量化分析 B 实验报告

学院名称：	数字经济与管理学院
专业名称：	大数据管理与应用
班　　级：	数据管理 2302
作　　者：	林子晗
学　　号：	2023113223

一、 概述

在本次期末大作业中，我通过爬取天财新闻上的所有新闻数据，进行数据的统计和分析，制作了多张图表来可视化这些数据。具体来说，我制作了以下几张图片：

1. 发布新闻数量按月统计的折线图：展示每月发布新闻的数量变化趋势。

2. 题目与内容摘要中高频词统计的折线图：展示出现次数最多的前 50 个字词及其出现的次数。

3. 供稿单位发布新闻数量的饼状图：展示各供稿单位发布新闻数量的占比。

4. 供稿单位平均阅读量的柱状图：展示各供稿单位发布新闻的平均阅读量。

5. 按周几统计平均阅读量的柱状图：展示在不同周数新闻的平均阅读量。

6. 新闻供稿单位发布新闻的平均字数和平均图片数量的散点图：展示各供稿单位发布新闻的平均字数和平均图片数量。

本次实验的目标是通过数据分析了解天财新闻的发布规律、学生阅读习惯和供稿单位的活跃度。主要通过 get 方法获取页面元素从而获得数据，在处理后得出结论并进行数据的可视化。

二、 实验路径

在本次实验中，我按照以下流程进行操作和处理：

1.数据爬取和保存：

- 使用 Python 的 requests 和 BeautifulSoup 库爬取天财新闻网站上的新闻数据。
- 将首页中爬取到的每篇新闻的内容（包括标题、摘要、发布日期、URL）保存到一个字典中。
- 再根据字典中的 URL 对页面的详细信息(包括具体发布日期、供稿单位、阅读量、文章字数、图片数量）进行爬取，并且保存在一个字典中。
- 将所有新闻数据以 JSON 格式保存到本地，方便后续处理和分析。

2.数据处理和存储：

- 从 JSON 文件中读取所有新闻数据，并使用内置库的 Counter 类将数据加载为一个 Counter 进行处理。
- 对新闻数据进行清洗，例如处理缺失值，日期转换等。

3.数据分析和图表绘制：

- 新闻数量按月统计：按月对新闻数量进行统计，并使用 matplotlib 绘制折线图。
- 词频统计：对新闻标题和摘要中的汉字进行统计，首先使用 jieba 库对文本进行分词处理，然后使用 Counter 类统计每个字词的出现次数，并绘制折线图。
- 供稿单位新闻数量统计：按供稿单位对新闻数量进行统计，取前 15 并使用 matplotlib 绘制饼状图。
- 供稿单位平均阅读量统计：按供稿单位计算新闻的平均阅读量，使用 matplotlib 绘制柱状图。
- 按星期几统计平均阅读量：根据新闻发布日期统计不同星期几的平均阅读量，使用 matplotlib 绘制柱状图。
- 供稿单位平均字数和平均图片数量统计：计算各供稿单位发布新闻的平均字数和平均图片数量，并使用 matplotlib 绘制散点图。

三、 数据分析

1. 学校新闻月发文量

变量定义：

每一篇新闻表示为 d_i，全部新闻集合为 $D = \{d_1, d_2, \ldots, d_n\}$。

统计过程：

本部分统计自 2020 年 12 月至 2023 年 6 月，假设其某个月发布文章的数量记为 M_t，获取 d_i 的月份记作 $T(d_i)$，初始化 $M_t=0$。

$$M_{T(d_i)} = M_{T(d_i)} + 1, \qquad \forall i \in \{1, 2, \ldots, n\}$$

对所有可能的 $T(d_i)$进行排序并作为横坐标，其所对应的发文量作为纵坐标，所得学校新闻月份发文量如图 1 所示。由图 1 可知，在统计区间内，2021 年至 2023 年，在每年 1、2 月，6、7 月左右发文量较少，由此可推断该时间段通常可能为每年寒暑假期间。

