微博数据分析报告

时间：2020年6月10 日

[1.需求分析](#_heading=h.3znysh7)

[2.面临的问题](#_heading=h.2et92p0)

[3.初步分析](#_heading=h.tyjcwt)

[3.1 分析方向](#_heading=h.3dy6vkm)

[3.2 爬取过程](#_heading=h.1t3h5sf)

[3.2.1 抓包过程](#_heading=h.4d34og8)

[3.2.1.1 prelogin](#_heading=h.2s8eyo1)

[3.2.1.2 login](#_heading=h.17dp8vu)

[3.2.2 登陆过程](#_heading=h.26in1rg)

[3.2.3 BeautifulSoup解析HTML页面](#_heading=h.lnxbz9)

[3.2.3.1 利用美味汤判断登陆状态](#_heading=h.35nkun2)

[3.2.3.2 爬取腾讯机构号的数据](#_heading=h.1ksv4uv)

[3.2.3.3 初步数据分析](#_heading=h.44sinio)

[3.2.3.4 数据分析中遇到的问题](#_heading=h.2jxsxqh)

[4.数据提取和进一步数据分析](#_heading=h.z337ya)

[4.1我们需要哪些数据（针对腾讯视频官方号）](#_heading=h.3j2qqm3)

[4.2 数据的初步分析](#_heading=h.1y810tw)

[4.2.1对腾讯视频微博的点赞数，评论数以及转发数进行分析](#_heading=h.4i7ojhp)

[5.拟合模型](#_heading=h.2xcytpi)

[6.词频分析](#_heading=h.1ci93xb)

[7.个人总结（个人进步及存在的问题）](#_heading=h.3whwml4)

[通过本次项目系统学习了爬虫理论，但对于scrapy库的理解还是有些欠缺，对于python面向对象的知识有了一定的了解，复习了一下python可视化分析以及R语言进行简单线性模型的分析。](#_heading=h.2bn6wsx)

[8.参考文献](#_heading=h.qsh70q)

# 1.需求分析

对腾讯在微博上的机构号运营情况做数据分析，以推动后续的优化工作。

# 2.面临的问题

* 如何判断哪些是腾讯机构号？
* 拟定数学分析报告的维度划分。
* 使用爬虫以及numpy、seaborn、pandas、Matplotlib等对数据进行数据分析
* 为数据分析报告添加图表、文字分析，根据分析结果给出优化建议

# 3.初步分析

## 3.1 分析方向

对于问题一腾讯机构号的划分，我们很容易得出结论，在微博中搜索腾讯关键词，然后选取搜索中的‘找人’功能键，此时返回的结果即为我们想要的结果。在找到腾讯机构号后，我们发现搜索结果众多，有多达50页的数据，本分析报告只选取其中一部分腾讯机构号进行分析。



图-微博搜索找人关键词‘腾讯’



图-微博搜索找人关键词‘腾讯’返回50页结果

## 3.2 爬取过程

### 3.2.1 抓包过程

#### 3.2.1.1 prelogin

使用charles进行抓包，具体过程详见reference。主要抓去login信息，在抓取登陆信息时，首先清理浏览器缓存，目的是清理掉以往的登陆信息，其次在login界面输入账号密码登陆账号，最后在charles中查看抓到的登陆信息，可能需要多次抓包来获得登陆所需要的全部信息。

