数据挖掘第四周作业

选用数据集Movies Dataset from Pirated Sites

1120193285 张秋雨 计算机学院计算机科学与技术专业07111908班

github地址：https://github.com/BIT-QiuYu/DM_homework_week4 (https://github.com/BIT-QiuYu/DM_homework_week4)

数据概览

In [1]:

```
import pandas as pd
import numpy as np
import movies_dataset
```

In [2]:

```
movies_dataset.show_col()
```

Out[2]:

```
Index(['Unnamed: 0', 'IMDb-rating', 'appropriate_for', 'director', 'downloads',
       'id', 'industry', 'language', 'posted_date', 'release_date', 'run_time',
       'storyline', 'title', 'views', 'writer'],
      dtype='object')
```

每行数据包括15个列，分别代表：

- **Unnamed: 0** 序号
- **IMDb-rating** 电影评分
- **appropriate_for** 电影分级
- **director** 导演
- **downloads** 电影下载次数
- **id** 电影的id
- **industry** 电影的出品公司
- **language** 电影使用的语言
- **posted_date** 电影的发布日期
- **release_date** 首次上映日期
- **run_time** 电影时长
- **storyline** 电影的主要故事
- **title** 电影标题
- **views** 观看次数
- **writer** 编剧

接下来的内容将对以上属性中的数值属性，有意义的非唯一标称属性，以及可推导获得的有意义的属性进行数据分析与预处理。

In [3]:

```
helper = movies_dataset.col_helper(1)
```

1 序号 (无缺失值)

In [4]:

```
helper.select_col('Unnamed: 0')
# 缺失值个数
n_b = helper.count_none()
```

0

没有缺失值

2 IMDb-rating (有缺失值)

In [5]:

```
helper.select_col('IMDb-rating')
```

In [6]:

```
helper.count_none('')
```

841

Out[6]:

841

有841个缺失值，这里猜测，由于盗版电影网站不一定会那么全面的数据，因此猜测是数据不全造成的

这里可以进行缺失值的处理方式有以下几种：

删除所有缺失数据

将缺失数据用中位数或平均数值代替

利用数据对象的相关性进行填补

利用属性的相关性进行填补

数据摘要

In [7]:

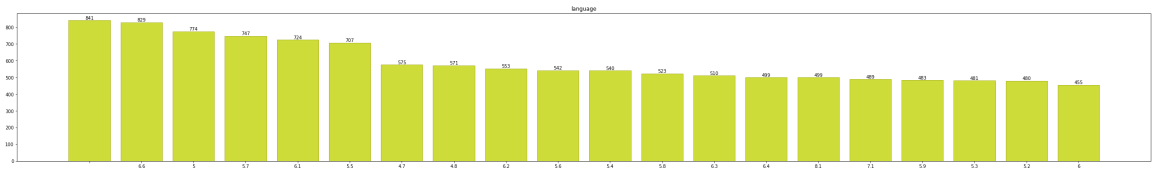
```
helper.data2['IMDb-rating'].value_counts()
```

Out[7]:

```
       841
6.6    829
5      774
5.7    747
6.1    724
      ...
1.5      2
9.1      1
9.9      1
9.5      1
1.3      1
Name: IMDb-rating, Length: 86, dtype: int64
```

In [8]:

```
helper.hist_bar(45,6,20)
```



In [9]:

```
index11,row11 = movies_dataset.get_row_index('IMDb-rating','')
print(row11)
```

```
       8296.00 ',    ], dtype=object), array([472,    ,    ,    ,  3370.00 ', 3685
89, 'Wrestling', 'English',
       '03 Dec, 2022', 'Dec 02 2022', '', '', 'WWE Smackdown 2022-12-02',
       '8853.00 ', ''], dtype=object), array([490, '', '', '', '215.00 ', 36853
8, 'Wrestling', 'English',
       '02 Dec, 2022', 'Dec 01 2022', '', '', 'TNA.Impact 2022-12-01',
       '977.00 ', ''], dtype=object), array([507, '', '', '', '3338.00 ', 36839
1, 'Wrestling', 'English',
       '29 Nov, 2022', 'Nov 28 2022', '', '', 'WWE Raw 2022-11-28',
       '8155.00 ', ''], dtype=object), array([518, '', '', '', '2847.00 ', 3682
89, 'Wrestling', '',
       '26 Nov, 2022', 'Nov 25 2022', '', '', 'WWE Smackdown 2022-11-26',
       '7356.00 ', ''], dtype=object), array([536, '', '', '', '206.00 ', 36823
6, 'Wrestling', 'English',
       '25 Nov, 2022', 'Nov 24 2022', '', '', 'TNA.Impact 2022-11-24',
       '915.00 ', ''], dtype=object), array([539, '', '', '', '761.00 ', 36819
8, 'Hollywood / English',
       'English,Hindi', '24 Nov, 2022', 'Nov 23 2022', '',
       'After\r\n inheriting a farm at Christmas time, a widowed father makes a
bumpy \r\nadjustment to village life - while his kids hatch a plan to stay ther
```

In [10]:

```
movies_dataset.delete_row(index11,2)
```

In [11]:

```
# 删除后缺失值个数
helper.count_none_after()
```

0

Out[11]:

0

数据分布

In [12]:

```
helper2 = movies_dataset.col_helper(2)
```

In [13]:

```
helper2.select_col('IMDb-rating')
```

In [14]:

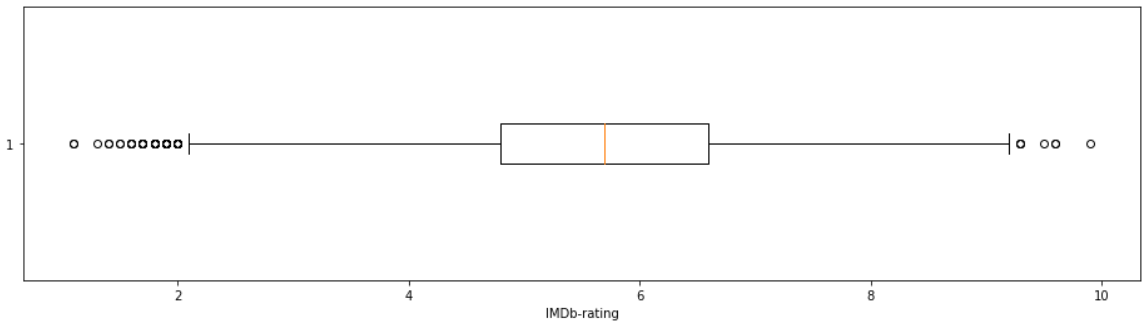
```
# 五数概括
helper2.five_number()
```

Min: 1.1
Q1: 4.8
Q2: 5.7
Q3: 6.6
Max: 9.9

In [15]:

```
# 盒图
helper2.box(16,4,'IMDb-rating')
```

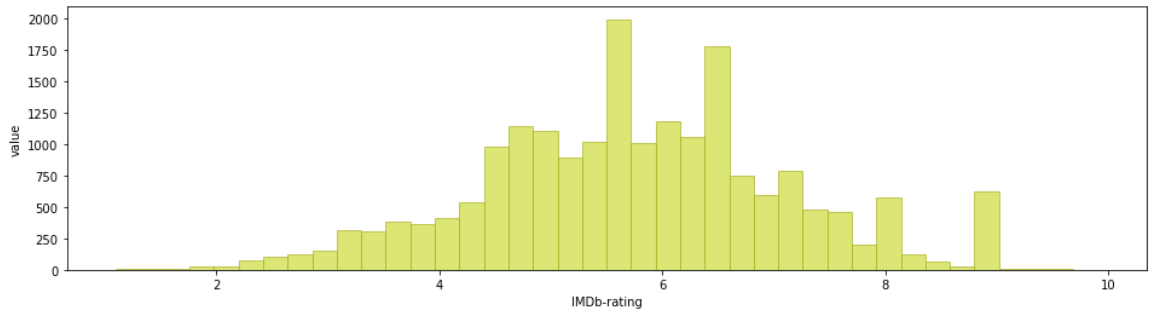
[4.8 6.4 5.2 ... 7.2 7.7 8.]



9.299999999999999

In [16]:

```
# 直方图
helper2.normal_hist(16,4,'IMDb-rating','value')
```



从盒图和直方图可以看出，大多数的电影评分接近正态分布，符合一般性规律

3 appropriate_for (有缺失值)

In [17]:

```
helper.select_col('appropriate_for')
```

In [18]:

```
helper.count_none('')
```

9476

Out[18]:

9476

有9476个缺失值，这里猜测，由于盗版电影网站不一定会那么全面的数据，因此猜测是数据不全造成的

这里可以进行缺失值的处理方式有以下几种：

删除所有缺失数据

将缺失数据用中位数或平均数值代替

利用数据对象的相关性进行填补

利用属性的相关性进行填补

In [19]:

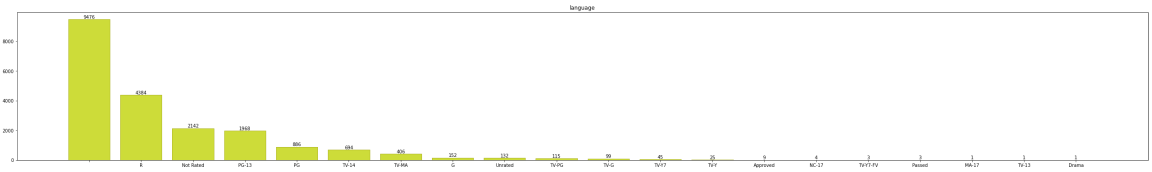
```
helper.data2['appropriate_for'].value_counts()
```

Out[19]:

```
                9476
R               4384
Not Rated       2142
PG-13           1968
PG               886
TV-14            694
TV-MA            406
G                152
Unrated          132
TV-PG            115
TV-G              99
TV-Y7             45
TV-Y              25
Approved           9
NC-17              4
TV-Y7-FV           3
Passed             3
MA-17              1
TV-13              1
Drama              1
Drama, Romance     1
18+                1
Name: appropriate_for, dtype: int64
```

