

4.3 平面填充

这个话题聊一聊常见的平面填充方案。

沿横轴填充

`fill_between()` 是 `matplotlib.pyplot` 库中的一个函数，用于绘制两个曲线之间的填充区域。`fill_between(x, y1, y2, ...)` 函数可以接受两个数组 `x` 和 `y1`，以及另外一个数组 `y2`，它们都是相同长度的。这个函数会将 `y1` 和 `y2` 之间的区域填充，并在 `x` 上绘制。图 1 所示三个例子展示曲线和水平线之间沿横轴填充。`fill_between()` 可以使用参数 `where` 来指定 `y1` 和 `y2` 之间的填充区域，可以使用参数 `facecolor` 来指定填充颜色，使用参数 `alpha` 来指定填充区域的透明度。

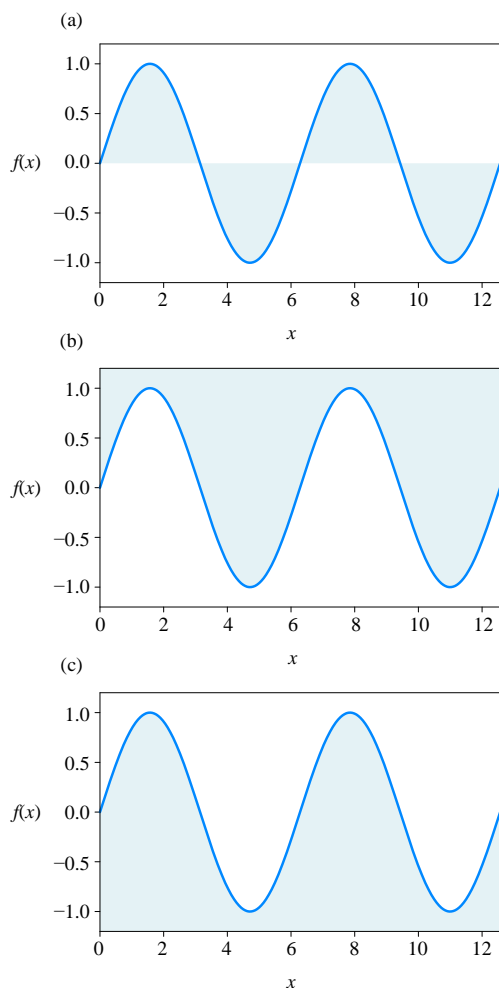


图 1. 曲线和水平线之间沿横轴填充

如图 2 所示，通过设置条件，我们还可以给满足不同条件的区域填充不同颜色。图 3 还给出两个例子，展示两条曲线之间沿横轴填充。

Jupyter 笔记 BK_2_Topic_4.03_1.ipynb 绘制图 1、图 2、图 3。

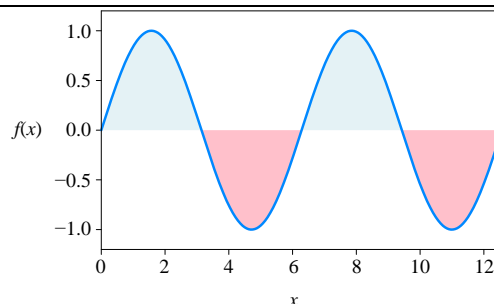


图 2. 曲线和水平线之间沿横轴填充，不同颜色

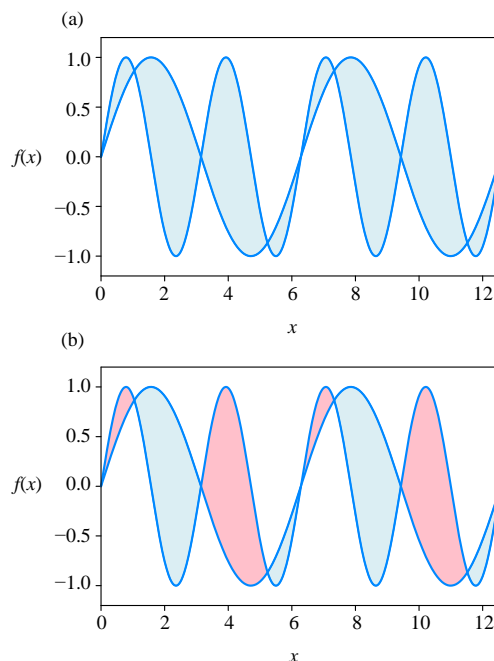


图 3. 两条曲线之间沿横轴填充

沿纵轴填充

`fill_betweenx()` 则可以用来绘制两个曲线在 `y` 轴方向之间的填充区域。`fill_betweenx(y, x1, x2, ...)` 函数接受两个数组 `y` 和 `x1`，以及另外一个数组 `x2`，它们都是相同长度的。这个函数会将 `x1` 和 `x2` 之间的区域填充，并在

