《Python数据处理编程》课内实验指导书

内容 简 介

Python数据处理编程是计算机和大数据专业学生的一门专业必修课程，着重讲述基于Python的数据处理的基本概念和技术方法。对于学生从事大数据分析、数据挖掘、机器学习、深度学习等系统的研发、使用和维护有重要意义。本课程概念多、内容涉及面广、系统性强。通过本课程的学习，学生应能从数学模型、编程实现的角度去了解、分析和研究基于Python的数据处理应用系统，建立起对基于Python的数据处理应用系统的全面认识，树立全面地、发展地看问题的观点，从而加深对各种基于Python的数据处理应用系统的了解。

本课程的学习应注重理论与实践相结合，因此实验教学是教学环节中必不可少的重要内容。通过实验教学的学习，使学生熟练掌握有关基于Python的数据处理编程的基本概念、基本原理和基本思想，掌握对基于Python的数据处理系统进行设计、分析和计算的方法。

实验部分包括实验目的、实验内容和实验所需环境等，介绍了实验所需的一些基础知识和技巧。在实验中给出的实验题，跟课堂教学的内容都有密切的关系，所以需要将课堂上讲授的例子程序融会贯通，掌握实验所需的一些基本方法和工具，并在吃透例子程序的基础上，积极独立思考设计和编写满足实验要求的程序。

中南大学计算机学院鲁鸣鸣制定

上机实验要求及规范

Python数据处理编程课程具有比较强的实践性。上机实验是一个重要的教学环节。一般情况下学生能够重视实验环节，对于编写程序上机练习具有一定的积极性。但是容易忽略实验的总结，忽略实验报告的撰写。对于一名大学生必须严格训练分析总结能力、书面表达能力。需要逐步培养书写科学实验报告以及科技论文的能力。拿到一个题目，一般不要急于编程。正确的方法是：首先理解问题，明确给定的条件和要求解决的问题，然后按照自顶向下，逐步求精，分而治之的策略，按照面向对象的程序设计思路，逐一地解决子问题。

一、实验报告的基本要求：

一般性、较小规模的上机实验题，必须遵循下列要求。养成良好的习惯。

姓名 班级 学号 日期 题目

i. 问题描述

ii. 设计简要描述

iii. 程序清单（带有必要的注释）

iv. 结果分析（原始图示，测试数据与运行记录，分析正确性；）

v. 调试报告：

实验者必须重视最后这两个环节，否则等同于没有完成实验任务。这里可以体现个人特色、或创造性思维。具体内容包括：测试数据与运行记录；调试中遇到的主要问题，自己是如何解决的；经验和体会等。