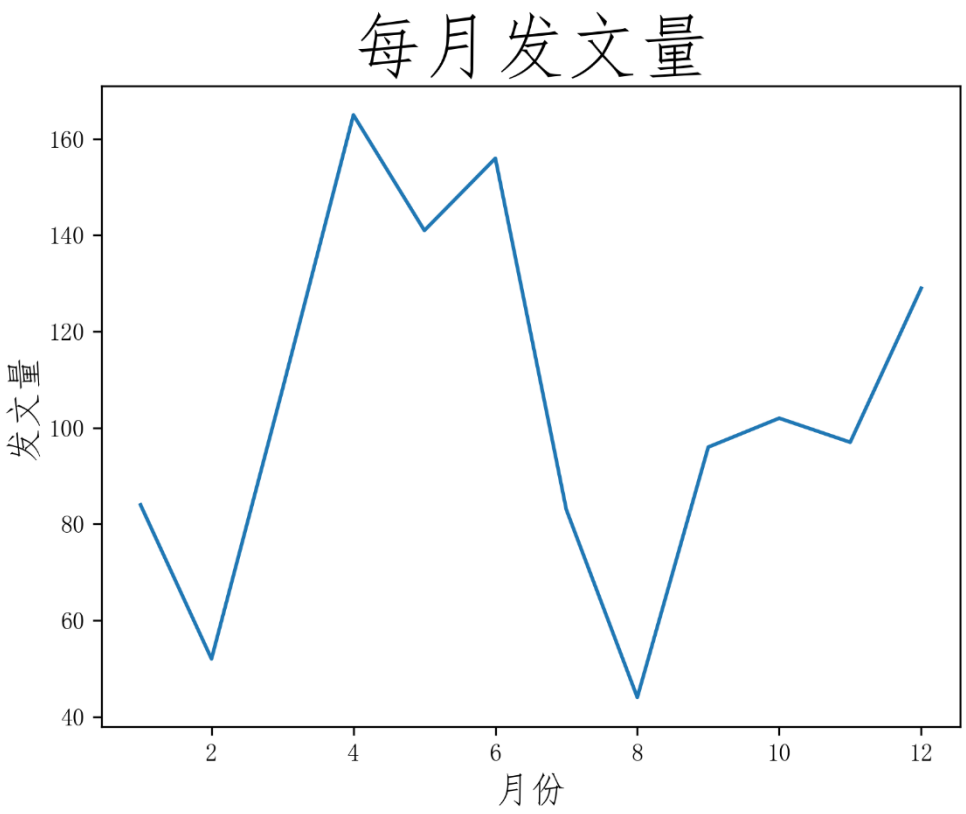

图 1 学校新闻每月发文量

2. 标题与摘要高频词统计

变量定义：

对于每篇新闻 d_i，每个字词记为 w，标题和摘要中的所有字词构成集合 $W(d_i)$，每个字词的出现次数记为 F(w)，初始化为 0。

统计过程：

统计所有新闻标题和摘要中的两字词和三字词出现次数，具体步骤如下：

- 初始化一个计数器，用于统计字词的出现次数。
- 遍历每篇新闻 d_i，提取其标题和摘要中的所有字词。
- 对于每个字词 w:

$$F(w) = F(w) + 1$$

统计结果中根据出现次数前五十的字词及其出现次数，绘制折线图。横坐标为词语，纵坐标为词语出现的次数。如图二所示。从图二中可知，‘我校’、‘工作’、‘教育’等词语出现频率最高。

