20250418 爬虫

1.urllib_基本使用.py

```
# 使用urllib来获取百度的源码
import urllib.request

# (1) 定义一个Url
url = 'http://www.baidu.com'

# (2)模拟浏览器向服务器发送请求，
response = urllib.request.urlopen(url)

# （3）获取响应中的页面源码，
# content = response.read().decode('utf-8')
#
# print(content)

# read() 字节形式读取二进制 扩展：rede(5)返回前几个字节

# readline() 读取一行
# content = response.readline()
# print(content)  # b'<!DOCTYPE html>\n'


# readlines() 一行一行读取 直至结束
# content = response.readlines()
# print(content)  # 读取所有的数据

# getcode() 获取状态码
# content = response.getcode()
# print(content)  # 200

# geturl() 获取url
# content = response.geturl()
# print(content)  # http://www.baidu.com

# getheaders() 获取header
print(response.getheaders())
```

2.urllib_下载.py

```
import urllib.request

# 1.下载网页
# url_page = "http://www.baidu.com"
# 在python中，url代表的是下载地址，filename文件的名字
# urllib.request.urlretrieve(url_page, './temp/1.baidu.html')

# 2.下载图片
# url_img = 'https://ww4.sinaimg.cn/mw690/008BY4DGly1iOgg1ukut7j31p62m7b29.jpg'
# urllib.request.urlretrieve(url_img, "./temp/2chen.jpg")
```

```
# 3.下载视频
url_video = 'http://t.cn/A6rFJejQ'
urllib.request.urlretrieve(url_video, './temp/3chen.mp4')
```

3.urllib_请求对象定制.py

```
import urllib.request

url = "http://www.baidu.com/s?wd=%E9%99%88%E4%B8%BD%E5%90%9B"

headers = {
    'user-agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36'
}


request = urllib.request.Request(url=url, headers=headers)

response = urllib.request.urlopen(request)

content = response.read().decode('utf-8')

print(content)
```

扩充1：get和post,delete,put ----restful

扩充2：json格式

4.urllib_百度翻译.py

```
import urllib.request
import urllib.parse
import json

url = 'https://fanyi.baidu.com/sug'
headers = {
    'user-agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36'
}
data = {
    'kw': 'spider'
}
# post 请求的参数，必须要进行编码
data = urllib.parse.urlencode(data).encode('utf-8')
request = urllib.request.Request(url=url,data=data,headers=headers)
response = urllib.request.urlopen(request)
content = response.read().decode('utf-8')
obj = json.loads(content)
print(type(obj))
```

5.urllib_豆瓣电影第一页.py

```
# https://movie.douban.com/j/chart/top_list?
type=5&interval_id=100%3A90&action=&start=0&limit=20

import urllib.request
import urllib.parse
import json

url = 'https://movie.douban.com/j/chart/top_list?
type=5&interval_id=100%3A90&action=&start=0&limit=20'

headers = {
    'user-agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36'
}

# (1)请求对象定制
request = urllib.request.Request(url=url, headers=headers)

# （2）获取相应的数据
response = urllib.request.urlopen(request)
content = response.read().decode('utf-8')

# (3)数据保存到本地
fp = open('./temp/douban1.json', 'w', encoding='utf-8')
fp.write(content)
```

6.urllib_豆瓣电影十页.py

```
# 观察路由
"""
    第一页：https://movie.douban.com/j/chart/top_list?
type=5&interval_id=100%3A90&action=&start=0&limit=20
    第二页：https://movie.douban.com/j/chart/top_list?
type=5&interval_id=100%3A90&action=&start=20&limit=20
    第三页：https://movie.douban.com/j/chart/top_list?
type=5&interval_id=100%3A90&action=&start=40&limit=20
    第四页：https://movie.douban.com/j/chart/top_list?
type=5&interval_id=100%3A90&action=&start=60&limit=20
    (page-1)*20
"""

"""
    需求：下载豆瓣电影前十页的数据
    （1）请求对象定制
    （2）获取相应数据
    （3）下载数据
"""

import urllib.request
import urllib.parse
import json

def create_request(page):
    base_url = 'https://movie.douban.com/j/chart/top_list?
type=5&interval_id=100%3A90&action=&'
    data = {
        'start':(page - 1) * 20,
```

```
        'limit':20
    }
    data = urllib.parse.urlencode(data)
    url = base_url + data

    headers = {
        'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36'
    }
    return urllib.request.Request(url=url, headers=headers)

def get_content(request):
    response = urllib.request.urlopen(request)
    return response.read().decode('utf-8')

def down_load(page, content):
    with open('./temp/douban_' + str(page) + '.json', 'w', encoding='utf-8') as
fp:
        fp.write(content)


if __name__ == '__main__':
    start_page = int(input("请输入起始页："))
    end_page = int(input("请输入结束页:"))

    for page in range(start_page, end_page + 1):
        # 每一页都有自己的请求对象
        request = create_request(page)

        # 获取相应数据
        content = get_content(request)

        # 下载
        down_load(page, content)
```

7.urllib_kfc下载.py

```
import urllib.request
import urllib.parse

def create_request(page):
    base_url = 'https://www.kfc.com.cn/kfccda/ashx/GetStoreList.ashx?op=cname'

    data = {
        'cname': '成都',
        'pid': '',
        'pageIndex': page,
        'pageSize': '10'
    }

    data = urllib.parse.urlencode(data).encode('utf-8')

    headers = {
        'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36'
    }
```

```
    return urllib.request.Request(url=base_url, data=data, headers=headers)


def get_content(request):
    reponse = urllib.request.urlopen(request)
    return reponse.read().decode('utf-8')

def down_load(page, content):
    with open('./temp/kfc_' + str(page) + '.json', 'w', encoding='utf-8') as fp:
        fp.write(content)

if __name__ == '__main__':
    start_page = int(input("请输入起始页："))
    end_page = int(input("请输入结束页："))
    for page in range(start_page, end_page + 1):
        # 请求对象定制
        request = create_request(page)
        # 获取网页源码
        content = get_content(request)
        # 下载数据
        down_load(page, content)
```

