

从“彩礼吧”看彩礼

——关于彩礼话题的数据分析

目录

目录	1
一、选题背景	2
二、数据获取	2
2.1 数据来源与爬取方式	2
2.2 字段与样本概况	2
三、数据预处理	3
3.1 字段筛选与清洗	3
3.2 构造“热度”指标并筛选高热度帖子	4
四、描述性统计与基础分析	4
4.1 高频词与整体话题氛围	4
4.2 用户性别分布	6
4.3 发帖时间统计（以 11 月为例）	7
五、深度关系分析	9
5.1 地域与彩礼相关性分析	9
5.2 帖子热度影响因素	11
六、文本内容理解与生成	12
6.1 大模型文本模拟生成：一则反对彩礼的贴文	12
七、总结	13
7.1 关于彩礼话题的发现	13
7.2 方法与技能上的收获	14

一、选题背景

“彩礼”作为中国传统婚俗中的重要环节，如今已成为一个触及代际观念、性别平等与家庭伦理的社会焦点。一方面，许多支持者视其为对女方家庭付出的合理回馈，是婚姻仪式感、社会认可与祝福的物质载体；另一方面，愈演愈烈的高额与攀比式彩礼，则被广泛批评为加剧年轻人婚育压力、激化两性对立，甚至异化了婚姻的情感本质，使之带上交易色彩。相关讨论在各类网络平台持续高热，尤其在百度“彩礼吧”这类匿名性较强的社区，普通网民得以相对直白地宣泄焦虑、表达不满或陈述期待。为系统梳理不同性别、立场普通民众的真实态度，本次作业将以“网民对彩礼的看法”为题，对“彩礼吧”内容展开一次结合描述性统计与文本挖掘的综合分析，尝试揭示表层观点之下的情感与逻辑。

二、数据获取

2.1 数据来源与爬取方式

本次分析主要使用百度贴吧数据，爬取贴吧“彩礼吧”。

使用助教提供的 [dataHelper.py](#) 工具，发现输出的 csv 文件输出为乱码。通过向大模型询问后，修正了原代码。运行后在命令行中输入待爬取的贴吧名称“彩礼”，自动抓取若干页帖子并保存为 CSV 文件。生成的主数据文件为：`./result/彩礼.csv`。

2.2 字段与样本概况

通过修改并运行 [checkData.py](#) 的脚本，对 `彩礼.csv` 进行检查，可以得到：

```
C:\Users\YF\Downloads\2025秋季学期计算概论C大作业\2025秋季学期计算概论C大作业\tieba>python checkData.py
-----
数据的第一个为：
user_name          秋虫不知晦朔
nick_name          光荣的红花少年
level              13
glevel             9
gender             1
is_vip             False
title              【彩礼吧】水经验专用贴
text               【彩礼吧】水经验专用贴\n需要水经验的来着集合喽！（如果真有人需要的话）
view               11152
reply              905
share              0
agree              79
disagree           1
create_time        2024-04-11 21:16:16
last_time          2025-11-22 15:58:44
Name: 0, dtype: object
-----
数据的字段为：
Index(['user_name', 'nick_name', 'level', 'glevel', 'gender', 'is_vip',
       'title', 'text', 'view', 'reply', 'share', 'agree', 'disagree',
       'create_time', 'last_time'],
      dtype='object')
-----
获取数据的数目为：
9740
```

图 1 运行 checkData.py 得到的结果

可以看出，彩礼.csv 中包含 9740 条数据，数据字段包括'user_name', 'nick_name', 'level', 'glevel', 'gender', 'is_vip', 'title', 'text', 'view', 'reply', 'share', 'agree', 'disagree', 'create_time', 'last_time'

三、数据预处理

3.1 字段筛选与清洗

在后续分析中，部分字段如 nick_name、glevel、is_vip 与本次研究问题相关性较弱，因此在预处理阶段适当删去，只保留与用户属性、帖子内容、互动行为密切相关的字段（如 user_name、level、gender、title、text、view、reply、share、agree、disagree、create_time、last_time 等）。

提示词：

在后续分析中，部分字段如 nick_name、glevel、is_vip 与本次研究问题相关性较弱，因此在预处理阶段适当删去，只保留与用户属性、帖子内容、互动行为密切相关的字段（如 user_name、level、gender、title、text、view、reply、share、agree、disagree、create_time、last_time 等）。

大数据回复：

```
1 import pandas as pd
2 import os
3
4 # 让用户输入要清洗的CSV文件名（如“相亲吧.csv”）
5 file_name = input("请输入要清洗的CSV文件名（含后缀，如：相亲吧.csv）：")
6 file_path = f"./result/{file_name}"
7
8 # 检查文件是否存在
9 if os.path.exists(file_path):
10     # 读取数据
11     df = pd.read_csv(file_path, encoding='utf-8-sig')
12
13     # 数据清洗操作示例
14     print("原始数据形状：", df.shape)
15
16     # 1. 删除缺失值（可选）
17     df = df.dropna(subset=['user_name', 'title']) # 关键字段缺失则删除
18
19     # 2. 去重（可选）
20     df = df.drop_duplicates(subset=['user_name', 'title', 'create_time'])
21
22     # 3. 清理文本（去除特殊字符）
23     df['text'] = df['text'].astype(str).str.replace(r'[\u4e00-\u9fa5a-zA-Z0-9\s]', '', regex=True)
24
25     # 保存清洗后的数据
26     clean_file_path = f"./result/clean_{file_name}"
27     df.to_csv(clean_file_path, index=False, encoding='utf-8-sig')
28     print(f"清洗完成！已保存至：{clean_file_path}")
29 else:
30     print("文件不存在，请检查文件名！")
31
```

图 2 cleaning.py

运行 cleaning.py，得到内容如下：

```
C:\Users\YF\Downloads\2025秋季学期计算概论C大作业\2025秋季学期计算概论C大作业\tieba>python cleaning.py
请输入要清洗的CSV文件名（含后缀，如：相亲吧.csv）：彩礼.csv
原始数据形状：(9740, 15)
清洗完成！已保存至：./result/clean_彩礼.csv

C:\Users\YF\Downloads\2025秋季学期计算概论C大作业\2025秋季学期计算概论C大作业\tieba>
```

