CS 212

Team Assignment

Group 17

Jiacheng Shen shenjch18@lzu.edu.cn **320180949191**

Ziyao Wang wangziyao2018@lzu.edu.cn **320180940361**

Motong Tian tianmt18@lzu.edu.cn **320180940301**

Yichen Wang ychwang2018@lzu.edu.cn **320180940341**

Zixiang Song songzx18@lzu.edu.cn **320180940221**

Ke Lei leik18@lzu.edu.cn **320180939861**

Jiang Zhou zhoujiang18@lzu.edu.cn **320180940681**

Wei Zhu zhuw2018@lzu.edu.cn **320180940701**

Abstract

This group assignment is focus on the data of kernel to discover the features of the kernel. There are mainly four steps to analyze the feature:grab the data from kernel, change the format of the data for the visualization of the data and the test of the metric use the metric to analyze the feature of the kernel draw a conclusion.

Here in our assignment, we focus on the metric and the analyze around the metric.

CONTENTS

1 Hypothesis

We have three hypothesis:

1. Release time is positively related to the number of bugs.
2. The number of bugs will be less as the patch release.
3. Patchlevel release time is regular to some extend.

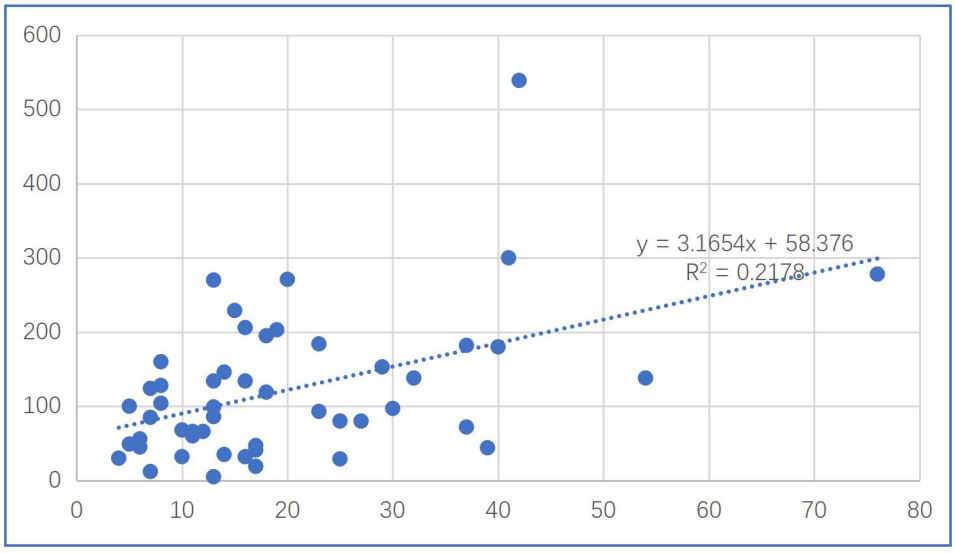
2 Metrics

To prove out the hypothesis above, we choose three metrics respectively:

1. Related coefficient based on scatter diagram.
2. Bug quantity distribution based on line chart.
3. Release timing distribution based on histogram.

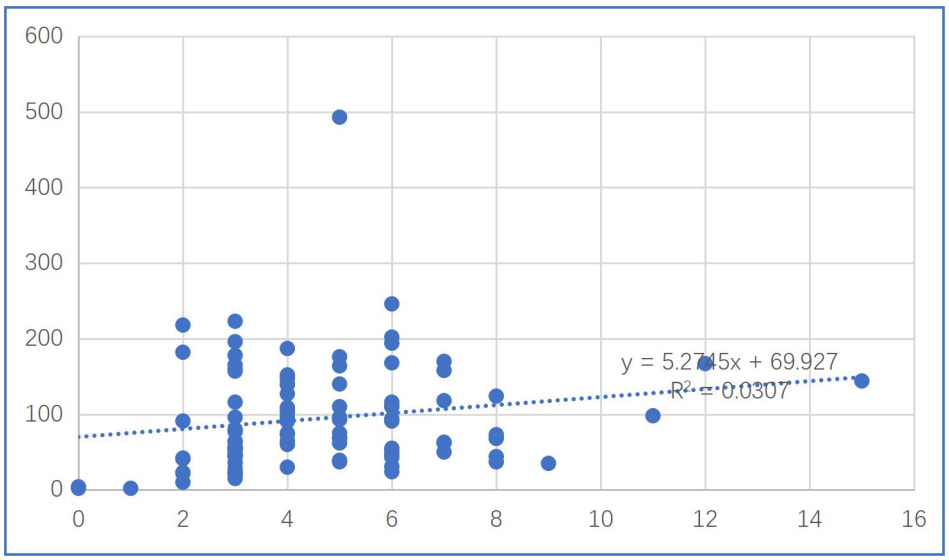
2.1 Scatter Diagram: release timing vs. bugs quantity

