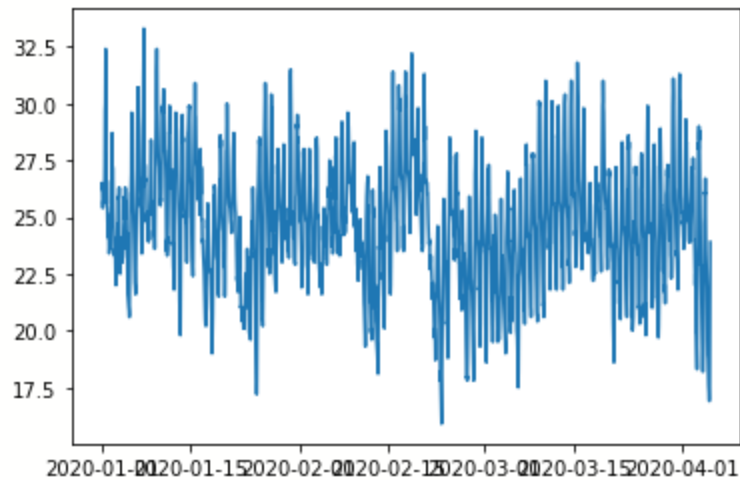
```
#mpld3.enable_notebook()
```

%% Exemplo 01 - abrir um arquivo e transformar em Dataframe

```
uri='https://raw.githubusercontent.com/sakagamiyoshiaki/CDM/master/InmetFl  
oripa2020.txt'
```

```
df = pd.read_csv(uri,header=0); # o comando pd.read_ já tranforma em  
dataframe
```

```
plt.plot(df['temp_inst'])
```



```
df
df.dtypes
df.T
df.sort_index(axis=0, ascending=False) # Ordenar o índice de forma
ascendente

df['timestamp'] = pd.to_datetime(df['data'].astype(str) + ' ' +
df['hora'].astype(str), format='%d/%m/%Y %H')
df.index=df['timestamp']

plt.plot(df['2020-02-10 00:00:00':'2020-02-15 00:00:00']['temp_inst'])

plt.plot(df['2020-02-10 00:00:00':'2020-02-15
00:00:00']['temp_inst','temp_max','temp_min'])

df2=df.iloc[0:100,[3,6,9]] # seleção por posicionamento - coluna
especifica

df2.to_csv('saida.csv') # salva os dados em um arquivo csv
```

Bom dia !

```
#Principais bibliotecas
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt

#!pip install mpld3
#import mpld3
#mpld3.enable_notebook()

uri='https://raw.githubusercontent.com/sakagamiyoshiaki/CDM/master/InmetFloripa2020.txt'
df = pd.read_csv(uri,header=0); # o comando pd.read_ já transforma em
dataframe
df['timestamp'] = pd.to_datetime(df['data'].astype(str) + ' ' +
df['hora'].astype(str), format='%d/%m/%Y %H')
df.index=df['timestamp']
df=df.sort_index(axis=0, ascending=True) # Ordenar o índice de forma
ascendente

#df['precipitacao'].hist()
df.hist('temp_inst')

df['tk']=df['temp_inst']+273

f=lambda x:x+273.15 # função definda (converte temperatura Celsius em
kelvin)
df['temp_inst']=df['temp_inst'].apply(f) # aplica uma função em um
Dataframe

df2.drop(df2.index[1:4])
```

```

df3=df.iloc[0:20,1:7]
df3

df3['temp_inst'][df3['temp_inst']>29]=np.nan; excluindo valores acima 29
graus e substituindo por NaN
df3

df3['temp_inst'][(df3['temp_inst']>25) & (df3['temp_inst']<30)]=np.nan;
#excluindo valores acima de 25 e abaixo de 29 graus e substituindo por NaN
df3

df3['temp_inst'][(df3['temp_inst']<26) | (df3['temp_inst']>30)]=np.nan;
#excluindo valores acima de 25 e abaixo de 29 graus e substituindo por NaN
df3

df8=df.iloc[0:20,1:7]

df9=df8.copy() # copia do dataframe (não modufica quando df altera)
df10=df8      # clone do dataframe (modifica junto com o df)

df8.loc[(df8['temp_inst']<25.5) |
(df8['temp_inst']>30),'temp_inst']=np.nan
df9

```

Aula CDM - 15 - 22/05/2020 - Lista 10

Consistência de dados

```

#criar uma serie artificial completa
dt=pd.date_range(start='2010-03-01 00:00:00',end='2010-03-31
23:55:00',freq='05min'); # criar uma série temporal inteira
idx=pd.DatetimeIndex(dt) # serie temporal e transforma em index
df2 = df2.reindex(idx) # reindexar

```

```
# separa apenas os dados não duplicados
df6=df5.loc[~df5.index.duplicated(keep='last')]

# remover os dados manualmente
df6=df2.copy()
plt.plot(df6.tp)
df6.tp['2010-03-12 07:00:00':'2010-03-14 14:00:00']=np.nan

# preencher a serie
df7=df6.fillna(0) # preencher com zero todo o dataframe
df8=df6.fillna(method='bfill') # preencher com os dados anterior
df10=df6.interpolate() # preencher fazendo interpolação
```

Aula CDM - 16 - 25/05/2020 - Lista 11

Revisão do biblioteca Pandas - Dataframe

Lista 12 - Juntar os dataframes

- Join, merge e concat
- Concat - Multi-Index
-