二、实验报告的提高要求：

阶段性、较大规模的上机实验题，应该遵循下列要求。养成科学的习惯。

（1）问题描述

（2）需求和规格说明

（3）描述问题，简述题目要解决的问题是什么。规定软件做什么。原题条件不足时补全。

（4）概要设计：功能模块的划分

（5）详细设计：每部分模块的设计，含数据结构的设计，算法的描述（流程图或PDL）

a.设计思想：主要算法基本思想。

b.设计表示：每个函数的头和规格说明；列出每个函数所调用和被调用的函数，也可以通过调用关系图表达。

（6）实现注释：各项功能的实现程度、在完成基本要求的基础上还有什么功能。

（7）用户手册：即使用说明书。

（8）调试报告：调试过程中遇到的主要问题是如何解决的；设计的回顾、讨论和分析；时间复杂度、空间复杂度分析；改进设想；经验和体会等。

实验二 **可视化大学专业性别差异**

一、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。

二、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。

三、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。具体任务如下：

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。

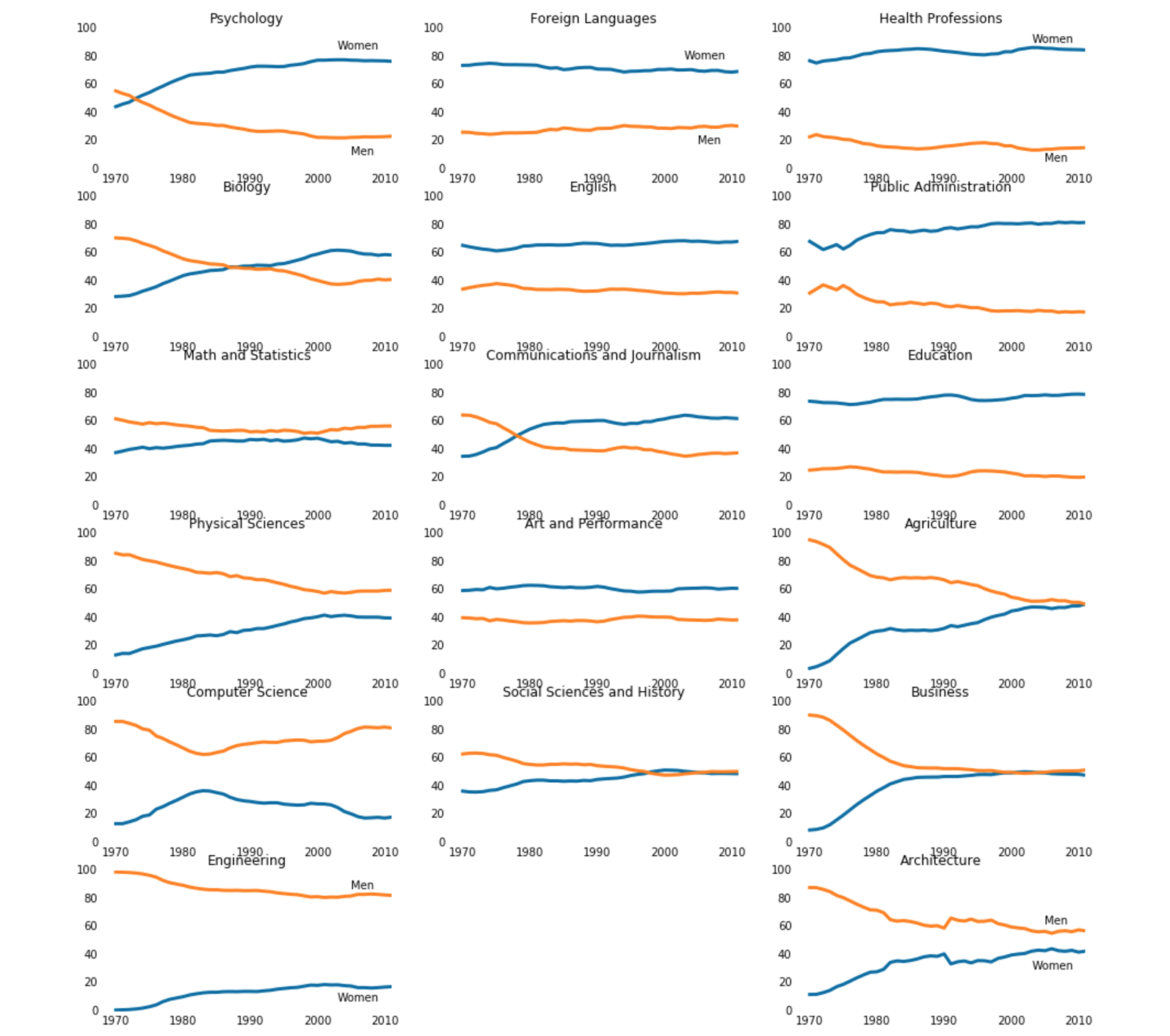


图1 17个专业的折线图

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。

1. **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：

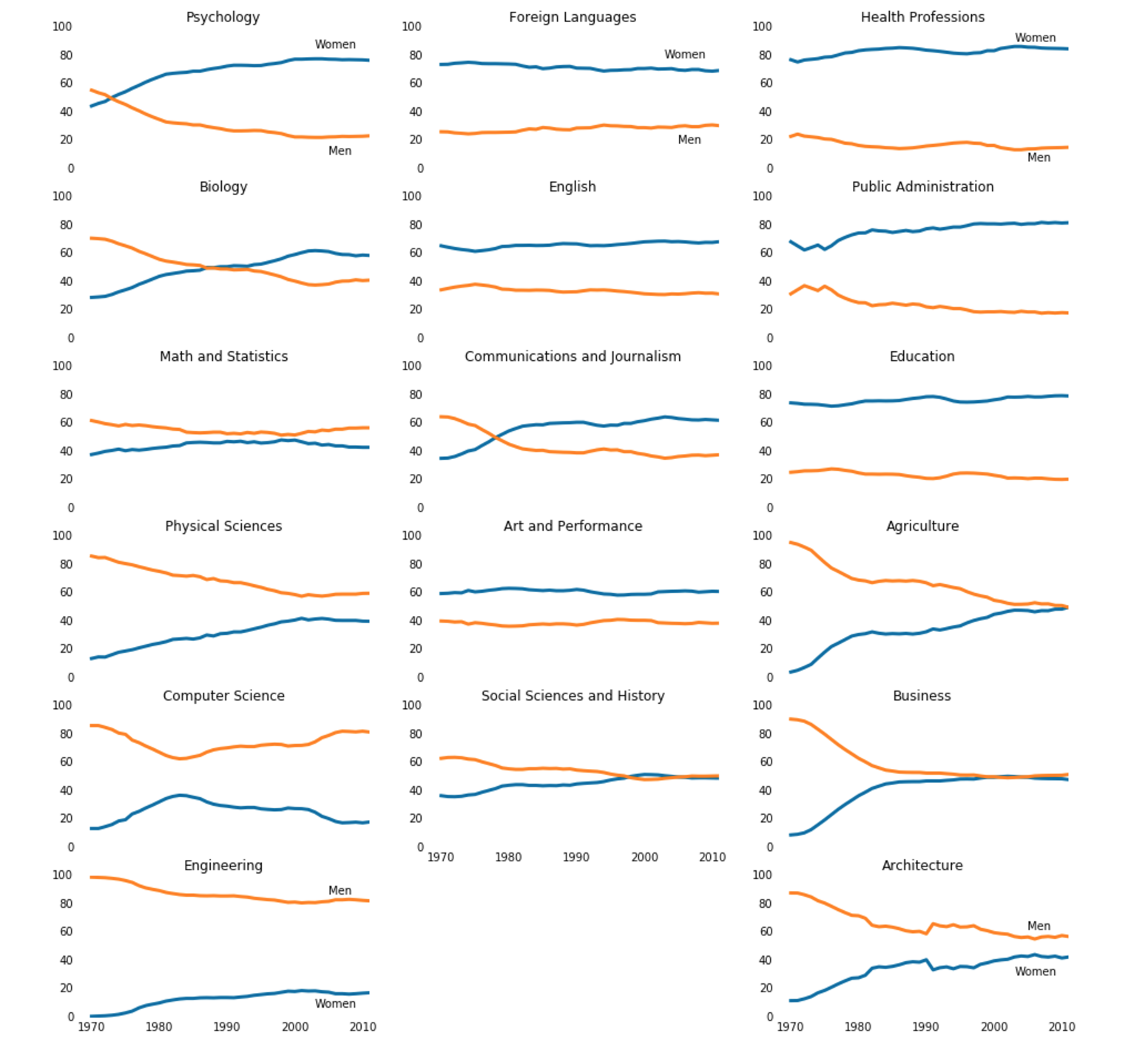
ax.tick\_params(bottom=False, top=False, left=False, right=False, labelbottom=False)