在prelogin中，浏览器会给我们发送一些信息，例如servertime，nonce以及pubkey(公钥)。

登陆微博时候我们所需要提交的表单信息包括：servertime，nonce，su，sp，\_(在prelogin中出现，是时间戳)等。

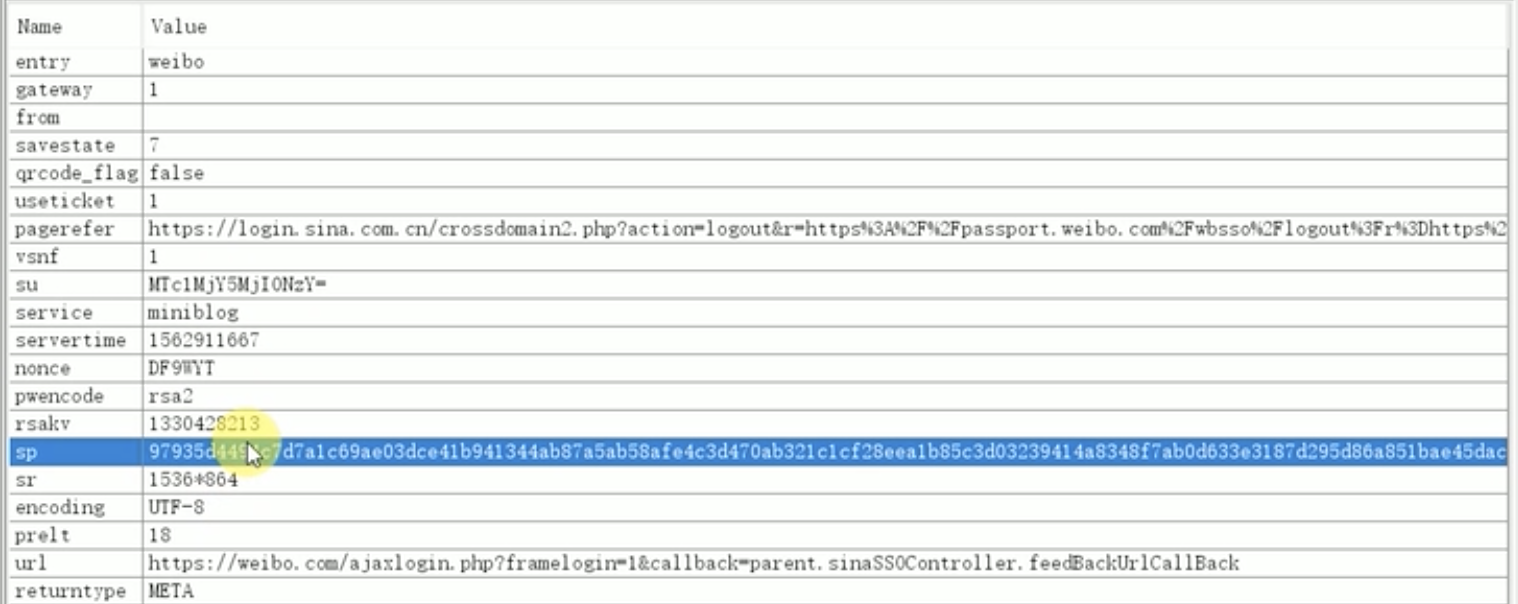


图-login表单



图-prelogin表单

在抓包分析中，最重要的是获取到login时候需要的js文件：此例中为ssologin.js文件，这里面会包含登陆时候处理我们表单信息的代码。

* sp：password(rsa和wsse两种加密方式，选取rsa，利用公钥进行加密)
* su：username(base64加密方式对用户名进行加密)

在prelogin中从浏览器得到的servertime，nonce以及pubkey会用于login时的登陆表单。

对于su，浏览器首先进行了urlencode（例如：腾讯 = %E8%85%BE%E8%AE%AF&，是一个编码过程），因此我们需要引入urllib库，使用urllib.parse.quote对字符串进行url编码。通过

base64.b64encode(urllib.parse.quote(self.user\_name).encode("utf-8")).decode("utf-8")获取su。b64encode不能够直接进行编码，需要utf-8编码再解码，然后获取我们想要得到的su。

随后我们需要在prelogin中取得我们需要的servertime，nonce以及pubkey。prelogin表单中的‘\_’即为十三位的时间戳，通过import time包中的time.time()方法获得，至此我们获得了username和十三位的时间戳进行prelogin。

之后我们制作header，制作header是要注意模拟浏览器访问：'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0'，并注意不要添加host和cookie信息。获取信息后，引入json包，获取json格式的信息（字典）。

#### 3.2.1.2 login

根据抓包得到的json代码，我们知道了password的加密为rsa非对称加密。下面是对password进行加密的过程：

#RSAKey.setPublic(me.rsaPubkey, "10001");

#password = RSAKey.encrypt([me.servertime, me.nonce].join("\t") + "\n" + password)

public\_key = rsa.PublicKey(int(self.get\_pre\_login()["pubkey"], 16), int("10001", 16))

password\_string = str(self.get\_pre\_login()["servertime"]) + '\t' + str(self.get\_pre\_login()["nonce"]) + '\n' + self.pass\_word

return binascii.b2a\_hex(rsa.encrypt(password\_string.encode("utf-8"), public\_key)).decode("utf-8")

我们在prelogin过程中得到了servertime以及nonce，因此利用json中的字符串逻辑：[me.servertime, me.nonce].join("\t") + "\n" + password，我们可以得到password\_string，之后对其进行utf-8编码，我们可以看到sp为十六进制的密码，因此需要引入binascii包进行转码，再解码之后得到十六进制的密码。