8.urllib_异常.py

```
import urllib.request
import urllib.parse
import urllib.error

# url = 'https://blog.csdn.net/qq_41684621/article/details/113851644lll'
url = 'http://www.zhangsandewangzhi.com'

headers = {
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36'
}

try:
    request = urllib.request.Request(url=url, headers=headers)
    response = urllib.request.urlopen(request)
    content = response.read().decode('utf-8')
    print(content)
except urllib.error.HTTPError:
    print('系统在升级.....')
except urllib.error.URLError:
    print('在升级啊.....')
```

9.urllib_微博.py

```
# 适用场景：数据采集，需要绕过登录，然后进入到某个页面
# 个人信息界面是utf-8,字符集gb2312

import urllib.request
import urllib.parse
```

```
# url = 'https://weibo.com/ajax/profile/info?uid=2164895442'
url = 'https://weibo.com/u/2164895442'

# headers = {
#     'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36'
# }

headers = {
    # ':authority': 'weibo.cn',
    # ':method': 'GET',
    # ':path': '/6451491586/info',
    # ':scheme': 'https',
    'accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,ima
ge/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9',
    # 'accept-encoding': 'gzip, deflate, br',
    'accept-language': 'zh-CN,zh;q=0.9',
    'cache-control': 'max-age=0',
    #     cookie中携带着你的登陆信息    如果有登陆之后的cookie  那么我们就可以携带着cookie
进入到任何页面
    'cookie': '_T_WM=24c44910ba98d188fced94ba0da5960e;
SUBP=0033WrSXqPxfM725Ws9jqgMF55529P9D9WFxxfgNNUmXi4YiaYZKr_J_5NHD95QcSh-
pSh.pSKncWs4DqcjiqgSXIgvVPcpD; SUB=_2A25MKKG_DeRhGeBK7lMV-
S_JwzqIHXVv0s_3rDV6PUJbktCOLXL2kW1NR6e0UHkCGcyvxTYyKB2OV9aloJJ7mUNz;
SSOLoginState=1630327279',
    # referer  判断当前路径是不是由上一个路径进来的     一般情况下 是做图片防盗链
    'referer': 'https://weibo.cn/',
    'sec-ch-ua': '"Chromium";v="92", " Not A;Brand";v="99", "Google
Chrome";v="92"',
    'sec-ch-ua-mobile': '?0',
    'sec-fetch-dest': 'document',
    'sec-fetch-mode': 'navigate',
    'sec-fetch-site': 'same-origin',
    'sec-fetch-user': '?1',
    'upgrade-insecure-requests': '1',
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/92.0.4515.159 Safari/537.36',
}

request = urllib.request.Request(url=url,headers=headers)
response = urllib.request.urlopen(request)
content = response.read().decode('gb2312')

# 将数据保存到本地
with open('./temp/weibo.html', 'w', encoding='gb2312') as fp:
    fp.write(content)
```

20250419 解析

10.解析_xpath基本用法.py

```
from lxml import etree

# xpath 解析
```

```
# （1）本地文件      etree.parse
# （2）服务器响应的数据，response.read().decode('utf-8')  重点    etree.HTML

# xpath解析本地文件
tree = etree.parse("10.解析_xpath的基本使用.html")

# 查找ul下面的li
# li_list = tree.xpath("//body/ul/li")  # 4

# 查找所有有id的属性的li标签
# li_list = tree.xpath('//ul/li[@id]')  # 4

# text()获取标签的内容
# li_list = tree.xpath("//ul/li[@id]/text()")  # ['北京', '上海', '深圳', '武汉']

# 找到id=l1的li标签,注意，引号问题
# li_list = tree.xpath("//ul/li[@id='l1']/text()")  # ['北京']

# 查询id中包含l的标签
# li_list = tree.xpath('//ul/li[contains(@id,"l")]/text()')  # ['北京', '上海']

# 查询id的值以l开头
# li_list = tree.xpath('//ul/li[starts-with(@id,"l")]/text()')  # ['北京', '上海']

# 查找id=l1同时class=c1
# li_list = tree.xpath("//ul/li[@id='l1' and @class='c1']/text()")  # ['北京']

# 查询id=l1和id=l2
li_list = tree.xpath("//ul/li[@id='l1']/text() | //ul/li[@id='l2']/text()")  #
['北京', '上海']

print(li_list)
print(len(li_list))
```

11.解析_获取百度一下.py

```
# 1.获取网页内容
# 2.解析，解析服务器响应的文件，etree.HTML
# 3.打印

import urllib.request
from lxml import etree

url = 'http://www.baidu.com'

headers = {
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36'
}

request = urllib.request.Request(url=url, headers=headers)
response = urllib.request.urlopen(request)
content = response.read().decode('utf-8')

# print(content)
tree = etree.HTML(content)
```

```
# 获取想要的数据
result = tree.xpath("//input[@id='su']/@value")[0]

print(result)  # 百度一下
```

