Python数据分析

庄佩玉

2017.12.21

一点自我介绍

wc

为什么是Python

一个公式

$e^{i\pi} + 1 = 0$

```
In [2]: # in Python
        import math
```

```
In [ ]: math.e ** (math.pi * 1j) + 1
```

天文数字？

```
In [7]: # 2 ** 1024
```

简单易用

ipython, jupyter notebook

强大的第三方库

numpy, pandas, scipy, etc...

可重复性与可测试性

准备工作

- 安装 Anaconda (下载 (https://www.anaconda.com/download/))
- 安装下列pip库
  - numpy (mkl)
  - pandas 0.21.1
  - jupyter
  - matplotlib
  - optional: scipy
  - optional: pandas-datareader

Windows: 已编译的包可以在这里下载： https://www.lfd.uci.edu/~gohlke/pythonlibs/#numpy (https://www.lfd.uci.edu/~gohlke/pythonlibs/#numpy)

检查你的环境

In [324]:
```
import pandas as pd
import numpy as np
# use matplotlib for plotting
%matplotlib inline
```

有关ipython和jupyter notebook

- 默认shell的替代
- 代码提示和帮助
- magic command
  - `cd`, `ls`, `pwd`
  - `%timeit`, `%run`
  - `!ver`, `!systeminfo`
  - `who`, `whos` ...

数据处理

- 收集数据
- 观察数据
- 访问数据
- 数据清洗
- 数据分析
- 数据可视化
- 出具报告

收集数据

- 从文件读取数据
- 从库/网络获取数据

In [ ]:

```
# read 'nations.tsv'
# nations = ...

# read stock, 'vmw'
# vmware = ...
```

In [ ]:

```
nations = pd.read_csv('nations.tsv', sep='\t')

from pandas_datareader import data
vmware = data.DataReader('VMW', 'yahoo')
```

观察数据

检查头尾数据

In [ ]:

```
# nations
```

In [ ]:

```
type(nations)
nations.head()
nations.tail()
```

检查索引，类型和列

In [ ]:

```
# nations
```

In [ ]:

```
nations.index
nations.dtypes
nations.columns
```

简单汇总

In [ ]:

```
# nations

# vmware
```

In [ ]:

```
nations.describe(include='all')
nations.info()

vmware.info()
```

作图，持久化

In [ ]:

```
# vmware
```

In [ ]:

```
vmware.plot()

# 持久化
vmware.to_csv('vmware.csv')
```

访问数据

访问列

In [ ]:
```
# nations column/columns
```

In [ ]:
```
nations.year
type(nations.year)
nations['year']
nations[['year', 'gdpPercap']]
```

行与索引

In [ ]:
```
# nations
```

In [ ]:
```
nations.loc[0]
#check type
type(nations.loc[0])
nations.loc[[0, 3, 5]]

nations.set_index('country').iloc[::12]
nations.set_index('country').loc[['China']]

nations.reset_index()
```

单元格

In [ ]:
```
# index: 0, column: country

# index: 0, 3, 5, column: country, pop
```

In [ ]:
```
nations.loc[0, 'country']
nations.loc[[0, 3, 5], ['country', 'pop']]
by_country = nations.set_index('country')
by_country.loc[['China', 'United Kingdom'], ['pop', 'year']]
by_country.iloc[range(10, 15), [2, 3]]
by_country.iloc[range(15, 20), :]
```

修改数据

In [ ]:
```
# make a copy
df = nations.copy()

# drop

# append

# modify
```

In [ ]:

```
df = nations.copy()
df.drop(0)
df.drop('pop', axis=1)
df.append({
    'country': 'My country',
    'continent': 'Unknown',
    'year': 2017,
    'lifeExp': 0.0,
    'pop': np.nan,
    'gdpPercap': np.nan
}, ignore_index=True).tail()
df.tail()

df.iloc[0, 2] = np.nan
df.fillna(12345)
```

数据清洗

格式化

In [350]:

```
vmw_csv = pd.read_csv('vmware.csv')
# format date
```

In [ ]:

```
type(vmw_csv.iloc[0, 0])
formatted_date = pd.to_datetime(vmw_csv['Date'])
vmw_csv['Date'] = formatted_date
type(vmw_csv.iloc[0, 0])
```

列运算

In [ ]:

```
vmw_csv = pd.read_csv('vmware.csv')
# diff on close

# percentage on close

# using lambda / function
```

