STATISTICAL RATING OF OPERATING SYSTEM SOFTWARE USING ONLINE SOFTWARE REPOSITORY DATA

Supported by tetfund

P. I. Ogunyinka[1]*, P. I. Ogunyinka[2], B. F. Iheanyichukwu[3], O. M. Ekundayo[4], Oladipupo Adekoya[5] and S. O. Folorunso[1]

[1]Department of Mathematical Sciences, Olabisi Onabanjo University, Ago-Iwoye, Nigeria

[2]Department of Information Technology, Apple Microfinance Bank, Oru-Ijebu, Ogun State, Nigeria

[3]Olabisi Onabanjo University Library, Olabisi Onabanjo University, Ago-Iwoye, Ogun State, Nigeria

[4]Department of Computer Science, Federal University of Technology, Akure, Ondo State, Nigeria

[5]Department of Computer Engineering & Built Environment, Birmingham City University, Birmingham, United Kingdom

*Corresponding author: ogunyinka.peter@gmail.com

Received: May 30, 2020 **Accepted:** September 05, 2020

Abstract: Operating System Software (OSS) had been rated with different yardsticks. However, this study had used new yardsticks. The yardsticks were estimated total software download and the Internal Software Download Error (ISDE) based on how these OSS visit the SourceForge Online Software Repository (OSR). The six OSS were Windows, Linus, Macintosh, Android, BSD and Solaris OSS. Two-phase sampling estimation method was used while the software download size based on the OSS type was used as the study variable (y) while the software filesize was used as the auxiliary variable(x). It was discovered that 70.51, 8.20, 6.30, 0.77, 0.08 and 0.04% FOSS were downloaded by Windows, Linux, Macintosh, Android, BSD and Solaris, respectively from the sourceforge OSR. Based on the computed percentage of total software download by the six OSS on the repository, Windows, Linux, Macintosh, Android, BSD and Solaris OSS were rated as first, third, fourth, fifth, sixth and seventh most used OSS, respectively. The computed Percentage Coefficient of Variation (PCV) was used to examine the ISDE of FOSS on these OSS. Based on the ISDE, Windows, Linux, Macintosh, Android, BSD and Solaris were rated as first, fourth, fifth, seventh, second and third, respectively. It was concluded that Windows was the most used and the most consistently used OSS. It was recommended that Free and Open Source Software (FOSS) developers should invest into the development of FOSS that use Windows while FOSS developers for other OSS should invest more into aggressive online marketing and promotion of such application software after development.

Keywords: Operating system software, online software repository, internal software download error

Introduction

Operating System Software is (OSS) the first software that is installed on every modern computer system. It hosts other software as it serves as the foundational software for all other types of application software, utility software, among others. OSS manages the resources on the computer system or a computer network. It also gives the user the interface for accessing the hardware and other software installed on the computer system. The OSS depends on the hardware while the applications and networks depend on the OSS. Similarly, the hypertext browser in the W^3 model also depends on the OSS (Berners-Lee *et al.*, 1992). Fig. 1 shows the conceptual model of a computer system. Alhassan and Bach (2014) concluded that OSS has wide range of definitions but summarized OSS as the spirit and mind that brings life to hardware. Silberschatz *et al.* (2003) highlighted the functions of OSS as process management, central processing unit scheduling, memory management, file system management, Input/Output (I/O) system management, communication management and network management.

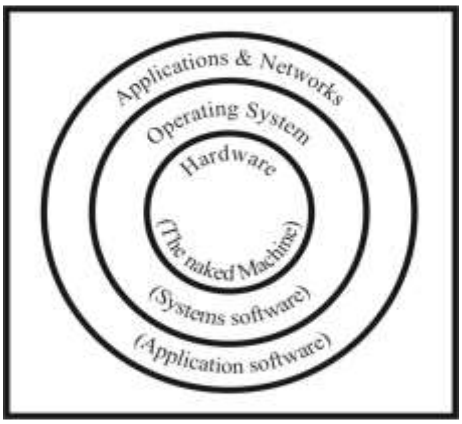

Fig. 1: The conceptual model of a computer system

The list of OSS is not limited to Windows XP, Windows Vista, Windows 7, Windows 8, Windows 10, Windows 10 Mobile, Android, Apple Macintosh, Contiki, Unix, Google Chrome, Free Berkeley Software Distribution (BSD), iOS, Blackberry and Linux. Just like software generally, literature has clarified OSS based on the types of license (Free or Proprietary OSS), device (mobile or computer system) or network OSS. However, attention has been channeled on Free and Open Source (FOS) Operating system software. This includes Linus and Unix.

