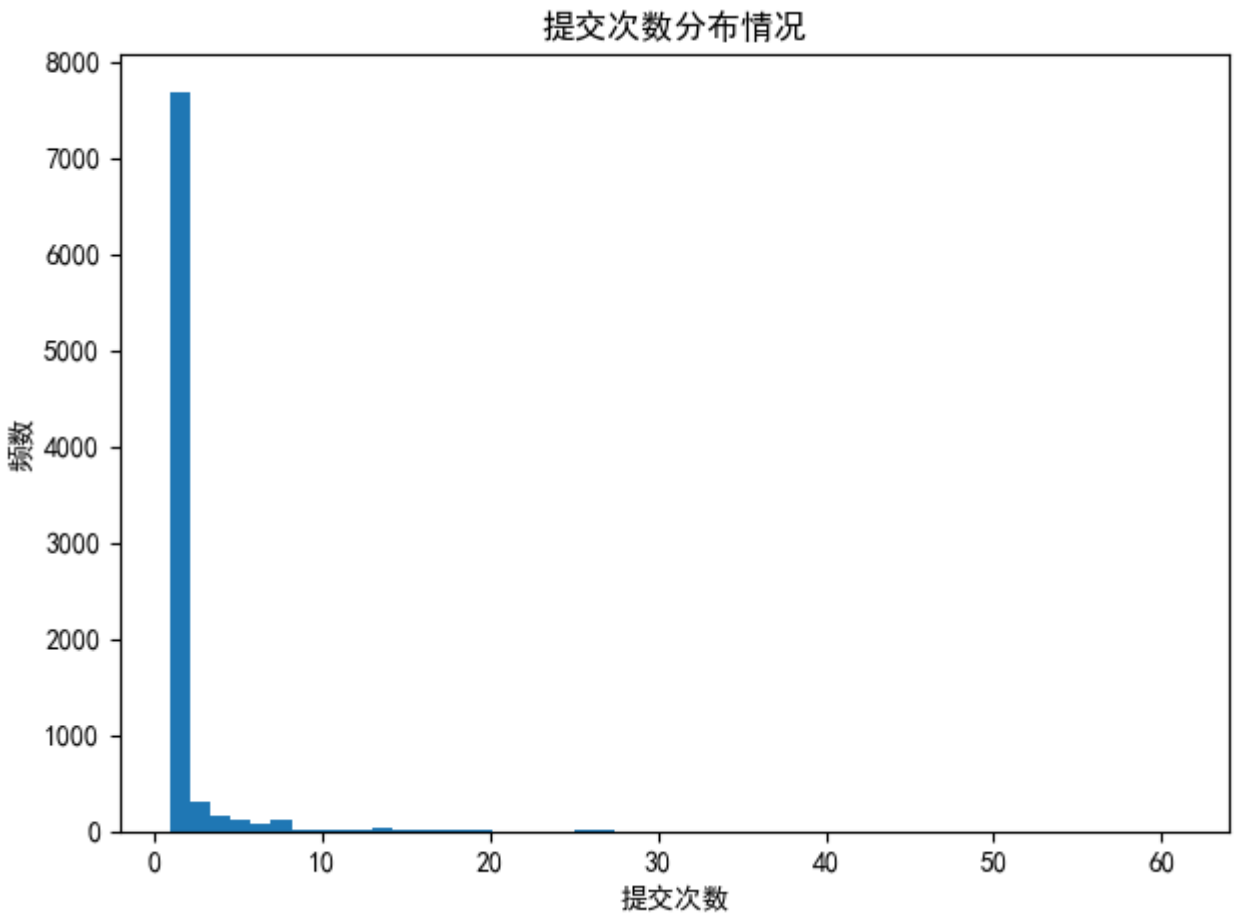
线性表、字符串(dim2)分析

1.提交次数

结束了第一阶段的数据读取，并做了相应的数据预处理和数据过滤后，dim2有效样本数目为8658份，其中只提交过一次的样本数目为6846，占比79.1%。



