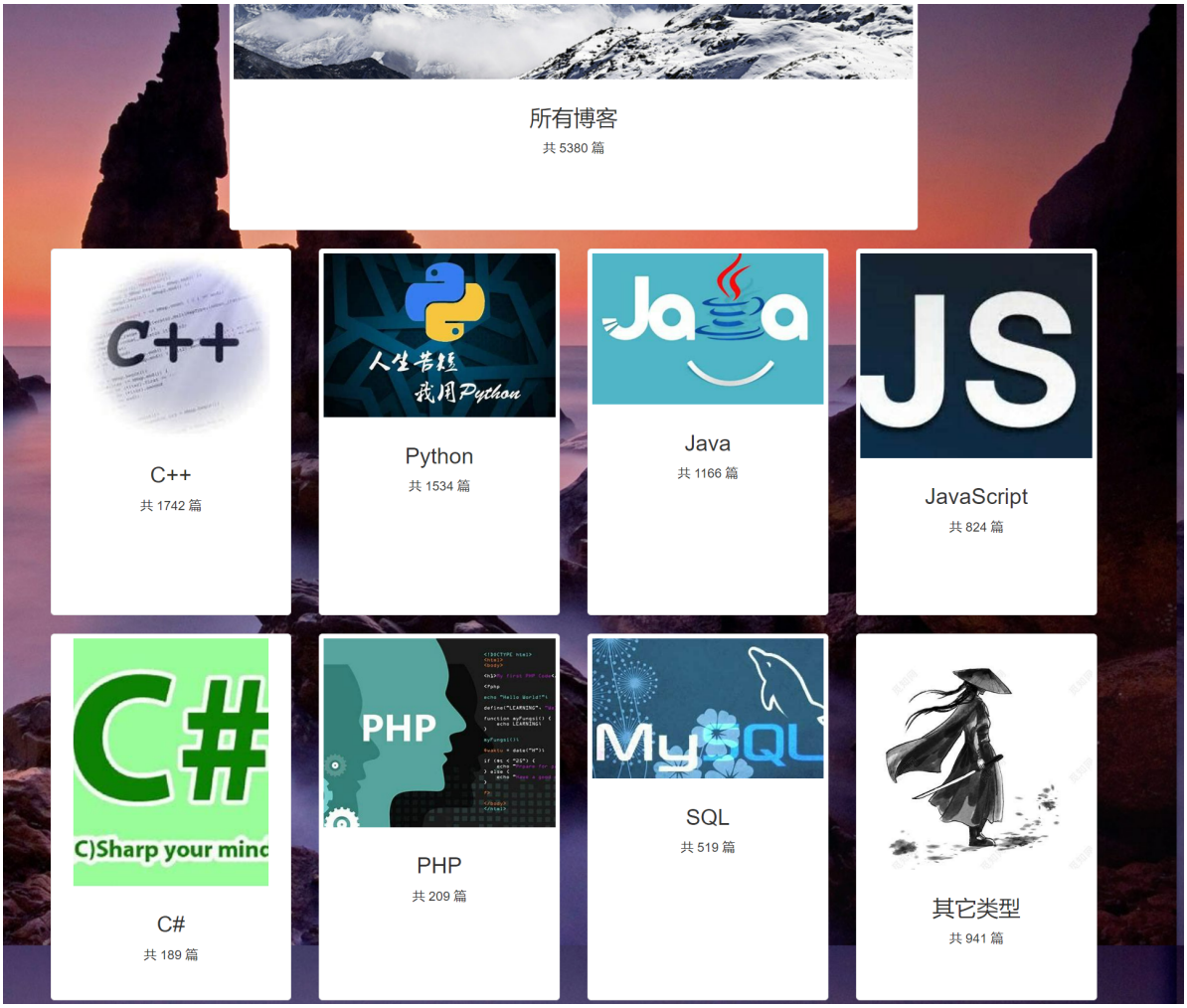
CSDN爬取数据分析

样本情况

- 从CSDN网站的语言分类平台上爬取共5385条博客，博客内容包括c++，c#等7种编程语言，其它数据归类到其它类型中。
- 主要爬取的数据包括：文章题目、正文、发表日期时间、点赞收藏量、作者头像、点赞量等



