Survey of University Server Software

Chris Sobczak

April 18, 2021

1 Abstract

Universities, especially in the United States have consistently increased the cost of attendance for their students with no transparent cause. Some investigation and speculation includes administrative costs, tenure, research expenses and increasing security threats (Ehrenberg 2009). A nominally large amount of money is spent on proprietary software licenses for services like Microsoft Office 360, SAS, SPSS and licenses for their own web services like Microsoft Express, Red Hat Enterprise Linux, and other server infrastructure. The goal of the survey is to determine the market share that open source server software occupies in the academic setting. The figure to be estimated is the proportion of university web services licenses. Are universities taking advantage of the secure, free, and open source industry standard technologies like nginx and Apache? This study target population includes all computers that universities run for their websites, mail servers and other digital services that provide a portal for students and members of the public to access. The main focus of the study is on those public domain services, and not services like ftp servers and other internal affairs.

2 Background

Open source software or "free and open source software" (FOSS) is a piece or package of software that is distributed with the source code available for auditing and editing. The statistical software R is a great example of a flourishing FOSS project where users are able to contribute to the project and create new packages. Another example is the OpenBSD operating system. In the case of OpenBSD and other FOSS operating systems, the public has the ability to know exactly what is going on behind the scenes. This is not to say that every user is required to read and understand the low level source code, but it is a principle that the project is transparent and the software community can audit it, creating a web of trust.

In contrast, there is no reason for a consumer to trust in the word of a company that distributes closed source software. This organization can claim whatever they want about the security and privacy of their software but without having the right to audit the source code, there is no reason to believe any of those claims. This raises security concerns in proprietary software since if the only way a software package is considered *secure* is by the fact that no one knows how it works except the producer, then as soon as the code becomes available (due to a leak), it can all be considered compromised. This happens constantly with the Microsoft Windows operating system, as we have seen most recently with the Solar Winds attack (Willett 2021). FOSS is secure through industry standard security schemes, cryptography and by the knowledge of the thousands of programmers who have access to it (compared to the relatively smaller number of people who work on a single companies development team).

Finally, maybe the more important reason for institutions, FOSS is also (most of the time) free of cost (in addition to "free as in freedom"). Universities and other public institutions spend millions of dollars each year on software licenses for their students, staff and web services and other infrastructure. Increasing costs for post-secondary education and healthcare administration costs can be attributed to some extent to these rising software licensing prices and the constant need for cybersecurity improvements. The increasing costs make higher education less accessible and harm the most vulnerable populations who rely on affordable healthcare. A very obvious way to reduce costs is be replacing these proprietary packages with equivalent and more secure, free, and open source software (Fitzgerald and Kenny 2004).

In addition to the problematic cost of closed source, proprietary software, these costly, licensed software packages are most vulnerable to security threats. Security should be a high priority of all organizations but especially at public institutions where all of the software that staff and students interact with poses a threat to those individuals. Providing one for profit company exclusive access to your staff and students, in the age of surveillance capitalism is quite reckless (Yeung 2018).

So this study, establishes a base-line estimate of the proportion of free and open source servers being used by universities around the world. The questions that are answered include: Are there specific countries whose universities use more open source servers than others? What kinds of services are associated with certain operating systems?

3 Methods

This survey was conducted using cluster sampling. The primary sampling unit (psu) being the name of an institution, and the primary unit of interest being the associated domain(s). Each of the schools highest level registered domain names are probed to extract all their associated subdomains, defining the secondary sampling units (ssus). A census was then taken of these ssus to determine the software run at each public domain name and calculate the proportion of the school's services that run open source software.

To provide an example, if Simon Fraser University (SFU) was draw into our sample, the unit of interest would be their registered domain name `sfu.ca`. Using the tool `findomain`, all of the subdomains under `sfu.ca` are collected into a file and probed to see if the service is up. Some example subdomains are `mailgate.sfu.ca`, `imapnew.sfu.ca`, and `canvas.sfu.ca` for the schools email services and canvas portal. This is only a very short list of some of the possible subdomains. It is common to have thousands of subdomains associated to the same top name for each school. Finally, filtering out all of the services that are not up, the header of the service is pulled using `curl`, which includes information about the software running at that address. The results are cleaned aggregated and proportions of the resulting categories are tabulated. Details on the tools I used in subsection 5.1 Tools.

