International Journal of Database Theory and Application
Vol.9, No.3 (2016), pp. 11-18
http://dx.doi.org/10.14257/ijdta.2016.9.3.02

# Bootstrap Correlation Analysis of Function Point Elements

Masood Uzzafer

*Associate Professor*
*Amity University Dubai*
*muzzafer@amityuniversity.ae*

***Abstract***

*This research work investigates the correlation of software function point elements using bootstrap simulation. The correlation of software function point elements plays an important role in understanding the software size; the correlation among function point elements suggests that they measure the same attribute of a software project. Bootstrapping is an effective method to study the statistical properties of correlation coefficients; bootstrap produces a histogram of the possible values of correlation coefficients, which helps to understand the range and spread of the correlation among different function point elements, rater then generating a single point estimate of the correlation.*

***Keywords****: Software Project size, Effort estimation, Function point, Statistical Analysis*

## 1. Introduction

Function Point (FP) has attracted much attention among software engineering researchers and practitioners. The FP were introduced by IBM in 70's, since then their nature, behaviour, impact, distribution and correlation have been studied with interest. The idea behind FP's is to standardize the measurement of the various software functions to estimate the software development effort which is independent of the computer language, development methodology, technology and the capability of the team developed the software. The international Point users Group (IFPUG) is a membership governed, non-profit organization committed to promoting and supporting the FP. There have been various releases of the FP's by the IFPUG with the latest 'Counting Practices Manual – 4.3.1 which was released in 2010.

Software engineering researchers [1-3] have studied the correlations among function point elements. The correlations among function point elements suggest that they are measuring the same attribute of a software project by the proportion of the correlation coefficient. Uncorrelated FP suggests that they measures unique attributes of a software development project. Therefore, software size estimates based on uncorrelated function point elements represents the true size estimate of a software project. Whereas, correlated FP capture the same attribute of the software project. Small FP correlation values suggest that the FP are unique having certain similar attributes or parts of attribute and large values of correlation suggest that these FP's captures the same attribute, thus considering the FP that are strongly correlated leads to double counting the attributes of a software project. This causes misleading software size estimates.

This paper studies the correlation among different FP using the bootstrap technique. Bootstrapping is a simulation method that estimates the properties of a statistics. Through bootstrapping the distribution of an estimator is established by estimating it from samples taken from an approximating distribution.

ISSN: 2005-4270 IJDTA

The paper is organized as follows: Section 2 describes the FP, Sections 3 discusses the dataset used for bootstrap analysis, Section 4 focuses on the correlation among FP using the bootstrapping technique, and Section 5 draws some conclusions.

## 2. Function Points Description

The FP is used to estimate the size of a software development project. The FP is categorized into five elements to capture different attribute of software projects that helps to model the size of software projects; these elements are: Internal Logical Files (ILF), External Interface Files (EIF), External Inputs (EI), External Outputs (EO) and External Enquiry (EQ). The software size estimation processes begins with the counting the five FP elements. Furthermore, the associated file numbers of a software project are also counted. The associated file numbers are Data Element Type (DET), File Type Referenced (FTR) and Record Element Types (RET). Each FP element is assigned a complexity level (Low, Average, High) based on its RET, DET and FTR number. The complexity metrics for five elements is shown in Table 1. Each function component is then assigned a weight according to its complexity is shown in Table 2.

**Table 1. Function Point Element Complexity Metrics**

| ILF/EIF | | | | EI | | | | EO/EQ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | DET | | | | DET | | | | DET | | |
| RET | 1-19 | 20-50 | 51+ | FTR | 1-4 | 5-15 | 16+ | FTR | 1-5 | 6-19 | 20 |
| 1 | Low | Low | Avg | 0-1 | Low | Low | Avg | 0-1 | Low | Low | Avg |
| 2-5 | Low | Avg | High | 2 | Low | Avg | High | 2-3 | Low | Avg | High |
| 6+ | Avg | High | High | 3+ | Avg | High | High | 4+ | Avg | High | High |

**Table 2. Function Point Complexity Weights**

| Component | Low | Average | High |
|---|---|---|---|
| External Inputs | 3 | 4 | 6 |
| External Outputs | 4 | 5 | 7 |
| External Inquiries | 3 | 4 | 6 |
| Internal Logical Files | 7 | 10 | 15 |
| External Interface Files | 5 | 7 | 10 |