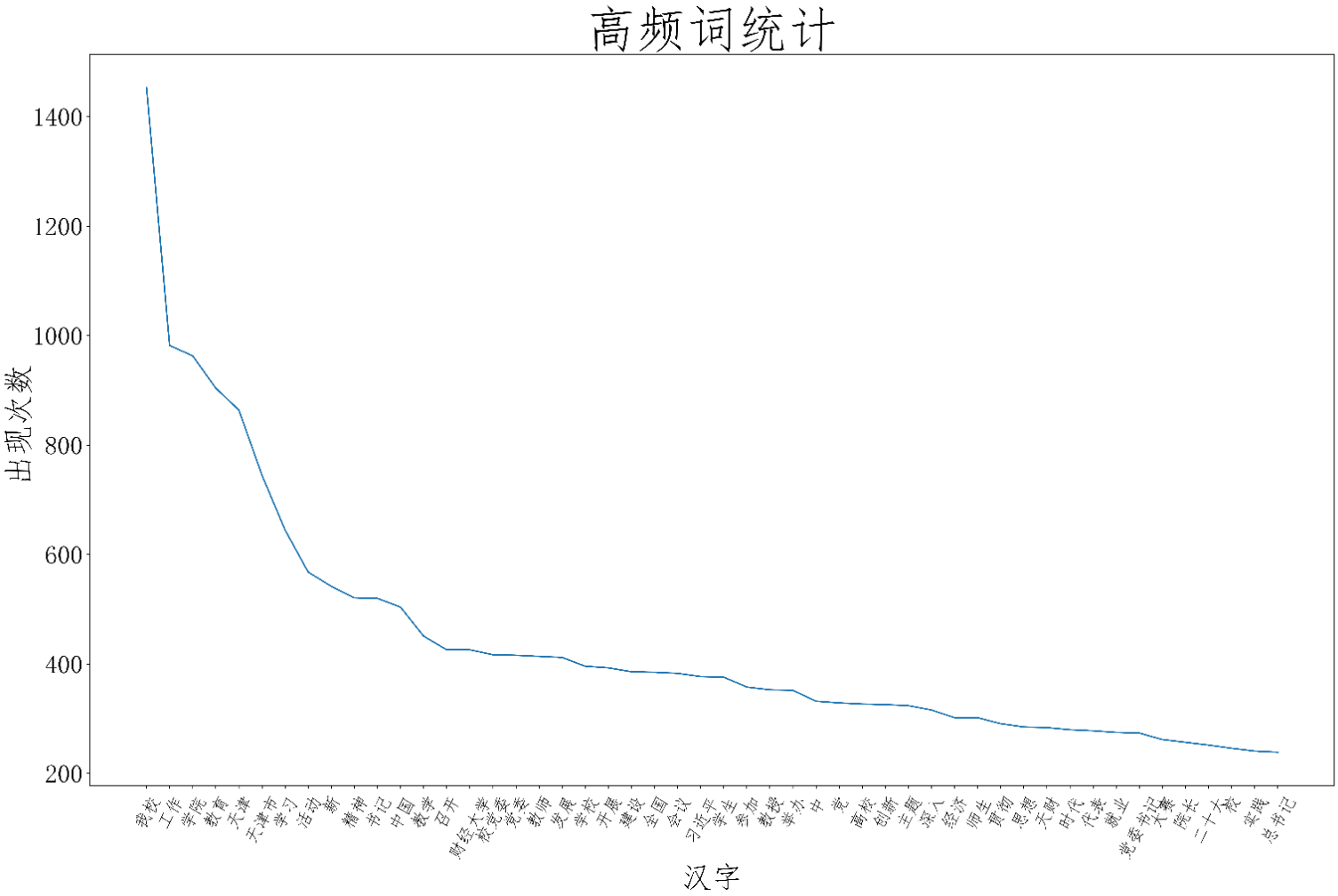

图 2 题目摘要词频统计

3. 供稿单位新闻数量统计

变量定义：

对于每篇新闻 d_i，其供稿单位记为 $C(d_i)$，全部供稿单位的集合记为 $C = \{C(d_1),C(d_2),\ldots\ldots,C(d_n)\}$，每个供稿单位发布新闻数记为 $N(C)$，初始化为 0。