For example, pulling the server information from `mail.sfu.ca` for SFU's mail service results in `Microsoft-IIS/10.0` as it is a Microsoft Exchange mail server. All responses including this, `MS`, `Win64`, and other server names that are licensed with anything except the *GPL*, *BSD*, *MIT*, or *Apache* licenses are considered proprietary, whereas those responses with the mentioned licenses are considered open source.

3.1 Survey Design

Using cluster sampling, a simple random sample (SRS) is taken of all the schools in the sampling frame (Hipo 2021). This sampling frame is an open source dataset of most of the universities in the world. Inevitably this list will have omitted some schools, but the set contains 9693 schools which is slightly on the low end of estimates, which may introduce bias in the conclusions of this study.

3.2 Sample Size Selection

The goal of this study is to estimate the proportion of university servers running an open source operating system using a 95% confidence interval with a margin of error of 0.03. Therefore, using Lohr 2019, "...surveys in which one of the main responses of interest is a proportion, it is often easiest to use that response in setting the sample size. For large populations, $S^2 \approx p(1-p)$, which attains its maximal value when $p = 1/2$. So using $n_0 = 1.96^2/(4e^2)$ will result in a 95% CI with width at most $2e$."

$$n_0 = \frac{z_{\alpha/2}^2 S^2}{e^2} = \frac{1.96^2(\frac{1}{2})(1-\frac{1}{2})}{0.03^2} \approx 1067$$

No need to use the finite population correction adjustment since the sample size is reasonable compared to the population size and the full dataset can be collected with a reasonable amount of resources.

3.3 Taking the Sample

With an appropriate sample size of $n = 1067$, using an R script to draw the sample, the selected school domains are saved in the `data.Rda` file found in the GitHub repository, along with `sample.R`, used to draw the sample. Within the sample, 1049 schools have only one registered domain, 16 schools have two registered domains and 2 have three domains.

$$N = 9693,\ n = 1067,\ m_0 \approx 2539008$$

The following 18 schools that were drawn in the sample have more than one domain name registered: Augusta University, University of Manchester, Universidad del País Vasco, Chinju National University of Education, Royal Holloway and Bedford New College, Northeastern University, University of the Pacific, Kwangju University, Kwangwoon University, University of Massachusetts at Lowell, St. Mary's University, University of Essex, Chonnam National University, Hanshin University, Savannah College of Art and Design, University of Technology Sydney, Universitat Pompeu Fabra, and Kyungil University.

These 16 schools with two domains and 2 schools with three domains accounts for the total 1087 ($1067 + (16 \text{ duplicates}) + (2 \times 2 \text{ more duplicates}) = 1087$) domains in the `domains` file of the GitHub repository. This file contained a few duplicates and subdomains for the school, so some of the domains were repeated and some were just included within the schools `subdomains/` file, reducing it to 1081 files in `subdomains/`.

All domains were separated onto their own line of the `probing/domains` file and processed with `findomain`. When extracting the subdomains, the `gen-subdomains` script only processed 1079 domains, identifying an inconsistency. Three of these domains that were not processed were `aloma.edu`, `student.uts.edu.au`, and `www.clcmn.edu`. In the case of `aloma.edu`, this is just a typo in the sampling frame for the Alamo Colleges' domain, which naturally is corrected to `alamo.edu`. The next missing domain `student.uts.edu.au`, for the University of Technology Sydney in Australia which is just a subdomain for their website `uts.edu.au`, identified as a duplicate domain. Finally, `www.clcmn.edu` for Central Lakes College-Brainerd is another subdomain for their college that was already extracted from `clcmn.edu`, another duplicated domain.

In the GitHub repository, the file `domains` is the audited file containing all of the highest level domains for the sample that the full observational information could be extracted. Domains omitted from this file were unreachable as mentioned in the discussion. From this file, the domains belonging to the same school have been concatenated so that results can be organized by psu before calculating proportions.