Software comparison could help in identifying the strength of one OSS over the other. It could, also, help the developers and consumers of such OSS in making reasonable decisions. Chen *et al.* (1996) compared Windows for workgroups, NetBSD (of Unix OSS) and Windows NT OSS using micro-benchmarks. These micro-benchmarks used intel'spentinium processor hardware counters. The study concluded that Windows for workgroup is most expensive in accessing system functionality. However, the application workload benchmark revealed that the subsystems (like graphics sub-system and buffer Cache sub-system) functionalities determined the general performance of these concerned OSS. Koopman *et al.* (1997) developed a 5-point robustness benchmarks for the comparing OSS. These benchmarks were Catastrophic, Restart, Abort, Silent and Hindering (summarized as CRASH). Al-Rayes (2012) compared Linux and Windows OSS with respect to cost, strategic IT choices, application availability, IT staff skills and competencies, company standards, performance and application deployment. It was concluded that while Linux OSS may be Free and Open Source Software (FOSS), the trade-off of Linux over Windows may not be obvious. Jindal and Jain (2012) compared Google Android, Symbian and Apple iOS using history, application development of App Store, programming of Software Development Kit (SDK) and reliability and security of the OSS. It was concluded that Symbian OSS had significant gap to cover, the Google Android OSS enjoyed the benefits of FOSS community while the Apple iOS was the ever enhancing OSS. Reusing (2012) compared Tiny OSS and Contiki networking OSS using the programming model, execution model, resource use, energy consumption, hardware platforms and tool chain benchmarks. It was concluded that Contiki OSS is a good decision if flexibility is under consideration while Tiny OSS is the better decision when resources may be scarce. Gupta *et al.* (2013) compared Microsoft Windows 7 and 8 OSS using OSS performance features as benchmark. Such features included startup time, wakeup time, shutdown time, 3D graphics and multi-media performance time, application opening and file copying times. It was revealed that Windows 8 significantly out-performed Windows 7 OSS.

Chim *et al.* (2013) compared Linux and Windows OSS using factors like cost, strategic IT choices, IT staff skills and competencies, application deployment, general performance, application availability and company standards. It was observed that Linux OSS could be FOSS OSS but the cost of associated drivers could be high. It was, finally, concluded that the “risk/return” trade-off of Linux to Windows could be very negligible as against first appearance of cost effectiveness. Alhassan and Bach (2014) compared Windows, Unix, Macintosh and Linux OSS with the aim to know factors that influence OSS users to purchase OSS. Investigation revealed that Windows OSS drives fifty percent of the OSS market sales while Macintosh and Linus OSS drive the remaining fifty percent of the OSS market sales. Further investigation also revealed that when OSS security and stability are considered, Mac and Linux OSS are rated high while Windows and Mac OSS are rate high when OSS application and availability are considered. It was concluded that OSS users’ requirement and users’ intention for OSS are the major factors for acquisition of OSS. Joshy and Ramadas (2016) had a survey and comparative study of TinyOS, Mantis and SOS. The comparison was justified based on the advantages, disadvantages and the most widely used OSS. Finally, Padhya *et al.* (2016) compared Android, iOS, Blackberry and Windows OSS. The benchmarks used included developer website, number of programming language, license (FOSS or Proprietary), App store, number of application, side loading, battery usage, customizability, security and voice assistance. However, it was concluded that Blackberry and Windows phone OSS are behind in the comparative study.

w3schools.com (w3schools, 2020), since March 2003, collects and reports the OSS statistics of every device that visits the website. It does counting and percentage comparison of Windows, Mac, Chrome OSS and Mobile OSS. Report as at February 2020, showed that Windows 10, Windows 8, Windows 7, Windows Vista and Windows XP take 59.1, 3.5, 9.8, 0 and 0.2%, respectively. Linux, Mac, Chrome OSS and Mobile OSS account for 5.9, 9.9, 0.0 and 11.4%, respectively. Google trend (Google Trend, 2020) uses geographical search data to accounted for 100, 74, 1, 16% and <1% popularity for Windows, Android, Mac, Linux and Solaris OSS, respectively. Fig. 2 represents the google trends result as at March 28, 2020.

Literature have compared different OSS based on different benchmarks. Survey revealed summary of OSS comparison benchmarks as reliability, performance, security, file system, device drivers, commercial applications, free applications, development environment or developers community, development infrastructure, type of license and support cost of ownership, among others. However, none has made comparison considering the Internal Software Download Error (ISDE) by the OSS benchmarks. The consistent download assessment by OSS on OSR could be used to assess the ISDE. The consistent download assessment determines the users’ quality download attention produced by the OSS through download of such OSS. This study would compare different OSS based on the estimated total software download and ISDE.