统计过程：

- 初始化一个计数器，用于统计每个供稿单位的新闻发布数量。
- 遍历每篇新闻 d_i，获取其供稿单位 $C(d_i)$
- 对于每个供稿单位 S：

$$\boldsymbol{N(s) = N(s) + 1}$$

统计结果中每个供稿单位发布新闻的数量，并绘制饼状图。每个扇形表示一个供稿单位，扇形面积表示其发布新闻的数量占比。如图 3 所示，从图 3 中可知，教务处与组织部供稿数最多。

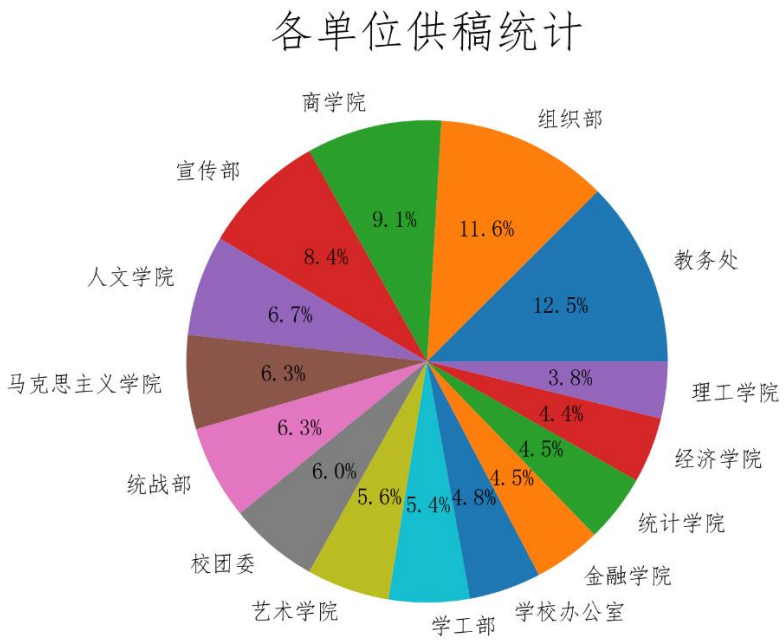

图 3 供稿单位新闻数量统计

4. 新闻供稿单位平均阅读量

变量定义：

对于每篇新闻 d_i，其供稿单位记为 $S(d_i)$，阅读量记为 $R(d_i)$。每个供稿单位的新闻总数记为 $D(s)$，其平均阅读量记为$\bar{R}(s)$，总阅读量为 $T(S)$。

统计过程：

- 初始化一个字典，用于存储每个供稿单位的总阅读量和新闻数量。
- 遍历每篇新闻 d_i，获取其供稿单位 $S(d_i)$和阅读量 $R(d_i)$。
- 对于每个供稿单位 S:

