Pandas爬取历史天气数据

天气数据网址：<http://www.tianqihoubao.com/>

代码：

import calendar

import datetime

import os

import random

import re

import time

from pprint import pprint

import numpy as np

import pandas as pd

from dateutil.parser import parse

def get\_month\_period(month\_begin=1, month\_end=0):

'''

获得自然月份间隔时间段, 默认取前一个自然月

:param month\_begin: 几个月前的第一天

:param month\_end: 几个月前的结束第一天

:return: e.g（2018,4,1 ，2018，5,1）

'''

now = datetime.datetime.now()

day = datetime.datetime.strptime(datetime.datetime.strftime(now.replace(day=1), "%Y-%m-%d"), "%Y-%m-%d")

def get\_day(shijian, zhouqi):

for i in range(zhouqi):

last\_month\_last\_day = shijian - datetime.timedelta(days=1)

cc = calendar.monthrange(last\_month\_last\_day.year, last\_month\_last\_day.month)

last\_month\_first\_day = shijian - datetime.timedelta(days=cc[1])

shijian = last\_month\_first\_day

i += 1

return (last\_month\_first\_day, last\_month\_last\_day)

begin = get\_day(day, month\_begin)[0]

end = get\_day(day, month\_end + 1)[1] + datetime.timedelta(days=1)

return begin, end

def get\_weather\_data(city='hangzhou', time\_func\_name=get\_month\_period, \*args):

begin, end = time\_func\_name(\*args)

print(begin, end)

# 获得需要爬取的日期区间

date\_list = [date.strftime("%Y%m") for date in pd.date\_range(begin, end, freq='M')]

# 构建url

url\_list = ["http://www.tianqihoubao.com/lishi/{}/month/{}.html".format(city, date) for date in date\_list]

pprint(url\_list)

# 合并后的天气信息文件

filepath = os.path.join(os.path.abspath(os.getcwd()), 'data',

"weather-{}-{}-{}.xlsx".format(city, date\_list[0], date\_list[-1]))

if os.path.exists(filepath):

weather\_data = pd.read\_excel(filepath)

else:

# 抓取天气信息

weather\_data = pd.DataFrame(columns=["日期", "天气状况", "气温", "风力风向"])

for index, url in enumerate(url\_list):

weatherDataFilePath = os.path.join(os.path.abspath(os.getcwd()), 'data',

"weather-{}-{}.xlsx".format(city, date\_list[index]))

# print(weatherDataFilePath)

try:

weather\_df = pd.read\_excel(weatherDataFilePath, header=0)

# 不完整月份的天气数据补充

current\_date = datetime.datetime.strptime(date\_list[index], '%Y%m')

if weather\_df.shape[0] < calendar.monthrange(current\_date.year, current\_date.month)[1]:

weather\_df = pd.DataFrame(pd.read\_html(url, encoding='GBK', header=0)[0])

weather\_df.to\_excel(weatherDataFilePath, index=None)

except Exception:

weather\_df = pd.DataFrame(pd.read\_html(url, encoding='GBK', header=0)[0])

weather\_df.to\_excel(weatherDataFilePath, index=None)

# 随机等待 [1-10]秒 发送请求

time.sleep(random.randint(1, 10))

weather\_data = pd.concat([weather\_data, weather\_df], ignore\_index=True)

weather\_data.to\_excel(filepath, index=None)

return weather\_data, filepath

def clean\_fengli(x):

'''正则表达式清洗风力数据的格式'''

pattern1 = re.compile('(\d+)(\W+)(\d+)') # 1-2, 1~2

pattern2 = re.compile('(\d\*)(\W+)(\d+)') # <2 <=2

pattern3 = re.compile('(\d+)') # 2

if re.match(pattern1, x):

return np.mean((int(re.match(pattern1, x).groups()[0]), int(re.match(pattern1, x).groups()[2])))

elif re.match(pattern2, x):

return int(re.match(pattern2, x).group()[1]) - 0.5

else:

return int(re.match(pattern3, x).group(0))

def clean\_weather\_data(df, filepath, remove=True):

'''使用正则表达式清洗天气数据'''

ptianqi = re.compile('\w+')

pwendu = re.compile('\d+')

pfengli = re.compile('(\w+)\s+(\d\*\W+\d+)')

df['主天气状况'] = df.loc[:, '天气状况'].apply(lambda x: ptianqi.findall(x)[0])

df['次天气状况'] = df.loc[:, '天气状况'].apply(lambda x: ptianqi.findall(x)[1])

df['主风向'] = df.loc[:, '风力风向'].apply(lambda x: pfengli.findall(x)[0][0])

df['主风力'] = df.loc[:, '风力风向'].apply(lambda x: pfengli.findall(x)[0][1])

df['主风力'] = df.loc[:, '主风力'].apply(lambda x: clean\_fengli(x))

df['次风向'] = df.loc[:, '风力风向'].apply(lambda x: pfengli.findall(x)[1][0])

df['次风力'] = df.loc[:, '风力风向'].apply(lambda x: pfengli.findall(x)[1][1])

df['次风力'] = df.loc[:, '次风力'].apply(lambda x: clean\_fengli(x))

df['最高温度'] = df.loc[:, '气温'].apply(lambda x: pwendu.findall(x)[0])

df['最低温度'] = df.loc[:, '气温'].apply(lambda x: pwendu.findall(x)[1])

df["日期"] = df["日期"].apply(lambda x: parse("-".join(re.match('(\d+)\w\*(\d{2})\w\*(\d{2,})', x).groups())))

if remove:

os.remove(filepath)

df.drop(columns=["天气状况", "气温", "风力风向"], inplace=True)

# 存储所有清洗好的天气数据

df.to\_excel(filepath.replace('weather-', 'weatherCleaned-'), index=False)

return df # [日期 主天气状况 次天气状况 主风向 主风力 次风向 次风力 最高温度 最低温度]

if \_\_name\_\_ == '\_\_main\_\_':

weather\_data, filepath = get\_weather\_data('hangzhou', get\_month\_period, 3)

clean\_weather\_data(weather\_data, filepath, remove=True)