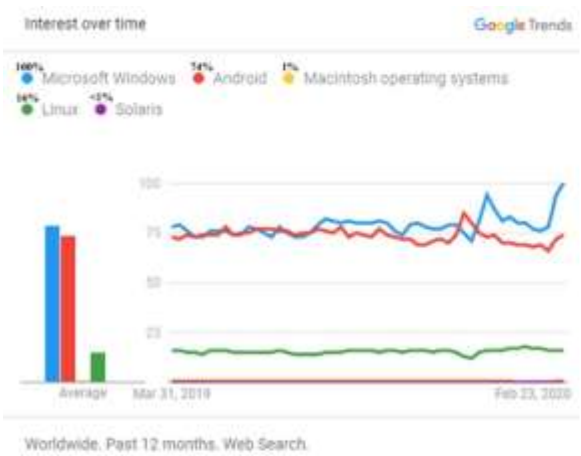

Fig. 2: OSS popularity by Google Trends

Materials and Methods

Review of two-phase sampling in survey statistics

The presence of auxiliary information has been proved to be significantly relevant in survey statistics. Auxiliary information can be utilised at the pre-sample selection, sample selection and the post-sample (estimation) stages for the estimation of study variable(s). Auxiliary information could either be quantitative (defined as auxiliary variable) or qualitative (defined as auxiliary attribute) characteristic. Graunt (1662) was assumed to be the pioneer Statistician that implemented auxiliary information in the estimation of England population. However, Bowley (1926) and Neyman (1934, 1938) were assumed to be the certified Survey Statistics studies that officially used auxiliary information. Neyman (1938) was the first to describe double sampling method for efficient estimation of human population. However, Keen (2005) reported on the reason that led to change of *double sampling* to *two-phase sampling.* Keen emphased that *double sampling* is a choice of terminology that describes a method in Statistical Quality Control. Hence, having the same terminology in Survey Statistics could mislead the audience.

In two-phase sampling, the first phase sampling with size n_1 is taken from the population of size N. Only information on the auxiliary variable is collected at the first phase sampling. A second phase sampling with size n_2 is conducted to obtain auxiliary information and the corresponding study variable information, such that $n_1 > n_2$. The second phase sample could be a proper subset of n_1 (nested two-phase sampling) or subset of N (non-nested two-phase sampling). Hidiroglou (2001) has reported the efficiency of nested two-phase sampling over non-nested two-phase sampling.

Two-phase sampling had been used with ratio and regression estimators to form two-phase sampling for ratio and regression estimators, respectively. Following the simple regression model,

$$y = \alpha + \beta x,$$

Where: y and x are study and auxiliary variables, β is the regression coefficient and α is the interception on y axis.

Table 1 shows the conditions for the use of ratio and regression estimators. In addition to the conclusions, two-phase sampling can be used when the population parameter of the auxiliary variable is not available or almost impossible to obtain due to cost.

Two-phase sampling for regression is presented as,

$$\bar{y}_{dl} = \bar{y}_2 + \hat{\beta}(\bar{x}_1 - \bar{x}_2). \quad (1)$$

Where: $\bar{y}_2$ is the sample mean of the study variable obtained at the second phase sampling, $\bar{x}_1$ is the sample mean of the auxiliary variable obtained at the first phase sampling, $\bar{x}_2$ is the sample mean of the auxiliary variable obtained at the second phase sampling and $\hat{\beta}$ is the estimated regression coefficient of y on x. $\bar{y}_{dl}$ is the unbiased estimator of the study variable.

However, the minimized Mean Square Error (MSE) of $\bar{y}_{dl}$ is presented as

$$MSE(\bar{y}_{dl}) \cong \theta_1 S_y^2 + \theta_2(S_y^2 + \hat{\beta}^2 S_x^2 - 2\hat{\beta}S_{yx}). \quad (2)$$

Where: $\theta_1 = \left(\frac{1}{n_1} - \frac{1}{N}\right), \theta_2 = \left(\frac{1}{n_2} - \frac{1}{n_1}\right)$, S_y^2 is the variance of the study variable, S_x^2 is the variance of the auxiliary variable and S_{yx} is the covariance of y and x.

Using logarithm non-linear data transformation, the study variable and the auxiliary variable would be $y_i^* = log_{10}y_i$ and $x_i^* = log_{10}x_i$, respectively. Hence, the transformation of equation (1) will be,

$$\bar{y}_{dl}^* = \bar{y}_2^* + \hat{\beta}^*(\bar{x}_1^* - \bar{x}_2^*). \quad (3)$$

The corresponding estimated minimized MSE of equation (3) is presented as,

$$M\hat{S}E_{min}(\bar{y}_{dl}^*) \cong \theta_1 \hat{S}_y^{2*} + \theta_2(\hat{S}_y^{2*} + \hat{\beta}^{2*}\hat{S}_x^{2*} - 2\hat{\beta}^*\hat{S}_{yx}^*). \quad (4)$$

Where: $\hat{S}_y^{2*}$ is the estimated variance of the transformed study variable, $\hat{S}_x^{2*}$ is the estimated variance of the transformed auxiliary variable, $\hat{\beta}^*$ is the estimated regression coefficient of the transformed study variable on the auxiliary variable and $\hat{S}_{yx}^*$ is the estimated covariance of the transformed y and x.