12.解析_站长素材.py

```
"""
    第一页：https://sc.chinaz.com/tupian/qinglvtupian.html
    第二页：https://sc.chinaz.com/tupian/qinglvtupian_2.html
    第三页：https://sc.chinaz.com/tupian/qinglvtupian_3.html
    第四页：https://sc.chinaz.com/tupian/qinglvtupian_4.html
"""

import urllib.request
from lxml import etree

def create_request(page):
    if page == 1:
        url = 'https://sc.chinaz.com/tupian/qinglvtupian.html'
    else:
        url = 'https://sc.chinaz.com/tupian/qinglvtupian_' + str(page) + '.html'

    headers = {
        'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36'
    }

    return urllib.request.Request(url=url, headers=headers)


def get_content(request):
    response = urllib.request.urlopen(request)
    return response.read().decode('utf-8')

def down_load(content):
    # 下载图片，urllib.request.urlretrieve(图片地址，图片名字)
    tree = etree.HTML(content)
    # 获取文字信息
    name_list = tree.xpath("//div[@class='container']/div/div/img/@alt")
    # 获取图片
    src_list = tree.xpath("//div[@class='container']/div/div/img/@data-
original")

    # 下载图片并保存
    for i in range(len(name_list)):
        name = name_list[i]
        src = src_list[i]
        url = "https:" + src

        # 保存数据
        urllib.request.urlretrieve(url=url, filename="./temp/" + name + ".jpg")


if __name__ == '__main__':
    start_page = int(input("请输入起始页："))
    end_page = int(input("请输入结束页："))
```

```
    for page in range(start_page, end_page + 1):
        # 1.请求对象定制
        request = create_request(page)
        # 2.获取网页源码
        content = get_content(request)
        # 3.下载
        down_load(content)
```

13.解析_jsonpath基本用法.py

```
import json
import jsonpath

obj = json.load(open('./tool/store.json', 'r', encoding='utf-8'))

# 书店的所有作者
# print(jsonpath.jsonpath(obj,"$.store.book[*].author"))  # ['六道', '天蚕土豆',
'唐家三少', '南派三叔']

# 所有作者
# print(jsonpath.jsonpath(obj,'$..author'))  # ['六道', '天蚕土豆', '唐家三少', '南派
三叔']

# store下面的所有元素
# print(jsonpath.jsonpath(obj,'$.store.*'))  # 所有内容

# store里面所有东西的价格
# print(jsonpath.jsonpath(obj,'$.store..price'))  # [8.95, 12.99, 8.99, 22.99,
19.95]

# 第三本书
# print(jsonpath.jsonpath(obj,'$..book[2]'))  # [{'category': '修真', 'author':
'唐家三少', 'title': '斗罗大陆', 'isbn': '0-553-21311-3', 'price': 8.99}]

# 最后一本书
# print(jsonpath.jsonpath(obj,'$..book[(@.length - 1)]'))

# 前面两本书
# print(jsonpath.jsonpath(obj, '$..book[:2]'))

# 条件过滤需要在（）的前面添加一个
# 过滤出所有包含isbn的书
# print(jsonpath.jsonpath(obj,'$..book[?(@.isbn)]'))

# 那本书超过十块钱
# print(jsonpath.jsonpath(obj,'$..book[?(@.price>10)]'))
```

14.解析_jsonpath淘票票.py

```
import urllib.request
import json
import jsonpath
```

```
url = 'https://dianying.taobao.com/cityAction.json?
activityId&_ksTS=1745045424373_108&jsoncallback=jsonp109&action=cityAction&n_s=n
ew&event_submit_doGetAllRegion=true'

headers = {
    # ':authority': 'dianying.taobao.com',
    # ':method': 'GET',
    # ':path': '/cityAction.json?
activityId&_ksTS=1629789477003_137&jsoncallback=jsonp138&action=cityAction&n_s=n
ew&event_submit_doGetAllRegion=true',
    # ':scheme': 'https',
    'accept': 'text/javascript, application/javascript, application/ecmascript,
application/x-ecmascript, */*; q=0.01',
    # 'accept-encoding': 'gzip, deflate, br',
    'accept-language': 'zh-CN,zh;q=0.9',
    'cookie': 'cna=UkO6F8VULRwCAXTqq7dbS5A8; miid=949542021157939863;
sgcookie=E100F01JK9XMmyoZRigjfmZKExNdRHQqPf4v9NIWIC1nnpnxyNgROLshAf0gz7lGnkKvwCn
u1umyfirMSAWtubqc4g%3D%3D; tracknick=action_li; _cc_=UIHiLt3xSw%3D%3D;
enc=dA18hg7jG1xapfVGPHoQCAkPQ4as1%2FEUqsG4M6AcAjHFFUM54HWpBv4AAm0MbQgqO%2BiZ5qkU
eLIxljrHkOW%2BtQ%3D%3D; hng=CN%7Czh-CN%7CCNY%7C156; thw=cn;
_m_h5_tk=3ca69de1b9ad7dce614840fcd015dcdb_1629776735568;
_m_h5_tk_enc=ab56df54999d1d2cac2f82753ae29f82;
t=874e6ce33295bf6b95cfcfaff0af0db6; xlly_s=1;
cookie2=13acd8f4dafac4f7bd2177d6710d60fe; v=0; _tb_token_=e65ebbe536158;
tfstk=cGhRB7mNpnxkDmUx7YpDAMNM2gTGZbWLxUZN9U4ulewe025didli6j5AFPI8MEC..;
l=eBrgmF1cOsMXqSxaBO5aFurza77tzIRb8sPzaNbMiInca6OdtFt_rNCK2Ns9SdtjgtfFBetPVKlOcR
CEF3apbgiMW_N-1NKDSxJ6-;
isg=BBoas2yXLzHdGp3pCh7XVmpja8A8S54lyLj1RySTHq14l7vRDNufNAjpZ2MLRxa9',
    'referer': 'https://dianying.taobao.com/',
    'sec-ch-ua': '"Chromium";v="92", " Not A;Brand";v="99", "Google
Chrome";v="92"',
    'sec-ch-ua-mobile': '?0',
    'sec-fetch-dest': 'empty',
    'sec-fetch-mode': 'cors',
    'sec-fetch-site': 'same-origin',
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/92.0.4515.159 Safari/537.36',
    'x-requested-with': 'XMLHttpRequest',
}

request = urllib.request.Request(url=url, headers=headers)
response = urllib.request.urlopen(request)
content = response.read().decode('utf-8')

# 由于不是标准的json,所以切割
content = content.split('(')[1].split(')')[0]
# print(content)

# 保存数据
with open('14.解析_jsonpath淘票票.json', 'w', encoding='utf-8') as fp:
    fp.write(content)

# 读取
obj = json.load(open('14.解析_jsonpath淘票票.json', 'r', encoding='utf-8'))

city_list = jsonpath.jsonpath(obj, '$..regionName')
print(city_list)
```