v4.1



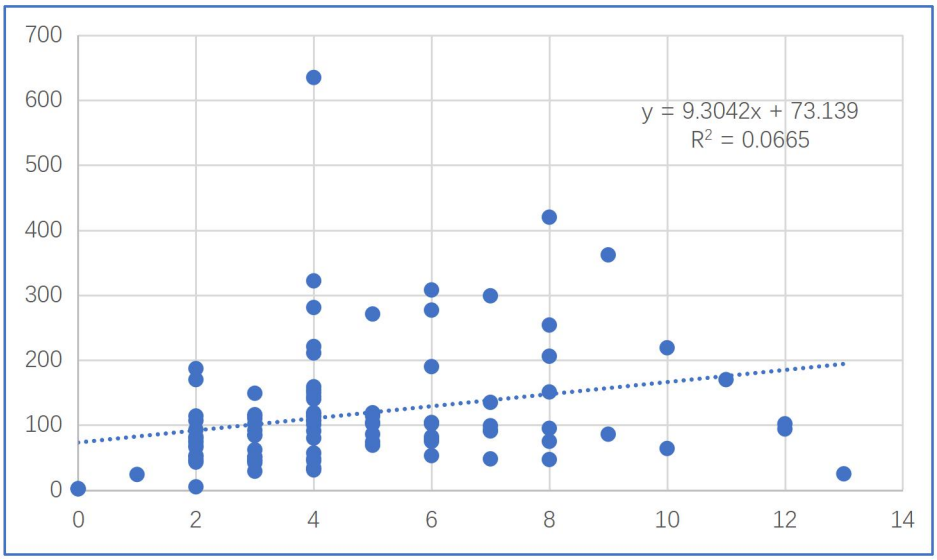
r=0.466, r>0.3, it means that the handle bugs and release interval has the media positive relationship.

v4.14



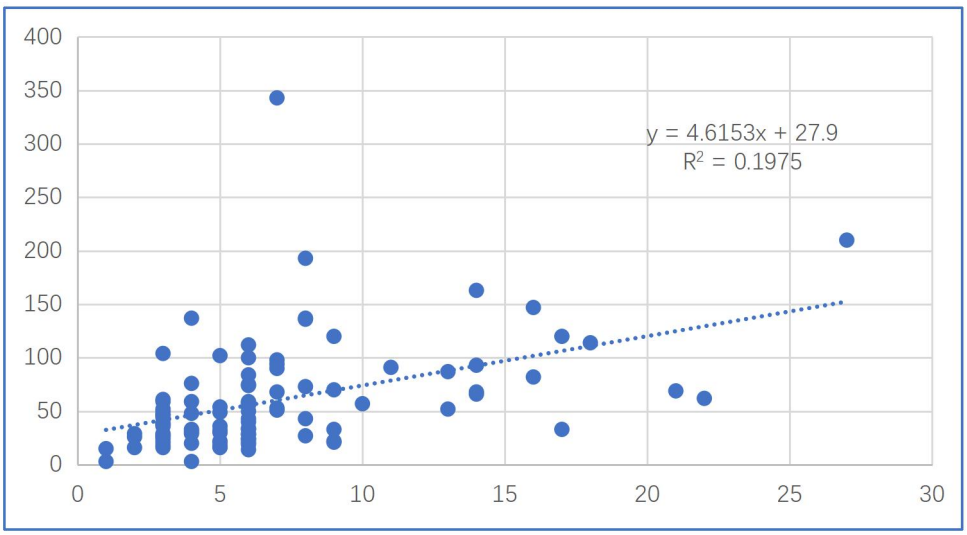
r=0.175<0.3, it means hat the handle bugs and release interval has the weak positive relationship.