Consequently to the data transformation, the unit of measurement of the study variable would have been distorted. To obtain the actual unit of measurement for equation (3), back transformation would be conducted to obtain

$$\bar{y}_{dl}' = 10^{(\bar{y}_{dl}^*)}. \quad (5)$$

The estimated population total is presented as,

$$\hat{Y}_{dl} = N\bar{y}_{dl}', \quad (6)$$

and the estimated MSE of $\hat{Y}_{dl}$ is presented as,

$$M\hat{S}E_{min}(\hat{Y}_{dl}) = N^2 M\hat{S}E_{min}\left(\bar{y}_{dl}'\right), \quad (7)$$

Where: ***N***= Population size

The Coefficient of Variation (CV) is a statistical measure of variability for the experiment in different units of measurements. The CV has the major advantage of converting experiment to a dimensionless output. Hence, it facilitates easy comparison of different experiments. Manuel (2013) documented a report on the creation of CV by Croxton *et al.* (1967). The CV is presented as,

$$PCV = \frac{\sqrt{M\hat{S}E}}{Mean} = \frac{\sqrt{M\hat{S}E_{min}(\bar{y}_{dl}^*)}}{\bar{y}_{dl}^*}, \quad (8)$$

Where: $M\hat{S}E$ = estimated Mean Square Error

However, the Percentage Coefficient of Variation (PCV) presents the CV in percentage. It is defined as,

$PCV = CV * 100\%$.

Table 1: Conditions for the use of ratio, regression, product and difference estimators

Estimators	Auxiliary Characteristics	Correlation Coefficient (ρ)	Linearity assumption	Intercept on y axis (α)	Value of regression coefficient (β)
Ratio	Must be available	Positive and high	Must be obeyed	Must be zero	Not applicable
Regression	Must be available	Positive and high	Must be obeyed	Is not be zero	Not fixed

Table 2: The common statistical transformation techniques

S/N	Method	Transformation	Regression Equation	Predicted/Back transformation value ($\hat{Y}$)
01	Standard linear regression	None	$Y = b_0 + b_1X$	$\hat{Y} = b_0 + b_1X$
02	Exponential model	Dependent variable ($log_{10}Y$)	$log_{10}Y = b_0 + b_1X$	$\hat{Y} = 10^{(b_0+b_1X)}$
03	Quadratic model	Dependent variable ($Sqrt(Y)$)	$Sqrt(Y) = b_0 + b_1X$	$\hat{Y} = (b_0 + b_1X)^2$
04	Reciprocal model	Dependent variable (y^{-1})	$y^{-1} = b_0 + b_1X$	$\hat{Y} = 1/(b_0 + b_1X)$
05	Logarithm transformation	Independent variable ($log_{10}X$)	$Y = b_0 + b_1log_{10}X$	$\hat{Y} = b_0 + b_1log_{10}X$
06	Power model	Dependent variable $log_{10}Y$ and independent variable $log_{10}X$	$log_{10}Y = b_0 + b_1log_{10}X$	$\hat{Y} = 10^{(b_0+b_1log_{10}X)}$
07	Square model	Independent variable (X^2)	$Y = b_0 + b_1X^2$	$\hat{Y} = b_0 + b_1X^2$

Data transformation

Osborne (2002) established that the presence of outliers in the dataset could lead to the statistical violation including linearity assumption between y_i's and x_i's. Consequently to this violation would be increase in the probability of committing type-I or type-II error. Data transformation technique would be used to correct the effect of outliers in the data set (Ogunyinka and Badmus, 2014). The Logarithm ($\log_{10} y_i$ and $\log_{10} x_i$) non-linear data transformation method would be used in this study. Consequently to data transformation method is change in the unit of measurement. Hence, back transformation tool (Miller, 1984) would be a necessary tool to untransform analysed result back to its original unit of measurement. Miller, also, confirmed that back transformation method is associated with bias increment. The estimated $MSE_{min}(\bar{y}_{dl}^*)$ would significantly account for this bias.

Table 2 shows the non-linear data transformation and back transformation methods as presented by Ogunyinka and Badmus (2014).

