07时间日期

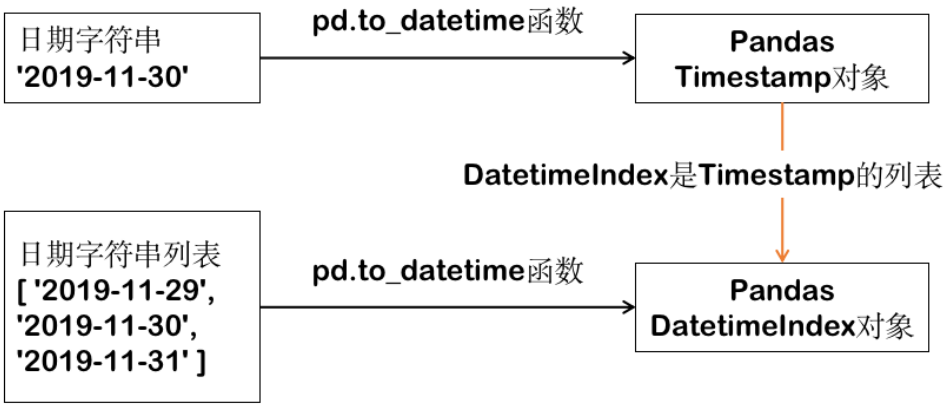
Pandas日期处理的作用：将2018-01-01、1/1/2018等多种日期格式映射成统一的格式对象，在该对象上提供强大的功能支持

几个概念：

1. `pd.to_datetime`：pandas的一个函数，能将字符串、列表、series变成日期形式
2. `Timestamp`：pandas表示日期的对象形式
3. `DatetimeIndex`：pandas表示日期的对象列表形式

其中：

- `DatetimeIndex`是`Timestamp`的列表形式
- `pd.to_datetime`对单个日期字符串处理会得到`Timestamp`
- `pd.to_datetime`对日期字符串列表处理会得到`DatetimeIndex`



一、问题：怎样统计每周、每月、每季度的最高温度？

1、读取天气数据到dataframe

```
fpath = "res/beijing_tianqi_2018.csv"
df = pd.read_csv(fpath)
# 替换掉温度的后缀℃
df.loc[:, "bWendu"] = df["bWendu"].str.replace("℃", "").astype('int32')
df.loc[:, "yWendu"] = df["yWendu"].str.replace("℃", "").astype('int32')
df.head()
```

2、将日期列转换成pandas的日期

```
df.set_index(pd.to_datetime(df["ymd"]), inplace=True)
df.head()
# DatetimeIndex是Timestamp的列表形式
df.index[0]
```

3、 方便的对DatetimeIndex进行查询

```
# 筛选固定的某一天
df.loc['2018-01-05']
# 日期区间
df.loc['2018-01-05':'2018-01-10']
# 按月份前缀筛选
df.loc['2018-03']
# 按月份前缀筛选
df.loc["2018-07":"2018-09"]
# 按年份前缀筛选
df.loc["2018"]
```

4、方便的获取周、月、季度

`Timestamp`、`DatetimeIndex`支持大量的属性可以获取日期分量：
https://pandas.pydata.org/pandas-docs/stable/user_guide/timeseries.html#time-date-components

```
# 周数字列表
df.index.week
# 月数字列表
df.index.month
# 季度数字列表
df.index.quarter
```

5、统计每周、每月、每个季度的最高温度

```
# 统计每周的数据
df.groupby(df.index.week)["bWendu"].max()
# 统计每个月的数据
df.groupby(df.index.month)["bWendu"].max()
# 统计每个季度的数据
df.groupby(df.index.quarter)["bWendu"].max()
```

二、Pandas怎么处理日期索引的缺失？

可以用两种方法实现：
1、`DataFrame.reindex`，调整`dataframe`的索引以适应新的索引
2、`DataFrame.resample`，可以对时间序列重采样，支持补充缺失值

0.构造数据

```
df = pd.DataFrame({
    "pdate": ["2019-12-01", "2019-12-02", "2019-12-04", "2019-12-05"],
    "pv": [100, 200, 400, 500],
    "uv": [10, 20, 40, 50],
})

df  # 缺失了2019-12-03的数据
df.set_index("pdate").plot()
plt.show()
```

方法1：使用`pandas.reindex`方法

```
df_date = df.set_index("pdate")
df_date
df_date.index
# 1.将df的索引设置为日期索引
df_date = df_date.set_index(pd.to_datetime(df_date.index))
df_date
df_date.index
# 2.使用pandas.reindex填充缺失的索引
# 生成完整的日期序列
pdates = pd.date_range(start="2019-12-01", end="2019-12-05")
pdates
# 填充0
df_date_new = df_date.reindex(pdates, fill_value=0)
df_date_new
df_date_new.plot() # 画图
plt.show()
```

方法2：使用`pandas.resample`方法

`resample`的含义：
改变数据的时间频率，比如把天数据变成月份，或者把小时数据变成分钟级别

`resample`的语法：
`(DataFrame or Series).resample(arguments).(aggregate function)`

`resample`的采样规则参数：
https://pandas.pydata.org/pandas-docs/stable/user_guide/timeseries.html#offset-aliases

```python
# 1.先将索引变成日期索引
df_new2 =
df.set_index(pd.to_datetime(df["pdate"])).drop("pdate", axis=1)
df_new2
df_new2.index
# 2.使用resample的方法按照天重采样
# 由于采样会让区间变成一个值，所以需要指定mean等采样值的设定方法
df_new2 = df_new2.resample("D").mean().fillna(0)
df_new2
# resample的使用方式
df_new2.resample("2D").mean()
```