B 站“每周必看”排行榜视频数据分析

——《数据工程》期末作业

刘智宇　1950083　自动化

摘要：近年来，随着视频网站和应用程序发展迅猛，人们更加依赖于通过视频的方式获取信息。对学生党，B 站是一个主流的视频网站。为此，本文希望对 B 站的“每周必看”排行榜进行分析和研究。首先利用 Python 爬虫技术从 B 站的“每周必看”排行榜自动化地获取上榜视频的类别标识、观看量、点赞量等关键数据信息，然后对上述收集到的关键数据信息进行分析并利用数据可视化技术进行展示。从而得出 B 站“每周必看”用户群体画像、用户所关注的热点方向以及受欢迎的视频的特征，为有志于成为 B 站 UP 主的用户提供一个参考。

关键词：B 站、视频、排行榜、Python 爬虫、可视化

一、案例背景介绍

随着视频网站和短视频 APP 的快速崛起，人们似乎更加习惯于通过视频来获取知识和信息。通过短视频，人们可以快速获取碎片化的信息；而通过时长较长的视频课，学生们也可以系统的学习自己感兴趣的知识，对校内知识进行自主的巩固、拓展和提高。

本人作为一名学生党，常常使用 B 站进行校内外各类课程的学习。在课余时间我会在 B 站上冲浪、娱乐、放松，有时也会自己发布视频分享生活和学习心得。所以，我希望可以通过使用本学期余有灵老师所教授的 Python 网络爬虫技术，自动化的获取 B 站“每周必看”排行榜上的视频及其信息。

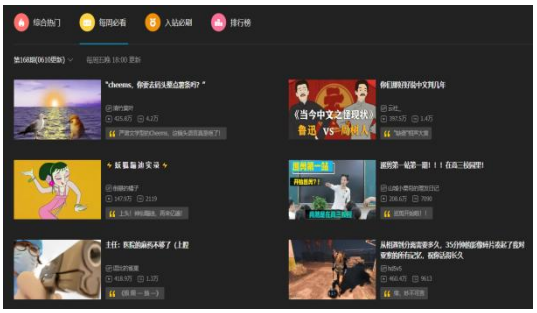
由于 B 站的“本周必看”排行榜刊登了近一个星期 B 站用户最关注、最感兴趣的几个视频，所以通过分析这些上榜视频的类别标签、关键词、观看量、点赞量、收藏量、分享量等信息，可以间接获取 B 站用户最喜欢的视频类型以及用户群体最关注的热点话题。将这些信息和数据通过可视化技术进行展示，可以更加直观的感受到受欢迎的视频所共有的特征。

今后如果有用户希望成为一名 UP 主，希望上传一些视频并获得关注，本次数据分析也可以作为一个参考，进而指导他们更有针对性的进行视频的创作和投稿。下面附一张我自己的 B 站首页截图，可以看到投稿的视频播放量较为惨淡，所以我以后投稿视频的时候也会按照本次分析结果对视频进行内容、时长等进行调整。

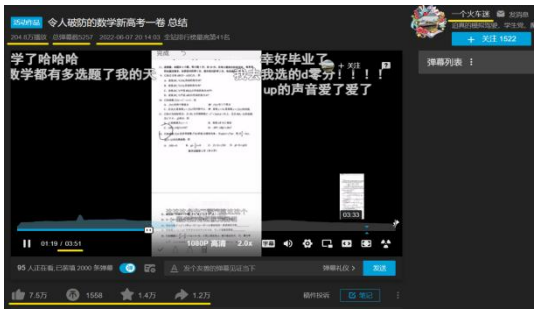
二、案例相关数据集表述

由于此次数据分析的选题是出于我自己的兴趣，所以并没有刻意去网站上搜索数据集，而是通过余有灵老师所教授的 Python 网络爬虫技术，自己从 B 站上获取的。由于从每周必看排行榜上只能获取视频名称、UP 主名称、播放量、弹幕数等简单数据信息，无法有效的进行数据分析，我们还需要进入视频详情页继续获取视频信息。

排行榜和视频详情页如下图所示：

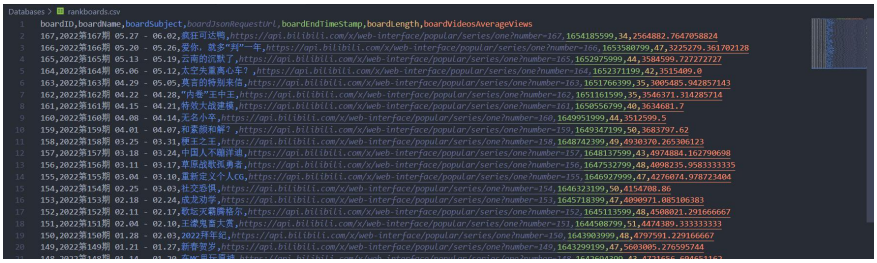


“每周必看”排行榜



视频详情页

对于每周必看排行榜，我获取了近一年的排行榜，共计 50 期。通过爬虫自动化的获取排行榜 ID、排行榜名称、排行榜主题、排行榜 URL、排行榜截止时间的时间戳等信息，并将其保存在 csv 文件中作为数据集，如下图所示。