v4.19



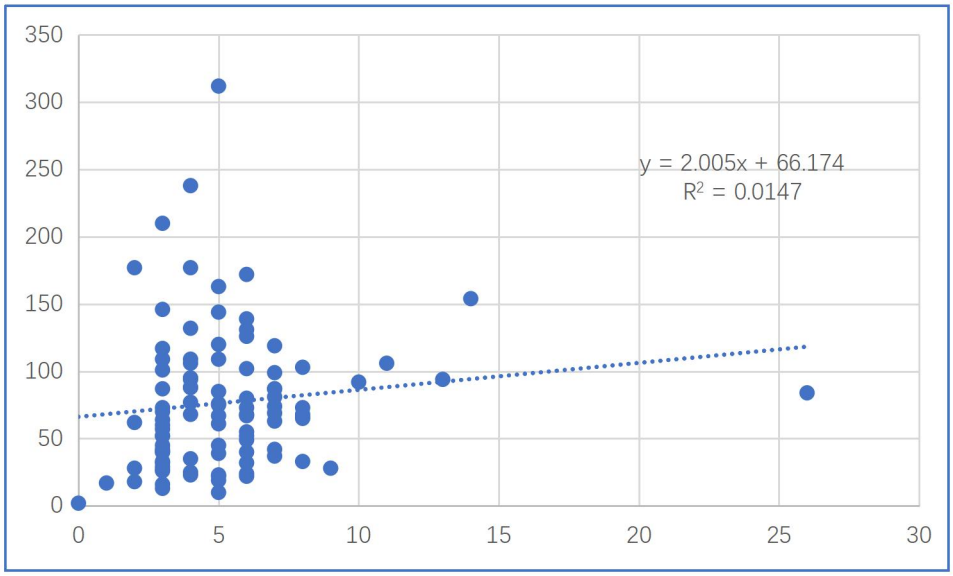
r=0.257, r<0.3, it means hat the handle bugs and release interval has the weak positive relationship.

v4.4



r=0.444, r>0.3,It means that the handle bugs and release interval has the media positive relationship.

v4.9



r=0.121, r<0.3, it means that the handle bugs and release interval has the weak positive relationship.

Summery

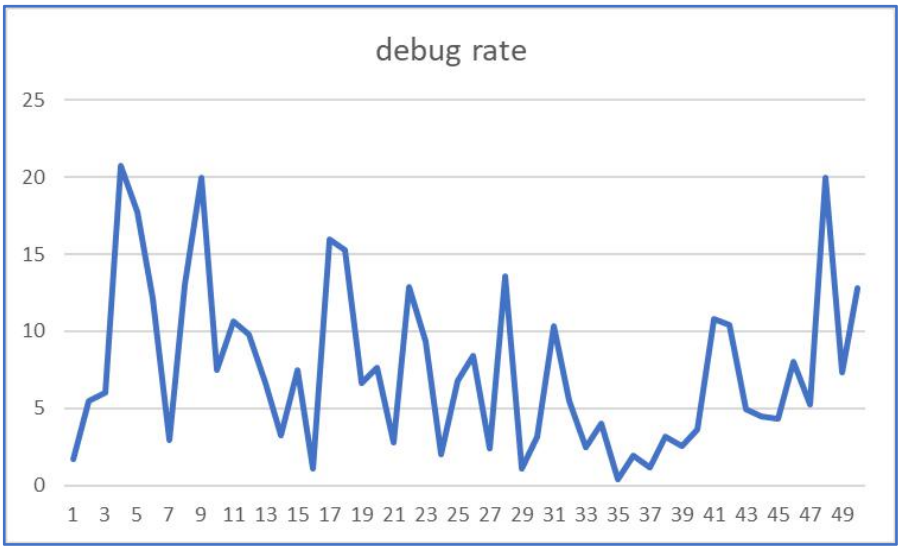
v4.1	0.466	media positive
v4.14	0.175	weak positive
v4.19	0.257	weak positive
v4.4	0.444	media positive
v4.9	0.121	weak positive

The metric is well -defined as it can indicate the degree of correlation between bugs and release time. By comparing the diagrams, it is easy to find that the length of the days between two releases more or less has a positive relationship with the number of bugs in every release of the kernel. The correlation coefficient r is around 0.4 to 0.2 among the diagrams, so the metric is robust. Release time is positively related to the number of bugs according to analysis.