### 

### 3.2.2 登陆过程

本分析报告主要使用requests库模拟登陆微博获取我们需要的数据，使用requests库的session会话对象，可以跨请求保持某些参数，首先我们使用session成功登陆微博，之后再次使用该session对象对网页进行操作会默认调用之前保存的cookie、post表单等登陆信息。

登陆代码如下：

self.session.post("https://passport.weibo.com/wbsso/login", params = params, verify = False, headers = header)

* [https://passport.weibo.com/wbsso/login为登陆url](https://passport.weibo.com/wbsso/login%E4%B8%BA%E7%99%BB%E9%99%86url)
* header为登陆网页的请求头
* params为登陆网页需要post的参数
* verify = False 需要设置为false，方便抓包软件抓包

利用login()函数进行登陆，我们需要准备postdata，即需要向浏览器提供的信息，其中su，nonce，sp，servertime我们都已经获得，rsakv是从浏览器prelogin时候获得的因此可以调用函数得到rsakv值，此外需要修改returntype为text方便阅读。



图-postdata用于传递给浏览器的表单信息汇总

在login之后我们得到了ticket用于passport.weibo.com，接着我们构建login-ticket界面所需的参数：ssosavestate以及ticket。ssosavestate即为时间戳。之后携带新的parameters登陆

passport.weibo.com界面完成登陆。

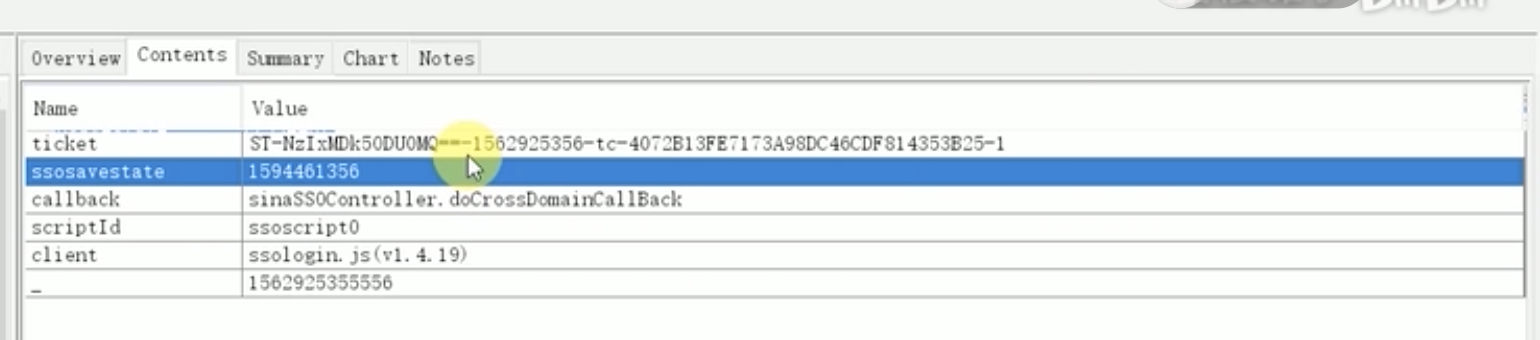


图-login-ticket界面参数

由于我们登陆的时候的设置为'Connection': 'keep-alive'，所以我们可以再次接入微博首页查看我们的登陆状态。这里我们使用session.post(‘[https://www.weibo.com’, verify](about:blank) = False, headers = header)来登陆微博首页。

### 3.2.3 BeautifulSoup解析HTML页面

#### 3.2.3.1 利用美味汤判断登陆状态

登陆微博首页后我们可以利用美味汤来检测是否登陆成功，此时import bs4库。

response = session.post('https://weibo.com',verify = False, headers = header)

soup = BeautifulSoup(response.text,'html.parser')

print(soup.find('title'))