此外，我将从视频详情页获取的信息，如视频 BV 号（唯一标识符）、视频名称、视频作者 ID、视频观看量等信息存储在另一个 csv 文件中作为另一个数据库，共计 2094 条数据，如下图所示：

最后我还将有视频上榜的 UP 信息汇总到第三个数据集中。如下图所示：

```
userID,userName,userFollows,userFans,userVideos
13407784,大宽物理,317,528254,222
9030440,修一j,310,473540,118
137429365,火山哥哥,367,694247,20
335692980,百万剪辑狮,175,1051309,90
627888730,星有野,93,3025391,56
5970160,小潮院长,99,7653209,267
14804670,无穷小亮的科普日常,87,7570794,225
99157282,盗月社食遇记,609,6839290,484
11688464,天才小熊猫本人,11,2115838,9
177291194,DECO27_Official,3,473190,34
```

三、案例实现过程

本次数据工程大作业案例实现主要分为两步：

1、通过 Python 网络爬虫技术获取所需数据并存储到本地，便于后续数据分析。

2、读取保存在本地的数据，通过可视化方法对数据进行展示和分析。

在这里先说明数据的获取及保存的实现过程。在编写爬虫获取数据前，我分析了需要获取的关键信息，并利用了面向对象思想设计了三个类：B 站排行榜类、B 站视频类、B 站用户类。三个类的成员变量如下图所示：

B 站视频类

```
class VideoClass():
    def __init__(self, videoUrl):
        self.videoBVID = None
        self.videoCID = 0
        self.videoName = None
        self.videoOwnerID = 0
        self.videoUrl = videoUrl
        self.videoViews = 0
        self.videoDanmaku = 0
        self.videoLength = 0
        self.videoLikes = 0
        self.videoCoins = 0
        self.videoFavorites = 0
        self.videoComments = 0
        self.videoShares = 0
        self.videoRecommendReason = None
        self.videoTag = None
        self.videoReleaseTimeStamp = 0
        self.videoBoardID = 0
        self.videoBoardRank = 0
```

B 站用户类

```
class UserClass():
    # UserList = []
    def __init__(self, userVideoUrl):
        self.userID = 0
        self.userName = None
        self.userFollows = 0
        self.userFans = 0
        self.userVideos = 0
```

B 站排行榜类

```
class RankBoardClass():
    def __init__(self, boardJsonRequestUrl):
        self.boardJson = None
        self.boardID = None
        self.boardName = ""
        self.boardSubject = ""
        self.boardJsonRequestUrl = boardJsonRequestUrl
        self.boardEndTimeStamp = None
        self.boardLength = 0
        self.boardVideosAverageViews = 0.0
        self.boardVideosList = []
```

在设计完成员变量后，我进一步对成员函数进行编写，每个类实现的成员函数如下一页的表格所示。由于函数具体实现较为繁琐这里不进行完整展示，我后续会将完整所有代码开源到 Github 和网盘上，链接见附录：

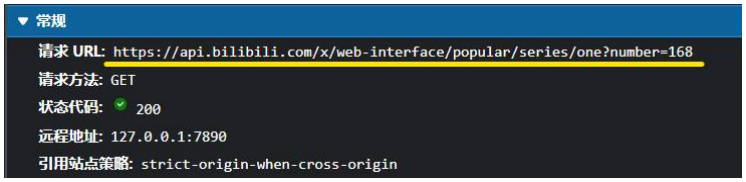
类名	函数定义	函数功能解释
Video Class	def showVideoInfo(self):	展示视频各种信息
	def initVideoLength(self):	获取视频时长，由于无法和其他数据一同获取故单独处理
	def initVideoInfoFromBoard(self, videoBoardRank, boardJson, boardID):	从“每周必看”排行榜先获取视频的部分信息
	def saveOneselfToDatabase(self, databasePath):	将本条视频数据存储在数据库中
	def initAllVideoInfo(self, videoBoardRank, boardJson, boardID):	调用上面实现的成员函数获取视频全部信息，并且将数据保存在 csv 文件中
User Class	def showUserInfo(self):	展示用户各种信息
	def initPartUserInfoFromBoard(self, videoBoardRank, boardJson):	从“每周必看”排行榜先获取用户的部分信息
	def initOtherUserInfo(self):	从用户首页获取其他信息
	def saveOneselfToDatabase(self, databasePath):	将本条用户数据存储在数据库中
	def initAllUserInfo(self, videoBoardRank, boardJson):	调用上面的成员函数获取用户全部信息并且保存在 csv 文件中
Rank Board Class	def showBoardInfo(self):	展示排行榜各种信息
	def addOneVideoToBoardList(self, videoRank):	创建一个视频对象添加在排行榜列表中（类似于树结构）
	def addAllVideoToBoardList(self):	将所有视频对象添加在排行榜中
	def computeBoardVideosAverageViews(self):	计算当前排行榜所有视频的平均播放量
	def addOneUser(self, videoRank):	创建一个用户对象并保存到 csv 文件
	def addAllUsers(self):	调用上面的函数保存排行榜视频所属 UP 主信息到 csv 文件
	def saveOneselfToDatabase(self, databasePath):	将当前排行榜的关键信息数据保存到 csv 文件中
	def initBoardInfo(self):	调用成员函数获取排行榜全部信息，并且将数据保存在 csv 文件中，同时也会调用 user 和 video 类的 initInfo 函数对数据进行保存