$$T(S) = T(s) + R(d_i)$$

$$D(s) = D(s) + 1$$

- 计算每个供稿单位的平均阅读量：

$$\overline{R}(s) = T(s) / D(s)$$

统计结果中每个供稿单位的平均阅读量，并绘制柱状图。横坐标为供稿单位，纵坐标为平均阅读量。如图 4 所示。从图 4 中可知，金融学院与经济学院的供稿平均阅读量最高

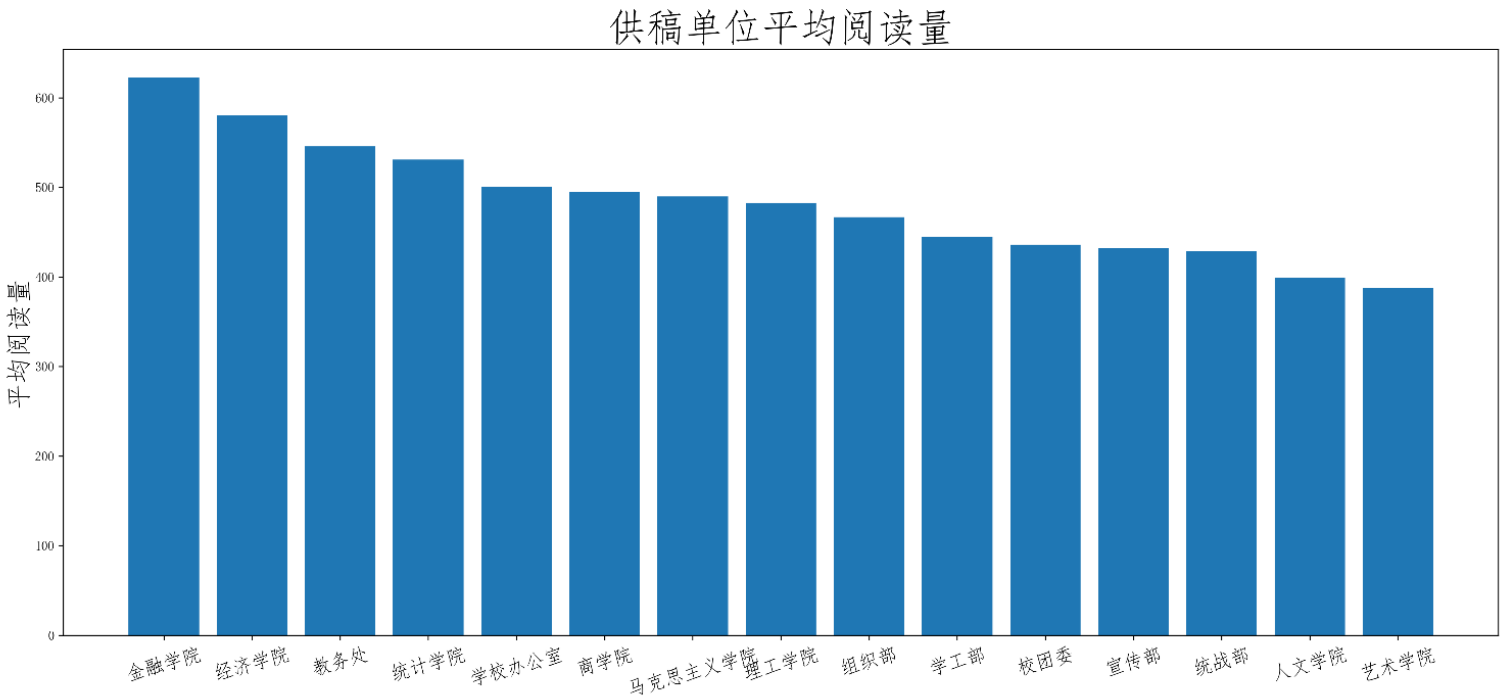

图 4 新闻供稿单位平均阅读量

5. 按周几统计平均阅读量

变量定义：

对于每篇新闻 d_i，其发布日期为 $T(d_i)$，阅读量记为 $R(d_i)$。所有新闻按周几分类的集合记为 $W = \{W_1, W_2, \ldots, W_7\}$，其中 W_j 表示星期 j 的新闻集合，其平均阅读量记为 $\overline{R}(W_j)$。

统计过程：

- 初始化一个字典，用于存储每周几的总阅读量和新闻数量。
- 遍历每篇新闻 d_i，获取发布日期 $T(d_i)$ 对应的周几 $W(T(d_i))$ 和阅读量 $R(d_i)$。
- 对于每周 j：