Then the unadjusted FP (UFP) count is computed from the equation 1,

$$\text{UFP} = \sum_{i=i}^{5} \sum_{j=1}^{3} w_{ij} x_{ij} \qquad (1)$$

Where $w_{ij}$ is the complexity weight and $x_{ij}$ is the count for each FP. Next step is the estimation Value Adjustment Factor (VAF). The VAF is calculated from 14 General System Characteristics (GSC) using equation 2. These characteristics are 1) Data Communication 2) Distributed Functions 3) Performance 4) heavily used configuration 5) transaction rate 6) on-line data entry 7) end user efficiency 8) on-line update 9) complex processing 10) reusability 11) installation ease 12) operational ease 13) multiple sites and 14) facilities change. VAF is the sum of all the GSC,

$$\text{VAF} = 0.65 + 0.01 \sum_{i=1}^{14} c_i \qquad (2)$$

Where $c_i$ are the GSC values. Finally the UFP and VAF are multiplied to get the adjusted FP (FP) count, where FP represents the size of the software project,

$$FP = UFP \times VAF \tag{3}$$

## 3. Understanding the Dataset

The bootstrap correlation analysis is performed on the function point dataset of the International Software Benchmarking Standards (ISBSG) repository [4]. ISBSG performs the data validation of the contributed datasets to ensure the data quality and consistency. The obtained repository contains data from 3024 different projects, where almost all the projects used IFPUG standard [5] for function points. Projects which used other methods then IFPUG were excluded from this research study. Dataset with the missing function point values were also excluded.

In the selected projects largest projects were contributed by the financial industry (banking, financial services, and accounting) the rest of the projects were from engineering (software, hardware and telecommunication), insurance, public administration, government, manufacturing, consulting and education. This dataset is not homogenous and the variety in the dataset ensures that the dataset represents different scenarios and possibilities within the software development industry.

The Box plot all five FP elements (EO, EQ, EI, ILF and EIF) of the selected dataset are shown in Figure 1. Box plots helps to understand the measure of central tendency and dispersion of any dataset.

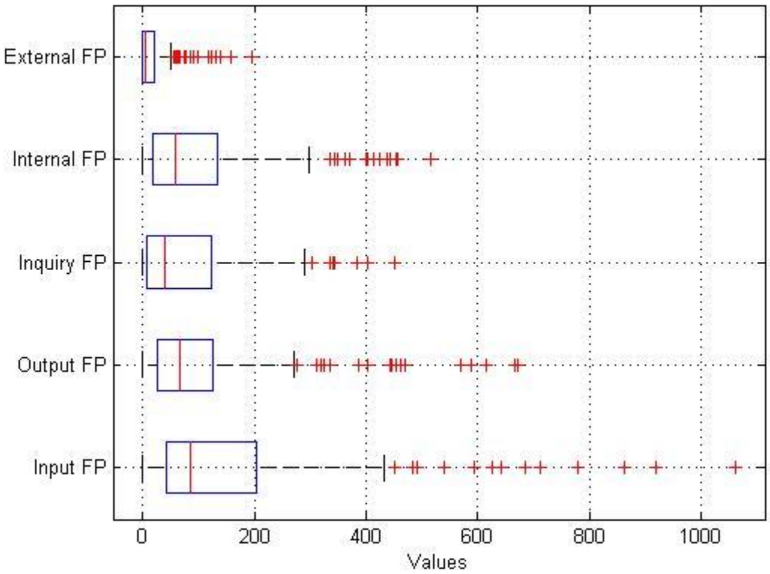

**Figure 1. Box Plot of Function Point Elements**

FP elements which are 3 times the standard deviation away from the sample mean are classified as the outliers and are removed. The line in the middle of the box represents the median if the line is not in the center of the box that is an indication of the skewness. Skewness is a measure of asymmetry of the data around the sample mean/median. The lower and upper lines of the box are 25th and 75th percentiles, respectively. The distance between the upper and lower lines is the inter-quartile range. Whiskers, lines extending above and below the box, represent the rest of the data. The length of the whiskers is set to 1.5 times the inter-quartile range. Plus signs show the data point which the 1.5 times away from the inter-quartile range. Table 3 compiles the median and percentiles of the selected function point elements.

The inspection of the Box plot reveals that the input function point element (EI) has the longest tail above the upper whisker and its values are more widely spread over the upper whisker than other function point elements. In-addition, the median line of EI is not in the middle of the box representing a positive skew meaning that the data values are more spread out after the median. This phenomenon is also observed with other function point elements and found to be common in all function point elements. The external function point has the smallest set of values with fairly small upper and none existing lower whisker.