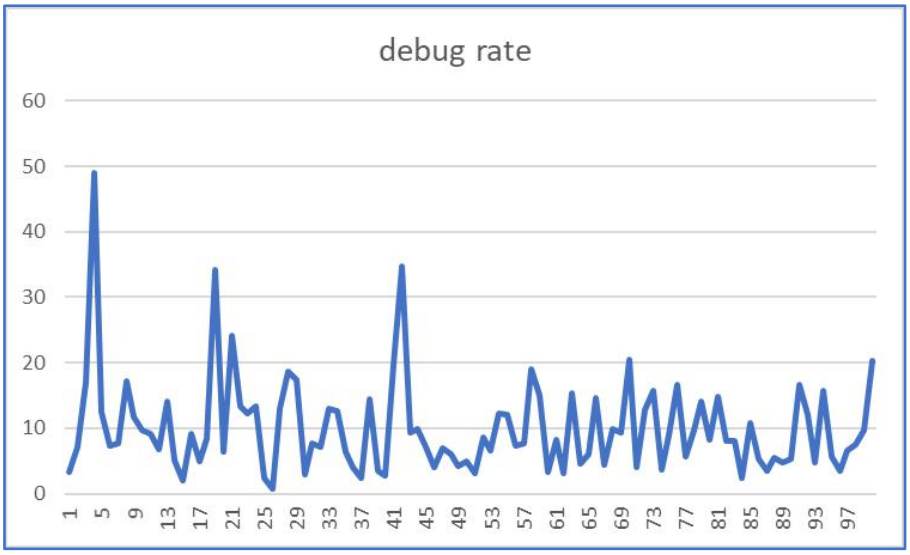
This tells us that at least one of the most significant factors that influence the release time should be the number of bugs need to be fixed in one release.