4 Results

The full results dataset of proportion of servers at each psu is available on the GitHub as `proportions.csv`. The proportion of each license status is represented in the columns, in addition to the sum for the FOSS licenses to compare with the non-FOSS proportion. After the proportion is measured in each of the psus, the population estimate is calculated using the ratio estimator

$$\hat{\bar{y}}_r = \frac{\sum_{i \in S} M_i \bar{y}_i}{\sum_{i \in S} M_i} = 0.640260$$

with a standard error of

$$\mathrm{SE}(\hat{\bar{y}}_r) = \sqrt{\left(1 - \frac{n}{N}\right) \frac{1}{n\bar{M}^2} \frac{\sum_{i \in S} M_i^2 \left(\bar{y}_i - \hat{\bar{y}}_r\right)^2}{n-1}} = 0.0233870$$

4.1 Nonresponses

It is possible that the nonresponses are missing completely at random, but it is more likely that the nonresponses are missing at random, given covariates (Lohr 2019). This is more likely since it is possible that the unreachable schools and domains are correlated with other factors especially the location of the school (country). Figure 1 in subsection 5.3 Nonresponse per Country of the Appendix shows that the countries with the largest missing proportions of schools are those that are furthest from the location the study was conducted. It is also possible but less likely that the nonresponse is correlated with the actual license of software used (the response variable in this study), so we will be adjusting according to the "Missing at Random" scheme in Lohr 2019. Figure 2 validates this assumption, the countries with the highest proportions of FOSS software are generally of the East, with a mix down the list.

In order to compensate for the nonresponse schools in the sample, estimates are adjusted using weighting class adjustment using countries as auxiliary variables. Table 1 shows the sum of weights, sample size, and respondents for each class (subsection 5.2 Weighting Classes of the Appendix). Only countries with nonresponses in the class are show. Those not included in the picture maintain the same N/n weighting.

For each of the countries of the table, $\hat{\phi}_c$ is defined as

$$\hat{\phi}_c = \frac{\text{sum of weights of respondents in class } c}{\text{sum of weights of a full sample in class } c}$$

With c being the country, when the full sample was successfully returned, $\hat{\phi}_c = 1$, causing no impact on the calculated estimates for that country. The weight factor is then the reciprocal of $\hat{\phi}_c$, $\frac{1}{\hat{\phi}_c}$. The weight factor for countries not shown in Table 1 is N/n. The new weights are calculated with

$$\tilde{w}_i = w_i \sum_c \frac{x_{ci}}{\hat{\phi}_c}$$

Let $x_{ci} = 1$ if unit i is in class c, and 0 otherwise, so that the new weights are adjusted according to their class specific $\hat{\phi}_c$. Our estimate is then calculated using the estimator $\hat{\bar{y}}_{wc}$, defined as

$$\hat{\bar{y}}_{wc} = \frac{\sum_{i \in S} \tilde{w}_i y_i}{\sum_{i \in S} \tilde{w}_i} = 0.572150$$

The weighting class adjustment estimator reduces the proportion of FOSS software in use, suggesting that those countries that were less reachable showed an even lower proportion of FOSS software. This estimate for the proportions of licenses, although a slim majority, it is still surprisingly low. Considering the benefits of open source software, you would expect a university to use proprietary software only when the specific legacy software that they need to run can *only* be run on a proprietary platform.

Despite that, we see that a big proportion of universities prefer paying for the enterprise licenses and security risks, over free and secure alternatives. It shows that within academic institutions, there is room to change and save money by replacing the expensive software with free alternatives.

4.2 Discussion and Conclusion

Overall, the data supports the claim of the paper, that universities, like other public institutions have room to improve and save money on software and make better security choices. The goal of this study is to hopefully shed light on an opportunity for public institutions to improve their own services and operation through choice of software. The most common proprietary software found in the study is the Microsoft Express server, the Windows email server distribution, like the one run at `mail.sfu.ca`. Some FOSS alternatives to this include citadel, kolabcommunity, and Open-Xchange.