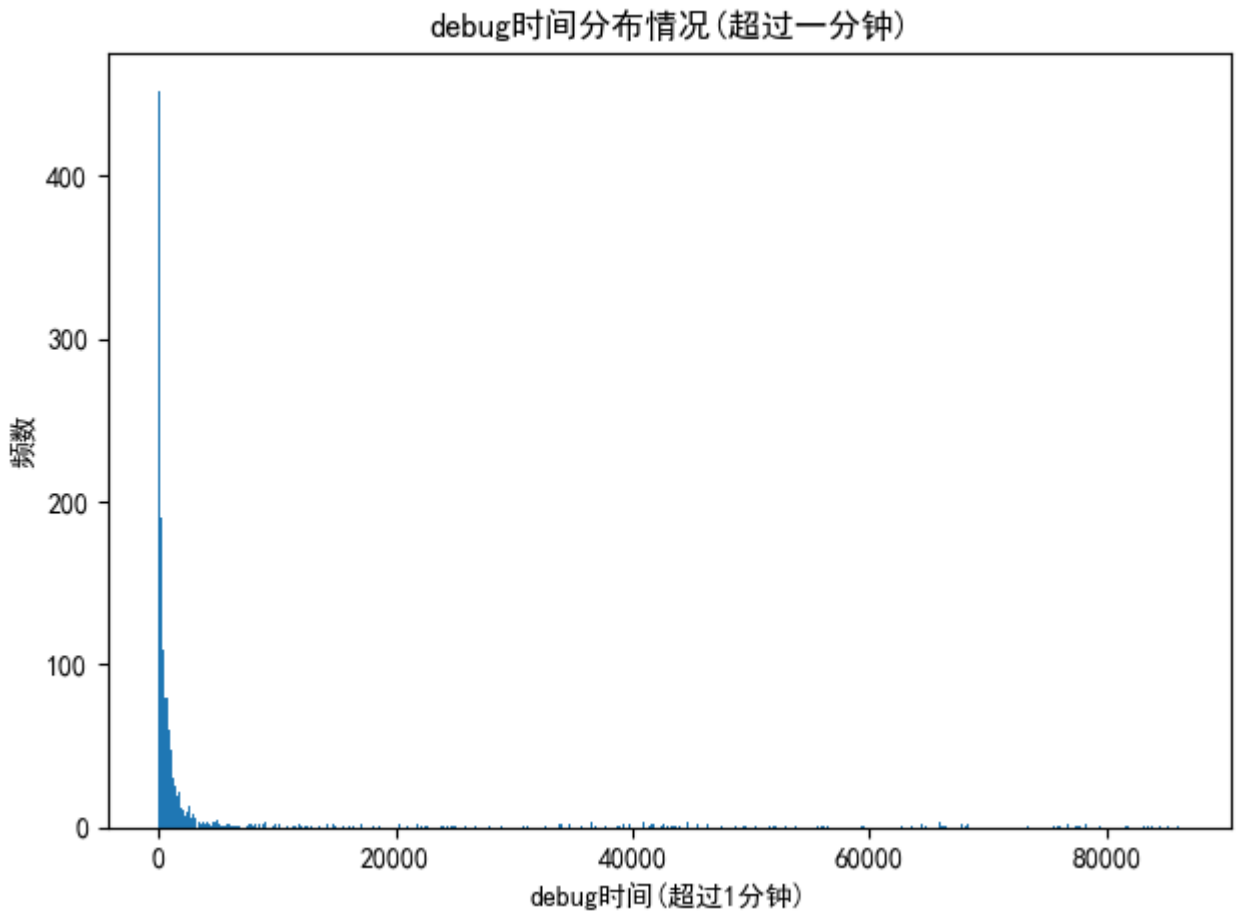
提交次数统计

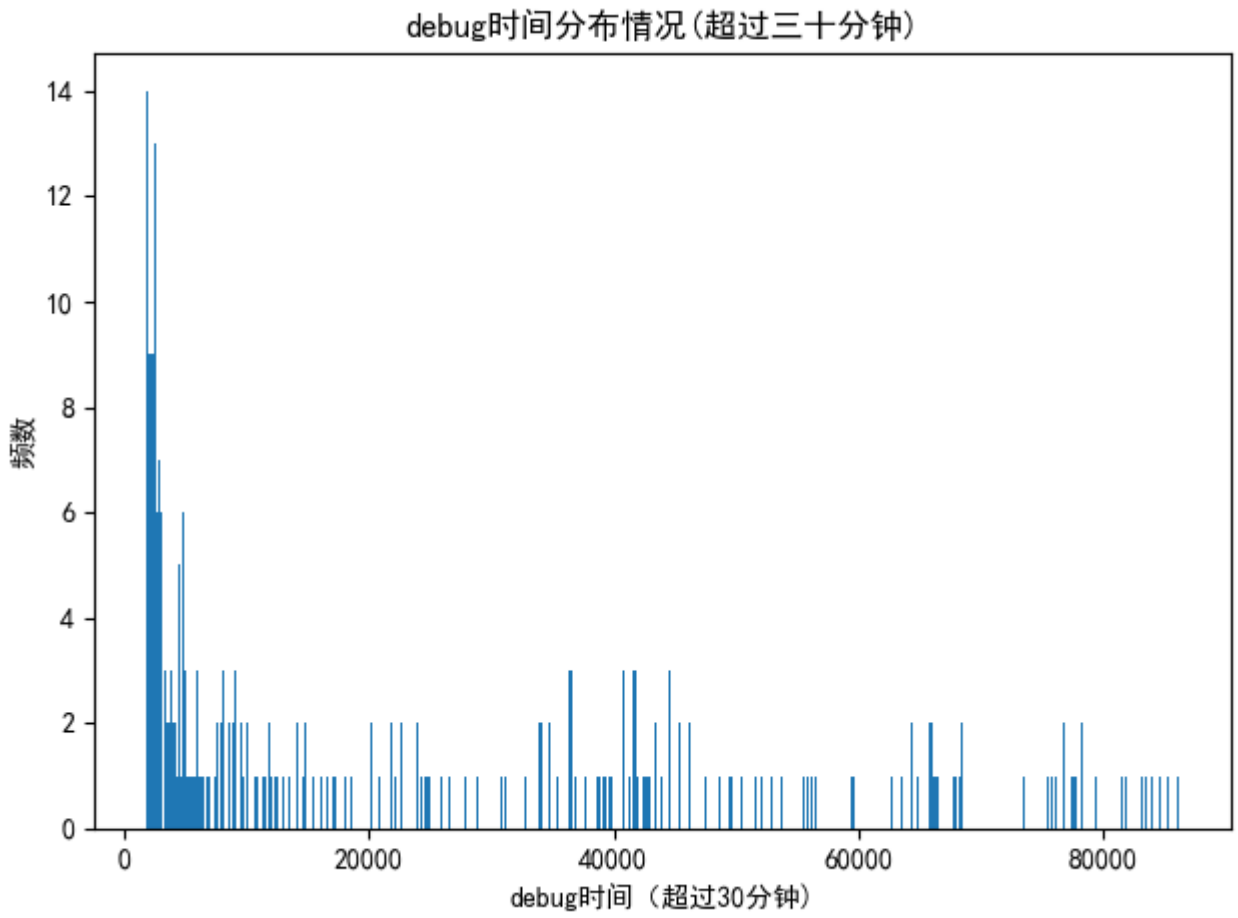
2.debug时间

> 注：最后一次提交时间-第一次提交时间，如果最后一次提交之前获得满分，则将首次获得满分的时间作为被减数

由于在提交次数的分析中，有79.1%的样本只提交过一次，结合大多数同学的编程习惯（先在本地编写调试），在debug时间的分析中，我们考虑时间超过1分钟的样本和超过30分钟的样本。

