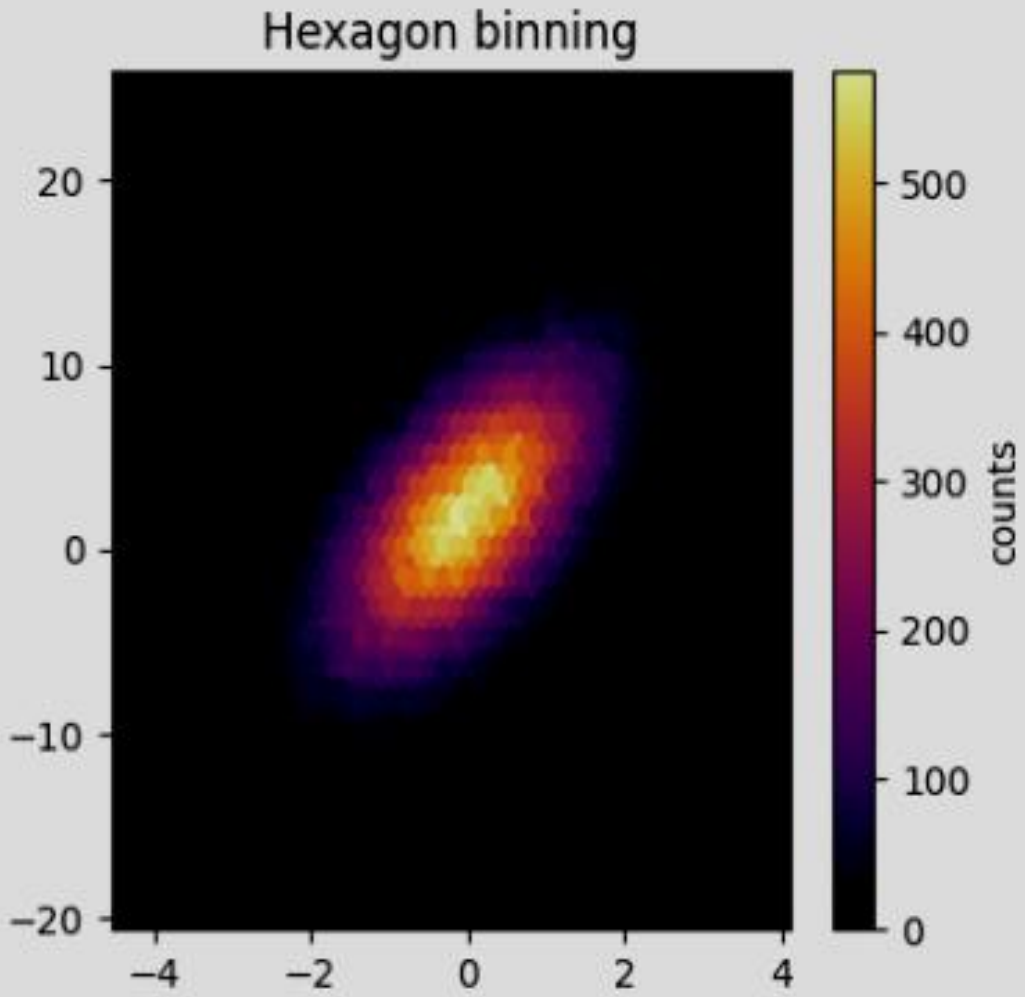
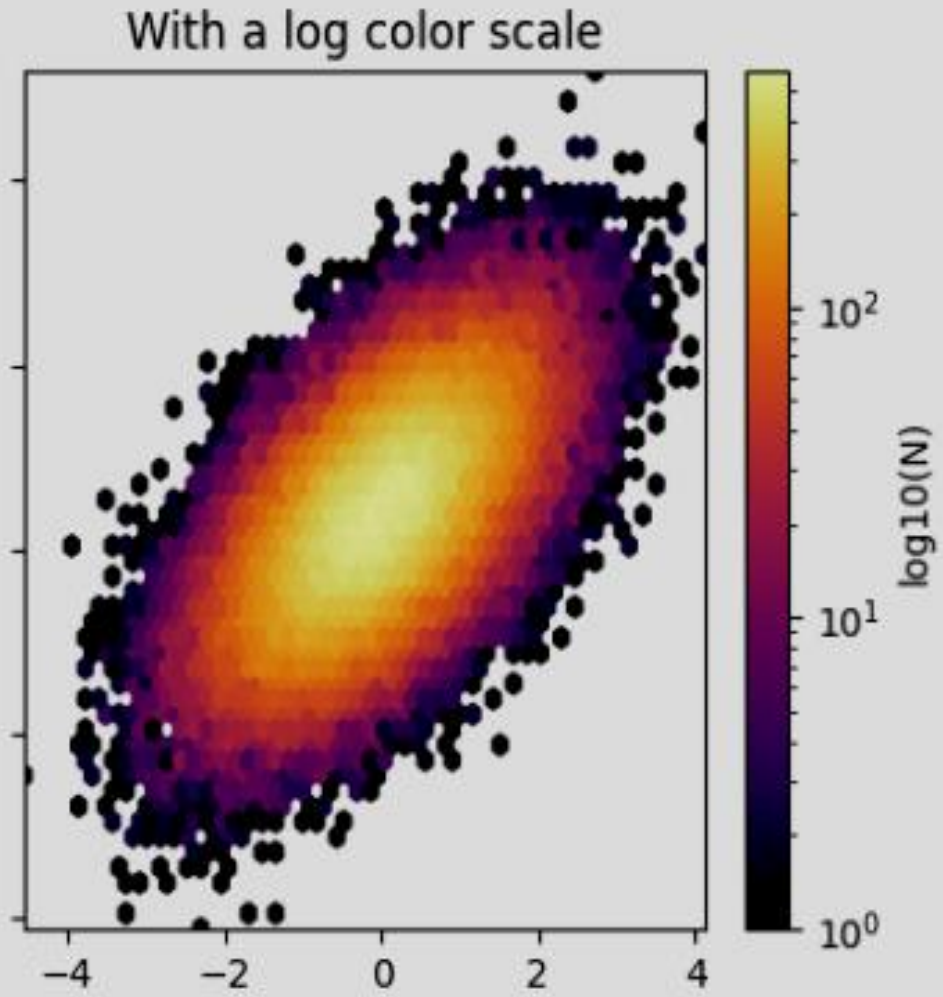
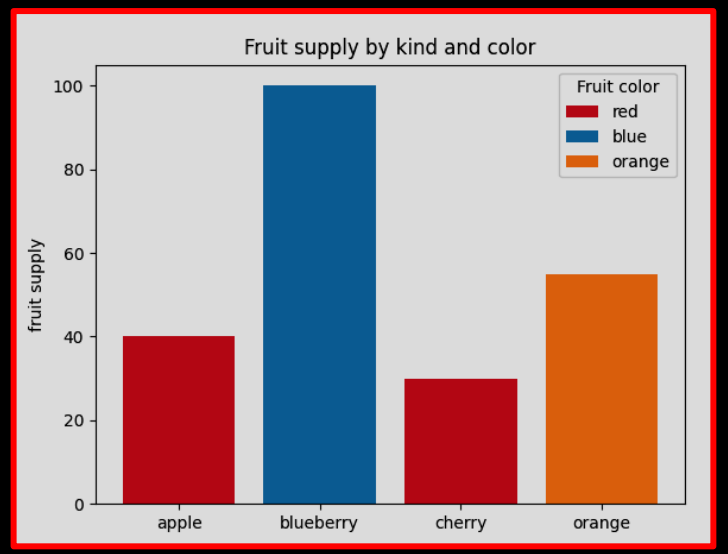
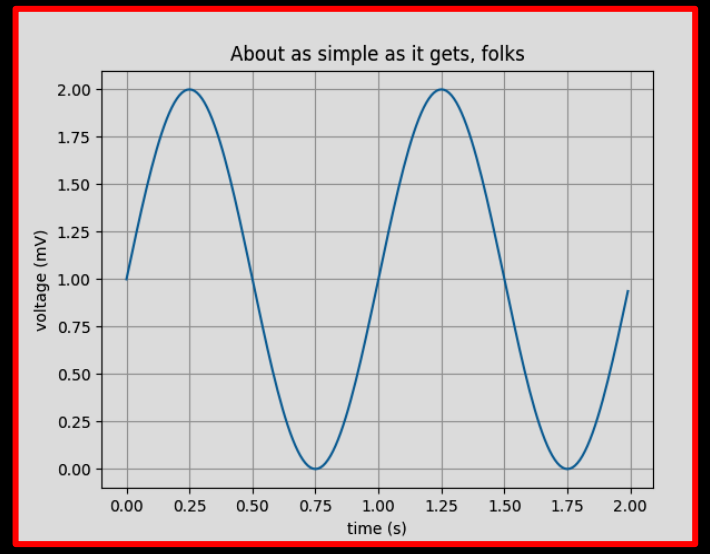


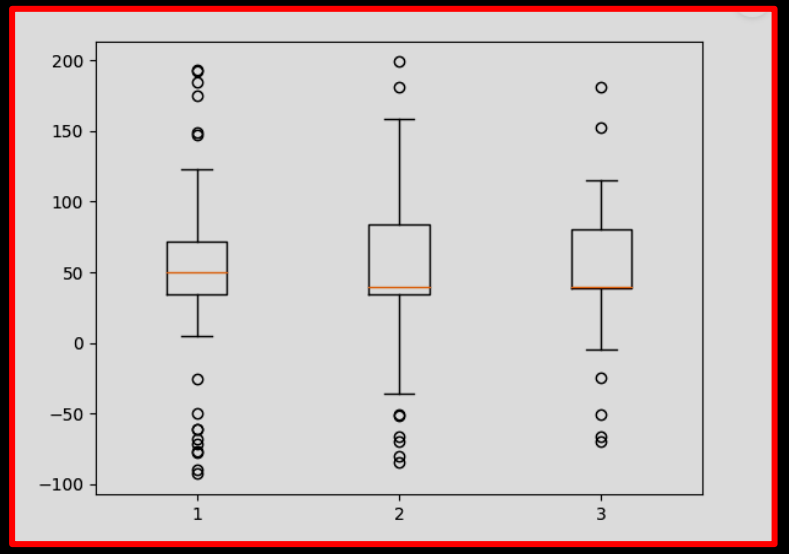


Pablo Ricardo de abreu
10/2022

O que é Biblioteca Matplotlib?



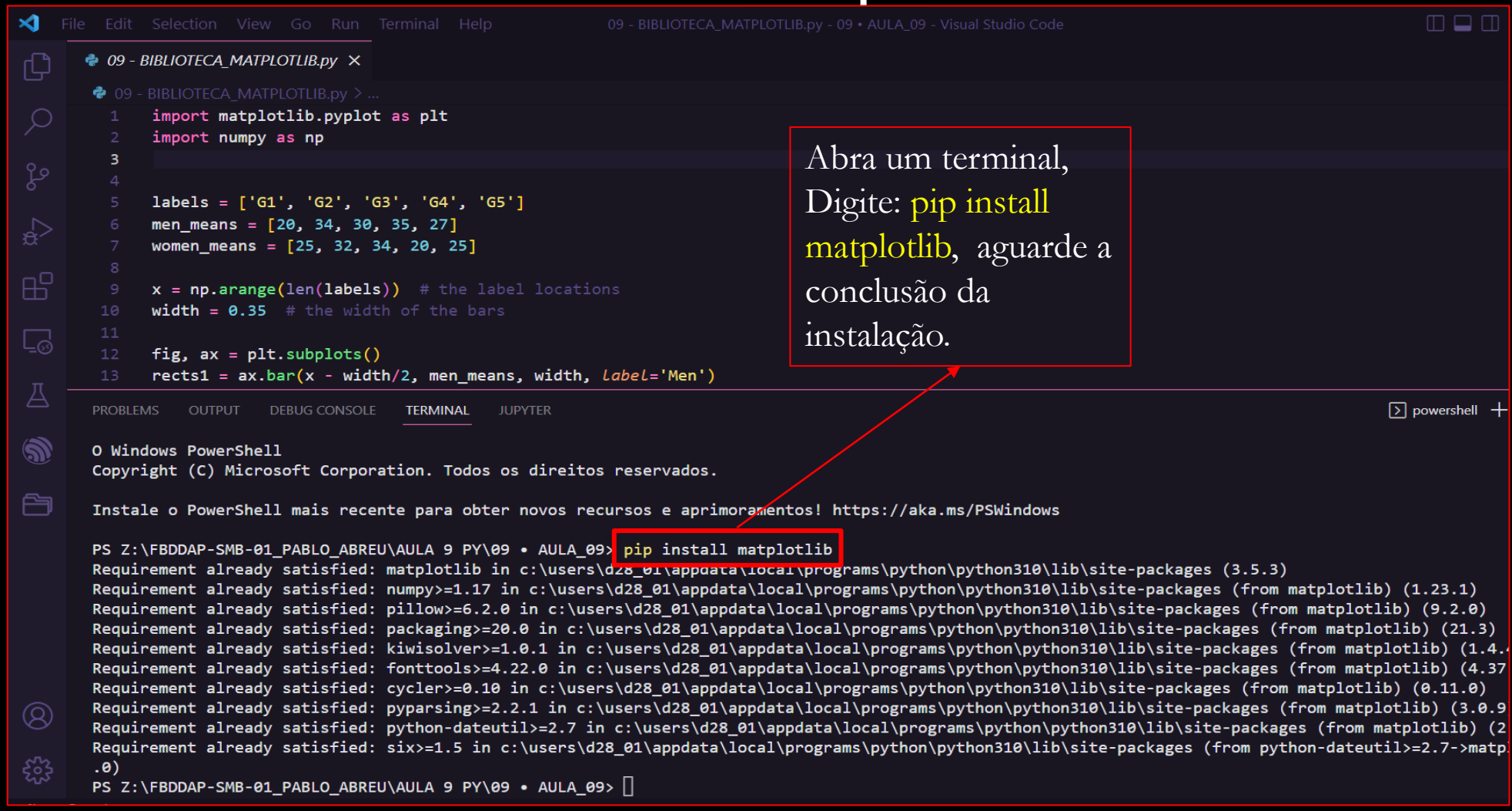




A biblioteca Matplotlib, escrita originalmente por **John D. Hunter** em 2003 é uma biblioteca abrangente para criar visualizações estáticas, animadas e interativas em Python. **Matplotlib** é fácil de usar e uma biblioteca de visualização incrível em Python, oferece uma interface de programação **orientada a objetos** para incluir gráficos através de kits de ferramentas de interface gráfica como o **Tkinter**, **Wxpython**, **Qt** ou **GTK**. É construído em arrayes NumPy e projetado para funcionar como um apoio cientifico mais amplo e consiste em vários gráficos como linha, barra, dispersão, histograma, etc.