There are plenty more alternative schemes for running an email server like these, that could replace Express, it is just a matter of the information technicians of the school making the right software decisions. That is what it comes down to the most, *knowing* there is even a decision to make. The technology industry giants have such a strangle hold on the market and use malicious, anticompetitive practices that cause consumers and businesses to think that they have no choice except to pay for proprietary licenses (Caballero Gutiérrez 2015).

4.3 Bias

Although the sample size for the study is very large, some bias in the sample could contribute to errors in the conclusions. One possible source of bias is the sampling frame potentially not including *all* universities in the world. Inevitably, this is the case, however it seems reasonable to consider the sample draw from the frame is representative of the target population (244 schools from the US, 23 from Canada, 10 from Italy, and so on).

Another potential source of bias in this study includes a few technical limitations. First, the `findomain` tool relies on databases and other historical DNS records and methods to identify subdomains, but it could miss some, most likely missing newly registered subdomains which may not represent the true makeup of the schools services. Additional bias could be introduced with `curl`, when it is not able to make a connection, or the header of the document not included the required server information for this study, the response is counted as an `NA`. However, it is almost always the case, that these missed, and unidentified servers are internal services, observational units that are outside the scope of the target population, not considered in this study.

Finally, a few of the schools in the sample were unreachable for probing The most common reason for the nonresponse was the absence of a legitimate SSL certificate at any of the given domains or subdomains preventing the data from being collected with curl (*curl - SSL CA Certificates* n.d.).

In conclusion, the results of this study should be taken to show where institutions can save money and improve their services. More research should be done to study the nominal costs of the licenses used compared to other cost reducing options. Additionally, with the significant increase in cyber attacks on all institutions, but especially medical care providers (hospitals and clinics), the security advantages of FOSS are significant.

References

Caballero Gutiérrez, José Luis (July 2015). "Antitrust and high technology industries: Microsoft in the spotlight". In: *Journal of Intellectual Property Law & Practice* 10.10, pp. 798–799. ISSN: 1747-1532. DOI: `10.1093/jiplp/jpv118`. eprint: `https://academic.oup.com/jiplp/article-pdf/10/10/798/5171629/jpv118.pdf`. URL: `https://doi.org/10.1093/jiplp/jpv118`.

curl - SSL CA Certificates (n.d.). `https://curl.se/docs/sslcerts.html`. (Accessed on 04/17/2021).

Ehrenberg, Ronald G (2009). *Tuition Rising: Why College Costs So Much : With a New Preface.* eng. Harvard University Press. ISBN: 0674009886.

Fitzgerald, B. and T. Kenny (2004). "Developing an information systems infrastructure with open source software". In: *IEEE Software* 21.1, pp. 50–55. DOI: `10.1109/MS.2004.1259216`.

Hipo (2021). *university-domains-list*. `https://github.com/Hipo/university-domains-list`.

Lohr, Sharon L (2019). *Sampling: Design and Analysis*. eng. 2nd ed. Chapman & Hall/CRC Texts in Statistical Science Series. Milton: CRC Press. ISBN: 0367273462.

Willett, Marcus (2021). "Lessons of the SolarWinds Hack". In: *Survival* 63.2, pp. 7–26. DOI: `10.1080/00396338.2021.1906001`. eprint: `https://doi.org/10.1080/00396338.2021.1906001`. URL: `https://doi.org/10.1080/00396338.2021.1906001`.

Yeung, Karen (2018). "Five fears about mass predictive personalization in an age of surveillance capitalism". eng. In: *International data privacy law* 8.3, pp. 258–269. ISSN: 2044-3994.

5 Appendix

5.1 Tools

The study was conducted using a combination of R, and shell scripting. Whenever possible, the basic GNU utilities are preferred over more complex ones, as well as the scripting style being POSIX compliant instead of Bash specific. The following tools are those tools greater than the GNU basic utilities that were used for the study. All project source files can be found at this GitHub repository.