15.解析_bs基本用法.py

```
from bs4 import BeautifulSoup

# 通过解析本地文件，来将bs4的基本语法进行讲解

soup = BeautifulSoup(open('15.解析_bs4的基本使用.html', encoding='utf-8'), 'lxml')

# 根据标签名字找结点,找到符合条件的第一个元素
# print(soup.a)
# 获取标签的属性和属性值
# print(soup.a.attrs)

# bs4的一些函数
#（1）find()
# 返回的是第一个符合条件的数据
# print(soup.find('a'))

# 根据title的值来找到对应的标签对象
# print(soup.find('a', title='a2'))

# 根据class的值找到对应的标签对象，注意的是class要添加下划线，因为class是一个关键字
# print(soup.find('a', class_='a1'))

# （2）find_all
# 查找所有满足条件
# print(soup.find_all('a'))

# 如果要获取a和span
# print(soup.find_all(['a', 'span']))

# limit的作用是查找前几个数据
# print(soup.find_all('li', limit=2))

# (3)select【推荐】全部东西都可以使用选择器来获取
# print(soup.select('a'))  # 标签选择器

# print(soup.select('.a1'))  #类选择器

# print(soup.select('#l1'))  # id选择器

# print(soup.select('li[id]'))  # 属性选择器

# print(soup.select('li[id="l2"]'))  # 属性选择器

# print(soup.select('div li'))  # 后代选择器

# print(soup.select('div>ul>li'))  # 子代选择器

# print(soup.select('a,li'))  # 并集选择器

# 节点获取
# obj = soup.select('#d1')[0]
# print(obj.get_text())  # 获取节点的值

obj = soup.select('#p1')[0]
print(obj.name)  # p
```

```
print(obj.attrs)  # {'id': 'p1', 'class': ['p1']}
# 以下结果都是一样，都是获得元素的class属性，p1
print(obj.attrs.get('class'))
print(obj.get('class'))
print(obj['class'])
```

16.requests_基本使用.py

```
import requests

url = 'http://www.baidu.com'

response = requests.get(url=url)

print(response.text)
```

17.案例_古诗文网.py

```
# 通过登录，直接进入主页
# 通过找登录接口我们发现，参数比较多

import requests

url = 'https://www.gushiwen.cn/user/login.aspx?
from=http://www.gushiwen.cn/user/collect.aspx'

headers = {
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36'
}

# 获取页面的源码
response = requests.get(url=url,headers=headers)
content = response.text

# 解析页面源码，获取__VIEWSTATE和__VIEWSTATEGENERATOR
from bs4 import BeautifulSoup
soup = BeautifulSoup(content, 'lxml')

# 获取不知道的两个隐藏的文字内容
viewstate = soup.select("#__VIEWSTATE")[0].attrs.get('value')
viewstategenerator = soup.select("#__VIEWSTATEGENERATOR")[0].attrs.get('value')

# 获取验证码图片
code = soup.select("#imgCode")[0].attrs.get('src')
code_url = 'https://www.gushiwen.cn/' + code

# 下面有错，但是按照思路正常写
# import urllib.request
# urllib.request.urlretrieve(url=code_url, filename='./temp/code.jpg')
# requests里面有一个方法session(),通过session的返回值，就能使用请求变成一个对象
session = requests.session()
# 验证码的url内容
response_code = session.get(code_url)
```

```
# 注意此时要使用二进制数据，因为我们要使用图片下载
content_code = response_code.content

# 将二进制写入文件
with open('./temp/code.jpg','wb') as fp:
    fp.write(content_code)


code_name = input("请输入你的验证码：")

# 点击登录
url_post = 'https://www.gushiwen.cn/user/login.aspx?
from=http%3a%2f%2fwww.gushiwen.cn%2fuser%2fcollect.aspx'

data_post = {
    "__VIEWSTATE": viewstate,
    "__VIEWSTATEGENERATOR": viewstategenerator,
    "from": 'http://www.gushiwen.cn/user/collect.aspx',
    "email": '18280111255',
    "pwd": 'luo510626',
    "code": code_name,
    "denglu": '登录'
}

response_post = session.post(url=url,headers=headers,data=data_post)
content = response_post.text
with open('./temp/gushiwen.html', 'w', encoding='utf-8') as fp:
    fp.write(content)
```