Pablo Ricardo de abreu
10/2022

Instalando no Python a biblioteca Matplotlib



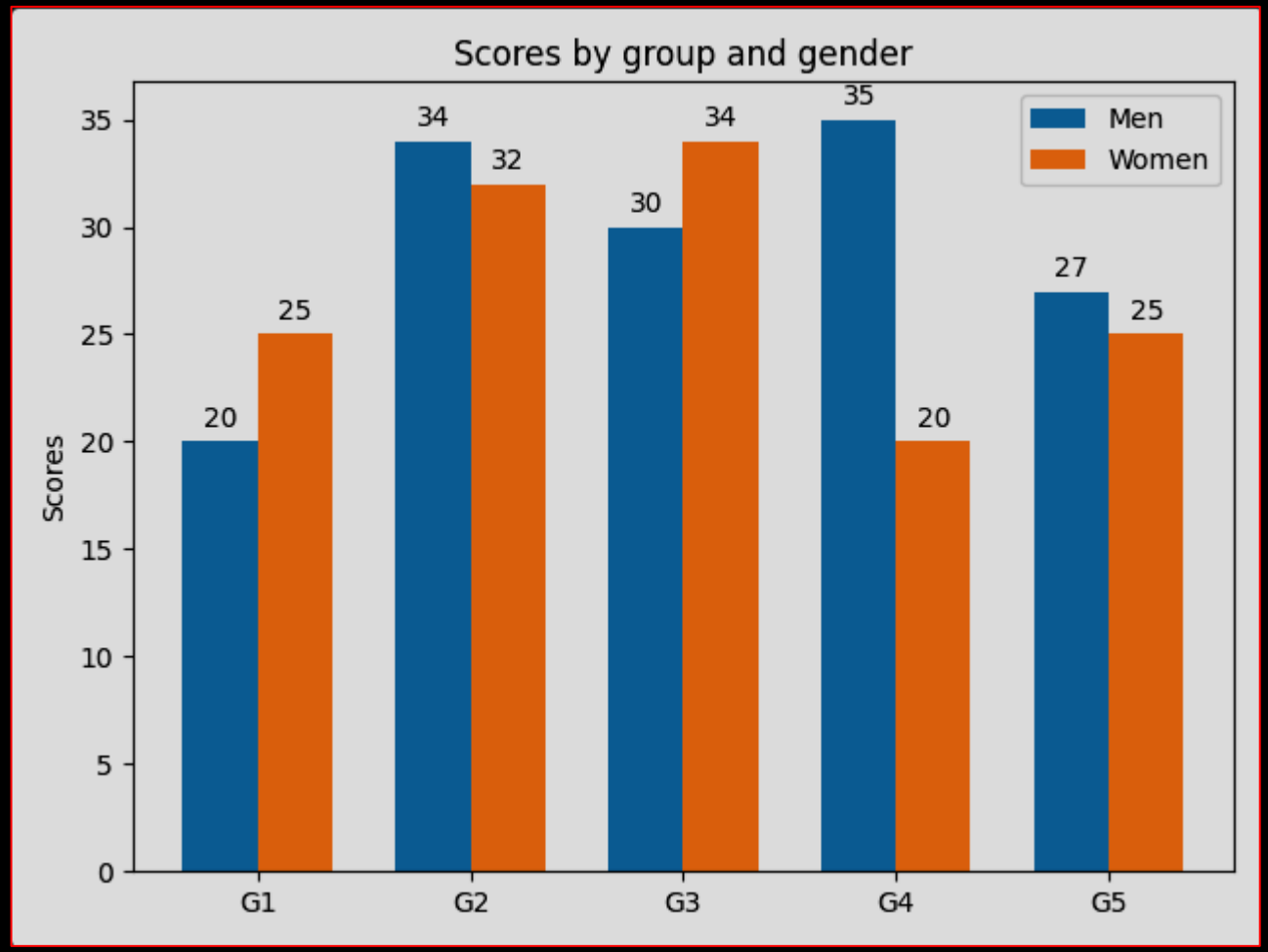
Abra um terminal, Digite: pip install matplotlib, aguarde a conclusão da instalação.

Pablo Ricardo de abreu
10/2022

Quais os principais gráficos ?

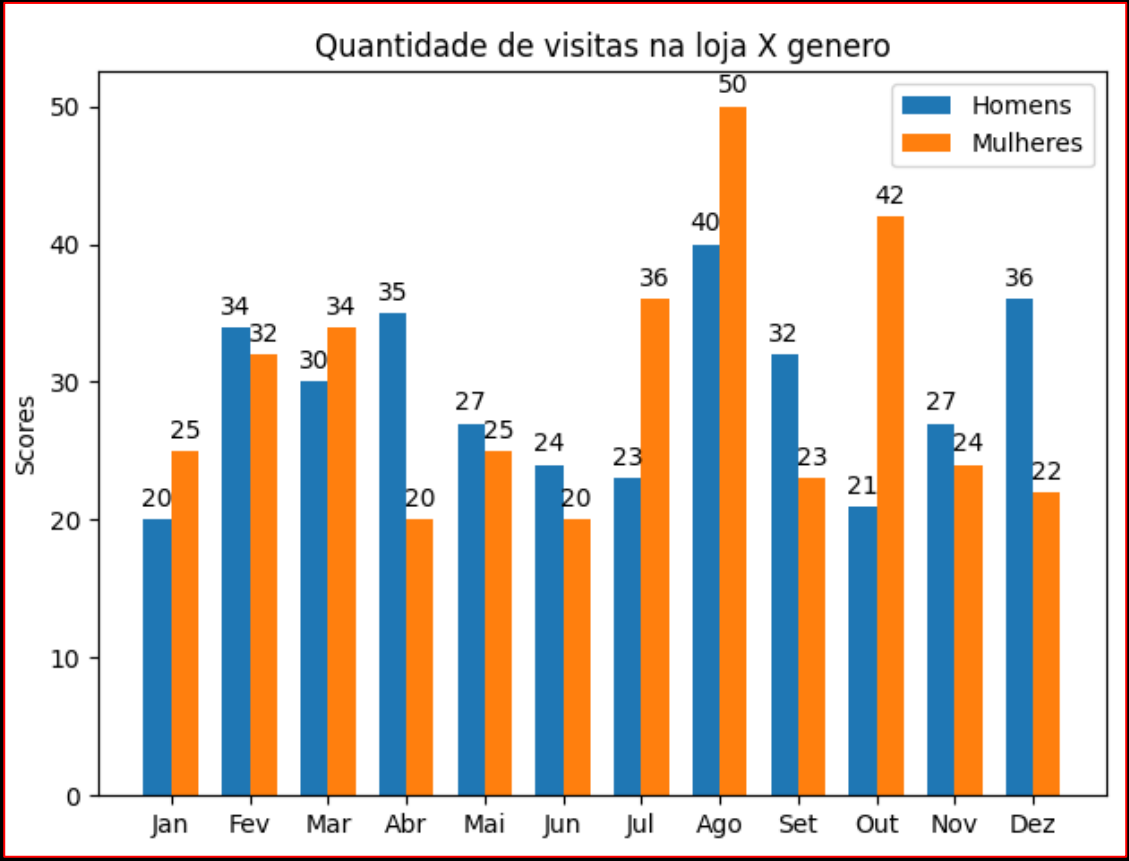
- Gráfico de barras :

- Mais utilizado para dados simples, onde a analise seja rápida, utilizamos para verificar evoluções no tempo ou comparações de duas variáveis nos eixos X e Y. Categorias ou grupos são comumente representados no eixo horizontal (eixo X) e uma escala quantitativa ou numérica é plotada no eixo vertical (eixo Y). Os gráficos de barras são usados para representar dados contínuos e descontínuos ou dados discretos.



Pablo Ricardo de abreu
10/2022

Exemplo de código para gráficos de Barras



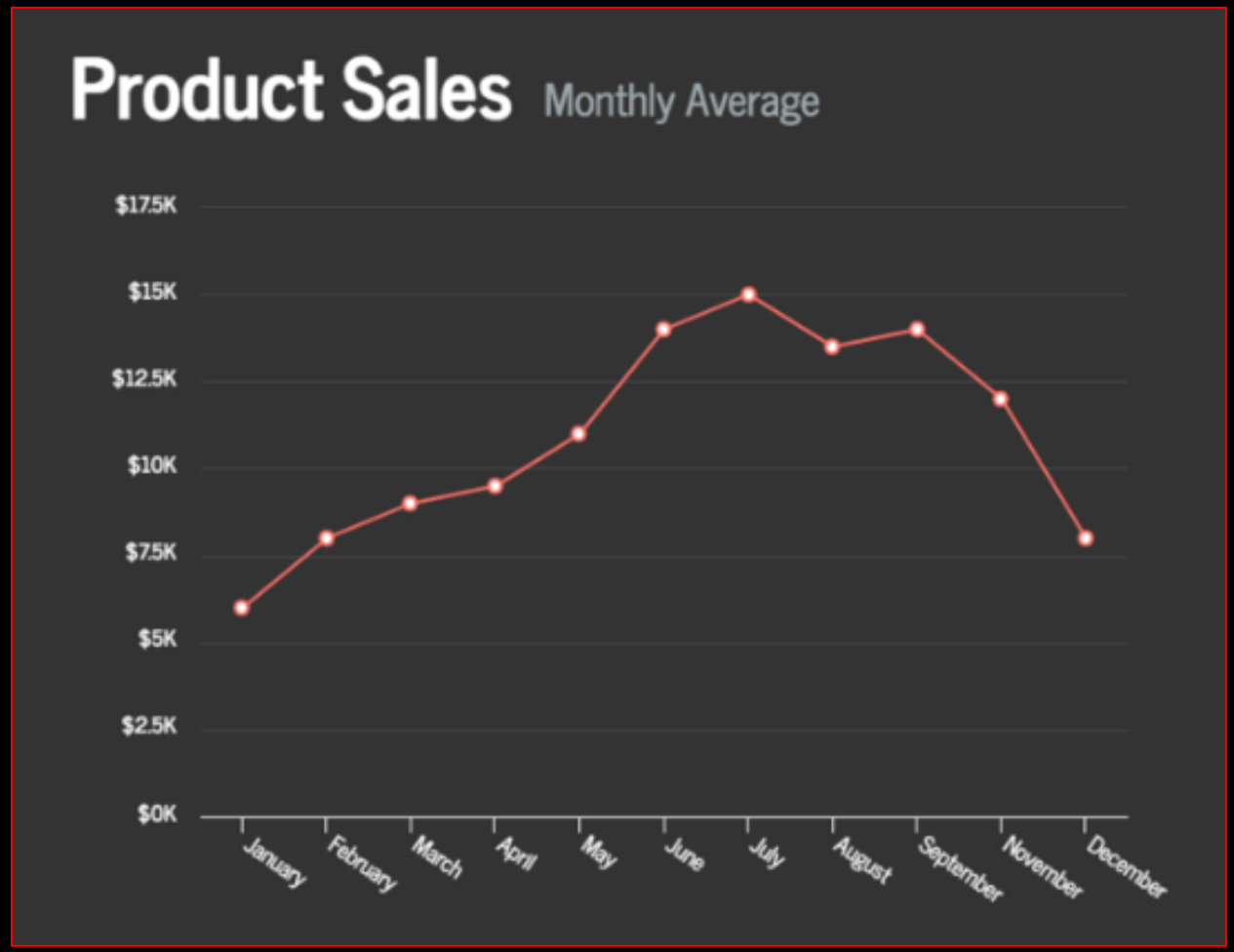
09 - BIBLIOTECA_MATPLOTLIB.py > ...

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

#dados
labels = ['Jan', 'Fev', 'Mar', 'Abr', 'Mai','Jun','Jul','Ago','Set','Out','Nov','Dez']
men_means = [20, 34, 30, 35, 27,24,23,40,32,21,27,36]
women_means = [25, 32, 34, 20, 25, 20, 36, 50, 23, 42,24,22]
#eixo X
x = np.arange(len(labels))  # the label locations
width = 0.35  # the width of the bars
#Unindo dados no mesmo eixo criando rotulo de dados
fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, men_means, width, label='Homens')
rects2 = ax.bar(x + width/2, women_means, width, label='Mulheres')

