

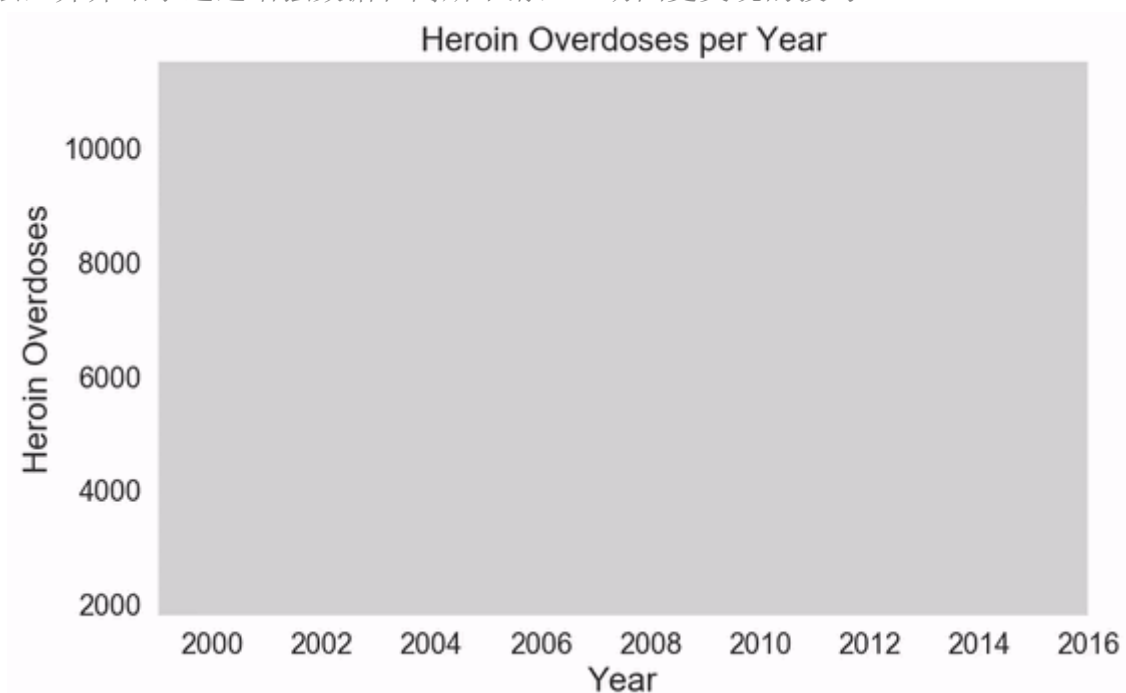
matplotlib秘技：让可视化图形动起来

V. Kakerbeck

来源：论智

未经允许，禁止二次转载

编者按：其实matplotlib有一个少有人知的功能`animation.FuncAnimation`，可以接受你编写的动画函数创建动图。Viviane Kakerbeck通过一个例子展示了这一功能的用法，并介绍了通过增强数据和高斯平滑，让动图更美观的技巧。



美国的过量服用海洛因致死数，使用seaborn创建

Python的matplotlib和seaborn是非常好用的绘图库。但它们创建的都是静态图像，难以通过动态、美观的方式描述数据值的变化。如果你的下一次演示或者下一篇博客文章，能用动态图形展示数据的发展，该有多好？更妙的是，你可以继续使用matplotlib、seaborn或者其他你喜欢用的库。

我最近为一部关于美国的阿片样物质危机的纪录片制作了一些动态图形，所以我会在这篇文章中使用相关的数据。数据来自美国国家药物滥用研究所和CDC的公开数据，可以从以下网址下载：https://www.drugabuse.gov/sites/default/files/overdose_data_1999-2015.xls

本文将使用matplotlib和seaborn绘制图形，同时使用numpy和pandas处理数据。matplotlib提供了一些可以用来制作动画的函数。闲话少叙，让我们开始吧，首先，是引入所有依赖。

```
1. import numpy as np

2. import pandas as pd

3. import seaborn as sns

4. import matplotlib

5. import matplotlib.pyplot as plt

6. import matplotlib.animation as animation
```

