文本(数据)分析步骤

1. 数据采集
2. 数据清洗整理
3. 数据分析

一般2和3常常混在一起

数据采集 ¶

数据采集(网络爬虫)是一种高效获取数据的新方法，但是必须满足一定的前提条件：

1. 采集者有权限，能访问的
2. 数据是能够网络上看到的
3. 知道这些数据对应的网址 不满足上面三条，凭空是无法获取我们想要的数据。在本文研究中，矿泉水相关评论可以在京东上搜到产品，我们能访问能看到，经过一定的网络访问分析，第三条评论数据的网址也能拿到。

下面列出的是各矿泉水产品链接及对应的评论网址模板

农夫山泉

https://item.jd.com/12211948808.html

https://sclub.jd.com/comment/productPageComments.action?callback=fetchJSON_comment98vv10805&productId=12211948808&score=0&sortType=5&page={page}&pageSize=10&isShadowSku=0&rid=0&fold=1

百岁山

https://item.jd.com/952862.html

https://sclub.jd.com/comment/productPageComments.action?callback=fetchJSON_comment98vv44765&productId=952862&score=0&sortType=5&page={page}&pageSize=10&isShadowSku=0&rid=0&fold=1

依云

https://item.jd.com/1384057.html

https://sclub.jd.com/comment/productPageComments.action?callback=fetchJSON_comment98vv6519&productId=1384057&score=0&sortType=6&page={page}&pageSize=10&isShadowSku=0&rid=0&fold=1

巴黎水

https://item.jd.com/1109759.html

https://sclub.jd.com/comment/productPageComments.action?callback=fetchJSON_comment98vv15686&productId=1109759&score=0&sortType=5&page={page}&pageSize=10&isShadowSku=0&rid=0&fold=1

西藏5100

https://item.jd.com/952875.html

https://sclub.jd.com/comment/productPageComments.action?callback=fetchJSON_comment98vv2664&productId=952875&score=0&sortType=5&page={page}&pageSize=10&isShadowSku=0&rid=0&fold=1

怡宝

https://item.jd.com/5258536.html

https://sclub.jd.com/comment/productPageComments.action?callback=fetchJSON_comment98vv3789&productId=5258536&score=0&sortType=5&page={page}&pageSize=10&isShadowSku=0&rid=0&fold=1

龙采冰海

https://item.jd.com/13139977525.html

https://sclub.jd.com/comment/productPageComments.action?callback=fetchJSON_comment98vv104&productId=13139977525&score=0&sortType=5&page={page}&pageSize=10&isShadowSku=0&rid=0&fold=1

京东产品评论只能显示100页，每页10个评论，相当于每个产品链接我们能获取1000条评论数据。

In [46]:

```
import requests
import json
import csv


#定义数据采集函数
def get_data(productname, product_url, comment_template_url):
    """
    productname: 产品名
    product_url: 在京东上的产品链接
    comment_template_url: 该product_url产品页对应的评论数据网址模板
    """
    csvf = open('data/{}.csv'.format(productname), 'a+', encoding='gbk')
    writer = csv.writer(csvf)
    writer.writerow(('date', 'comment'))
    cookies = {'cookie': 'shshshfpa=d998ad68-8ef8-97d9-ce77-f63bb2d33de2-1571647
890; shshshfpb=bZ1QOzBh7YhaSdc4Fr3dVdw%3D%3D; unpl=V2_ZzNtbUdXSxZ3XUVdfUpbBGIERl
URVxYRc1oSVyxJWQBiBkINclRCFX0URlRnGl4UZAMZX0FcQRJFCEVkexhdBGMBEVxLVHMlRQtGZHsYbA
VjBRJaR1FKHH0NRlB%2bG1kNbgsVWUBncxJ1AXZkKEkEWD9cRDMAExVFNjhHZHopXABmChZZQl5CFEVD
KFU2GVgDZwQXW0teSxB1DENWfhFVDWAHEG1DZ0A%3d; __jdv=76161171|baidu|-|organic|not s
et|1575355760921; user-key=6289d37e-1567-4003-b5f9-dc4e45949d34; cn=0; areaId=1
0; ipLoc-djd=10-698-45817-0; PCSYCityID=CN_230000_230100_0; __jda=122270672.1571
647889415434989653.1571647889.1576125858.1576400368.6; __jdc=122270672; 3AB9D23F
7A4B3C9B=NC3EVKVFFONHN6WQUVJGZPHNO7SHXRK23BHE7T3BSUUMK3HEXLJWN6NTTUGBQPXBSXYZKPA
XV6BFW2AC5TXQBFOLVQ; shshshfp=ea79df3ba9ee59ec3f289bf52cae6f66; shshshsID=c07947
544c53ae265da1d661640a492c_5_1576400562558; __jdb=122270672.5.157164788941543498
9653|6.1576400368; JSESSIONID=A6FD75A7DDCD0D3354A75F4DF9475F44.s1'}
    headers = {"user-agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_1) Ap
pleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.79 Safari/537.36",
               "referer": product_url}

    #京东评论只能抓100页评论
    for page in range(0, 100):
        url = comment_template_url.format(page=1)
        resp = requests.get(url, headers=headers, cookies=cookies)
        try:
            alldatas = json.loads(resp.text[27:-2])
        except:
            alldatas = json.loads(resp.text[26:-2])
        else:
            alldatas = json.loads(resp.text[25:-2])

        for comment in alldatas['comments']:
            commnt = comment['content']
            date = comment['creationTime']
            #print(date, commnt)
            writer.writerow((date, commnt))

    csvf.close()

#执行采集数据(以依云为例)
productname='依云'
product_url = 'https://item.jd.com/1384057.html'
comment_template_url = 'https://sclub.jd.com/comment/productPageComments.action?
callback=fetchJSON_comment98vv6519&productId=1384057&score=0&sortType=6&page={pa
ge}&pageSize=10&isShadowSku=0&rid=0&fold=1'
get_data(productname, product_url, comment_template_url)
```

文本分析

在本节主要是文本的词频可视化。步骤:

1. 读取文本数据
2. 分词/统计词频
3. 输出词云图

In [12]:

```
import re
import os
import jieba
import csv
import pandas as pd
from pyecharts import options as opts
from pyecharts.charts import Page, WordCloud
from pyecharts.globals import SymbolType


#定义词云图函数
def word_cloud(product):
    df = pd.read_csv('data/{}.csv'.format(product), encoding='gbk')
    texts = ''.join(df['comment'])
    #剔除非中文的内容（只保留中文）
    texts = ''.join(re.findall(r'[\u4e00-\u9fa5]+', texts))

    #jieba分词
    wordlist = jieba.lcut(texts)
    wordset = [w for w in set(wordlist) if len(w)>1]
    wordfreq = []
    #词语计数
    for word in wordset:
        freq = wordlist.count(word)
        wordfreq.append((word, freq))


    #词频排序
    wordfreq = sorted(wordfreq, key=lambda k:k[1], reverse=True)
    wordcloud =WordCloud()

    wordcloud.add("",
                  wordfreq,
                  word_size_range=[20,100])

    wordcloud.set_global_opts(title_opts=opts.TitleOpts(title=product))


    wordcloud.render('output/{}.html'.format(product))


# 执行词云图函数(以依云为例)
word_cloud(product='依云')
```