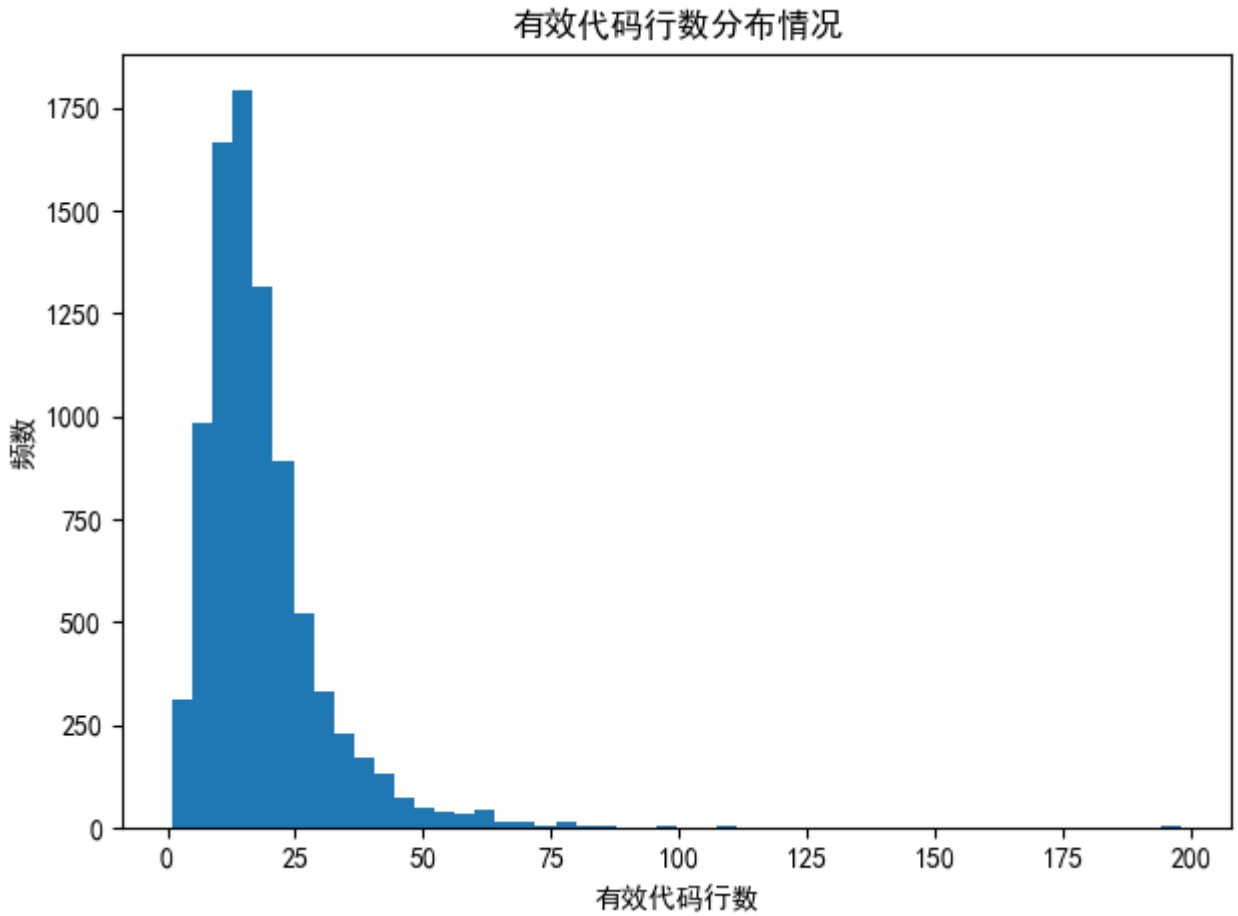
其中debug时间超过1分钟的样本数为1306，debug时间超过30分钟的样本数为269


debug时间分布情况(超过一分钟)
频数
0
100
200
300
400
0
20000
40000
60000
80000
debug时间(超过1分钟)


debug时间分布情况(超过三十分钟)
频数
0
2
4
6
8
10
12
14
0
20000
40000
60000
80000
debug时间（超过30分钟）