$$Total(W_j) = Total(W_j) + R(d_i)$$

$$Count(W_j) = Count(W_j) + 1$$

- 计算每周 j 的平均阅读量：

$$\overline{R}(W_j) = \frac{Total(W_j)}{Count(W_j)}$$

统计结果中每周 j 的平均阅读量，并绘制柱状图。横坐标为周几，纵坐标为平均阅读量。如图 5 所示。从图 5 可知，每周二与每周四发布的文章阅读量最高。

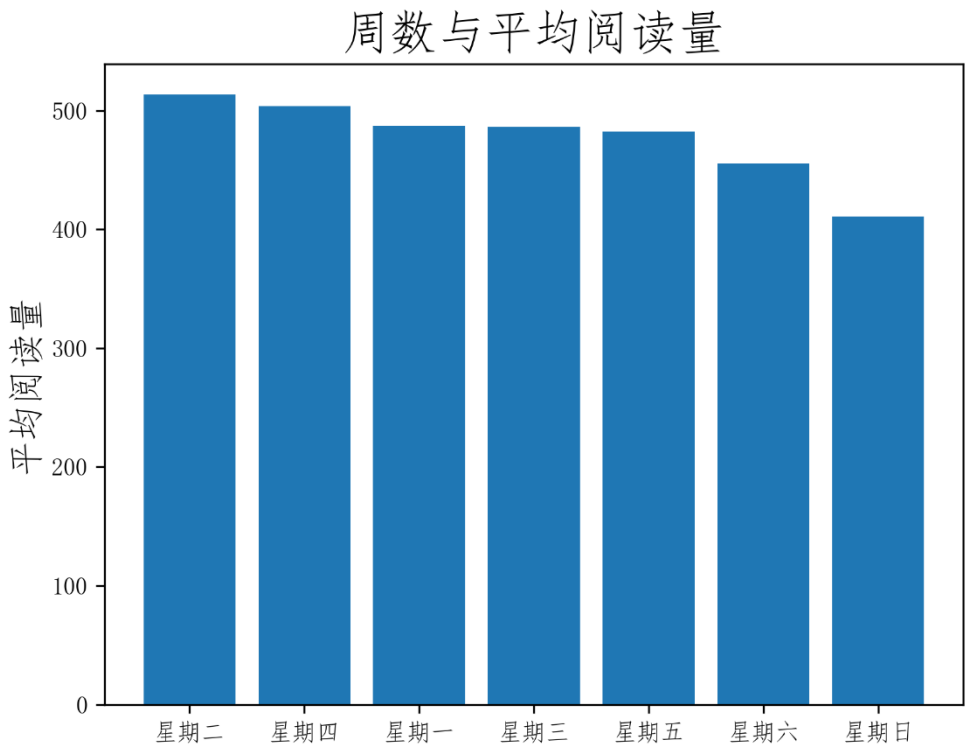

图 5 按周几统计的平均阅读量

6. 新闻供稿单位平均字数和平均图片数量

变量定义：

对于每篇新闻 d_i，其供稿单位记为 $S(d_i)$，字数记为 $W(d_i)$，图片数量记为 $P(d_i)$。每个供稿单位的新闻集合记为 $D(s)$，其平均字数和平均图片分别记为 $\overline{W}(s)$,$\overline{P}(s)$。

统计过程：

- 初始化一个字典，用于存储每个供稿单位发布新闻的平均字数和平均图片数量。
- 遍历每篇新闻 d_i，获取其供稿单位 $S(d_i)$，字数 $W(d_i)$和图片数量 $P(d_i)$。
- 对于每个供稿单位 s：

$$total\_words(s) = total\_words(s) + W(d_i)$$

$$total\_pictures(s) = total\_pictures(s) + P(d_i)$$

$$count(s) = count(s) + 1$$

- 计算每个供稿单位的平均字数和平均图片数：

$$\overline{W}(s) = \frac{total\_words(s)}{count(s)}$$

$$\overline{P}(s) = \frac{total\_pictures(s)}{count(s)}$$

统计结果中包含每个供稿单位的平均字数和平均图片数量，绘制散点图。横坐标为平均字数，纵坐标为平均图片数量。如图 6 所示，从图 6 中可知，越靠近右上角，稿件的平均字数和平均图片数就更多。