#<title>我的首页 微博-随时随地发

我们可以通过soup.find('title')查看页面标题，发现标题出现了‘我的首页’字样，代表登陆成功。

#### 3.2.3.2 爬取腾讯机构号的数据

我们的目的是对微博上的机构号运营情况做数据分析，以推动后续的优化工作。在爬取数据之前我们首先要知道哪些数据需要爬取。

* 第一步：最直观的数据就是每个腾讯机构号的关注数，粉丝数以及发博数。获取这部分数据有三个子步骤：

1. 获取搜索结果的前n页url信息
2. 获取前n页相关用户的url信息
3. 爬取每位用户的关注数，粉丝数以及发博数
4. 存储为csv文件

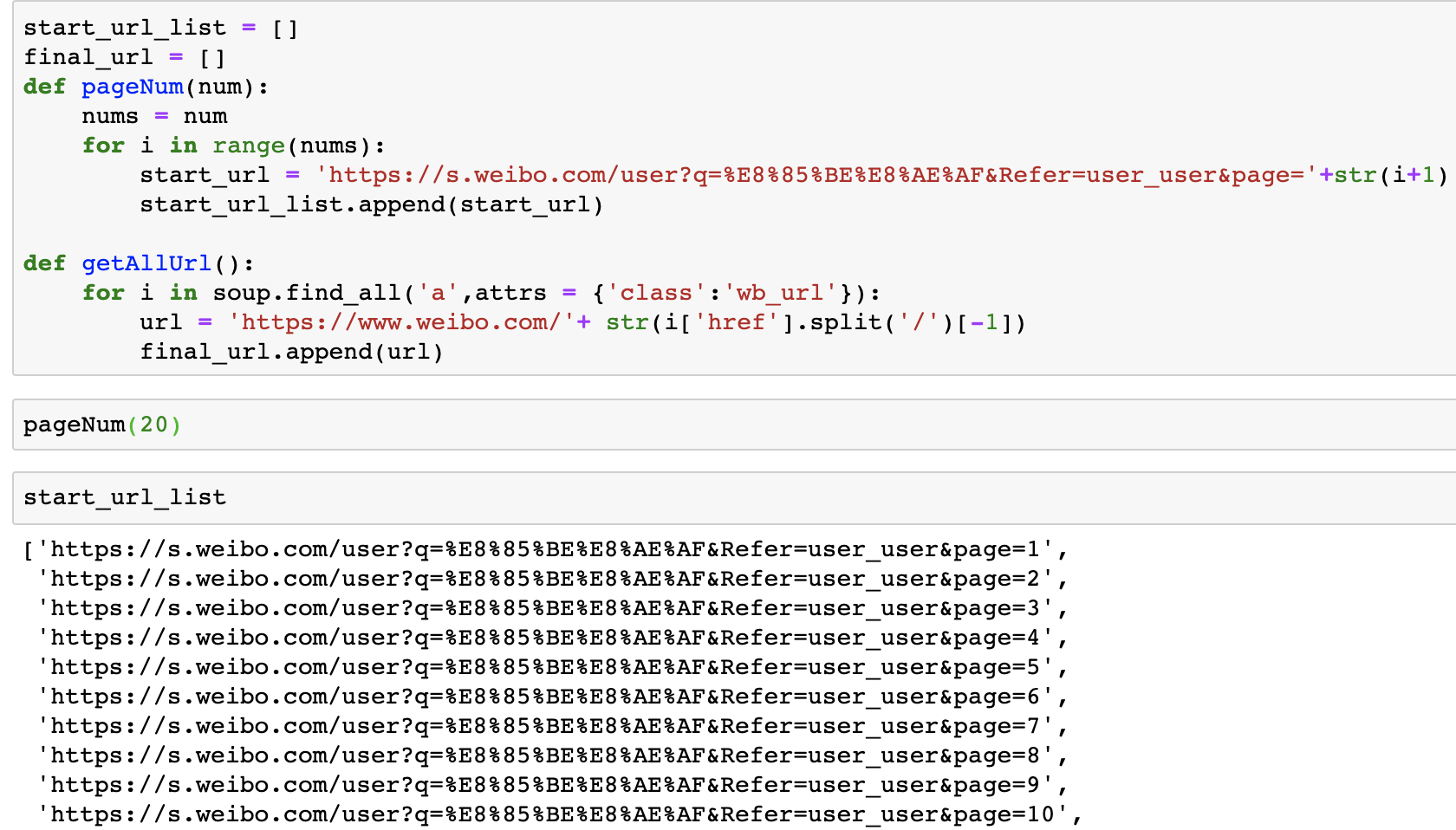


图-获取‘腾讯找人’前n页URL信息



图-获取前n页相关用户的url信息



图-获取前100位用户的关注数，粉丝数以及发博数

爬取结果如表所示(仅展示前11条)：

表-前100位腾讯机构号的关注数，粉丝数以及发博数

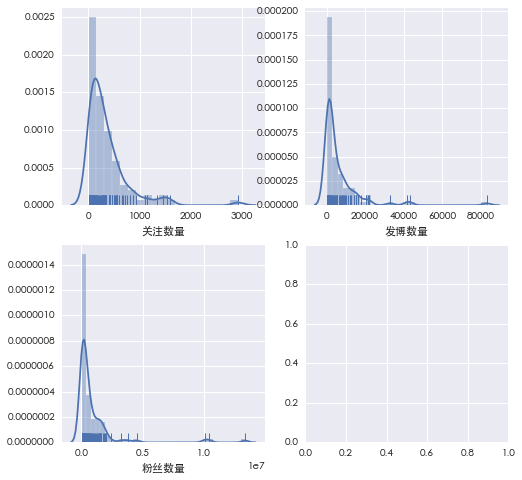
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 微博名称 | 关注数量 | 粉丝数量 | 发博数量 |
| 0 | 腾讯公司 | 165 | 671621 | 1064 |
| 1 | 腾讯视频 | 2921 | 13374071 | 83179 |
| 2 | 腾讯视频VIP | 684 | 1844971 | 22119 |
| 3 | 腾讯微信团队 | 42 | 501627 | 1053 |
| 4 | 腾讯微视 | 865 | 1073297 | 6980 |
| 5 | 腾讯电视剧 | 518 | 1406616 | 21349 |
| 6 | 腾讯新闻出品 | 312 | 608258 | 7961 |
| 7 | 腾讯动漫 | 867 | 1209234 | 15207 |