# Adicione alguns textos para rótulos,
# rótulos de marcação de título e eixo x personalizado, etc.
ax.set_ylabel('Scores')
ax.set_title('Quantidade de visitas na loja X genero')
ax.set_xticks(x, labels)
ax.legend()
ax.bar_label(rects1, padding=3)
ax.bar_label(rects2, padding=3)
fig.tight_layout()
plt.show()
```

Pablo Ricardo de abreu
10/2022

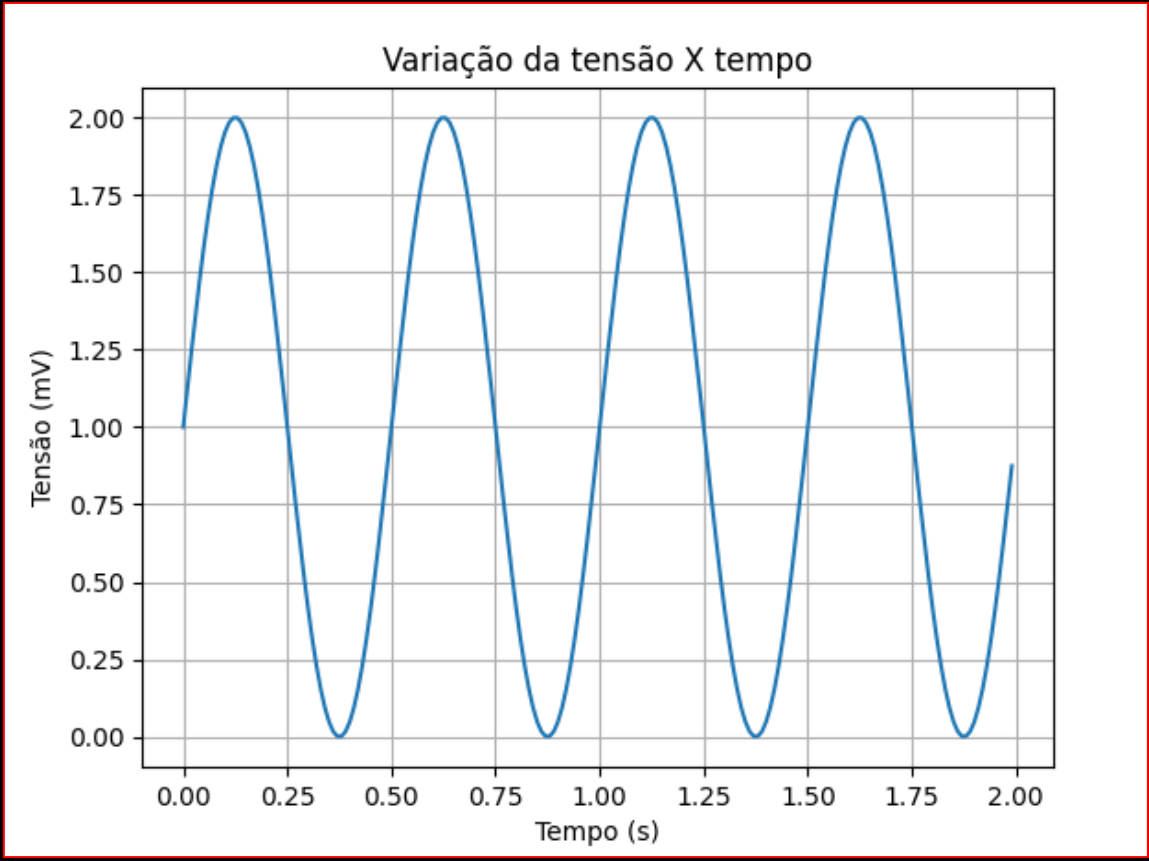
Quais os principais gráficos ?

- Gráfico de linha :
- Utilizado para analisar tendências de uma variável, podemos usar esse tipo de gráfico para verificar se ação da bolsa esta subindo ou caindo, uma onda senoidal do circuito eletrônico, temperatura de um ambiente ao longo do dia, custo de um produto no varejo ou no atacado e Etc.



Pablo Ricardo de abreu
10/2022

Exemplo de código para gráficos de linha



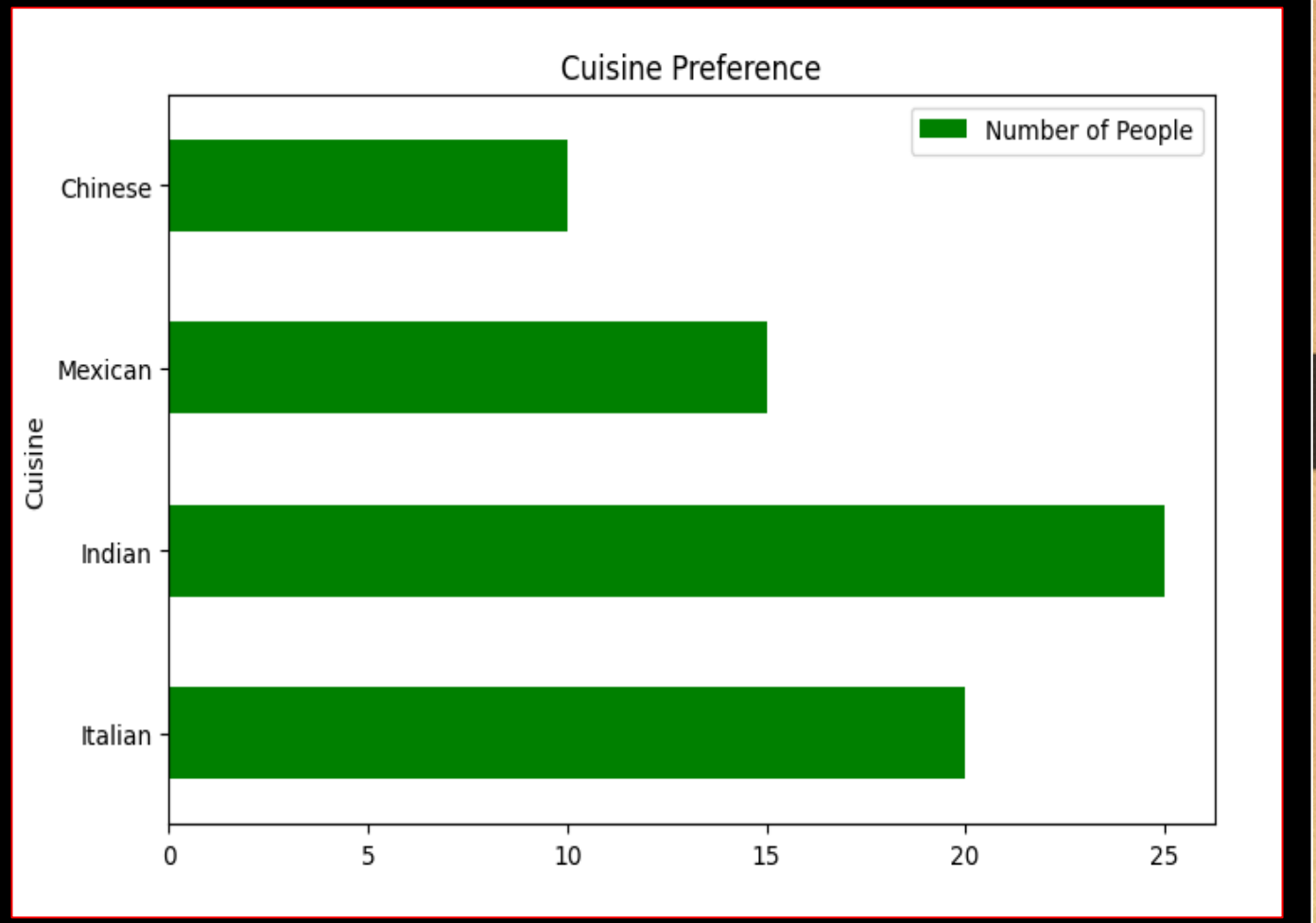
09 - BIBLIOTECA_MATPLOTLIB_LINHAS.py > ...

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

# dados para impressão
t = np.arange(0.0, 2.0, 0.01)
s = 1 + np.sin(4 * np.pi * t)

fig, ax = plt.subplots()
ax.plot(t, s)
ax.set(xlabel='Tempo (s)', ylabel='Tensão (mV)',
       title='Variação da tensão X tempo')
ax.grid()

#fig.savefig("test.png")
plt.show()

```

Pablo Ricardo de abreu
10/2022

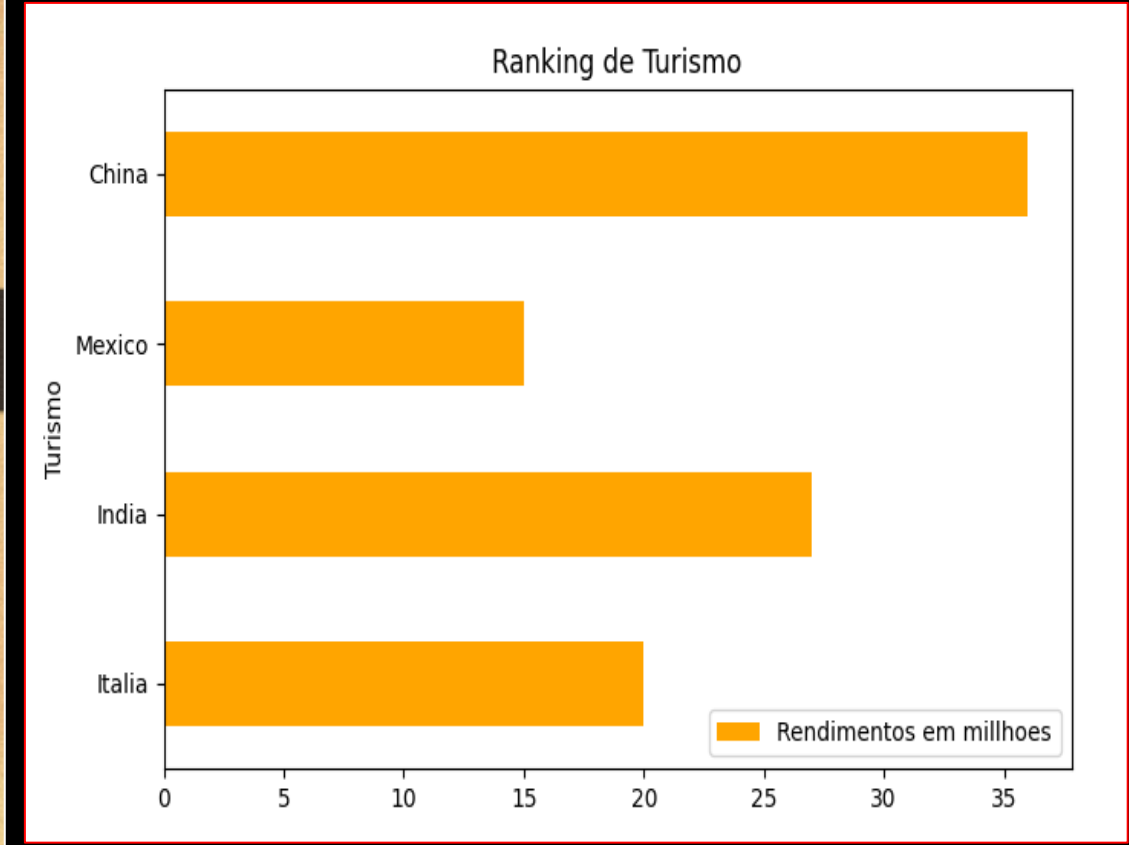
Quais os principais gráficos ?

- Gráfico de Horizontal de barras :
- Quando precisamos criar um ranking com uma base de dados, usamos um gráfico desse tipo, onde mostra a maior e a menor pontuação ou frequência listados por barras horizontais em ordem decrescente. Ao contrario do gráfico de barras verticais, esse modelo nos remete a uma corrida e que esta com a vantagem tem o tamanho da sua barra maior.



Pablo Ricardo de abreu
10/2022

Exemplo de código para gráficos Horizontal de barra



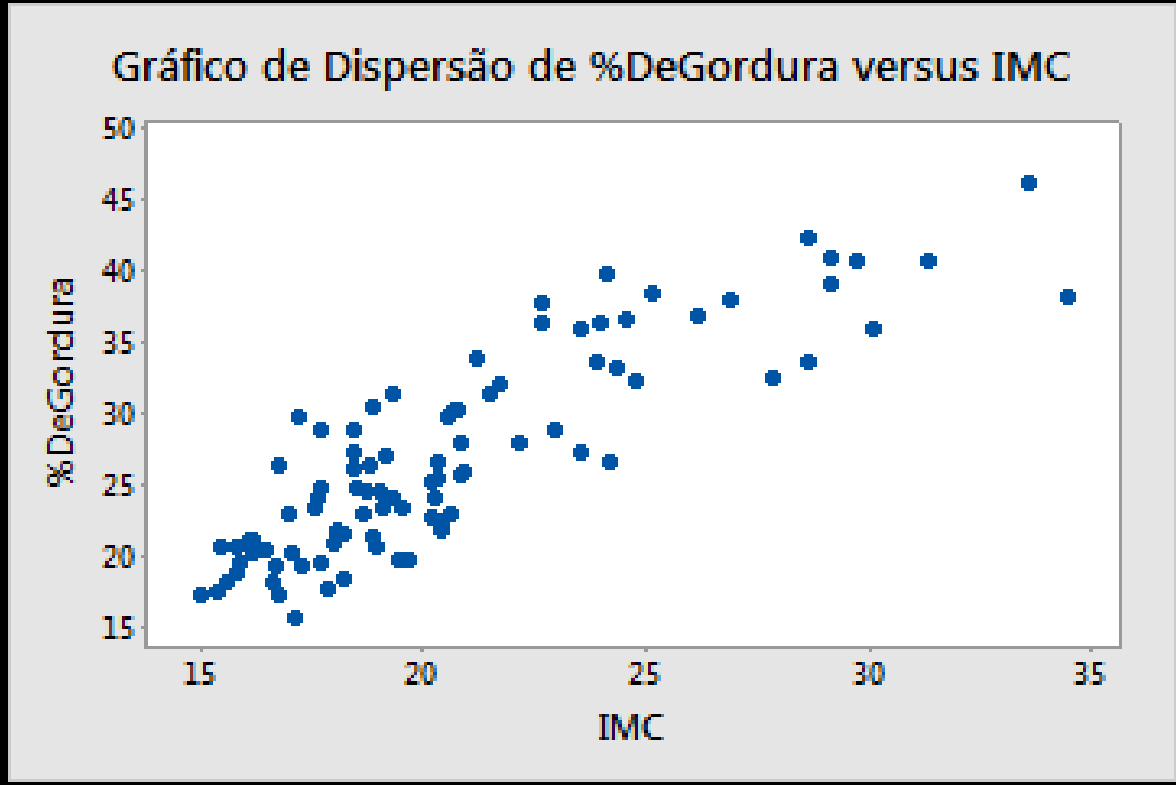
09 - BIBLIOTECA_MATPLOTLIB_HBARRAS.py > ...

```python
import pandas as pd
import matplotlib.pyplot  as plt

# Criando sua base de dados
df = pd.DataFrame({'Turismo': ['Italia', 'India', 'Mexico', 'China'],
                   'Rendimentos em millhoes': [20, 27, 15, 36]})

# impressão do grafico de barra horizontal veja: barh
df.plot.barh(x='Turismo', y='Rendimentos em millhoes',
             title='Ranking de Turismo', color='orange')

plt.show()

```

Pablo Ricardo de abreu
10/2022

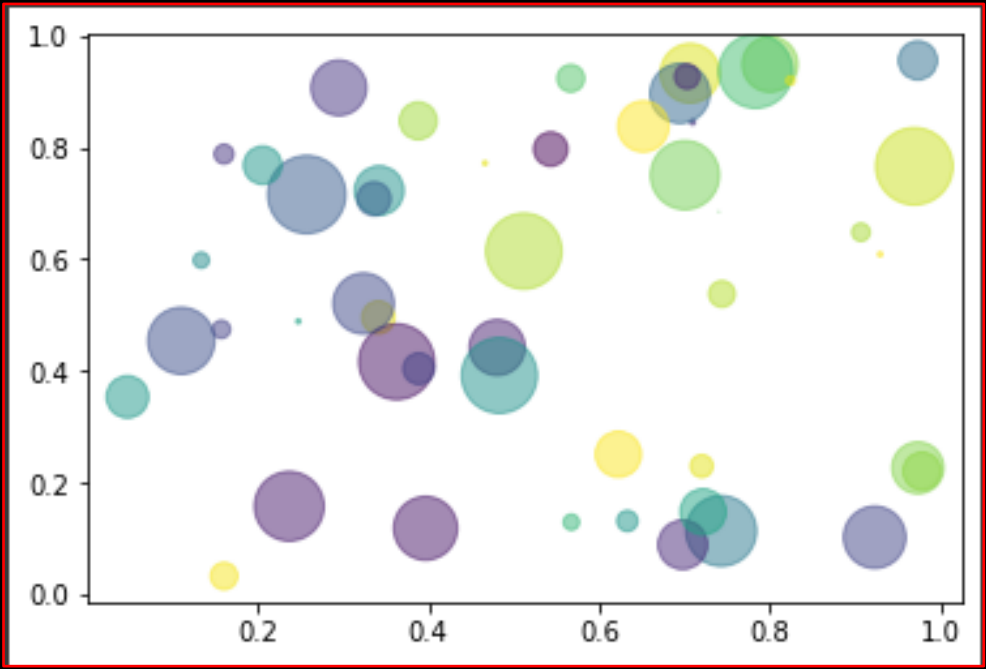
Quais os principais gráficos ?

- Gráfico de dispersão.
- O gráfico de dispersão é uma ferramenta representada em um gráfico, representada por um eixo horizontal e outro vertical. Dessa forma, é fácil identificar visualmente a correlação entre dois pontos e identificar diferentes tipos de correlação de causa e efeito entre eles.



