# 第1章 数据可视化与matplotlib

1. **填空题**
2. 数据可视化是从数据空间到 图形 空间的映射。
3. 狭义的数据可视化是将数据以 图表 、图形、地图的方式进行呈现。
4. 箱型图 是一种利用数据中的5个统计量描述数据的图表。
5. matplotlib是一个主要用于绘制 2D 图表的Python库。
6. Anaconda 提供了包管理器、环境管理器，包括诸如numpy、pandas、matplotlib、scipy等180多个科学计算包及其依赖项。
7. **判断题**
8. 数据可视化是一个抽象的过程。（√）
9. 散点图可以清晰地展示数据增减的趋势、速率、规律、峰值等特征。（× ）
10. 柱形图与直方图展示的效果完全相同。（ × ）
11. matplotlib只能采用面向对象的方式开发程序。（x ）
12. **选择题**
13. 下列选项中，关于数据可视化描述错误的是（ C）。
14. 数据可视化可以简单地理解为将不易描述的事物形成可感知画面的过程
15. 数据可视化的目的是准确地、高效地、全面地传递信息
16. 数据表格是数据可视化最基础的应用
17. 数据可视化对后期数据挖掘具有深远的影响
18. 关于常见图表的说法中，下列描述正确的是（C ）。
19. 柱形图可以反映数据增减的趋势
20. 条形图是横置的直方图
21. 饼图用于显示数据中各项大小与各项总和的比例
22. 雷达图是一种可以展示多变量关系的图表
23. 下列图表中，可以反映三个变量之间关系的是（D ）。
24. 折线图
25. 柱形图
26. 散点图
27. 气泡图
28. 下列哪个可视化库可以生成Echarts 图表？（ D）
29. matplotlib
30. seaborn
31. bokeh
32. pyecharts
33. 下列选项中，属于数据之间逻辑关系的是（ABCD ）。（多选）
34. 比较
35. 分布
36. 构成
37. 联系

# 第2章 使用matplotlib绘制简单图表

1. **填空题**
2. plot()函数会返回一个包含多个 Line2D 类对象的列表。
3. 常见的 堆积图 包括堆积面积图、堆积柱形图和堆积条形图。
4. pyplot绘制的直方图默认有 10 个矩形条。
5. **判断题**
6. pyplot只能使用errorbar()函数绘制误差棒图。（x ）
7. pyplot可以使用barh()函数绘制堆积条形图。（ √）
8. pyplot绘制的箱形图默认不显示异常值。（ x）
9. **选择题**
10. 下列函数中，可以快速绘制雷达图的是（D ）。
11. bar()
12. barh()
13. hist()
14. polar()
15. 当pyplot调用barh()函数绘图时，可以通过哪个参数设置图表的刻度标签？（C ）
16. width
17. height
18. tick\_label
19. align
20. 请阅读下面一段代码：

plt.bar(x, y1, tick\_label=["A", "B", "C", "D"])

plt.bar(x, y2, bottom=y1, tick\_label=["A", "B", "C", "D"])

以上代码中bar()函数的bottom参数的作用是（ B ）。

1. 将后绘制的柱形置于先绘制的柱形下方
2. 将后绘制的柱形置于先绘制的柱形上方
3. 将后绘制的柱形置于先绘制的柱形左方
4. 将后绘制的柱形置于先绘制的柱形右方
5. 下列选项中，程序运行的效果为圆环图的是（ D ）。

import numpy as np

import matplotlib.pyplot as plt

data = np.array([20, 50, 10, 15, 30, 55])

pie\_labels = np.array(['A', 'B', 'C', 'D', 'E', 'F'])

plt.pie(data, labels=pie\_labels)

plt.show()

import numpy as np

import matplotlib.pyplot as plt

data = np.array([20, 50, 10, 15, 30, 55])

pie\_labels = np.array(['A', 'B', 'C', 'D', 'E', 'F'])

plt.pie(data, radius=1.5, labels=pie\_labels)

plt.show()