然后我们加载数据，将其转换成pandas的DataFrame。我还编写了一个辅助函数，可以从感兴趣的行加载数据，之后绘图会用到。

```
1. overdoses = pd.read_excel('overdose_data_1999-2015.xls', sheetname='Online', skiprows =6)

2. def get_data(table, rownum, title):

3.     data = pd.DataFrame(table.loc[rownum][2:]).astype(float)

4.     data.columns = {title}

5.     return data
```

准备就绪，下面是本文主要部分，如何绘制动画。

首先，如果你和我一样，用的是jupyter notebook，那么我建议你使用%matplotlib notebook指令，这样可以直接在notebook中查看动画效果，无需等待保存后再查看。

我使用了之前编写的辅助函数`get_data`取得海洛因服用过量数，并将其封装入一个两列的pandas DataFrame，一列表示年份，一列表示服用过量数。

```
1. %matplotlib notebook

2. title = 'Heroin Overdoses'

3. d = get_data(overdoses, 18, title)

4. x = np.array(d.index)

5. y = np.array(d['Heroin Overdoses'])

6. overdose = pd.DataFrame(y, x)

7. #XN, YN = augment(x, y, 10)

8. #augmented = pd.DataFrame(YN, XN)

9. overdose.columns = {title}
```

接着我们初始化一个写入器（writer），基于ffmpeg记录20 fps（比特率为1800）。当然，你可以按照需要调整这些参数。

```
1. Writer = animation.writers['ffmpeg']

2. writer = Writer(fps=20, metadata=dict(artist='Me'), bitrate=1800)
```

接下来创建一个带标签的图形。别忘了限定x轴和y轴的范围，以免动画在显示数据时出现跳跃现象。

```
1. fig = plt.figure(figsize=(10,6))

2. plt.xlim(1999, 2016)

3. plt.ylim(np.min(overdose)[0], np.max(overdose)[0])

4. plt.xlabel('Year', fontsize=20)

5. plt.ylabel(title, fontsize=20)

6. plt.title('Heroin Overdoses per Year', fontsize=20)
```

制作动画的关键是定义一个动画函数，指定视频的每一帧发生了什么。这里*i*表示动画帧的索引。你可以选择在*i*帧中可见的数据范围。之后我使用seaborn的线图绘制选定数据。最后两行我调整了一些尺寸，使图形看起来更美观。

```
1. def animate(i):

2.     data = overdose.iloc[:int(i+1)] # 选定数据范围

3.     p = sns.lineplot(x=data.index, y=data[title], data=data, color="r")

4.     p.tick_params(labelsize=17)

5.     plt.setp(p.lines, linewidth=7)
```