| 8 | 腾讯体育 | 345 | 3243903 | 43098 |
| 9 | 腾讯新闻 | 401 | 653387 | 14426 |
| 10 | 和陌生人说话-腾讯新闻 | 257 | 160746 | 621 |

#### 3.2.3.3 初步数据分析

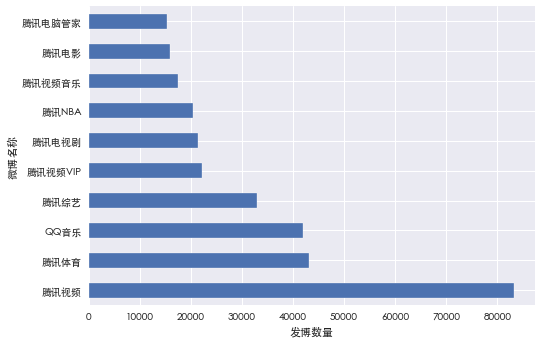
通过绘制各个腾讯机构号的条形图和密度图我们发现：

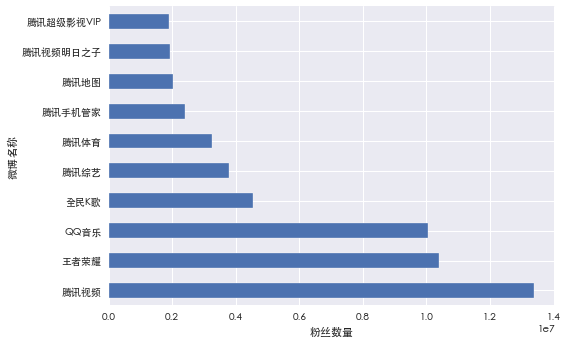
* 对于关注数量，大多数腾讯机构号的关注数量少于1000。
* 对于发博数量，有少数腾讯机构号的发博数量大于40000，我们可以将此类腾讯机构号用户视为非常活跃的腾讯机构号。
* 对于粉丝数量，有少数腾讯机构号的粉丝数量超过1000万，我们可以着重对这些

机构号进行分析。



紧接着我们找到关注数量和发博数量TOP10的腾讯机构号





我们发现腾讯视频为发博数量最多的腾讯机构号，同时腾讯视频也是粉丝数最多的腾讯机构号。

#### 3.2.3.4 数据分析中遇到的问题

主要问题是我们通过进入页面-F12-select an element in the page to inspect it定位我们所需的数据，但是微博中的一些信息被保存成JavaScript格式（比如某微博的点赞数），难以进行直接爬取，所以选择手动方式获取粉丝数量和发博数量均最多的腾讯机构号（腾讯视频）的一些数据。

在随后的数据提取部分我们使用Chen Lei(weibo-crawler,2020)的数据抓取工具。



图-数据分析中遇到的问题

# 4.数据提取和进一步数据分析

## 4.1我们需要哪些数据（针对腾讯视频官方号）

* 发博主体：腾讯视频
* 发博日期 ：发出微博对应日期
* 转发数 ：某微博被转发次数
* 评论数 ：某微博下评论数
* 点赞数 ：某微博被点赞次数
* 字数 ：某条微博字数
* 是否有视频：发出微博是否包含视频
* 是否有图片：发出微博是否包含图片
* 文字图片比： 字数/图片数

