

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего  
образования  
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра  
инфокоммуникаций  
Институт цифрового  
развития**

**ОТЧЁТ**  
**по лабораторной работе №3.5**  
Дисциплина: «Визуализация  
данных с помощью matplotlib»

Выполнила: студентка 2  
курса группы Пиж-б-о-21-  
1  
Джолдошова Мээрим  
Бекболотовна

Ставрополь 2023  
Цель работы: исследовать базовые возможности визуализации данных на  
плоскости средствами библиотеки matplotlib языка программирования Python.

1. Проработать примеры лабораторной работы в отдельном ноутбуке.

Select items to perform actions on them.



Рисунок 1 – Проработанные примеры лабораторной работы

2. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения линейного графика, условие которой предварительно необходимо согласовать с преподавателем.



Рисунок 2 – Выполненное 1 индивидуальное задание

3. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения столбчатой диаграммы, условие которой предварительно необходимо согласовать с преподавателем.



Рисунок 3 – Выполненное 2 индивидуальное задание

4. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.) требующей построения круговой диаграммы, условие которой предварительно необходимо согласовать с преподавателем.

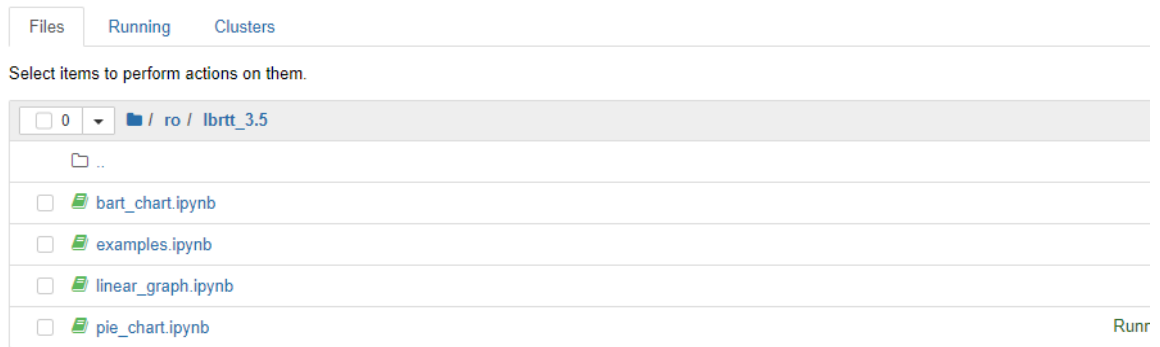


Рисунок 4 – Выполненное 3 индивидуальное задание

5. Найти какое-либо изображение в сети Интернет. Создать ноутбук, в котором будет отображено выбранное изображение средствами библиотеки matplotlib по URL из сети Интернет.



Рисунок 5 – Выполненное 4 индивидуальное задание

### Вопросы для защиты работы

1. Как выполнить построение линейного графика с помощью matplotlib?

`plot([x], y, [fmt], *, data=None, **kwargs)`

`plot([x], y, [fmt], [x2], y2, [fmt2], ..., **kwargs)`

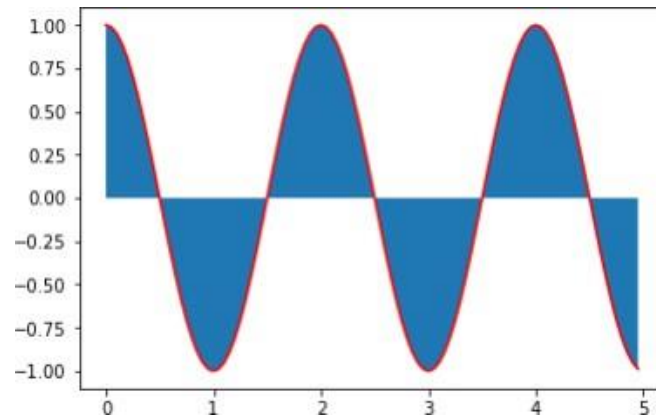
2. Как выполнить заливку области между графиком и осью? Между двумя графиками?