20250419 数据分析

一、综述

\1. 为什么要学习数据分析

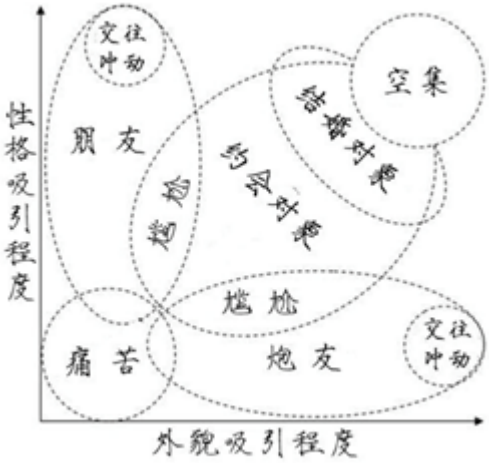
\2. 什么是数据分析

\3. 环境安装

\4. 认识jupyter notebook

1.1 为什么要学习数据分析

Departmer	Name	GroupName	ModifiedDa
1	Engineering	Research and Development	7/31/2004
2	Tool Design	Research and Development	7/31/2004
3	Sales	Sales and Marketing	7/31/2004
4	Marketing	Sales and Marketing	7/31/2004
5	Purchasing	Inventory Management	7/31/2004
6	Research and Development	Research and Development	7/31/2004
7	Production	Manufacturing	7/31/2004
8	Production Control	Manufacturing	7/31/2004
9	Human Resources	Executive General and Administration	8/30/2006
10	Finance	Executive General and Administration	7/31/2004
11	Information Services	Executive General and Administration	9/28/2007
12	Document Control	Quality Assurance	7/31/2004
13	Quality Assurance	Quality Assurance	7/31/2004
14	Facilities and Maintenance	Executive General and Administration	1/10/2005
16	Shipping and Receiving	Inventory Management	7/31/2004



\1. 有岗位需求

\2. 是python数据科学得基础

\3. 是机器学习课程的基础

1.2 什么是数据分析

数据分析是用适当的方法对收集来的大量数据进行分析，帮助人们作出判断，以便采取适当行动。

1.2.1 数据分析的流程

1.3 认识jupyter notebook

jupyter notebook:一款编程/文档/笔记/展示软件

启动命令：jupyter notebook

二、matplotlib折线图

\1. 什么是matplotlib

\2. matplotlib基本要点

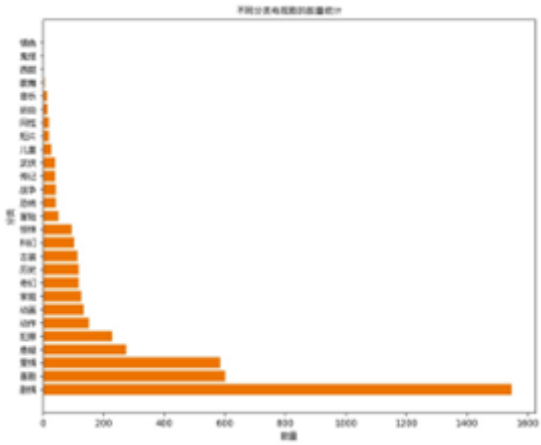
\3. matplotlib的散点图、直方图、柱状图

\4. 更多的画图工具

2.1 为什么要学习matplotlib

\1. 能将数据进行可视化，更直观的呈现

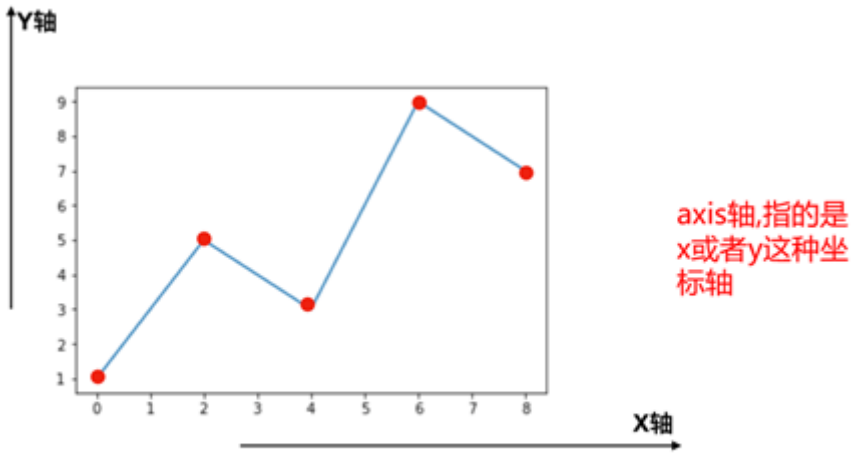
\2. 使数据更加客观、更具说服力



2.2 什么是matplotlib

matplotlib:最流行的Python底层绘图库,主要做数据可视化表格,名字取材于MATLAB,模仿MATLAB构建

2.3 matplotlib基本要点



那么上面的每一个红色的点是什么呢?

每个红色的点是坐标，把5个点的坐标连接成一条线，组成了一个折线图。

那么到底如何把它通过代码画出来呢?