In [20]:

```
helper.hist_bar(45,6,20)
```



In [21]:

```
index11,row11 = movies_dataset.get_row_index('appropriate_for','')
print(index11)
```

```
7, 2388, 2391, 2393, 2394, 2397, 2401, 2402, 2403, 2408, 2410, 2411, 2413, 241
4, 2415, 2417, 2419, 2420, 2421, 2422, 2423, 2425, 2426, 2427, 2428, 2429, 243
1, 2432, 2433, 2434, 2435, 2437, 2438, 2440, 2442, 2445, 2446, 2447, 2448, 245
2, 2453, 2457, 2458, 2459, 2461, 2463, 2464, 2466, 2468, 2471, 2472, 2473, 247
5, 2476, 2477, 2478, 2479, 2481, 2483, 2484, 2487, 2488, 2489, 2494, 2496, 249
8, 2499, 2501, 2502, 2503, 2505, 2507, 2508, 2509, 2510, 2512, 2514, 2516, 251
7, 2519, 2523, 2525, 2526, 2527, 2529, 2530, 2533, 2536, 2537, 2538, 2539, 254
2, 2545, 2546, 2547, 2549, 2551, 2553, 2556, 2557, 2558, 2559, 2560, 2561, 256
2, 2563, 2564, 2566, 2570, 2573, 2576, 2577, 2579, 2580, 2582, 2585, 2587, 258
9, 2590, 2591, 2593, 2595, 2597, 2598, 2599, 2601, 2602, 2603, 2605, 2606, 260
7, 2612, 2613, 2619, 2623, 2624, 2626, 2627, 2628, 2630, 2631, 2633, 2634, 263
5, 2637, 2639, 2641, 2642, 2645, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 265
8, 2661, 2662, 2664, 2666, 2667, 2672, 2673, 2675, 2677, 2678, 2679, 2681, 268
3, 2685, 2690, 2691, 2692, 2693, 2694, 2696, 2697, 2698, 2700, 2704, 2708, 270
9, 2710, 2711, 2713, 2714, 2715, 2721, 2722, 2723, 2725, 2727, 2728, 2729, 273
1, 2733, 2734, 2735, 2736, 2737, 2739, 2741, 2744, 2745, 2746, 2747, 2748, 274
9, 2751, 2753, 2755, 2756, 2757, 2758, 2760, 2761, 2765, 2766, 2767, 2769, 277
1, 2780, 2784, 2785, 2791, 2793, 2794, 2795, 2796, 2797, 2798, 2800, 2801, 280
2, 2803, 2807, 2809, 2810, 2811, 2813, 2815, 2816, 2821, 2822, 2823, 2828, 283
0, 2831, 2835, 2836, 2837, 2838, 2840, 2841, 2843, 2846, 2849, 2850, 2851, 285
```

In [22]:

```
movies_dataset.delete_row(index11,3)
```

In [23]:

```
# 删除后缺失值个数
helper.count_none_after()
```

```
0
```

Out[23]:

```
0
```

4 其余非数值属性

director 导演

id 电影编号

industry 电影的出品公司

language 电影使用的语言

posted_date 电影的发布日期

release_date 首次上映日期

storyline 电影的主要故事

title 电影标题

writer 编剧

处理源根同电影分级

In [24]:

```
helper.select_col('director')
```

In [25]:

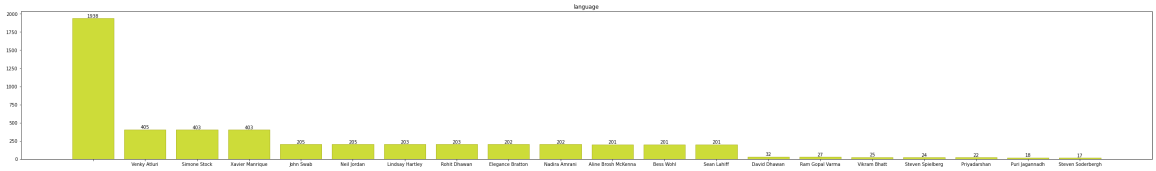
```
helper.data2['director'].value_counts()
```

Out[25]:

```
                  1938
Venky Atluri       405
Simone Stock       403
Xavier Manrique    403
John Swab          205
                  ...
David G. Evans       1
Theresa Rebeck       1
Mark Grentell        1
Nick Searcy          1
Becca Gleason        1
Name: director, Length: 9673, dtype: int64
```

In [26]:

```
helper.hist_bar(45,6,20)
```



In [27]:

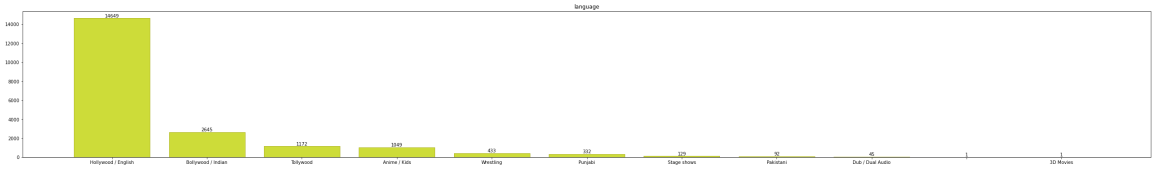
```
helper.select_col('industry')
helper.data2['industry'].value_counts()
```

Out[27]:

```
Hollywood / English    14649
Bollywood / Indian      2645
Tollywood               1172
Anime / Kids            1049
Wrestling                433
Punjabi                  332
Stage shows              129
Pakistani                 92
Dub / Dual Audio          45
                           1
3D Movies                  1
Name: industry, dtype: int64
```

In [28]:

```
helper.hist_bar(45,6,20)
```



In [29]:

```
helper.select_col('language')
helper.data2['language'].value_counts()
```

