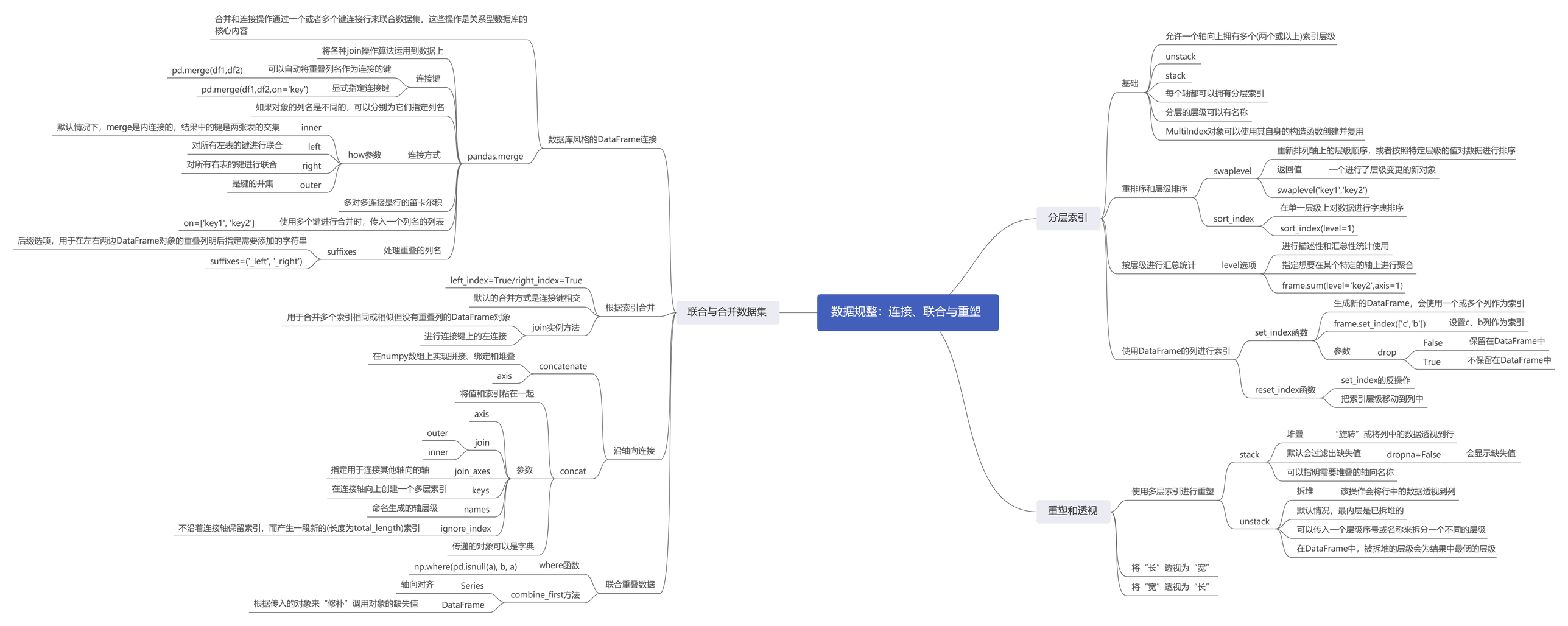
数据规整：连接、联合与重塑
联合与合并数据集
数据库风格的DataFrame连接
合并和连接操作通过一个或者多个键连接行来联合数据集。这些操作是关系型数据库的核心内容
pandas.merge
将各种join操作算法运用到数据上
连接键
可以自动将重叠列名作为连接的键
pd.merge(df1,df2)
显式指定连接键
pd.merge(df1,df2,on='key')
如果对象的列名是不同的，可以分别为它们指定列名
连接方式
how参数
inner
默认情况下，merge是内连接的，结果中的键是两张表的交集
left
对所有左表的键进行联合
right
对所有右表的键进行联合
outer
是键的并集
多对多连接是行的笛卡尔积
使用多个键进行合并时，传入一个列名的列表
on=['key1', 'key2']
处理重叠的列名
suffixes
后缀选项，用于在左右两边DataFrame对象的重叠列明后指定需要添加的字符串
suffixes=('_left', '_right')
根据索引合并
left_index=True/right_index=True
默认的合并方式是连接键相交
join实例方法
用于合并多个索引相同或相似但没有重叠列的DataFrame对象
进行连接键上的左连接
沿轴向连接
concatenate
在numpy数组上实现拼接、绑定和堆叠
axis
concat
将值和索引粘在一起
参数
axis
join
outer
inner
join_axes
指定用于连接其他轴向的轴
keys
在连接轴向上创建一个多层索引
names
命名生成的轴层级
ignore_index
不沿着连接轴保留索引，而产生一段新的(长度为total_length)索引
传递的对象可以是字典
联合重叠数据
where函数
np.where(pd.isnull(a), b, a)
combine_first方法
Series
轴向对齐
DataFrame
根据传入的对象来“修补”调用对象的缺失值
分层索引
基础
允许一个轴向上拥有多个(两个或以上)索引层级
unstack
stack
每个轴都可以拥有分层索引
分层的层级可以有名称
MultiIndex对象可以使用其自身的构造函数创建并复用
重排序和层级排序
swaplevel
重新排列轴上的层级顺序，或者按照特定层级的值对数据进行排序
返回值
一个进行了层级变更的新对象
swaplevel('key1','key2')
sort_index
在单一层级上对数据进行字典排序
sort_index(level=1)
按层级进行汇总统计
level选项
进行描述性和汇总性统计使用
指定想要在某个特定的轴上进行聚合
frame.sum(level='key2',axis=1)
使用DataFrame的列进行索引
set_index函数
生成新的DataFrame，会使用一个或多个列作为索引
frame.set_index(['c','b'])
设置c、b列作为索引
参数
drop
False
保留在DataFrame中
True
不保留在DataFrame中
reset_index函数
set_index的反操作
把索引层级移动到列中
重塑和透视
使用多层索引进行重塑
stack
堆叠
“旋转”或将列中的数据透视到行
默认会过滤出缺失值
dropna=False
会显示缺失值
可以指明需要堆叠的轴向名称
unstack
拆堆
该操作会将行中的数据透视到列
默认情况，最内层是已拆堆的
可以传入一个层级序号或名称来拆分一个不同的层级
在DataFrame中，被拆堆的层级会为结果中最低的层级
将“长”透视为“宽”
将“宽”透视为“长”