Pablo Ricardo de abreu
10/2022

Exemplo de código para gráficos dispersão

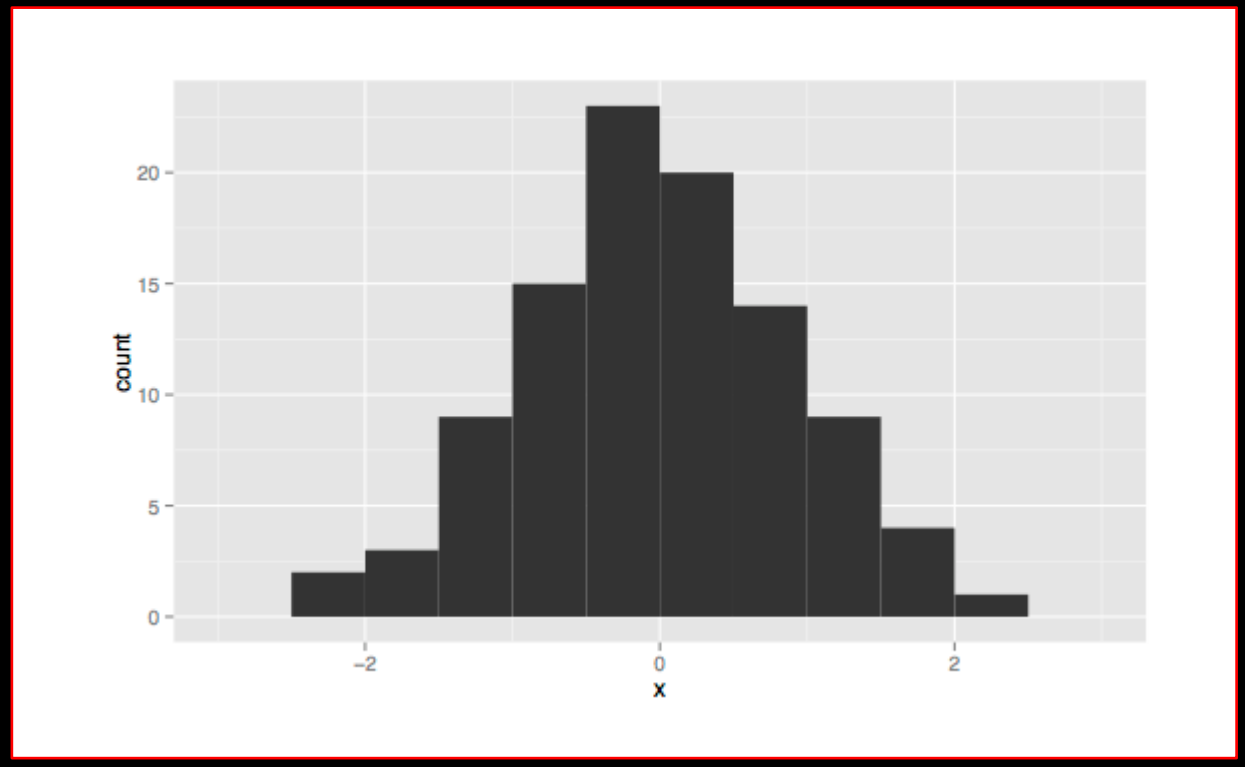



```python
import numpy as np
import matplotlib.pyplot as plt

# Fixing random state for reproducibility
np.random.seed(19680801)


N = 50
x = np.random.rand(N)
y = np.random.rand(N)
colors = np.random.rand(N)
area = (30 * np.random.rand(N))**2  # 0 to 15 point radii

plt.scatter(x, y, s=area, c=colors, alpha=0.5)
plt.show()
```

Pablo Ricardo de abreu
10/2022

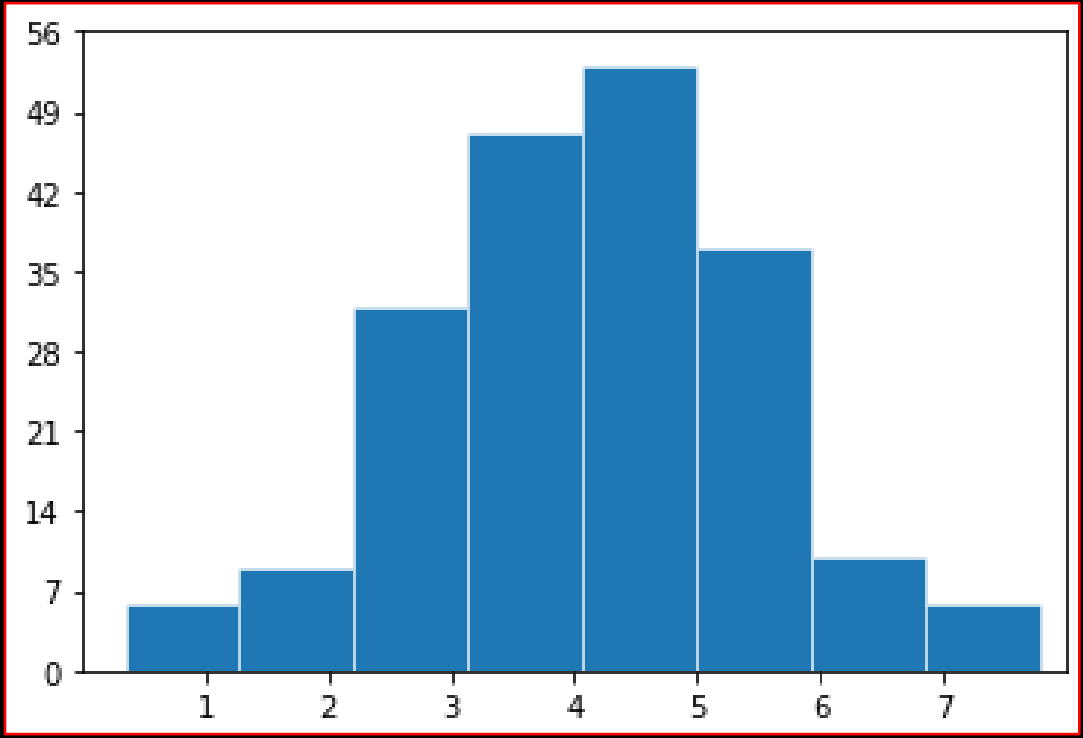
Quais os principais gráficos ?

- Gráfico Histograma
- Um histograma é uma espécie de gráfico de barras que demonstra uma distribuição de frequências. No histograma, a base de cada uma das barras representa uma classe e a altura representa a quantidade ou frequência absoluta com que o valor de cada classe ocorre. Ao mesmo tempo, ele pode ser utilizado como um indicador de dispersão de processos.



Pablo Ricardo de abreu
10/2022

Exemplo de código para gráficos tipo Histograma

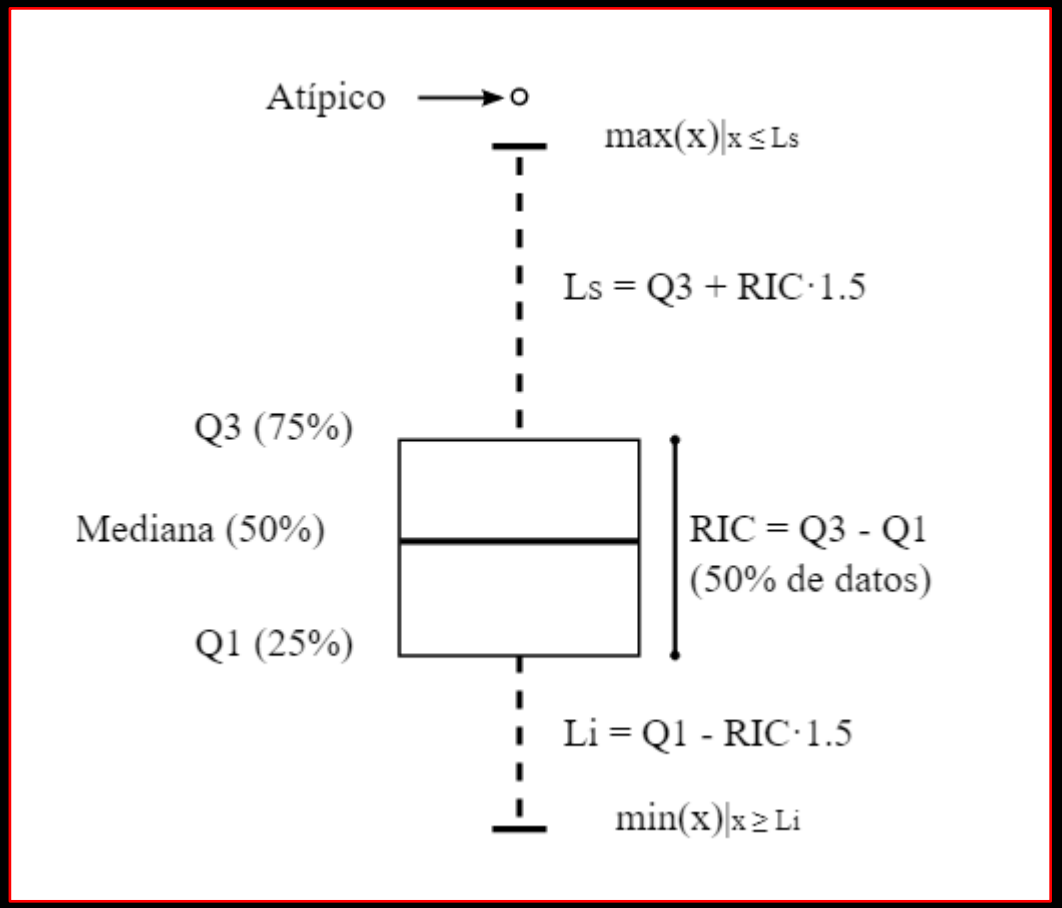



```python
import matplotlib.pyplot as plt
import numpy as np



# make data
np.random.seed(1)
x = 4 + np.random.normal(0, 1.5, 200)

# plot:
fig, ax = plt.subplots()

ax.hist(x, bins=8, linewidth=0.5, edgecolor="white")

ax.set(xlim=(0, 8), xticks=np.arange(1, 8),
       ylim=(0, 56), yticks=np.linspace(0, 56, 9))

plt.show()
```

Pablo Ricardo de abreu
10/2022

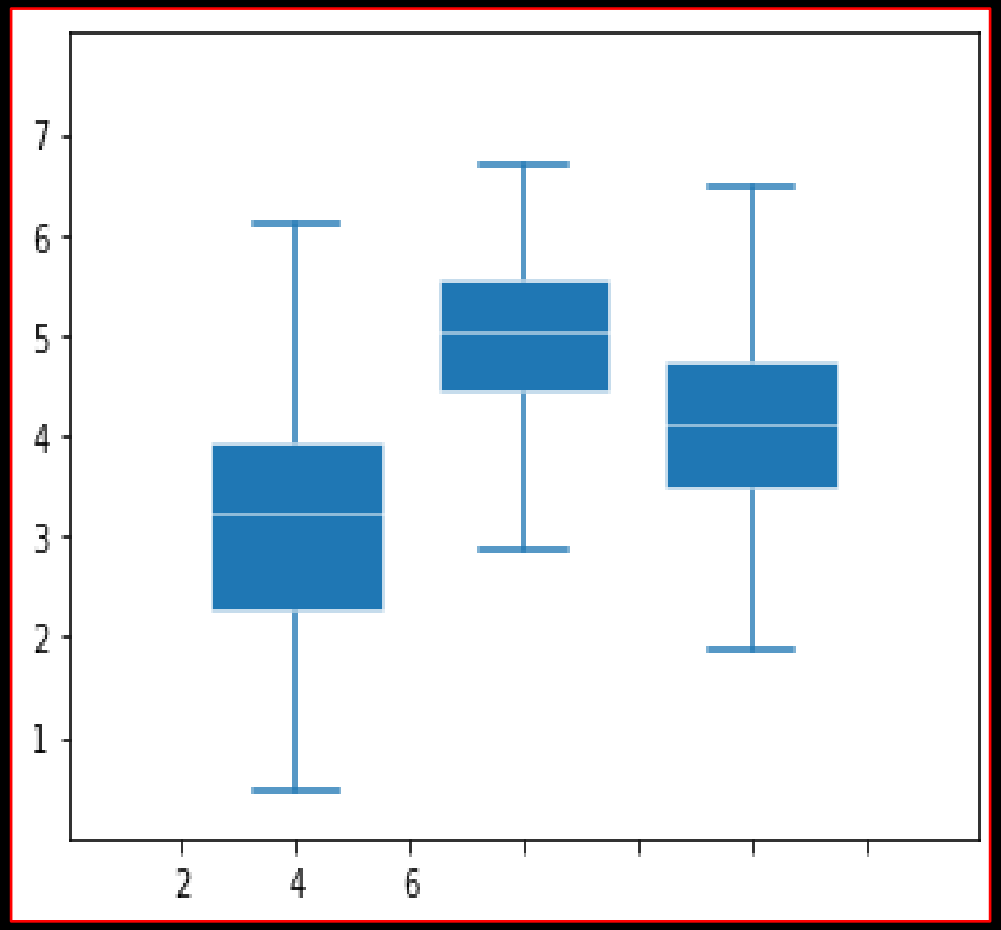
Quais os principais gráficos ?

- Gráfico Boxplot
- Um boxplot (ou diagrama de caixa, numa tradução livre) mostra a distribuição quantitativa dos dados de um jeito que facilita a comparação entre as variáveis, ou através dos níveis categóricos das variáveis. Essa caixa (“box”) mostra os quartis do dataset enquanto os “whiskers” mostram o resto da distribuição, exceto os pontos que são chamados de outliers.



Pablo Ricardo de abreu
10/2022

Exemplo de código para gráficos boxplot

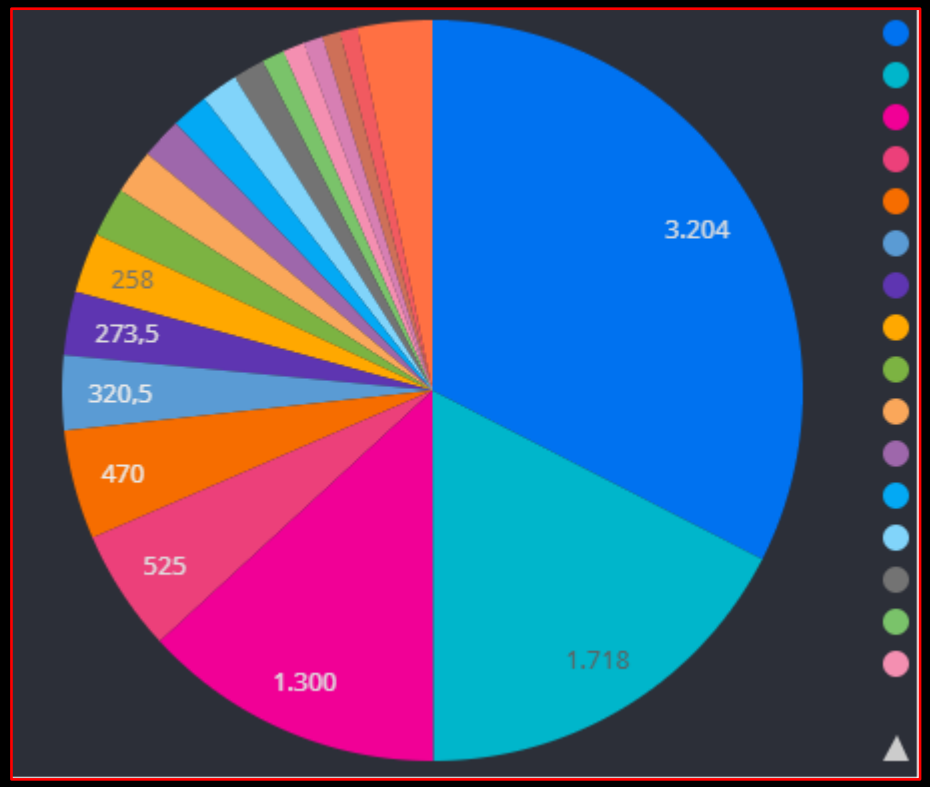



```python
import matplotlib.pyplot as plt
import numpy as np



# make data:
np.random.seed(10)
D = np.random.normal((3, 5, 4), (1.25, 1.00, 1.25), (100, 3))

# plot
fig, ax = plt.subplots()
VP = ax.boxplot(D, positions=[2, 4, 6], widths=1.5, patch_artist=True,
                showmeans=False, showfliers=False,
                medianprops={"color": "white", "linewidth": 0.5},
                boxprops={"facecolor": "C0", "edgecolor": "white",
                          "linewidth": 0.5},
                whiskerprops={"color": "C0", "linewidth": 1.5},
                capprops={"color": "C0", "linewidth": 1.5})

ax.set(xlim=(0, 8), xticks=np.arange(1, 8),
       ylim=(0, 8), yticks=np.arange(1, 8))

plt.show()
```

Pablo Ricardo de abreu
10/2022

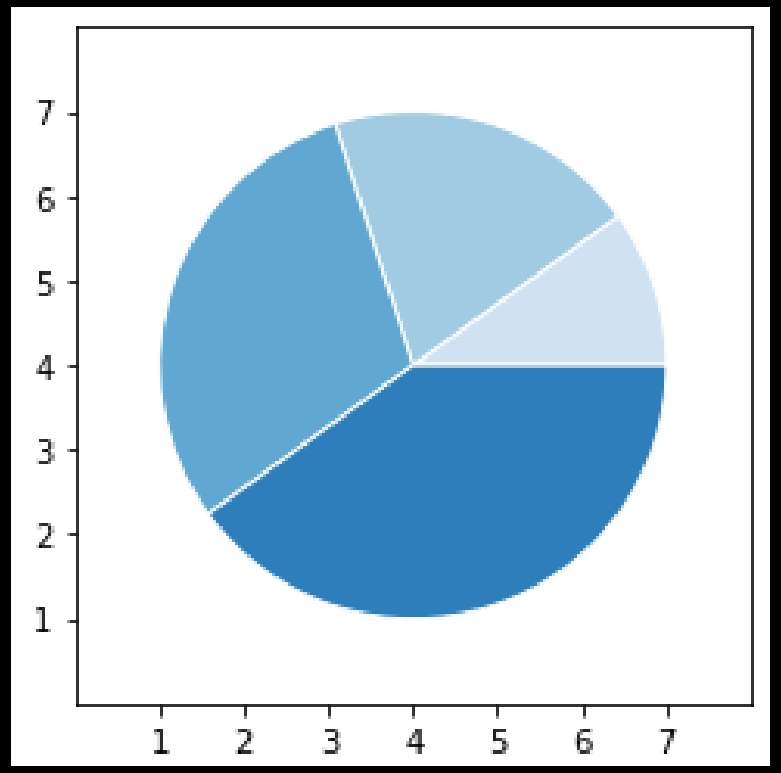
Quais os principais gráficos ?

