实战案例2：YouTube视频趋势分析

作者： Robin
日期： 2018/02
提问： 小象问答
数据集来源： kaggle

1. 案例描述

YouTube（世界著名的视频分享网站）列出了该平台上热门视频的最新视频。 据Variety杂志称，“为了确定年度热门视频，YouTube网站使用了多种因素的组合，其中包括了衡量用户交互的因素（观看次数，分享次数，评论和喜好）。 注意，这里的热门视频并不是整年观看次数最多的视频”。

2. 数据集描述

- Kaggle提供的数据集包括4个国家的热门YouTube视频的每日记录。每个国家的数据文件为一个CSV文件及一个JSON文件：
  - CAvideos.csv
  - CA_category_id.json
  - DEvideos.csv
  - DE_category_id.json
  - GBvideos.csv
  - GB_category_id.json
  - USvideos.csv
  - US_category_id.json
- 数据字典
  - CSV文件:
    - **video_id**: 视频id，字符串
    - **trending_date**: 视频上榜的日期，字符串
    - **title**: 视频标题，字符串
    - **channel_title**: 所属频道标题，字符串
    - **category_id**: 所属类别编号，整型
    - **publish_time**: 视频发布时间，时间类型
    - **tags**: 视频标签，字符串
    - **views**: 观看次数，整型
    - **likes**: 点赞次数，整型
    - **dislikes**: 被踩次数，整型
    - **comment_count**: 评论次数，整型
    - **comments_disabled**: 评论是否关闭，布尔值
    - **ratings_disabled**: 打分是否关闭，布尔值
    - **video_error_or_removed**: 视频出错或者被删，布尔值
    - **description**: 视频详情，字符串

- JSON文件
  - JSON文件中包含category_id用于和CSV文件中的category_id做匹配，从而获取category的标题。

3. 任务描述

- 绘制每个国家指定列的的top10，如category，channel_title等
- 统计视频发布后上榜的天数
- 查看views,likes,dislikes,comment_count的关系

4. 主要代码解释

- 代码结构

```
lect03_proj
├── data
    ├── *.csv   # CSV数据文件
    ├── *.json  # JSON数据文件
├── output
    ├── *.csv   # 分析结果CSV文件
    ├── *.png   # 分析结果PNG文件
    ├── *.html  # 分析结果HTML文件
├── config.y    # 配置文件
├── main_win.py     # 适用于windows的主程序
├── main_mac.py     # 适用于mac的主程序
├── lect03_proj_readme.pdf  # 案例讲解文档
```

- **main_win.py**

解决matplotlib在Win系统显示中文问题，注意会对一些特殊语言的字符产生影响，比如德语等。

```
# 仅适用于Windows
plt.rcParams['font.sans-serif'] = ['SimHei']  # 指定默认字体
plt.rcParams['axes.unicode_minus'] = False  # 解决保存图像是负号'-'显示为方块的问题
```

- **main_mac.py**

解决matplotlib在Mac系统显示中文问题。

```
def get_chinese_font():
    """
        获取Mac系统中文字体
    """
    return FontProperties(fname='/System/Library/Fonts/PingFang.ttc')

def plot_top10_by_country(video_df, col_name, title, save_filename):
    ...
    # 使用fontproperties改变默认的字体
    fig.suptitle(title, fontproperties=get_chinese_font())
    ...
```

- **main_?.py**

Pandas中日期类型处理

```
def combine_video_data():
    ...
    # 根据时间的格式将数据处理成时间类型，用于后续的操作
    # 比如，计算时间差等。详细用法会在第八课中介绍
    video_df['trending_date'] = pd.to_datetime(video_df['trending_date'], format='%y.%d.%m')
    video_df['publish_time'] = pd.to_datetime(video_df['publish_time'], format='%Y-%m-
%dT%H:%M:%S.%fZ')
    ...
```

- **main_?.py**

DataFrame中的map操作

```
def combine_video_data():
    ...
    # 通过map操作添加category名称列
    # 注意，这里的map()不同于Python本身的map()函数
    video_df['category'] = video_df['category_id'].map(category_dict)
    ...
```

- **main_?.py**

过长的字符串处理

```
def plot_top10_by_country(video_df, col_name, save_filename):
    ...
    # 处理x轴的刻度标签，如果标签长度超过15个字符，用省略号代替
    x_labels = [label[:12] + '...' if len(label) > 15 else label for label in top10_df.index]
    ...
```

- **main_?.py**

EChart中同时绘制多个图表时，使用Overlap()进行合并

```
def plot_days_to_trend(video_df, save_filename):
    ...
    bar = Bar('视频发布后2个月的情况')
    ...
    line = Line()
    ...
    overlap = Overlap()
    overlap.add(bar)
    overlap.add(line)
    ...
```

- **main_?.py**

Pandas中相关系数的计算。两个变量之间的皮尔逊相关系数定义为两个变量之间的协方差和标准差的商：

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

皮尔逊相关系数的变化范围为-1到1。 系数的值为1意味着 X 和 Y 可以很好的由直线方程来描述，所有的数据点都很好的落在一条 直线上，且 Y 随着 X 的增加而增加。系数的值为−1意味着所有的数据点都落在直线上，且 Y 随着 X 的增加而减少。系数的值为0意味着两个变量之间没有线性关系。

```
def plot_relationship_of_cols(all_video_df, cols):
    ...
    # 计算每两个列之间的皮尔逊相关系数
    corr_df = sel_video_df.corr()
    ...
```

5. 案例总结

- 该项目通过分析YouTube数据巩固了Pandas数据操作及Python中常用的可视化工具:
    - Pandas数据合并
    - matplotlib数据可视化
    - Pandas数据可视化
    - Searborn数据可视化
    - pyecharts数据可视化

6. 课后练习

- 按月份统计不同国家发布的视频数量，并用柱状图进行可视化

参考资料

1. Pandas数据可视化
2. matplotlib可视化案例
3. seaborn可视化案例
4. pyecharts使用
5. 皮尔逊相关系数