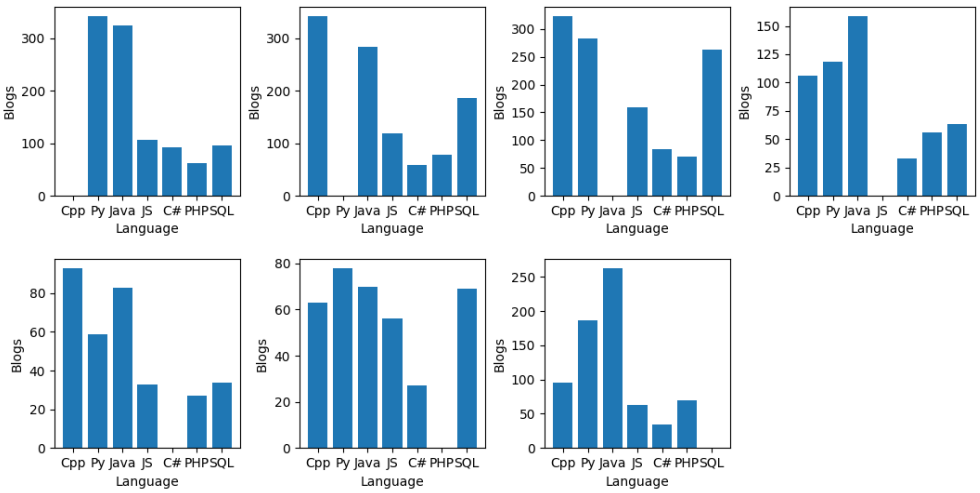
分析结论

话题一：主流语言

- 下面这段代码统计了在带有所有带有某种语言标签的数据种带有其它语言标签的数量，并据此绘制了柱状统计图。以图一为例，在所有带有`Cpp`标签的数据中，带有`Python`和`Java`标签的博客都达到了300条以上，而其它语言标签数量均在100条以下

```
import json
from scipy.integrate import quad
import numpy as np
import matplotlib.pyplot as plt
import dexplot as dxp
import pandas as pd
with open('all_data.json','r',encoding='utf-8') as JsonFile:
    data = json.load(JsonFile)
```

```
all_dict = {'Cpp':  {'Cpp': 0, 'Python': 0,'Java': 0 , 'JavaScript': 0, 'Csharp': 0, 'PHP': 0, 'SQL': 0}, 'Python':  {'Cpp': 0, 'Python': 0,'Java': 0 , 'JavaScript': 0, 'Csharp': 0, 'PHP': 0, 'SQL': 0},'Java':  {'Cpp': 0, 'Python': 0,'Java': 0 , 'JavaScript': 0, 'Csharp': 0, 'PHP': 0, 'SQL': 0} , 'JavaScript':  {'Cpp': 0, 'Python': 0,'Java': 0 , 'JavaScript': 0, 'Csharp': 0, 'PHP': 0, 'SQL': 0}, 'Csharp':  {'Cpp': 0, 'Python': 0,'Java': 0 , 'JavaScript': 0, 'Csharp': 0, 'PHP': 0, 'SQL': 0}, 'PHP':  {'Cpp': 0, 'Python': 0,'Java': 0 , 'JavaScript': 0, 'Csharp': 0, 'PHP': 0, 'SQL': 0}, 'SQL':  {'Cpp': 0, 'Python': 0,'Java': 0 , 'JavaScript': 0, 'Csharp': 0, 'PHP': 0, 'SQL': 0}}
for each in data:
    for lan in each['language'] :
        if lan =='Others':
            continue
        for lan_to in each['language']:
            if lan_to == 'Others' or lan == lan_to:
                continue
            all_dict[lan][lan_to] += 1
num=1
for each in all_dict:
    plt.subplot(2, 4,num )
    num+=1
    lst = list(all_dict[each].values())
    plt.bar(range(7),lst)
    plt.xticks(range(7),['Cpp','Py','Java','JS','C#','PHP','SQL'])
    plt.xlabel('Language')
    plt.ylabel('Blogs')
plt.show()
```



据此可以得出下面的结论：

- `Cpp`、`Java`、`JavaScript`**在七种语言中热度较高，覆盖面较广**，有两点数据支撑：
    1. 从标签总数来看，带有这三种语言标签的数量均在1000条以上，远大于其它语言
    2. 从语言标签的包含关系来看，无论是在哪种语言标签下，这三种语言均占据了数量前三的位置
- `Java`与`JavaScript`和`SQL`常常被放到一起讨论，说明这三种语言的联系相对更紧密