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：



1. **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。

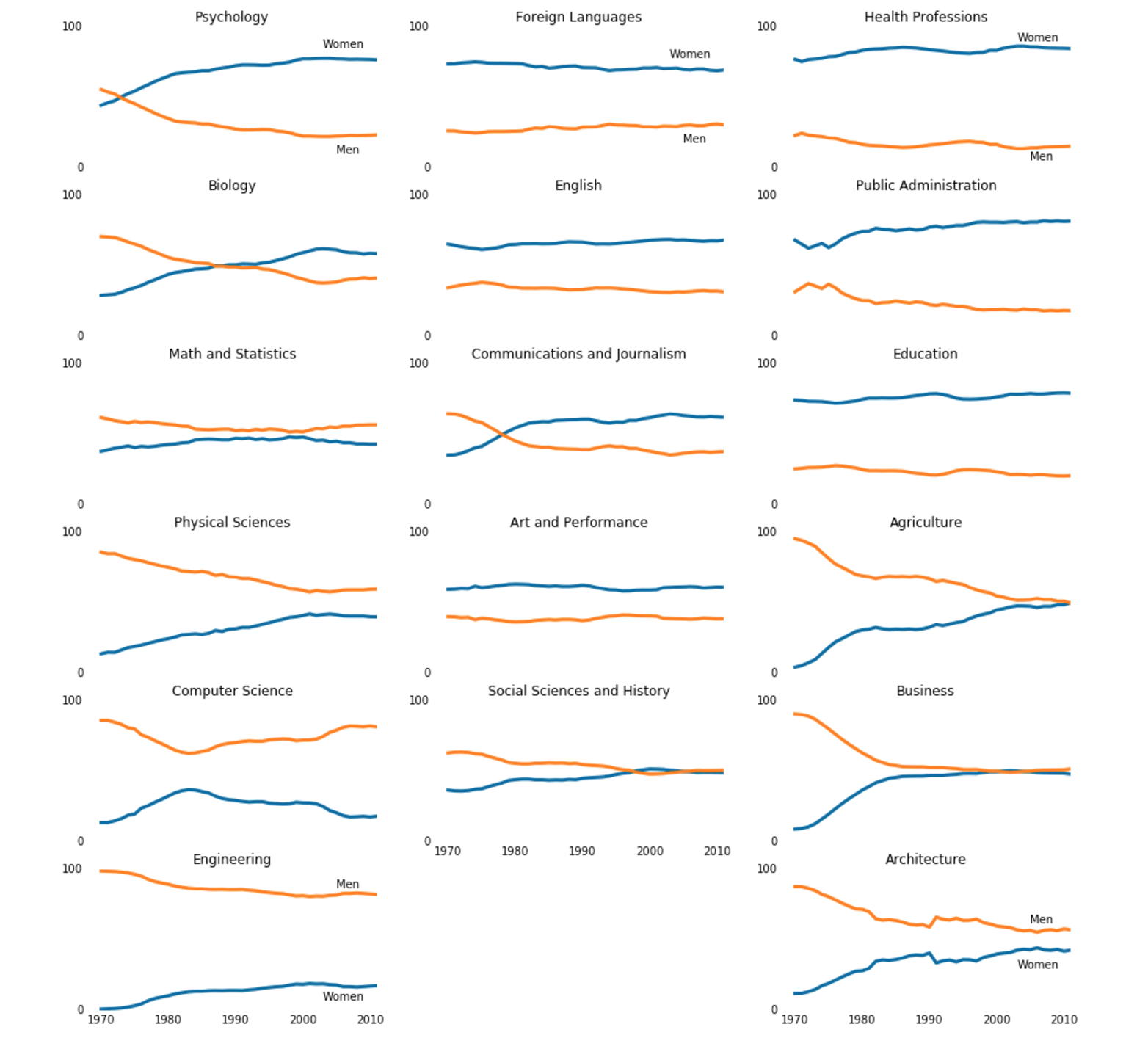
为了使绘出来的图更具有可读性，我们还可以简化y轴标签。目前，所有17个折线图都有6个y轴标签，即使它们在整个网格中是一致的，但它们仍然会增加视觉混乱。通过只保留起始和结束标签（0和100），我们可以保留y轴标签的一些好处但又可以进一步使得图更具有可读性。