`x = np.arange(0.0, 5, 0.01)`

`y = np.cos(x*np.pi)`

```
plt.plot(x, y, c = "r")
```

```
plt.fill_between(x, y)
```



3. Как выполнить выборочную заливку, которая удовлетворяет некоторому условию?

Заливка области между 0.5 и y, при условии, что  $y \geq 0.5$ :

```
plt.plot(x, y, c="r")
```

```
plt.grid()
```

```
plt.fill_between(x, 0.5, y, where=(y>=0.5))
```

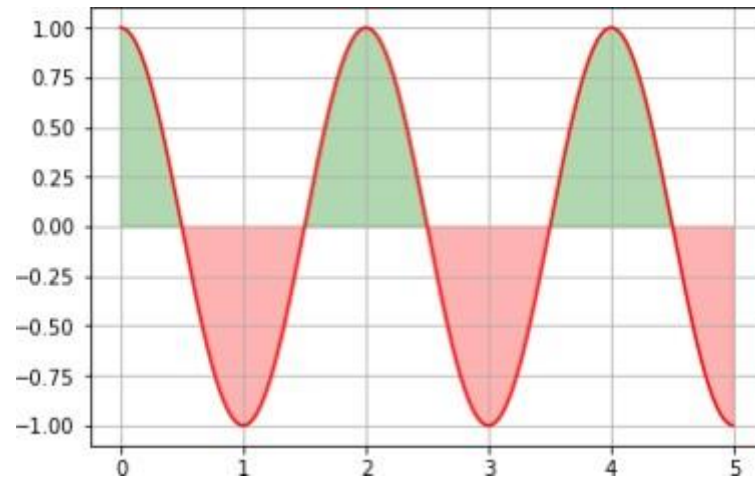
4. Как выполнить двухцветную заливку?

```
plt.plot(x, y, c="r")
```

```
plt.grid()
```

```
plt.fill_between(x, y, where=y>=0, color="g", alpha=0.3)
```

```
plt.fill_between(x, y, where=y<=0, color="r", alpha=0.3)
```



## 5. Как выполнить маркировку графиков?

```
plt.plot(x, y, marker="o", c="g")
```

Типы маркеров:

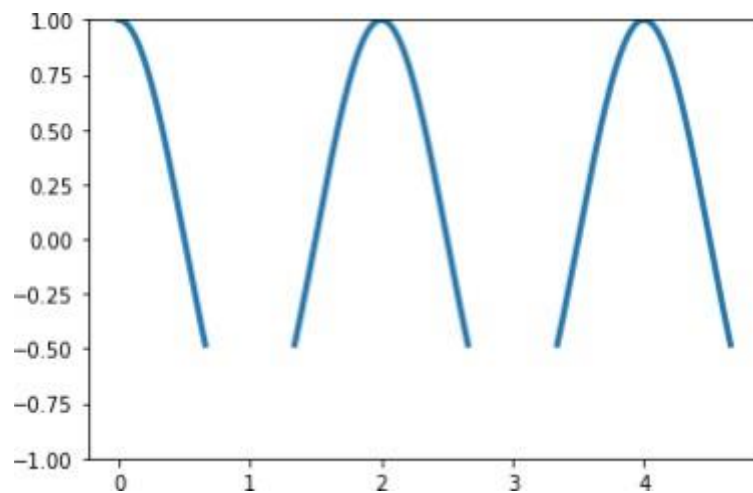
Символ	Описание
'.'	Точка ( <i>point marker</i> )
','	Пиксель ( <i>pixel marker</i> )
'o'	Окружность ( <i>circle marker</i> )
'v'	Треугольник, направленный вниз ( <i>triangle_down marker</i> )
'^'	Треугольник, направленный вверх ( <i>triangle_up marker</i> )
'<'	Треугольник, направленный влево ( <i>triangle_left marker</i> )
'>'	Треугольник, направленный вправо ( <i>triangle_right marker</i> )
'1'	Треугольник, направленный вниз ( <i>tri_down marker</i> )
'2'	Треугольник, направленный вверх ( <i>tri_up marker</i> )
'3'	Треугольник, направленный влево ( <i>tri_left marker</i> )
'4'	Треугольник, направленный вправо ( <i>tri_right marker</i> )
's'	Квадрат ( <i>square marker</i> )
'p'	Пятиугольник ( <i>pentagon marker</i> )
'*'	Звезда ( <i>star marker</i> )
'h'	Шестиугольник ( <i>hexagon1 marker</i> )
'H'	Шестиугольник ( <i>hexagon2 marker</i> )
'+'	Плюс ( <i>plus marker</i> )
'x'	Х-образный маркер ( <i>x marker</i> )
'D'	Ромб ( <i>diamond marker</i> )
'd'	Ромб ( <i>thin_diamond marker</i> )
' '	Вертикальная линия ( <i>vline marker</i> )
'_'	Горизонтальная линия ( <i>hline marker</i> )

