

NumPy e Matplotlib

Paulo Almeida
André Grégio



JUSTIÇA 4.0: INOVAÇÃO E EFETIVIDADE NA REALIZAÇÃO DA JUSTIÇA PARA TODOS
PROJETO DE EXECUÇÃO NACIONAL BRA/20/015

1.

NumPy - Operações Lógicas



Where

A função `numpy.where(condition, resultado_if, resultado_else)` retorna um array com os elementos que satisfazem determinada condição

`condition` é a condição

`resultado_if` são os elementos selecionados quando a condição é satisfeita

`Resultado_else` são os elementos selecionados quando a condição não é satisfeita

Where

```
import numpy as np
```

```
valores = np.random.randint(0, 10, (5, 4))
```

```
print(valores)
```

```
resultado = np.where(valores < 5, valores, 10*valores)
```

```
print(resultado)
```

Where

É possível combinar o where com conectivos lógicos

Coloque cada condição entre parêntesis ()

& representa um **e** lógico

| representa um **ou** lógico

Where

```
import numpy as np
```

```
valores = np.random.randint(0, 10, (5, 4))
```

```
print(valores)
```

```
resultado = np.where((valores >= 3) & (valores <= 7), valores, 10*valores)
```

```
print(resultado)
```

Aplicando condições diretamente

É possível aplicar condições diretamente no operador []

```
import numpy as np

valores = np.random.randint(0, 10, (5, 4))
print(valores)
valores[(valores < 3) | (valores == 9)] = 33
print(valores)
```

Matrizes de máscaras

É possível ainda criar matrizes de “máscaras”, que podem ser usadas para algum pós-processamento

```
import numpy as np
```

```
valores = np.random.randint(0, 10, (5, 4))
```

```
mascara = valores < 5
```

```
print(mascara)
```

```
mascara = mascara.astype(int)
```

```
print(mascara)
```


Veja mais

Veja mais sobre where e operadores lógicos em
numpy.org/doc/stable/reference/generated/numpy.where.html
numpy.org/doc/stable/reference/routines.logic.html
www.geeksforgeeks.org/numpy-array-logical-operations

Próximo Tópico

Ordenação e Estatística Básica



2.

NumPy - Ordenação e Estatística Básica



Ordenando

Para ordenar um array NumPy, basta chamar a função `sort()`

```
import numpy as np

valores = np.random.randint(0, 6, (5, 5))
print(valores)
valores.sort(axis=0)
print(valores)
```

Média

A média dos elementos pode ser computada através da função `mean()`

De maneira similar, é possível calcular

- Média ponderada: `average()`

- Desvio padrão: `std()`

- Variância: `var()`

- ...

Veja mais em

numpy.org/doc/stable/reference/routines.statistics.html

Máximo e Mínimo

As funções `max()` e `min()` podem ser usadas para se encontrar o maior e o menor valor em um Array NumPy

Argmax e Argmin

É possível encontrar o índice do maior e menor valor no array utilizando `argmax()` e `argmin()`

Caso exista mais de uma cópia do maior/menor valor, o índice da primeira ocorrência é retornado

Um exemplo

Considere o arquivo TRE-BA.csv disponibilizado

A coluna 62 (mov_count) contém a contagem de movimentos de cada processo

Vamos calcular:

- A quantidade média de movimentos

- O desvio padrão

- O maior e menor número de movimentos

- O índice dos (primeiros) processos com o maior e menor número de movimentos

Um exemplo

```
import numpy as np
```

```
data = np.genfromtxt('TRE-BA.csv', delimiter=';',  
skip_header = 1, usecols = (62), dtype=(np.intc))
```

```
print(data.mean())  
print(data.max())  
print(data.min())  
print(data.argmax())  
print(data.argmin())
```

Próximo tópico

Matplotlib



3. Matplotlib



Matplotlib

Pacote para criar gráficos
Principalmente gráficos bidimensionais

Projeto iniciado em 2002 por John Hunter

Muitos dos comandos e lógicas envolvidas nos gráficos são compatíveis com ferramentas como MATLAB e Octave

Um exemplo

```
import matplotlib.pyplot as plt
import numpy as np
```

```
x = np.arange(0,10,1)
y = x**2 # $f(x) = x^2$ 
```

```
plt.plot(x,y)
plt.show()
```

Um exemplo

```
import matplotlib.pyplot as plt
import numpy as np
```

```
x = np.arange(0,10,1)
y = x**2 #f(x) = x^2
```

```
plt.plot(x,y)
plt.show()
```

A função `matplotlib.pyplot.plot(x,y)` recebe dois vetores contendo os valores de `x`, e os respectivos valores de `y`.

