# Acceptance sampling in Quality Control: from theory to the Web

## In PHP, Python, and gnuplot

Miguel CASQUILHO[1]
Department of Chemical Engineering, IST, *Universidade de Lisboa* (University of Lisbon), and CERENA, ***C**entro de **Re**cursos **Na**turais e Ambiente* ("Centre for Natural Resources and the Environment")
Ave. Rovisco Pais, IST, South Tower (Bldg. 20); 1049-001 Lisboa (Lisbon), Portugal
*mcasquilho@tecnico.ulisboa.pt*

Elisabete A. CAROLINO
H&TRC- Health & Technology Research Center, *ESTeSL- Escola Superior de Tecnologia da Saúde, Instituto Politécnico de Lisboa* (Polytechnic Institute of Lisbon)
Ave. Dom João II, lote 4.69.01, Parque das Nações; 1990-096 Lisboa (Lisbon), Portugal
*etcarolino@estesl.ipl.pt*

André ESPÍRITO SANTO
Dept. of Computer Science and Eng.[ing], IST, *Universidade de Lisboa*
Ave. Rovisco Pais, IST, Bldg. 09 (Computer Science); 1049-001 Lisboa (Lisbon), Portugal
*andre.espirito.santo@tecnico.ulisboa.pt*

Sandro CARVALHO
Polytechnic Institute of Cávado and Ave (IPCA), Applied Artificial Intelligence Laboratory (2AI) and Laboratório de Inteligência Artificial e Ciência de Computadores (LIACC)
4750-810 Barcelos, Portugal
*scarvalho@ipca.pt*

João Luís de MIRANDA
Polytechnic Institute of Portalegre, and CERENA
Campus Politécnico 10; 7300-555 Portalegre
*jlmiranda@ipportalegre.pt*

João M. BORDADO
Dept. of Chemical Eng.[ing], IST, *Universidade de Lisboa*, and CERENA
Ave. Rovisco Pais, IST, South Tower (Bldg. 20); 1049-001 Lisboa (Lisbon), Portugal
*jcbordado@tecnico.ulisboa.pt*

*Abstract* — **Quality is nowadays a ubiquitous component in manufacture and many other activities. We present a computation related to Acceptance Sampling, which, together with Statistical Process Control, makes the fundamentals of statistical Quality Control. We provide a webpage where a user's typical sample from a process can be inserted, leading to "accept" or "reject" vs. given specifications. We have several goals: to offer, on a webpage, the statistical procedure computation, which is otherwise complex; to combine, on a Linux platform, programming languages, PHP and Python, and a graphical utility, gnuplot; and to stress the suggestion of the Web as a computing medium. The webpage is freely accessible to a user just by means of a browser, *i.e.*, installing no software, thus needing no special power or matching operating system, this being an example adaptable to many other problems. The study also draws attention both to the use of the Web for scientific computing, and to the convenience of this use in scientific publications. Web-based computing in general is advocated, this route using the same executable programs as classical computing, which is the core technical difficulty. In our technological era, this still insufficiently explored approach is here made accessible, inviting the sharing between academia and industry.**

***Keywords*** **– Web-based computing; Acceptance Sampling; Statistical Quality Control; scientific computing; Engineering education.**

[1] Corresponding author.

*2023 18th Iberian Conference on Information Systems and Technologies (CISTI)*
*20 – 23 June 2023, Aveiro, Portugal*
ISBN: 978-989-33-4792-8

## I. INTRODUCTION[2]

We understand web-based computing as resources that are made accessible through a webpage[3], available just using a browser, to solve a problem of a given type, the data being supplied by the user to get a result. In this text, we describe a web-based solver to perform the procedure of "Acceptance Sampling". Acceptance Sampling (AS) permits to inspect the (external) input and output of a process, whereas Statistical Process Control (SPC) monitors the (internal) performance of the process, as suggested in Figure 1.

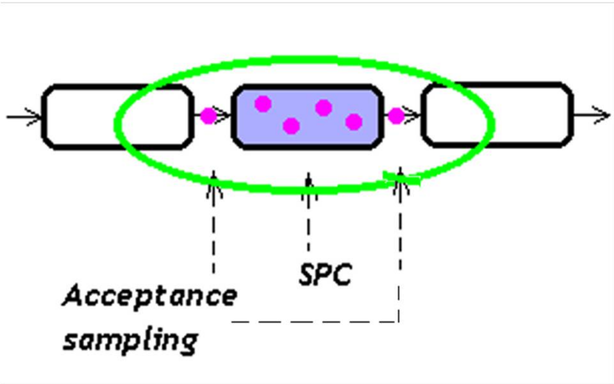


Figure 1 "Quality Control: acceptance sampling (AS) and statistical process control (SPC).

