In [ ]:

```
bj09idx.reindex(idx)
```

In [ ]:

```
bj09idx[bj09idx.index.duplicated()]
```

In [ ]:

```
bj09idx['2009-03-08']
```

In [ ]:

```
bj09idx[~bj09idx.index.duplicated()].reindex(idx)
```

In [ ]:

```
bj09idx[~bj09idx.index.duplicated()].reindex(idx, method = 'bfill')
```

9.4.3 序列数值平移

df.shift(

periods = 1 : 希望移动的周期数
freq : 时间频度字符串
axis = 0

)

In [ ]:

```
bj08idx.shift(3)
```

9.5 实战：建立时间索引

要求:

自行练习分别使用Date (LST)和年、月、日、时变量建立DatetimeIndex
尝试只使用年、月、日建立Period对象，然后转换为DatetimeIndex
基于DatetimeIndex，进一步完成原先在第7章练习中完成过的任务
- 计算出每年PM2.5的平均值、中位数、最大值、最小值
- 计算出每年PM2.5值大于200、300、500的天数
- 将PM2.5数据整理为以年为行，月为列，单元格为最大值的宽表形式
- 将2009年和2012年的数据分别提取出来，然后合并为一个数据框
- 将数据转换为每年一列的宽表格式

10 数据的图形展示

10.1 配置绘图系统环境

In [ ]:

```
df2['总分'].plot.box(title='总分的箱图分布', ylim=(60, 80))
```

In [ ]:

```
# 图形在Pandas页面同步显示的问题
%matplotlib inline
```

In [ ]:

```
# 中文字符兼容问题
import matplotlib
matplotlib.rcParams['font.sans-serif'] = ['SimHei']
```

In [ ]:

```
# 绘图功能的进一步美化和功能增强包，参考http://seaborn.pydata.org/
import seaborn
seaborn.set_style("whitegrid")
# 注意有中文兼容问题，需要重新导入中文设定！
```

In [ ]:

```
# 进一步在一些细节上的美化和优化
import matplotlib.pyplot as plt
plt.figure()
```

10.2 绘图命令基本框架

df.plot(

```
    绘图用数据
        data : 数据框
        x = None: 行变量的名称/顺序号
        y = None : 列变量的名称/顺序号

    kind = 'line' : 需要绘制的图形种类
        'line' : line plot (default)
        'bar' : vertical bar plot
            'barh' : horizontal bar plot
        'hist' : histogram
        'box' : boxplot
        'kde' : Kernel Density Estimation plot
            'density' : same as 'kde'
        'area' : area plot
        'pie' : pie plot
        'scatter' : scatter plot
            'hexbin' : hexbin plot

    各种辅助命令
        figsize : a tuple (width, height) in inches
        xlim / ylim : X/Y轴的取值范围, 2-tuple/list格式
        logx / logy / loglog = False : 对X/Y/双轴同时使用对数尺度
        title : string or list
        Alpha : 图形透明度, 0-1

    图组命令
        subplots = False : 是否分图组绘制图形
        sharex : 是否使用相同的X坐标
            ax = None时, 取值为True, 否则取值为False
        sharey = False : 是否使用相同的Y坐标
        ax = None : 需要叠加的 matplotlib绘图对象
)
```

图形种类的等价写法

```
    df.plot.kind()
```

In [ ]:

```
df2['总分'].plot.box(title='总分的箱图分布', ylim=(60, 80))
```

In [ ]:

```
# 考察过去一段时间的数据分布
bj08[-100:].Value.plot(figsize=(12,8))
```

In [ ]:

```
bj.groupby(bj.Year).Value.plot() # 有无seaborn修饰时的结果不同
```

10.3 条图

需要先自行完成数据汇总，绘图函数只能完成绘图工作

10.3.1 简单条图

In [ ]:

```
# 条图
pd.value_counts(df2.类型).plot.bar()
```

In [ ]:

```
pd.value_counts(df2.类型).plot.barh()
```

10.3.2 复式条图

行索引构成大分组，变量列构成小分组

In [ ]:

```
import numpy as np

dfnew = pd.DataFrame(np.random.rand(10, 4), columns=['a', 'b', 'c', 'd'])
print(dfnew)
dfnew.plot.bar()
```