- Gráfico Pizza.
- O gráfico de setores, comumente chamado de gráfico de pizza, é uma visualização que representa um valor relativo de cada categoria estabelecida em relação a um todo. Porém, essa visualização tem sido fortemente criticada quanto a sua utilização por alguns fatores, e são eles que veremos agora.



Pablo Ricardo de abreu
10/2022

Exemplo de código para gráficos pizza

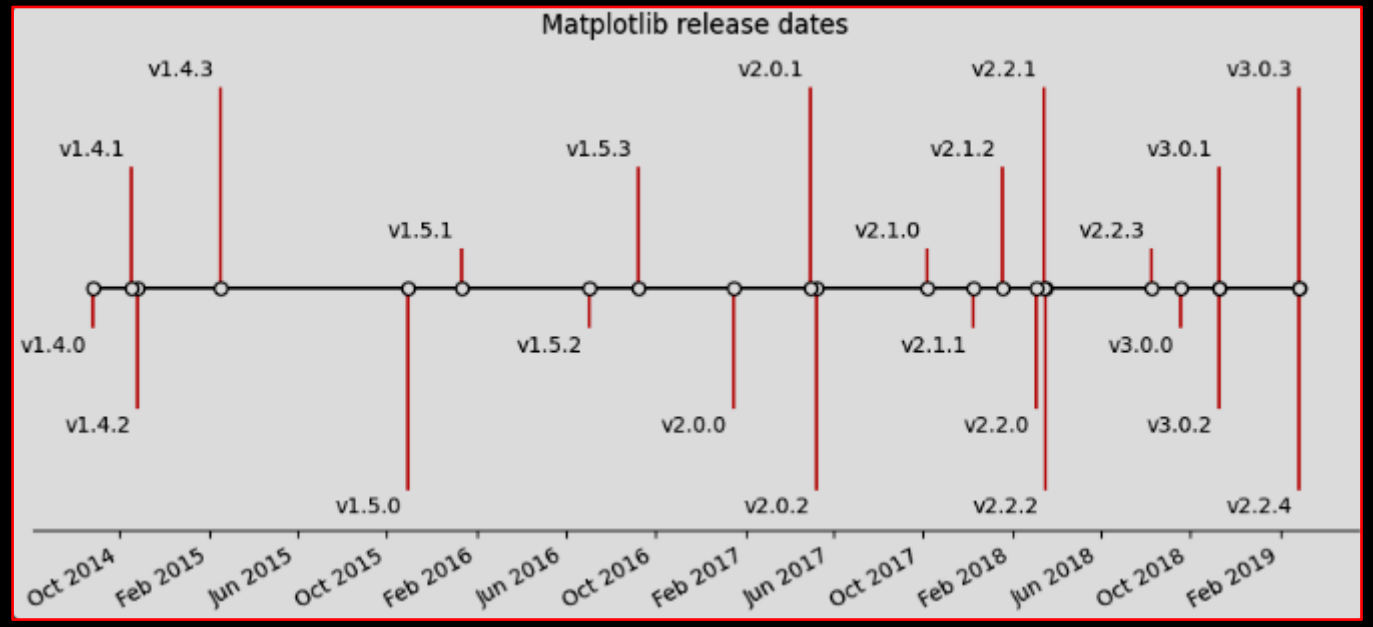



```python
import matplotlib.pyplot as plt
import numpy as np




# make data
x = [1, 2, 3, 4]
colors = plt.get_cmap('Blues')(np.linspace(0.2, 0.7, len(x)))

# plot
fig, ax = plt.subplots()
ax.pie(x, colors=colors, radius=3, center=(4, 4),
       wedgeprops={"linewidth": 1, "edgecolor": "white"}, frame=True)

ax.set(xlim=(0, 8), xticks=np.arange(1, 8),
       ylim=(0, 8), yticks=np.arange(1, 8))

plt.show()
```

Pablo Ricardo de abreu
10/2022

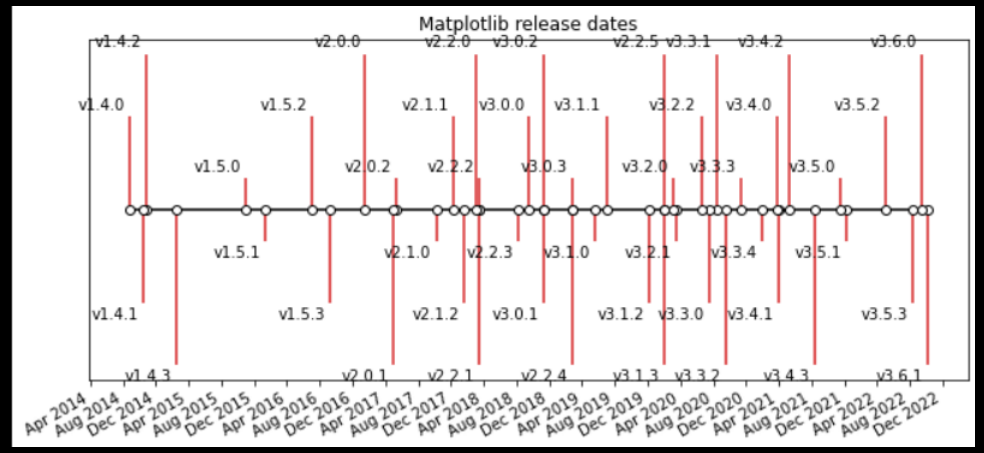
Quais os principais gráficos ?

- Gráfico linha do tempo
- Mais utilizado em exibição de um histórico em forma de linha com os marcadores sendo uma data um ano expecifico.



Pablo Ricardo de abreu
10/2022

Exemplo de código para gráficos linha do tempo



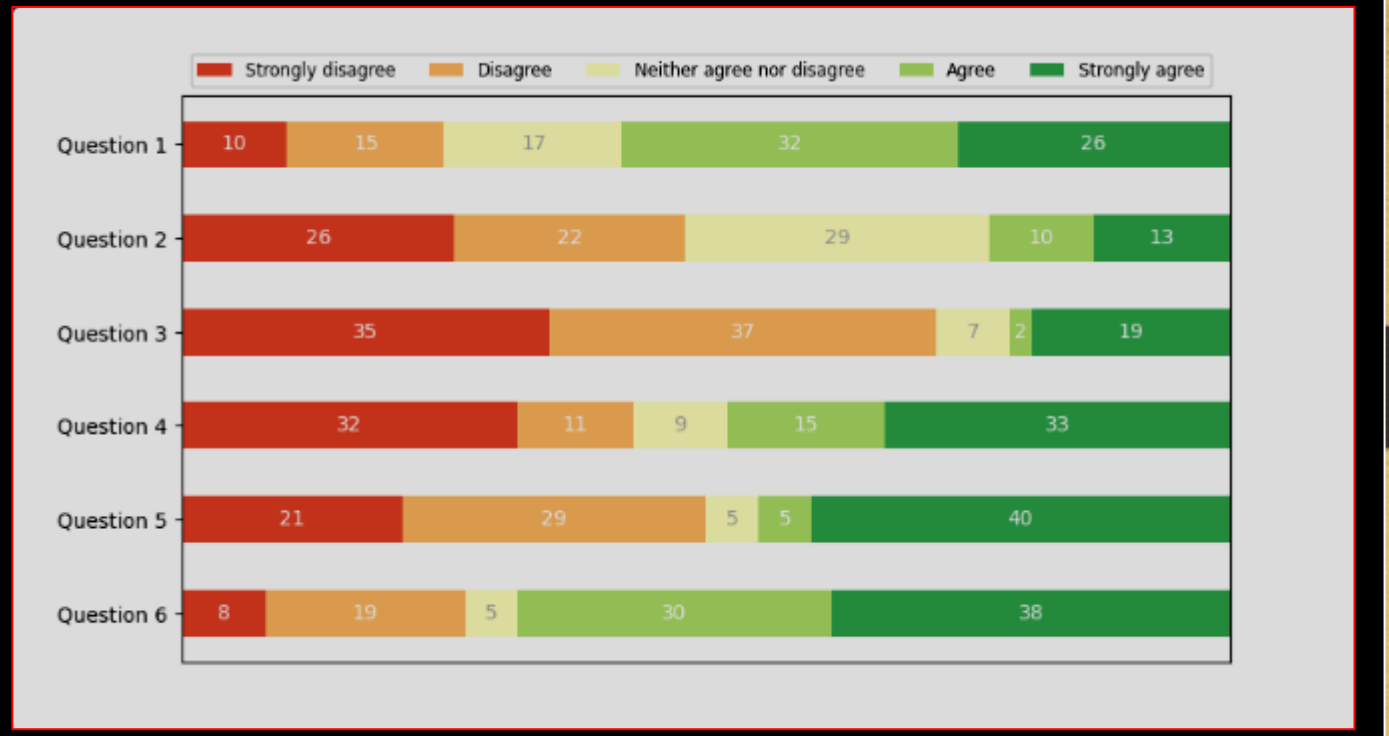

```python
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.dates as mdates
from datetime import datetime

try:
    # Try to fetch a list of Matplotlib releases and their dates
    # from https://api.github.com/repos/matplotlib/matplotlib/releases
    import urllib.request
    import json

    url = 'https://api.github.com/repos/matplotlib/matplotlib/releases'
    url += '?per_page=100'
    data = json.loads(urllib.request.urlopen(url, timeout=.4).read().decode())

    dates = []
    names = []
    for item in data:
        if 'rc' not in item['tag_name'] and 'b' not in item['tag_name']:
            dates.append(item['published_at'].split("T")[0])
            names.append(item['tag_name'])
    # Convert date strings (e.g. 2014-10-18) to datetime
    dates = [datetime.strptime(d, "%Y-%m-%d") for d in dates]

except Exception:
    # In case the above fails, e.g. because of missing internet connection
    # use the following lists as fallback.
    names = ['v2.2.4', 'v3.0.3', 'v3.0.2', 'v3.0.1', 'v3.0.0', 'v2.2.3',
             'v2.2.2', 'v2.2.1', 'v2.2.0', 'v2.1.2', 'v2.1.1', 'v2.1.0',
             'v2.0.2', 'v2.0.1', 'v2.0.0', 'v1.5.3', 'v1.5.2', 'v1.5.1',
             'v1.5.0', 'v1.4.3', 'v1.4.2', 'v1.4.1', 'v1.4.0']

    dates = ['2019-02-26', '2019-02-26', '2018-11-10', '2018-11-10',
             '2018-09-18', '2018-08-10', '2018-03-17', '2018-03-16',
             '2018-03-06', '2018-01-18', '2017-12-10', '2017-10-07',
             '2017-05-10', '2017-05-02', '2017-01-17', '2016-09-09',
             '2016-07-03', '2016-01-10', '2015-10-29', '2015-02-16',
             '2014-10-26', '2014-10-18', '2014-08-26']
```

```python
    # Convert date strings (e.g. 2014-10-18) to datetime
    dates = [datetime.strptime(d, "%Y-%m-%d") for d in dates]

    # Choose some nice levels
levels = np.tile([-5, 5, -3, 3, -1, 1],
                 int(np.ceil(len(dates)/6)))[:len(dates)]

# Create figure and plot a stem plot with the date
fig, ax = plt.subplots(figsize=(8.8, 4), constrained_layout=True)
ax.set(title="Matplotlib release dates")

ax.vlines(dates, 0, levels, color="tab:red")  # The vertical stems.
ax.plot(dates, np.zeros_like(dates), "-o",
        color="k", markerfacecolor="w")  # Baseline and markers on it.

# annotate lines
for d, l, r in zip(dates, levels, names):
    ax.annotate(r, xy=(d, l),
                xytext=(-3, np.sign(l)*3), textcoords="offset points",
                horizontalalignment="right",
                verticalalignment="bottom" if l > 0 else "top")

# format xaxis with 4 month intervals
ax.xaxis.set_major_locator(mdates.MonthLocator(interval=4))
ax.xaxis.set_major_formatter(mdates.DateFormatter("%b %Y"))
plt.setp(ax.get_xticklabels(), rotation=30, ha="right")

# remove y axis and spines
ax.yaxis.set_visible(False)
ax.spines[["left", "top", "right"]].set_visible(False)

ax.margins(y=0.1)
plt.show()
```

