

## Plotting with Python

```
In [1]: import abc
import jieba
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from wordcloud import WordCloud
from PIL import Image
from pathlib import Path
```

```
In [2]: class Plotter(metaclass=abc.ABCMeta):
    @abc.abstractmethod
    def plot(data, *args, **kwargs):
        pass
```

```
In [3]: class Plotter(metaclass=abc.ABCMeta):
    @abc.abstractmethod
    def plot(data, *args, **kwargs):
        pass

    class PointPlotter(Plotter):
        def plot(self, points):
            plt.plot(points, 'bo')
            plt.show()

    class ArrayPlotter(Plotter):
        def plot(self, array):
            if len(array) == 2:
                plt.plot(*array)
            elif len(array) == 3:
                fig = plt.figure()
                ax = Axes3D(fig)
                ax.plot3D(*array)
            plt.show()

    class TextPlotter(Plotter):
        def __init__(self, text):
            self.text = text

        def plot(self):
            with open(r'stopwords.txt', encoding='utf-8') as f:
                stop = f.read().split('\n')
            tokens = self.text.iloc[:, 0].apply(
                lambda x: [i for i in jieba.lcut(x) if i not in stop], 1)
            tokens = tokens.loc[tokens.isin([]) == False]
            all_words = []
            for i in tokens:
                all_words.extend(i)
            count = pd.Series(all_words).value_counts()
            wc = WordCloud(font_path='simhei.ttf', background_color='white')
            wc = wc.fit_words(count)
            plt.imshow(wc)
            plt.axis('off')
            plt.show()

    class ImagePlotter(Plotter):
        def __init__(self, image_path, format = "png"):
            self.image_path = image_path
```

```

        image_path = Path(image_path)
        self.images = [] # 列表images存储图片
        if image_path.is_file(): # 图片文件
            self.images.append(Image.open(image_path))
            self.format = image_path.suffix[1:]
        elif image_path.is_dir(): # 图片目录
            for file in image_path.glob("*. "+format):
                self.images.append(Image.open(file))

    def plot(self, row = 1, col = 1):
        for i in range(len(self.images)):
            plt.subplot(row, col, i+1)
            plt.imshow(self.images[i])

class GifPlotter(Plotter):
    def __init__(self, image_path, format = "png"):
        self.image_path = image_path
        image_path = Path(image_path)
        self.images = [] # 列表images存储图片
        if image_path.is_file(): # 图片文件
            self.images.append(Image.open(image_path))
            self.format = image_path.suffix[1:]
        elif image_path.is_dir(): # 图片目录
            for file in image_path.glob("*. "+format):
                self.images.append(Image.open(file))

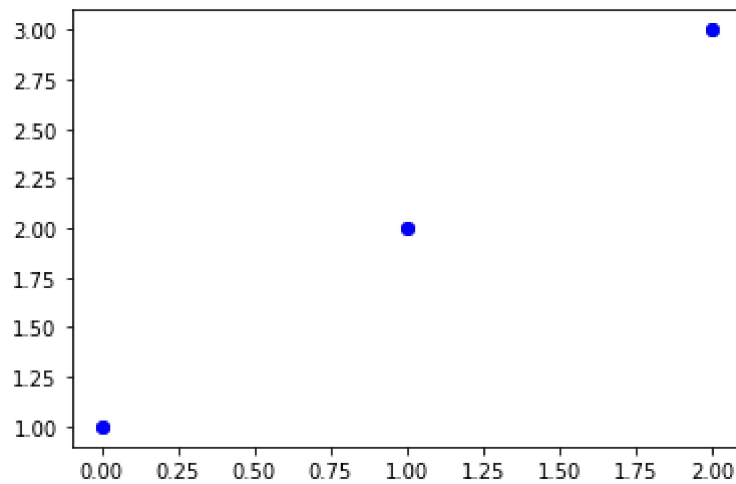
    def plot(self, save_path, duration = 0.5):
        self.images[0].save(save_path, save_all=True,
                            append_images=self.images[1:], duration=duration)