2.2 Line Chart: patchlevel releases vs. bugs quantity

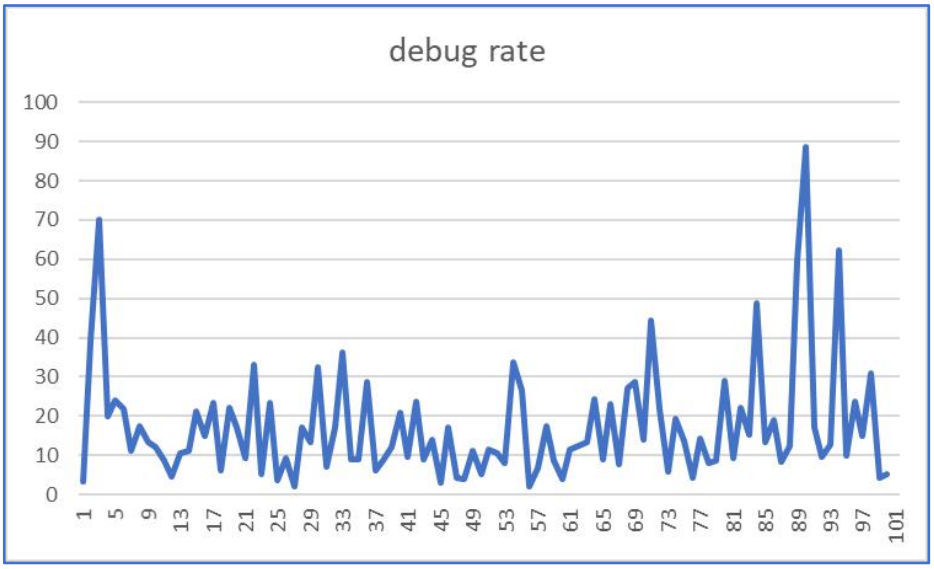
v4.1



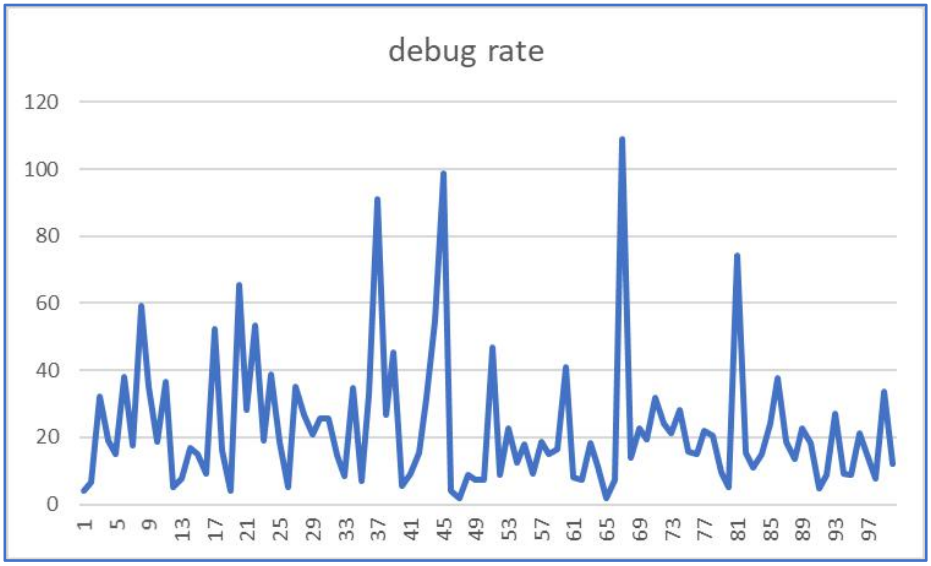
v4.4


debug rate
60
50
40
30
20
10
0
1
5
9
13
17
21
25
29
33
37
41
45
49
53
57
61
65
69
73
77
81
85
89
93
97

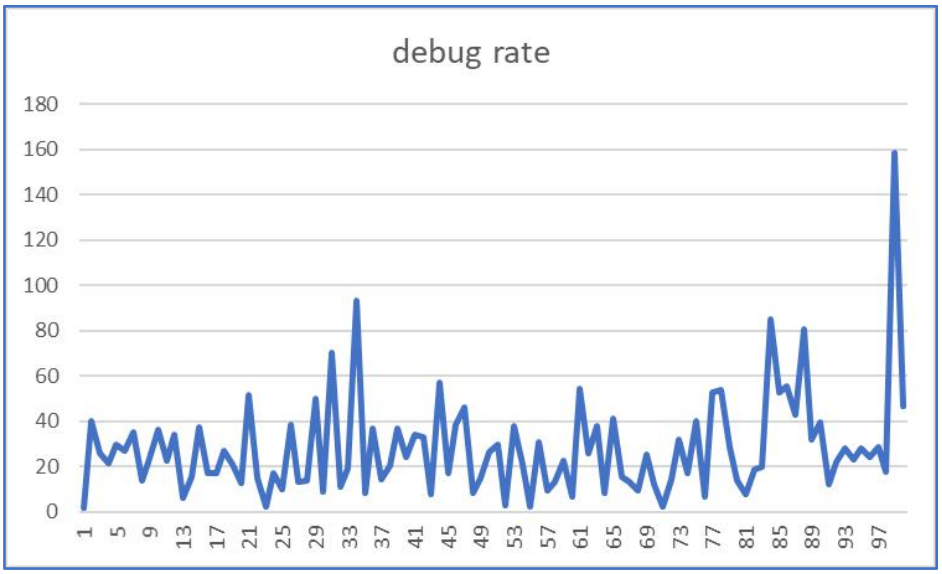
v4.9


debug rate
100
90
80
70
60
50
40
30
20
10
0
1
5
9
13
17
21
25
29
33
37
41
45
49
53
57
61
65
69
73
77
81
85
89
93
97
101

v4.14


debug rate
120
100
80
60
40
20
0
1
5
9
13
17
21
25
29
33
37
41
45
49
53
57
61
65
69
73
77
81
85
89
93
97

v4.19



Summery

v4.1	7.556
v4.4	10.0312
v4.9	17.334
v4.14	22.968
v4.19	28.01

The metric is well-defined, as it can indicate the variation of number of bugs as the release going on. The metric is not robust though, as the distributions of the five versions are different. And our hypothesis is proved to be wrong, as the number of bugs appears to be randomly happened during development of kernel.