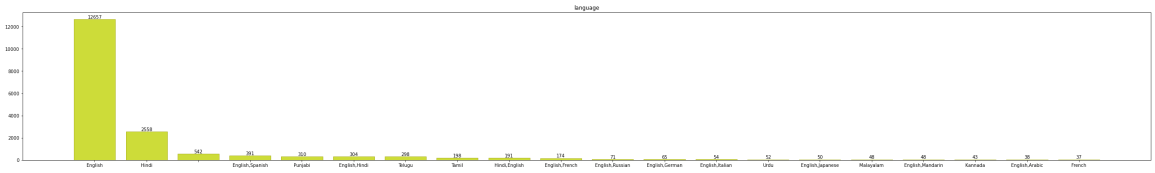
Out[29]:

English	12657
Hindi	2558
	542
English,Spanish	391
Punjabi	310
...	
English,Mandarin,Turkish,Indonesian,Russian	1
English,Polynesian,Spanish	1
English,Cheyenne,French	1
English,American Sign Language,Russian,French	1
Spanish,German,English	1

Name: language, Length: 1169, dtype: int64

In [30]:

```
helper.hist_bar(45,6,20)
```



In [31]:

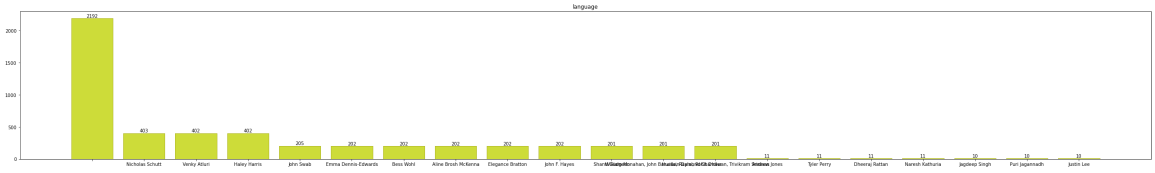
```
helper.select_col('writer')
helper.data2['writer'].value_counts()
```

Out[31]:

```
                                              2192
Nicholas Schutt                                403
Venky Atluri                                   402
Haley Harris                                   402
John Swab                                      205
                                              ...
Barbara Samuels, Joseph Boyden                   1
Maria Allred                                     1
Pia Mechler                                      1
Paul Flannery, David Ryan Keith                  1
Khwaja Ahmad Abbas, Khwaja Ahmad Abbas           1
Name: writer, Length: 13604, dtype: int64
```

In [32]:

```
helper.hist_bar(45,6,20)
```



5 downloads 电影下载次数

In [33]:

```
helper.select_col('downloads')
```

In [34]:

```
helper.count_none('')
```

1

Out[34]:

1

有1个缺失值

In [35]:

```
helper.data2['downloads'].value_counts()
```

Out[35]:

```
75.00          403
622.00         212
378.00         209
1782.00        187
466.00         170
              ...
3721.00          1
13947.00         1
51963.00         1
19225.00         1
3276.00          1
Name: downloads, Length: 10626, dtype: int64
```

In [36]:

```
index11,row11 = movies_dataset.get_row_index('downloads','')
print(index11)
movies_dataset.delete_row(index11,4)
```

```
[149]
```

In [37]:

```
helper4 = movies_dataset.col_helper(4)
helper4.select_col('downloads')
```

In [38]:

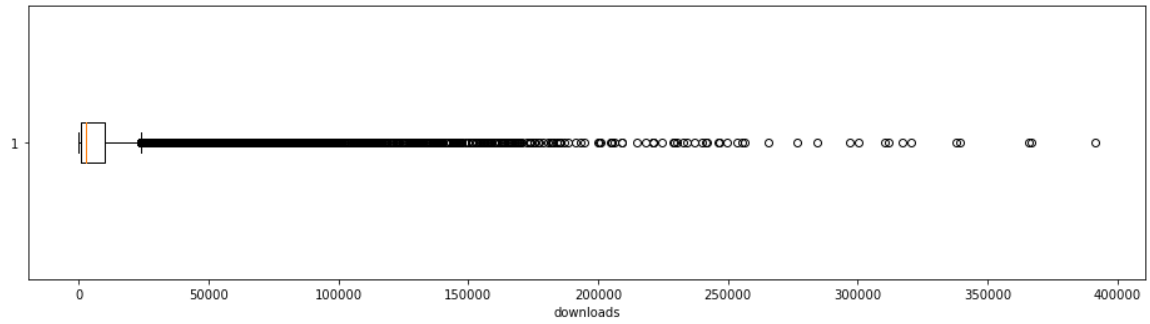
```
# 五数概括
helper4.five_number()
```

```
Min: 0.0
Q1: 855.5
Q2: 2716.0
Q3: 10070.0
Max: 391272.0
```

In [39]:

```
# 盒图
helper4.box2(16,4,'downloads')
```

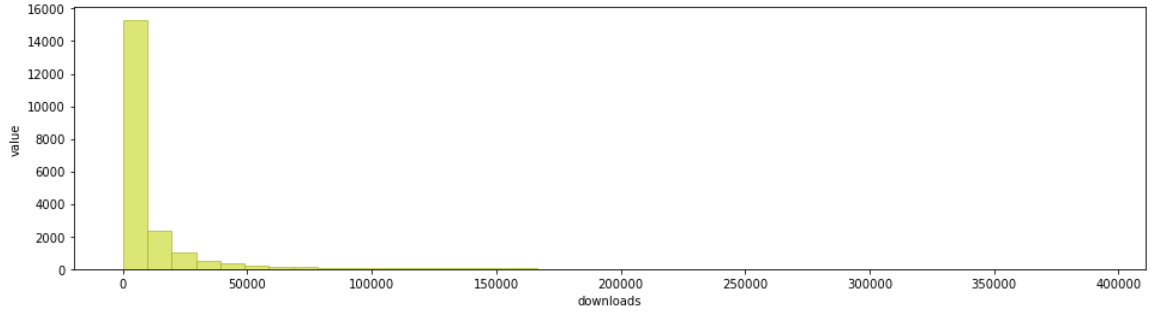
[304. 73. 1427. ... 3276. 309. 2613.]