Pablo Ricardo de abreu
10/2022

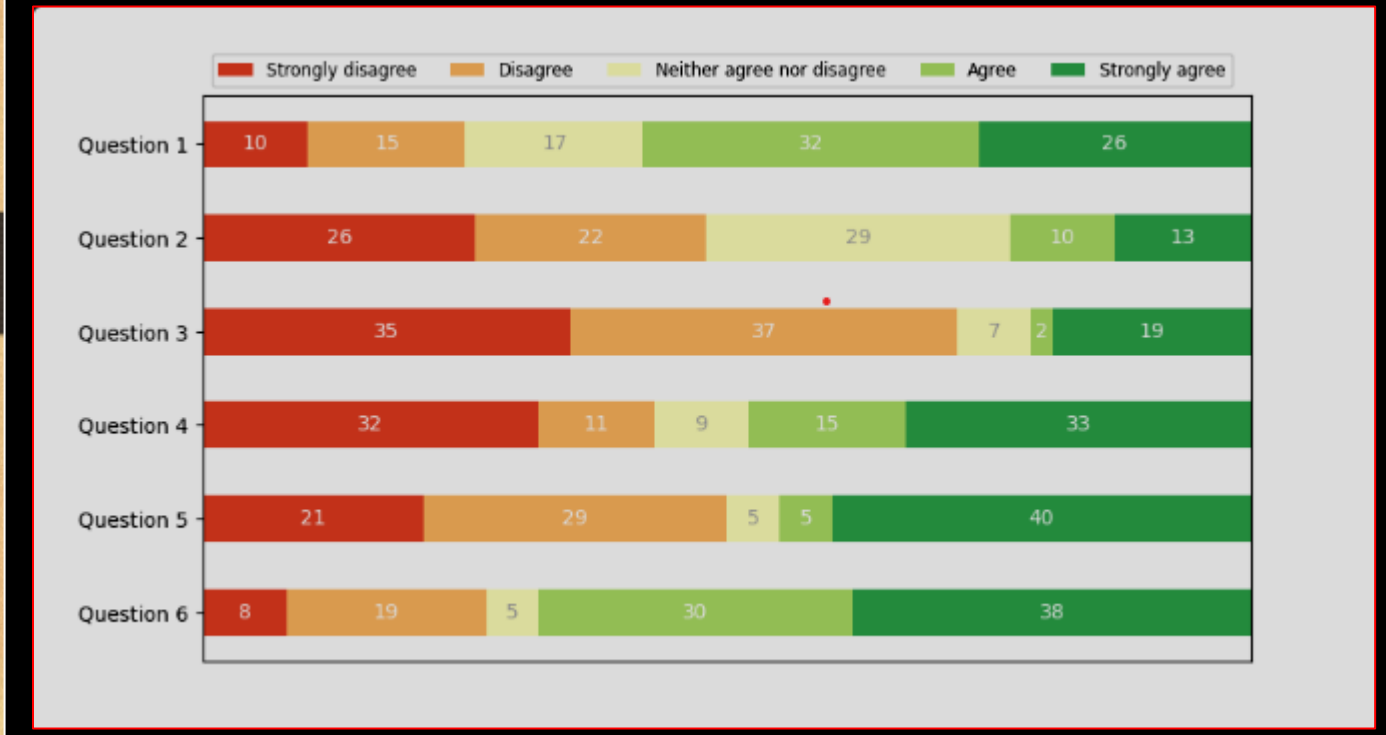
Quais os principais gráficos ?

- Gráfico de barras horizontais modificado.
- Os gráficos de barras empilhadas podem ser usados para visualizar distribuições discretas.Este exemplo visualiza o resultado de uma pesquisa na qual as pessoas podem avaliar sua concordância com as perguntas em uma escala de cinco elementos.O empilhamento horizontal é obtido chamando barh()para cada categoria e passando o ponto de partida como a soma cumulativa das barras já desenhadas através do parâmetro left.



Pablo Ricardo de abreu
10/2022

Exemplo de código para gráficos barras horizontais modificado

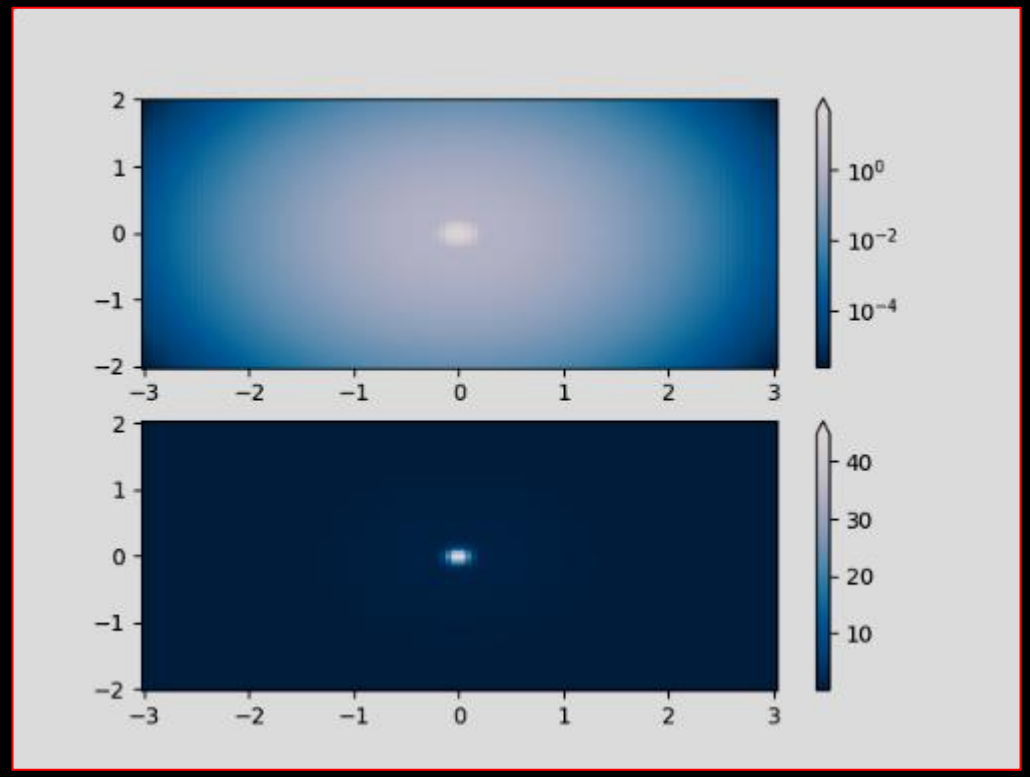



```python
import numpy as np
import matplotlib.pyplot as plt


category_names = ['Strongly disagree', 'Disagree',
                  'Neither agree nor disagree', 'Agree', 'Strongly agree']
results = {
    'Question 1': [10, 15, 17, 32, 26],
    'Question 2': [26, 22, 29, 10, 13],
    'Question 3': [35, 37, 7, 2, 19],
    'Question 4': [32, 11, 9, 15, 33],
    'Question 5': [21, 29, 5, 5, 40],
    'Question 6': [8, 19, 5, 30, 38]
}


def survey(results, category_names):
    """
    Parameters
    ----------
    results : dict
        A mapping from question labels to a list of answers per category.
        It is assumed all lists contain the same number of entries and that
        it matches the length of *category_names*.
    category_names : list of str
        The category labels.
    """
    labels = list(results.keys())
    data = np.array(list(results.values()))
    data_cum = data.cumsum(axis=1)
    category_colors = plt.colormaps['RdYlGn'](
        np.linspace(0.15, 0.85, data.shape[1]))

    fig, ax = plt.subplots(figsize=(9.2, 5))
    ax.invert_yaxis()
    ax.xaxis.set_visible(False)
    ax.set_xlim(0, np.sum(data, axis=1).max())

    for i, (colname, color) in enumerate(zip(category_names, category_colors)):
        widths = data[:, i]
        starts = data_cum[:, i] - widths
        rects = ax.barh(labels, widths, left=starts, height=0.5,
                        label=colname, color=color)

        r, g, b, _ = color
        text_color = 'white' if r * g * b < 0.5 else 'darkgrey'
        ax.bar_label(rects, label_type='center', color=text_color)
    ax.legend(ncol=len(category_names), bbox_to_anchor=(0, 1),
              loc='lower left', fontsize='small')

    return fig, ax


survey(results, category_names)
plt.show()
```

Pablo Ricardo de abreu
10/2022

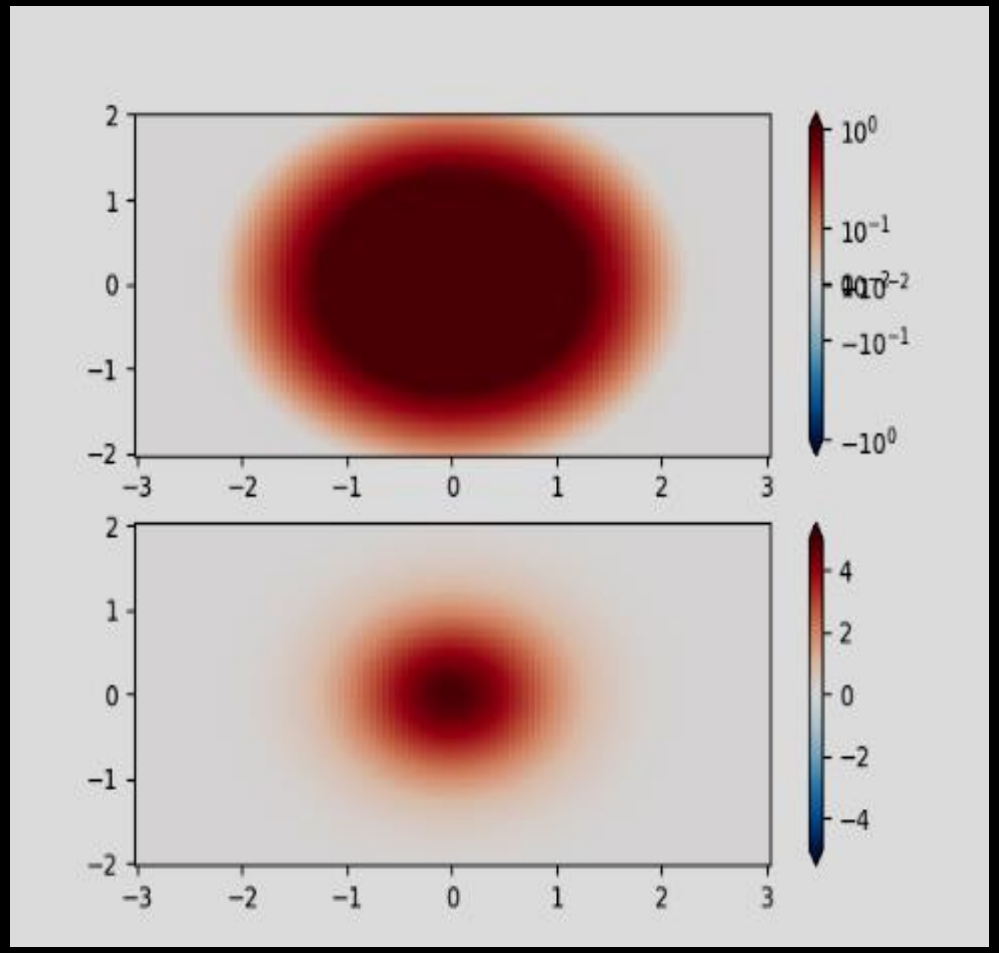
Quais os principais gráficos ?

- Gráfico mapa de cores.
- Demonstração do uso de norma para mapear mapas de cores em dados de maneiras não lineares.



Pablo Ricardo de abreu
10/2022

Exemplo de código para gráficos Mapa de cores

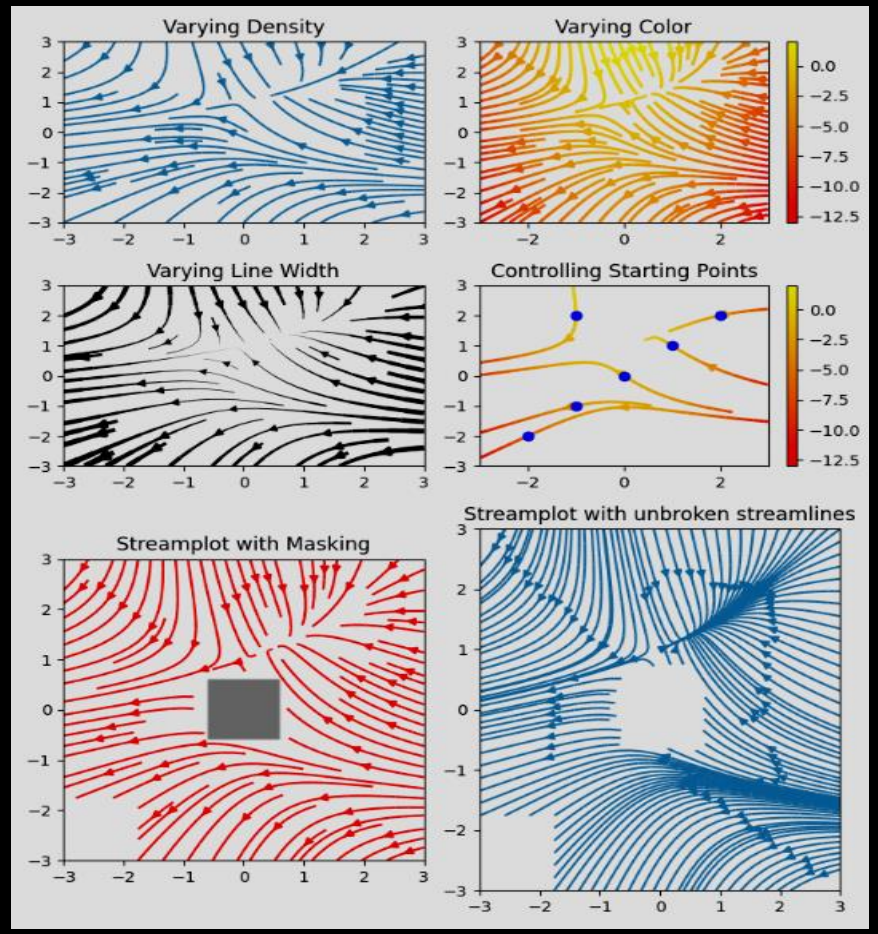



```python
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as colors

N = 100
X, Y = np.mgrid[-3:3:complex(0, N), -2:2:complex(0, N)]

# A low hump with a spike coming out of the top.  Needs to have
# z/colour axis on a log scale so we see both hump and spike.  linear
# scale only shows the spike.

Z1 = np.exp(-X**2 - Y**2)
Z2 = np.exp(-(X * 10)**2 - (Y * 10)**2)
Z = Z1 + 50 * Z2

fig, ax = plt.subplots(2, 1)

pcm = ax[0].pcolor(X, Y, Z,norm=colors.LogNorm(vmin=Z.min(), vmax=Z.max()),cmap='PuBu_r', shading='nearest')
fig.colorbar(pcm, ax=ax[0], extend='max')

pcm = ax[1].pcolor(X, Y, Z, cmap='PuBu_r', shading='nearest')
fig.colorbar(pcm, ax=ax[1], extend='max')

X, Y = np.mgrid[0:3:complex(0, N), 0:2:complex(0, N)]
Z1 = (1 + np.sin(Y * 10.)) * X**2

fig, ax = plt.subplots(2, 1)

pcm = ax[0].pcolormesh(X, Y, Z1, norm=colors.PowerNorm(gamma=1. / 2.),
                       cmap='PuBu_r', shading='nearest')
fig.colorbar(pcm, ax=ax[0], extend='max')

pcm = ax[1].pcolormesh(X, Y, Z1, cmap='PuBu_r', shading='nearest')
fig.colorbar(pcm, ax=ax[1], extend='max')

X, Y = np.mgrid[-3:3:complex(0, N), -2:2:complex(0, N)]
Z1 = 5 * np.exp(-X**2 - Y**2)
Z2 = np.exp(-(X - 1)**2 - (Y - 1)**2)
Z = (Z1 - Z2) * 2

fig, ax = plt.subplots(2, 1)

pcm = ax[0].pcolormesh(X, Y, Z1,norm=colors.SymLogNorm(linthresh=0.03, linscale=0.03,vmin=-1.0, vmax=1.0, base=10),cmap='RdBu_r', shading='nearest')
fig.colorbar(pcm, ax=ax[0], extend='both')

pcm = ax[1].pcolormesh(X, Y, Z1, cmap='RdBu_r', vmin=-np.max(Z1),shading='nearest')
fig.colorbar(pcm, ax=ax[1], extend='both')
```