图 3 运行 cleeaning.py

可以得知，现在已经得到清洗后的数据，可以开始进行数据分析了。

3.2 构造“热度”指标并筛选高热度帖子

为了集中分析讨论度最高的内容，我定义了一个综合热度指标：

```
[text{hot} = 3 times text{view} + 5 times text{reply} + 3 times text{share}
+ 1 times text{agree} + 1 times text{disagree}]
```

现在将综合热度指标输入大模型，要求统计热度最高的 50 个帖子数据。

提示词：

我收集到了彩礼贴吧的数据，放在了'./result/彩礼.csv'路径下。写一段python代码，统计一下热度最高的帖子（view*3 +reply*5+share*3+agree*1+disagree*1），找到100个热度最高的帖子，写成csv文件存进./result，命名为“热度100_彩礼”。

运行大模型输出的代码 [hot.py](#), 得到以下内容：

```
C:\Users\YF\Downloads\2025秋季学期计算概论C大作业\2025秋季学期计算概论C大作业\tieba>python hot.py
已生成热度最高的100个帖子文件，路径：./result/热度100_彩礼.csv
```

图 4 运行 hot.py

四、描述性统计与基础分析

本节从整体上分析“彩礼吧是什么样的一个讨论空间”，从词汇氛围、性别构成和时间分布三个维度展开。

4.1 高频词与整体话题氛围

首先使用 jieba 分词对 text 字段进行分词，并结合自定义停用词表 [stopwords2.txt](#) 去除无实义词（如“男方”“女方”“不是”“没有”等）。在此基础上统计词频，得到 Top 15 高频词。

提示词：

我收集到了彩礼贴吧的数据，放在了'./result/彩礼.csv'路径下。写一段python 代码，统计一下 text 中使用频率最高的 15 个词，可以使用 jieba 库，停用词用同目录下的 [stopwords2.txt](#)。

大模型生成代码：

```
1 import pandas as pd
2 import jieba
3 from collections import Counter
4 import re
5
6 # 定义加载停用词的函数
7 def load_stopwords(file_path):
8     """加载停用词文件"""
9     try:
10         with open(file_path, 'r', encoding='utf-8') as f:
11             stop_words = [word.strip() for word in f.readlines()]
12             return set(stop_words)
13     except FileNotFoundError:
14         print(f"错误: 找不到停用词文件 {file_path}")
15         # 如果文件不存在，返回默认的基础停用词集合
16         return {'的', '了', '是', '我', '你', '他', '这', '那', '谁', '哪', '不', '在', '有', '和'}
17
18 # 读取数据
19 df = pd.read_csv('./result/彩礼.csv')
20 df['text'] = df['text'].fillna('')
21
22 # 加载停用词 (确保文件路径正确)
23 stop_words = load_stopwords('./cn_stopwords.txt')
24
25 # 文本预处理: 去除特殊字符
26 def preprocess_text(text):
27     text = re.sub(r'[\u4e00-\u9fa5]', '', text) # 仅保留中文
28     return text.strip()
29
30 df['processed_text'] = df['text'].apply(preprocess_text)
31
32 # 合并所有文本并分词
33 all_text = ''.join(df['processed_text'].tolist())
34 words = jieba.cut(all_text)
35
36 # 过滤停用词并转换为1的向量
37 words = [word for word in words if word not in stop_words and len(word) > 1]
38
39 # 统计词频
40 word_freq = Counter(words)
41 top10_words = word_freq.most_common(10)
42
43 # 输出结果
44 print(f"文本中使用频率最高的十个词: ")
45 for i, (word, count) in enumerate(top10_words, 1):
46     print(f"{i}. {word} - {count}")
```

图 5 frequency.py

运行 [frequency.py](#)，得到结果如下：



图 6 运行 frequency.py

为了进一步加强数据可读性，我选择 wordcloud 库制作词语云图，并将需求输入给大模型。

提示词：

写一段 python 代码，统计词频最高的 80 个词，并且生成一个词云图。停用词储存在同目录下的 stopwords2.txt。

运行 `wordcloudan.py` 得到:

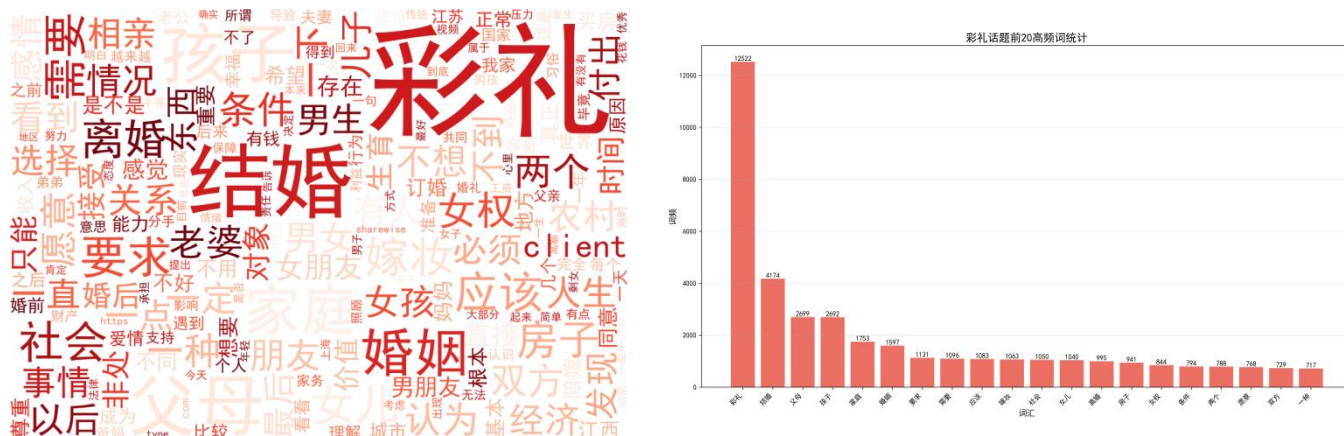


图 7 wordcloudan.py 运行结果

高频词中,既包含“花钱”“房子”“车”“嫁妆”等与物质条件直接相关的词汇,也出现“父母”“孩子”“孝顺”“补偿”“公平”等与家庭关系、价值观相关的表达。可以看出,彩礼讨论并不仅仅是“多少钱”的问题,还牵涉到代际关系、性别分工与“是否公平”的价值冲突。同时,“需要”“应该”“要求”等强烈语气的高频词也反映出网民对于彩礼的态度比较强硬。

