**Python数据处理编程**

**实验报告**

****

|  |  |
| --- | --- |
| **姓名** |  |
| **学号** |  |
| **班级** |  |
| **专业** |  |
| **年级** |  |

目录

[一、 问题描述](#_Toc16459_WPSOffice_Level1) [1](#_Toc16459_WPSOffice_Level1)

[1.1 实验目的](#_Toc16459_WPSOffice_Level2) [1](#_Toc16459_WPSOffice_Level2)

[1.2 实验开发环境和工具](#_Toc18983_WPSOffice_Level2) [1](#_Toc18983_WPSOffice_Level2)

[1.3 实验内容](#_Toc26830_WPSOffice_Level2) [1](#_Toc26830_WPSOffice_Level2)

[二、 需求和规格说明](#_Toc18983_WPSOffice_Level1) [1](#_Toc18983_WPSOffice_Level1)

[三、 概要设计](#_Toc26830_WPSOffice_Level1) [5](#_Toc26830_WPSOffice_Level1)

[四、详细设计](#_Toc23828_WPSOffice_Level1) [5](#_Toc23828_WPSOffice_Level1)

[4.1设计思想](#_Toc18863_WPSOffice_Level2) [5](#_Toc18863_WPSOffice_Level2)

[五、结果分析：](#_Toc16578_WPSOffice_Level1) [12](#_Toc16578_WPSOffice_Level1)

[5.1实现注释程序清单](#_Toc29867_WPSOffice_Level2) [12](#_Toc29867_WPSOffice_Level2)

[5.2结果分析](#_Toc3801_WPSOffice_Level2) [13](#_Toc3801_WPSOffice_Level2)

[六、用户手册](#_Toc166_WPSOffice_Level1) [26](#_Toc166_WPSOffice_Level1)

[七、调试及总结](#_Toc15069_WPSOffice_Level1) [26](#_Toc15069_WPSOffice_Level1)

**实验二 可视化大学专业性别差异**

1. **问题描述** 
   1. **实验目的**

进一步熟悉和掌握数据可视化的相关知识，了解和掌握如何美化图表，掌握可视化中的颜色、布局与标注，了解和掌握如何自定义颜色、线宽、布局和注释，以提高读者从图表中提取信息的能力。

* 1. **实验开发环境和工具**

可以在Windows或Linux操作系统上搭建开发环境，使用Anaconda科学计算包安装Python 3，里面会包含大多数科研常用的库。.ipynb文件推荐使用Anaconda内置的web版Jupyter Notebook。

* 1. **实验内容**

在本实验中，我们将扩展实验一的工作，即可视化大学学位之间的性别差距。到目前为止，我们主要关注的是STEM学位，但现在我们将生成折线图来比较所有学位类别。在本指导项目的最后一步中，我们将探索如何将创建的最终关系图导出为图像文件。

1. **需求和规格说明**

**1.比较所有的专业：**

因为有17个专业需要生成折线图，所以我们将使用6行3列的子块网格布局。然后，我们可以按以下方式将学位分为STEM、文科和其他专业类别：

stem\_cats = ['Psychology', 'Biology', 'Math and Statistics', 'Physical Sciences', 'Computer Science', 'Engineering']

lib\_arts\_cats = ['Foreign Languages', 'English', 'Communications and Journalism', 'Art and Performance', 'Social Sciences and History']

other\_cats = ['Health Professions', 'Public Administration', 'Education', 'Agriculture', 'Business', 'Architecture']

相应的折线图如图1所示。在之前的任务中，stem\_cats的排名是根据性别差距来排列的，而这三个排名都是根据授予女性的学位百分比来排列的。你可能还注意到，虽然stem\_cats和other\_cats有六个学位类别作为元素，但lib\_arts\_cats只有五个。实验中不仅需要修改for循环以生成我们在上一个任务中编写的折线图，还需要添加两个新的for循环以生成文科学位和其他学位的线图。