23891.75

In [40]:

```
# 直方图
helper4.normal_hist(16,4,'downloads','value')
```



从盒图和直方图可以看出，大多数的电影观看下载数适中，符合“盗版网站”的特点。

6 run_time 电影时长

In [41]:

```
helper.select_col('run_time')
```

In [42]:

```
helper.count_none('')
```

1768

Out[42]:

1768

有1768个缺失值，这里猜测，由于盗版电影网站不一定会那么全面的数据，因此猜测是数据不全造成的

这里可以进行缺失值的处理方式有以下几种：

删除所有缺失数据

将缺失数据用中位数或平均数值代替

利用数据对象的相关性进行填补

利用属性的相关性进行填补

In [43]:

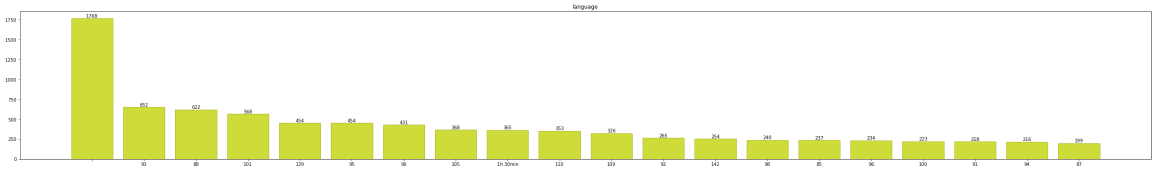
```
helper.data2['run_time'].value_counts()
```

Out[43]:

```
            1768
93           652
88           622
101          568
139          454
            ...
74 min         1
288            1
220            1
49min          1
3h 13min       1
Name: run_time, Length: 416, dtype: int64
```

In [44]:

```
helper.hist_bar(45,6,20)
```



In [45]:

```
index11,row11 = movies_dataset.get_row_index('run_time','')
print(index11)
```