import numpy as np

import matplotlib.pyplot as plt

data = np.array([20, 50, 10, 15, 30, 55])

pie\_labels = np.array(['A', 'B', 'C', 'D', 'E', 'F'])

plt.pie(data, radius=1.5, explode=[0, 0.2, 0, 0, 0, 0],labels=pie\_labels)

plt.show()

import numpy as np

import matplotlib.pyplot as plt

data = np.array([20, 50, 10, 15, 30, 55])

pie\_labels = np.array(['A', 'B', 'C', 'D', 'E', 'F'])

plt.pie(data, radius=1.5, wedgeprops={'width': 0.6},labels=pie\_labels)

plt.show()

1. 关于使用boxplot()函数绘制的箱形图，下列描述正确的是（ C ）。
2. 箱形图中异常值对应的符号默认为星号
3. 箱形图只能垂直摆放，无法水平摆放
4. 箱形图默认显示箱体
5. 箱形图默认不会显示异常值

# 第3章 图表辅助元素的定制

1. **填空题**
2. 图表的辅助元素是指除了根据数据绘制的 图形 之外的元素。
3. 图例是一个列举图表中各组图形 标识 方式的方框图。
4. 指向型注释文本是通过 指示箭头 的注释方式对图形进行解释的文本。
5. 参考线 是标记坐标轴上特殊值的一条直线。
6. matplotlib自带的引擎可以自动识别数学字符串，并将该数学字符串解析成相应的 数学公式 。
7. **判断题**
8. matplotlib中图例一直位于图表的右上方，它的位置是不可变的。（ x ）
9. 参考线可以为图形数据与特殊值之间的比较提供参考。（ √ ）
10. 坐标轴的标签代表图表名称，一般位于图表顶部居中的位置。（ X ）
11. 若坐标轴没有刻度，则无法显示网格。（ √ ）
12. 坐标轴的刻度范围取决于数据的最大值和最小值。（ √ ）
13. **选择题**
14. 关于图表辅助元素的说法中，下列描述错误的是（ B ）。
15. 标题一般位于图表的顶部中心，可以帮助用户理解图表要说明的内容
16. 参考区域是标记坐标轴上特殊值的一条直线
17. 图例由图例标识和图例项构成，可以帮助用户理解每组图表的含义
18. 表格主要用于强调比较难以理解的数据
19. 下列函数中，可以设置坐标轴刻度标签的是（ C ）。
20. xlim()
21. grid ()
22. xticks()
23. axhline()
24. 当pyplot使用legend()函数添加图例时，可以通过以下哪个参数控制图例的列数？（ B ）
25. loc
26. ncol
27. bbox\_to\_anchor
28. fancybox
29. 下列选项中，可以为图表添加一条值为1.5的水平参考线的是（ A ）。

plt.axhline(y=1.5, ls='--', linewidth=1.5)

plt.axhline(y=1, ls='--', linewidth=1.5)

plt.axvline(x=1.5, ls='--', linewidth=1.5)

plt.axvline(x=1, ls='--', linewidth=1.5)

1. 请阅读下面一段代码：

r'$\alpha^i < \beta^i$'

以上代码对应的数学公式为（ D ）。

# 第4章 图表样式的美化

1. **填空题**
2. matplotlib载入时会将包含全部配置项的字典赋值给变量 rcParams ，方便开发者采用访问字典的方式设置或获取配置项。
3. 在数据可视化中， 颜色 通常被用于编码数据的分类或定序属性。
4. 数据标记 指代表单个数据的圆点或其它符号等，用于帮助用户强调数据的位置。
5. matplotlib中文本都是一个 Text 类的对象。
6. matplotlib可以使用 use() 函数切换图表的主题风格。
7. **判断题**
8. matplotlib中线条的类型默认是长虚线。（ x ）
9. matplotlib中折线图的线条默认不显示数据标记。（ √）
10. 任何类型的图表都可以添加数据标记。（ x ）
11. matplotlib支持使用多种方式表示颜色。（ √ ）
12. matplotlib默认不支持显示中文。（ √ ）
13. **选择题**
14. 关于图表样式的说法，下列描述正确的是（ A ）。
15. matplotlib会读取matplotlibrc文件的配置信息以指定图表的默认样式
16. 图表的样式只能通过代码的方式进行修改
17. matplotlib不能修改matplotlibrc文件的配置信息
18. matplotlibrc文件一定保存在当前工作路径下
19. 下列选项中，表示的颜色不是黑色的是（ D ）。
20. 'k'
21. '#000000'
22. (0.0, 0.0, 0.0)
23. 'b'
24. 请阅读下面一段代码：