Pablo Ricardo de abreu
10/2022

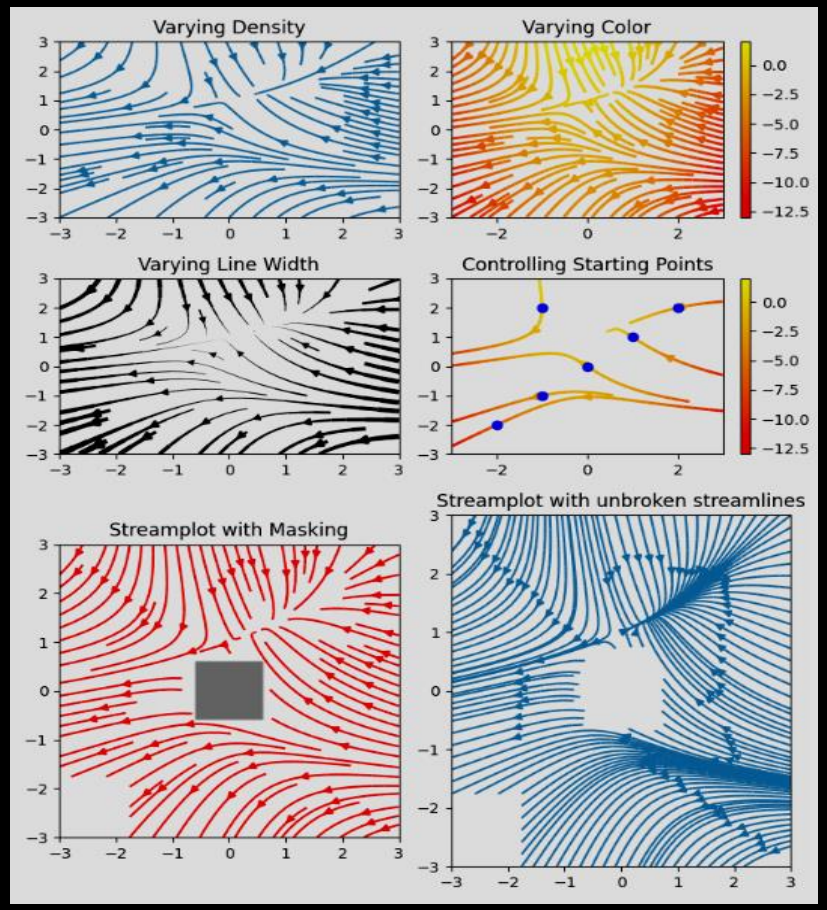
Quais os principais gráficos ?

- **Gráfico de fluxo.**
- Um gráfico de fluxo, ou gráfico simplificado, é usado para exibir campos vetoriais 2D. Este exemplo mostra alguns recursos da streamplot função:Variando a cor ao longo de uma linha de corrente.Variando a densidade de linhas de corrente.Variando a largura da linha ao longo de uma linha de corrente. Controlar os pontos de partida das linhas de corrente.Simplifica pular regiões mascaradas e valores NaN. Agiliza ininterruptamente mesmo ao exceder o limite de linhas dentro de uma única célula de grade.



Pablo Ricardo de abreu
10/2022

Exemplo de código para gráficos de fluxo



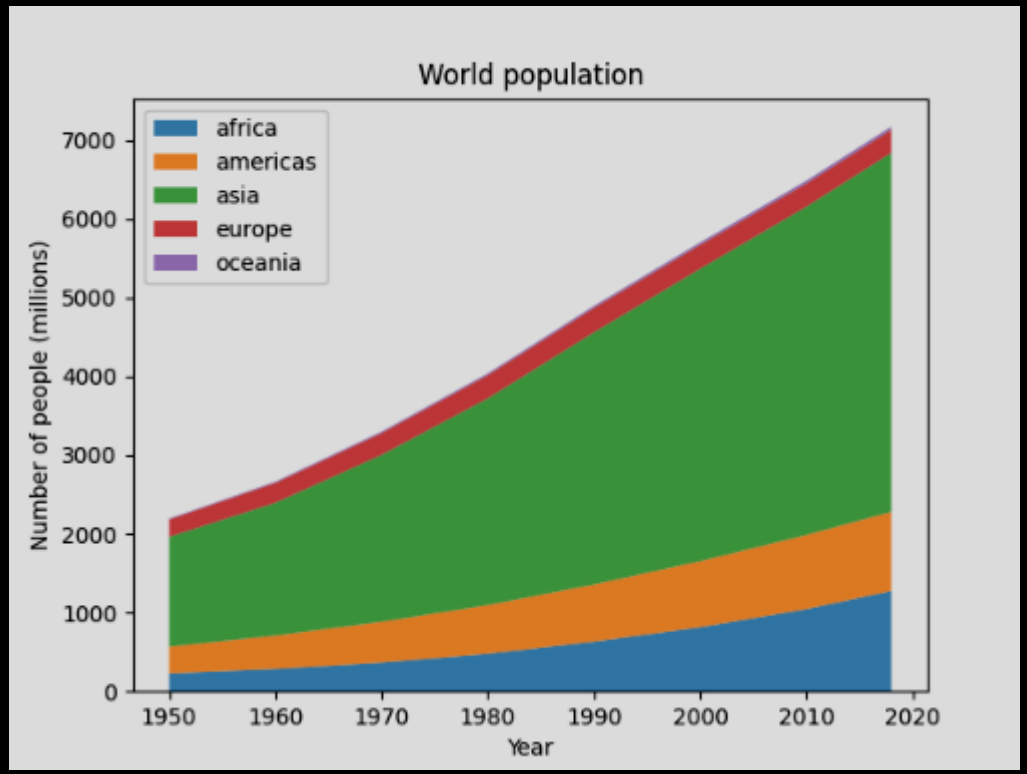

```python
import numpy as np
import matplotlib.pyplot as plt


w = 3
Y, X = np.mgrid[-w:w:100j, -w:w:100j]
U = -1 - X**2 + Y
V = 1 + X - Y**2
speed = np.sqrt(U**2 + V**2)

fig, axs = plt.subplots(3, 2, figsize=(7, 9), height_ratios=[1, 1, 2])
axs = axs.flat

#  Varying density along a streamline
axs[0].streamplot(X, Y, U, V, density=[0.5, 1])
axs[0].set_title('Varying Density')

# Varying color along a streamline
strm = axs[1].streamplot(X, Y, U, V, color=U, linewidth=2, cmap='autumn')
fig.colorbar(strm.lines)
axs[1].set_title('Varying Color')

#  Varying line width along a streamline
lw = 5*speed / speed.max()
axs[2].streamplot(X, Y, U, V, density=0.6, color='k', linewidth=lw)
axs[2].set_title('Varying Line Width')

# Controlling the starting points of the streamlines
seed_points = np.array([[-2, -1, 0, 1, 2, -1], [-2, -1,  0, 1, 2, 2]])

strm = axs[3].streamplot(X, Y, U, V, color=U, linewidth=2,
                         cmap='autumn', start_points=seed_points.T)
fig.colorbar(strm.lines)
axs[3].set_title('Controlling Starting Points')

# Displaying the starting points with blue symbols.
axs[3].plot(seed_points[0], seed_points[1], 'bo')
axs[3].set(xlim=(-w, w), ylim=(-w, w))

# Create a mask
mask = np.zeros(U.shape, dtype=bool)
mask[40:60, 40:60] = True
U[:20, :20] = np.nan
U = np.ma.array(U, mask=mask)

axs[4].streamplot(X, Y, U, V, color='r')
axs[4].set_title('Streamplot with Masking')

axs[4].imshow(~mask, extent=(-w, w, -w, w), alpha=0.5, cmap='gray',
              aspect='auto')
axs[4].set_aspect('equal')

axs[5].streamplot(X, Y, U, V, broken_streamlines=False)
axs[5].set_title('Streamplot with unbroken streamlines')

plt.tight_layout()
plt.show()
```

Pablo Ricardo de abreu
10/2022

Quais os principais gráficos ?

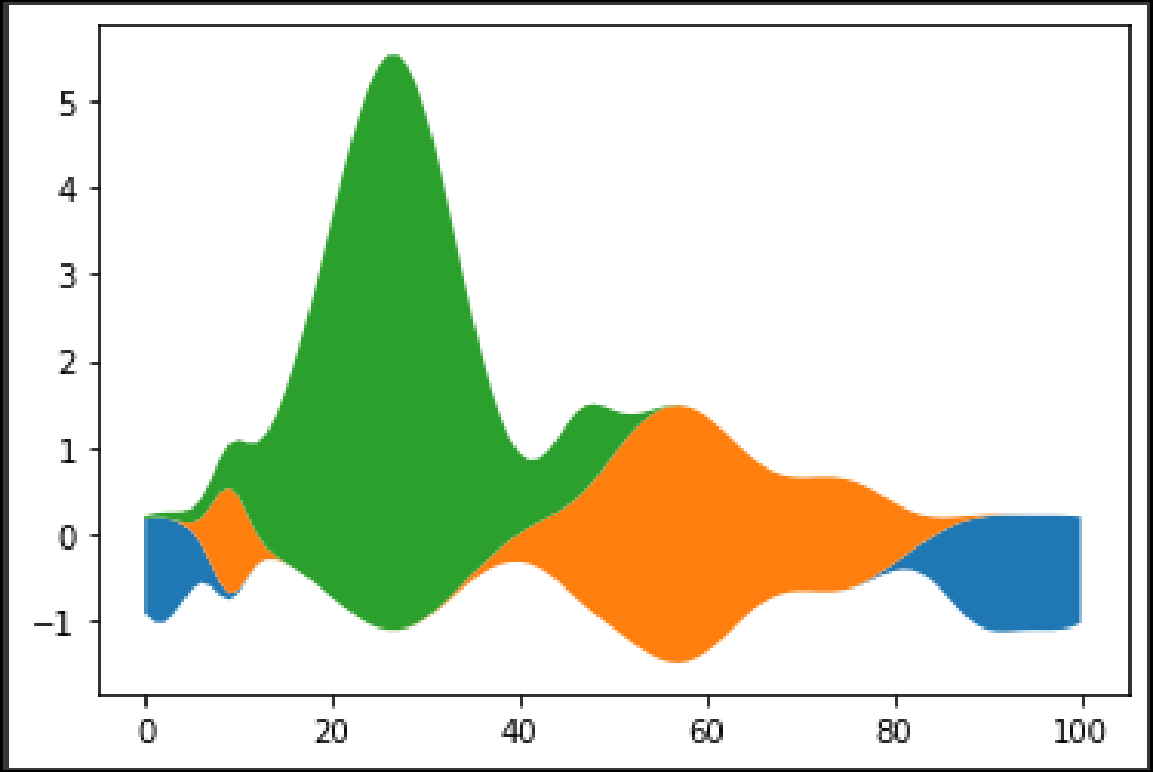
- Gráfico de Áreas.

- Stackplots desenham vários conjuntos de dados como áreas empilhadas verticalmente. Isso é útil quando os valores de dados individuais e adicionalmente seu valor cumulativo são de interesse.



Pablo Ricardo de abreu
10/2022

Exemplo de código para gráficos de áreas

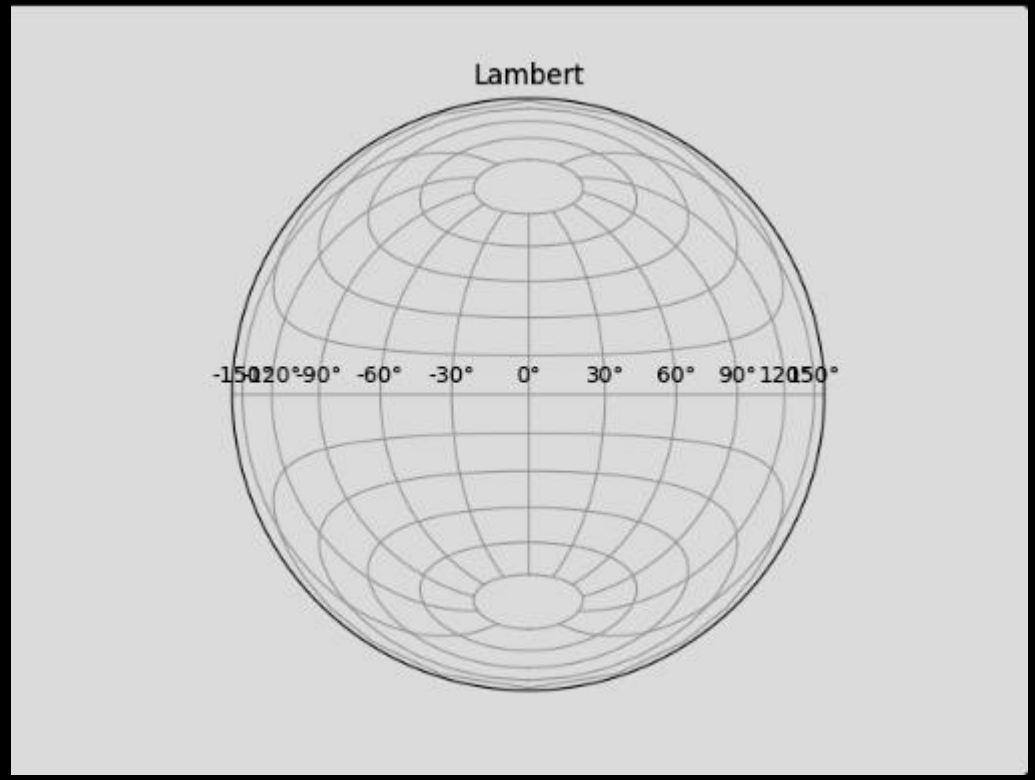



```python
# Fixing random state for reproducibility
np.random.seed(19680801)


def gaussian_mixture(x, n=5):
    """Return a random mixture of *n* Gaussians, evaluated at positions *x*."""
    def add_random_gaussian(a):
        amplitude = 1 / (.1 + np.random.random())
        dx = x[-1] - x[0]
        x0 = (2 * np.random.random() - .5) * dx
        z = 10 / (.1 + np.random.random()) / dx
        a += amplitude * np.exp(-(z * (x - x0))**2)
    a = np.zeros_like(x)
    for j in range(n):
        add_random_gaussian(a)
    return a


x = np.linspace(0, 100, 101)
ys = [gaussian_mixture(x) for _ in range(3)]

fig, ax = plt.subplots()
ax.stackplot(x, ys, baseline='wiggle')
plt.show()
```

Pablo Ricardo de abreu
10/2022

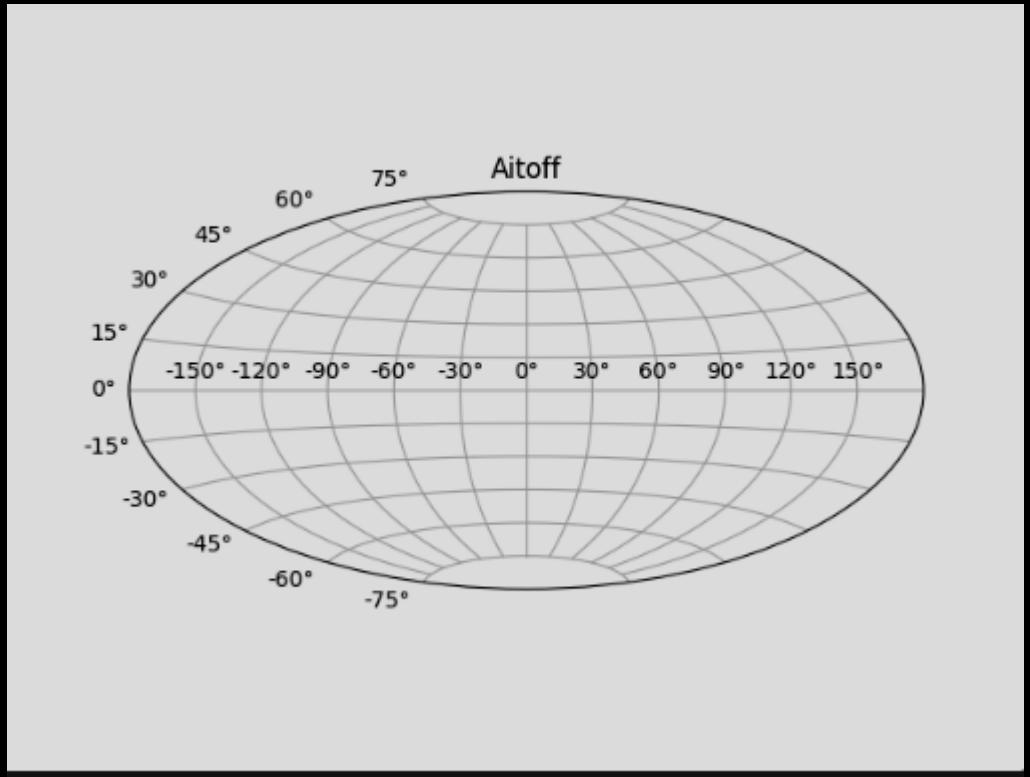
Quais os principais gráficos ?