y 上绘制。图 5 所示为沿纵轴方向填充的 6 个例子。

Jupyter 笔记 BK_2_Topic_4.03_2.ipynb 绘制图 5。

填充阴影线

Matplotlib 中我们还可以用 `hatch` 给各种填充增加阴影线。`hatch` 是 `matplotlib` 库中的一个属性，用于给某些图形元素添加填充图案。要使用 `hatch` 属性，需要将它设置为一个字符串，该字符串描述了所需的填充图案类型。`matplotlib` 库中提供了多种不同的填充图案类型，下面给出几个例子：

```

'/'：斜杠填充
'\': 反斜杠填充
'.'：点状填充
'o'：圆形填充
'-': 横向线性填充
'+': 十字线填充
'x': 斜十字线填充
'|': 纵向线性填充

```

BK_2_Topic_4.03_3.ipynb 给出几种常见的阴影线，请大家自行学习。也请大家在前两个 Jupyter 笔记使用不同的填充阴影线。

参考填充色块

本书前文介绍过如何绘制水平、竖直参考线，类似地，我们也可以绘制参考填充色块。`axhspan` 是 `matplotlib` 库中的一个函数，用于在一个子图中绘制一个水平的矩形。

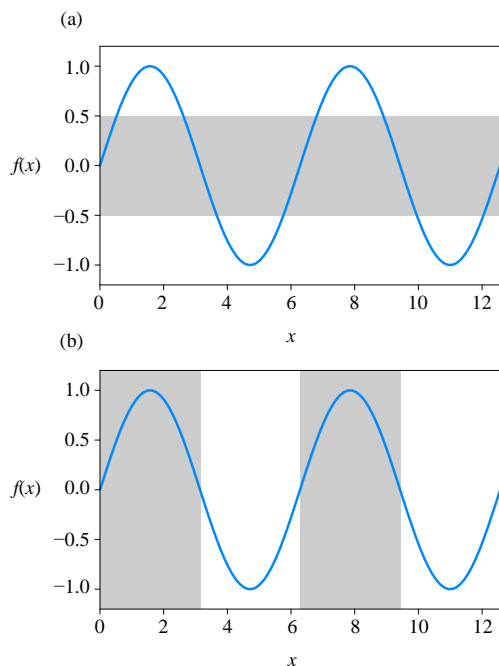


图 4. 参考填充色块

这个函数通常用于强调某个区域的范围或表示一个特定的数据区间，如图 4 (a) 所示。

`axhspan` 函数接受四个参数：`ymin`、`ymax`、`xmin` 和 `xmax`。其中，`ymin` 和 `ymax` 表示矩形的纵向范围，`xmin` 和 `xmax` 表示矩形的横向范围。类似地，`axvspan` 可以绘制竖直方向参考填充色块，如图 4 (b) 所示。

封闭填充

`fill` 是 `matplotlib` 库中常用的一个函数，用于绘制填充区域。使用 `fill` 函数可以将一个多边形区域填充成指定的颜色。

`fill` 函数接受两个参数：`x` 坐标数组和 `y` 坐标数组，用于指定要填充的多边形区域的顶点坐标。`x` 和 `y` 的长度必须相同，且每个元素都对应一个多边形的顶点。图 6 给出的例子还用到了旋转，《矩阵力量》会介绍如何利用线性代数工具完成旋转操作。

Jupyter 笔记 BK_2_Topic_4.03_5.ipynb 绘制图 6。

堆叠面积图

`stackplot` 函数是 `matplotlib` 库中用于绘制堆叠面积图的函数。堆叠面积图可以用于展示多个数据序列之间的比例关系，常用于显示时间序列数据的变化趋势。此外，更常用的是直接用 `dataframe` 绘制堆叠面积图，如图 7 所示。

