Objective: comparing the efforts of fixing cross-project bugs and other bugs

Data: we select three kinds indicators which could be easily collected from the bug reports to reflect the difficulty or efforts of fixing a bug

- fixing time: the duration between the reported time to the closed time of a bug
- the number of comments: the total number of comments in a bug report
- the number of participants: the total number of developers participating the discussion of a bug

Method: The Wilcoxon rank-sum test with the significant level of 0.05

The null hypothesis $\boldsymbol{H_0}$ and its alternative hypothesis $\boldsymbol{H_A}$ are stated as follows:

- $\boldsymbol{H_0}$: The fixing time / the number of comments / the number of participants of cross-project bugs is the same as that of within-project bugs.
- $\boldsymbol{H_A}$: The fixing time / the number of comments / the number of participants of cross-project bugs is significantly different from that of within-project bugs.

The null hypothesis $\boldsymbol{H_0}$ is rejected if the p-value of the Wilcoxon rank-sum test is less than 0.05, which means that the fixing time / the number of comments / the number of participants of cross-project bugs is significantly different from that of within-project bugs

Result: TABLE I. shows the p-values obtained from the Wilcoxon rank-sum tests for comparing the fixing time, the number of comments, and the number of participants between cross-project and within-project bugs in the seven studied projects. The *p* values larger than 0.05 are marked in red.

TABLE I. The *p* values of Wilcoxon rank-sum tests

Project	fixing time	#comments	#participants
NumPy	1.21E-04	3.22E-07	6.24E-05
SciPy	1.85E-02	2.30E-07	5.20E-05
Ipython	1.07E-01	8.18E-05	1.08E-06
Pandas	3.04E-04	9.76E-07	5.70E-04
Matplotlib	1.28E+00	3.42E-04	2.62E-02
Astropy	8.60E-03	2.42E-04	6.46E-01
Scikit-learn	1.94E-02	3.28E-02	1.51E-01

From the table, it is easy to see that the fixing time, the number of comments, and the number of participants in fixing cross-project bugs are significant different from those in fixing within-project bugs in nearly all projects (except for the fixing time in *Ipython* and *Matplotlib*, and the number of participants in *Astropy* and *Scikit-learn*).

To determine whether fixing cross-project bugs needs more or less effort than fixing within-project ones, we compare the median values of the three indicators in the two kinds of bugs. The results are shown in TABLE II. It can be seen that in all cases except the two marked in red, the fixing time, the number of comments, and the number of participants for cross-project bugs all show higher median values than those for within-project values.

TABLE II. The median values of the three indicators

		fixing time	**#comments**	**#participants**
NumPy	cross-project	551367	7	4
	with-project	117979	3	3
SciPy	cross-project	1710502	6.5	4
	with-project	730569	3	3
Ipython	cross-project	1698186	6	4
	with-project	1402518	3	3
Pandas	cross-project	796093	6	3
	with-project	268966	2	2
Matplotlib	cross-project	3549699	9	4
	with-project	3717850	4	3
Astropy	cross-project	1091204	8	3
	with-project	478721	5	3
Scikit-learn	cross-project	9675766	8	4
	with-project	1802469	6	3

Combining the results of the Wilcoxon rank-sum tests and the comparison of median values, we know that the fixing time, the number of comments, and the number of participants for cross-project bugs are significantly larger than those for within-project bugs.

Conclusion: it takes significantly more efforts to repair cross-project bugs than within-project bugs in the studied projects.