[10, 12, 16, 18, 22, 23, 32, 35, 44, 55, 60, 63, 66, 77, 91, 92, 99, 100, 105, 11
1, 112, 123, 142, 149, 150, 151, 156, 182, 183, 186, 193, 196, 205, 224, 230, 243,
248, 266, 267, 279, 280, 282, 291, 297, 310, 311, 319, 322, 329, 339, 340, 352, 35
6, 367, 368, 371, 374, 375, 376, 381, 396, 399, 400, 403, 413, 418, 424, 443, 453,
457, 462, 465, 472, 490, 507, 510, 514, 518, 529, 530, 536, 539, 540, 541, 543, 54
4, 545, 546, 550, 552, 553, 554, 563, 579, 600, 616, 621, 629, 642, 647, 665, 674,
675, 677, 683, 694, 717, 726, 738, 762, 772, 782, 798, 815, 823, 825, 837, 853, 85
5, 856, 867, 884, 907, 933, 935, 938, 972, 977, 981, 1027, 1030, 1076, 1128, 1193,
1256, 1264, 1290, 1291, 1349, 1390, 1414, 1426, 1506, 1508, 1515, 1523, 1571, 163
2, 1636, 1748, 1762, 1770, 1859, 1868, 1903, 1925, 1952, 1993, 1997, 2031, 2043, 2
046, 2083, 2089, 2102, 2127, 2180, 2288, 2299, 2308, 2323, 2336, 2402, 2410, 2477,
2479, 2564, 2603, 2828, 2831, 2838, 2840, 2849, 2850, 2851, 2860, 2861, 2869, 287
0, 2881, 2885, 2886, 2891, 2892, 2893, 2904, 2905, 2911, 2913, 2915, 2919, 2920, 2
922, 2925, 2926, 2928, 3225, 3244, 3246, 3247, 3256, 3257, 3258, 3259, 3260, 3261,
3267, 3275, 3276, 3277, 3281, 3282, 3283, 3285, 3287, 3319, 3330, 3344, 3358, 339
1, 3435, 3442, 3508, 3566, 3609, 3620, 3621, 3635, 3637, 3665, 3667, 3707, 3784, 3
890, 3921, 3968, 3988, 3994, 4011, 4034, 4049, 4052, 4075, 4083, 4103, 4109, 4120,
4150, 4170, 4182, 4200, 4204, 4241, 4320, 4329, 4347, 4386, 4391, 4411, 4412, 453
6, 4576, 4605, 4632, 4642, 4717, 4723, 4782, 4816, 4955, 4999, 5021, 5025, 5036, 5
070, 5076, 5079, 5105, 5117, 5137, 5204, 5215, 5226, 5284, 5285, 5293, 5296, 5301,
5341, 5342, 5348, 5413, 5433, 5437, 5571, 5662, 5667, 5680, 5681, 5766, 5842, 588
4, 5940, 6061, 6082, 6120, 6387, 6415, 6525, 6532, 6577, 6613, 6624, 6652, 6662, 6
701, 6720, 6738, 6745, 6791, 6875, 6939, 6952, 6973, 7085, 7216, 7267, 7320, 7365,
7378, 7422, 7628, 7710, 7715, 7717, 7814, 7855, 7865, 7903, 7970, 7984, 8106, 812
7, 8290, 8304, 8321, 8374, 8391, 8457, 8542, 8631, 8693, 8699, 8719, 8768, 8776, 8
838, 8839, 8840, 8845, 8850, 8882, 8883, 8884, 8892, 8908, 8926, 8927, 8928, 8970,
8971, 8972, 9014, 9015, 9016, 9058, 9059, 9060, 9066, 9102, 9103, 9104, 9146, 914
7, 9148, 9190, 9191, 9192, 9228, 9234, 9235, 9236, 9272, 9278, 9279, 9280, 9293, 9
322, 9323, 9324, 9333, 9366, 9367, 9368, 9372, 9410, 9411, 9412, 9454, 9455, 9456,
9498, 9499, 9500, 9542, 9543, 9544, 9586, 9587, 9588, 9600, 9607, 9630, 9631, 963
2, 9662, 9668, 9674, 9675, 9676, 9718, 9719, 9720, 9744, 9762, 9763, 9764, 9794, 9
797, 9806, 9807, 9808, 9842, 9850, 9851, 9852, 9894, 9895, 9896, 9938, 9939, 9940,
9982, 9983, 9984, 10026, 10027, 10028, 10070, 10071, 10072, 10114, 10115, 10116, 1
0158, 10159, 10160, 10202, 10203, 10204, 10217, 10222, 10246, 10247, 10248, 10264,
10277, 10290, 10291, 10292, 10327, 10334, 10335, 10336, 10365, 10378, 10379, 1038
0, 10422, 10423, 10424, 10466, 10467, 10468, 10480, 10485, 10490, 10510, 10511, 10
512, 10554, 10555, 10556, 10598, 10599, 10600, 10642, 10643, 10644, 10660, 10686,
10687, 10688, 10708, 10730, 10731, 10732, 10750, 10762, 10774, 10775, 10776, 1081
0, 10811, 10812, 10813, 10818, 10819, 10820, 10826, 10837, 10838, 10839, 10840, 10
841, 10843, 10844, 10845, 10846, 10847, 10855, 10862, 10863, 10864, 10877, 10878,
10879, 10887, 10889, 10890, 10891, 10895, 10896, 10897, 10898, 10899, 10906, 1090
7, 10908, 10920, 10921, 10922, 10923, 10924, 10927, 10928, 10930, 10931, 10950, 10
951, 10952, 10994, 10995, 10996, 11022, 11038, 11039, 11040, 11064, 11082, 11083,
11084, 11093, 11096, 11126, 11127, 11128, 11162, 11170, 11171, 11172, 11214, 1121
5, 11216, 11226, 11227, 11248, 11258, 11259, 11260, 11286, 11302, 11303, 11304, 11
346, 11347, 11348, 11370, 11385, 11390, 11391, 11392, 11399, 11400, 11413, 11419,
11434, 11435, 11436, 11456, 11478, 11479, 11480, 11522, 11523, 11524, 11537, 1155
1, 11566, 11567, 11568, 11610, 11611, 11612, 11646, 11650, 11654, 11655, 11656, 11
662, 11698, 11699, 11700, 11714, 11719, 11722, 11742, 11743, 11744, 11763, 11780,
11786, 11787, 11788, 11794, 11802, 11811, 11818, 11830, 11831, 11832, 11855, 1187
4, 11875, 11876, 11895, 11897, 11918, 11919, 11920, 11952, 11962, 11963, 11964, 11
973, 12006, 12007, 12008, 12017, 12039, 12050, 12051, 12052, 12094, 12095, 12096,
12103, 12104, 12119, 12138, 12139, 12140, 12153, 12166, 12182, 12183, 12184, 1220
3, 12211, 12216, 12226, 12227, 12228, 12239, 12262, 12270, 12271, 12272, 12285, 12
314, 12315, 12316, 12325, 12358, 12359, 12360, 12386, 12392, 12402, 12403, 12404,
12413, 12430, 12431, 12446, 12447, 12448, 12456, 12470, 12490, 12491, 12492, 1251
6, 12534, 12535, 12536, 12544, 12548, 12578, 12579, 12580, 12599, 12622, 12623, 12
624, 12666, 12667, 12668, 12710, 12711, 12712, 12728, 12754, 12755, 12756, 12778,
12785, 12798, 12799, 12800, 12829, 12842, 12843, 12844, 12848, 12870, 12886, 1288
7, 12888, 12905, 12930, 12931, 12932, 12941, 12974, 12975, 12976, 13018, 13019, 13
020, 13062, 13063, 13064, 13106, 13107, 13108, 13119, 13150, 13151, 13152, 13194,