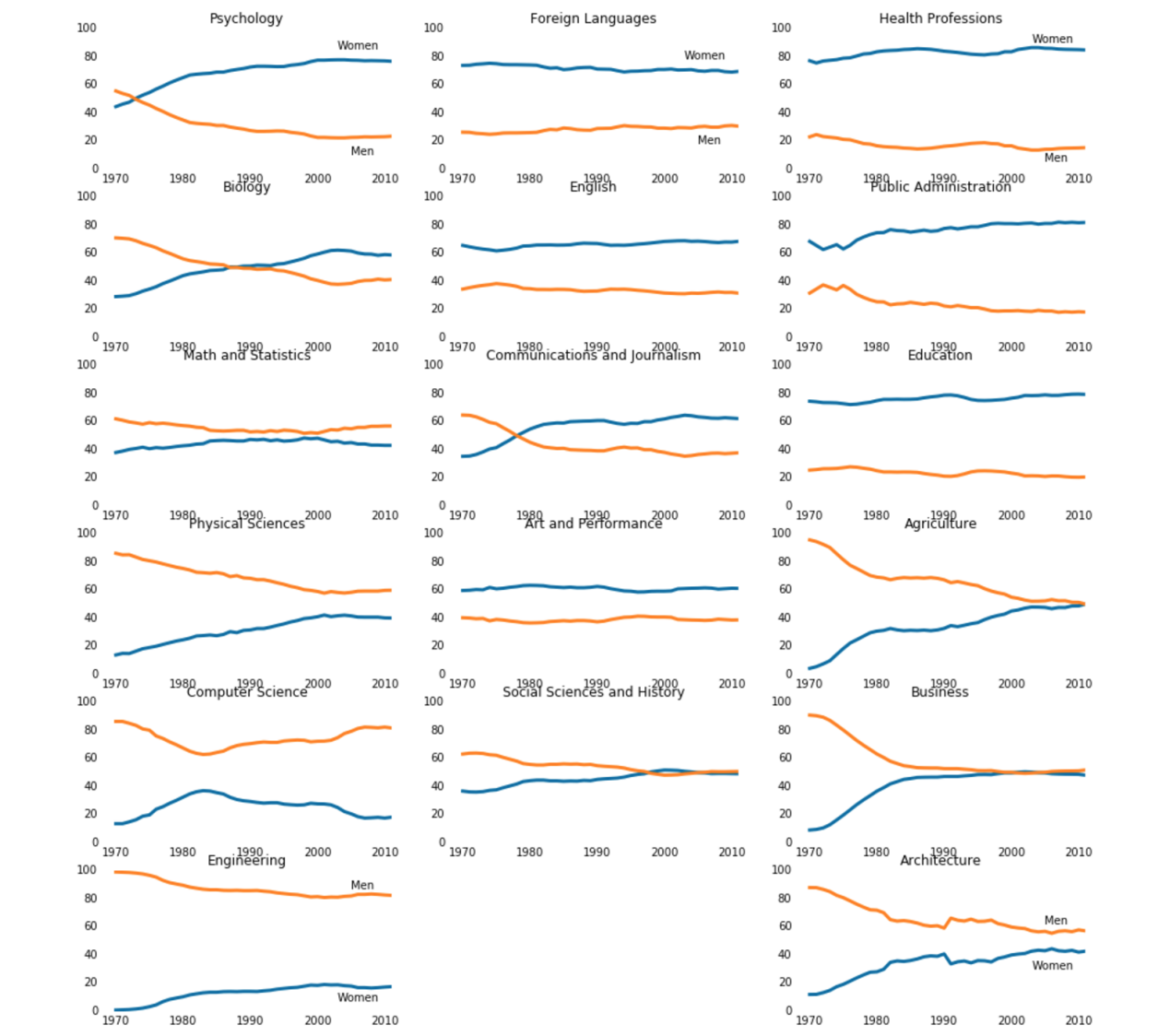


图1 17个专业的折线图

具体要求如下：

生成6行3列的subplots网格。

在第一列中：

为每个stem\_cats类别中的专业，将女生和男生的百分比都生成折线图。

在最上面和最下面的plots中为Women和Men添加文本注释。

在第二列中：

为每一个lib\_arts\_cats类别的专业，将女生和男生的百分比都生成折线图。

为最上面的plots中的Women和Men添加文本注释。

在第三列中：

为每个other\_cats类别中的专业，将女生和男生的百分比都生成折线图。

在最上面和最下面的plots中为Women和Men添加文本注释。

**2.隐藏x轴标签：**

当一个图中有17个折线图时，非数据元素会很容易扰乱视图。最直接突出的问题是一些折线图的标题与上面折线图的x轴标签重叠。如果我们删除每个折线图的标题，浏览者将不知道每个折线图所对应的专业。因此，我们可以移除一列中每个折线图（最下面的除外）的x轴标签。我们可以通过修改对方法Axes.tick\_params()的调用并将labelbottom设置为False来实现，如下所示：

ax.tick\_params(bottom=False, top=False, left=False, right=False, labelbottom=False)

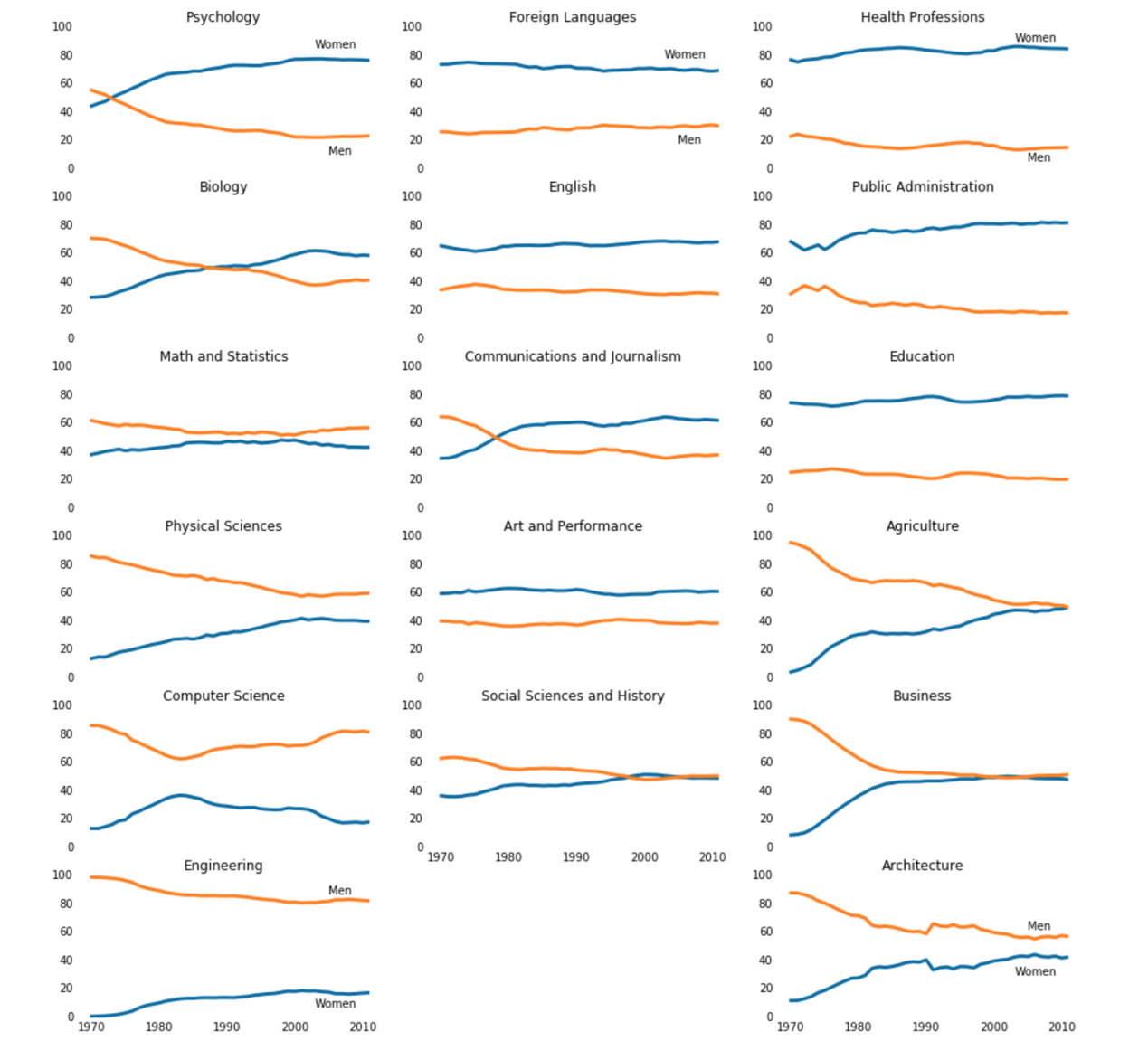
这将禁用所有折线图的x轴标签。可通过以下代码为每列中最下面的折线图启用x轴标签：

ax.tick\_params(labelbottom=True)