## 6. Как выполнить обрезку графиков?

Для того, чтобы отобразить только часть графика, которая отвечает определенному условию используйте предварительное маскирование данных с помощью функции `masked_where` из пакета `numpy`.

```
x = np.arange(0.0, 5, 0.01)
y = np.cos(x * np.pi)
y_masked = np.ma.masked_where(y < -0.5, y)
plt.ylim(-1, 1)

plt.plot(x, y_masked, linewidth=3)
```



## 7. Как построить ступенчатый график? В чем особенность ступенчатого графика?

Такой график строится с помощью функции `step()`, которая принимает следующий набор параметров:

- `x`: `array_like` - набор данных для оси абсцисс
- `y`: `array_like` - набор данных для оси ординат
- `fmt`: `str`, optional - задает отображение линии (см. функцию `plot()`).
- `data`: `indexable object`, optional - метки.
- `where`: `{'pre', 'post', 'mid'}`, optional, по умолчанию `'pre'` - определяет место, где будет установлен шаг.

‘pre’: значение  $y$  ставится слева от значения  $x$ , т.е. значение  $y[i]$  определяется для интервала  $(x[i-1]; x[i])$ .

‘post’: значение  $y$  ставится справа от значения  $x$ , т.е. значение  $y[i]$  определяется для интервала  $(x[i]; x[i+1])$ .

‘mid’: значение  $y$  ставится в середине интервала.

Код:

```
x = np.arange(0, 7)
```

```
y = x
```

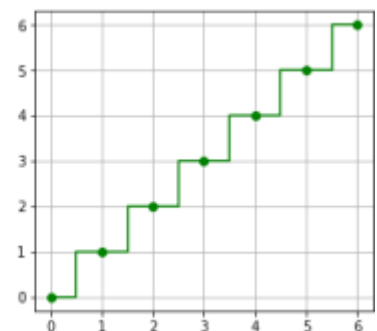
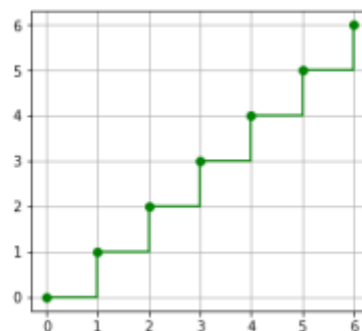
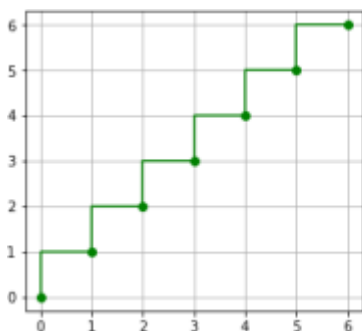
```
where_set = ['pre', 'post', 'mid']
```

```
fig, axs = plt.subplots(1, 3, figsize=(15, 4))
```

```
for i, ax in enumerate(axs):
```

```
    ax.step(x, y, "g-o", where=where_set[i])
```

```
    ax.grid()
```



8. Как построить стековый график? В чем особенность стекового графика?