4.2 用户性别分布

借鉴示例中的 [gender.py](#)，对原数据中的 gender 字段进行统计。为了能更清晰地展示性别比，要求改变饼图颜色、注释。提示词：

模仿 `gender.py`，来统计彩礼这个 `csv` 的性别，并且对颜色、注释进行调整

得到代码运行结果:

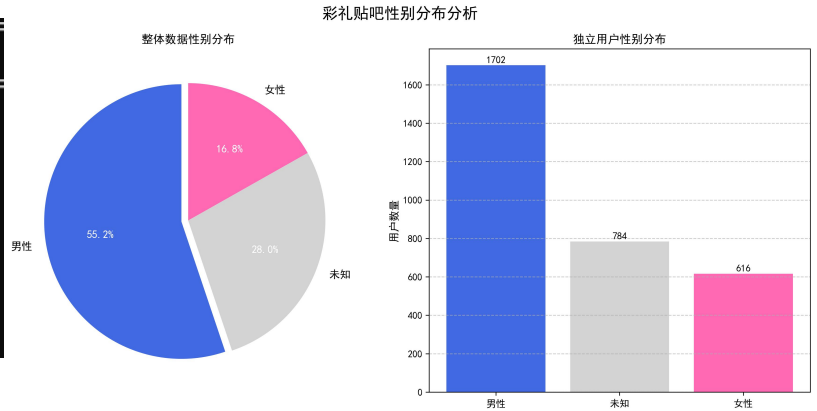
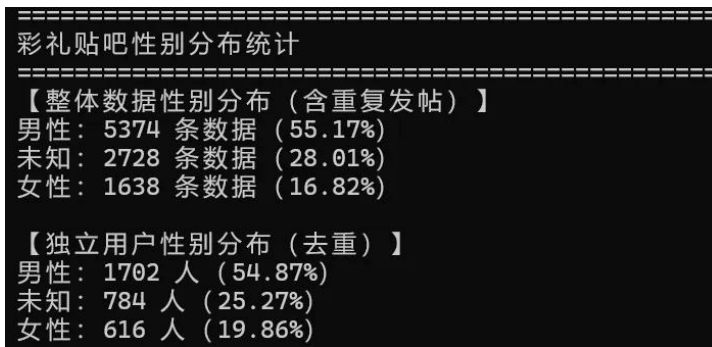


表 8 gender.py 代码运行结果

通过数据分析发现，男性发帖显著多于女性，说明彩礼话题在该贴吧中更容易由男性发起或参与，可能与“支付彩礼的一方往往是男方家庭”这种现实压力有关。在绝大多数涉及彩礼的婚恋传统中，男方及其家庭是彩礼的支付方。这直接关系到他们的经济储蓄、家庭债务乃至生活规划。因此，男性对彩礼金额、女方家庭要求等话题有更直接、更强烈的经济焦虑和讨论需求。男性常常感觉自己处于被要求、被评估的位置。发帖是一种寻求帮助、共鸣和情绪宣泄的途径。

4.3 发帖时间统计（以 11 月为例）

为了分析发帖时间规律，我决定以 2025 年 11 月为例，以日（24 小时）、周（7 天）分别为尺度考察发帖量随时间的变化。

提示词：以 2025 年 11 月为例，以日（24 小时）、周（7 天）分别为尺度，分析帖子发布时间与发贴量之间的关系。

大模型回复：


```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import numpy as np
4 from datetime import datetime
5
6 # 设置中文字体
7 plt.rcParams['font.sans-serif'] = ['SimHei']
8 plt.rcParams['axes.unicode_minus'] = False
9
10 # 加载数据
11 file_path = './result/彩礼.csv'
12 df = pd.read_csv(file_path)
13
14 # ===== 时间数据预处理 (仅保留2025年11月) =====
15 df['create_time'] = pd.to_datetime(df['create_time'], errors='coerce')
16
17 # 筛选2025年11月数据
18 df_202511 = df[
19     (df['create_time'].dt.year == 2025) &
20     (df['create_time'].dt.month == 11)
21 ].copy()
22
23 # 过滤无效时间数据
24 df_202511 = df_202511.dropna(subset=['create_time'])
25
26 if len(df_202511) == 0:
27     print("暂无2025年11月彩礼数据!")
28 else:
29     # 探索时间特征 (11月内)
30     df_202511['day'] = df_202511['create_time'].dt.day # 日期 (1-30)
31     df_202511['hour'] = df_202511['create_time'].dt.hour # 小时
32     df_202511['weekday'] = df_202511['create_time'].dt.weekday # 星期 (0=周一-6=周日)
33     df_202511['week'] = df_202511['create_time'].dt.isocalendar().week # 周数
34
35 # ===== 11月内发帖量统计 =====
36 # 1. 按日期统计 (1-30日)
37 daily_posts = df_202511.groupby('day').size().reset_index(name='发帖量')
38 daily_posts['日期'] = '11月' + daily_posts['day'].astype(str) + '日'
39
40 # 2. 按小时统计
41 hourly_posts = df_202511.groupby('hour').size().reset_index(name='发帖量')
42
43 # 3. 按星期统计
44 weekday_mapping = {0: '周一', 1: '周二', 2: '周三', 3: '周四', 4: '周五', 5: '周六', 6: '周日'}
45 df_202511['星期'] = df_202511['weekday'].map(weekday_mapping)
46 weekday_posts = df_202511.groupby('星期').size().reset_index(name='发帖量')
47
48 # 4. 按周统计 (11月包含的周)
49 week_posts = df_202511.groupby('week').size().reset_index(name='发帖量')
50 week_posts['周数'] = '第' + week_posts['week'].astype(str) + '周'

