数据可视化 课堂笔记

1. 数据筛选和处理

Numpy

方法	描述
np.array()	数组
np.arange(start,end,step)	按步长step生成数组
np.linspace(start,end,num)	按数量num生成数组
np.info()	查看numpy帮助
np.reshape()	重构矩阵结构生成新数组对象
np.resize()	重构矩阵结构不生成新数组对象
np.ones()	幺矩阵
np.zeros()	零矩阵
np.eye()	单位矩阵
np.full()	填充矩阵
np.ones_like()	和给定矩阵形状相同的幺矩阵
np.zeros_like()	和给定矩阵形状相同的零矩阵
np.full_like()	和给定矩阵形状相同的填充矩阵
np.square()、np.power()、np.log()、...	幂指对数等运算
np.sin()、np.cos()...	三角函数运算
np.floor()、np.ceil()、np.round() ...	取整运算
np.amax()、np.amin()、np.sum()、np.median()...	聚合运算
np.random()	随机运算
np.loadtxt()、np.savetxt()	文本文件存取
np.tofile()、np.fromfile()	数据文件存取

属性	描述
shape	矩阵形状
ndim	矩阵维度
size	元素数量
itemsize	元素数据大小
dtype	元素数据类型

Pandas

方法	描述
pd.Series	生成系列Series对象
pd.DataFrame()	生成数据表DataFrame对象
pd.to_csv()、pd.read_csv()	读写csv文件
pd.to_excel()、pd.read_excel()	读写excel文件
pd.to_json()、pd.read_json()	读写json文件

2. Matplotlib.pyplot模块

plot()函数 - 绘制线图

```
plt.plot(x, y, 'xxx', linestyle=, marker=, color=, linewidth=, label= )
```

- x：点的横坐标，可迭代对象
- y：点的纵坐标，可迭代对象
- linestyle：线的样式，字符串

linestyle	线形
'-'	实线
'--'	虚线
'-.'	点划线
':'	点虚线
' '	无线

- linewidth：线的粗细，数值
- marker：点的样式，字符串

marker	标记点
'.'	点
','	像素
'^' 'v' '>' '<'	上下左右三角形
'1' '2' '3' '4'	上下左右三叉线
'o'	圆形
's' 'D'	方形
'p'	五边形
'h' 'H'	六边形
'*'	五角星
'+' 'x'	十字交叉
'_' '|'	横线和竖线

- markersize： 点的大小，数值
- alpha ： 透明度，0~1数值
- color： 颜色，字符串

color	字符串
'r'， '#FF0000'	红
'g'， '#008000'	绿
'b'， '#0000FF'	蓝
'y'， '#FFFF00'	黄
'c'， '#00FFFF'	青
'm'，'#FF00FF'	品
'k'， '#000000'	黑
'w'， '#FFFFFF'	白

- label：图标签，可用作legend文字

legend()函数 - 生成图例

```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(-10,10,20)
y1 = x**2 + 2*x + 1
y2 = - x**2 - 2*x - 21
plt.plot(x,y1,linestyle='-',linewidth=3, marker='o',color='r',label='y1')
plt.plot(x,y2,linestyle='--',marker='.',color='b',label='y2')
```

```
# 按位置设置标签和字号
plt.legend(labels=['y1','y2'], fontsize=15)

# 标签支持公式，参考LaTex公式语法
plt.legend(labels=['$y=x^2+2x+1$','$y=-x^2-2x-21$'])

# 指定图例位置
plt.legend(loc='upper center')

# 不显示边框
plt.legend(frameon=False)

# 分两列显示图例
plt.legend(ncol=2)

# 显示图例边框阴影
plt.legend(shadow=True)
```