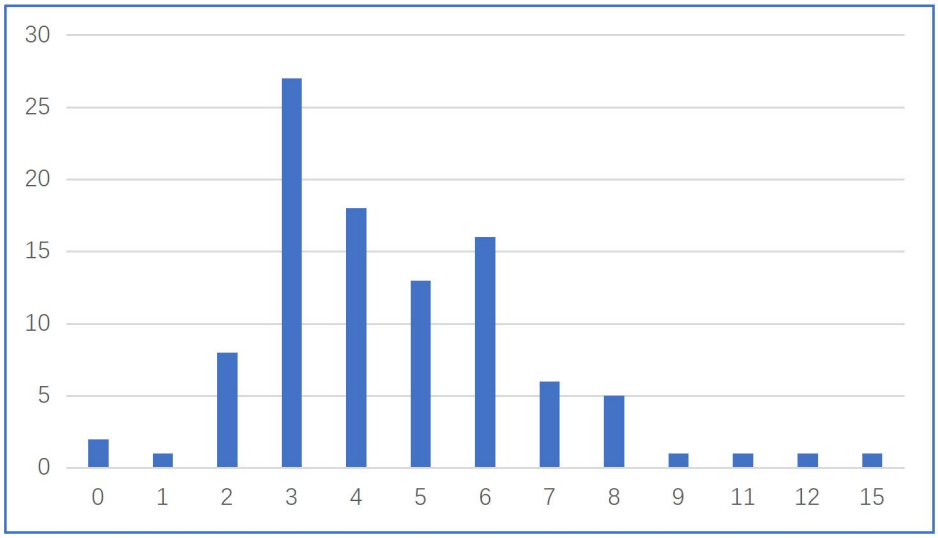
We could see that the most of the number of the bugs is around a constant, but there are still some peaks between some of the releases. It reflects that in a certain version of the kernel, the number of the bugs between two releases could be estimated by the average number of the bugs before. But there is still some special cases may caused by the users which will lead the increase of the bugs.

Another point we should focus on is the average number of the bugs between different versions of the kernel. It seems like a trend of increase. Maybe the latest version it is, the more comprehensive it will be. This directly leads to the increase of the bugs.

2.3 Histogram: patchlevel releases timing in days

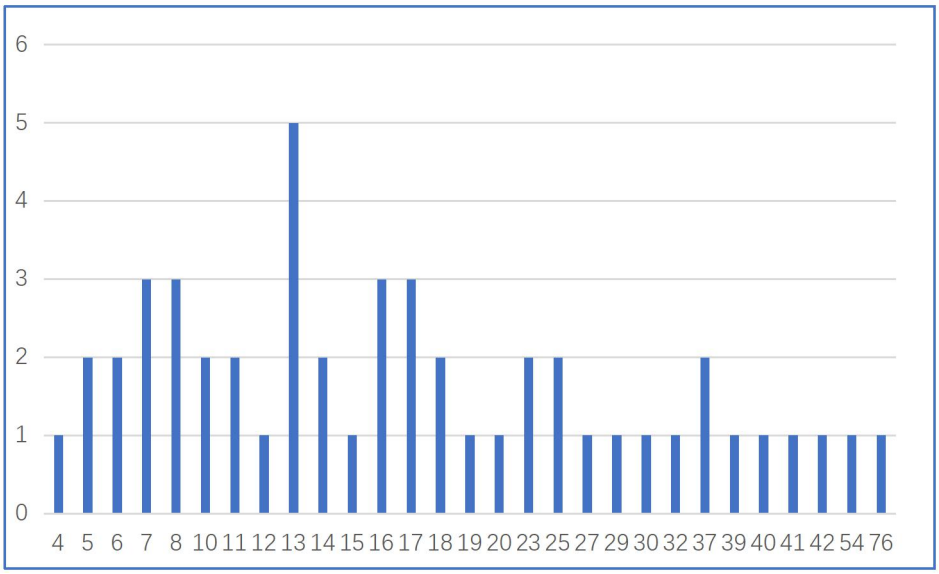
For the histogram, I choose the coefficient of skew. I suppose that the histogram is right-skewed.