具体要求如下：

除了每列最下面的折线图之外，对所有折线图禁用x轴标签。

完成的图应当如下所示：



**3.设置y轴标签**

删除除最下面的折线图以外的所有折线图的x轴标签能够解决重叠文本的问题。这使得绘出来的图更清晰也更具有可读性。这样做所带来的一个问题是：现在浏览者比较难以辨别出哪几年可能发生了一些有趣的趋势变化。然而，这是可以接受的，因为我们的主要关注的是使浏览者能够迅速地获得一个宏观的印象：哪些专业更容易发生性别失衡，以及随着时间的推移，这种情况是如何改变的。

为了使绘出来的图更具有可读性，我们还可以简化y轴标签。目前，所有17个折线图都有6个y轴标签，即使它们在整个网格中是一致的，但它们仍然会增加视觉混乱。通过只保留起始和结束标签（0和100），我们可以保留y轴标签的一些好处但又可以进一步使得图更具有可读性。

我们可以使用轴设置方法Axes.set\_yticks()指定要显示的标签。以下代码仅允许显示标签0和标签100：

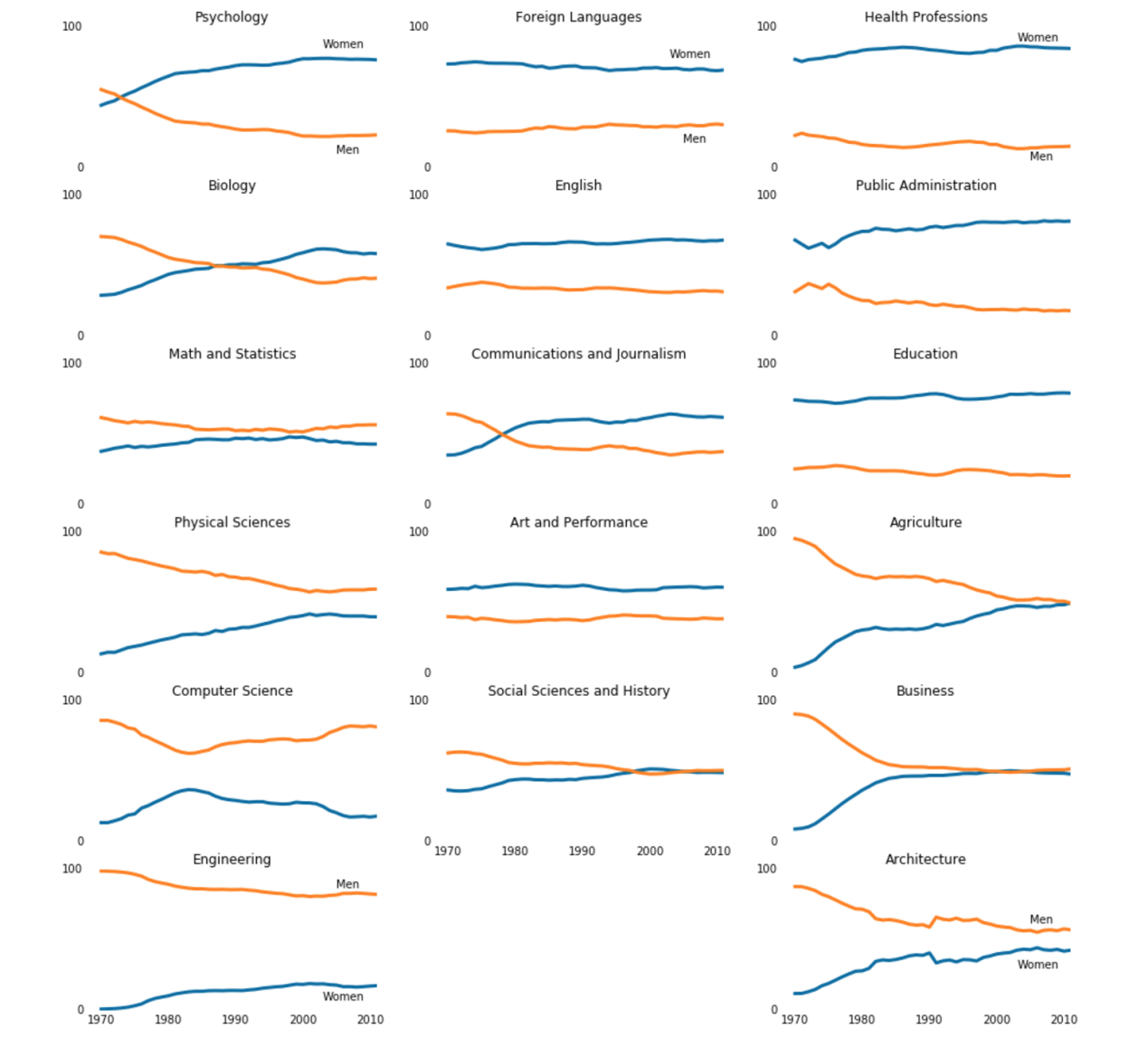
ax.set\_yticks([0,100])

具体要求如下：

对于所有折线图：

仅启用数值0和100处的y轴标签。

完成的图应当如下所示：



**4.添加水平线**

虽然去掉了大部分y轴的标签确实减少了标签所带来的干扰，但这也使得人们很难理解哪些学位接近50-50的性别比例。虽然保留所有y轴标签可以达到这一目的，但实际上我们可以使用y轴标签50位置的水平线实现同样的目的。这样既减少了标签所带来的干扰又能实现观察哪些学位接近50-50的性别比例的目的，所以是一种更好的可视化方式。

我们可以使用ax.axhline() 方法在整个子图上生成一条水平线。唯一需要的参数是该线所处的y轴位置，如下所示：

ax.axhline(50)