## 4.2 数据的初步分析

### 4.2.1对腾讯视频微博的点赞数，评论数以及转发数进行分析

4.2.1.2 点赞数

获取‘腾讯视频’微博的时间为2020-05-01 至 2020-05-14 2020-06-04 至 2020-06-11

我们首先分别获取点赞数，评论数以及转发数前十的微博，之所以分析这些微博，是因为这些微博具有很高的热度，分析其微博正文可以让我们揣测用户更想要看到什么，以及当腾讯官方号发布什么信息时，他们更乐于点赞转发以及评论。



图-点赞数前十的微博信息

通过数据我们发现点赞数超过30W的热点微博相关‘话题’为：极限挑战，创造营2020，创3主题曲MV，肖战手绘，鹅娘侃侃聊。

其中极限挑战，创造营2020，创3主题曲MV，肖战等话题本身都具有很高的热度，获取到高关注度不足为奇，而鹅娘侃侃聊的话题获得了超过30W的点赞数，我们需要对此微博的内容进行进一步的分析。此条微博的内容为：

#鹅娘侃侃聊#明晚8点：你会在腾讯视频看\_\_\_\_\_\_。发博日期为2020-05-01，而通过进一步的了解，2020-05-02晚上8点是创3首播的日子。

鹅娘侃侃聊话题的另外一条微博也进入了点赞数TOP10，微博内容为：母亲节快到了，有没有你印象深刻的关于母亲的影视剧，这是一个热点节日+提问的微博，而影视剧又是一个大家熟悉的话题。

对鹅娘侃侃聊话题的建议：

* 结合腾讯热点剧，热点综艺与网友进行互动，提前一日预热为最佳
* 结合热点节日进行提问，例如：
* 春节：结合热点明星/热点春节节目提问
* 元宵节：元宵节你们那里叫汤圆还是元宵
* 高考：有没有印象深刻的校园剧

4.2.1.2 评论数和转发数

评论数和转发数最高的微博都是和肖战手绘相关的一条微博，由此可见当腾讯视频发送和热点明星相关的微博时候可以获得更高的关注度。

建议：腾讯系官方号需要合理蹭热度

# 5.拟合模型

我们需要解决的问题是如何让官方号获得更高的关注，也就是如何让每一条微博获取更多的点赞数，转发数以及评论数。在4.进一步数据分析中，我们发现点赞数越高的微博，转发数和评论数也会更高，因此我们模型的响应变量设定为点赞数。

response：点赞数 like

explanatory variables（解释变量）：

* 字数 words ：某条微博字数
* 是否有视频 video：发出微博是否包含视频
* 是否有图片 picture：发出微博是否包含图片
* 工具 tools：发博方式（android发送/携带超话发送/客户端发送/其他）
* @用户数 at\_num：发出微博时候@的人数
* 是否原创 original：是否为原创微博

表-解释变量总结表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Variable** | **Mean** | **SD** | **Min.** | **1st Q.** | **Median** | **3rd Q.** | **Max.** |
| like | 8307.55 | 26723.81 | 23 | 520.0 | 2119 | 5641.5 | 433606 |
| words | 104.86 | 56.20 | 20 | 68.0 | 96 | 130.0 | 747 |
| at\_num | 1.99 | 2.39 | 0 | 1.0 | 1 | 2.0 | 18 |
| comment | 511.58 | 1004.41 | 2 | 90.0 | 238 | 606.5 | 20954 |
| forward | 796.92 | 2039.63 | 0 | 55.5 | 174 | 745.0 | 32971 |
| picture | 0.41 | 0.49 | 0 | 0.0 | 0 | 1.0 | 1 |
| video | 0.59 | 0.49 | 0 | 0.0 | 1 | 1.0 | 1 |

在表中我们可以发现腾讯视频所发微博中，最大点赞数为433606，平均点赞数8307.55，平均字数104.86，平均每条微博@1.99人，平均每条微博下511.58条评论，796.92转发，相比于发图片微博更喜欢发视频微博。