Sourceforge open source repository (OSR) data collection procedure and data transformation

Among some of the transactions that regularly take place on sourceforge repository (www.sourceforge.net) include number of software raters, software average rating, software filesize, software download number based on visitor's country and software download number based on Operating System (OS) type. This study would make use of software filesize and software download size based on OS type. It was assumed that there is relationship between software filesize and software download size based on OS type. In this study, the auxiliary variable is represented as x which is the software filesize while the study variable, represented with y, is the software download size based on OS type. Data were mined on sourceforge repository between March 1, 2013 and April 31, 2014 for seven OSS using Okikisoft data miner (Sourceforge, 2014). Data mined included the software filesize and the software download size based on the OSS. The OSS considered were Windows, Linux, Macintosh, Android, BSD and Solaris OSS. Okikisoft mined transaction details for 1048 software on Sourceforge repository for Seven (including the unclassified) OSS. The software filesize variable was in different units (Byte, KiloByte (KB), MegaByte (MB) and GigaByte (GB)). However, this study converted the MB, GB and the byte into KB for the purpose of computation. The seven OSS were Windows, Android, Linus, Macintosh, Solaris, BSD and Unknown OS. In this study,

Figure 3 shows the graphical analyses on the original data and the transformed data on the obedience of linear assumption. Figure 3a confirms that the original data violated the linearity assumption. Applying non-linear data transformation to the two variables "software download size based on OS type" (represented with y) and "software filesize" (represented with x) revealed that the nonlinear transformed variables obeyed the linearity assumption and significant coefficients of determination were obtained (Fig. 3b). This study would obtain the sample mean and the estimated population total of software download size based on OS type. The MSE and the Percentage Coefficient of Variation (PCV) would be estimated to represent the Internal Software Download Error (ISDE) for each of the seven OS types. The estimated ISDE would be used to access the consistent use of the OS on the repository. Finally, the seven OS types would be ranked based on the computed ISDE.

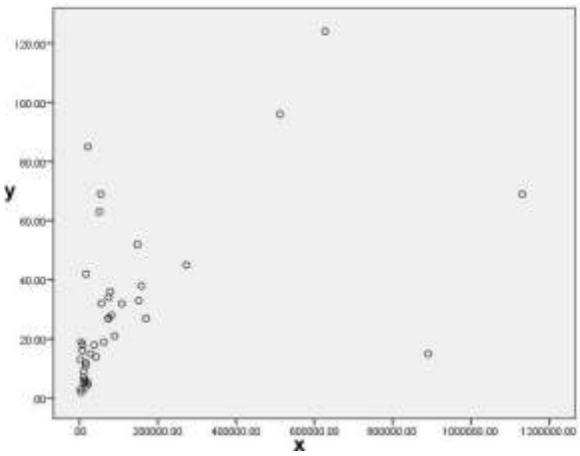

Fig. 3a: Graph of y against x (Original data)

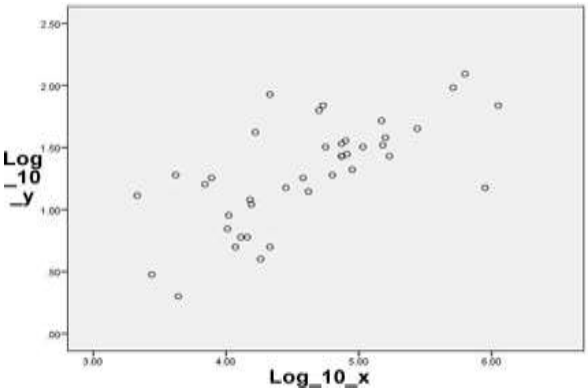

Fig. 3b: Graph of $Log_{10}y$ against $Log_{10}x$ (transformed data)

Estimation using Two-phase sampling for regression method

Figure 3a shows the violation of linearity assumption by the original data while Fig. 3b shows how the transformed data conformed to linearity assumption. Table 3 shows the analyses results for the six considered OSS. Two-phase sampling for regression estimation method had been used to obtain the sample mean $\left(\bar{y}'_{dl}\right)$, estimated population total $\left(\hat{Y}_{dl}\right)$, Mean Square Error (MSE) of the estimates and the Percentage Coefficient of Variation (PCV). Tables 4 and 5 show the results of the analyses and the corresponding ratings of the six OSS.

Results and Discussion

The aim of this study is to rate OSS based on the consistent FOSS download by OSS on the Online Software Repository (OSR). The OSS considered were Windows, Linus, Macintosh, Android, BSD and Solaris OSS. Data mined included the software filesize and the software download size based on the OSS. A relationship was assumed between the software filesize and software download size based on the OSS. The software filesize was used as the auxiliary variable (x) while the software download size based on the OSS was used as the study variable(y). It was confirmed that the variables violated the linearity assumption. Hence, nonlinear data transformation method was used to correct the violated assumption. However, back transformation which was used to restore the transformed data back to its original unit of measurement was associated with increase of bias in the estimates. Since the bias is observed in all the estimates and the focus is on the rating of the OSS, this study has decided to use the percentage coefficient of variation to obtain the rating for each OSS. However, the PCV was used to rate the OSS based on the consistent FOSS download by the OSS on the sourceforge OSR.