**Table 3. Median and Percentiles of the Function Point Elements**

| | **Median** | **25th percentile** | **50th percentile** | **75th percentile** | **100th percentile** |
|---|---|---|---|---|---|
| Input (EI) | 86.5 | 41.5 | 86.5 | 204.5 | 1061 |
| Output (EO) | 67 | 26 | 67 | 125 | 673 |
| Inquiry (EQ) | 39 | 7.25 | 39 | 122.5 | 450 |
| Internal (ILF) | 58 | 17.75 | 58 | 133 | 516 |
| External (EIF) | 5 | 0 | 5 | 20 | 195 |

Figure 2 shows the histogram of the UFP of the dataset. The minimum project size is recorded to be 13 UFP while the largest project size is 4943 UFP with the overall mean of 579.33 UFP with the standard deviation of 715.46 UFP. Majority of the software projects sizes were in the range of 13 to 500 UFP, whereas few projects sizes were more than 2000 UFP.

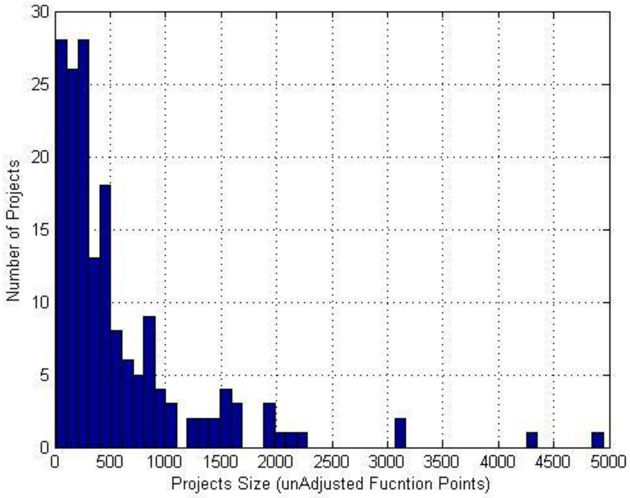

**Figure 2. Project Sizes of the Dataset (UFP)**

## 4. Function Points Correlations through Bootstrapping

The uncorrelated function point elements are orthogonal to each other which means that they have nothing in common. Hence, they measure different attributes of a software project. The lack of orthogonality causes correlated function point elements which create new dimensions in the statistical properties analysis of the function point elements. Correlated function point elements suggests that same attribute of a software project is counted more than once and there is a degree of influence of function point elements over each other; therefore, understanding the correlation among FP elements is critical. This phenomenon can be studied with the correlation among function point elements. Previous studies [1-3], have noted the correlation among function point elements and have reported the observed correlation coefficients. This study takes a step forward and focuses on to understand the nature of the correlation among function point elements by estimating the possible distribution of correlation coefficients. Therefore, instead of having a single value quantification of the correlation, which does not quantify the stated objective; this research study promotes a histogram view of the correlation coefficients.

Therefore, to get a deep understanding of the correlation among function point elements advanced statistical analysis techniques are needed such as bootstrap analysis of the correlation coefficients.

Bootstrap is a procedure to estimate the sampling distribution of an estimator by sampling with replacement from the original sample. Bootstrap is often used with the purpose of estimating the mean, median and correlation coefficients of a data given data sample. The simulation of bootstrap involves choosing samples with replacement from a dataset for analysis. Sampling with replacement means that every sample is returned to the dataset after sampling so a particular data sample could appear multiple times in a given bootstrap analysis. The simulation process is repeated multiple times to get the sampling distribution of the correlation coefficients.

Function Point elements of the selected IFPUG dataset are analysed for correlations through bootstrap simulation which generates the distribution of the correlation coefficients among two function point elements. The obtained correlation coefficients values are plotted against the number of times that value is observed during the simulation. The process is repeated for all the pairs of function point elements, which produces histograms of correlation coefficients for each pair of function point elements. The histogram of the correlation coefficients are interpreted as follows: two function point elements have no correlation when the correlation histogram is centred around 0 and have weak correlation when the histogram is centred around any value less than 0.5. Whereas, function point elements are strongly correlated when the histogram is centred on any value greater than 0.5. The correlation histograms of each pair of function point elements are shown in Figure 3.

The external input (EI) function point element shows weak correlation with external output (EO) and external inquiry (EQ) function point elements while it shows strong correlation with internal logical files (ILF) function point element and shows no correlation with external logical files (EIF) function point element.