```

```

51
52 # ===== 输出统计结果 =====
53 print("="*50)
54 print("2025年11月帖子发布时间分析")
55 print("="*50)
56 print(f"11月总发帖量: {len(df_202511)} 条")
57 print(f"11月发帖日期范围: {df_202511['create_time'].min()} 至 {df_202511['create_time'].max()}")
58 print(f"11月日均发帖量: {len(df_202511)/len(df_202511['day'].unique()):.2f} 条")
59
60 print("\n【11月按日期发帖量】")
61 print(daily_posts[daily_posts['发帖量'] > 0]) # 仅显示有发帖的日期
62
63 print("\n【11月按小时发帖量】")
64 print(hourly_posts[hourly_posts['发帖量'] > 0].sort_values('hour'))
65
66 print("\n【11月按星期发帖量】")
67 print(weekday_posts)
68
69 # ===== 可视化分析 =====
70 fig, axes = plt.subplots(2, 2, figsize=(16, 12))
71 fig.suptitle("2025年11月帖子发布时间分布分析", fontsize=18, fontweight='bold')
72
73 # 1. 按日期分布
74 ax1 = axes[0, 0]
75 ax1.bar(daily_posts['日期'], daily_posts['发帖量'], color='#1f77b4', alpha=0.8)
76 ax1.set_title("11月按日期发帖量分布", fontsize=14)
77 ax1.set_xlabel('日期')
78 ax1.set_ylabel('发帖量')
79 ax1.tick_params(axis='x', rotation=45)
80 ax1.grid(axis='y', linestyle='--', alpha=0.7)
81
82 # 2. 按小时分布
83 ax2 = axes[0, 1]
84 ax2.plot(hourly_posts['hour'], hourly_posts['发帖量'], marker='s', linewidth=2, color='#ff7f0e')
85 ax2.fill_between(hourly_posts['hour'], hourly_posts['发帖量'], alpha=0.3, color='#ff7f0e')
86 ax2.set_title("11月按小时发帖量趋势", fontsize=14)
87 ax2.set_xlabel('小时 (24小时制)')
88 ax2.set_ylabel('发帖量')
89 ax2.set_xticks(range(0, 24, 2))
90 ax2.grid(linestyle='--', alpha=0.7)
91
92 # 3. 按星期分布
93 ax3 = axes[1, 0]
94 ax3.bar(weekday_posts['星期'], weekday_posts['发帖量'], color='#2ca02c', alpha=0.8)
95 ax3.set_title("11月按星期发帖量分布", fontsize=14)
96 ax3.set_xlabel('星期')
97 ax3.set_ylabel('发帖量')
98 ax3.grid(axis='y', linestyle='--', alpha=0.7)
99
100 # 4. 按周分布
101 ax4 = axes[1, 1]
102 ax4.pie(week_posts['周数'], labels=week_posts['周数'], autopct='%1.1f%%',
103        colors=['#d62728', '#9467bd', '#8c564b', '#e377c2'], startangle=90)
104 ax4.set_title("11月按周发帖量占比", fontsize=14)
105
106 plt.tight_layout()
107 plt.savefig("./result/2025年11月发帖时间分析.png", dpi=300, bbox_inches='tight')
108
109 # ===== 峰值分析 =====
110 peak_day = daily_posts.loc[daily_posts['发帖量'].idxmax()]
111 peak_hour = hourly_posts.loc[hourly_posts['发帖量'].idxmax()]
112 peak_weekday = weekday_posts.loc[weekday_posts['发帖量'].idxmax()]
113
114 print("\n" + "="*50)
115 print("11月发帖量峰值分析")
116 print("="*50)
117 print(f"峰值日期: {peak_day['日期']} ({peak_day['发帖量']} 条)")
118 print(f"峰值时段: {peak_hour['hour']} 点 ({peak_hour['发帖量']} 条)")
119 print(f"峰值星期: {peak_weekday['星期']} ({peak_weekday['发帖量']} 条)")
120
121 plt.show()

```

运行得到:



表 9 运行 timequantity.py

晚间（20 - 24 点）发帖高峰明显，可能说明用户更倾向于在下班或休息时间围绕彩礼展开情绪性讨论。结合彩礼话题的敏感性，这种“夜间高峰”也可能与用户在相对私密时段更愿意表达压力、焦虑和吐槽有关。

11 月发帖高峰出现在第二周，可能是与“双十一”（光棍节）有关，说明网民的情绪会受到周围环境（如特殊节日）的影响。同时可能由于某一突发事件，导致某个或某群网友反应较大，高强度发帖。

五、深度关系分析

在掌握了整体分布之后，本节进一步探究不同变量之间的关系，例如地域与彩礼金额、影响热度因素等。

5.1 地域与彩礼相关性分析

注意到在之前研究中，高频词中出现部分省份名称。我国彩礼金额差异显著，从数万到数十万不等。其数额分布与地域经济水平、传统习俗及性别比例等因素密切相关，呈现出鲜明的区域特色。接下来通过 python 代码分析帖文中有关不同地域彩礼差异。

提示词：现在来考虑地域因素。写一段 python 代码，要求统计所有数据中包括地域的部分，比如含有“江浙”“北京”“鄂”“东北”“西北”“东三省”的语句。一般来说地域都会带上一个数字，比如“江浙沪 18.8w”“东三省 6 万 8”，或者用一个虚量，比如“多”“少”“不多”，并且输出地域彩礼平均数、具体内容。

代码输出结果：

地域	平均金额_万元	中位数_万元	样本数
上海	35.18	15.00	49
东三省	12.37	10.00	26
北京	24.65	4.89	56
华北	51000.96	9.72	34
华南	15.89	4.69	13
四川	22.94	9.00	29
天津	82.61	15.00	13
安徽	20.20	15.90	24
山东	16.64	6.00	37
广东	21.31	11.00	54
江浙沪	27.95	9.00	89
江西	43.12	21.90	78
河南	21.00	12.53	44
湖北	21.76	12.00	19
湖南	13.64	11.83	18
福建	21.18	15.00	15
西北	15.84	8.00	35
西南	22.95	9.39	34

地域	平均金额_万元	中位数_万元	样本数
上海	35.18	15.00	49
东三省	12.37	10.00	26
北京	24.65	4.89	56
华北	51000.96	9.72	34
华南	15.89	4.69	13
四川	22.94	9.00	29
天津	82.61	15.00	13
安徽	20.20	15.90	24
山东	16.64	6.00	37
广东	21.31	11.00	54
江浙沪	27.95	9.00	89
江西	43.12	21.90	78
河南	21.00	12.53	44
湖北	21.76	12.00	19
湖南	13.64	11.83	18
福建	21.18	15.00	15
西北	15.84	8.00	35
西南	22.95	9.39	34