The first and second phase sample sizes ($n_1$and n_2, respectively) and the population size (N) for each of the OSS types are displayed in Table 3. Similarly, Table 3 shows the computation of the sample means of the auxiliary variables for both first $(\bar{x}_1)$ and second $(\bar{x}_2)$ phase sampling and the sample mean of the study variable $(\bar{y}_2)$ at the second phase sampling. It also shows the computed sample mean of the study variable $(\bar{y}^*_{dl})$ using the transformed data and the computed sample mean of the study variable $(\bar{y}_{dl})$ after effecting back transformation. Similarly, the estimated population total $\left(\hat{Y}_{dl}\right)$ after effecting back transformation.

Table 3: Two-phase sampling for regression analyses for the six OSS using $n_1 = 114$and $n_2 = 50$

OSS	Windows	Linux	Macintosh	Android	BSD	Solaris	Unknown OSS
N	84497	34013	15307	1021	9342	1013	2134
$\bar{y}_2$	4.0964	3.1538	3.0416	2.1334	1.1095	0.8753	3.3965
$\bar{x}_2$	3.6720	3.6364	3.6806	3.6543	3.5184	3.6249	3.6687
$\bar{x}_1$	3.7468	3.7082	3.7276	3.7161	3.6565	3.7032	3.7224
$\bar{y}_{dl}$	4.1027	3.1684	3.0535	2.1404	1.1501	0.8634	3.4037
$\bar{y}^*_{dl}$	12669.0492	1473.7064	1131.1326	138.1610	14.1291	7.3015	2533.1359
$M\hat{S}E_{min}(\bar{y}^*_{dl})$	0.0081	0.0106	0.0150	0.0197	0.0081	0.0102	0.0168
$(\hat{Y}_{dl})$	1,070,496,653	124,523,770	95,577,308	11,674,189	1,193,868	616,954	214,042,384
$M\hat{S}E_{min}(\hat{Y}_{dl})$	57828846.48	75923086.82	106772251.5	140878185.4	58138960.68	72637158.92	119793288.1

Table 4: Ranking of OSS based on the estimated total download of FOSS on SourceForge OSR

OSS	Windows	Linux	Macintosh	Android	BSD	Solaris	Unknown OSS
$(\hat{Y}_{dl})$	1070496652.59	124523770.37	95577308.25	11674189.44	1193868.35	616953.82	214042383.71
$\%(\hat{Y}_{dl})$	70.51%	8.20%	6.30%	0.77%	0.08%	0.04%	14.10%
$\text{Rank}(\hat{Y}_{dl})$	1	3	4	5	6	7	2

Table 5: Ranking of OSS based on the estimated PCV for OSS on SourceForge OSR

OSS	Windows	Linux	Macintosh	Android	BSD	Solaris	Unknown OSS
$\bar{y}_{dl}$	4.1027	3.1684	3.0535	2.1404	1.1501	0.8634	3.4037
$M\hat{S}E_{min}(\bar{y}^*_{dl})$	0.0081	0.0106	0.0150	0.0197	0.0081	0.0102	0.0168
$P\hat{C}V(\bar{y}^*_{dl})$	2.2%	3.3%	4.0%	6.6%	7.8%	11.7%	3.8%
$\text{Rank}\left(P\hat{C}V(\bar{y}^*_{dl})\right)$	1	2	4	5	6	7	3

The estimated population total download $(\hat{Y}_{dl})$ is shown in Table 4. Figs. 4 and 5 shows that approximately 71% (1,070,496,653 estimated total downloads) of the devices that visit sourceforge OSR uses Windows OSS, approximately 8.2% (124,523,770 estimated total downloads) of the devices that visits this OSR uses Linux OSS, 6% (95,577,308 estimated total downloads) uses Macintosh OSS, 1% (11,674,189 estimated total download) uses Android OSS while 0.1% (1,193,868 estimated total download) uses BSD OSS and 0.04% (616,954 estimated total download) uses Solaris OSS. It was observed that 14.10% (214,042,384 estimated total download) of the OSS that visit sourceforge OSR could not be identified. This implies that majority of the visitors to sourceforge OSR uses Windows OSS and would probably prefer to download FOSS that are Windows OSS based. It could be observed that even when Linux OSS is a free and open source OSS; it is yet to gain popularity among the OSR users. This study identifies that 14.10% unidentified OSS is high. This requires that the sourceforge administrator should improve on the OSR to identify more of the OSS that visit the OSR.