5.1.1 Extracting All SSUs From Each PSU

The tool `findomain` was used to collect all subdomains associated with a school. A simple `findomain -t $domain` can be run to collect a full set of the subdomains for the given `$domain`. However, as discussed, there may be limitations to any tool attempting to collect all of these associated subdomains. The R script outputs the base second and third level subdomains into a file with one domain per line, for which I run `findomain -t`. The tool takes a domain and searches various databases and tests the domain for subdomains associated with it. I take this and output all the subdomains into a file for each school, and then test if the service is up.

5.1.2 Probing Hosts

In order to save time in the data collection phase, I first validated the output given from `findomain` by probing each provided domain with the DNS lookup utility `dig`. This tool checks a DNS server to see if the given domain actually exists and if the host is up. Without this step, probing from each of the SSUs would take much longer.

5.1.3 Extracting Info From SSUs

The `probe-knownhosts` scripts are shell scrips that run the following curl command on each line of each psu file, collecting the server data.

```
curl -I -s --connect-timeout 2 $domain
```

This outputs the header of the page that might look like the following from `mail.sfu.ca`:

```
HTTP/1.1 302 Moved Temporarily
Content-Length: 0
Location: https://mail.sfu.ca/owa/
Server: Microsoft-IIS/10.0
X-FEServer: ITS-EXCHANGE06
X-RequestId: 66b65c01-b923-40f8-8337-ef6018996422
```

The `Server` lines are extracted with `grep` and added to the results file for that school.

5.2 Weighting Classes

Table 1: Weighting Class Adjustment Factors

Country	Sample Size	Responses	Sum of Sample Weights	Sum of Respondent Weights	$\hat{\phi}_c$	Weight factor
Afghanistan	4	3	36.33	27.25	0.75	1.33
Argentina	15	14	136.26	127.17	0.93	1.07
Belarus	5	4	45.42	36.33	0.80	1.25
Belgium	6	5	54.50	45.42	0.83	1.20
Brazil	22	20	199.84	181.67	0.91	1.10
Canada	23	22	208.93	199.84	0.96	1.05
China	44	39	399.68	354.27	0.89	1.13
Colombia	13	8	118.09	72.67	0.62	1.62
Ecuador	3	2	27.25	18.17	0.67	1.50
Ethiopia	4	3	36.33	27.25	0.75	1.33
France	30	25	272.51	227.09	0.83	1.20
India	44	35	399.68	317.93	0.80	1.26
Indonesia	16	12	145.34	109.00	0.75	1.33
Iran	21	17	190.76	154.42	0.81	1.24
Iraq	7	6	63.59	54.50	0.86	1.17
Italy	10	9	90.84	81.75	0.90	1.11
Japan	63	56	572.28	508.69	0.89	1.12
Korea	29	26	263.43	236.18	0.90	1.12
Latvia	3	2	27.25	18.17	0.67	1.50
Lebanon	3	2	27.25	18.17	0.67	1.50
Malaysia	18	16	163.51	145.34	0.89	1.12
Mongolia	3	2	27.25	18.17	0.67	1.50
Nigeria	10	8	90.84	72.67	0.80	1.25
Panama	4	2	36.33	18.17	0.50	2.00
Philippines	13	10	118.09	90.84	0.77	1.30
Poland	11	10	99.92	90.84	0.91	1.10
Puerto Rico	3	2	27.25	18.17	0.67	1.50
Russia	26	23	236.18	208.93	0.88	1.13
Rwanda	2	1	18.17	9.08	0.50	2.00
Saudi Arabia	8	7	72.67	63.59	0.88	1.14
Spain	8	7	72.67	63.59	0.88	1.14
Sudan	8	4	72.67	36.33	0.50	2.00
Syria	4	3	36.33	27.25	0.75	1.33
Taiwan	10	7	90.84	63.59	0.70	1.43
Tunisia	2	1	18.17	9.08	0.50	2.00
Turkey	27	25	245.26	227.09	0.93	1.08
Uganda	4	3	36.33	27.25	0.75	1.33
Ukraine	9	8	81.75	72.67	0.89	1.12
United Kingdom	18	16	163.51	145.34	0.89	1.12
United States	244	222	2216.43	2016.59	0.91	1.10
Uzbekistan	4	3	36.33	27.25	0.75	1.33
Venezuela	4	3	36.33	27.25	0.75	1.33