Existem variações para a chamada dessa função. Veja em matplotlib.org/3.5.1/api/_as_gen/matplotlib.pyplot.plot.html

Um exemplo

```
import matplotlib.pyplot as plt  
import numpy as np
```

```
x = np.arange(0,10,1)  
y = x**2 #f(x) = x^2
```

```
plt.plot(x,y)  
plt.show()
```

A função show() exibe o gráfico

Múltiplas Séries

Uma das formas de se adicionar múltiplas séries, é chamando `matplotlib.pyplot.plot(x,y)` múltiplas vezes

```
import matplotlib.pyplot as plt
import numpy as np
```

```
x = np.arange(0,10,1)
y2 = x**2 #f(x) = x^2
y3 = x**3 #f(x) = x^3
```

```
plt.plot(x,y2)
plt.plot(x,y3)
plt.show()
```


Estilizando

É possível trocar as cores e estilo das linhas

Para uma listagem completa dos estilos de cores e linhas
matplotlib.org/3.5.1/tutorials/colors/colors.html

matplotlib.org/2.1.2/api/_as_gen/matplotlib.pyplot.plot.html

Estilizando

```
import matplotlib.pyplot as plt
import numpy as np
```

```
x = np.arange(0,10,1)
y2 = x**2 #f(x) = x^2
y3 = x**3 #f(x) = x^3
```

```
plt.plot(x,y2, "b-.")
plt.plot(x,y3, "k:")
plt.show()
```

b- para azul (blue) com linha tracejada
k: para preto com linha pontilhada

Legendas

É possível adicionar legendas

Nas séries

Adicione a série com `plt.plot(..., label = "legenda")`

Antes de chamar `show()`, configure a posição com
`plt.legend(loc="posição")`

Veja as posições possíveis em

matplotlib.org/3.5.0/api/_as_gen/matplotlib.pyplot.legend.html

Legendas

Legendas nos eixos x e y podem se adicionadas através de

```
plt.ylabel("Nome eixo y")  
plt.xlabel("Nome eixo x")
```

Exemplo

```
import matplotlib.pyplot as plt
import numpy as np
```

```
x = np.arange(0,10,1)
y2 = x**2 #f(x) = x^2
y3 = x**3 #f(x) = x^3
```

```
plt.plot(x,y2, "b-.", label = "x^2")
plt.plot(x,y3, "k:", label = "x^3")
plt.legend(loc="upper left")
plt.ylabel("Eixo Y")
plt.xlabel("Eixo X")
plt.show()
```

Próximo Tópico

Múltiplos plots e tipos de gráfico



4.

Múltiplos plots e tipos de gráfico



Dados

Considere os arquivos PETR3.SA.csv e PRIO3.SA.csv que contêm os valores das ações da Petrobras e PetroRio entre os dias 04/01/2021 e 15/03/2022

Os dados foram retirados de finance.yahoo.com

Um exemplo

Vamos considerar o preço de fechamento das ações

Um exemplo

```
import matplotlib.pyplot as plt
from datetime import datetime
import numpy as np
```

```
dataPetr = np.genfromtxt('PETR3.SA.csv', delimiter=',', skip_header = 1, usecols = (0,5),
converters={0: lambda x: datetime.strptime(x.decode('ascii'), "%Y-%m-%d")})
dias = np.empty((dataPetr.shape[0]), dtype=datetime)
valores = np.empty((dataPetr.shape[0]), dtype=np.float64)
```

```
i = 0
for d in dataPetr:
    dias[i] = d[0]
    valores[i] = d[1]
    i += 1
```

```
plt.plot(dias, valores, "b-.", label = "PETR3")
plt.legend(loc="upper left")
plt.ylabel("Preço (R$)")
plt.xlabel("Data")
plt.show()
```