It means the Cs should >0



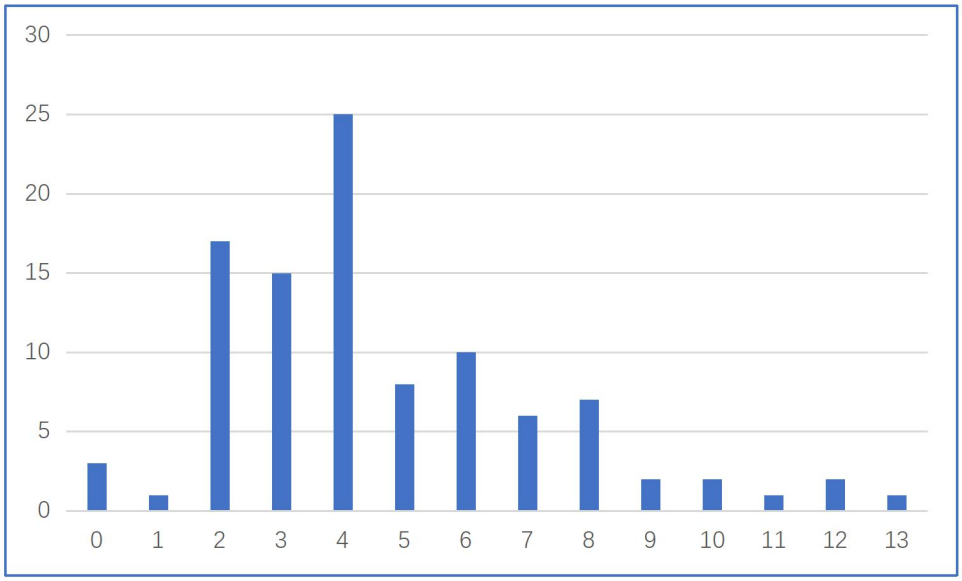
v4.14

Cs=1.317, it means the histogram is right skewed. When the horizontal axis is 3, the maximum value is 27.



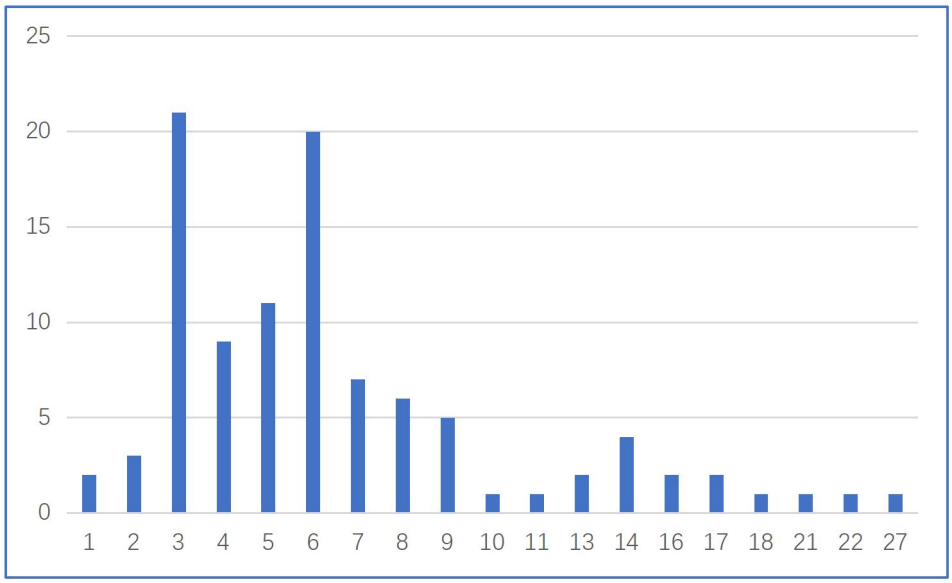
v4.1

Cs=1.745, it means the histogram is strongly right skewed. When the horizontal axis is 13, the maximum value is 5.



