## **Correlation between Maintainability index and Post-Release Defect Density**

***Calculating Spearman Correlation***

Spearman’s Rank correlation coefficient is one of the most-prominent technique which can be used to find out the strength and correlation between two variables. We have calculated the spearman correlation coefficient using the following method for one project and for the rest of the projects we have used the online tool combining all the versions of a project

***Method used to calculate the Spearman correlation***

* Create a table from your data and get the ordered pairs of two variables.
* Rank the two data sets. Ranking is achieved by giving the ranking '1' to the biggest number in a column, '2' to the second biggest value and so on. The smallest value in the column will get the lowest ranking. This should be done for both sets of measurements or the variables used to find the correlation for.
* Tied scores are given the mean (average) rank.
* Find the difference in the ranks (d).
* Square the differences (d²) To remove negative values and then sum them
* Calculate the coefficient (***Rs***) using the formula mentioned below.

When written in mathematical notation the Spearman Rank formula looks like this:



Here,

ρ= Spearman rank correlation

di= the difference between the ranks of corresponding variables

n= number of observations

We have used the formula of spearman correlation coefficient by considering the Post-Release Defect Density values as X- values and the Maintainability index values as Y- values.

Table gives the values of the Post-Release Defect Density, Maintainability Index , as well as Spearman Correlation Coefficient for the particular version for each project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Projects | Versions | Maintainability Index | Post release Defect density | Spearman coefficient |
| Apache Commons DbUtils | 1.2 1.3 1.4 1.5 1.6 | 77.74 77.75 74.03 72.92 70.23 | 0.002629 0.000514 0.001873 0.000929 0.000582 | 0.14 |
| Apache Commons Lang | 3.4 3.5 3.6 3.7 3.8 | 45.73 51.97 51.96 52.5 52.44 | 0.000455 0.0001723 0.0002106 0.00005125 0.0000396 | -0.9 |
| Apache Commons Collections | 3 4 4.1 4.2 4.3 | 69.03 66.52 65.01 76.30 76.29 | 0.00106502 0.0009584 0.0007167 0.0002634 0.0000159 | 0.3 |
| Apache Commons Math | 3.2 3.3 3.4 3.5 3.6 | 67.58 67.48 67.06 66.95 60.56 | 0.000233 0.000189 0.0000513 0.0000193 0.00006225 | 0.7 |
| JFreeChart | 1.0.4 1.0.6 1.0.7 1.0.8 1.5.0 | 63.82 63.00 56.04 63.56 62.56 | 0.0004797 0.0000898 0.0001585 0.0001586 0.0008947 | 0.2 |

By analysis of the result as seen in Table, we have studied that each project has correlation between weak to medium. So that, we can say that having a better/higher maintainability index ensures less software maintenance costs. However, it might not always ensure a bug free system. In addition, there are also some other factors that affects the value of the post release defect density like the experience of the developers and testers , the type of the defects taken into account and the time required for the calculation of the post release defect density calculation.