Adicionando PRI03

```
import matplotlib.pyplot as plt
from datetime import datetime
import numpy as np
```

```
data_petr = np.genfromtxt('PETR3.SA.csv', delimiter=',', skip_header = 1, usecols = (0,5), converters={0: lambda
x: datetime.strptime(x.decode('ascii'), "%Y-%m-%d")})
dias = np.empty((data_petr.shape[0]), dtype=datetime)
valores_petr = np.empty((data_petr.shape[0]), dtype=np.float64)
```

```
data_prio = np.genfromtxt('PRI03.SA.csv', delimiter=',', skip_header = 1, usecols = (0,5), converters={0: lambda
x: datetime.strptime(x.decode('ascii'), "%Y-%m-%d")})
valores_prio = np.empty((data_prio.shape[0]), dtype=np.float64)
```

```
i = 0
while(i < data_petr.shape[0]):
    dias[i] = data_petr[i][0]
    valores_petr[i] = data_petr[i][1]
    valores_prio[i] = data_prio[i][1]
    i += 1

plt.plot(dias, valores_petr, "b-.", label = "PETR3")
plt.plot(dias, valores_prio, "k:", label = "PRI03")
plt.legend(loc="upper left")
plt.ylabel("Preço (R$)")
plt.xlabel("Data")
plt.show()
```

Figure

Manipulações mais complexas podem exigir o uso de uma variável do tipo Figure

Retornada pela função `plt.figure()`

add_subplot

É possível adicionar sub gráficos (subplots) em uma Figure através de `add_subplot(linhas, colunas, idx)`

Considerando que o gráfico completo é um grid (matriz) de $N \times M$

Linhas: indica o número de linhas no grid

Colunas: indica o número de colunas no grid

Idx: índice do subgráfico no gráfico

Adicionando em subgráficos

Funções como `plt.plot()`, `plt.legend()`, `plt.xlabel()` vão criar gráficos e alterar o último subplot adicionado

Exemplo

```
#...
```

```
fig = plt.figure()
fig.add_subplot(2, 1, 1, title= "Gráfico da Petrobras")
plt.plot(dias,valores_petr, "b-.", label = "PETR3")
plt.legend(loc="upper left")
plt.ylabel("Preço (R$)")
plt.xlabel("Data")
fig.add_subplot(2, 1, 2, title= "Gráfico da PetroRio")
plt.plot(dias,valores_prio, "k:", label = "PRI03")
plt.legend(loc="upper left")
plt.ylabel("Preço (R$)")
plt.xlabel("Data")
plt.show()
```

Tipos de Gráficos

Existem diversos outros tipos de gráficos

- Gráficos de barra

- Scatter

- Pizza

- Boxplot

- ...

Veja uma lista aqui matplotlib.org/3.5.1/plot_types/index.html

Exemplo

Mostrar como gráficos de barras os preços das ações referentes aos 30 primeiros dias das séries

Exemplo

Primeiros 30 dias

```
#...
```

```
i = 0
while(i < 30):
    dias[i] = data_petr[i][0]
    valores_petr[i] = data_petr[i][1]
    valores_prio[i] = data_prio[i][1]
    i += 1
```

```
fig = plt.figure()
fig.add_subplot(2, 1, 1)
plt.bar(dias, valores_petr, label = "PETR3")
plt.legend(loc="upper left")
plt.ylabel("Preço (R$)")
plt.xlabel("Data")
fig.add_subplot(2, 1, 2)
plt.bar(dias, valores_prio, label = "PRI03")
plt.legend(loc="upper left")
plt.ylabel("Preço (R$)")
plt.xlabel("Data")
plt.show()
```

Salvando

É possível salvar um gráfico através da função `plt.savefig("Caminho de destino")`

A função infere automaticamente o tipo do arquivo pela extensão do caminho. Por exemplo, para salvar em pdf:

`plt.savefig("imagem.pdf")`

Veja mais em

matplotlib.org/3.5.1/api/_as_gen/matplotlib.pyplot.savefig.html

Próximo Tópico: Histogramas



5. Histogramas



Histograma

Um histograma geralmente é representado através de um gráfico de barras

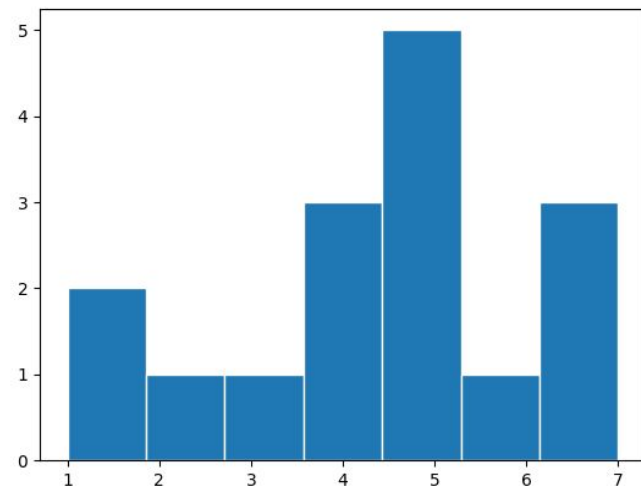
Gera uma aproximação visual da distribuição dos dados

Exemplo

Considerando o vetor a seguir

2	1	1	4	5	5	3	4	4	5	5	5	6	7	7	7
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

O histograma ao lado mostra graficamente a quantidade de vezes que cada número aparece no vetor.



Matplotlib

Para gerar um histograma via Matplotlib, basta chamar a função hist()

```
import matplotlib.pyplot as plt  
import numpy as np
```

```
x = np.array([2,1,1,4,5,5,3,4,4,5,5,5,6,7,7,7])
```

```
plt.hist(x, bins=7, edgecolor="white")  
plt.show()
```


Bins

Muitas vezes, desejamos agrupar os dados em bins

Por exemplo, desejamos agrupar todos os valores entre 1 e 3 em uma única barra, os valores entre 4 e 6 em outra, ...

Cada agrupamento é **um bin**

Bins também são conhecidos como *buckets* (baldes)

Exemplo

Considere o arquivo ExemploNotas.csv disponibilizado
O arquivo contém o nome e a nota de diversos alunos

Desejamos criar um histograma mostrando o número de
alunos com nota entre 0 e 10, 11 e 20, 21 e 30, ...

Exemplo

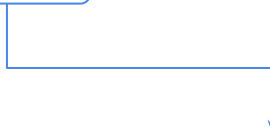
```
import matplotlib.pyplot as plt
import numpy as np

data = np.genfromtxt('ExemploNotas.csv',
delimiter=';', skip_header = 1, usecols = (1),
dtype=np.int32)
plt.hist(data, bins=10, edgecolor="white")
plt.show()
```

Exemplo

```
import matplotlib.pyplot as plt
import numpy as np
```

```
data = np.genfromtxt('ExemploNotas.csv',
delimiter=';', skip_header = 1, usecols = (1),
dtype=np.int32)
plt.hist(data, bins=10, edgecolor="white")
plt.show()
```



O matplotlib vai automaticamente definir o tamanho de cada bin, de forma que todos os bins tenham o mesmo tamanho, e sejam gerados 10 bins.

Bins

É possível ainda especificar um tamanho para cada bin, e o tamanho dos bins não precisa ser o mesmo para todos.

```
import matplotlib.pyplot as plt
import numpy as np

data = np.genfromtxt('ExemploNotas.csv', delimiter=';',
skip_header = 1, usecols = (1), dtype=np.int32)
plt.hist(data, bins=(0,25,50,70,80,90,100),edgecolor="white")
plt.show()
```

Bins

É possível ainda especificar um tamanho para cada bin, e o tamanho dos bins não precisa ser o mesmo para todos.

```
import matplotlib.pyplot as plt
import numpy as np

data = np.genfromtxt('ExemploNotas.csv', delimiter=';',
skip_header = 1, usecols = (1), dtype=np.int32)
plt.hist(data, bins=(0,25,50,70,80,90,100), edgecolor="white")
plt.show()
```

Um bin compreende dados entre 0 e 25. Outro entre 25 e 50, ...

6.

Teste seus conhecimentos



Teste seus conhecimentos

1. Replique tudo que foi ensinado durante as aulas no seu computador para fixar os conhecimentos.
2. Faça um programa que lê os dados dos arquivos do CNJ, e processa a coluna *mov_count*. O programa deve exibir um histograma mostrando quantos processos possuem entre 0 e 25 movimentos, outro bin para processos com movimentos entre 25 e 50 movimentos, ... (cada bin tem tamanho 25).



Obrigado!

Bons Estudos!!!