

```

#importação de bibliotecas

import pandas as pd #para o dataframe

import matplotlib.pyplot as plt #para os gráficos

import mpncyberpunk #para o tema dos gráficos, está marcado como não utilizado pois não estou
usando suas funções


#função para ordenar as listas em ordem crescente

def listaSort(lista, listaCount):

    for i in range(len(lista)): #dois loops para que cada elemento da lista sofra a comparação

        for j in range(len(lista)): #com todos os outros elementos

            if int(lista[int(i)]) < int(lista[int(j)]): #se o primeiro elemento for maior

                lista[int(i)], lista[int(j)] = lista[int(j)], lista[int(i)] #eles trocam de valor

                listaCount[int(i)], listaCount[int(j)] = listaCount[int(j)], listaCount[int(i)] #e trocam o valor da
quantidade

        return (lista, listaCount) #retorna a lista com o campo da data e a lista de quantidades


filename = r"exercicioedjalma.csv" #Lê o arquivo CSV

df = pd.read_csv(filename) #transforma-o em DataFrame usando o pandas

dicionario = df.to_dict() #transforma o DataFrame em um dicionário


# Adiciona todas as datas em uma lista

datas = [] #declaração de lista

for i in dicionario.keys(): #acessa os elementos das chaves

    datas.append(i) #adiciona na lista

    for j in dicionario[i].values(): #acessa o resto dos valores de cada chave

        datas.append(j) #e os adiciona na lista


# Divide e armazena as datas em suas respectivas listas(dia, mês e ano)

anos = [] #declaração de lista

```

```

meses = [] #declaração de lista
dias = [] #declaração de lista
for data in datas: #acessa os elementos da lista
    data = data.split('/') #separa o seu conteúdo usando o caractere da '/'
    if data[0][0] == '0': #se o primeiro dígito dos dias for zero
        data[0] = data[0][1:] #o valor desse primeiro elemento do split é apenas a casa das unidades
    if data[1][0] == '0': #se o primeiro dígito dos meses for zero
        data[1] = data[1][1:] #o valor desse segundo elemento do split é apenas a casa das unidades
    dias.append(data[0]) #adiciona o primeiro elemento do split na lista dias
    meses.append(data[1]) #adiciona o segundo elemento do split na lista meses
    anos.append(data[2]) #adiciona o terceiro elemento do split na lista anos

#separa a lista dias em 2 listas, para ordenar por dia e contar quantas vezes se repete
dia = [] #declaração de lista
diaCount = [] #declaração de lista
for i in dias: #acessa os elementos da lista dias
    if i not in dia: #se o elemento não existe na nova lista
        dia.append(i) #adiciona-o na lista
        diaCount.append(1) #e adiciona um valor na lista de quantidades
        continue #vira ao início do loop for
    diaCount[dia.index(i)] += 1 #caso já exista adiciona um valor no índice de quantidades no mesmo
    índice da lista dia
dia, diaCount = listaSort(dia, diaCount) #as duas listas recebem o retorno da função de ordenação

#separa a lista meses em 2 listas, para ordenar por mês e contar quantas vezes se repete
mes = [] #declaração de lista
mesCount = [] #declaração de lista
for i in meses: #acessa os elementos da lista meses
    if i not in mes: #se o elemento não existe na nova lista

```

```
mes.append(i) #adiciona-o na lista

mesCount.append(1) #e adiciona um valor na lista de quantidades

continue #vola ao inicio do loop for

mesCount[mes.index(i)] += 1 #caso já exista adiciona um valor no indice de quantidades no mesmo
indice da lista mes

mes, mesCount = listaSort(mes, mesCount) #as duas listas recebem o retorno da função de ordenação
```

```
#separa a lista anos em 2 listas, para ordenar por ano e contar quantas vezes se repete
```

```
ano = [] #declaração de lista
```

```
anoCount = [] #declaração de lista
```

```
for i in anos: #acessa os elementos da lista anos
```

```
    if i not in ano: #se o elemento não existe na lista nova
```

```
        ano.append(i) #adiciona-o na lista
```

```
        anoCount.append(1) #e adiciona um valor na lista de quantidades
```

```
        continue #volta ao inicio do loop for
```

```
    anoCount[ano.index(i)] += 1 #caso já exista adiciona um valor no indice de quantidades no mesmo
    indice da lista ano
```

```
ano, anoCount = listaSort(ano, anoCount) #as duas listas recebem o retorno da função de ordenação
```

```
#cria o gráfico dos dias
```

```
with plt.style.context('cyberpunk'): #tema
```

```
    plt.figure(figsize=(15, 5)) #tamanho
```

```
    plt.bar(dia, diaCount) #define a comparação dos dias com sua quantidade
```

```
    plt.title('Repetição dos dias') #titulo
```

```
    plt.xlabel('Dias') #nome da label do eixo X
```

```
    plt.ylabel('Quantidade') #nome da label do eixo Y
```

```
    plt.show() #mostra o gráfico
```

#cria o gráfico dos meses

with plt.style.context('cyberpunk'): #tema

plt.figure(figsize=(10, 5)) #tamanho

plt.bar(mes, mesCount) #define a comparação dos meses com sua quantidade

plt.title('Repetição dos meses') #titulo

plt.xlabel('Meses') #nome da label do eixo X

plt.ylabel('Quantidade') #nome da label do eixo Y

plt.show() #mostra o gráfico

#cria o gráfico dos anos

with plt.style.context('cyberpunk'): #tema

plt.figure(figsize=(25, 5)) #tamanho

plt.bar(ano, anoCount) #define a comparação dos anos com sua quantidade

plt.title('Repetição dos anos') #titulo

plt.xlabel('Anos') #nome da label do eixo X

plt.ylabel('Quantidade') #nome da label do eixo Y

plt.show() #mostra o gráfico