5.3 Nonresponses per Country

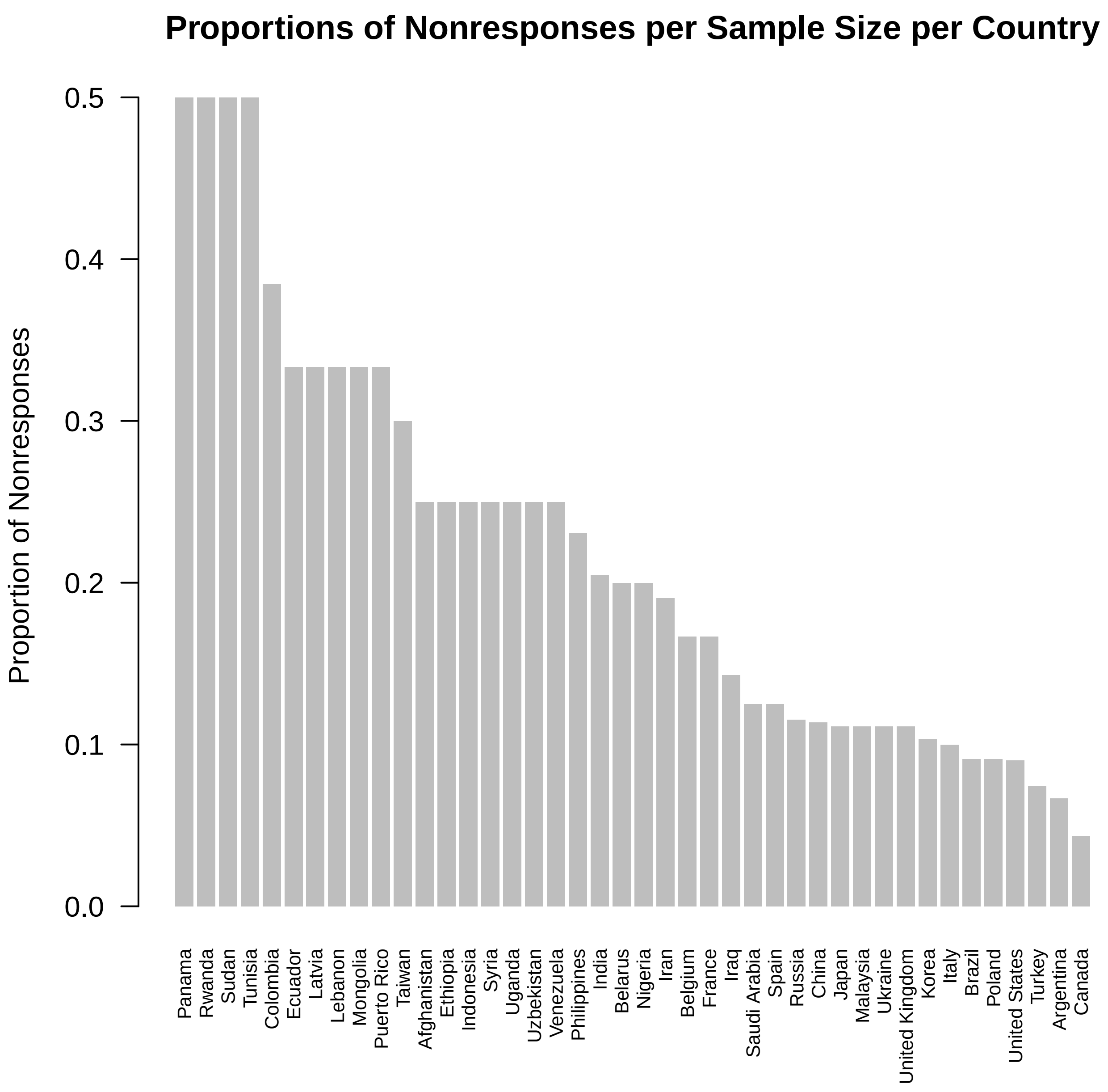

Figure 1: Proportion of sample that is missed from the results as nonresponse