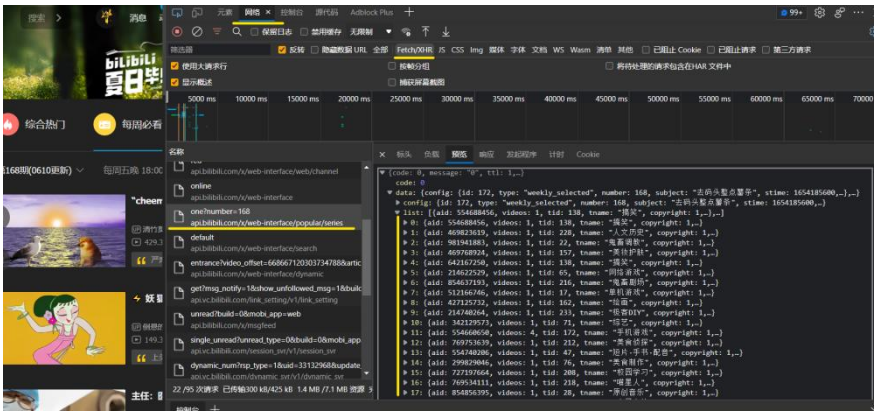
需要说明的是在使用爬虫获取 B 站数据时，我遇到了几处困难：

①为了获取较为全面的信息，我不得不通过跳转多个页面才能将一期“每周必看”排行榜的一个视频及其创作者的关键数据获取完整。

②此外由于 B 站可能设置了一些反爬手段，导致课上所教的添加浏览器请求头后，通过网页原始 URL 获取 HTML 文件的方法失效。

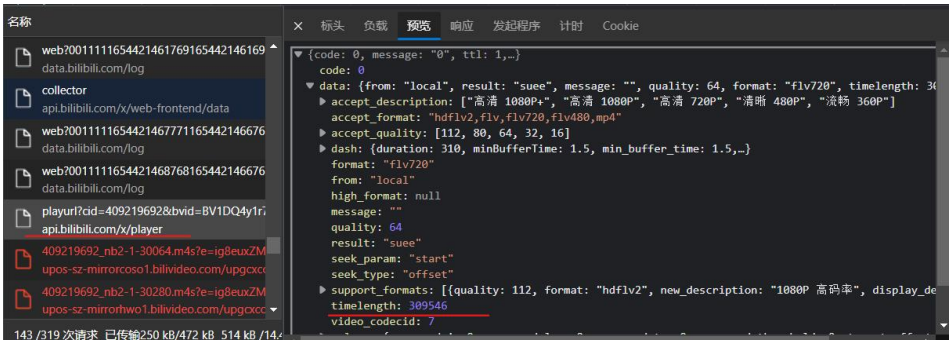
起初，由于是个爬虫方面的新手，我并不知道这些问题应该如何解决，有多次想要更换选题。通过和余有灵老师的多次沟通，我反复查阅资料并不断尝试（包括 selenium、scrapy 等方式），最后终于用较为简单的方法实现这些关键数据的获取：就是不直接通过网页原本的 URL 获取 HTML，而是直接去访问 B 站的 api 接口，得到 json 数据，然后可以使用 request 包从 json 中直接提取相关数据。如下图所示：





排行榜 api 的 URL 规律也较为明显，即”number=”后填入想要访问的排行榜期号，即可获取该期的排行榜的 json 文件。将排行榜对应的 json 文件中上榜视频 BV 号和 CID 等数据的提取并保存后，我们就可以通过 BV 号以及 cid 编号跳转到视频页，获取视频的时长等数据。

值得注意的是，视频长度信息仍然需要通过访问 B 站视频播放器 api 接口获取相应 json 文件后才能提取（如下图所示），需要将 cid 和 BV 号填在 api 的 URL 规定位置。在编写报告时，我发现 B 站已经使用了新的 api 接口，旧 api 虽然还是可以访问以获取所需数据，但是在 Edge 的开发人员工具中已经无法捕获相应的请求和响应了。为了防止旧 api 接口的淘汰，我会在之后调整代码，保证爬虫的时效性。



在获取完视频信息后，还要获取用户数据（包括投稿视频数量、粉丝数、关注数等信息），该流程、步骤和上面的基本相同，这里不再赘述了。

获取到某一期排行榜的榜单信息以及对应的全部视频和视频作者的各项数据后，我将这3份数据分别保存在3个csv文件中。为了方便调用，我将数据存储写在各个类中，作为类的成语变量，并且在initInfo()后会自动调用saveOneselfToDatabase()函数（如下图所示，使用视频类的进行展示），这样就免于显式指明数据存储，对于本次爬虫显然更加方便。

```
def saveOneselfToDatabase(self, databasePath):
    with open(file=databasePath, mode='a', encoding="utf-8", newline="") as databaseFile:
        databaseWriter = csv.writer(databaseFile)
        if os.path.getsize(databasePath) == 0:
            print(databasePath, "is Empty. Do initialization.")
            databaseWriter.writerow(["userID", "userName", "userFollows", "userFans",\
                                     "userVideos"])
        databaseWriter.writerow([self.userID, self.userName, self.userFollows,\
                                 self.userFans, self.userVideos])
```