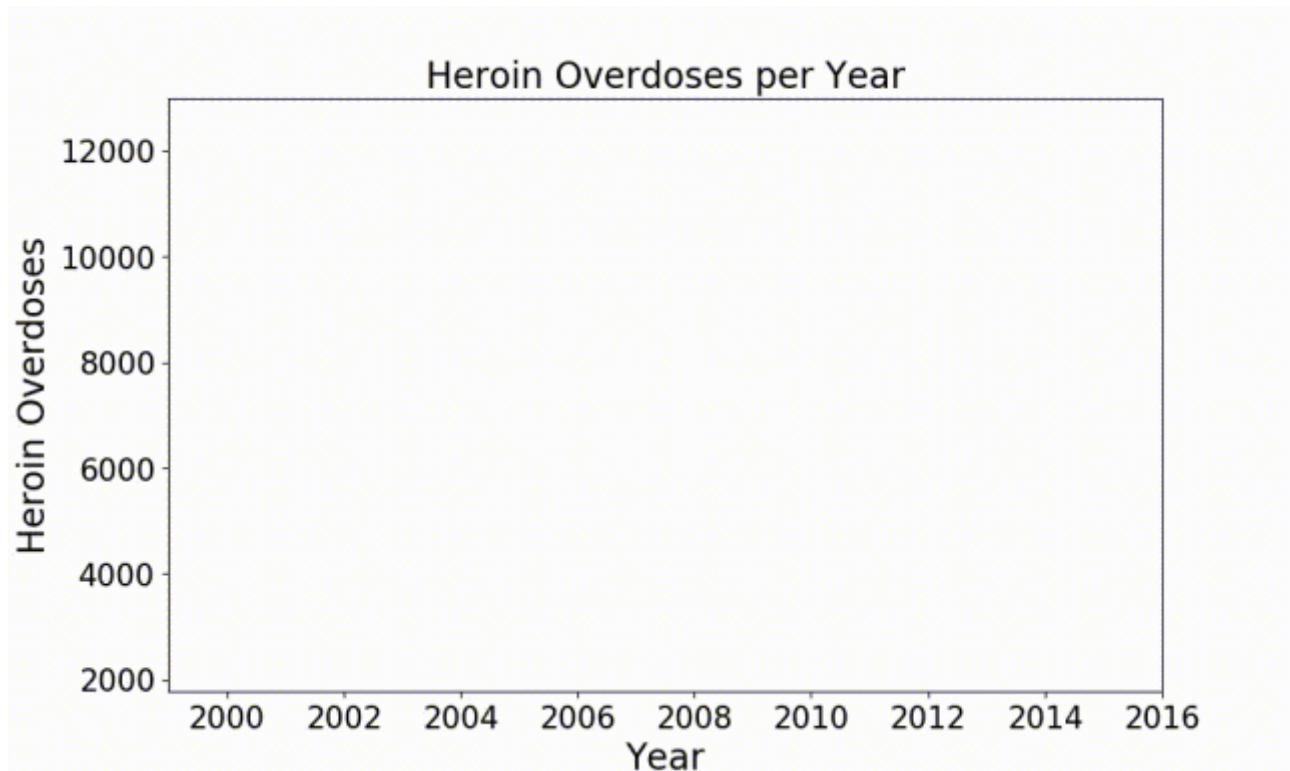
定义了动画函数后，使用matplotlib.animation.FuncAnimation定义动画应当包含多少帧，也就是说，通过frames参数定义调用animate(i)的频率。

```
1. ani = matplotlib.animation.FuncAnimation(fig, animate, frames=17, repeat=True)
```

之后只需调用ani.save()就可以将动画保存为mp4文件。如果你想在保存之前先看下效果，那么可以使用plt.show()。

```
1. ani.save('HeroinOverdosesJumpy.mp4', writer=writer)
```

好了，让我们来看下效果。



看起来效果还可以，但是感觉有点抖动。为了缓解抖动的现象，我们可以在已有数据中插入一些中间值，平滑一下。为此我们定义以下的数据增强函数：

```
1. def augment(xold, yold, numsteps):  
  
2.     xnew = []  
  
3.     ynew = []  
  
4.     for i in range(len(xold)-1):  
  
5.         difX = xold[i+1]-xold[i]  
  
6.         stepsX = difX/numsteps  
  
7.         difY = yold[i+1]-yold[i]
```

```

8.         stepsY = difY/numsteps

9.         for s in range(numsteps):

10.             xnew = np.append(xnew, xold[i]+s*stepsX)

11.             ynew = np.append(ynew, yold[i]+s*stepsY)

12.     return xnew, ynew

```

之后我们只需在数据上应用这一增强函数，并相应地增加matplotlib.animation.FuncAnimation函数中的帧数。这里我调用augment函数时使用了参数numsteps=10，也就是说，我将数据点增加到160个，相应地，帧数设置为frames=160。增强数据之后，结果看起来平滑了很多，但在数据值变动处，曲线仍有一些尖角，看起来不是特别美观。

为了让图像看起来更美观，我们实现了一个高斯平滑函数：

```

1. def smoothListGaussian(listin, strippedXs=False, degree=5):

2.     window=degree*2-1

3.     weight=np.array([1.0]*window)

4.     weightGauss=[]

5.     for i in range(window):

6.         i=i-degree+1

7.         frac=i/float(window)

8.         gauss=1/(np.exp((4*(frac))**2))

9.         weightGauss.append(gauss)

```

```

10.         weight=np.array(weightGauss)*weight

11.         smoothed=[0.0]*(len(listin)-window)

12.         for i in range(len(smoothed)):
            smoothed[i]=sum(np.array(listin[i:i+window])*weight)/sum(weight)

13.         return smoothe