Для построения стекового графика используется функция `stackplot()`. Суть его в том, что графики отображаются друг над другом, и каждый следующий является суммой предыдущего и заданного набора данных:

```
x = np.arange(0, 11, 1)
```

```
y1 = np.array([(-0.2)*i**2+2*i for i in x])
```

```
y2 = np.array([(-0.4)*i**2+4*i for i in x])
```

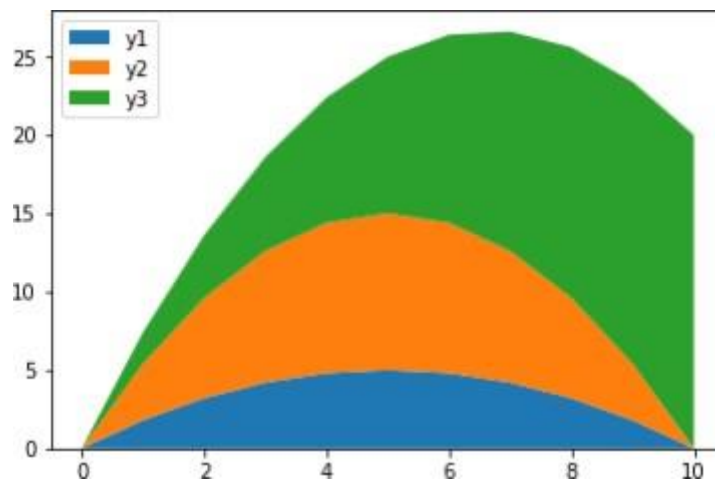
```
y3 = np.array([2*i for i in x])
```

```
labels = ["y1", "y2", "y3"]
```

```
fig, ax = plt.subplots()
```

```
ax.stackplot(x, y1, y2, y3, labels=labels)
```

```
ax.legend(loc='upper left')
```



9. Как построить stem-график? В чем особенность stem-графика?

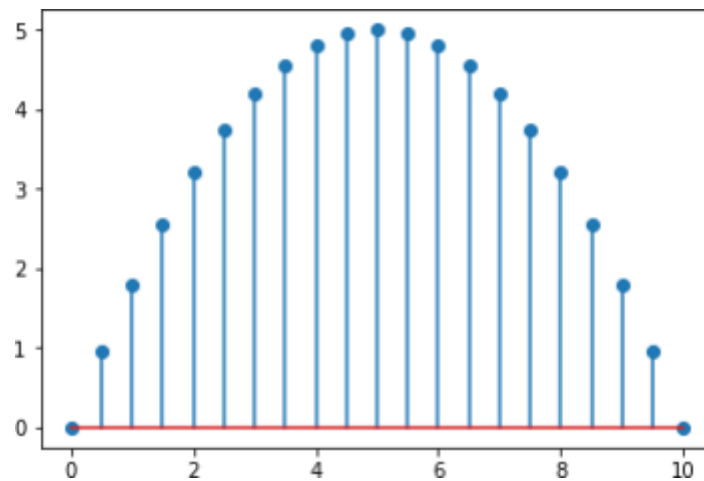
Визуально этот график выглядит как набор линий от точки с координатами (x, y) до базовой линии, в верхней точке ставится маркер.

```
x = np.arange(0, 10.5, 0.5)
```

```
y = np.array([(-0.2)*i**2+2*i for i in x])
```

```
plt.stem(x, y)
```





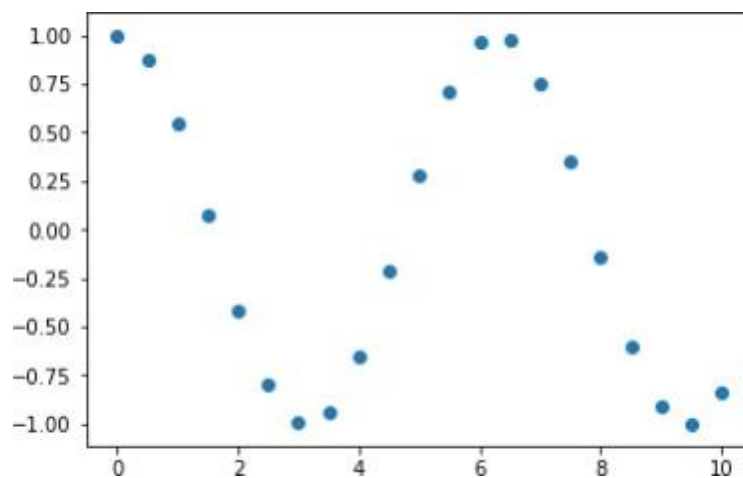
10. Как построить точечный график? В чем особенность точечного графика?