彩形式	五金三金	仪式服务	家电家具	房产相关	现金类	车辆相关
东三省	11	3	3	14	19	13
北京	23	5	6	47	59	26
华北	10	3	1	16	20	12
华南	3	0	1	4	9	5
四川	8	4	2	12	19	8
天津	5	2	1	3	7	4
安徽	10	2	3	15	14	9
山东	10	5	5	17	20	10
广东	11	9	3	20	37	15
江浙沪	47	28	25	95	110	66
江西	24	14	6	34	73	22
河南	24	13	6	32	49	24
湖北	4	3	1	9	17	6
湖南	11	9	4	16	25	10
福建	5	4	2	7	10	3
西北	9	4	2	11	16	11
西南	4	6	3	13	19	8

表 10 shengfentongji.py 运行结果

进一步进行可视化处理，提示词：

生成柱状图，不要华北；不考虑地域差异，分析不同类别的彩礼占比，用饼状图。

大数据生成了 [map.py](#) 和 [cat.py](#):

```
1 import pandas as pd
2 from pyecharts import options as opts
3 from pyecharts.charts import Bar
4 from pyecharts.globals import ThemeType
5 import os
6
7 # 数据 (剔除华北)
8 data = {
9     '省份': ['上海', '东三省', '北京', '华南', '四川', '天津', '安徽', '山东', '广东', '江浙沪', '江西', '河南', '湖北', '湖南', '福建', '西北', '贵州'],
10     '平均金额_万元': [15.18, 12.37, 24.49, 15.89, 22.94, 22.40, 18.3, 18.84, 23.34, 27.95, 43.11, 21, 21.76, 13.64, 21.18, 15.84, 22.91],
11     '中位数_万元': [15, 18, 4.89, 4.69, 9, 15, 15.4, 6, 11, 9, 21.9, 22.53, 12, 11.83, 15, 8, 9.39],
12     '样本数': [49, 28, 56, 15, 29, 13, 24, 37, 54, 89, 78, 44, 19, 15, 15, 54]
13 }
14
15 df = pd.DataFrame(data)
16
17 # 按平均金额降序排列
18 df_sorted = df.sort_values('平均金额_万元', ascending=False)
19
20 # 创建柱状图
21 bar_chart = (
22     Bar.from_opts(df_sorted[['省份', '平均金额_万元', '中位数_万元']])
23     .add_series('平均金额_万元', 'df_sorted[平均金额_万元].tolist()', '省份', '平均金额_万元')
24     .add_series('中位数_万元', 'df_sorted[中位数_万元].tolist()', '省份', '中位数_万元')
25     .set_global_opts(title='各地区彩礼金额对比柱状图 (剔除华北)', subtitle='平均金额 vs 中位数',
26                      tooltip_opts=TooltipOpts(trigger='item', formatter='{b}<br>平均金额: (c)万元<br>中位数: (d)万元<br>样本数: (e)', extra_link_func=lambda p: {'function': lambda p: df_sorted[p['省份']].tolist()[p['省份'].tolist().index(p['省份'])]}),
27                      datazoom_opts=DataZoomOpts(type_='slider', orient='horizontal'),
28                      set_series_opts=SeriesOpts(is_show=True, position='right', font_size=10))
29
30 # 保存图片
31 output_path = os.path.join(os.path.dirname(__file__), '彩礼金额对比柱状图_剔除华北.html')
32 bar_chart.render(output_path)
33
34 print(f'柱状图生成成功: {output_path}')
35 print('数据源: 各地区彩礼金额对比柱状图 (剔除华北)')
36
37 for idx, row in df_sorted.iterrows():
38     print(f'省份: {row[\"省份\"]} | 平均金额: {row[\"平均金额_万元\"]} 万元 | 中位数: {row[\"中位数_万元\"]} 万元 | 样本数: {row[\"样本数\"]}')
```