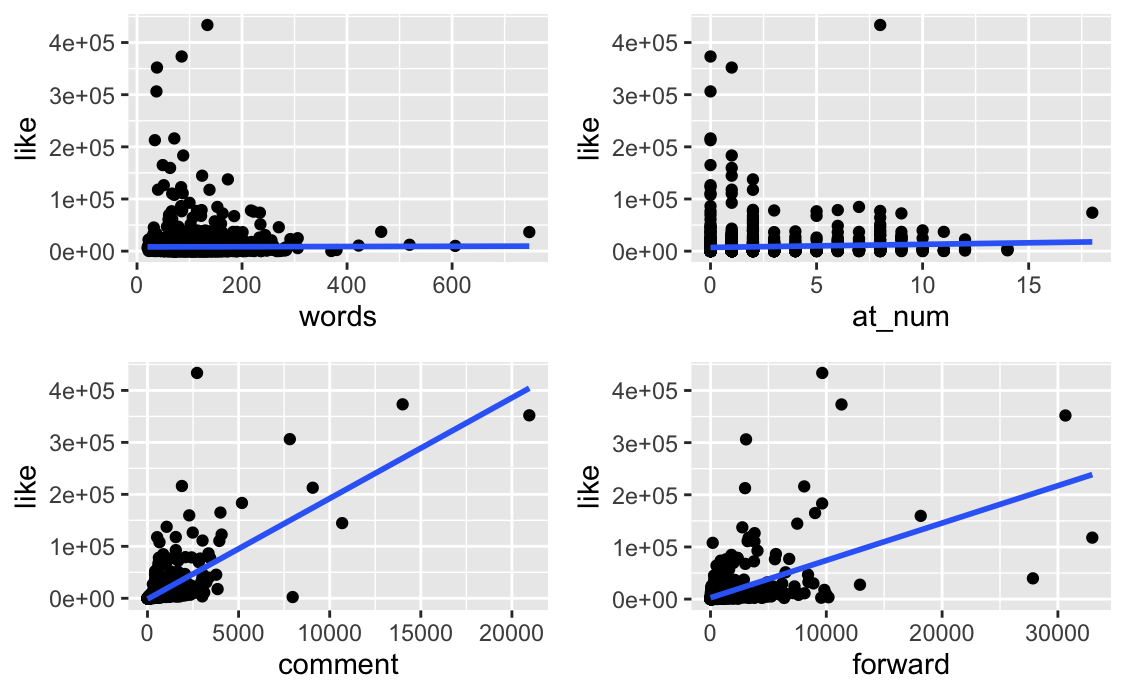


图-解释变量与点赞数的关系

从图中我们可以看出评论和转发数与点赞数有很强的线性关系，二发博文字数和@人数与点赞数的关系并不明显，不过在随后的分析中，由于点赞数的数字量级过大我们对它使用log-transform。