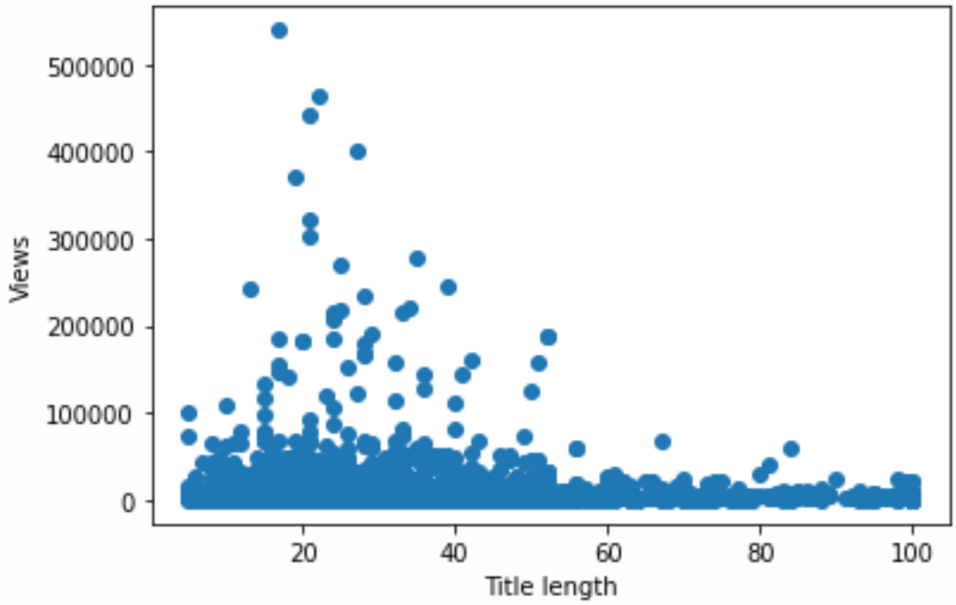
话题二：文章受欢迎程度的影响因素

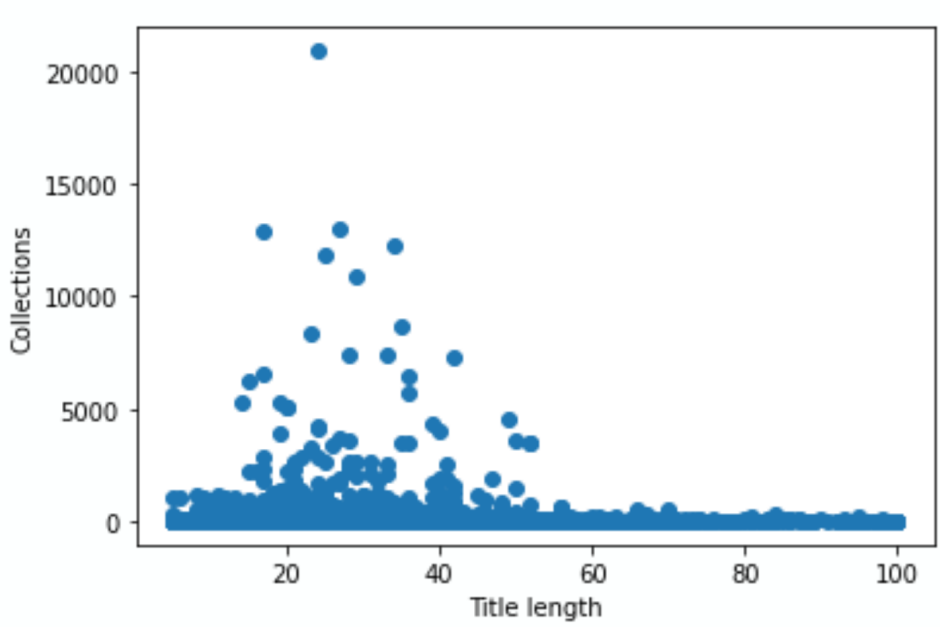
根据收集的数据，文章受欢迎程度主要可以由文章的点赞量、收藏量和阅读量来衡量

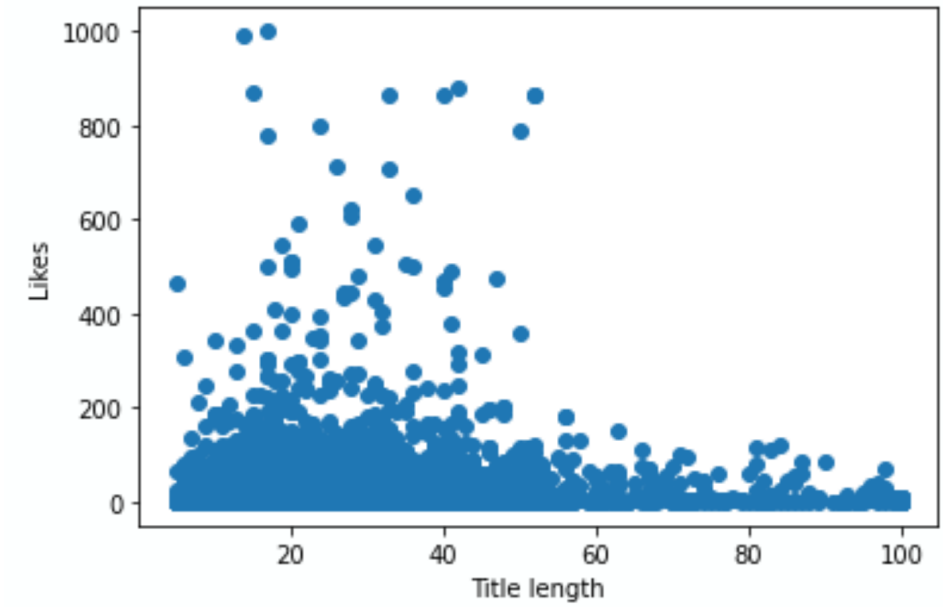
下段代码绘制了点赞量、收藏量、阅读量和标题长度的关系

```
import matplotlib.pyplot as plt
import json
import random
with open('all_data.json','r',encoding='utf-8') as JsonFile:
    data = json.load(JsonFile)
x=[]
y=[]
rec_index = []
num = 0
for each in data:
    x.append(len(each['Title']))
    y.append(int(each['Views'])) #还可以是Likes和Collections

plt.scatter(x,y)
plt.show()
```