通过下面的小例子我们来看一下matplotlib该如何简单的使用

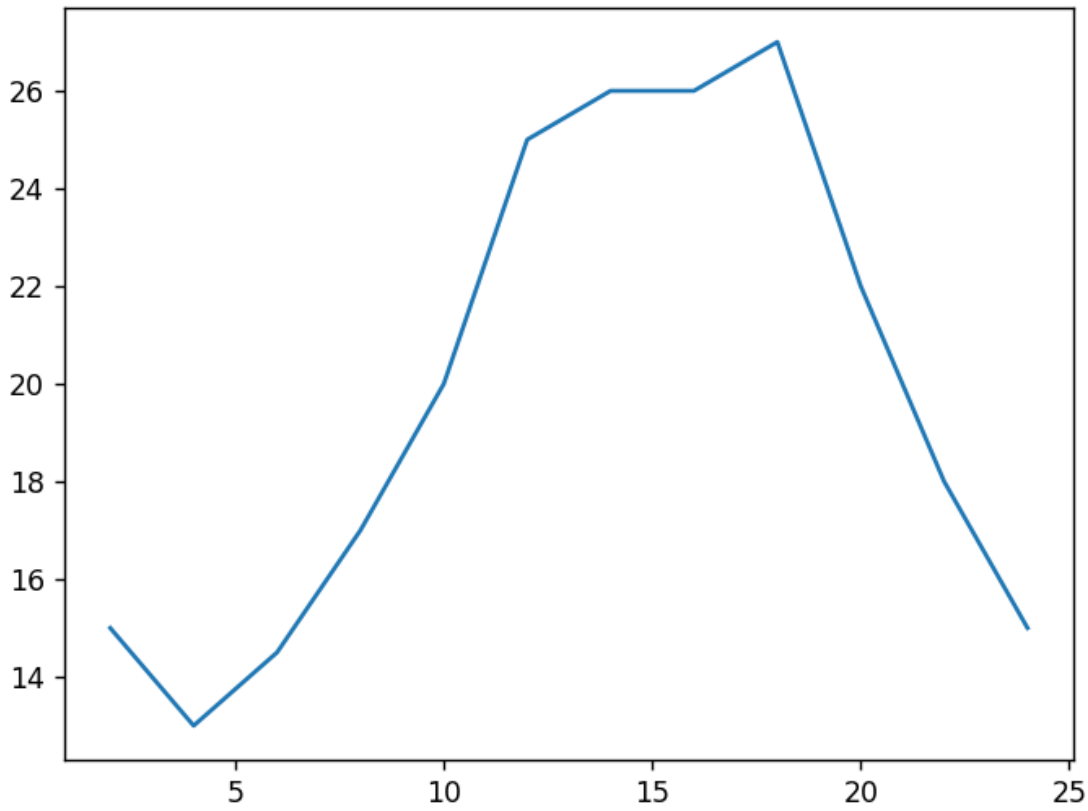
假设一天中每隔两个小时(range(2,26,2))的气温（℃）分别是

[15,13,14.5,17,20,25,26,26,27,22,18,15]

代码：（18.mat_气温变化.py）

```
from matplotlib import pyplot as plt

x = range(2, 26, 2)
y = [15,13,14.5,17,20,25,26,26,27,22,18,15]

# 绘制
plt.plot(x, y)
plt.show()
```

效果图：



但是目前存在以下几个问题：

\1. 设置图片大小（想要一个高清无码大图）

\2. 保存到本地

\3. 描述信息，比如x轴和y轴标识什么，这个图表示什么

\4. 调整x或y的刻度的间距

\5. 线条的样式（比如颜色，透明度等）

\6. 标记出特殊的点（比如告诉别人最高点和最低点在那里）

\7. 给图片添加一个水印（防伪，防止盗用）

2.4 设置图片大小

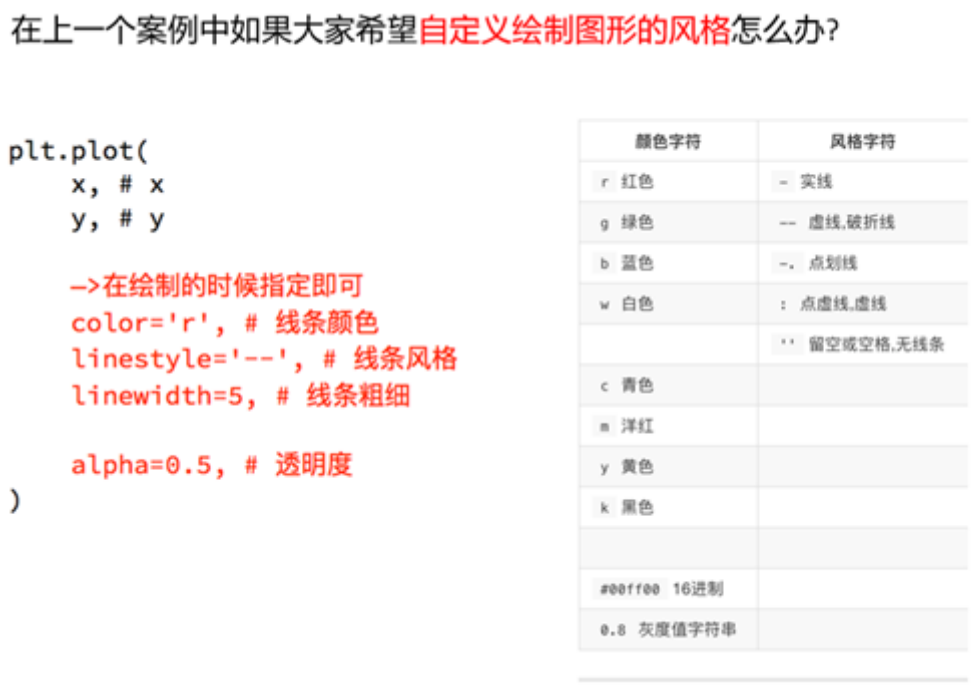
```
from matplotlib import pyplot as plt

x = range(2, 26, 2)
y = [15,13,14.5,17,20,25,26,26,27,22,18,15]

# 设置图片大小
plt.figure(figsize=(20, 8), dpi=80)
'''
    figure图形图标的意思，在这里指的就是我们画的图
    通过实例化一个figure并且传递参数
'''

# 绘制
plt.plot(x, y)

# x轴的刻度
# plt.xticks(range(2, 26))
# 设置一个小数
# _xtick_labels = [i/2 for i in range(4, 49)]  # 每隔0.5
# plt.xticks(_xtick_labels[::3])  # 切片
plt.xticks(range(25, 50))

plt.yticks(range(min(y), max(y) + 1))  # y轴



# 保存图片
plt.savefig("./temp/temp1.png")


plt.show()
```