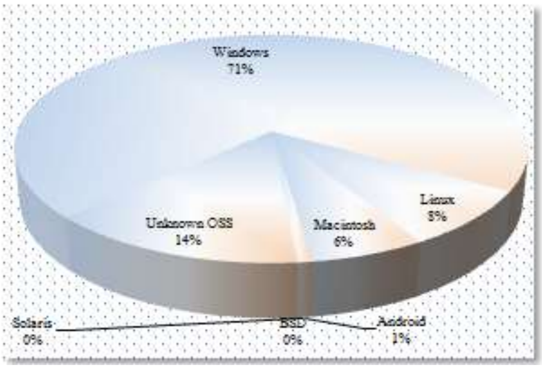

Fig. 4: Pie Chart of the estimated download total

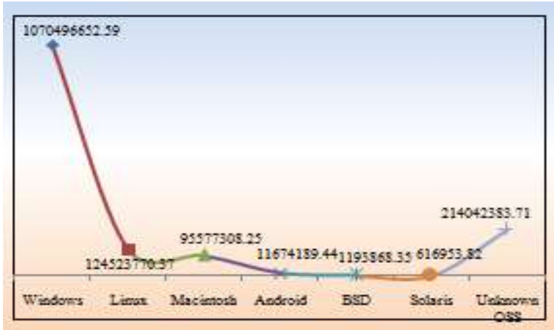

Fig. 5: Line Chart of the estimated download total

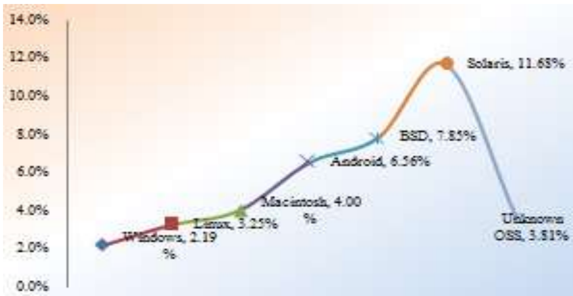

Fig. 6: Line Chart of the estimated PCV for the six OSS

The rating of the OSS based on the consistent download assessment of OSS on the sourceforge OSR was done using the estimated PCV. Fig. 6 shows the distribution of the estimated PCV used for the Internal Software Download Error (ISDE) for the six OSS. Windows OSS has the least estimated OSS ISDE of 2.19% PCV while Solaris OSS had the highest ISDE of 11.68% PCV. Table 5 shows the rating of the OSS based on the estimated OSS ISDE. The lower the PCV (or ISDE), the less consistent is such OSS to the download of software on sourceforge OSR and vice versa. Windows OSS is the most consistent OSS for FOSS download while Solaris is the least consistent OSS for FOSS download on sourceforge OSR. Windows, Linux, Macintosh, Android, BSD and Solaris

OSS were rated as first, second, fourth, fifth, sixth and seventh OSS while the unidentified OSS was rated third most consistent OSS on sourceforge OSR.

Consequently to the analyses results on the rating of OSS based on the estimated download size and the estimated consistent download assessment on these OSS, the following conclusions could be inferred:

a. Windows OSS had the highest total download and the most consistent download of FOSS on sourceforge OSR.
b. While Linux OSS had third most total download of FOSS but had the second most consistent download of FOSS on sourceforge OSR.
c. Windows OSS had the significant total download and consistent download of FOSS than other competitive and popular free and open source OSS like Linux and Android OSS based FOSS. Consequently, it is recommended that FOSS developers should invest into Windows OSS based FOSS. It would require less investment to market and promote such Windows based FOSS. On the other hand, Linux and Android based FOSS developers should incorporate the cost of marketing and promotion in the cost of such FOSS production in order for such FOSS to receive quality attention from the users.
d. Wikipedia (2020) reported that the primary revenue model for SourceForge is through placement of advertising banner sales on the webpages. For sourceforge OSR to make more money through this model, this study recommends that more adverts should be placed on the webpages of the Windows OSS based FOSS on the OSR.

Sourceforge OSR, as at the time of mining these data from the repository, had high percentage (14%) of OSS that visit the OSR which were not classified. There is need for the OSR administrator to upgrade the OSR to account for other type of OSS that visits the repository.

Conclusion

This study had rated six OSS (Windows, Linus, Macintosh, Android, BSD and Solaris OSS) based on the estimated total download of FOSS and consistent download assessment of FOSS on sourceforge OSR. Windows OSS based was rated as the most consistent OSS in download FOSS. It was concluded that Windows based OSS is more consistent than other competitive and popular free and open source operating system software like Linux and Android OSS. It was recommended that FOSS developers, irrespective, of OSR, should invest into Windows based FOSS. However, Linux and Android OSS based FOSS developers should add software marketing and promotion cost to the total software production cost in order for such FOSS to gain popular use. Similarly, sourceforge and other OSR administrators were advised to upgrade their OSR to be able to identify more OSS that may visit the OSR.