另外，在排行榜类的initInfo()函数中调用了视频类和用户类的initInfo()，所有相当于我们只需通过调用排行榜类的初始化函数即可将该榜的全部数据进行保存，在我看来，这样的封装十分的“优雅”（如下图所示，只需接收排行榜期号就可以完成数据爬取与存储）。

```
def getOneRankBoardAllInfo(boardID):
    print(SeparateBar)
    print("获取第%3d期排行榜数据信息"%(boardID))
    startTime = time.time()
    fullBoardJsonRequestUrl = RankBoardJsonRequestUrlPrefix + str(boardID)
    tempRankBoard = RankBoardClass(boardJsonRequestUrl=fullBoardJsonRequestUrl)
    tempRankBoard.initBoardInfo()  # Init 的时候自动将信息存入
    print("RankBoardName    :", tempRankBoard.boardName)
    print("RankBoardLEngth  :", tempRankBoard.boardLength)
    endTime = time.time()
    spendTime = endTime-startTime
    print("获取第%3d排行榜数据信息，耗时%.3f秒"%(boardID, spendTime))
    print(SeparateBar)
    return tempRankBoard
```

最后，我爬取了近50期排行榜，也就是近一年的“每周必看视频”用于此次B站排行榜视频数据分析。我首先使用pandas包将数据从csv文件中读取到DataFrame中以便分析随后使用matplotlib包绘制了一些折线图，直观展示了B站“每周必看”排行榜的关键数据的趋势（如视频平均播放量、上榜视频数量等）。

```
plt.figure(figsize=(20,5))
plt.subplot(1, 2, 1)
plt.title("排行榜视频数量折线图")
plt.xlabel("排行榜期数")
plt.ylabel("排行榜视频数量")
plt.grid()
plt.plot(BoardIDs, BoardLengths, linestyle='-', color='red', marker='x', linewidth=1.5)
```

随后对排行榜视频的标签进行归类和统计，得到近一年 B 站用户最关注的视频分区的饼图（具体图片见后续结论部分，这里仅展示部分代码），可以由此分析出 B 站用户最喜爱的视频类型，进而可以推断出用户画像。

```
print(TagDict)
print(SeparateBar)
plt.figure(figsize=(15,15))
textProps = {"fontsize":18, "color":"w"}
plt.title("上榜视频标签分析饼图", fontdict={"fontsize":25, "color":"w"})
plt.pie(TagDict.values(), labels=TagDict.keys(), autopct="%1.1f%%", \
textprops=textProps, shadow=False, startangle=180);
```

除了视频分区，我认为视频的标题也是十分重要信息，从上榜“每周必看”的视频的标题，我们能够更进一步的获取 B 站用户最关注的热点话题。同时，从视频的推荐理由中，我们也能获得很多启发，它可以告诉 B 站的各个 UP 主观众更看重视频的哪些特质。这两个信息可以为视频创作者们今后的视频制作提供一些启发和思路。由于这两个数据都是字符串类型的，所以需要利用自然语言处理等高级手段进行关键词提取、分析和可视化展示。这里我使用了余有灵老师所推荐的几个包，SnowNLP、wordcloud 等，可视化了词频（具体结果也会在结论中展示），这里仅展示部分代码。

```
def getWordCloud(imagePath, pltTitle, keywordsListWithFreq, recolor):
    backgroundImage = np.array(Image.open(imagePath))
    imageColors = wordcloud.ImageColorGenerator(backgroundImage)
    keywordsDictWithFreq = {}
    for oneTuple in keywordsListWithFreq:
        keywordsDictWithFreq[oneTuple[0]] = oneTuple[1]
    fontPath = r'c:\Windows\Fonts\simfang.ttf'
    WordCloud = wordcloud.WordCloud(mask=backgroundImage, background_color="white",
width=698, height=698, font_path=fontPath)
    WordCloudPic = WordCloud.generate_from_frequencies(keywordsDictWithFreq)
    plt.figure(figsize=(20,40))
    plt.subplot(1,2,1)
    plt.title(pltTitle)
    if recolor is True:
        WordCloudPic = WordCloudPic.recolor(color_func=imageColors)
    plt.imshow(WordCloudPic)
    plt.subplot(1,2,2)
    plt.title("模板原图")
    plt.imshow(backgroundImage)
```

最后，我还利用柱状图对对于视频的各个指标进行了跟踪分析，同时也对各个指标之间的相关性进行了分析，并且使用 seaborn 包中的 heatmap 函数进行相关性矩阵的展示。通过这些指标，我们可以分析用户群体的观看习惯（同样的具体分析和可视化在结论部分）。