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5 # 设置中文字体
6 plt.rcParams['font.sans-serif'] = ['SimHei'] # 黑体
7 plt.rcParams['axes.unicode_minus'] = False
8
9 # 数据计算
10 data = {
11     '五金三金': [11, 23, 10, 3, 8, 5, 10, 18, 11, 47, 24, 24, 4, 11, 5, 9, 41],
12     '仪式服务': [13, 5, 3, 8, 4, 2, 2, 5, 9, 28, 14, 13, 3, 9, 4, 4, 6],
13     '家电家具': [15, 6, 1, 1, 2, 1, 3, 5, 3, 25, 6, 6, 1, 4, 2, 2, 3],
14     '房产相关': [14, 47, 16, 4, 12, 3, 15, 17, 28, 95, 34, 32, 9, 16, 7, 11, 13],
15     '现金类': [19, 59, 28, 9, 19, 7, 14, 28, 37, 118, 73, 49, 17, 25, 18, 16, 19],
16     '车辆相关': [13, 26, 12, 5, 8, 4, 9, 18, 15, 66, 22, 24, 6, 18, 3, 11, 8]
17 }
18
19 df = pd.DataFrame(data)
20 categories = list(df.keys())
21 totals = [df[cat].sum() for cat in categories]
22 percentages = [(t/sum(totals))*100:1)% for t in totals]
23
24 # 创建饼图
25 fig, ax = plt.subplots(figsize=(10, 7))
26 colors = ['#ff9999', '#6666ff', '#99ff99', '#ffcc99', '#ff99cc', '#cc99ff']
27
28 wedges, texts, autotexts = ax.pie(
29     totals,
30     labels=categories,
31     colors=colors,
32     autopct='%1.1f%%',
33     startangle=90,
34     explode=[0.05 if t == max(totals) else 0 for t in totals] # 突出最大值
35 )
36
37 # 美化文字
38 for autotext in autotexts:
39     autotext.set_color('white')
40     autotext.set_fontweight('bold')
41
42 ax.set_title('彩礼类别占比分布', fontsize=16, fontweight='bold', pad=20)
43 plt.tight_layout()
44
45 # 保存图片
46 plt.savefig('彩礼类别占比饼图.png', dpi=300, bbox_inches='tight')
47 plt.show()
48
49 # 输出统计结果
50 print(f'彩礼类别统计: {categories}')
```

运行结果如下:

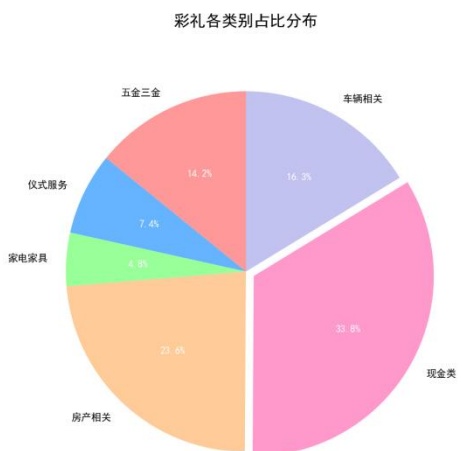
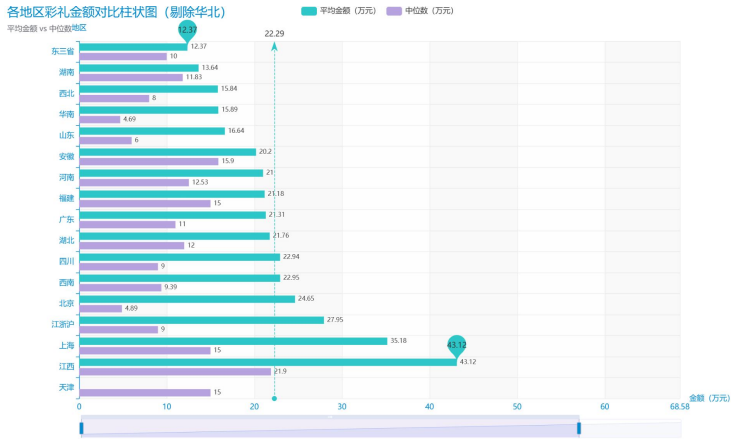


表 11 map.py 和 cat.py 运行结果

观察数据发现, 华北 510000.96 明显存在数据误差, 从数据统计中剔除。剔除华北数据后, 彩礼金额的地域梯度更明确: 江西、天津、上海构成第一阵营, 江浙沪及中西部部分省份形成中高值区, 东北及华南、西北为低值区, 总体呈现“东部沿海偏高、东北偏低、中部江西突出、一线城市分化”的特点, 地域习俗与经济水平仍是核心影响因素。

同时，北京、江浙沪对于彩礼的形式讨论较多，反映出一线城市彩礼不仅局限于现金，同时更加考察车、房等大城市中的稀缺资源，和现实中北京、江浙沪房价高、上车牌难等社会问题相呼应，体现了彩礼在不同地区形式的多样性。

5.2 帖子热度影响因素

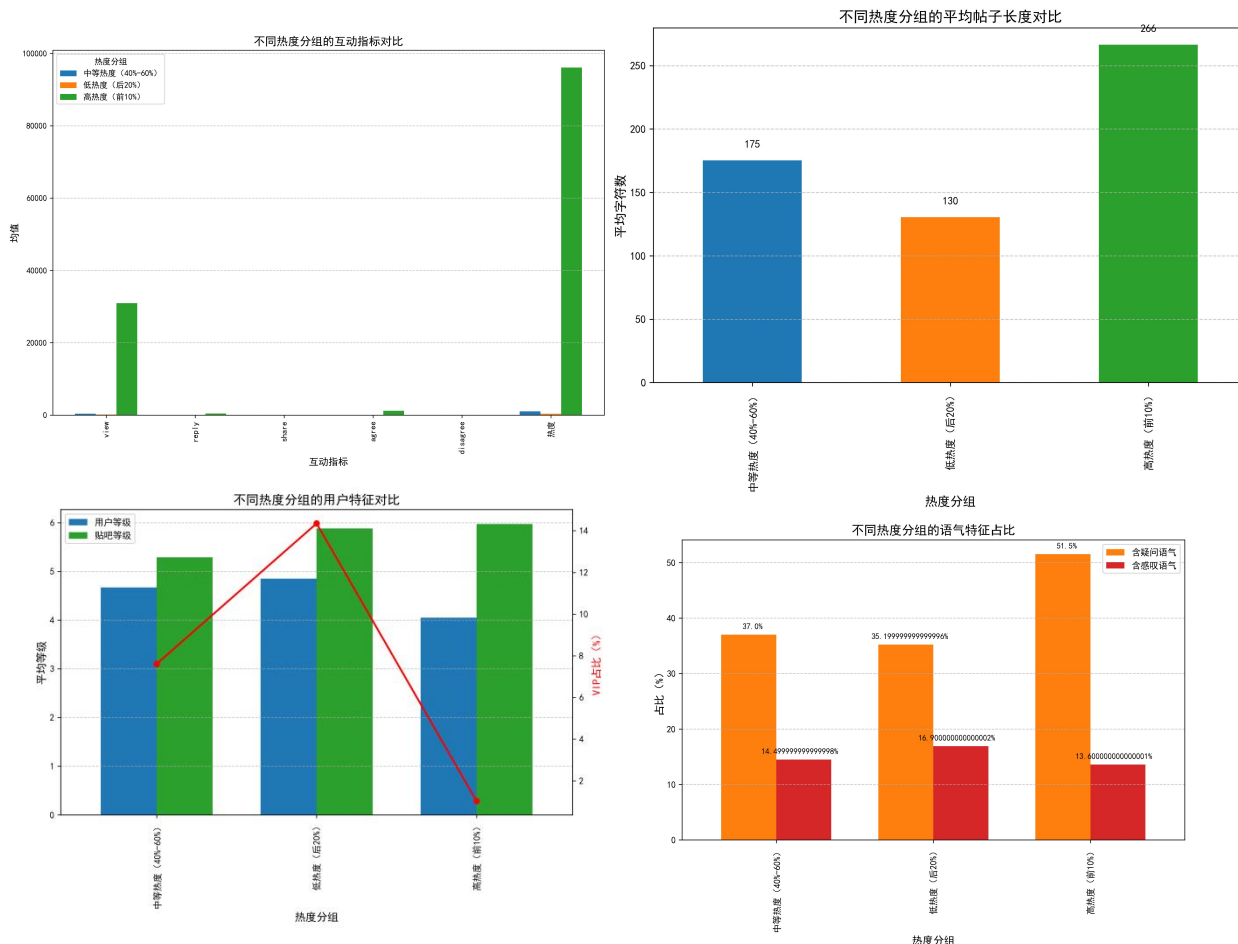
注意到在社区互动中，帖文的关注度存在显著差异，其热度可从零星浏览到上千互动不等。这种差异通常与话题的时效性、内容共鸣度等因素紧密相关，呈现出鲜明的传播规律。接下来将通过数据分析方法，探究影响帖子热度的关键因素及其作用模式。

提示词：相关性分析：计算帖子长度、内容语气、互动指标、用户特征与综合热度的相关性，判断哪类因素对“高热度”贡献最大。

大模型回复：

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import re
5 from scipy.stats import pearsonr
6
7 # 设置中文字体
8 plt.rcParams['font.sans-serif'] = ['SimHei']
9 plt.rcParams['axes.unicode_minus'] = False
10
11 # 1. 读取数据并预处理
12 df = pd.read_csv('./result/彩礼.csv')
13 df = df.fillna({
14     'view': 0, 'reply': 0, 'share': 0, 'agree': 0, 'disagree': 0,
15     'text': '', 'title': ''
16 })
17
18 # 计算综合热度
19 df['热度'] = df['view'] * 3 + df['reply'] * 5 + df['share'] * 3 + df['agree'] + df['disagree']
20
21 # 2. 按热度分位数分组
22 # 计算分位数的阈值
23 q10 = df['热度'].quantile(0.9) # 前10%高热度阈值
24 q40 = df['热度'].quantile(0.4) # 中等热度下限
25 q60 = df['热度'].quantile(0.6) # 中等热度上限
26 q20 = df['热度'].quantile(0.2) # 后20%低热度阈值
27
28 # 定义分组函数
29 def heat_group(heat):
30     if heat >= q10:
31         return '高热度 (前10%)'
32     elif q40 <= heat < q60:
33         return '中等热度 (40%-60%)'
34     elif heat <= q20:
35         return '低热度 (后20%)'
36     else:
37         return '其他'
38
39 df['热度分组'] = df['热度'].apply(heat_group)
40
41 # 只保留三组有效数据
42 df_grouped = df[df['热度分组'].isin(['高热度 (前10%)', '中等热度 (40%-60%)', '低热度 (后20%)'])]
43
44 # 3. 各组核心指标对比
45 # -----
46 # 3.1 互动指标均值对比
47 interaction_cols = ['view', 'reply', 'share', 'agree', 'disagree', '热度']
48 group_interaction = df_grouped.groupby('热度分组')[interaction_cols].mean().round(2)
49 print("==== 不同热度分组的互动指标均值对比 ===")
50 print(group_interaction)
51
52 # 3.2 帖子长度对比
53 fig, ax = plt.subplots(figsize=(12, 8))
54 group_interaction.T.plot(kind='bar', ax=ax, width=0.7)
55 ax.set_title('不同热度分组的互动指标对比', fontsize=14)