```

```

In [4]: point_plot = PointPlotter()
        point_plot.plot([(1,1), (2,2), (3, 3)])

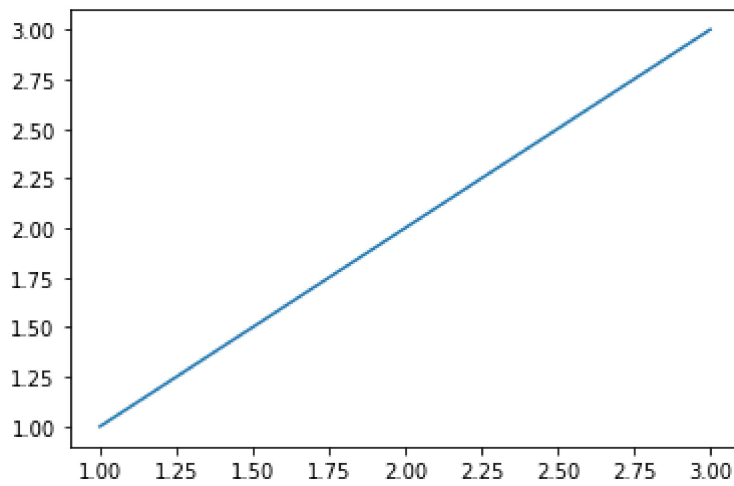
```



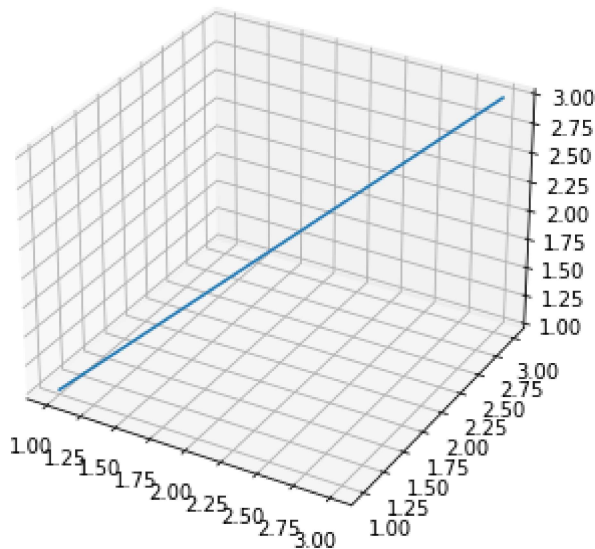
```

In [5]: array_plot = ArrayPlotter()
        array_plot.plot([(1, 2, 3), (1, 2, 3)]) # 2 Dimension
        array_plot.plot([(1, 2, 3), (1, 2, 3), (1, 2, 3)]) # 3 Dimension

```



```
<ipython-input-3-ae552c7ddd5f>:17: MatplotlibDeprecationWarning: Axes3D(fig) adding
itself to the figure is deprecated since 3.4. Pass the keyword argument auto_add_to_
figure=False and use fig.add_axes(ax) to suppress this warning. The default value of
auto_add_to_figure will change to False in mpl3.5 and True values will no longer wor
k in 3.6. This is consistent with other Axes classes.
... ax = Axes3D(fig)
```

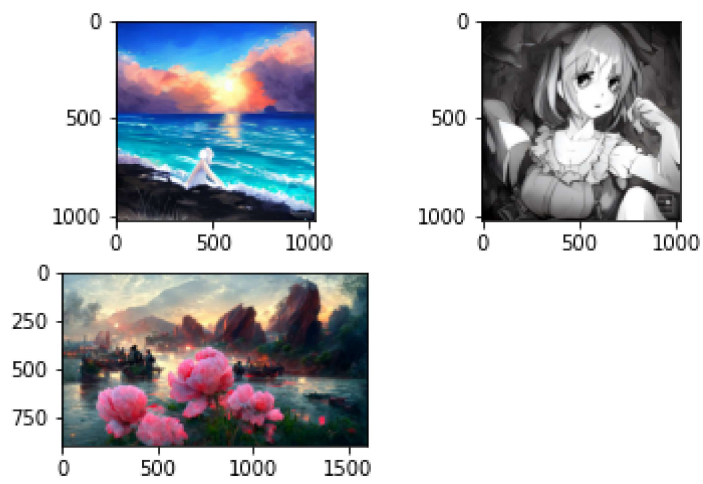


```
In [6]: text_plot = TextPlotter(pd.read_csv('comments.csv'))
text_plot.plot()
```

```
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\86198\AppData\Local\Temp\jieba.cache
Loading model cost 0.963 seconds.
Prefix dict has been built successfully.
```



```
In [7]: image_plot = ImagePlotter('.')
image_plot.plot(2, 2)
```



```
In [8]: gif_plot = GifPlotter('gif_data/')  
gif_plot.plot('gif_plot.gif')
```