四、结论

首先展示我通过 Python 网页爬虫技术获取的数据集基本信息，使用 pandas 的 info()方法，获取数据集的描述。可以看到共获取 2093 条视频数据、50 期排行榜数据、2093 个用户信息，以及每一条数据包含的各个列的信息描述。这里需要说明的是，数据集信息都是截至我进行数据爬取时的数据，后续的更新暂时不在我的考虑范畴内。

视频数据集

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2093 entries, 0 to 2092
Data columns (total 16 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   videoBVID             2093 non-null   object
 1   videoName             2093 non-null   object
 2   videoOwnerID          2093 non-null   int64
 3   videoViews            2093 non-null   int64
 4   videoDanmaku          2093 non-null   int64
 5   videoLength           2093 non-null   int64
 6   videoLikes            2093 non-null   int64
 7   videoCoins            2093 non-null   int64
 8   videoFavorites        2093 non-null   int64
 9   videoComments         2093 non-null   int64
 10  videoShares           2093 non-null   int64
 11  videoRecommendReason  2092 non-null   object
 12  videoTag              2093 non-null   object
 13  videoReleaseTimeStamp 2093 non-null   int64
 14  videoBoardID          2093 non-null   int64
 15  videoBoardRank        2093 non-null   int64
dtypes: int64(12), object(4)
memory usage: 261.8+ KB
videosDatabaseDF
 None
```

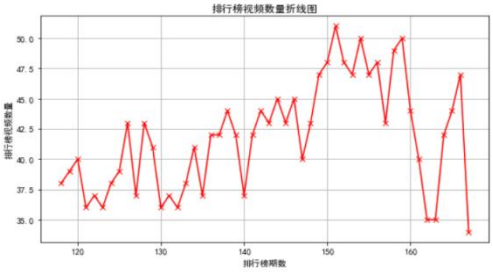
排行榜数据集

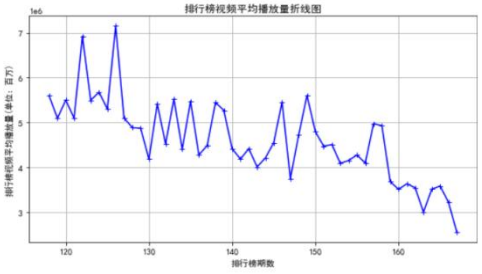
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 7 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   boardID                50 non-null     int64
 1   boardName              50 non-null     object
 2   boardSubject           50 non-null     object
 3   boardJsonRequestUrl    50 non-null     object
 4   boardEndTimeStamp      50 non-null     int64
 5   boardLength            50 non-null     int64
 6   boardVideosAverageViews 50 non-null    float64
dtypes: float64(1), int64(3), object(3)
memory usage: 2.9+ KB
rankBoardsDatabaseDF
 None
```