我们设置这条水平线的颜色为Color Blind 10 调色板中的一个颜色，它的RGB值为(171,171,171)。因为我们不想让这条线扰乱读者的观看体验，所以我们适当增加这条线的透明度。我们可以使用参数c设置颜色，使用参数alpha 设置透明度。传入alpha 参数的值必须在0到1之间:

ax.axhline(50, c=(171/255, 171/255, 171/255), alpha=0.3)

具体要求如下：

对所有图：

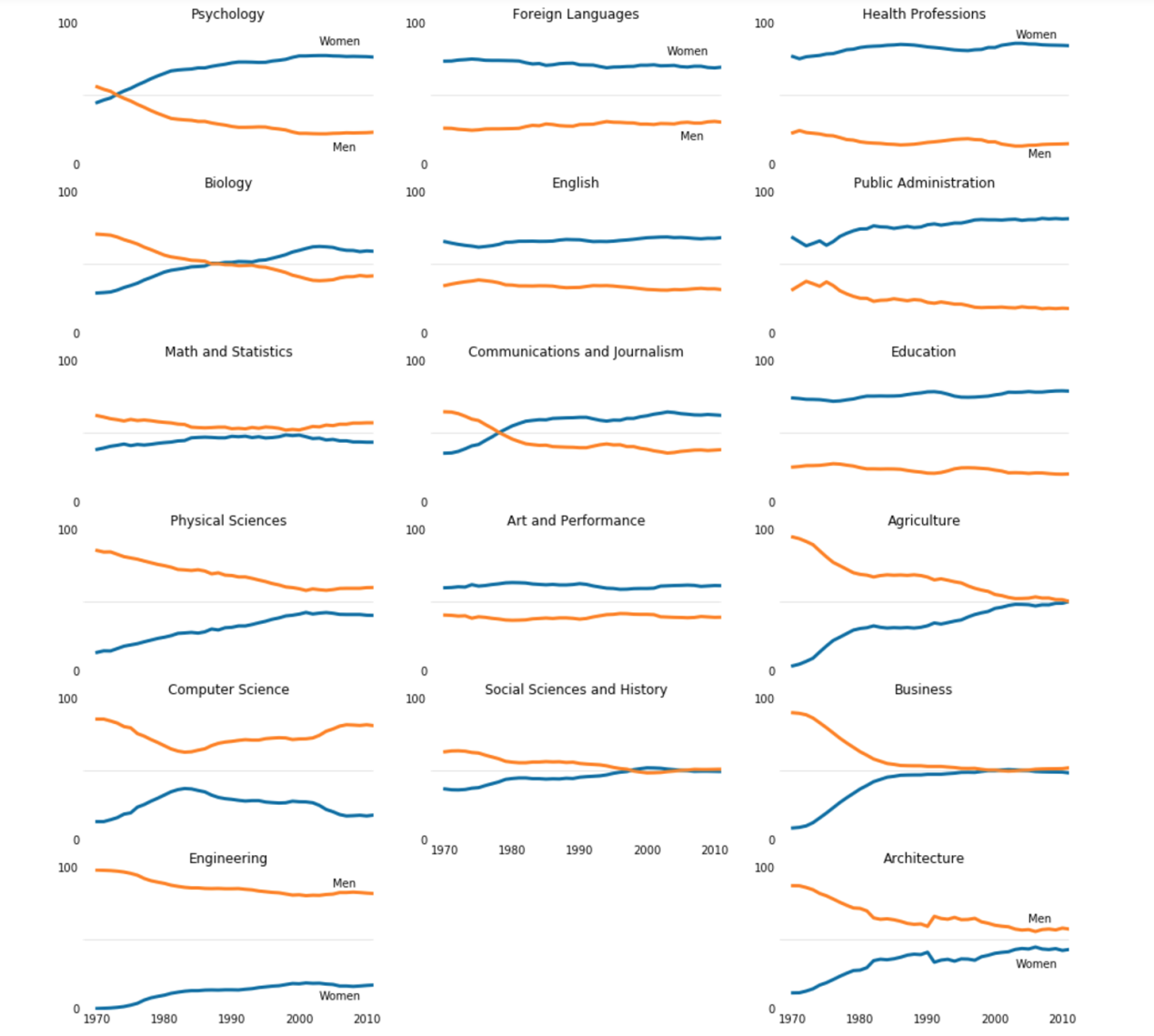
生成一条具有以下属性的水平线：

位置处于y轴的50

颜色为Color Blind 10 调色盘中的第三个颜色（浅灰）

透明度为0.3

效果如下图所示：



**5.将图保存到文件**

你可能还记得，我们可以通过很多方式使用matplotlib。它可以在JupyterNotebook界面、命令行或集成开发环境中使用。许多这些matplotlib的使用方法在工作流中各不相同，处理图像呈现的方式也各不相同。为了支持这些不同的用例，matplotlib可以针对不同的输出或后端。如果导入matplotlib并运行matplotlib.get\_backend() ，你将看到你当前所使用的后端。

在我们当前使用的后端中，可以通过调用Figure.savefig() 或pyplot.savefig() 来导出一张画布中的所有图表到一个图像文件中（注：必须在使用pyplot.show() 显示图像之前调用保存图像的方法），如下所示。

plt.plot(women\_degrees['Year'], women\_degrees['Biology'])

plt.savefig('biology\_degrees.png')

在上面的代码中，我们将折线图保存为PNG文件。图像将被导出到你的Jupyter Notebook服务器所运行的文件夹中。

具体要求如下：

导出所有的折线图到gender\_degrees.png 文件中。

**三、概要设计**

**1.比较所有的专业**

需要修改for循环以生成我们在上一个任务中编写的折线图，还需要添加两个新的for循环以生成文科学位和其他学位的线图。

**2.隐藏x轴标签**

非数据元素会很容易扰乱视图。最直接突出的问题是一些折线图的标题与上面折线图的x轴标签重叠。如果我们删除每个折线图的标题，浏览者将不知道每个折线图所对应的专业。因此，我们可以移除一列中每个折线图（最下面的除外）的x轴标签。我们可以通过修改对方法Axes.tick\_params()的调用并将labelbottom设置为False来实现，如下所示：

ax.tick\_params(bottom=False, top=False, left=False, right=False, labelbottom=False)

**3.设置y轴标签**

为了使绘出来的图更具有可读性，我们还可以简化y轴标签。目前，所有17个折线图都有6个y轴标签，即使它们在整个网格中是一致的，但它们仍然会增加视觉混乱。通过只保留起始和结束标签（0和100），我们可以保留y轴标签的一些好处但又可以进一步使得图更具有可读性。