import matplotlib.pyplot as plt

plt.scatter([1, 2, 3], [3, 4, 5], s=10, marker='^')

plt.show()

以上代码运行后，展示了一个带有（ D ）标记的散点图。

1. 正方形
2. 星形
3. 菱形
4. 正三角形
5. 下列函数中，用于切换图表主题风格的是（ C ）。
6. turn()
7. change()
8. use()
9. replace()
10. 下列函数中，用于填充多边形的是（ A ）。
11. fill()
12. fill\_between()
13. fill\_betweenx()
14. fill\_betweeny()

# 第5章 子图的绘制及坐标轴共享

1. **填空题**
2. matplotlib可以规划整个画布 等分 成m\*n（行\*列）的矩阵区域。
3. matplotlib使用subplots()绘制多个子图时可以通过 sharex 参数控制是否共享x轴。
4. 约束布局 是通过一系列限制来确定画布中元素的位置的方式。
5. matplotlib的 gridspec 是专门指定画布中子图位置的模块。
6. **判断题**
7. subplot(223)与subplot(2, 2, 3)是等价的。（√）
8. matplotlib使用subplot()可以一次性绘制多个子图。（X）
9. 同一画布的多个子图可以共享同方向的坐标轴。（√）
10. matplotlib默认未启用约束布局。（√）
11. 紧密布局适用于图表的所有元素，可以调整所有元素的位置。（X）
12. **选择题**
13. 下列函数中，可以一次性绘制多个子图的是（D）。
14. subplot()
15. subplot2grid()
16. twinx()
17. subplots()
18. 请阅读下面一段程序：

%matplotlib auto

import matplotlib.pyplot as plt

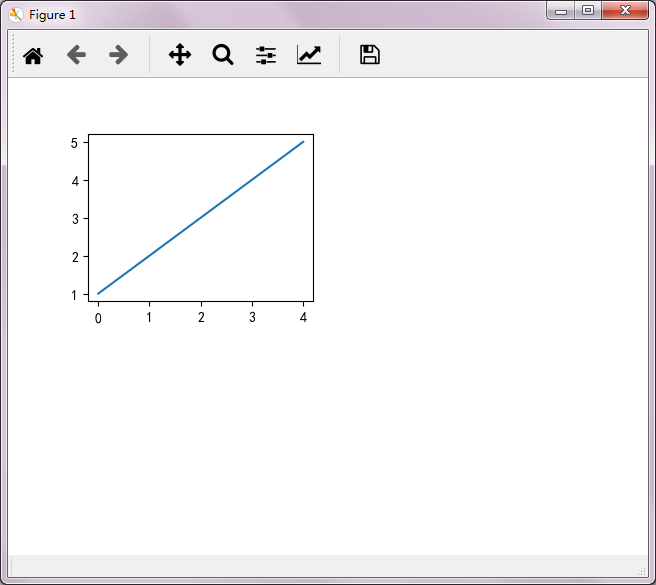
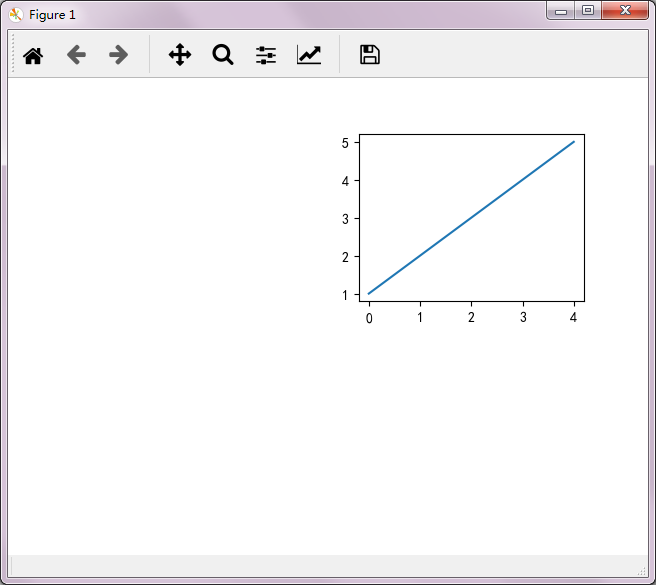
ax\_one = plt.subplot(223)