用户数据集

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2093 entries, 0 to 2092
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   userID       2093 non-null   int64
 1   userName     2093 non-null   object
 2   userFollows  2093 non-null   int64
 3   userFans     2093 non-null   int64
 4   userVideos   2093 non-null   int64
dtypes: int64(4), object(1)
memory usage: 81.9+ KB
usersDatabaseDF
 None
```

随后，展示排行榜相关信息的走势图，我主要关注了排行榜长度和排行榜平均播放量两个信息。可以从下图看出排行榜长度平均值在 42 左右，并且随着当周视频总体质量会有响应的波动。另外，从上榜视频平均播放量折线图，我们也可以看出在近一年的时间里，大部分时间平均播放量都维持在五百万左右。结合排行榜放榜的日期我发现，左侧的峰值对应的正是暑假的时候，而右侧的低谷则对应的学生期末的阶段。由此我们可以据此推断，B 站的用户有可能大部分为学生群体。

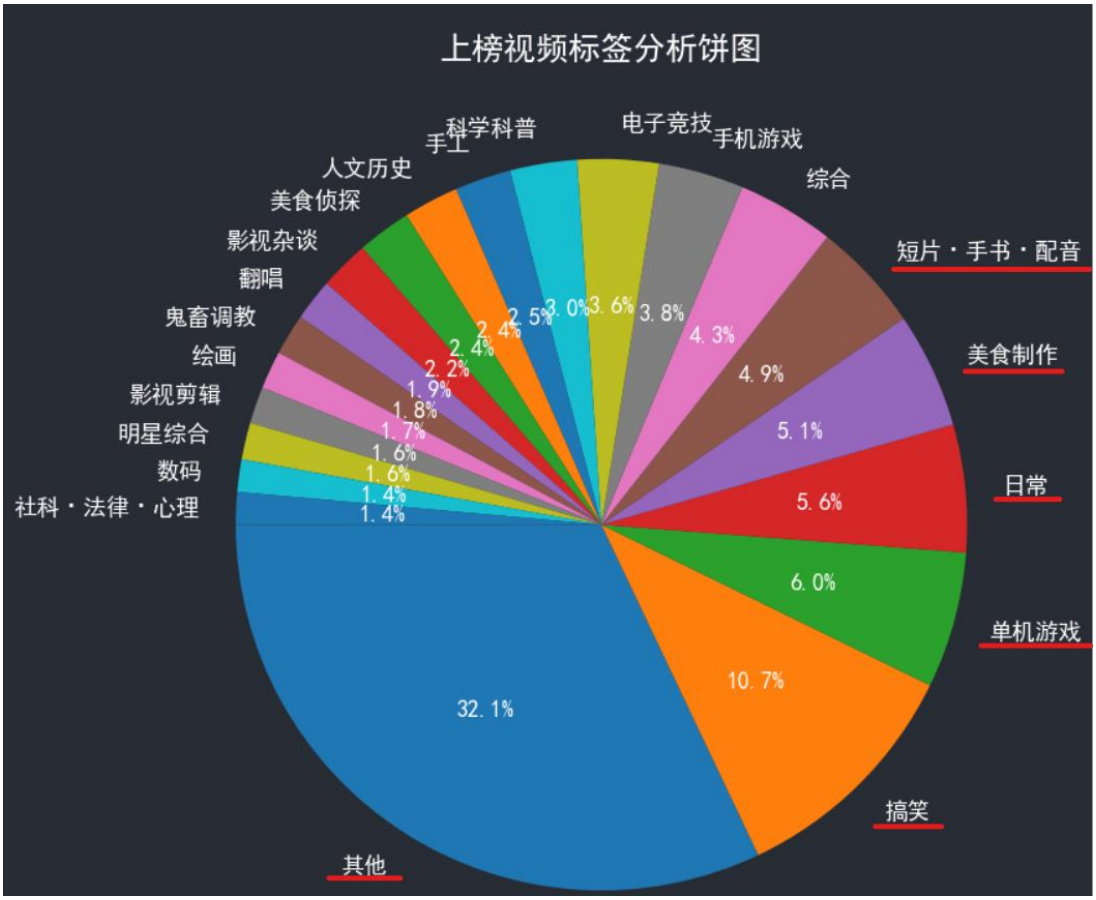




此外，上榜视频类别占比也很能说明问题，我在此通过饼图来进行展示和分析（红色下划线是我后期添加的）。显然，在众多视频类别中，“搞笑”是最多的占比 10.7%，高出第二名“单机游戏”近 5%，随后是“日常”、“美食制作”、“短片·手书·配音”等等。

由此可见，B 站用户普遍喜欢看一些放松类的搞笑视频，能够缓解平时的压力；也喜欢看看其他人分享自己的日常生活，如美食制作等等。同时除了排名第二，占比 6%的“单机游戏”，还有占比分别为 3.8%和 3.6%的“手机游戏”和“电子竞技”，显然 B 站用户的“游戏瘾”也比较大。

在饼图的左下角一大块区域对应的是“其他”，这个其他并非是视频的标签，而是由于分类过多，我将占比较小的都进行汇总。但由此也可以看出 B 站是一个包容性很强的平台，就算是比较小众的类别，如果视频本身的质量够格，也可以被平台发现，并向 B 站的所有用户进行推荐。



对于 B 站的各个 UP 主来说，视频标题也是很重要的，一个吸引人的标题往往能够增加视频的播放量、点赞量、投币量等等关键指标，进而给创作者带来更多的粉丝和收益。为此，我使用了余有灵老师推荐的几个工具包，对近一年的上榜视频的标题关键字进行了词频分析，并将词云进行展示，如下图所示（蓝色下划线是我后期添加的）：

上图中，字体较大的词语就是词频较高的关键词（颜色深浅只是为了迎合右侧的模板）。据此我们可以分析出如果视频标题中涉及热度较高的话题，如原神（一款近期火爆的游戏）、游戏、动画等，则比较容易获得大量的曝光度。

同时，用户们也十分看重视频创作者的付出，那些“硬核”、“耗时”、“自制”的视频获得推荐也是十分合理的，所以今后视频创作者对于视频也应该更加的用心付出，用自己的真心创作换用户真心的点赞、投币。

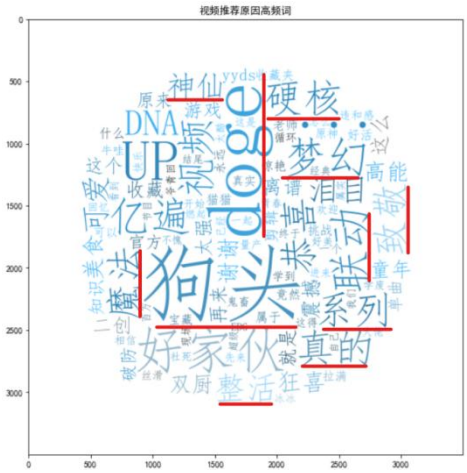
当然如果一个 UP 主是十分出名的，那么跟他相关的视频或是他自己创作的视频也理所应当能够比较火爆，比如“罗翔”老师有着 B 站最多的粉丝数，自然也是 B 站的一个“关键词”。

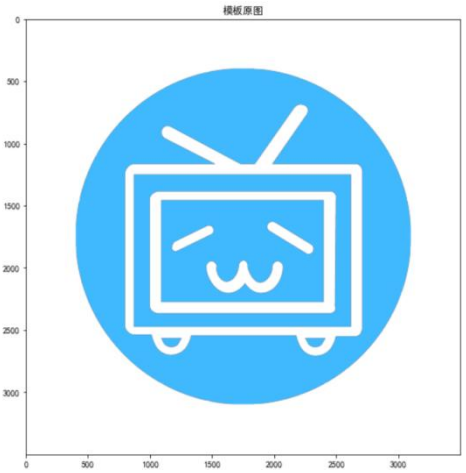
除了可以对视频标题进行进行词频分析，我们也可以对视频的上榜理由进行分析。我用类似的方法对视频推荐理由高频词进行了分析，如下图所示（红色下划线是我后期添加的）。