案例：10点-12点的每一分钟的气温变化，温度采用20-35的随机数

```
from matplotlib import pyplot as plt
import random
from matplotlib import font_manager

my_font = font_manager.FontProperties(fname='./tool/simhei.ttf')

x = range(0, 120)
y = [random.randint(20, 35) for i in range(120)]

plt.figure(figsize=(20, 8), dpi=80)
plt.plot(x, y)

# 设置x轴的刻度
_x = list(x)
_xtick_labels = ["10点{}分".format(i) for i in range(60)]
_xtick_labels += ["11点{}分".format(i) for i in range(60, 120)]

plt.xticks(_x[::3], _xtick_labels[::3], rotation=45, fontproperties=my_font)

# 展示描述信息
```

```
plt.xlabel("时间", fontproperties=my_font)
plt.ylabel('温度', fontproperties=my_font)
plt.title("10-12点每分钟的温度变化", fontproperties=my_font)

plt.show()
```

案例：20.动手折线图.py



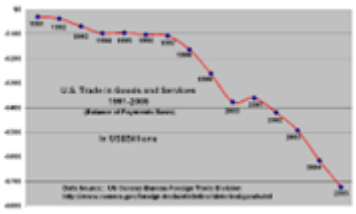

```
from matplotlib import pyplot as plt
from matplotlib import font_manager

my_font = font_manager.FontProperties(fname="./tool/simhei.ttf")

a = [1, 0, 1, 1, 2, 4, 3, 2, 3, 4, 4, 5, 6, 5, 4, 3, 3, 1, 1, 1]
b = [1, 0, 3, 1, 2, 2, 3, 3, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1]

x = range(11, 31)

plt.figure(figsize=(20, 8), dpi=80)

plt.plot(x, a, label="自己", color="red", linestyle='-.')
plt.plot(x, b, label="同桌", color="green", linestyle='--')

# 设置x轴刻度
_xtick_labels = ["{}岁".format(i) for i in x]
plt.xticks(x, _xtick_labels,fontproperties=my_font)
plt.yticks(range(0, 9))

# 添加网格
plt.grid(alpha=0.4)

# 添加图例
plt.legend(prop=my_font, loc="upper left")

plt.show()
```

2.5 matplotlib只能发折线图吗?

matplotlib能够绘制折线图,散点图,柱状图,直方图,箱线图,饼图等

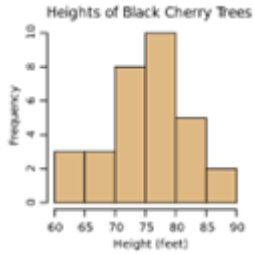
但是,我们需要知道不同的***统计图***到底能够表示出什么,以此来决定选择哪种***统计图***来更直观的呈现我们的数据

2.6 对比常用统计图



折线图:以折线的上升或下降来表示统计数量的增减变化的统计图

特点:能够显示数据的变化趋势，反映事物的变化情况。(变化)

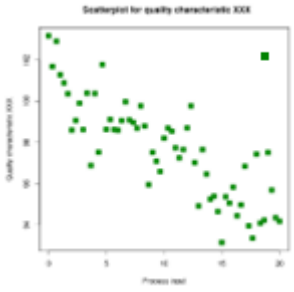


直方图:由一系列高度不等的纵向条纹或线段表示数据分布的情况。 一般用横轴表示数据范围，纵轴表示分布情况。

特点:绘制连***续性***的数据,展示一组或者多组数据的分布状况(统计)

条形图:排列在工作表的列或行中的数据可以绘制到条形图中。

特点:绘制连***离散***的数据,能够一眼看出各个数据的大小,比较数据之间的差别。(统计)



散点图:用两组数据构成多个坐标点，考察坐标点的分布,判断两变量之间是否存在某种关联或总结坐标点的分布模式。

特点:判断变量之间是否存在数量关联趋势,展示离群点(分布规律)