Для отображения точечного графика предназначена функция `scatter()`. В простейшем виде точечный график можно получить передав функции `scatter()` наборы точек для  $x$ ,  $y$  координат:

```
x = np.arange(0, 10.5, 0.5)
```

```
y = np.cos(x)
```

```
plt.scatter(x, y)
```



11. Как осуществляется построение столбчатых диаграмм с помощью matplotlib?

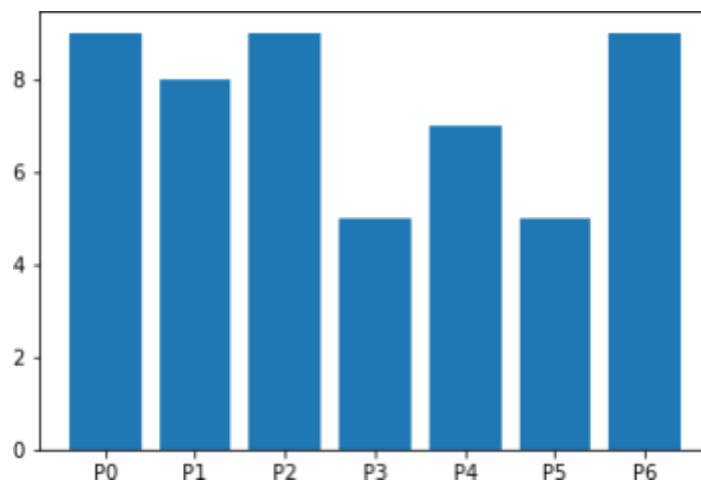
Для их построения используются функции: `bar()` – для построения вертикальной диаграммы `barh()` – для построения горизонтальной диаграммы.

```
np.random.seed(123)
```

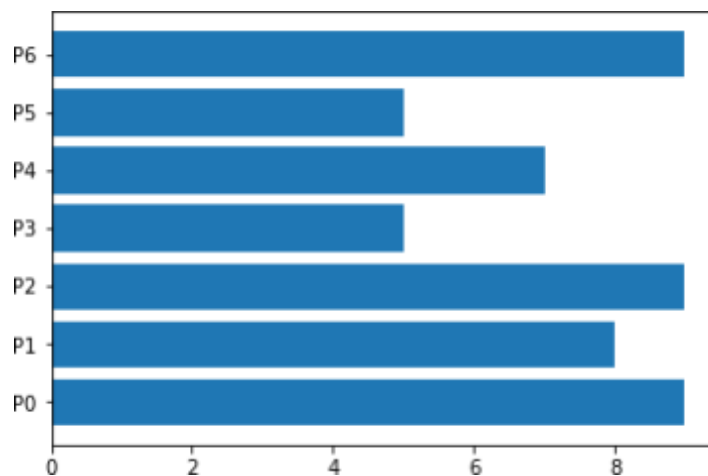
```
groups = [f"P{i}" for i in range(7)]
```

```
counts = np.random.randint(3, 10, len(groups))
```

```
plt.bar(groups, counts)
```



Если заменим `bar()` на `barh()` получим горизонтальную диаграмму:



12. Что такое групповая столбчатая диаграмма? Что такое столбчатая диаграмма с errorbar элементом?

Это столбчатая диаграмма, которая используется для одновременного отображения больших или нескольких наборов данных.

```
cat_par = [f"P{i}" for i in range(5)]
```

```
g1 = [10, 21, 34, 12, 27]
```

```
g2 = [17, 15, 25, 21, 26]
```

```
width = 0.3
```

```
x = np.arange(len(cat_par))
```

```
fig, ax = plt.subplots()
```

```
rects1 = ax.bar(x - width/2, g1, width, label='g1')
```