我们可以使用轴设置方法Axes.set\_yticks()指定要显示的标签。以下代码仅允许显示标签0和标签100：ax.set\_yticks([0,100])

1. **添加水平线**

可以使用ax.axhline() 方法在整个子图上生成一条水平线。唯一需要的参数是该线所处的y轴位置，如下所示：

ax.axhline(50)

我们设置这条水平线的颜色为Color Blind 10 调色板中的一个颜色，它的RGB值为(171,171,171)。因为我们不想让这条线扰乱读者的观看体验，所以我们适当增加这条线的透明度。我们可以使用参数c设置颜色，使用参数alpha 设置透明度。传入alpha 参数的值必须在0到1之间:

ax.axhline(50, c=(171/255, 171/255, 171/255), alpha=0.3)

**5.将图保存到文件**

可以通过调用Figure.savefig() 或pyplot.savefig() 来导出一张画布中的所有图表到一个图像文件中（注：必须在使用pyplot.show() 显示图像之前调用保存图像的方法）。

**四、详细设计**

**4.1设计思想**

**1.比较所有的专业**

import matplotlib.pyplot as plt

import pandas as pd

file = pd.read\_csv("percent-bachelors-degrees-women-usa.csv")

df = pd.DataFrame(file)

m\_cats = ['Psychology', 'Biology', 'Math and Statistics', 'Physical Sciences', 'Computer Science', 'Engineering']

lib\_arts\_cats = ['Foreign Languages', 'English', 'Communications and Journalism', 'Art and Performance', 'Social Sciences and History']

other\_cats = ['Health Professions', 'Public Administration', 'Education', 'Agriculture','Business', 'Architecture']

#file['Year'] = pd.to\_datetime(file['Year'])

i=0

j=0

k=0

fig, ax = plt.subplots(nrows=6, ncols=3, figsize=(12,18))

ax[5][1].remove()#删除空白图

for value in m\_cats:

fig1=plt.figure(figsize=(3,2))

ax[i][0].plot(df['Year'], df[value],c='orange',label='Women')

ax[i][0].plot(df['Year'], 100-df[value],c='blue',label='Men')

ax[i][0].set\_title(value)

ax[i][0].legend(loc='upper left')#在标签图中的左上角显示

i = i+1

for value in lib\_arts\_cats:

fig2=plt.figure(figsize=(3,2))

ax[j][1].plot(df['Year'], df[value],c='orange',label='Women')

ax[j][1].plot(df['Year'], 100-df[value],c='blue',label='Men')

ax[j][1].set\_title(value)

j = j+1

for value in other\_cats:

fig3=plt.figure(figsize=(3,2))

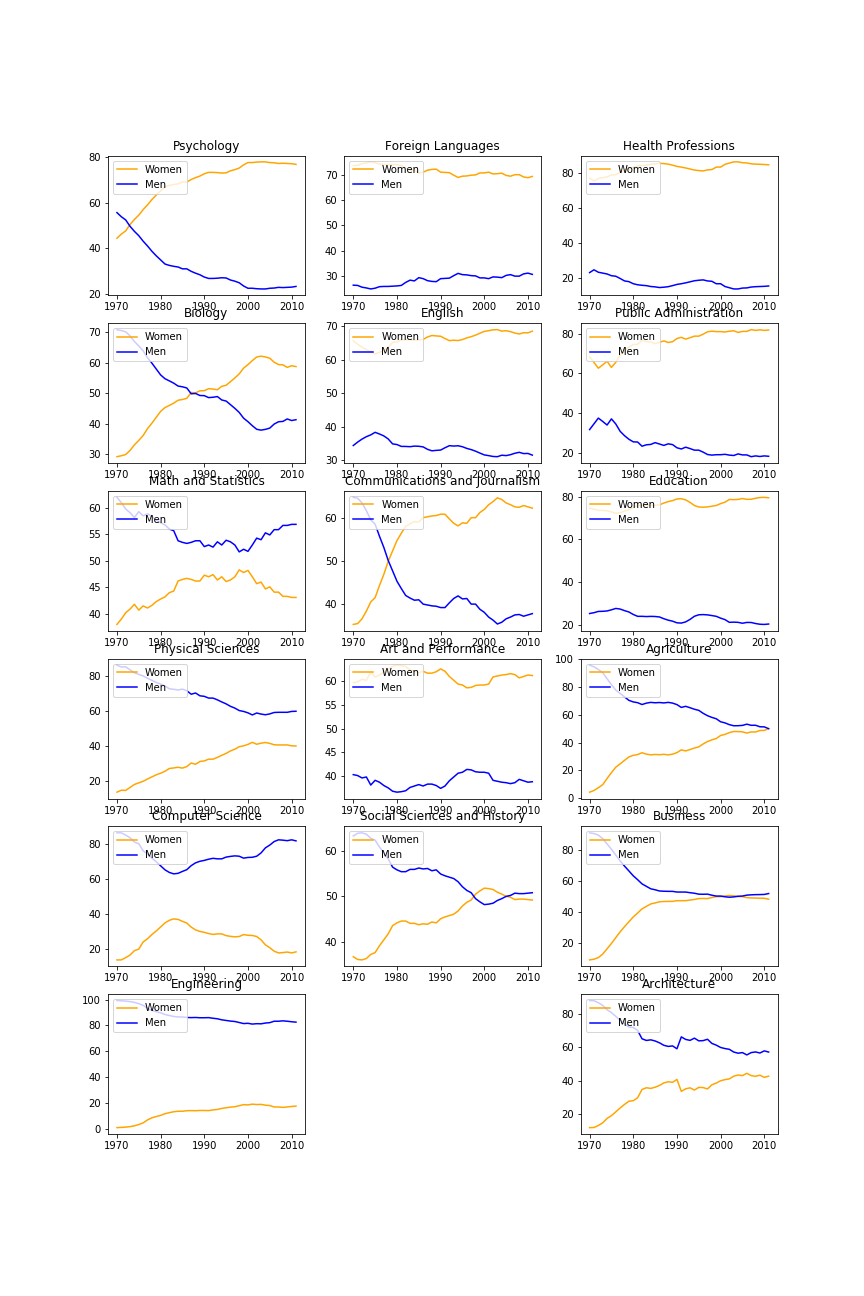
ax[k][2].plot(df['Year'], df[value],c='orange',label='Women')

ax[k][2].plot(df['Year'], 100-df[value],c='blue',label='Men')

ax[k][2].set\_title(value)