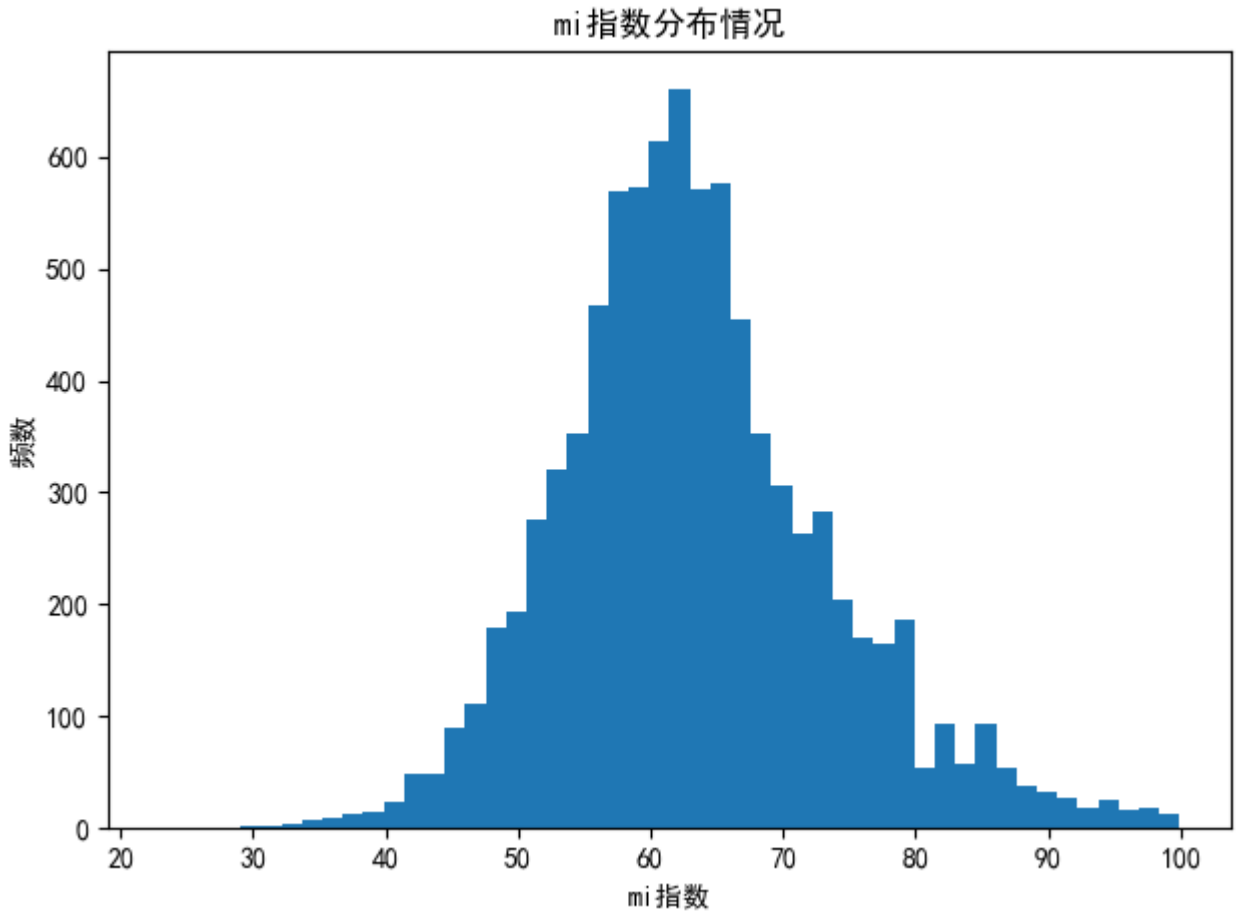
3.代码有效行数

大多数样本中，代码有效行数小于50行。有效代码行数小于50行的样本数为8435，这与线性表和字符串的题型相对简单有关。



4.mi指数

样本中mi指数的分布趋近正态分布



5.总结

从提交次数和debug时间的分析结果来看，大多数样本首次提交便取得了满分，结合自己编程时经验，此类样本应该是先在本地有过测试和debug过程。因此，我们在最终确定样本dim2得分时，只采用了debug时间、代码有效行数和mi指数作为判断依据，而且debug时间占比相对较少。

debug得分

在debug时间的分析中，我们得知debug时间超过30分钟的仅占3.1%，再结合个人编程习惯的差异，我们决定将debug时间小于30分钟的样本归为编程习惯较好、debug能力较强一类;debug时间超过30分钟的，每一分钟扣除0.01分,超过130分钟的计为0分。具体计算公式如下:

用户做的n题中第i题的debug时间为 :DT_i (单位:秒)

用户做的n题中第i题的debug得分为:

$$DTS_i = \begin{cases} 1, DT_i < 30 * 60 \\ 1 - \frac{DT_i - 30*60}{60} * 0.01, 130 * 60 \geq DT_i \geq 30 * 60 \\ 0, DT_i > 130 * 60 \end{cases}$$

用户的debug得分为

$$DT_{dim2}^{score} = \frac{\sum_{i=1}^{n} DTS_i}{n}$$

有效行数得分

计算有效行数得分时，是与整体情况做了参照，具体计算公式如下:

所有样本的代码行数均值为:VL_{avg}

用户做的n题中第i题的有效行数为:VL_i

用户的有效行数得分为

$$VL_{dim2}^{score} = \frac{\sum_{i=1}^{n} \frac{VL_{avg}}{VL_i}}{n}$$

mi指数得分

因为在可视化分析时，mi指数分布是最接近正态分布的，因此在设计公式时，将mi指数的影响做了一些强化，具体公式如下：

所有样本的dim2中mi指数均值记为：MI_{avg}

用户做的n题中第i题的mi指数为: MI_i

用户的mi指数得分为:

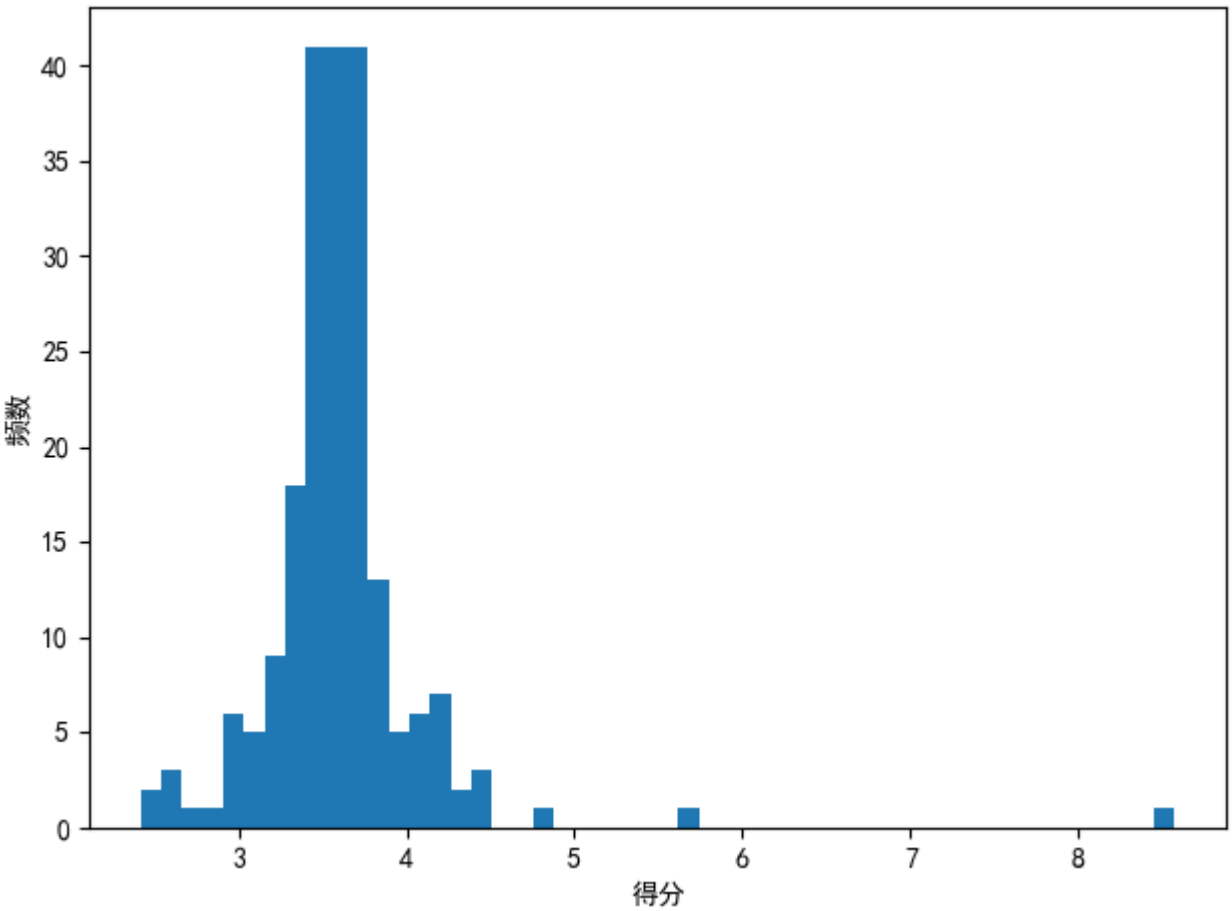
$$MI_{dim2}^{score} = \frac{\sum_{i=1}^{n} (\frac{MI_i}{MI_{avg}})^2}{n}$$

用户最终得分

因为考虑到debug时间的参考价值较小，而mi指数的标准更客观，因此最终的得分公式如下:

$$S_{dim2}^{score} = 0.5 * DT_{dim2}^{score} + VL_{dim2}^{score} + 1.5 * MI_{dim2}^{score}$$

最终得分分布如下图所示:


40
35
30
25
20
15
10
5
0
频数
3
4
5
6
7
8
得分