常用参数	描述
labels	图例标签列表
loc	图例位置
fontsize	字体大小
frameon	是否显示图例边框
ncol	图例列数，默认1列
title	图例标题
shadow	是否显示图例阴影

title()函数 - 设置标题

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-10,10,21)
y1 = x**2 + 2*x + 1
y2 = - x**2 - 2*x -21
plt.plot(x,y1,linestyle='-',linewidth=3,
marker='.',markersize=10,color='#FF0000')
plt.plot(x,y2,linestyle='--',marker='D')

plt.title('This is the title',fontstyle='oblique',fontweight='bold')
// 中文标题需设置中文字体字典
Fangsong = {'family' : 'Fangsong','weight' : 'normal','size' : 15}
plt.title('这是一个标题',fontdict=Fangsong, backgroundcolor='k',color='w')
```

常用参数	描述
fontsize	设置字体大小，默认12
fontweight	设置字体粗细，可选参数 ['light', 'normal', 'medium', 'semibold', 'bold', 'heavy', 'black']
fontstyle	设置字体类型，可选参数['normal' | 'italic' | 'oblique']
verticalalignment	设置水平对齐方式，可选参数 ['center' | 'top' | 'bottom' | 'baseline']
horizontalalignment	设置垂直对齐方式，可选参数 ['left' | 'right' | 'center']
rotation	旋转角度，可选参数为['vertical' | 'horizontal']也可以为角度
alpha	透明度，参数值0至1之间

xlabel()、ylabel()函数 - 设置x、y轴标签

```
// 常用参数
plt.xlabel('$X$',horizontalalignment='right',verticalalignment='top')
plt.ylabel('$y=x^2+2x+1$',horizontalalignment='left',verticalalignment='center')

// 设置支持中文字体
Fangsong = {'family' : 'Fangsong','weight' : 'normal','size' : 15}
plt.xlabel('X轴',fontdict=Fangsong)
plt.ylabel('Y轴',rotation=0,fontdict=Fangsong)
```

常用参数	描述
‘TEXT’	显示标签文字
size	字号
rotation	旋转
horizontalalignment	水平对齐方式
verticalalignment	垂直对齐方式
fontdict	字体(字典类型)

> **font:**
>
> ```
> # 字体字典：仿宋体
> Fangsong = {'family' : 'Fangsong','weight' : 'normal','size' : 15,}
> # 字体字典：简黑体
> Simhei = {'family' : 'SimHei','weight' : 'normal','size' : 15,}
>
> # matplotlib设置全局中文字体简体黑
> plt.rcParams['font.sans-serif'] = ['SimHei']
> # matplotlib正常显示中文负号
> plt.rcParams['axes.unicode_minus'] = False
> ```

xlim()、ylim()函数 - 设置x、y轴显示范围

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-10,10,21)
y1 = x**2 + 2*x + 1
y2 = - x**2 - 2*x -21
plt.plot(x,y1,linestyle='-',linewidth=3,
marker='.',markersize=10,color='#FF0000')
plt.plot(x,y2,linestyle='--',marker='D')
plt.xlim(0,5)
plt.ylim(-100,100)
```