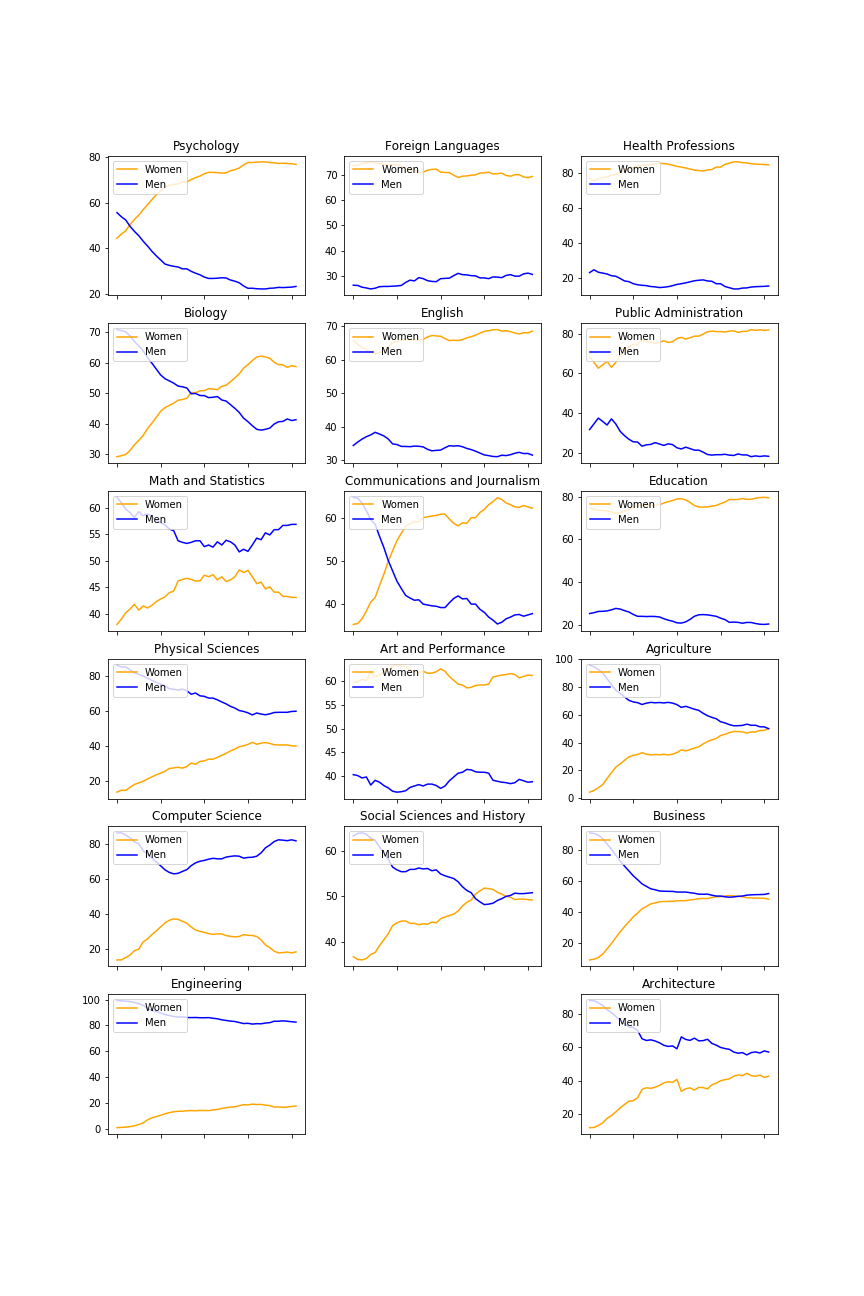
k = k+1

plt.show()