ax\_one.plot([1, 2, 3, 4, 5])

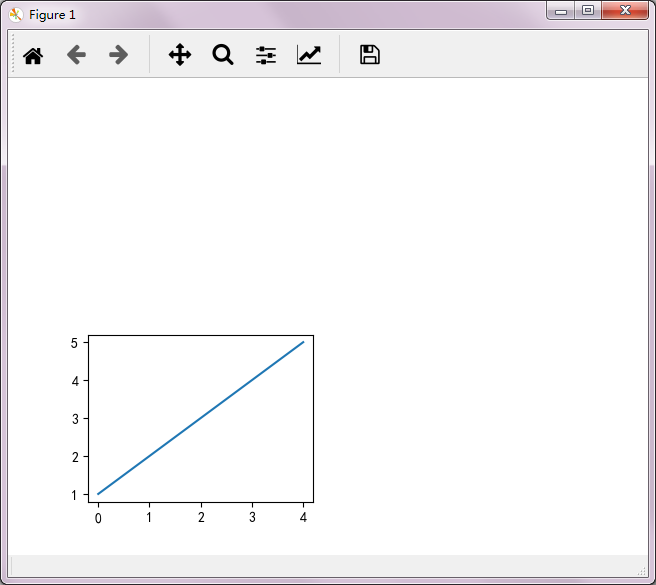
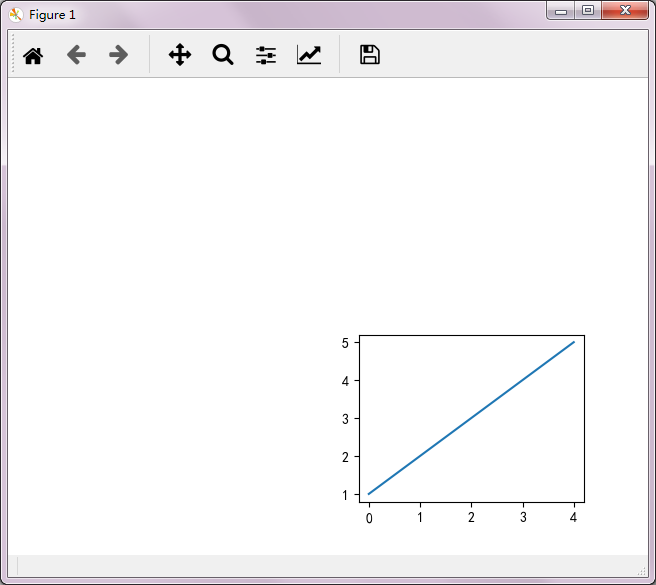
plt.show()

运行程序，效果为（C）。

1. B.

1. C.

1. 请阅读下面一段程序：

import matplotlib.pyplot as plt

plt.subplots(2, 2, sharex=\_\_\_\_\_\_\_\_)

plt.show()