三、matplotlib常用统计图

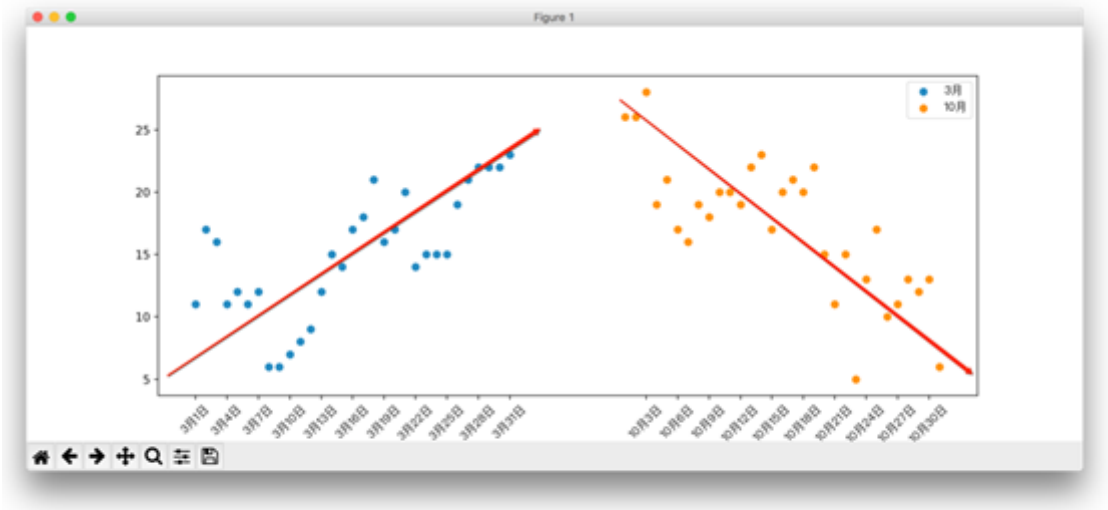
3.1 绘制散点图

假设通过爬虫你获得了北京2016年3，10月份每天白天的最高气温（分别位于列表a,b），那么此时如何寻找出气温和随时间（天）变化的某种规律?

a = [11,17,16,11,12,11,12,6,6,7,8,9,12,15,14,17,18,21,16,17,20,14,15,15,15,19,21,22,22,22,23]

b = [26,26,28,19,21,17,16,19,18,20,20,19,22,23,17,20,21,20,22,15,11,15,5,13,17,10,11,13,12,13,6]

数据来源：http://lishi.tianqi.com/beijing/index.html



技术要点:plt.scatter(x,y)

```
from matplotlib import pyplot as plt
from matplotlib import font_manager

my_font = font_manager.FontProperties(fname="./tool/simhei.ttf")

y_3 = [11, 17, 16, 11, 12, 11, 12, 6, 6, 7, 8, 9, 12, 15, 14, 17, 18, 21, 16, 17,
20, 14, 15, 15, 15, 19, 21, 22, 22,
       22, 23]

y_10 = [26, 26, 28, 19, 21, 17, 16, 19, 18, 20, 20, 19, 22, 23, 17, 20, 21, 20,
22, 15, 11, 15, 5, 13, 17, 10, 11, 13,
        12, 13, 6]

# x轴
x_3 = range(1, 32)
x_10 = range(51, 82)

# 设置图像大小
plt.figure(figsize=(20, 8), dpi=80)

# 使用scatter方法绘制散点图
plt.scatter(x_3, y_3, label="3月")
plt.scatter(x_10, y_10, label="10月")

# 设置x轴
_x = list(x_3) + list(x_10)
_xtick_labels = ['3月{}日'.format(i) for i in x_3]
```

```
_xtick_labels += ['10月{}日'.format(i-50) for i in x_10]
plt.xticks(_x[::3], _xtick_labels[::3],fontproperties=my_font,rotation=45)

# 绘制网格
plt.grid(alpha=0.4)

# 添加图例
plt.legend(prop=my_font)

# 展示
plt.show()
```

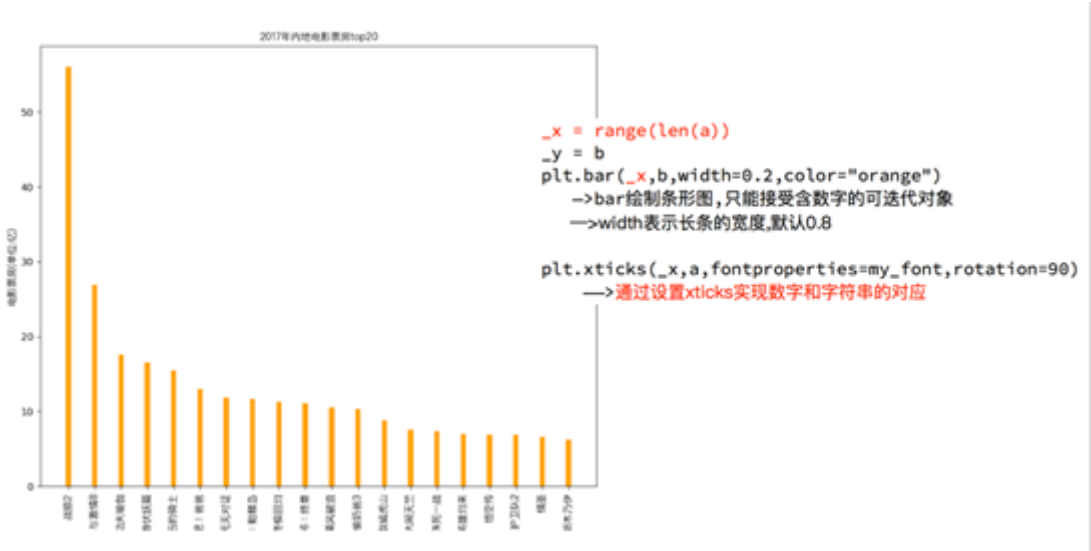
3.2 绘制条形图

假设你获取到了2017年内地电影票房前20的电影（列表a）和电影票房数据（列表b）,那么如何更加直观的展示该数据?

a = ["战狼2","速度与激情8","功夫瑜伽","西游伏妖篇","变形金刚5：最后的骑士","摔跤吧！爸爸","加勒比海盗5：死无对证","金刚：骷髅岛","极限特工：终极回归","生化危机6：终章","乘风破浪","神偷奶爸3","智取威虎山","大闹天竺","金刚狼3：殊死一战","蜘蛛侠：英雄归来","悟空传","银河护卫队2","情圣","新木乃伊",]

b=
[56.01,26.94,17.53,16.49,15.45,12.96,11.8,11.61,11.28,11.12,10.49,10.3,8.75,7.55,7.32,6.99,6.88,6.86,6.58,6.23] 单位:亿

数据来源: http://58921.com/alltime/2017



3.3 绘制直方图