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：



1. **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：

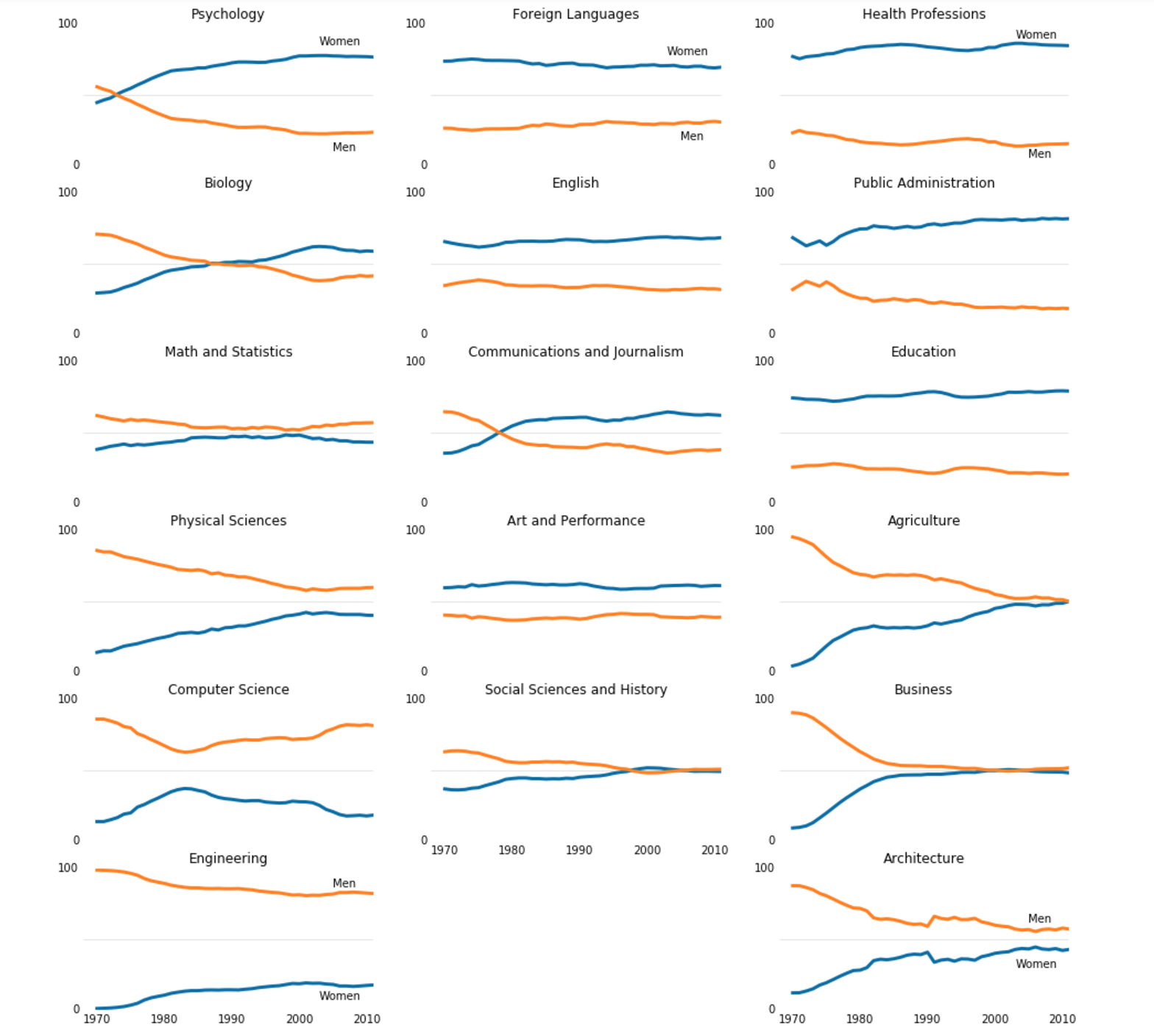
ax.axhline(50)

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：



1. **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。

参考文献

[1] https://www.educoder.net/shixuns/m3eztwlk/challenges

[2] https://www.educoder.net/shixuns/f7vu3wpm/challenges