xticks()、yticks()函数 - 设置x、y轴刻度

grid()函数 - 设置网格线（与xticks、yticks对齐）

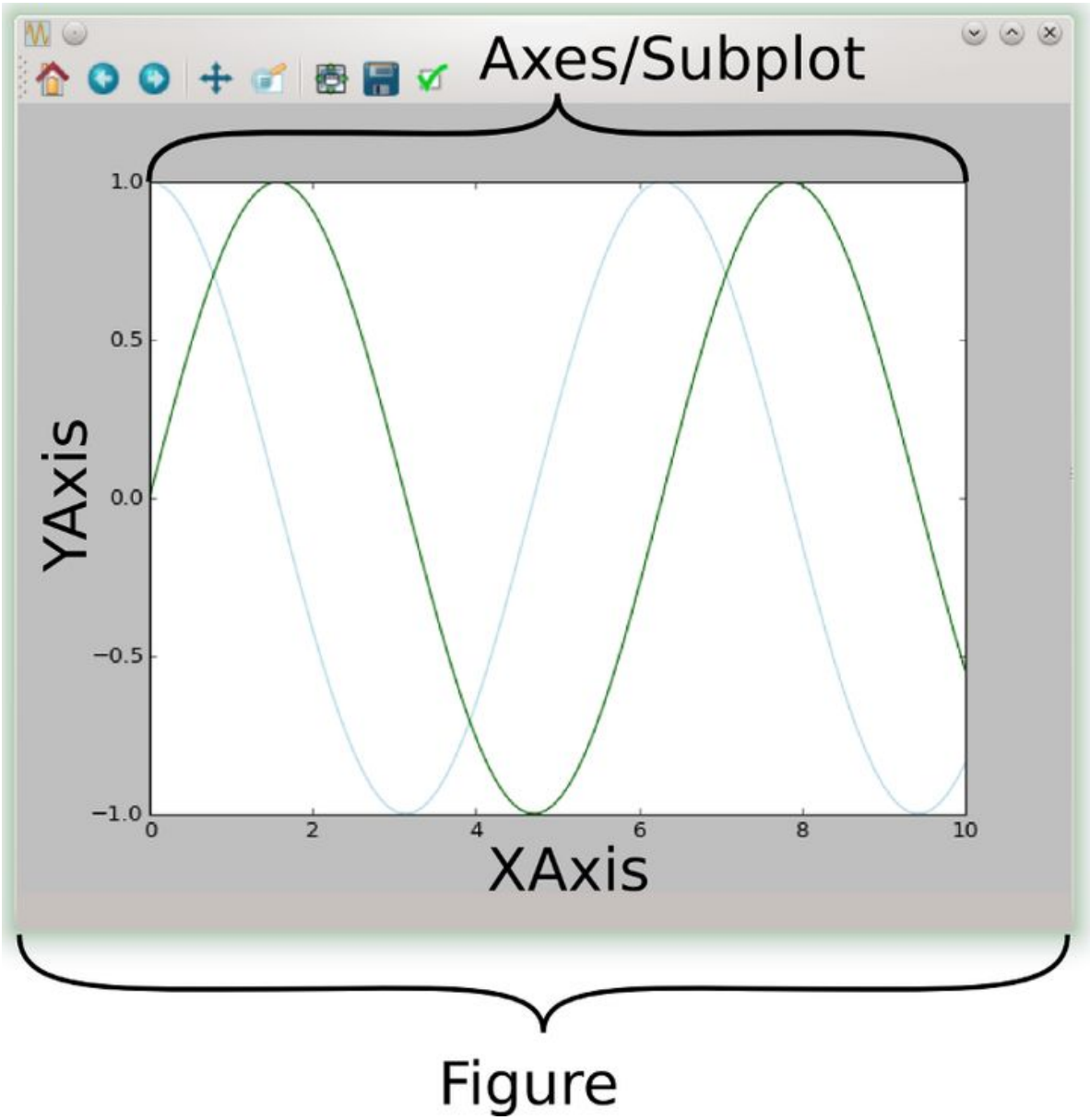
```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-10,10,21)
y1 = x**2 + 2*x + 1
y2 = -x**2 - 2*x -21
plt.plot(x,y1,linestyle='-',linewidth=3,
marker='.',markersize=10,color='#FF0000')
plt.plot(x,y2,linestyle='--',marker='D')
plt.xlim(0,5)
plt.ylim(-100,100)
# 设置刻度
plt.xticks(x, rotation=-90)
# 中文刻度显示需使用中文字体
plt.xticks(range(5),labels=['〇','一','二','三','四'], fontproperties='SimHei')
# 设置网格
plt.grid()
```

figure和axes

- **figure是作图的画布** : matplotlib.figure.Figure
- **你可以在figure上面铺展axes, 事实上, 你画的图其实都是画在axes上的** : matplotlib.pyplot.axes，**axes其实是在一幅画布上, 规划出的一个个科学作图的坐标轴系统**

```
plt.gcf() # 意为, get current figure
plt.gca() # 意为, get current axes
```



subplots()函数 - 生成子图矩阵框架

subplots()函数生成一个figure对象和figure容器中的axes坐标轴，如果有多个坐标轴，这些坐标轴被放到一个列表中

```
figure, axes = plt.subplots(n,m)
```