In [ ]:

```
vmw_csv['Diff'] = vmw_csv['Close'].diff()
vmw_csv['Change'] = vmw_csv['Diff'] / vmw_csv['Close'] * 100
vmw_csv = vmw_csv.assign(AnotherChange = lambda x: x.Diff / x.Close * 100)
```

过滤数据

In [ ]:

```
vmw_csv = pd.read_csv('vmware.csv')
# Close price > 100

# High > 100 and Low < 100

# using lambda
```

In [ ]:

```
vmw_csv[vmw_csv['Close'] > 100]
vmw_csv.apply(lambda x: x.High > 100 and x.Low < 100, axis=1)
vmw_csv[vmw_csv.apply(lambda x: x.High > 100 and x.Low < 100, axis=1)]
```

过滤索引

```
In [355]: country_indexed = nations.set_index('country')
          # name starts with 'C' and ends with 'a'
```

```
In [ ]: capital_c = country_indexed.apply(lambda x: x.name.startswith('C') and x.name.endsw
        ith('a'), axis=1)
        country_indexed[capital_c]
```

数据分析与绘图

排序

```
In [ ]: # nations 对lifeExp排序

        # nations 对lifeExp的平均值按照国家排序

        # 每隔10位作图
```

```
In [ ]: nations.sort_values(by='lifeExp', ascending=True).head(10)

        resampled = nations.groupby(['country'])['lifeExp'].mean().sort_values().iloc[::10]

        resampled.plot(kind='bar', figsize=(16, 5))
```

同列运算

```
In [ ]: temp = vmware.copy()
        # 10 day SMA for 'Close'

        # plot
```

```
In [ ]: temp = vmware.iloc[-100:-1, :].copy()
        temp['CloseSma10'] = vmware['Close'].rolling(10, min_periods=1).mean()
        temp['CloseSma30'] = vmware['Close'].rolling(30, min_periods=1).mean()

        temp[['Close', 'CloseSma10', 'CloseSma30']].plot(figsize=(15, 4))
        # why this?
        # * historical impact
```

插值

```
In [ ]: china = nations[nations['country'] == 'China']
        # insert values for each year
```

In [ ]:

```
china = nations[nations['country'] == 'China']
china.plot(y='gdpPercap', x='year', style='+-')
# insert values for each year
china = china.set_index('year')
years = range(1952, 2007)
populated = china.reindex(years).interpolate('quadratic')
populated['country'].fillna('China', inplace=True)
populated['continent'].fillna(populated['continent'].iloc[0], inplace=True)
populated.plot(y='gdpPercap', style='+-')

# methods: ‘linear’, ‘time’, ‘index’, ‘values’, ‘nearest’, ‘zero’,
# ‘slinear’, ‘quadratic’, ‘cubic’, ‘barycentric’, ‘krogh’, ‘polynomial’,
# ‘spline’, ‘piecewise_polynomial’, ‘from_derivatives’, ‘pchip’, ‘akima’
```

透视表

In [ ]:

```
col = 'lifeExp'
df = nations[['country', col, 'year']]
df = df[df['country'].isin(['China', 'Turkey', 'United States'])]
# draw lines for each country
# df.pivot_table(index=..., columns=..., values=...)
```

In [ ]:

```
# draw lines for each country
df = df.pivot_table(index='year', columns='country', values=col)
df.plot(figsize=[16, 6], style='+-')
```

分组

In [ ]:

```
# nations 按年分组求均值

# nations 按年代（10年）分组求平均

# 按照年份与大洲分组求平均
```

In [ ]:

```
nations.groupby('year').mean()# .plot(y='lifeExp')

nations.groupby(lambda x: nations.iloc[x]['year'] // 10).mean().head(20)

nations.groupby(['year', 'continent']).mean()

grouped = nations.groupby(['year', 'continent']).mean()
# flatten
grouped.reset_index()
```

多图

In [ ]:

```
# 对各大洲的GDP,预期寿命及人口作图不同的子图
```

In [ ]:

```
import matplotlib.pyplot as plt
import time
columns = ['lifeExp', 'pop', 'gdpPercap']
fig, axes = plt.subplots(nrows=len(columns), sharex=True)

for i, col in enumerate(columns):
    grouped = nations[['continent', col, 'year']].groupby(['continent', 'year'])
    if (col == 'pop'):
        cont = grouped.sum()
    else:
        cont = grouped.mean()
    cont = cont.unstack().T.reset_index()
    cont.plot(ax=axes[i], figsize=[14, 16], x='year',style='.-', title='{} by contenant'.format(col), kind='line')
```

Me