```

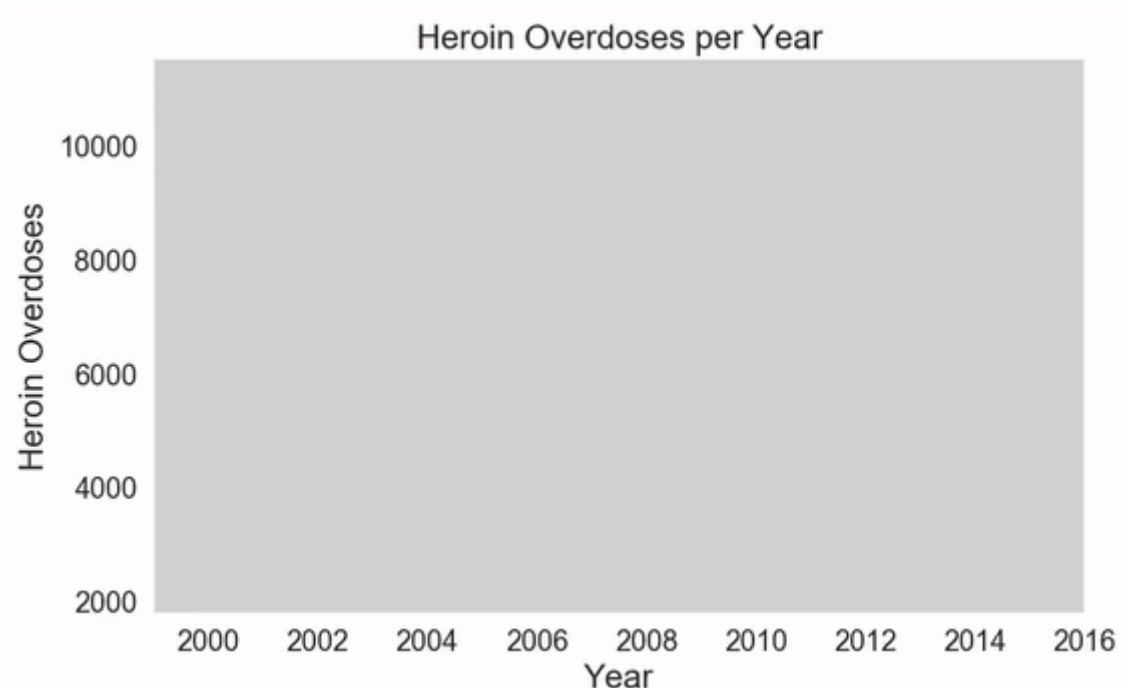
我在这里不会讨论高斯平滑的细节，感兴趣的读者可以看这篇文章：
<https://www.swharden.com/wp/2008-11-17-linear-data-smoothing-in-python/>

最后我们略微调整下颜色和风格，就得到了文章开头的动态图形：

```

1. sns.set(rc={'axes.facecolor':'lightgrey',
'figure.facecolor':'lightgrey','figure.edgecolor':'black','axes.grid':False})

```



本文通过一个例子展现了matplotlib动画函数的用法。当然，你可以将它用在任何你想要动画化的图形上。只需调整animate()函数中的参数和图形类型，便有无限可能。

我希望你喜欢matplotlib的整个功能，并能善加利用。另外，如果你对我之前提到的纪录片感兴趣，可以在YouTube上查看：<https://youtu.be/7xrvuSDLHiY>



历史推荐

[人人都是数据分析师，人人都能玩转Pandas](#) | [Numpy 精品系列教程汇总](#) | [我是如何入门机器学习的呢](#) | [谷歌机器学习43条黄金法则](#)

现在微信改版了

很多人说找不到AI派了

其实只需要[设为星标](#)就可以了

只需[三步](#)，轻松搞定星标设置

一篇让文科生也能读懂机器学习 的文章

原创：王伟同学 [AI派](#) 1周前

1. 点击上方蓝色“AI派”



2. 点击右上角“...”



AI派

AI派专注于分享人工智能领域知识，包括但不限于机器学习、深度学习、数据分析、自然语言处理、推荐系统等。致力于推进人工智能平民化，让每一个感兴趣的人都可以借助人工智能去做一些有意义的事情。

83篇原创文章 2703位朋友关注

进入公众号

取消关注

进入公众号

取消关注

设为星标

3. 点击“设为星标”

更多资料

推荐给朋友

设置

END



长按，识别二维码，加关注

文章转载自公众号

论智

论智

论智