figure

常用参数	描述
num	图编号
figsize	图大小
facecolor	窗口的背景颜色
edgecolor	窗口边框颜色
frameon	是否绘制窗口的图框

```
fig = plt.figure(figsize=(6,3))
fig.add_subplot(121) # 1行2列，第1个
fig.add_subplot(122) # 1行2列，第2个

fig.add_axes(121)
fig.add_axes(122)
```

bar() 函数 - 绘制条形图/柱状图

```
bar(x, height, width=0.8, bottom=None, *, align='center', data=None, **kwargs)
```

常用参数	描述
x	横坐标
height	条形高度
width	条形宽度
bottom	条形的起点
align	条形的对齐点
color	条形颜色
tick_label	下标签
orientation	竖条还是横条

scatter()函数 - 绘制散点图

```
scatter(x, y, s=None, c=None, marker=None, cmap=None, norm=None, vmin=None,
vmax=None,
alpha=None, linewidths=None, verts=None, edgecolors=None, hold=None, data=None,
**kwargs)
```

常用参数	描述
x,y	坐标向量
s	size 标记大小
c	color 标记颜色
marker	标记样式
cmap	设置色彩盘
norm	设置亮度，0~1
alpha	透明度，0~1
orientation	竖条还是横条
linewidths	线宽
edgecolors	轮廓颜色

pie() 函数 - 绘制饼图

```
pie(x, explode=None, labels=None, colors=None, autopct=None,
    pctdistance=0.6, shadow=False, labeldistance=1.1, startangle=None,
    radius=None, counterclock=True, wedgeprops=None, textprops=None,
    center=(0, 0), frame=False, rotatelabels=False, hold=None, data=None)
```

常用参数	描述
x	(每一块)的比例，如果sum(x) > 1会使用sum(x)归一化；
labels	(每一块)饼图外侧显示的说明文字；
explode	(每一块)离开中心距离；
startangle	起始绘制角度,默认图是从x轴正方向逆时针画起,如设定=90则从y轴正方向画起；
shadow	在饼图下面画一个阴影。默认值：False，即不画阴影；
labeldistance	label标记的绘制位置,相对于半径的比例，默认值为1.1, 如<1则绘制在饼图内侧；
autopct	控制饼图内百分比设置,可以使用format字符串或者format function '%1.1f'指小数点前后位数(没有用空格补齐)；
pctdistance	类似于labeldistance,指定autopct的位置刻度,默认值为0.6；
radius	控制饼图半径，默认值为1；
counterclock	指定指针方向；布尔值，可选参数，默认为：True，即逆时针。将值改为False即可改为顺时针。
wedgeprops	字典类型，可选参数，默认值：None。参数字典传递给wedge对象用来画一个饼图。例如：wedgeprops={'linewidth' 3}设置wedge线宽为3。
textprops	设置标签（labels）和比例文字的格式；字典类型，可选参数，默认值为：None。传递给text对象的字典参数。
center	浮点类型的列表，可选参数，默认值：(0,0)。图标中心位置。
frame	布尔类型，可选参数，默认值：False。如果是true，绘制带有表的轴框架。
rotatelabels	布尔类型，可选参数，默认为：False。如果为True，旋转每个label到指定的角度。

hist() 函数 - 绘制直方图

参数	描述
x	数据集，最终的直方图将对数据集进行统计
bins	统计的区间分布，分多少个桶，在range区间上的分桶数量
range	tuple, 显示的区间
density	bool，默认为false，显示的是频数统计结果，为True则显示频率统计结果，这里需要注意，频率统计结果=区间数目/(总数*区间宽度)，和normed效果一致，官方推荐使用density
histtype	可选{'bar', 'barstacked', 'step', 'stepfilled'}之一，默认为bar，推荐使用默认配置，step使用的是梯状，stepfilled则会对梯状内部进行填充，效果与bar类似
align	可选{'left', 'mid', 'right'}之一，默认为'mid'，控制柱状图的水平分布，left或者right，会有部分空白区域，推荐使用默认
log	bool，默认False,即y坐标轴是否选择指数刻度
stacked	bool，默认为False，是否为堆积状图

3. Seaborn 模块

Relational plots 关系图

`relplot`	**Figure-level interface for drawing relational plots onto a FacetGrid.**
`scatterplot`	Draw a scatter plot with possibility of several semantic groupings.
`lineplot`	Draw a line plot with possibility of several semantic groupings.