首先，字体最大的两个词语是“doge”、“狗头”和“整活”，由此我们能分析出两个信息：①B 站用户普遍比较幽默，间接反映了用户群体的年龄段并不是特别高，应该是学生为主；②这些评论的视频也是娱乐整活类型的，也应证了用户对于“搞笑”、“整活”类别的视频的喜爱。

此外，“硬核”、“魔法”、“神仙”等推荐词也反映了用户的关注方向。对于倾注了创作者大量的心血的视频，用户群体在推荐时也不吝美言。

同时，“联动”、“系列”视频也很受观众们的喜爱，所以日后 UP 主在创作视频时也可以多多考虑是否可以将有相关性的视频做成一个系列或者和其他 UP 主进行联动，以获得更好的视频反馈。

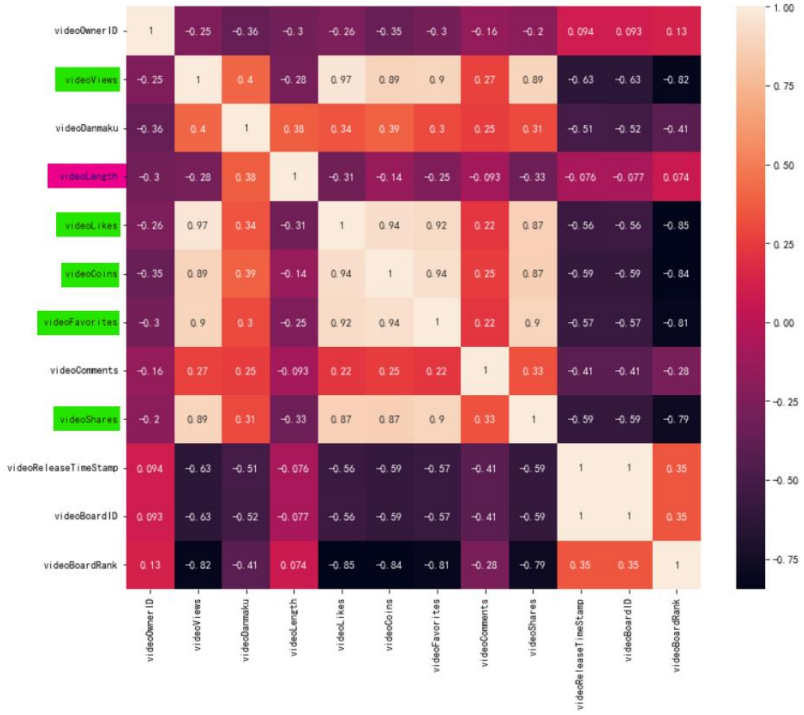




最后，我对于视频各个关键参数进行了相关性分析，并且用 seaborn 工具包中的热力图对相关性矩阵进行了展示（浅色代表强相关、深色代表弱相关）。不难发现，可以衡量视频的曝光度的参数：播放量、点赞量、投币数、收藏数、分享数等都有较强的相关性，说明对于高质量视频，观众是普遍愿意更多的去点赞、投币、分享它们。

这个现象其实是一个良性循环，因为创作者会因为自己的视频收获比较可观的“创作激励”，进而能够更好的全身心投入视频创作中。这种积极的创作环境或许也解释了 B 站近些年的迅速发展的原因。

同时也有值得警觉的部分，可以看出视频长度和点赞、投币、收藏等数据有一定的负相关，说明短视频逐渐在 B 站占据了主导的地位，这其实并不是什么好事。我个人在使用 B 站也有类似的感觉，类似抖音、快手等平台的短视频大量的涌入 B 站。这些短视频没有什么营养，很多也在打“擦边球”，却能收获不少的曝光度。短视频的进入在我看来是污染了 B 站原先的良好氛围，希望 B 站“不要忘记自己的初心”。



最后，我还评选出了几个近一年的 Top 榜单，如下表所示：

最高产 up Top20

userName	userFans
罗翔说刑法	22736797
老番茄	17268968
哔哩哔哩英雄联盟赛事	11566510
原神	11554915
共青团中央	9581901
绵羊料理	9553728
老师好我叫何同学	9496558
央视新闻	9326856
敬汉卿	9282411
木鱼水心	8818431
中国BOY超级大猩猩	8199975
凉风Kaze	8195211
小潮院长	7654114
无穷小亮的科普日常	7572052
敖厂长	7562612
观察者网	7511299
徐大虾晗	7422845
某幻君	7320763
记录生活的蛋黄派	7184457
papi酱	7112534

粉丝最多 up 主 Top20

userName	userVideos
star星视频	31173
央视网	16300
观察者网	13165
中国日报	7237
央视新闻	4585
四川观察	4532
哔哩哔哩英雄联盟赛事	4019
共青团中央	3810
央视频	3609
央视网快看	3599
英雄联盟	3435
人民网	3098
新华社	2984
大司马工作室	2441
任性的KIMKILLS	2386
广东共青团	2271
观视频工作室	2270
中国长安网	2243
王者荣耀	2075
吟游大司人	1839

观看次数最多视频 Top20

videoName	videoBoardID	videoViews
⚡萨日朗！！！⚡	122	37104185
《孤勇者》（《英雄联盟：双城之战》动画剧集中文主题曲）	138	36340242
【时代少年团】「五月粉丝见面会实况」-《世界上的另一个我》纯享版	121	32033332
破亿纪念!【猛男版】新宝岛 4K高清重置加强版	120	31296464