以上程序的横线处填充以下哪个取值，可以共享每列子图的坐标轴？（A）

1. 'col'
2. 'row'
3. 'all'
4. 'none'
5. 下列选项中，可以实现紧密布局的是（C）。
6. twinx()
7. constrained\_layout()
8. tight\_layout()
9. GridSpec()
10. 当matplotlib使用GridSpec()自定义布局结构时，可以通过（D）参数控制子图之间的间隙。
11. nrows
12. ncols
13. figure
14. wspace

# 第6章 坐标轴的定制

1. **填空题**
2. 坐标轴包括轴脊、刻度，其中刻度线可以细分为 主刻度线 和次刻度线。
3. 坐标轴一般将 轴脊 作为刻度的载体。
4. pyplot可以使用 axis() 函数显示或隐藏的全部轴脊。
5. **判断题**
6. matplotlib的坐标轴默认隐藏次刻度线。（√）
7. matplotlib可以使用Formatter的子类调整刻度的位置。（×）
8. matplotlib中刻度线的方向只能朝外。（×）
9. **选择题**
10. 下列选项中，可以获取坐标轴全部轴脊的是（C）。
11. xaxis
12. yaxis
13. spines
14. ticks
15. 下列方法中，用于设置主刻度标签格式的是（C）。
16. set\_major\_locator()
17. set\_minor\_locator()
18. set\_major\_formatter()
19. set\_minor\_formatter()
20. 请阅读下面一段代码：

line\_loc = LinearLocator(numticks=3)

ax.xaxis.set\_major\_locator(line\_loc)

下列哪个选项最有可能是x轴的效果？（D）









1. 下列选项中，可以隐藏坐标轴上轴脊的是（A）。

ax.spines['top'].set\_color('none')

ax.spines['right'].set\_color('none')

ax.spines['bottom'].set\_color('none')

ax.spines['left'].set\_color('none')

1. 下列方法中，用于移动轴脊位置的是（B）。
2. set\_color()
3. set\_position()
4. set\_ticks\_position()
5. set\_yticklabels()

# 第7章 绘制3D图表和统计地图

1. **填空题**
2. mplot3d是matplotlib中专门绘制 3D图标 \_\_的工具包。
3. FuncAnimation通过重复地调用同一 函数 \_\_来制作动画。
4. basemap工具包中默认使用的地图投影是 cyl \_\_。
5. basemap工具包中包含一个表示基础地图背景的 Basemap \_\_类。
6. **判断题**
7. matplotlib只能绘制2D图表。（X）
8. FuncAnimation是一个动画基类。（X）
9. 创建Basemap类的对象时可以指定地图投影的类型和要处理的地球区域。（√）
10. **选择题**
11. 下列选项中，用于绘制统计地图的是（B）。
12. mplot3d
13. basemap
14. animation
15. ticker
16. 下列方法中，用于绘制3D曲面图的是（C）。
17. plot()
18. plot\_wireframe()
19. plot\_surface()
20. plot\_trisurf()
21. 关于animation模块的说法中，下列描述错误的是（D）。
22. Animation类针对不同的行为派生了不同的子类
23. FuncAnimation类表示基于重复调用一个函数的动画
24. ArtistAnimation类表示基于一组Artist对象的动画
25. ArtistAnimation是一个动画基类
26. 请阅读下面一段代码：

map = Basemap(projection='stere', lat\_0=90, lon\_0=-105, llcrnrlat=23.41,

urcrnrlat=45.44, llcrnrlon=-118.67, urcrnrlon=-64.52,

rsphere=6371200., resolution='l', area\_thresh=10000)

以上代码中地图背景使用的投影类型为（B）。

1. Cylindrical Equal Area
2. Stereographic
3. Cylindrical Equidistant
4. South-Polar Stereographic
5. 下列方法中，用于绘制地图纬度线的是（A）。
6. drawparallels()
7. drawmeridians()
8. drawcoastlines()
9. drawrivers()