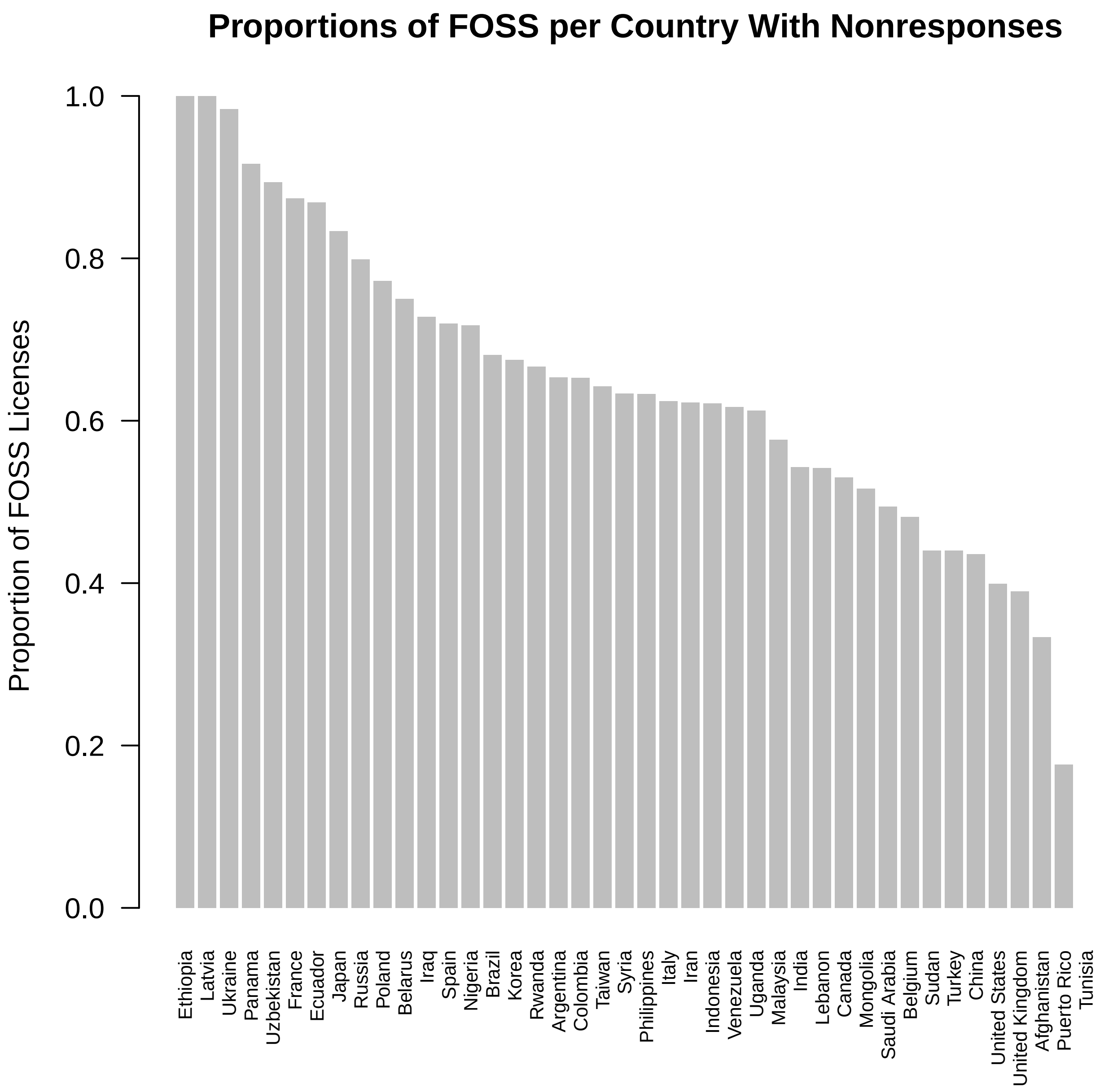

Figure 2: Of countries with nonresponses, the proportion of responses that are FOSS