- Gráfico de projeções geográficas.
- Mapeia as coordenadas que ajudam na cartografia de geolocalização.



Pablo Ricardo de abreu
10/2022

Exemplo de código para gráficos projeções geograficas



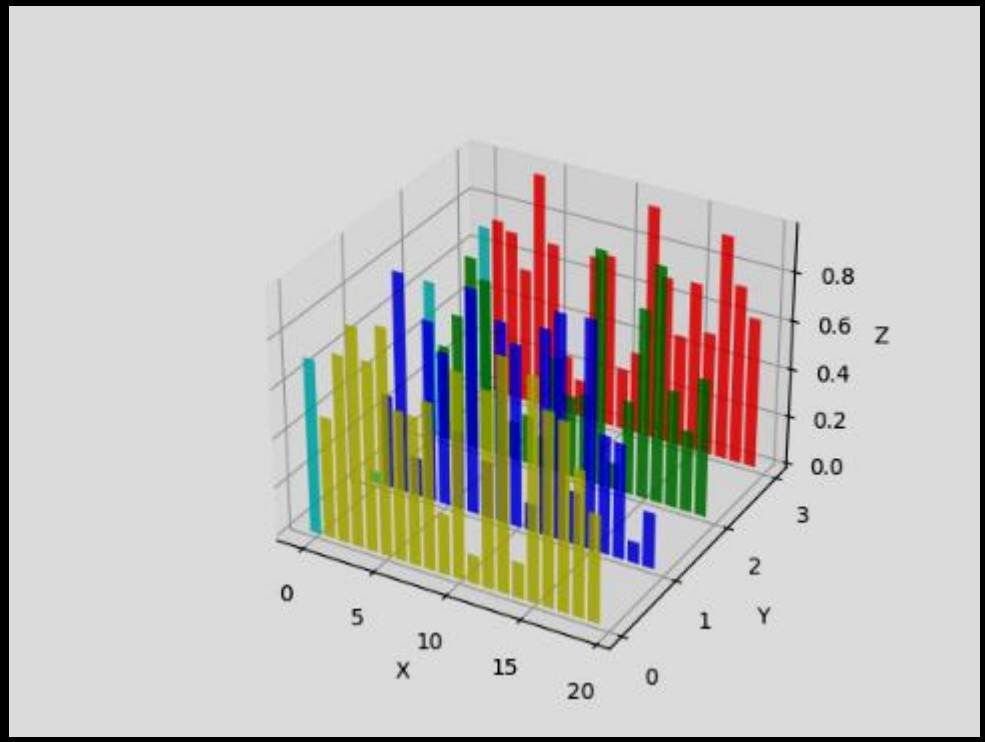

```python
import matplotlib.pyplot as plt
```

```python
plt.figure()
plt.subplot(projection="aitoff")
plt.title("Aitoff")
plt.grid(True)
```

Pablo Ricardo de abreu
10/2022

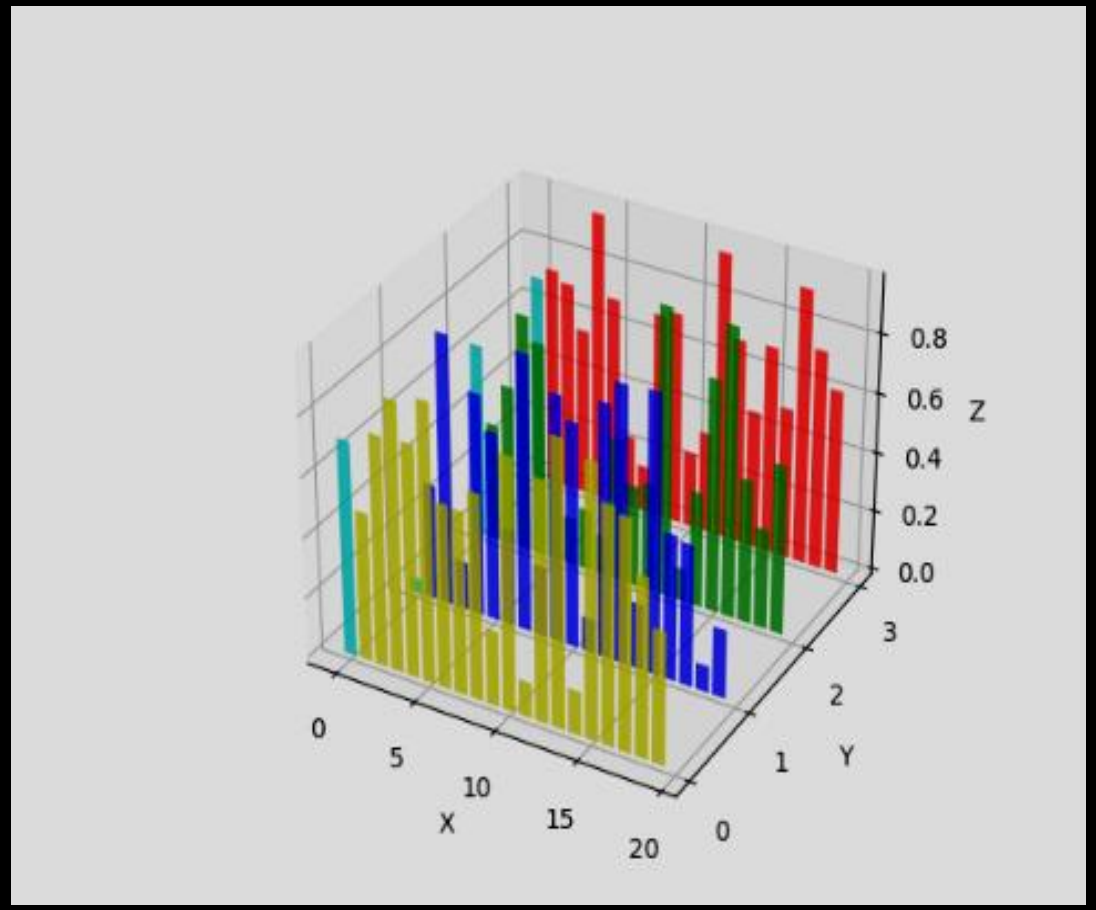
Quais os principais gráficos ?

- Gráfico de projeções de planos.
- Conseguimos demonstrar vários planos e comparações de vários outros parâmetros tudo isso em um único gráfico em 3D.



Pablo Ricardo de abreu
10/2022

Exemplo de código para gráficos projeções de planos

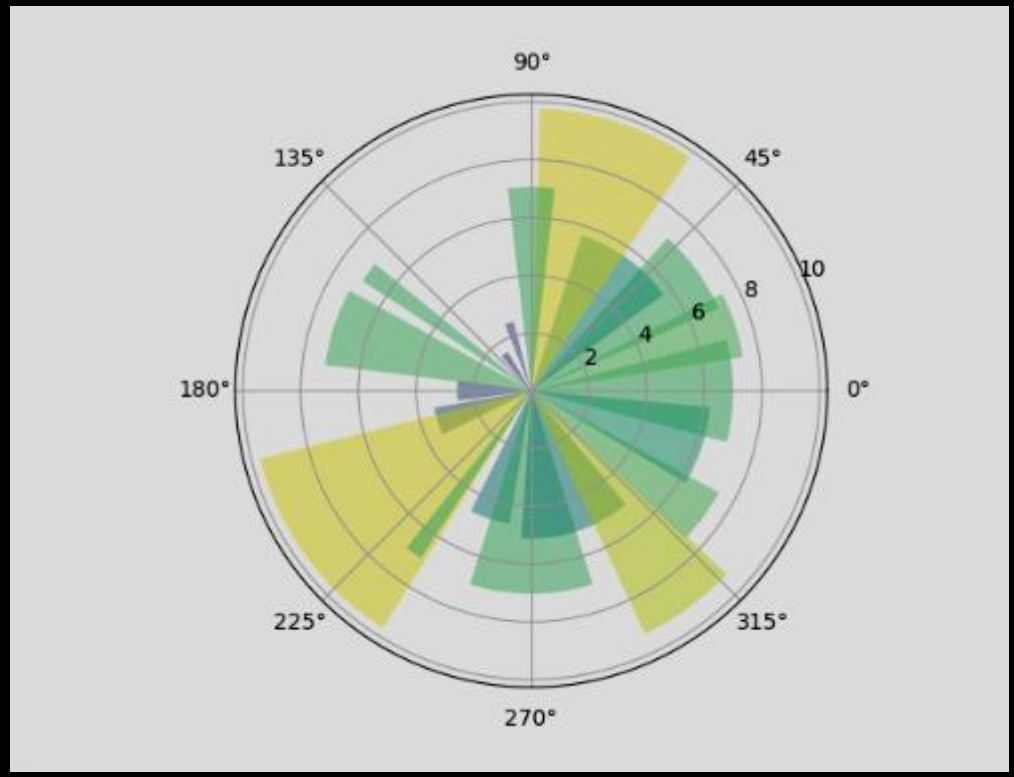



```python
import matplotlib.pyplot as plt
import numpy as np

# Fixing random state for reproducibility
np.random.seed(19680801)


fig = plt.figure()
ax = fig.add_subplot(projection='3d')

colors = ['r', 'g', 'b', 'y']
yticks = [3, 2, 1, 0]
for c, k in zip(colors, yticks):
    # Generate the random data for the y=k 'layer'.
    xs = np.arange(20)
    ys = np.random.rand(20)

    # You can provide either a single color or an array with the same length as
    # xs and ys. To demonstrate this, we color the first bar of each set cyan.
    cs = [c] * len(xs)
    cs[0] = 'c'

    # Plot the bar graph given by xs and ys on the plane y=k with 80% opacity.
    ax.bar(xs, ys, zs=k, zdir='y', color=cs, alpha=0.8)

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

# On the y axis let's only label the discrete values that we have data for.
ax.set_yticks(yticks)

plt.show()
```

Pablo Ricardo de abreu
10/2022

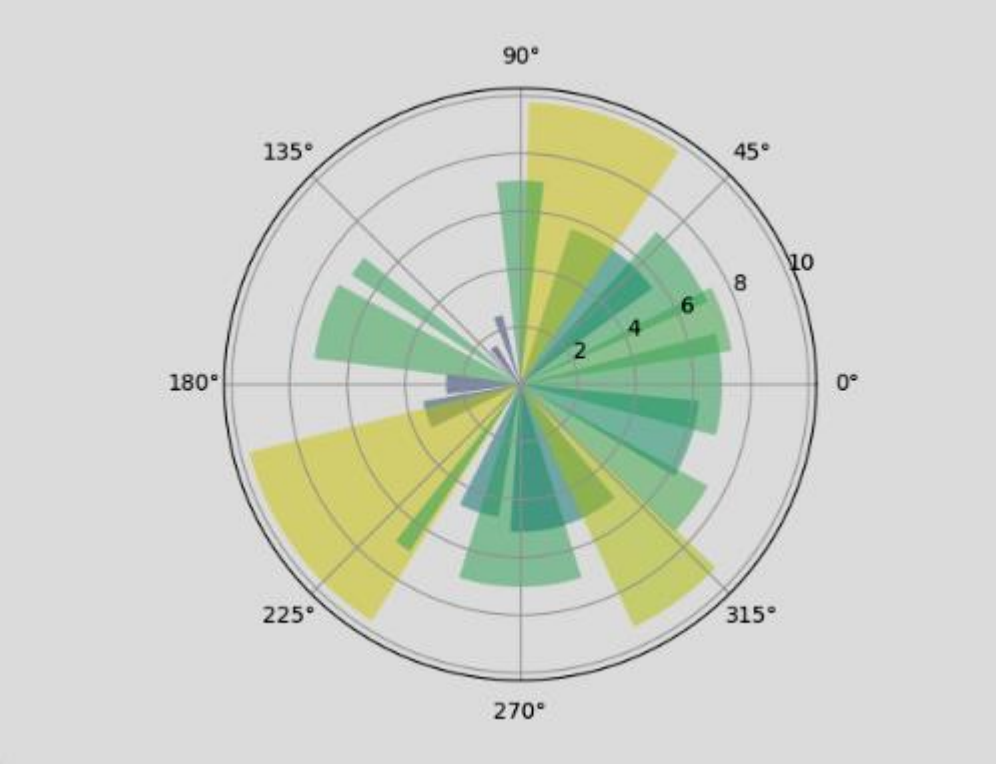
Quais os principais gráficos ?

- **Gráfico Radar.**
- **Em forma de um grade circular as plotagem dos valores são feitos em ângulos e cada raio representa um gradiente da variável.**



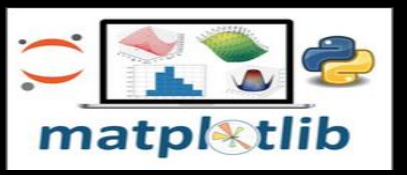
Pablo Ricardo de abreu
10/2022

Exemplo de código para gráficos projeções geograficas




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


# Fixing random state for reproducibility
np.random.seed(19680801)

# Compute pie slices
N = 20
theta = np.linspace(0.0, 2 * np.pi, N, endpoint=False)
radii = 10 * np.random.rand(N)
width = np.pi / 4 * np.random.rand(N)
colors = plt.cm.viridis(radii / 10.)

ax = plt.subplot(projection='polar')
ax.bar(theta, radii, width=width, bottom=0.0, color=colors, alpha=0.5)

plt.show()
```

Pablo Ricardo de abreu
10/2022



Obrigado!!!

Pablo Ricardo de Abreu
Github:https://github.com/Pabloabreu1277