56 ax.set_xlabel('互动指标', fontsize=12)
57 ax.set_ylabel('均值', fontsize=12)
58 ax.legend(title='热度分组', fontsize=10)
59 ax.grid(axis='y', linestyle='--', alpha=0.7)
60 plt.savefig('./result/热度分组互动指标对比.png', dpi=300, bbox_inches='tight')
61 plt.close()
62
63 # 3.3 帖子长度对比
64 df_grouped['帖子长度'] = df_grouped['text'].astype(str).agg(lambda x: len(x))
65 group_length = df_grouped.groupby('热度分组')['帖子长度'].agg(['mean', 'median']).round(2)
66 print("==== 不同热度分组的帖子长度对比 ===")
67 print(group_length)
68
69 # 可视化帖子长度对比
70 fig, ax = plt.subplots(figsize=(10, 6))
71 group_length['mean'].plot(kind='bar', color=['#1f77b4', '#ff7f0e', '#2ca02c'], ax=ax)
72 ax.set_title('不同热度分组的平均帖子长度对比', fontsize=14)
73 ax.set_xlabel('热度分组', fontsize=12)
74 ax.set_ylabel('平均字符数', fontsize=12)
75 ax.grid(axis='y', linestyle='--', alpha=0.7)
76 for i, v in enumerate(group_length['mean']):
77     ax.text(i, v * 10, f'{int(v)}', ha='center', fontsize=10)
78 plt.savefig('./result/热度分组帖子长度对比.png', dpi=300, bbox_inches='tight')
79 plt.close()
80
81 # 3.3 语气特征对比 (疑问/感叹语气占比)
82 # 定义语气判断函数
83 def has_question(text):
84     return 1 if re.search(r'[\?]', str(text)) else 0
85
86 def has_exclamation(text):
87     return 1 if re.search(r'[\!]', str(text)) else 0
88
89 # 计算语气特征
90 df_grouped['含疑问语气'] = (df_grouped['title'] + df_grouped['text']).astype(str).apply(has_question)
91 df_grouped['含感叹语气'] = (df_grouped['title'] + df_grouped['text']).astype(str).apply(has_exclamation)
92
93 # 统计含语气占比
94 group_tone = df_grouped.groupby('热度分组')[['含疑问语气', '含感叹语气']].mean().round(3) * 100
95 print("==== 不同热度分组的语气特征占比 (%) ===")
96 print(group_tone)
97
98 # 可视化语气特征对比
99 fig, ax = plt.subplots(figsize=(10, 6))
100 group_tone.plot(kind='bar', ax=ax, width=0.7, color=['#ff7f0e', '#d62728'])
101 ax.set_title('不同热度分组的语气特征占比', fontsize=14)
102 ax.set_xlabel('热度分组', fontsize=12)
103 ax.set_ylabel('占比 (%)', fontsize=12)
104 ax.legend(['含疑问语气', '含感叹语气'], fontsize=10)
105 ax.grid(axis='y', linestyle='--', alpha=0.7)
106 for i, col in enumerate(group_tone.columns):
107     for j, v in enumerate(group_tone[col]):
108         ax.text(j + (i-0.5)*0.5, v * 1, f'{v}%', ha='center', fontsize=9)
109 plt.savefig('./result/热度分组语气特征对比.png', dpi=300, bbox_inches='tight')
110 plt.close()
111
112 # 3.4 用户特征对比 (等级、性别、VIP)
113 user_cols = ['level', 'glevel', 'is_vip']
114 group_user = df_grouped.groupby('热度分组')[user_cols].agg({
115     'level': 'mean',
116     'glevel': 'mean',
117     'is_vip': lambda x: (x == 1).mean() * 100 # VIP占比
118 }).round(2)
119
120 group_user.rename(columns={'is_vip': 'VIP占比 (%)'}, inplace=True)
121 print("==== 不同热度分组的用户特征对比 ===")
122 print(group_user)
123
124 # 可视化用户特征对比
125 fig, ax = plt.subplots(figsize=(10, 6))
126 group_user[['level', 'glevel']].plot(kind='bar', ax=ax, width=0.7, color=['#1f77b4', '#2ca02c'])
127 ax2 = ax.twinx()
128 group_user['VIP占比 (%)'].plot(kind='line', ax=ax2, marker='o', color='red', linewidth=2)
129 ax.set_title('不同热度分组的用户特征对比', fontsize=14)
130 ax.set_xlabel('热度分组', fontsize=12)
131 ax.set_ylabel('平均等级', fontsize=12)
132 ax2.set_ylabel('VIP占比 (%)', fontsize=12, color='red')
133 ax.legend(['用户等级', '性别等级'], fontsize=10)
134 ax.grid(axis='y', linestyle='--', alpha=0.7)
135 plt.savefig('./result/热度分组用户特征对比.png', dpi=300, bbox_inches='tight')
136 plt.close()
137
138 # -----
139 # 4. 输出对比结果到CSV
140 # -----
141 # 合并所有对比结果
142 comparison_result = pd.concat([
143     group_interaction.add_prefix('互动_'),
144     group_length.add_prefix('长度_'),
145     group_tone.add_prefix('语气_'),
146     group_user.add_prefix('用户_')
147 ], axis=1)
148
149 comparison_result.to_csv('./result/热度分组对比分析.csv', encoding='utf-8-sig')
150 print("\n热度分组对比分析结果已保存至 ./result/热度分组对比分析.csv")
```

