## 准备工作

目标: 抓取豆瓣电影Top250 电影信息

1.引入常用的第三方包stringr 和 rvest。

```
install.packages("rvest") # 处理HTML
install.packages("stringr") # 处理字符串
library(rvest)
library(stringr)
```

- 2.测试XPath 并获取每个电影的具体信息
  - 获取电影名称

```
web=read_html("https://movie.douban.com/top250?start=0&filter=") # 读取页面 web %>% html_nodes(xpath="//span[@class='title'][1]") %>% html_text()
```

#### • 结果如下:

[1] "肖申克的救赎" 太冷"	"霸王别姬"	"阿甘正传"	"这个杀手不
[5] "泰坦尼克号" 单"	"美丽人生"	"千与千寻"	"辛德勒的名
[9] "盗梦空间" [13] "海上钢琴师" 天"	"忠犬八公的故事" "三傻大闹宝莱坞"	"楚门的世界" "机器人总动员"	"星际穿越" "放牛班的春
[17] "无间道" [21] "教父" [25] "怦然心动"	"大话西游之大圣娶亲" "当幸福来敲门"	"疯狂动物城" "龙猫"	"熔炉" "控方证人"

• 获取某电影其他元素的XPath如下:

```
# 获取 导演,主演,年份,国家,电影分类
web %>% html_nodes(xpath="//div[@class='info']//div[@class='bd']//p[@class]
[1]") %>% html_text()
# 获取 电影评分
web %>%
html_nodes(xpath="//div[@class='info']//div[@class='bd']//div//span[@class='r ating_num']") %>% html_text()
# 获取 电影评价人数
web %>%
html_nodes(xpath="//div[@class='info']//div[@class='bd']//div//span[last()]")
%>% html_text()
# 获取 电影详情链接
web %>% html_nodes(xpath="//div[@class='info']//div[@class='hd']//a[1]") %>%
html_attr('href')
```

### 开始抓取数据

```
if (!require("stringr"))
 install.packages("stringr")
if (!require("rvest"))
  install.packages("rvest")
df_douban_250 = data.frame()
for(page number in 0:9){
  content = read_html(str_c(
    "https://movie.douban.com/top250?start=", 25*page_number, "&filter="))
 middle_text = content %>% html_nodes(
   xpath="//div[@class='info']//div[@class='bd']//p[@class][1]") %>%
html text()
  middle_text_line_1 = sapply(str_split(str_trim(middle_text), '\n'), "[[",
1) #[1] "导演: 弗兰克·德拉邦特 Frank Darabont 主演: 蒂姆·罗宾斯 Tim Robbins
/..."
 middle_text_line_2 = sapply(str_split(str_trim(middle_text), '\n'), "[[",
2) #[1] "1994 / 美国 / 犯罪 剧情"
 m_director = sapply(str_split(middle_text_line_1, "[\u00A0]{3}"), "[[", 1)
# 注意:  字符需要用unicode去匹配
 m_actor = str_split(middle_text_line_1, "[\u00A0]{3}")
 for(i in 1:length(m_actor)){
   if(length(m_actor[[i]])<2){m_actor[[i]][2]=''} # 补齐操作
  }
```

```
m_actor = sapply(m_actor, "[[", 2)
 list year = str split(middle text line 2, "/")
 for(i in 1:length(list_year)){
   if(length(list year[[i]])>3){ #处理No.054的异常数据
     list_year[[i]][3] = list_year[[i]][length(list_year[[i]])]
     list_year[[i]][2] = list_year[[i]][length(list_year[[i]])-1]
   }
  }
 m_year = str_trim(sapply(list_year, "[[", 1))
 m country = str trim(sapply(list year, "[[", 2))
 m_type = str_trim(sapply(list_year, "[[", 3))
 m_name = content %>% html_nodes(
   xpath="//span[@class='title'][1]") %>% html_text()
 m_score = content %>% html_nodes(
xpath="//div[@class='info']//div[@class='bd']//div//span[@class='rating_num']
") %>% html_text()
 m commit count = content %>% html nodes(
   xpath="//div[@class='info']//div[@class='bd']//div//span[last()]") %>%
html_text()
 m_commit_count = str_sub(m_commit_count, end=-4)
 m detail url = content %>% html nodes(
   xpath="//div[@class='info']//div[@class='hd']//a[1]") %>%
html attr('href')
  df_one_page = data.frame(
   m_name, # 电影名称
   m_director, # 导演
   m_actor, # 主演
   m_year, # 上映年份
   m_country, # 国家
   m_type, # 所属分类
   m_score, # 评分
   m_commit_count, # 评论人数
   m_detail_url # 详细页面
  df_douban_250 = rbind(df_douban_250, df_one_page)
}
```

## 整理数据

#### 异常数据处理:

● No.054《大闹天宫》年份数据异常 1961(中国大陆) / 1964(中国大陆) / 1978(中国大陆) / 中国大陆 / 剧情 动画 奇幻 古装

需要从右向左按 / 字符split 3个字符串,才能符合其他的电影信息的格式。

由于stringr文档中的str\_split不支持从右向左,因此仅取了第一个年份,丢掉了第二和第三个。

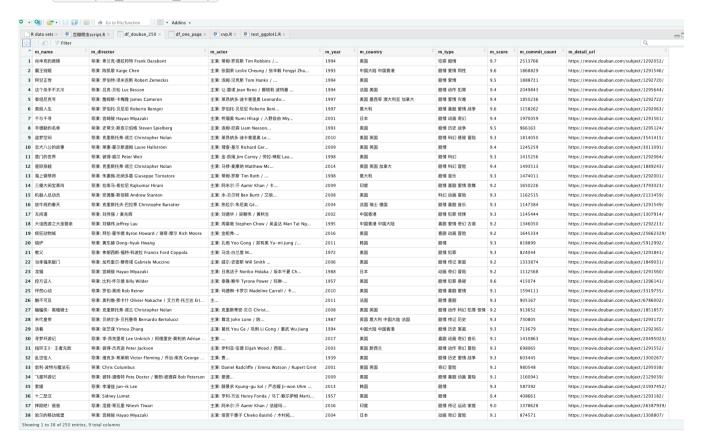
#### 需要补充的数据:

- No.063《窃听风云》 没有主演数据 导演: 弗洛里安·亨克尔·冯·多纳斯马尔克 Florian Henckel von Donnersmarck &n...
- No.235《黑客帝国2》没有主演数据导演: 拉娜·沃卓斯基 Lana Wachowski / 莉莉·沃卓斯基 Lilly Wachowski ...
- No.207《初恋这件小事》 没有主演数据 导演: 普特鹏·普罗萨卡·那·萨克那卡林 Puttipong Promsaka Na Sakolnakorn / 华森·波克彭...

还有一些电影的主演在排行榜页面上没有、需要进一步完善。

#### 查看抓取的数据:

使用 View(df\_douban\_250) 命令查看如下



#### 导出抓取的数据:

使用 write.save(df\_douban\_250, 'douban250.csv') 导出CSV文件。

# 参考列表