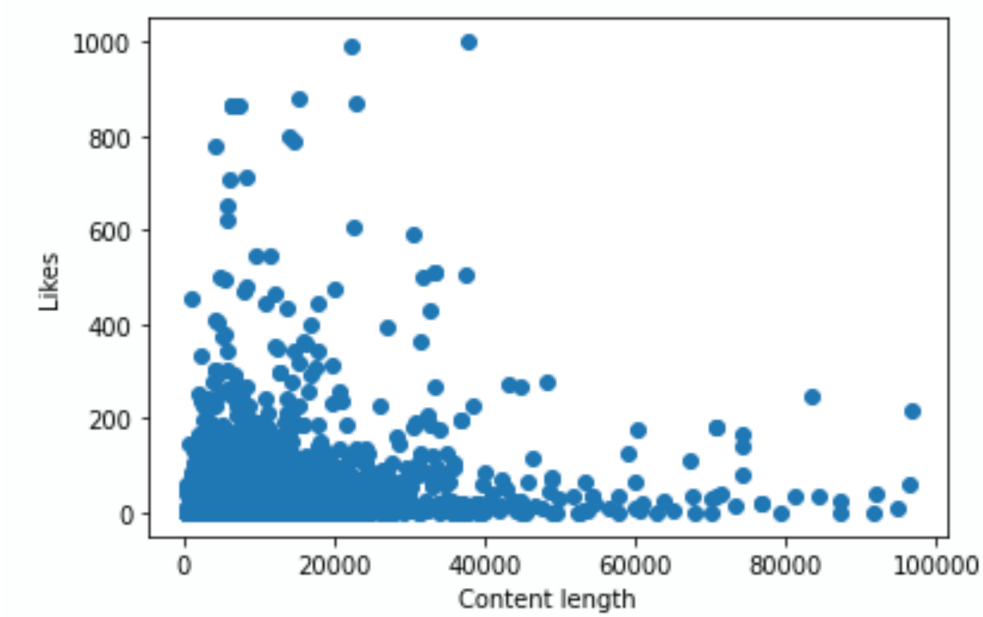


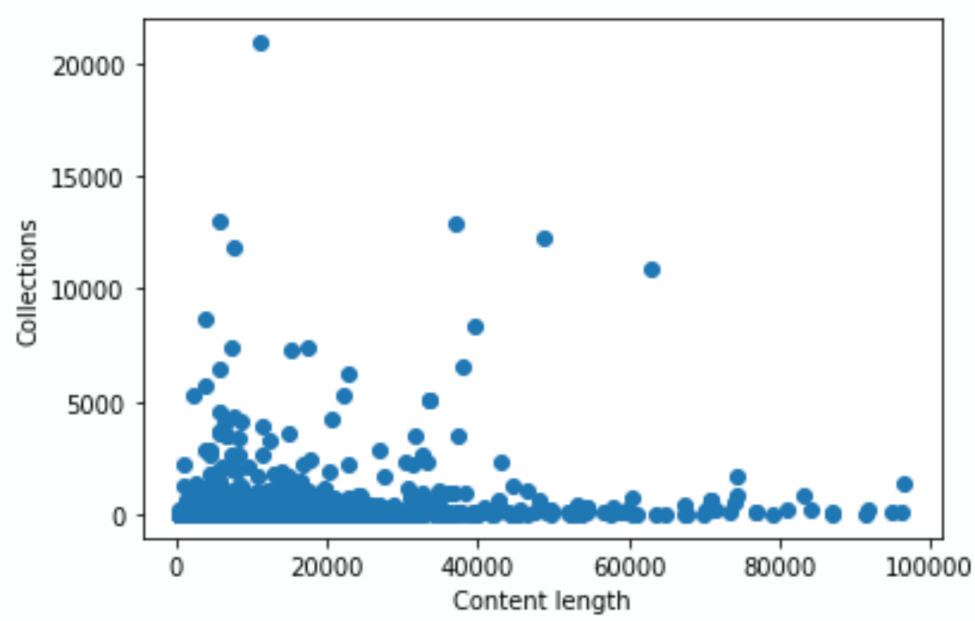


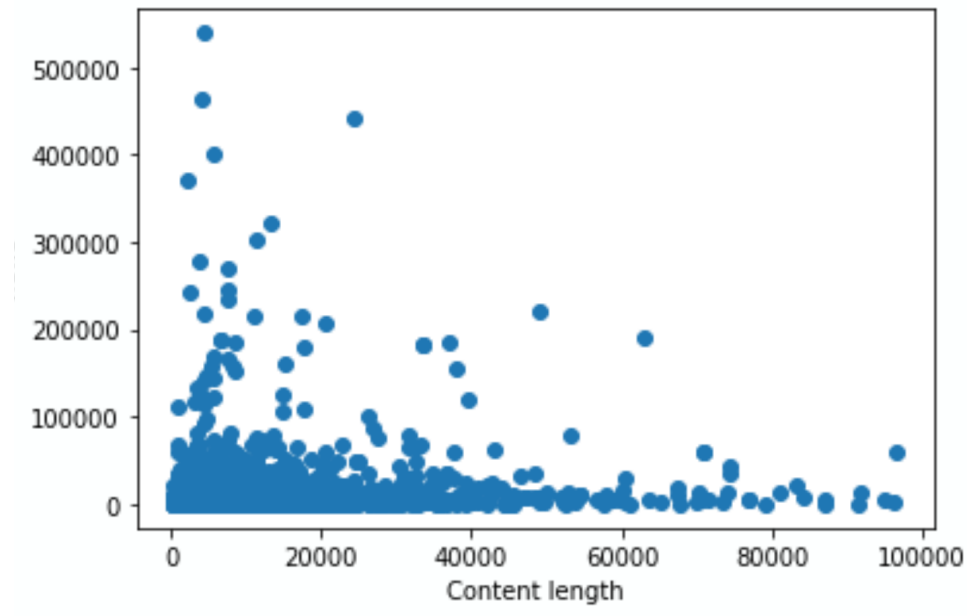


分析数据发现，标题长度在20到40个字符之间的文章更受欢迎，无论是在点赞量、收藏量还是阅读量上都明显更高，文章点赞量相比其它数据受标题长度影响更小一些。

采用相同的方法、进一步统计了点赞量、收藏量以及阅读量和正文长度的关系：







发现受欢迎的文章字符数大多在40000以内。

据此可以得出下面的结论：

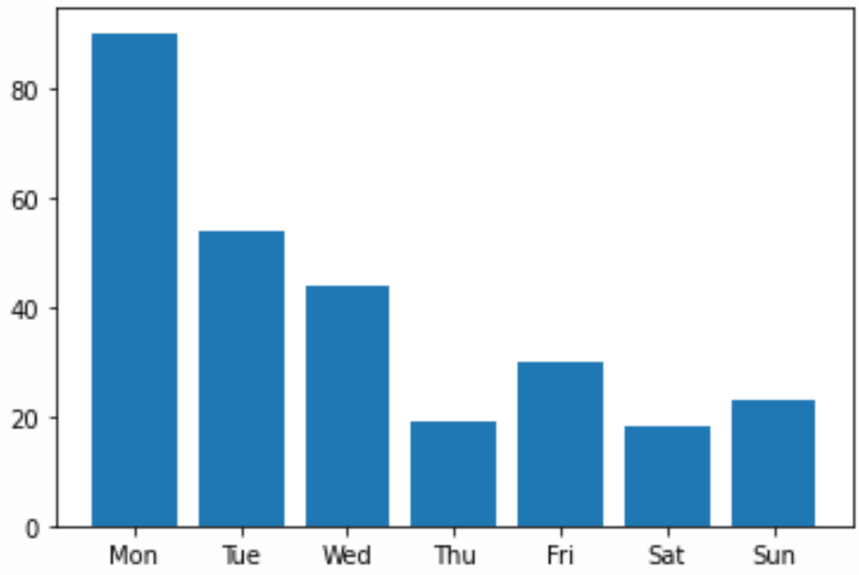
- **受欢迎的文章、标题和正文长度均适中（20-40字、40000以内）。总体来看，字数较少的文章更容易受到欢迎，点赞量相比收藏和观看受文章与标题长度影响比较小**

话题三：基于作者发布习惯的分析

通过将样本按照作者归类、可以观察到不同作者发表文章时间的特点，下面代码统计了每个作者发表文章数量最多的一天。

```
import matplotlib.pyplot as plt
import json
import random
import re
import datetime
with open('all_data.json','r',encoding='utf-8') as JsonFile:
    data = json.load(JsonFile)
author_dict={}
for each in data:
    if each['Author'] in author_dict:
        author_dict[each['Author']]+=1
    else:
        author_dict.update({each['Author']:1})
author_list=sorted(author_dict.items(), key = lambda d:d[1],reverse=True)
hot_author_list=[]
for each in author_list:
    if each[1]>=3:
        hot_author_list.append(each)
hot_author_dict = {}
for aut in hot_author_list:
    aut_time_list=[]
    for each in data:
        if each['Author']==aut[0]:
            aut_time_list.append(each['Date'])
    aut_time_list.sort(reverse=True)
    hot_author_dict.update({aut[0]:aut_time_list})
author_week_dict={}
for aut in hot_author_dict:
    week_dict = [0,0,0,0,0,0,0]
    for each in hot_author_dict[aut]:
        day = datetime.date(int(re.match(r'(\d+)-(\d+)-(\d+)',
each).group(1)),int(re.match(r'(\d+)-(\d+)-(\d+)',
each).group(2)),int(re.match(r'(\d+)-(\d+)-(\d+)',
each).group(3))).weekday()
        week_dict[day]+=1
        author_week_dict.update({aut:week_dict})
author_day = [0,0,0,0,0,0,0]
for each in author_week_dict:
    max = 0
    da = 0
    for x in author_week_dict[each]:
        if x>max:
            max=x
            da = author_week_dict[each].index(x)
    author_day[da]+=1
```

```
plt.xticks(range(7),['Mon','Tue','Wed','Thu','Fri','Sat','Sun',])
plt.bar(range(7),author_day)
plt.show()
print(author_day)
```



统计的结果略微有些违背直觉，CSDN的博主们普遍更喜欢在一周的上半段时间发布文章，而非通常认为的会在周末发文章，周一是大部分作者选择发布文章的高峰期。