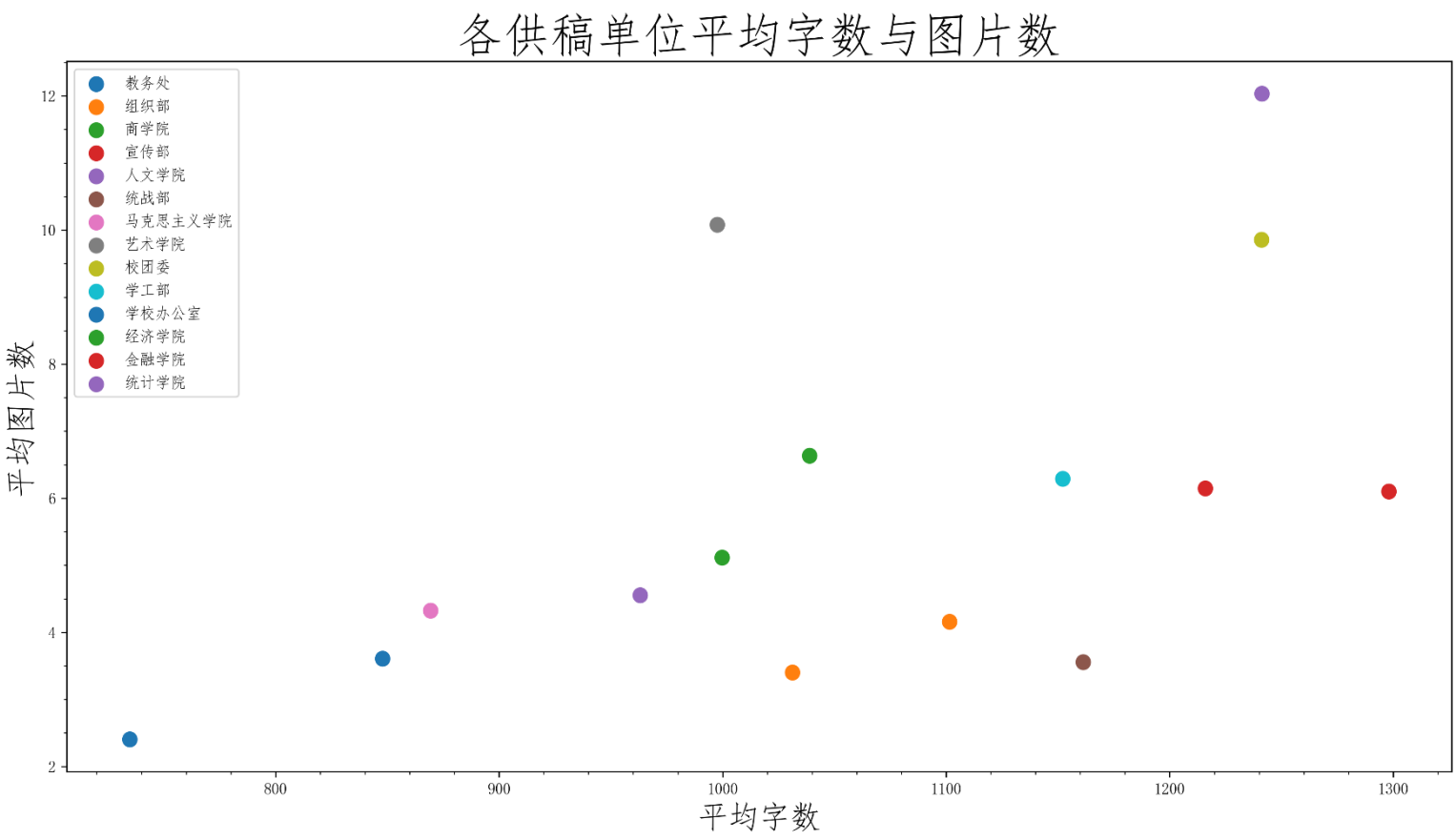

图 6 新闻供稿单位平均字数和平均图片数量

四、 总结

其实整体没有什么特别难的点，唯二个让我浪费了时间的点出现在学校 Web 的阅览量爬取还有微信链接里的时间获取（这个太麻烦了导致我根本不想拿数

据），第一时间没发现他是前端通过 js 返回的数据，让我一直感觉是自己代码有问题才一直拿到空数据。除此之外影响速度的点可能主要是对 matplotlib 这个库不太熟，导致出的图一直不符合自己预期。算下来总体时间稍微有一点久了。

最后，**爬虫爬的好，牢饭吃到饱。**