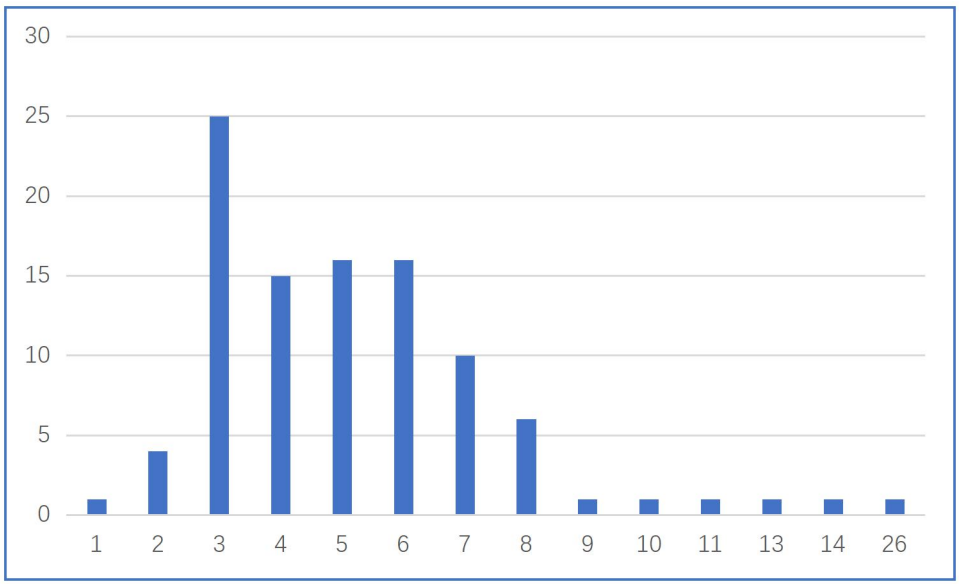
v4.19

Cs=1.274, it means the histogram is right skewed. When the horizontal axis is 4, the maximum value is 25.



v4.4

Cs=0.923, it means the histogram is right skewed. When the horizontal axis is 3, the maximum value is 21.



v4.9

Cs=1.116, it means the histogram is right skewed. When the horizontal axis is 3, the maximum value is 25.

Summery

v4.14	1.317	27
v4.1	1.745	5
v4.19	1.274	25
v4.4	0.923	21
v4.9	1.116	25

The metric is well-defined as it directly shows the frequency distribution of release time. And the metric is robust, because all the histograms are similar on skewness and peakness. So our hypothesis proved to be right: release time is regular.

As we can see the data above, it is clear that all the histogram is right-skewed, indicating that most of releases takes less than 8 days, which means releases happened within one week at most of the time of kernel development. Moreover, three days is mostly where the peaks point happen, which means three days is kind of a stable time that a large proportion of patch need to release or the basic of a version need to be developed completely, then it will move into so-called ‘non-new-feature’ phrase. Within 7 days, there are also some shorter peaks, which means the climax of bug fixing is brought about by testing.

3 Conclusions

As the data we analyze above, the kernel, as we think is a system which is stable in some perspectives—we fix its bugs in a regular pace. We could see it clearly in the chart we given above. It has relationship with the time. And between two releases the number of the bugs remains in a stable interval. There is also a trend that with the newest version of the kernel the average bugs it appears between the gap of the release would also increase. So I have the reason to believe that there will be more bugs between the gap of the release in the future.

What's more, the release time is regular to some extend, which ensures that the management and control of kernel development is rigorous at least in terms of timing controlling. And that shows the high quality of kernel development to some extent.

Another thing I would like to mention is that the bugs between the gap of the release would sometimes turn to be a peak. It may be caused by the other factors in the reality (the increase of the users, etc.). With the data above we can not ensure the factors.