# 前言



对于《大秦赋》，相信大家都不会陌生，由延艺导演，张鲁一主演的历史古装剧。

剧集讲述了秦始皇嬴政在吕不韦、李斯、王翦等的辅佐下平灭六国、一统天下，建立起中华历史上第一个大一统的中央集权国家的故事。  
 战国晚期，纷乱五百余年的华夏大地仍战火不息、 生灵涂炭。彼时六国势弱、秦国独强，天下统一之势渐显。巨商吕不韦携时在赵国为质的嬴异人逃归秦国，幼小的始 皇帝嬴政被弃留邯郸，屡遭生死劫难，也目睹战争带给百姓的痛苦与绝望，心中天下凝一之志由此而生。此后嬴政归秦，在咸阳政治漩涡中经历精神阵痛，蜕变成一个真正王者。为抓住一统天下的时机和力量，实现一统天下的理想抱负，精心谋划，暗中行动，终于平定嫪毐之乱，亲政，罢相，收复王权。此后又铲除宗室复辟势力，为东出灭国扫清障碍。在李斯、王翦、蒙恬等一班文臣武将的辅佐下扫灭六国，建立起中华历史上第一个大一统的中央集权国家。

当然，我今天的主题自然不是讲历史故事，而是爬虫。

现在开始进入主题......

# 需求分析

《大秦赋》这部剧在最近还算是比较热的一门电视剧了。作为喜欢写爬虫的人来说，获取观众的评论，做数据分析是再好不过了，所以今天就来爬取豆瓣中观众的短评。

## 评论数随时间的变化趋势

一般来说，一部剧在开播后的几天都是比较火的，观众的评论自然就不会少。但是当这部剧开播之后却没有你想象的那般美好，那观众就不会有看下去的欲望，既然观众都不看，那你觉得还会有评论吗？

## 24小时内的评论数的趋势

这里主要是分析观众观看这部剧的时间大概在什么范围内。一般看完就评论或者边看边评论，时间上就不会有太大的差别，当然这个也是大多数人的做法，应该会有极少的人在几天后才对前面的剧情进行评论的吧。那样谁还去翻前面的东西。

## 观众对该剧的评价

就像在淘宝买东西一样，如果你买的产品性价比特别高，很实惠。那我相信你不会吝啬那5星好评，当然如果你觉得我的文章好，也不会吝啬你的点赞。

## 观众的主要评论

在这里主要是看看观众对哪类人或者是哪些事情提及的比较多，方便之后的进一步分析。

# 数据获取的实现

巧妇难为无米之炊。

同样的，数据是做数据分析的基础，没有数据你做个嘚。

## 获取什么数据

从上面的需求分析可以得出我们现在需要获取三项数据：'星级'、'评论'、'评论时间'



## 网页分析

打开开发者工具，利用选择器可以快速定位数据的位置。



下面给出核心代码获取评论、星级与时间。

def get\_info(self):
  
 html = etree.HTML(self.login())
  
 time = html.xpath('//div[@class="comment"]/h3/span/span[3]/@title')
  
 star = html.xpath('//div[@class="comment"]/h3/span/span[2]/@title')
  
 content = html.xpath('//p[@class=" comment-content"]/span/text()')
  
 content = [i.replace('\n', '') for i in content]
  
 df = pd.DataFrame(
  
 {'content\_time': time,
  
 'star': star,
  
 'comment-content': content
  
 }
  
 )
  
 return df

## 关于反爬

经过测试这边的反爬只有User-Agent和cookies，只要将这两个信息添加进headers即可。

所以对于这种反爬手段还是比较容易解决的。因此这里可以稍微转变一下写法，使用session来保存cookies，核心代码如下所示：

def login(self):
  
 data = {
  
 'ck': '',
  
 'remember': 'true',
  
 'name': '18218138350',
  
 'password': '698350As?',
  
 }
  
 self.session.post(self.login\_url, data=data)

## 翻页爬取

通过网页查看短评，你会发现评论不单单只有1页，而是有好几页。但是通过运行上面的代码你会发现，只能获取到第一页的数据。

# 第一页
  
https://movie.douban.com/subject/26413293/comments?start=0&limit=20&status=P&sort=new\_score
  