Distribution plots 分布图

`displot`	**Figure-level interface for drawing distribution plots onto a FacetGrid.**
`histplot`	Plot univariate or bivariate histograms to show distributions of datasets.
`kdeplot`	Plot univariate or bivariate distributions using kernel density estimation.
`ecdfplot`	Plot empirical cumulative distribution functions.
`rugplot`	Plot marginal distributions by drawing ticks along the x and y axes.
`distplot`	DEPRECATED: Flexibly plot a univariate distribution of observations.

Categorical plots 分类图

catplot	**Figure-level interface for drawing categorical plots onto a FacetGrid.**
stripplot	Draw a scatterplot where one variable is categorical.
swarmplot	Draw a categorical scatterplot with non-overlapping points.
boxplot	Draw a box plot to show distributions with respect to categories.
violinplot	Draw a combination of boxplot and kernel density estimate.
boxenplot	Draw an enhanced box plot for larger datasets.
pointplot	Show point estimates and confidence intervals using scatter plot glyphs.
barplot	Show point estimates and confidence intervals as rectangular bars.
countplot	Show the counts of observations in each categorical bin using bars.

Regression plots 回归图

lmplot	**Plot data and regression model fits across a FacetGrid.**
regplot	Plot data and a linear regression model fit.
residplot	Plot the residuals of a linear regression.

Matrix plots 矩阵图

heatmap	**Plot rectangular data as a color-encoded matrix.**
clustermap	Plot a matrix dataset as a hierarchically-clustered heatmap.

Joint grids 联合图

jointplot	**Draw a plot of two variables with bivariate and univariate graphs.**
JointGrid	Grid for drawing a bivariate plot with marginal univariate plots.
JointGrid.plot	Draw the plot by passing functions for joint and marginal axes.
JointGrid.plot_joint	Draw a bivariate plot on the joint axes of the grid.
JointGrid.plot_marginals	Draw univariate plots on each marginal axes.

Pair grids

`pairplot`	**Plot pairwise relationships in a dataset.**
`PairGrid`	Subplot grid for plotting pairwise relationships in a dataset.
`PairGrid.map`	Plot with the same function in every subplot.
`PairGrid.map_diag`	Plot with a univariate function on each diagonal subplot.
`PairGrid.map_offdiag`	Plot with a bivariate function on the off-diagonal subplots.
`PairGrid.map_lower`	Plot with a bivariate function on the lower diagonal subplots.
`PairGrid.map_upper`	Plot with a bivariate function on the upper diagonal subplots.

关系数据可视化 （relplot())

relplot() 关系数据可视化主要有两种图折线图lineplot和散点图 scatterplot

- `scatterplot()` (with `kind="scatter"`; the default)

```
import seaborn as sns
sns.set_theme(style="darkgrid")

tips = sns.load_dataset("tips")  # 导入seaborn自带数据集
print(tips.head(5))
sns.relplot(x="total_bill", y="tip", data=tips);
```

```
sns.relplot(x="total_bill", y="tip", hue="smoker", data=tips);
```

- `lineplot()` (with `kind="line"`)

```
import seaborn as sns
sns.set_theme(style="darkgrid")

flights = sns.load_dataset("flights")
print(flights.head(5))
may_flights = flights.query("month == 'May'")
sns.lineplot(data=may_flights, x="year", y="passengers")
```

```
flights_wide = flights.pivot("year", "month", "passengers")
flights_wide.head()
sns.lineplot(data=flights_wide["May"])
```