Jupyter 笔记 BK_2_Topic_4.03_6.ipynb 绘制图 7。

添加图形元素

`add_patch` 是 `matplotlib` 库中 `Axes` 对象的一个方法，用于向一个子图中添加一个图形元素。这个方法可以添加多种不同类型的图形元素，例如矩形、多边形、圆形、椭圆、箭头等等。

在使用 `add_patch` 方法前，需要先创建一个对应的图形元素对象，例如 `Circle`、`Rectangle`、`Polygon`、`Ellipse`、`Arrow` 等。然后，可以使用 `add_patch` 方法将这个对象添加到指定的子图中。在添加完成后，可以使用 `set_*` (比如 `set_edgecolor`、`set_linewidth`) 方法或属性来设置图形元素的属性，例如填充颜色、边框颜色、边框宽度等。图 8 所示为利用长方形可视化最小二乘回归原理。

Jupyter 笔记 BK_2_Topic_4.03_7.ipynb 绘制图 8。

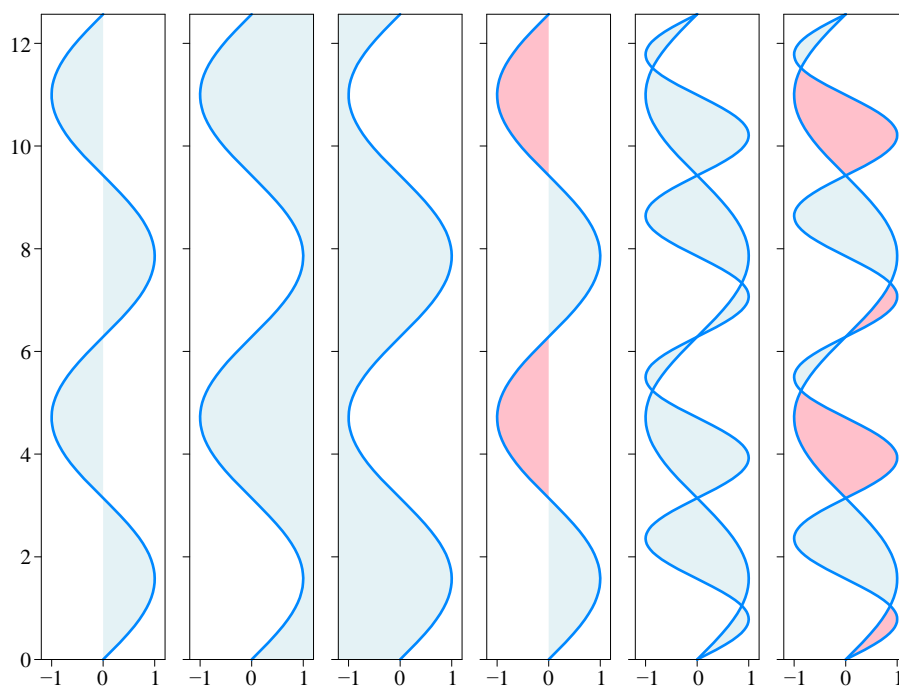


图 5. 沿纵轴方向填充

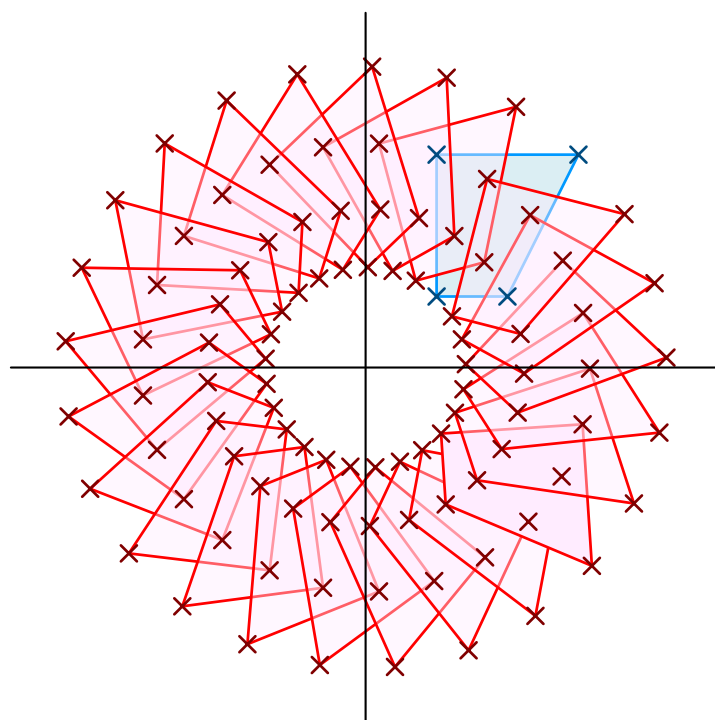


图 6. 用 fill 首尾连接封闭填充

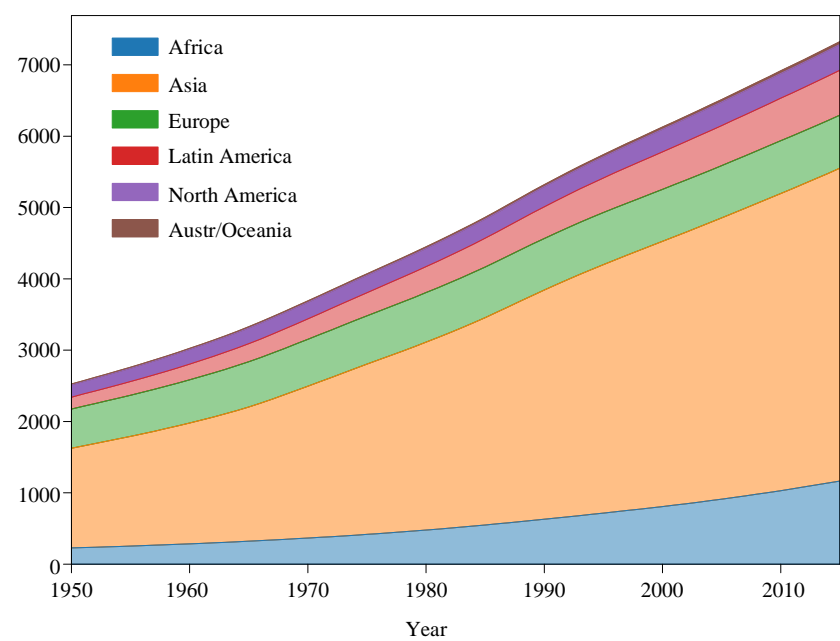


图 7. 用堆叠面积图绘制人口变化

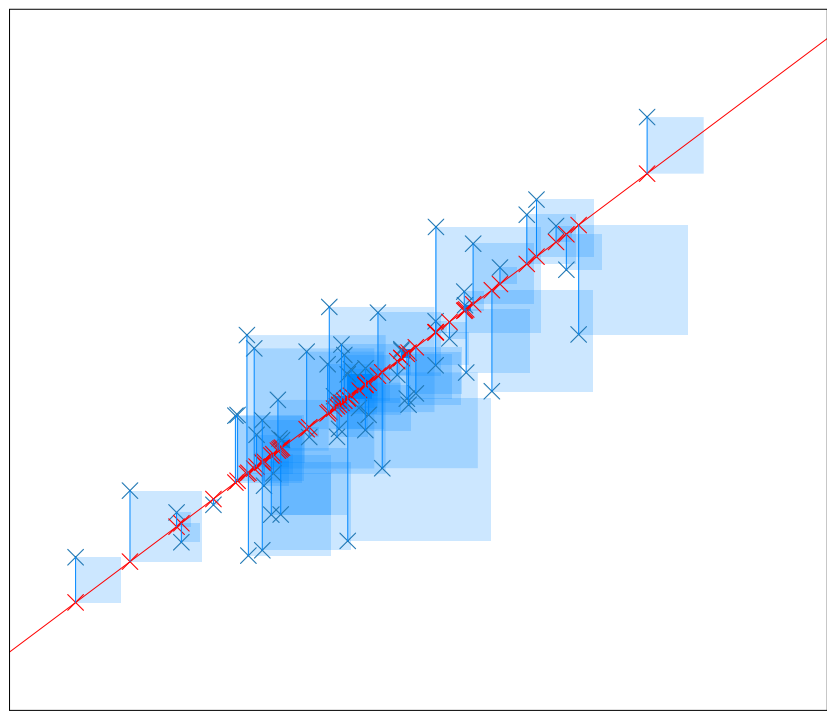


图 8. 添加几何元素