# 第二页
  
https://movie.douban.com/subject/26413293/comments?start=20&limit=20&status=P&sort=new\_score
  
# 第三页
  
https://movie.douban.com/subject/26413293/comments?start=40&limit=20&status=P&sort=new\_score

通过观察上面的链接你会发现，只有start参数是不一样的，并且是20的倍数。只要规律的改变start参数的值便可以实现翻页的效果。

但是在这里有一个问题，当start=480时，便无法继续翻页了，所有这个也是豆瓣反爬的手段之一，有兴趣的高手可以自己试试。

核心代码如下所示：

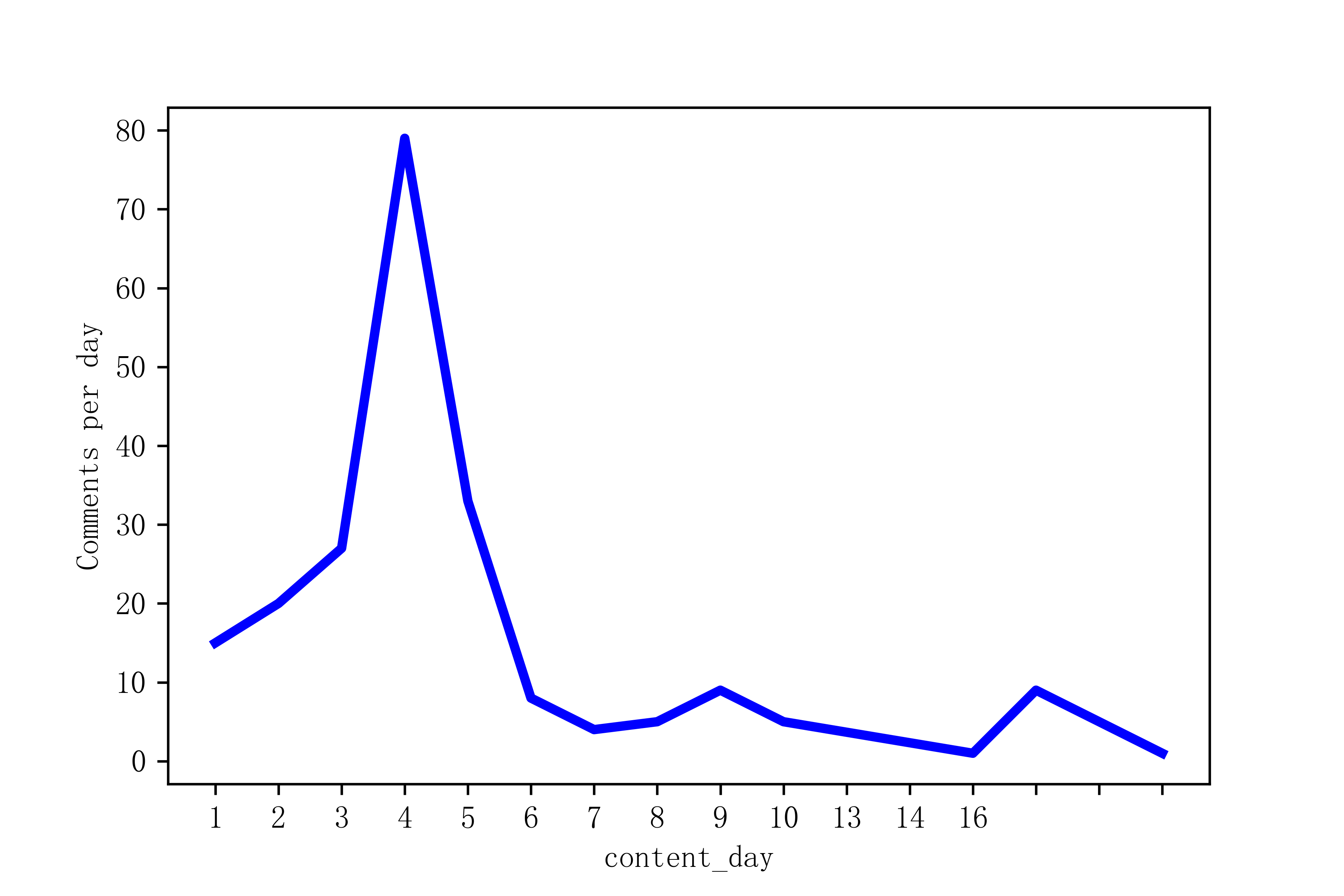
def get\_content\_url(self, i):
  