# 第8章 使用matplotlib绘制高级图表

1. **填空题**
2. 哑铃图主要用于展示两个数据点之间的 变化 。
3. hierarchy模快可以轻松地生成聚类数据并绘制 树状图 。
4. 人口金字塔图一般以 年龄 为纵轴、以人口数为横轴。
5. pyplot可以使用 contour() 函数绘制等高线图。
6. 棉棒图的 标记符号 表示数据点的数值。
7. **判断题**
8. 桑基图中所有主支宽度的总和等于所有分支宽度的总和。（√）
9. 华夫饼图一般由100个方格组成，其中每个方格代表2%。（X）
10. 矢量场的强度只能使用线条的长度来表示。（X）
11. 甘特图可以展示每个项目的进展情况。（√）
12. 漏斗图可以展示业务流程中数据的变化。（√）
13. **选择题**
14. 关于高级图表的说法中，下列描述正确的是（C）。
15. 树状图可以展示部分与整体的比例
16. 桑基图中每条分支的长度代表着数据流量的大小
17. 矢量场流线图可以表现矢量场的流态
18. 棉棒图主要比较数据点到基线之间线段的长度
19. 下列字段中，可以访问Sankey对象的中心标签的是（C）。
20. patch
21. flows
22. text
23. texts
24. Sankey类对象在添加完数据之后需要调用（B）方法完成绘制。
25. add()
26. finish()
27. over()
28. show()
29. 下列选项中，可以将层次聚类数据绘制为树状图的是（A）。
30. dendrogram()
31. linkage()
32. finish()
33. stem()
34. 下列选项中，哪个的效果是一个10行5列且以垂直方向绘制的华夫饼图？（A）

plt.figure(FigureClass=Waffle, rows=10, columns=5,

values=[45, 55], vertical=True)

plt.figure(FigureClass=Waffle, rows=10, columns=10,

values=[45, 55], vertical=True)

plt.figure(FigureClass=Waffle, rows=10, columns=5, values=[45, 55])

plt.figure(rows=10, columns=5, values=[45, 55], vertical=True)

# 第9章 可视化后起之秀——pyecharts

1. **填空题**
2. pyecharts 是一个针对Python用户开发的、用于生成 Echarts \_图表的库。
3. 链式调用是一种简化 同一对象 \_多次访问属性或调用方法的编码方式。
4. render\_embed()方法用于将图表 渲染 \_到Jupyter Notebook工具中。
5. pyecharts.charts的Tab类表示以 选项卡 \_形式显示的组合图表。
6. pyecharts将图表主题封装为全局变量 ThemeType \_引用类的属性。
7. **判断题**
8. 链式调用可以避免多次重复使用同一个对象变量。（√）
9. pyecharts可以将系列配置项传入set\_global\_options()方法来配置图表。（X）
10. pyecharts不允许用户使用自定义的主题风格。（X）
11. pyecharts可以轻松地整合Web框架。（√）
12. Timeline类表示时间线轮播的组合图表，它通过点击时间线的时间节点来切换图表。（√）
13. **选择题**
14. 下列选项中，可以创建漏斗图的是（C）。
15. Scatter
16. Map
17. Funnel
18. Sankey
19. 关于配置项的说法中，下列描述正确的是（D）。
20. 全局配置项是一些针对图表特定元素属性的配置项
21. 系列配置项是一些针对图表通用属性的配置项
22. pyecharts可以将InitOpts实例传入set\_global\_options()方法为图表设置初始化配置项。
23. pyecharts可以将系列配置项传入add()或add\_xx()方法来配置图表
24. 下列组合多图的方式中，可以采用左右布局的形式显示多个图表的是（ A ）。
25. 并行多图
26. 顺序多图
27. 选项卡多图
28. 时间线轮播多图
29. 下列方法中，可以将图表渲染到Jupyter Notebook工具的是（ B ）。
30. render()
31. render\_notebook()
32. render\_embed()
33. -load\_javascript()
34. 下列选项中，可以修改图表主题风格的是（ A ）。

Bar(init\_opts=opts.InitOpts(theme=ThemeType.ROMA))

Bar(init\_opts=opts.InitOpts(theme=ROMA))

Bar(theme=ROMA)

Bar(theme=ThemeType.ROMA)