Conflict of Interest

Authors have declared that there is no conflict of interest reported in this work.

References

Alhassan A Hussain & Bach Christain 2014. Operating systems and decision making. *Book of Proceedings of ASEE Zone1 Conference*, pp. 3-5.

Al-Rayes Hadeel Tariq 2012. Studying main differences between linux and windows operating systems. *Int. J. Electrical and Comp. Sci.*, IJECS-IJENS, 2(4): 25-31.

Berners-Lee T, Cailliau R., Groff J-F & Pollermann B. 1992. World-wide web: The information universe. *Electronic Networking Research, Applications and Policy*, 2(1): 52-58.

Bowley A 1926. Measurements of precision attained in sampling. *Butt. Inst. Int. Statist*., 22: 1-62.

Chen J Brandley, Endo Yauhiro, Chan Kee, Mazieres Davido, Dias Antonio, Seltzer Margo & Smith D Michael 1996. The measured performance of personal computer operating systems. *ACM Transactions on Comp. Sys.*, 14(1): 3-40.

Chim Monali, Rathod Vasundhara & Chawan Pramila 2013. Linux and windows operating systems. *J. Engr., Comp. and Appl. Sci.* (JEC&AS), 2(6): 15-20.

Croxton FE, Crowden DJ & Klein S 1967. Applied General Statistics, 3rd Edition, Prentice-Hall, New-York, p. 754.

Google Trends 2020. https://bit.ly/2WQmGaY, retrieved on March 28, 2020.

Graunt J 1662. Natural and Political Observations made upon the Bills of Mortality. London: Martin.

Gupta Kalpana, Kakkar Barkha & Sexana VivekSen 2013. A comparative study of two operating systems: Windows 7 and Windows 8. *Int. Refereed Multidis. J. Contem. Res.*, 1(2): 95-106.

Hidiroglou MA 2001. Double sampling. *Survey Methodology Statistics, Canada* 27(2): 143-154.

Jindal Gaurav& Jain Mayank 2012. A comparative study of mobile phone's operating systems. *Int. J. Comp. Applic. and Infor. Techn.*, 1(3): 10-15.

Joshy Athira & Ramadas Saranya 2016. A comparative study on network operating systems. *IOSR J. Comp. Engr.* (IOSR-JCE), ISSN: 2278-8727: 35-41.

Keen KJ 2005. Two-phase sampling. *Encyclopedia of Bio-Statistics* 8: 1 – 4. Doi:10.1002/047001815.b2aO5O94.

Koopman Philip, Sung John, Dingman Christopher, Siewiorek Daniel & Marz Ted 1997. Comparing Systems Using Robustness Benchmarks.

Manuel Martinez-Pons 2013. A coefficient of variability. *J. Math. and Stat.*, 9(11): 62-64, dou:10.3844/jmssp.2013.62.64.

Miller D 1984. Reducing transformation bias in curve fitting, *The American Statistician*, 30(2): 124-126.

Neyman J 1934. On the different two aspect of representative methods: The method of Stratified Sampling and the method of purposive selection. *J. Roy. Statist. Soc.*, 97: 558-606.

Neyman J 1938. Contribution to the theory of sampling human populations. *J. Amer. Statist. Assoc*., 33: 101-116.

Ogunyinka Peter I & Badmus Idowu 2014. Efficient linearity of online software repository variables. 5th series of proceedings. *Int. Conf. on Sci., Techn., Educ., Arts, Mgt. and Soc. Sci*. Isteams Research Nexus Conference, AfeBabalola University, Ado-Ekiti, Nigeria, pp. 453-458.

Osborne Jason 2002. Notes on the use of data transformations. *Practical Assessment, Research and Evaluation*, 8(6).

Padhya Bhargavi, Desai Prasad & Pawede Dipti 2016. Comparison of mobile operating systems. *Int. J. Innovative Res. Comp. and Commun. Engr.*, 4(8), Doi:10.15680/IJIRCCE.2016.0408132, 15281-15286.

Reusing Tobias 2012. Comparison of Operating Systems TinyOS and Contiki. Network Architectures and Services. Doi:10.2313/NET-2012-08-2_02.

Silberschatz A, Galvin P & Gagne G 2003. Applied Operating System Concepts. First Edition, John Wiley and Sons Inc., Danvers MA.

Sourceforge 2014. OkikiSoft Download. http://sourceforge.net/projects/okikisoft/?source=directory

w3schools 2020. OS Platform Statistics. https://www.w3schools.com/browsers/browsers_os.asp, Retrieved on March 28, 2020.

Wikipedia 2020. Source Forge. https://en.wikipedia.org/wiki/SourceForge, Retrieved on March 28, 2020.