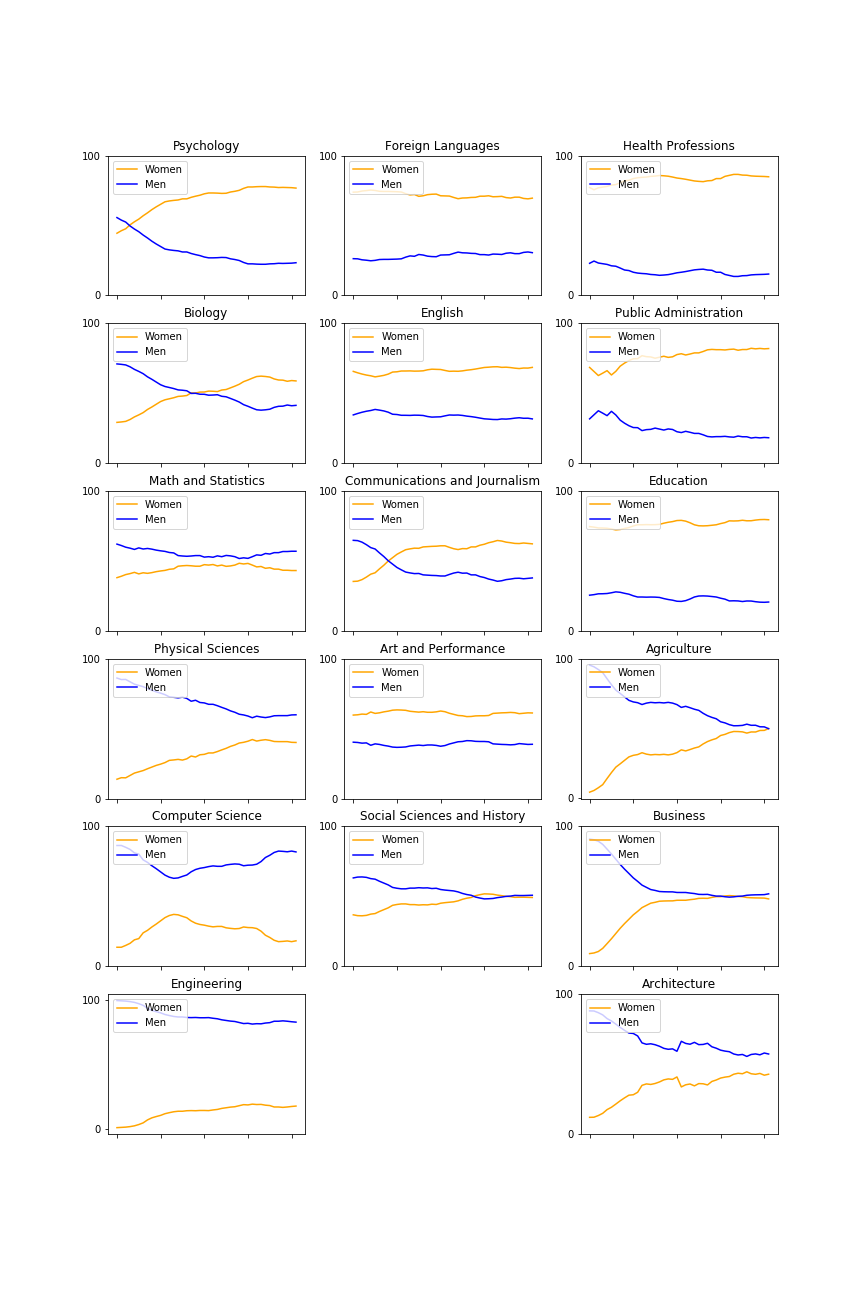
**2.隐藏x轴标签**

ax[i][0].tick\_params(labelbottom=False)#禁用所有折线图的x轴标签



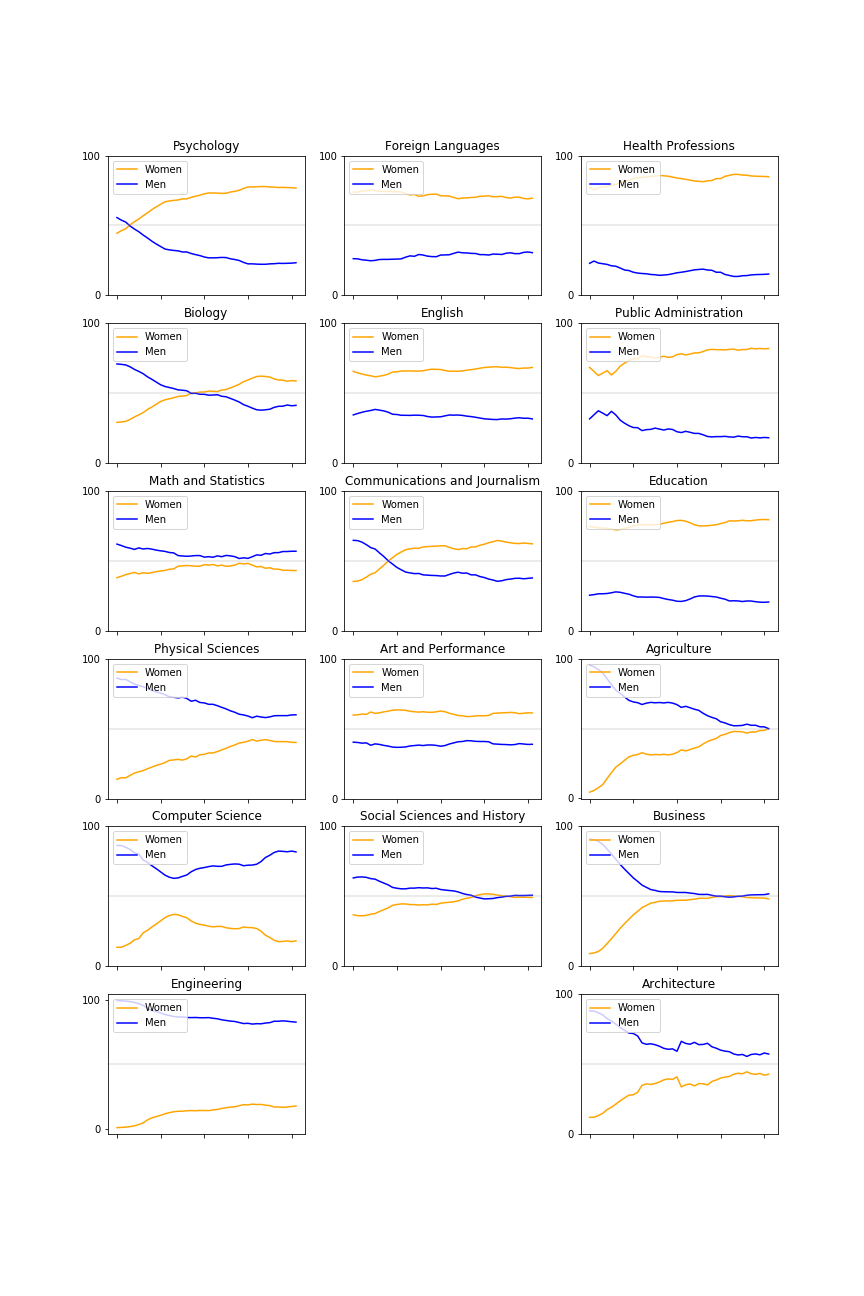
**3.设置y轴标签**

ax[i][0].set\_yticks([0,100])#只保留起始和结束标签（0和100）



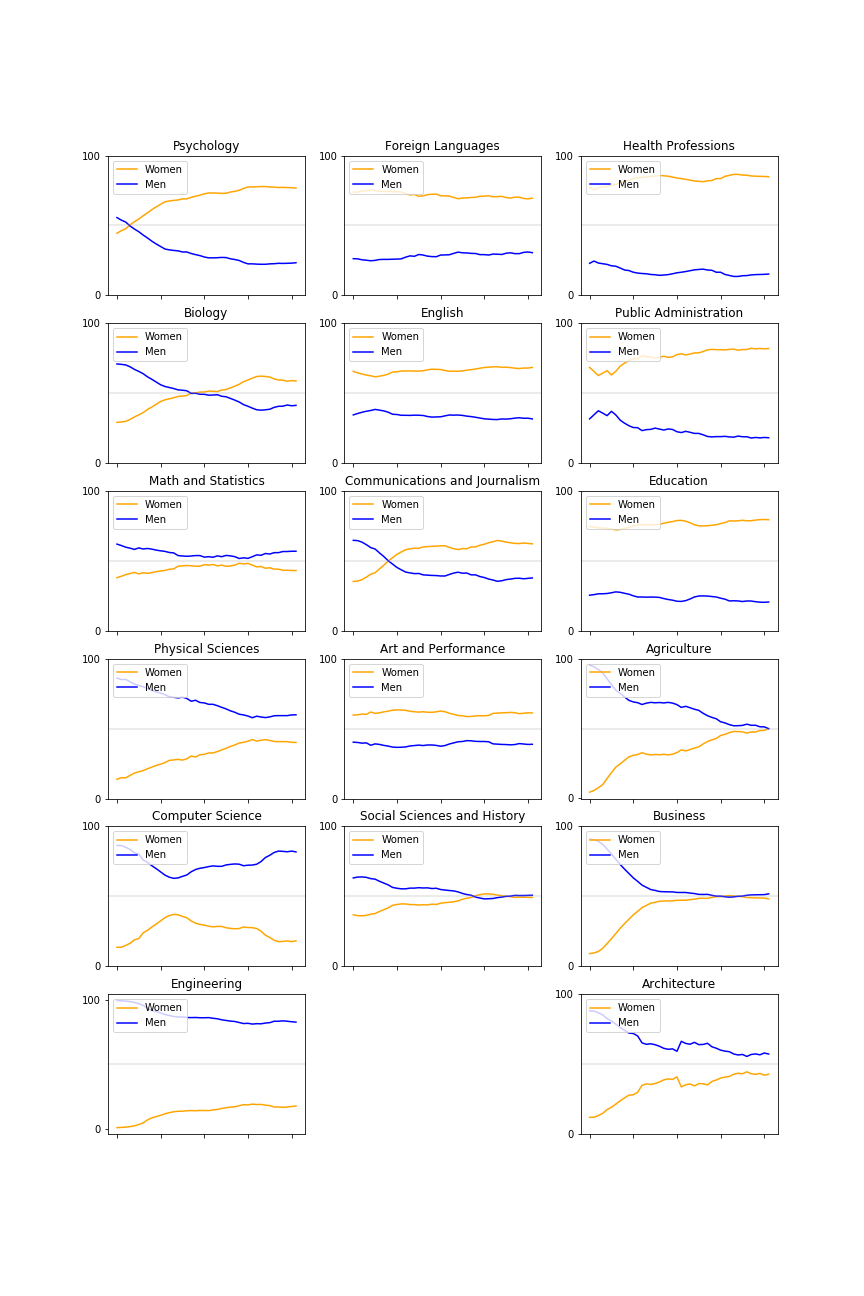
**4.添加水平线**

ax[i][0].axhline(50, c=(171/255, 171/255, 171/255), alpha=0.3)#显示中心线



**5.将图保存到文件**

fig.savefig('gender\_degrees.png')

****

**五、结果分析：**

**5.1实现注释程序清单**

import matplotlib.pyplot as plt

import pandas as pd

file = pd.read\_csv("percent-bachelors-degrees-women-usa.csv")

df = pd.DataFrame(file)

m\_cats = ['Psychology', 'Biology', 'Math and Statistics', 'Physical Sciences', 'Computer Science', 'Engineering']

lib\_arts\_cats = ['Foreign Languages', 'English', 'Communications and Journalism', 'Art and Performance', 'Social Sciences and History']

other\_cats = ['Health Professions', 'Public Administration', 'Education', 'Agriculture','Business', 'Architecture']

#file['Year'] = pd.to\_datetime(file['Year'])

i=0

j=0

k=0

fig, ax = plt.subplots(nrows=6, ncols=3, figsize=(12,18))

ax[5][1].remove()

for value in m\_cats:

fig1=plt.figure(figsize=(3,2))

ax[i][0].plot(df['Year'], df[value],c='orange',label='Women')

ax[i][0].plot(df['Year'], 100-df[value],c='blue',label='Men')

ax[i][0].set\_title(value)

ax[i][0].tick\_params(labelbottom=False)#禁用所有折线图的x轴标签

ax[i][0].set\_yticks([0,100])#只保留起始和结束标签（0和100）