13195, 13196, 13201, 13219, 13233, 13238, 13239, 13240, 13245, 13282, 13283, 1328
4, 13326, 13327, 13328, 13353, 13356, 13370, 13371, 13372, 13414, 13415, 13416, 13
422, 13428, 13453, 13458, 13459, 13460, 13469, 13502, 13503, 13504, 13546, 13547,
13548, 13558, 13586, 13590, 13591, 13592, 13609, 13618, 13624, 13634, 13635, 1363
6, 13678, 13679, 13680, 13706, 13713, 13722, 13723, 13724, 13766, 13767, 13768, 13
772, 13786, 13802, 13810, 13811, 13812, 13832, 13854, 13855, 13856, 13892, 13898,
13899, 13900, 13913, 13923, 13925, 13942, 13943, 13944, 13986, 13987, 13988, 1400
2, 14005, 14030, 14031, 14032, 14045, 14074, 14075, 14076, 14118, 14119, 14120, 14
128, 14162, 14163, 14164, 14206, 14207, 14208, 14217, 14221, 14238, 14250, 14251,
14252, 14262, 14264, 14283, 14294, 14295, 14296, 14302, 14306, 14319, 14338, 1433
9, 14340, 14377, 14382, 14383, 14384, 14391, 14392, 14399, 14403, 14417, 14421, 14
423, 14426, 14427, 14428, 14470, 14471, 14472, 14497, 14508, 14514, 14515, 14516,
14555, 14558, 14559, 14560, 14572, 14602, 14603, 14604, 14611, 14646, 14647, 1464
8, 14682, 14690, 14691, 14692, 14700, 14734, 14735, 14736, 14778, 14779, 14780, 14
809, 14812, 14822, 14823, 14824, 14832, 14834, 14866, 14867, 14868, 14880, 14895,
14899, 14910, 14911, 14912, 14926, 14931, 14954, 14955, 14956, 14982, 14998, 1499
9, 15000, 15005, 15034, 15042, 15043, 15044, 15057, 15073, 15086, 15087, 15088, 15
130, 15131, 15132, 15167, 15174, 15175, 15176, 15218, 15219, 15220, 15236, 15252,
15262, 15263, 15264, 15295, 15306, 15307, 15308, 15312, 15324, 15332, 15333, 1533
6, 15342, 15350, 15351, 15352, 15394, 15395, 15396, 15427, 15438, 15439, 15440, 15
482, 15483, 15484, 15512, 15513, 15526, 15527, 15528, 15553, 15570, 15571, 15572,
15576, 15595, 15611, 15614, 15615, 15616, 15658, 15659, 15660, 15667, 15702, 1570
3, 15704, 15746, 15747, 15748, 15752, 15755, 15778, 15790, 15791, 15792, 15834, 15
835, 15836, 15873, 15878, 15879, 15880, 15884, 15895, 15905, 15922, 15923, 15924,
15959, 15966, 15967, 15968, 15993, 16010, 16011, 16012, 16040, 16054, 16055, 1605
6, 16090, 16098, 16099, 16100, 16114, 16115, 16127, 16131, 16142, 16143, 16144, 16
161, 16170, 16186, 16187, 16188, 16206, 16220, 16230, 16231, 16232, 16239, 16247,
16255, 16274, 16275, 16276, 16280, 16286, 16311, 16312, 16313, 16318, 16319, 1632
0, 16329, 16330, 16332, 16333, 16334, 16337, 16341, 16342, 16353, 16354, 16355, 16
362, 16363, 16364, 16368, 16373, 16374, 16377, 16381, 16382, 16399, 16403, 16406,
16407, 16408, 16413, 16420, 16428, 16432, 16440, 16450, 16451, 16452, 16458, 1645
9, 16464, 16465, 16475, 16494, 16495, 16496, 16514, 16517, 16533, 16538, 16539, 16
540, 16550, 16551, 16552, 16561, 16574, 16578, 16582, 16583, 16584, 16588, 16589,
16590, 16595, 16613, 16614, 16615, 16616, 16618, 16626, 16627, 16628, 16644, 1664
8, 16672, 16680, 16690, 16695, 16696, 16714, 16715, 16716, 16751, 16
758, 16759, 16760, 16770, 16780, 16794, 16802, 16803, 16804, 16811, 16846, 16847,
16848, 16876, 16890, 16891, 16892, 16896, 16934, 16935, 16936, 16978, 16979, 1698
0, 16996, 17003, 17022, 17023, 17024, 17029, 17034, 17057, 17066, 17067, 17068, 17
081, 17084, 17086, 17110, 17111, 17112, 17123, 17137, 17140, 17147, 17148, 17149,
17150, 17154, 17155, 17156, 17182, 17186, 17190, 17193, 17198, 17199, 17200, 1722
2, 17242, 17243, 17244, 17279, 17286, 17287, 17288, 17294, 17324, 17330, 17331, 17
332, 17351, 17368, 17374, 17375, 17376, 17384, 17397, 17413, 17418, 17419, 17420,
17462, 17463, 17464, 17474, 17480, 17482, 17496, 17506, 17507, 17508, 17517, 1752
5, 17529, 17532, 17535, 17546, 17550, 17551, 17552, 17558, 17582, 17583, 17594, 17
595, 17596, 17601, 17629, 17630, 17638, 17639, 17640, 17661, 17678, 17680, 17681,
17683, 17691, 17712, 17718, 17719, 17722, 17728, 17729, 17752, 17756, 17757, 1776
2, 17794, 17795, 17799, 17814, 17832, 17833, 17855, 17856, 17870, 17871, 17877, 17
881, 17908, 17909, 17928, 17931, 17934, 17941, 17946, 17947, 17955, 17967, 17971,
17983, 17984, 17985, 18001, 18005, 18008, 18021, 18022, 18023, 18025, 18057, 1806
0, 18061, 18098, 18099, 18108, 18113, 18129, 18132, 18136, 18137, 18163, 18167, 18
168, 18174, 18175, 18181, 18188, 18198, 18207, 18212, 18213, 18218, 18250, 18251,
18285, 18288, 18289, 18297, 18316, 18318, 18326, 18327, 18331, 18362, 18363, 1836
4, 18365, 18366, 18393, 18400, 18402, 18403, 18416, 18435, 18440, 18441, 18458, 18
470, 18478, 18479, 18488, 18503, 18508, 18516, 18517, 18535, 18538, 18540, 18551,
18554, 18555, 18557, 18586, 18592, 18593, 18601, 18602, 18603, 18606, 18622, 1862
3, 18630, 18631, 18634, 18638, 18654, 18660, 18668, 18669, 18670, 18674, 18688, 18
694, 18698, 18705, 18706, 18707, 18710, 18723, 18729, 18733, 18737, 18743, 18744,
18745, 18754, 18770, 18771, 18773, 18782, 18783, 18797, 18798, 18808, 18820, 1882
1, 18828, 18832, 18851, 18858, 18859, 18873, 18878, 18880, 18890, 18896, 18897, 18
905, 18906, 18919, 18924, 18935, 18940, 18972, 18973, 18976, 18979, 19010, 19011,
19016, 19029, 19043, 19048, 19049, 19067, 19080, 19084, 19086, 19087, 19112, 1912