Whereas, external output (EO) function point element shows weak correlation with external inquiry (EQ) and internal logical files (ILF) function point element and no correlation with external interface files (EIF) function point element.

External inquiry (EQ) function point element shows weak correlation with internal logical files (ILF) and no correlation with external interface files (EIF).

While, the external interface files (EIF) function point elements shows no correlation internal logical files (ILF) function point element.

These results suggest that external input (EI) and internal logical files (ILF) are strongly correlated; while external interface files (EIF) are not correlated at all with any function point elements. While external input (EI) and internal logical files (ILF) have some correlations with external output (EO) and external enquiry (EQ) function point elements. These findings confirm the results of Chris [1].

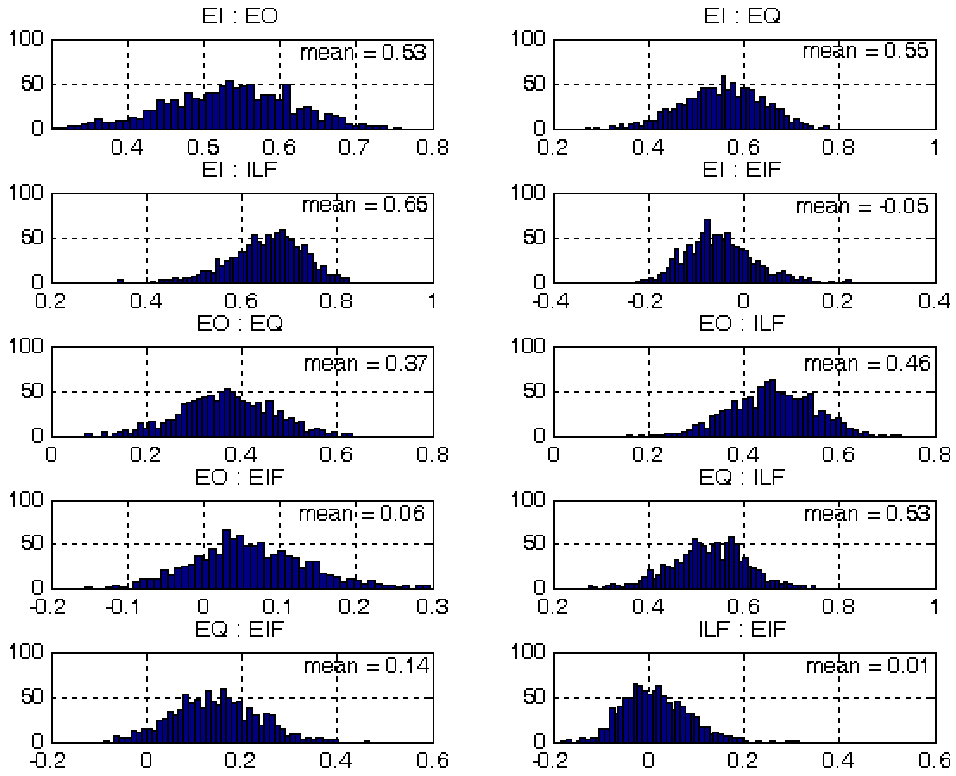

**Figure 1. Function Point Correlations through Bootstrap Analysis**

## 5. Conclusions

In-depth function point correlation analysis is presented with some interesting observations. The critical observation which came out of this research study is that external input (EI) and internal logical files (ILF) measures the significant portion of the same attribute of software development projects. Therefore, software size based on the function point elements may have the effects of double counting. Software practitioners should consider the correlation among function point elements when measuring the software size using function point elements.

## References

[1] C. J. Lokan, “An Empirical Study of the Correlation between Function Point Elements”, Software Metrics Symposium, Sixth International Proceedings, **(1999)**, pp. 200-206.

[2] D. R. Jeffery and J. Stathis, “Function Point Sizing: Structures, validity and applications”, Journal of Empirical Software Engineering, **(1996)**, pp. 11-30.

[3] B. Kitchenham and K. Kansala, “Intr.-item correlations among function points”, In Proceeding 15th International Conference on Software Engineering, IEEE, **(1993)**, pp. 477-480.

[4] “International Software Benchmarking Standards Group”, no. 9.

[5] “IFPUG. Function Point Counting Practices Manual”, no. 4.2

# Author

**Masood Uzzafer**, He has 20 years of experience of which 10 years in the high-tech software development industry and 10 years in research and academics. Dr. Masood has PhD from University of Nottingham, UK and he is also a PMP certified project manager. His research interests are Software cost estimation, project management and healthcare prediction models.