ax[i][0].axhline(50, c=(171/255, 171/255, 171/255), alpha=0.3)#显示中心线

ax[i][0].legend(loc='upper left')#在图中的左上角显示

i = i+1

for value in lib\_arts\_cats:

fig2=plt.figure(figsize=(3,2))

ax[j][1].plot(df['Year'], df[value],c='orange',label='Women')

ax[j][1].plot(df['Year'], 100-df[value],c='blue',label='Men')

ax[j][1].set\_title(value)

ax[j][1].tick\_params(labelbottom=False)

ax[j][1].set\_yticks([0,100])

ax[j][1].axhline(50, c=(171/255, 171/255, 171/255), alpha=0.3)

ax[j][1].legend(loc='upper left')

j = j+1

for value in other\_cats:

fig3=plt.figure(figsize=(3,2))

ax[k][2].plot(df['Year'], df[value],c='orange',label='Women')

ax[k][2].plot(df['Year'], 100-df[value],c='blue',label='Men')

ax[k][2].set\_title(value)

ax[k][2].tick\_params(labelbottom=False)

ax[k][2].set\_yticks([0,100])

ax[k][2].axhline(50, c=(171/255, 171/255, 171/255), alpha=0.3)

ax[k][2].legend(loc='upper left')

k = k+1

fig.savefig('gender\_degrees.png')

plt.show()

**5.2结果分析**

由各个专业男女生占比分布可知，不同专业受男女生的青睐程度不同

1. **用户手册**

若用户需要对类似问题进行分析时，只需按照步骤分析修改图图片显示要求，并将以上图表的数据进行修改即可。

**七、调试及总结**

1.在作图的时候，明明plt.plot()中设置了label的值，但是在图中plt.show()以后就是没有显示出来那个标签，此时，只需要添加一个函数plt.legend()用来标示不同图形的文本标签图例就行了。

2.ax[][]如果是多列绘图可任意充分利用ax的角标定位绘制图像

3.ax[][].remove()该函数可以删除subplot所固定的图片个数，即如本题可以定位删除没有数据的图。

4.for value in m\_cats:

fig1=plt.figure(figsize=(3,2))

ax[i][0].plot(df['Year'], df[value],c='orange',label='Women')

其中我们可以根据不同的value依此绘图。