In [46]:
```
movies_dataset.delete_row(index=11, 3)
```

In [47]:
```
helper5 = movies_dataset.col_helper(5)
helper5.select_col('run_time')
```

7 views 观看次数

In [48]:
```
helper.select_col('views')
```

In [49]:
```
helper.count_none()
```

Out[49]:
```
1
```

有1个缺失值，这里猜测，由于盗版电影网站不一定会那么全面的数据，因此猜测是数据不全造成的

这里可以进行缺失值的处理方式有以下几种：

删除所有缺失数据

将缺失数据用中位数或平均数值代替

利用数据对象的相关性进行填补

利用属性的相关性进行填补

In [50]:

```
helper.data2['views'].value_counts()
```

Out[50]:

```
6259.00    6
5859.00    5
5926.00    5
6043.00    5
14859.00   5
          ..
142440.00  1
1045070.00 1
1527768.00 1
6400.00    1
1419.00    1
Name: views, Length: 108227, dtype: int64
```

In [51]:

```
index11,row11 = movies_dataset.get_row_index('views','')
print(index11)
movies_dataset.delete_row(index11,6)
```

```
[149]
```

In [52]:

```
helper6 = movies_dataset.col_helper(6)
helper6.select_col('views')
```

In [53]:

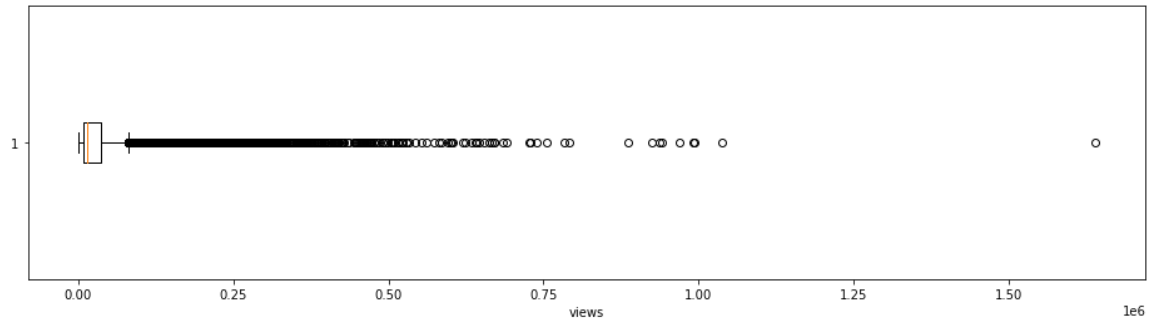
```
# 五数概括
helper6.five_number()
```

```
Min: 667.0
Q1: 7571.5
Q2: 15222.0
Q3: 36571.0
Max: 1638533.0
```

In [54]:

```
# 盒图
helper6.box(16,4,'views')
```

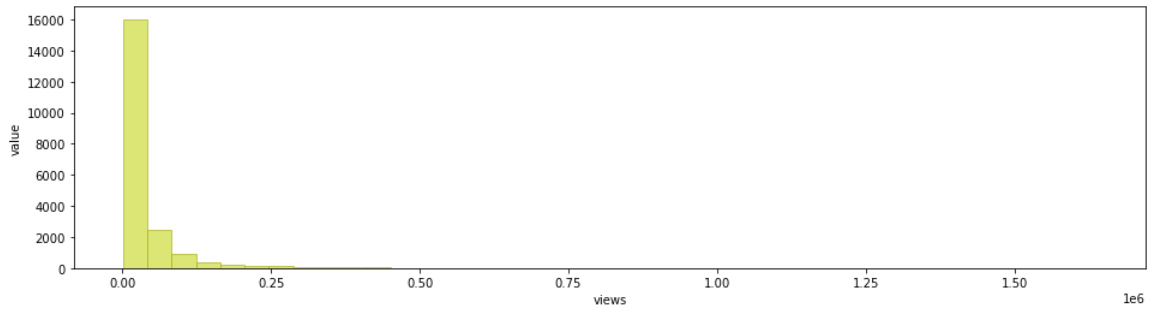
```
[ 2794.   1002.  14419. ...   7220.   1419.   6697.]
```




```
80070.25
```

In [55]:

```
# 直方图
helper6.normal_hist(16,4,'views','value')
```



从盒图和直方图可以看出，大多数的电影观看观看数适中，符合“盗版网站”的特点。