10.3.3 分段条图

plot.bar(stacked = True)

In [ ]:

```
dfnew.plot.bar(stacked = True)
```

10.4 直方图

10.4.1 简单直方图

plot.hist(

by : 在df中用于分组的变量列(绘制为图组)
bins = 10 : 需要拆分的组数

)

In [ ]:

```
#直方图
df2.总分.plot.hist(bins=20)
```

10.4.2 直方图图组

hist(

    by : 在df中用于分组的变量列(绘制为图组)

)

In [ ]:

```
df2.总分.hist(by = df2.类型, bins=20)
```

10.5 饼图

注意是每行代表一个饼块的数据结构，因此需要先自行汇总好变量频数

plot.pie(

    y : 指定需要绘制的变量列名称
    subplots = False : 多个变量列时要求分组绘图
    Labels
    Colors

)

10.5.1 简单饼图

In [ ]:

```
df2.类型.value_counts().plot.pie()
```

In [ ]:

```
pd.value_counts(df2.类型).plot.pie()
```

In [ ]:

```
df2.loc[:9,['名次','总分']].plot.pie(subplots = True, figsize=(8, 4))
```

10.5.2 Semicircle

当数值总和小于1时，绘制的是semicircle

In [ ]:

```
pd.Series([0.1,0.2,0.1,0.3]
          , index=['a', 'b', 'c', 'd']).plot.pie(figsize=(6, 6))
```

10.6 箱图

plot.box(

    vert = True : 是否纵向绘图

)

boxplot(

by : 在df中用于分组的变量列(绘制为图组)

)

In [ ]:

```
df2.plot.box(vert = False)
```

In [ ]:

```
df2.boxplot(by='类型')
```

10.7 散点图

plot.scatter(

s : 控制散点大小的变量列，不能使用df中的简写方式指定
c : 控制散点颜色的变量列

)

10.7.1 简单散点图

In [ ]:

```
df2.plot.scatter('总分', '名次')
```

10.7.2 对散点图进行修饰

In [ ]:

```
df2.plot.scatter(x= '总分', y='名次', c='名次')
```

In [ ]:

```
df2.plot.scatter(x= '总分', y='名次', s=df2.名次)
```

10.7.3 重叠散点图

使用matplotlib的ax对象进行图形叠加

```
ax = df.plot.scatter(x='', y='', color='', label='');
df.plot.scatter(x='', y='', color='', label='', ax=ax);
```

In [ ]:

```
ax = df2.plot.scatter(x='总分', y='名次'
                      , color='DarkBlue', label='Group 1');
df2.plot.scatter(x='名次', y='总分'
                 , color='DarkGreen', label='Group 2', ax = ax);
```

10.8 实战：图形探索PM2.5数据

基于前面数据整理实战中的成果，要求：

绘制分年度的PM2.5箱图（所有箱体在一张图上）
分图组绘制PM2.5的直方图
绘制一天24小时PM2.5均值变化的线图
绘制一天24小时PM2.5均值、中位数变化的重叠散点图
各年比较的PM2.5最大值超过100、200、300、500的天数的分段条图
提示：事先需要做较多的数据整理工作

11 数据特征的分析探索

11.1 数值变量的基本描述

df.describe(

percentiles : 需要输出的百分位数，列表格式提供，如[.25, .5, .75]
include = 'None' : 要求纳入分析的变量类型白名单
None (default) : 只纳入数值变量列
A list-like of dtypes : 列表格式提供希望纳入的类型
'all' : 全部纳入
exclude : 要求剔除出分析的变量类型黑名单，选项同上

)

In [ ]:

```
df2.describe(include = 'all')
```

11.2 分类变量的频数统计

Series.value_counts(

normalize = False : 是否返回构成比而不是原始频数
sort = True : 是否按照频数排序(否则按照原始顺序排列)
ascending = False : 是否升序排列
bins : 对数值变量直接进行分段，可看作是pd.cut的简便用法
dropna = True : 结果中是否包括NaN

)

In [ ]:

```
pd.value_counts(df2.类型)
```

In [ ]:

```
pd.value_counts(df2.类型, sort = False)
```