人类高质量玩具！我蚌埠住了，哈哈哈哈哈哈	126	25900519
前方高能！《孤勇者》女声版 超A燃炸！！	139	25620104
《黑神话：悟空》12分钟UE5实机演试集锦	126	25033560
不是吧不是吧，别人发都火了，我自己发还能火吗？	126	23941129
《原神》剧情PV-「神女劈观」	146	22974335
【何同学】我做了苹果放弃的产品...	135	22004589
【S11全球总决赛】决赛 11月6日 EDG vs DK	138	21528129
“总有一天，全城的狗，都要高看我！”	144	20890745
《孤勇者》前方核能！谁说女生不适合唱这歌？	140	20136334
《孤勇者》前方核能！谁说女生不适合唱这歌？	141	20136296
⚡中国人不蹦洋迪⚡	157	19866771
无伤速通催逝员	143	19080432
呜呜，这也太可爱了吧！胡桃摇玩具终于来了！	133	18958873
买瓜大队	122	18888689
无伤反杀刘华强	125	18444704
17个简单有趣的小食谱 有手就能做系列	127	17511309

五、改进思路

本次大作业虽然已经进行了很多的数据分析，但是像 B 站这种视频网站能提供的信息绝对是“海量”的，所以还是有很多数据是我没有用到的。限于时间、设备等原因，我暂时没法更加的深入分析。如果日后有条件，我一定会获取跟多方面的数据，并从更丰富的角度来分析 B 站能够提供数据。下面是我能想到的几个比较切实可行的解决方案。

首先，除了榜单上的视频，我也应该去获取一些其他的没有上榜的视频信息。这样我就同时拥有了正例和反例，后续就可以应用更多的机器学习算法，根据视频对一个视频是否能够上榜进行预测。这个功能显然是十分有用的。

同时，如果以后硬件允许的情况下，我会将视频下方的全部评论都爬取下来，这样可以使用自然语言处理的方法，分析出用户对视频的评价。这个功能显然可以更好的帮助 UP 主们找到各个视频的短板，进而能够更好的进行视频创作。

此外，由于 B 站的视频详情页情况较为复杂，我本次数据分析对于视频详情页的有些内容处理的略微有些粗糙，例如是否属于联合投稿、是否是系列视频等等。上述分析已经说明了联合投稿和系列视频容易受到欢迎，所以以后我也会将这种影响进行量化，并进行可视化展示。

我还认为对于视频来说，热度趋势还是比较重要的。有些视频是“永远的经典”，其热度会持续很长时间，而有些视频的热度只是昙花一现。所以时序的信息还是十分重要的，他可以更好的反映视频的影响力。在今后我有能力时，也一定会将时序信息加入对 B 站的数据统计分析中，使得其更有说服力。

另外，本次分析中，我调用了一些现成的工具包，它们大大简化了我进行数据分析以及可视化的难度。但是我们应该做到不仅会用，还应该清楚其底层实现过程，并且可以尝试通过代码进行实现。这样，我们的编程能力和水平才能“更上一层楼”。

最后，我以后还可以多多尝试使用 mysql、postgresql 等数据库管理系统进行数据从存储，当然还可以去尝试 docker、hadoop 等技术。这些技术可以让我更加贴近未来就业，而且在面对更加海量的数据时，也是更好的解决方案。

六、参考文献

本次 B 站排行榜数据分析大作业并没有用到什么参考文献，主要是通过运用余有灵老师课上所教授的知识以及自学的一些爬虫技巧完成的。今后我会多多阅读相关论文。通过学习他人的分析方法，可以拓展我自己的数据分析思路，我也能更好的把握数据分析的重点和先进技术。

七、附录（网盘链接地址）

Github 链接：https://github.com/leizhenyu-lzy/BigHomework/tree/main/BilibiliRanking

百度网盘链接：https://pan.baidu.com/s/1i7Bjr2kBGnoVcMRfFAOMiA 提取码：yfiu