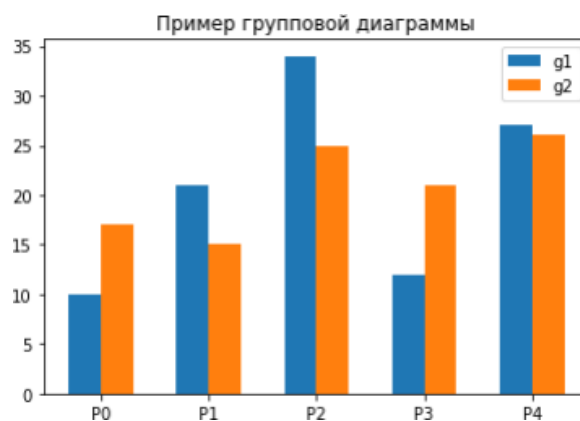
```
rects2 = ax.bar(x + width/2, g2, width, label='g2')
```

```
ax.set_title('Пример групповой диаграммы')
```

```
ax.set_xticks(x)
```

```
ax.set_xticklabels(cat_par)
```

```
ax.legend()
```



13. Как выполнить построение круговой диаграммы средствами matplotlib?

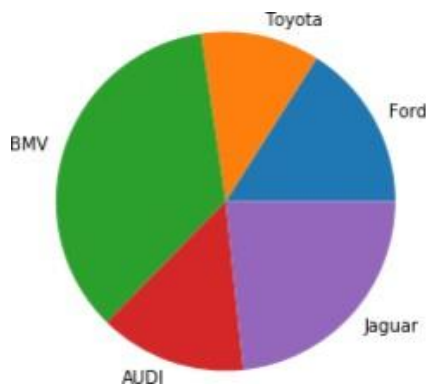
```
vals = [24, 17, 53, 21, 35]
```

```
labels = ["Ford", "Toyota", "BMV", "AUDI", "Jaguar"]
```

```
fig, ax = plt.subplots()
```

```
ax.pie(vals, labels=labels)
```

```
ax.axis("equal")
```



14. Что такое цветовая карта? Как осуществляется работа с цветовыми картами в matplotlib?

Цветовая карта представляет собой подготовленный набор цветов, который хорошо подходит для визуализации того или иного набора данных.

```
fig, axs = plt.subplots(1, 2, figsize=(10,3), constrained_layout=True)
```

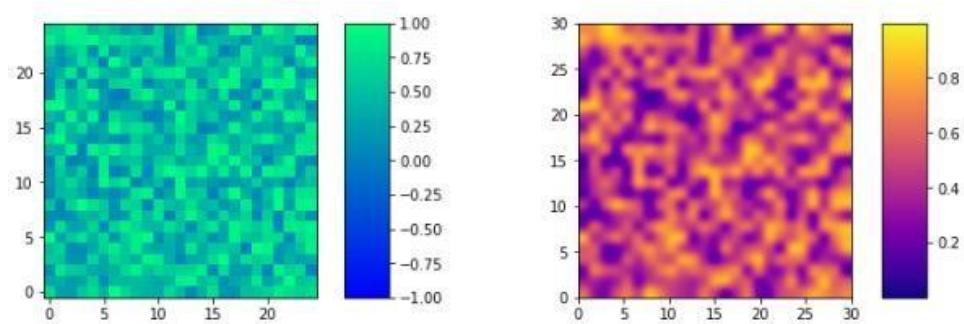
```
p1 = axs[0].imshow(data, cmap='winter', aspect='equal', vmin=-1, vmax=1, origin="lower")
```

```
fig.colorbar(p1, ax=axs[0])
```

```
p2 = axs[1].imshow(data, cmap='plasma', aspect='equal',
```

```
interpolation='gaussian', origin="lower", extent=(0, 30, 0, 30))
```

```
fig.colorbar(p2, ax=axs[1])
```



15. Как отобразить изображение средствами

matplotlib?from PIL import Image

import requests

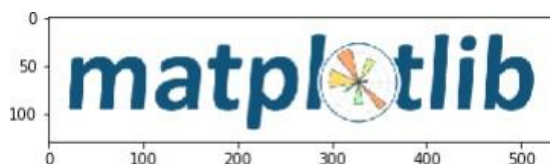
from io import BytesIO

response =

requests.get('https://matplotlib.org/\_static/logo2.png')img =

Image.open(BytesIO(response.content))

plt.imshow(img)



16. Как отобразить тепловую карту средствами

matplotlib?np.random.seed(123)

data = np.random.rand(5, 7)

plt.pcolormesh(data, cmap='plasma', edgecolors="k", shading='flat')