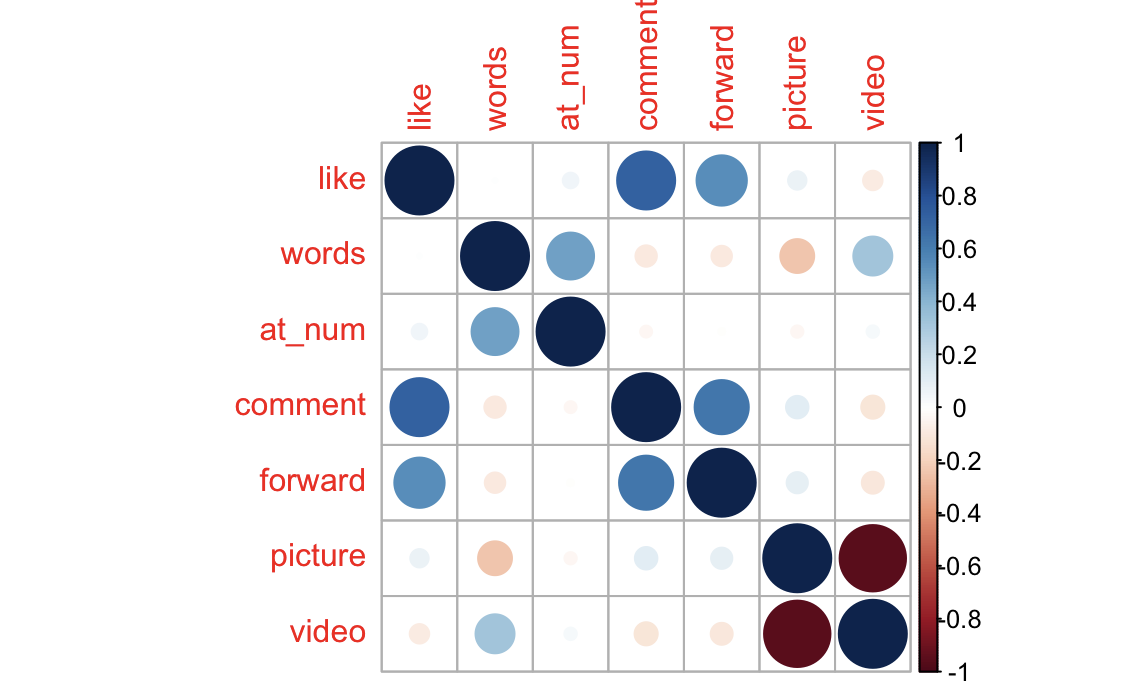


图-变量间correlation

从图中我们可以看到变量间的相关性，评论转发和点赞数呈现强相关，不过我们研究的问题是如何获取更多的点赞数，这种强相关会对我们的模型产生影响，在随后的分析中，剔除评论数和转发数。

通过调试我们最终拟合的线性模型如下：

log(like) ~ log(words) + at\_num + video + tools +picture+original

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **term** | **estimate** | **std\_error** | **statistic** | **p\_value** | **lower\_ci** | **upper\_ci** |
| intercept | 7.990 | 0.697 | 11.467 | 0.000 | 6.623 | 9.357 |
| log(words) | -0.284 | 0.120 | -2.363 | 0.018 | -0.519 | -0.048 |
| at\_num | 0.151 | 0.022 | 6.910 | 0.000 | 0.108 | 0.194 |
| video | -1.226 | 0.390 | -3.140 | 0.002 | -1.992 | -0.460 |
| toolsothers | 0.718 | 0.290 | 2.480 | 0.013 | 0.150 | 1.286 |
| toolsweibo.com | -0.197 | 0.151 | -1.304 | 0.192 | -0.494 | 0.099 |
| tools超话 | 0.880 | 0.179 | 4.922 | 0.000 | 0.529 | 1.231 |
| picture | -0.803 | 0.379 | -2.117 | 0.034 | -1.547 | -0.059 |
| originalTrue | 1.529 | 0.619 | 2.471 | 0.014 | 0.315 | 2.742 |

log(like) =7.990+ (-0.284)\* log(words) + 0.151\*at\_num -1.226\*video + tools +(-0.803)\*picture+1.529\*original

分析得到的线性模型，我们得到如下建议

建议：如果想获得更多的点赞数，字数要得到控制，尽量根据话题尽可能的多@明星，少发视频和图片，多使用超话工具，保证原创。

# 6.词频分析

根据爬虫爬到的微博正文信息，先把所有爬取到的微博内容做成txt文本，然后先去掉stopwords，再去掉空格之类的没有意义的词，之后使用value\_counts()函数得到词汇频数，生成一个词频表，词频TOP排序如下表。

表-腾讯视频词频TOP10

|  |  |
| --- | --- |
| 创造 | 887 |
| 2020 | 831 |
| 视频 | 688 |
| 腾讯 | 589 |
| 少女 | 252 |
| 超话 | 236 |
| R1SE | 235 |
| 教练 | 172 |
| 学员 | 168 |
| 舞台 | 151 |

生成的词云图：



从词云图中我们可以发现时下腾讯视频最热门的话题是火箭少女101（2020），此外极限挑战，风味人间，炙热的我们都有很高的热度。和腾讯视频合作的热点明星有黄子韬，R1SE以及宋茜等。

# 7.个人总结（个人进步及存在的问题）

# 通过本次项目系统学习了爬虫理论，但对于scrapy库的理解还是有些欠缺，对于python面向对象的知识有了一定的了解，复习了一下python可视化分析以及R语言进行简单线性模型的分析。

# 8.参考文献

* <https://www.bilibili.com/video/BV1qt411V7Aq?p=1> 解决登陆问题
* <https://github.com/dataabc/weibo-crawler#%E5%A6%82%E4%BD%95%E8%8E%B7%E5%8F%96cookie%E5%8F%AF%E9%80%89> 数据抓取工具
* <https://blog.csdn.net/Crystal_LYP/article/details/97634028> cmd命令运行python
* <https://www.bilibili.com/video/BV1ex411x7Em?from=search&seid=11097773534728523641> python面向对象
* <https://study.163.com/course/courseLearn.htm?courseId=1209598955&share=1&shareId=1410358713#/learn/video?lessonId=1280056350&courseId=1209598955> python数据分析实战 账号[datadata777@163.com](mailto:datadata777@163.com) 密码
* <http://www.icourse163.org/course/BIT-1001870001#/info> 北理工爬虫课程