 url = f'https://movie.douban.com/subject/26413293/comments?start={i}&limit=20&status=P&sort=new\_score'
  
 return url
  
   
 if \_\_name\_\_ == '\_\_main\_\_':
  
 douban = Douban()
  
 douban.login()
  
 df = pd.DataFrame(columns=['content\_time', 'star', 'comment-content'])
  
 for i in range(25):
  
 print(f'正在打印第{i+1}页')
  
 url = douban.get\_content\_url(i\*20)
  
 df1 = douban.get\_info(url)
  
 df = pd.concat([df, df1])
  
 time.sleep(3)
  
 df = df.reset\_index(drop=True)
  
 df.to\_csv('../data/conment-content\_all.csv', encoding='utf-8-sig')
  
 print('获取成功')

通过对上面代码的测试，已经将数据全部获取下来了。

接下来就是进入数据分析的精彩时刻了。

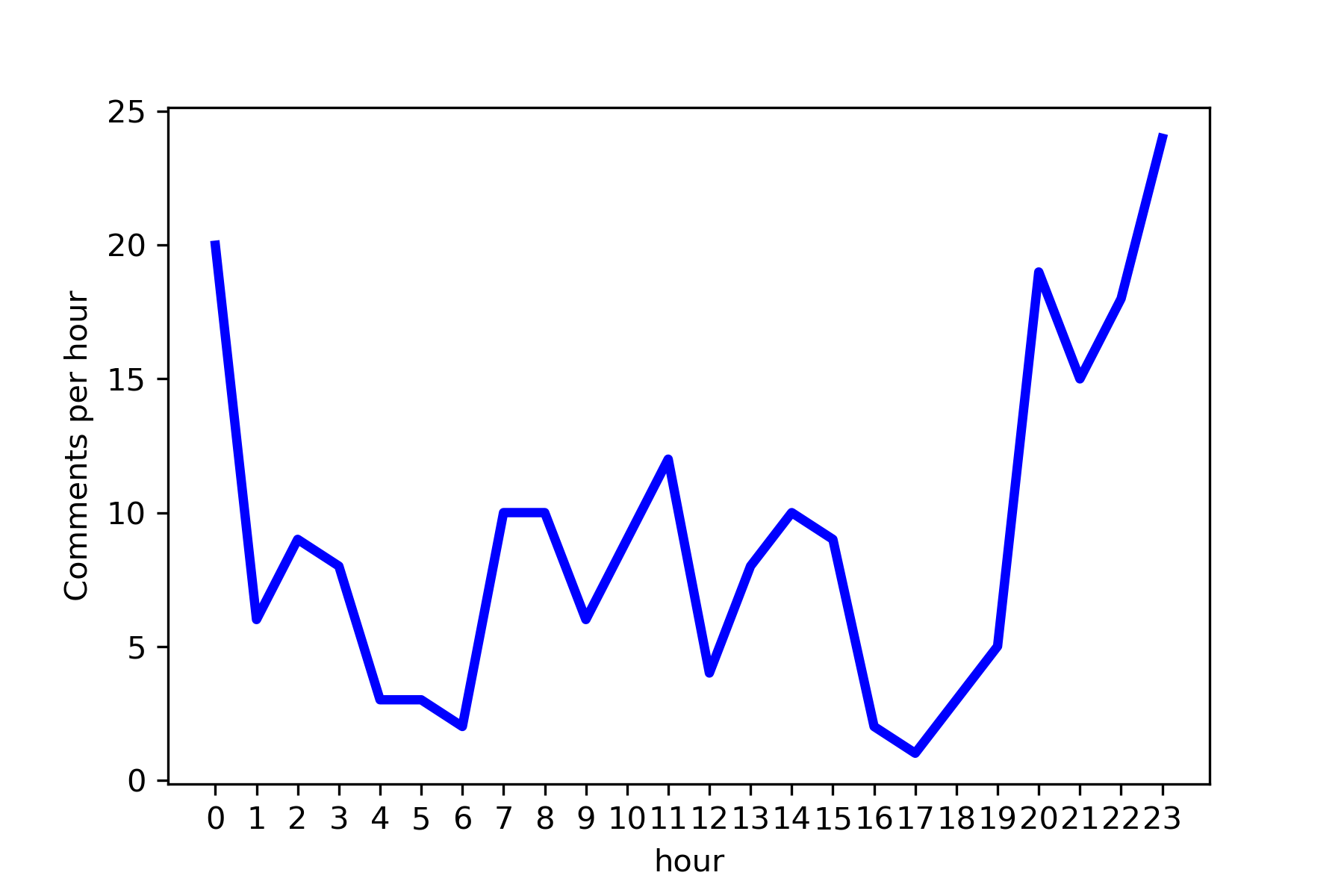