输出内容：



可以发现，前 20%热度的帖子热度远高于中热度、地热度帖子，说明热度集中度极高，网友热衷于在相同帖子下互动；高热度帖子的浏览量(view)和点赞（agree）明显高于另外两组，说明浏览量与点赞量对于帖子热度贡献高；高中-低热度帖子长度依次下降，说明长帖更容易获得高热度，这可能是由于长帖子一般是分享个人经历、分析个人观点、讲述个人故事等内容，具有吸引力；高热度帖子帖主用户等级低、贴吧等级高，说明其长期活跃于彩礼吧，输出内容更容易收获高热度；在三组中，疑问语气都多于感叹语气，高热图帖子尤为明显，说明寻求建议、帮助等发问类型的帖子更容易获得高热度。

六、文本内容理解与生成

在统计分析的基础上，本节转向对具体文本内容的理解与生成。

6.1 大模型文本模拟生成：一则反对彩礼的贴文

注意到当前社会讨论中，彩礼话题持续引发热议。彩礼习俗在我国社会语境中承载着复杂意涵，其存废之争涉及代际观念、性别平等、婚姻本质等多重维度，折射出传统与现代价值观的深刻碰撞。接下来通过大模型生成模拟文本，展现反对彩礼立场的主要论点与表达方式。

提示词：“‘彩礼’消失了几十年，这个陋习是什么时候……又指望它长大成忠犬？”这是一些贴文，模仿这些贴文写一条类似的、反对彩礼的贴文。

大模型输出：

你爸妈养你辛苦，凭什么要让刚跟你组建家庭的男方来买单？他爸妈把他从牙牙学语养到能独立赚钱，难道就轻松了？合着就你家父母的付出金贵，人家的付出就该白搭？

还有人说“要彩礼是为了给爸妈留养老钱”，这话更可笑。真有孝心，婚后自己多努力赚钱给爸妈买东西、陪他们说话，比拿着别人的钱充大方强百倍。靠要彩礼“尽孝”，本质上就是把自己该尽的责任，转嫁到了伴侣身上，说难听点，这跟“拿别人的钱孝敬自己爹妈”有啥区别？

更别说有些要完彩礼还说“这是传统”的，咋不想想传统里还有“嫁妆跟彩礼对等”“女方出嫁也带资产补贴小家庭”呢？只捡对自己有利的传统提，不利的全当没看见，这不就是双标吗？

说到底，彩礼从来不是什么“孝心证明”，也不是“婚姻保障”。真正想好好过日子的人，只会跟伴侣一起规划未来，而不是把彩礼当成结婚的“入场费”。那些把彩礼挂嘴边的，与其说是为了爸妈，不如说是自己既不想承担责任，又想占便宜罢了。

这条生成文本说明大模型并不仅仅能写代码、画图，还可以学习特定论坛的语言风格，模拟真实用户的发言方式；从内容上，它也代表了彩礼讨论中一类典型立场：质疑“拿别人的钱孝敬自己父母”；反对把彩礼当作道德绑架工具；倡导把重心放在两个人共同承担责任、规划生活上。

七、总结

本次大作业围绕“男女对彩礼的看法”这一主题，从百度贴吧“彩礼吧”以及相关文本出发，完成了从数据获取、预处理、描述性统计到文本分析与生成的完整流程，主要收获可以概括为两部分：

7.1 关于彩礼话题的发现

1. **讨论重心多维而复杂：**高频词表明，用户既关心金额与物质条件，又反复提到父母、补偿、公平、安全感等更深层次议题。
2. **性别与话语风格存在差异：**不同性别在彩礼话题中发帖的比例不同，表达风格也有偏向。
3. **地域因素显著：**从文本中抽取的信息显示，不同地区对“合理彩礼”的期待值差异较大，彩礼的具体实现形式差异也较大，体现出不同地域的经济社会文化特征多元性。

7.2 方法与技能上的收获

1. 学习了使用 pandas 对 CSV 数据进行读取、清洗、字段筛选与派生变量构造的基本流程。
2. 掌握了使用 matplotlib / wordcloud 等工具进行可视化，包括柱状图、饼图、折线图与词云。
3. 通过向大模型提问，学会了如何把自然语言需求转化为可执行的 Python 代码，在数据清洗、统计与绘图等环节都起到了加速作用。
4. 初步尝试了基于规则的语气分析与大模型文本生成，体会到传统数据分析与生成式 AI 结合的可能性。

总体而言，这次大作业不仅让我对“彩礼”这一社会热点问题有了更加立体的认识，也帮助我打通了“从提出问题→获取数据→分析与可视化→解释结果→用语言总结”的完整数据分析链路，为之后更复杂的项目打下了基础。