# 数据分析实现

## 评论数随时间的变化



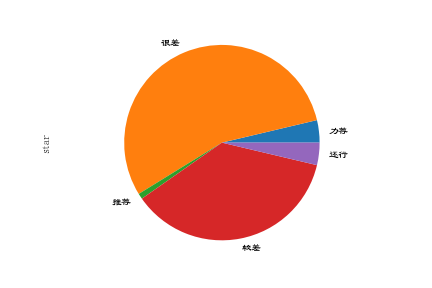
从上图，你会发现从该剧开播以来的前5天，评论数量都还是比较多的，但是到了第6天评论数量就猛跌，从这一方面也说明该剧的受欢迎程度不高。

## 24小时内评论数的变化



分析24小时内评论数的变化情况可以发现，观众在19点至24点都是观看该剧的高峰。19点就可以看剧，想想我也是很羡慕的呀。

## 评分情况

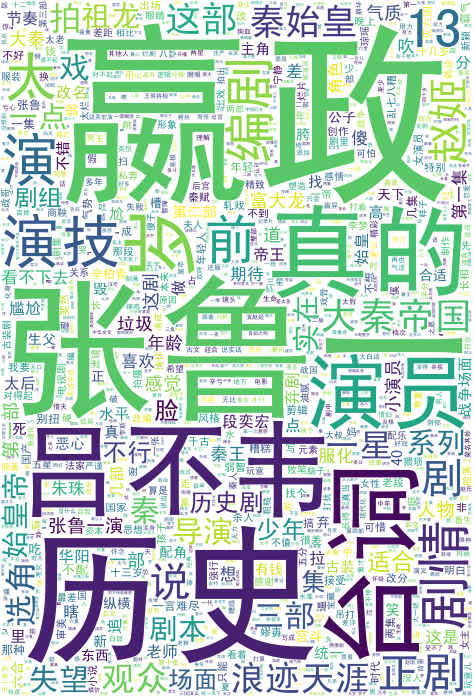


从图上可以看到，评分情况惨不忍睹。

大秦赋自开播以来口碑一路暴跌，从9.3到8.5到7.6再到如今的6.1，上映之前的万众期待似乎已经是远古年间的故事了。

从个人角度来说，我不喜欢大秦赋。被大秦帝国三部曲一点一点把口味养叼了的老观众对大秦赋有失落感是很正常的——为了讨好年轻群体将台词直白化、为了吸引更多受众加入了大量的“宫斗情感戏”、为了将嬴政的塑造走出一条前人没有走过的路，在剧情和人设上大下功夫；这些为了迎合市场而试图转型的举动在原有受众群体里等同于背叛。尤其是对于我这种从孙皓晖老师原著转至影视剧的拥趸来说，这种不满显得尤为有理有据。

## 评论内容



# 最后

本次分享到这里就结束了，如果你看到了这里，那么说明本文对你还是有帮助的，所有我也非常希望读者朋友们可以给我点赞、评论与转发，在下不胜感激。

**路漫漫其修远兮，吾将上下而求索。**

我是**啃书君**，一个专注于学习的人，**你懂的越多，你不懂的越多。**

更多精彩内容，我们下期再见！