As SPC has (rightly) been generally adopted in industry — namely, through the common "control charts"—, AS has (wrongly) been somewhat forgotten ([1]). This decline is also a consequence of the complexity of the procedure, ruled by the Standard Mil-Std-414 ([2]) (or ISO 3951-1:2022, internationally [3]), a standard which needs the consultation of several numerical tables. This procedure (of course commercially available) becomes straightforward in the free webpage we supply.

The subject in this text belongs to 'Computer Science Engineering' or 'Informatics'[4] (as we'll use subsequently). We feel that the "classical", non-informatic engineers benefit from a cooperation with their informatic colleagues, who would also take advantage from this forgotten "market" for their professional activities.

A web-based solver, as we conceive it, requires no software installation on the user's computer, which is a significant difference from the common applications. These latter (even if free) demand their installation (uninstallation if no longer needed), operating system compatibility, adequate power. The presence on the Internet dispenses with these requirements, and even invites collaborations, as we have since long insisted in (e.g., [5], [6], [7], [8], [9], [10], [11], [12], along two decades). The fact that many references are to our own publications indicates that web-computing is far from being common. The web environment for computing is one of the essential points in this document.

The problem addressed is succinctly explained in the next Section. As to the informatic problem, it is the operation of a web-based program, for which we found no resolution on the Web. Remark that we are not saying that there is no software to solve the problem of AS: we are pointing to a (free) webpage for the purpose. The program will be explained regarding its architecture and use.

In a previous article in this Conference [10], we mentioned "simple science" (perhaps, in the present case, an inadequate term) meaning the easy aptitude of the Web for computing. In our technological era, many people, wrongly, say "technological" instead of "informatic" (or computer science). Innumerable scientific articles, however, while showing their numeric results, do not give the reader any confirmation thereof, some publications offering their programs, yet possibly not matching our operating system or known languages (if free). In the 'FAIR' initiative, Wilkinson et al. ([13]) write: "There is an urgent need to improve the infrastructure supporting the reuse of scholarly data." Regarding programs, however, not data, the issue is the same or more stringent in the scientific literature. The editor of a statistics journal ([14]) wrote to "strongly encourage data and code sharing associated with newly submitted papers". This is a narrow advice, because of the programming language difficulties we mentioned.

In the next Sections: "The problem", the statistical problem is briefly described; "Web applications" are mentioned, in view of their scarcity; and "Results and discussion" show the architecture of our application, with its results. Some "Conclusions" are finally presented.

## II. THE PROBLEM

The problem of Acceptance Sampling is, schematically, as follows.

A buyer (such as a factory or company) wishes to purchase, say, $N = 120$ bags of a substance (bags of 25 kg), with specifications of (between): lower specification, $L = 24.9$ kg with AQL 0.65 %, and upper specification $U = 25.2$ kg with AQL 1 %. AQL, "acceptable quality limit", indicates the admissible fraction of failure to meet the respective specification, for the purpose of sampling inspection. (In this example, the buyer and the seller agree to establish a stricter failure for the lower specification, 0.65 %, than for the upper specification, 1 %.) From a random drawn sample of size $n = 10$ (why 10 ?), is the lot acceptable ?

The procedure is to take a random sample (from an assumed Gaussian population) of size $n = 10$ (a sample size from the cited Standard), and compute its mean and standard deviation. From these sole values, a conclusion to "accept" or "reject" will be obtained. Two "quality indexes", lower and upper (defined symmetrically), are computed, from $(\bar{x}, s)$, the sample mean and standard deviation:

[2] This work was sponsored by CERENA, funded by the Portuguese FCT (see Acknowledgements).
[3] We are using "webpage", instead of "web page" (Web page, etc.), for lack of a reliable standard.

[4] We think it is time to use the terms "Informatics" and "informatic engineering" — from Russian "информатика" (informátika) ([4]).

*2023 18th Iberian Conference on Information Systems and Technologies (CISTI)*
*20 – 23 June 2023, Aveiro, Portugal*
ISBN: 978-989-33-4792-8

$$Q_L = \frac{\bar{x} - L}{s} \qquad Q_U = \frac{U - \bar{x}}{s} \qquad (1)$$

The quality indexes (the higher, the better) are converted to "estimated fraction defectives", $p_L$ and $p_U$ (the lower, the better). These are compared with their "tabulated" maximum allowable values, $M_L$ and $M_U$, under three conditions:

$$\begin{aligned} &p_L \le M_L \ ? \\ &p_U \le M_U \ ? \\ &p_L + p_U \le \max(M_L, M_U) \ ? \end{aligned} \qquad (2)$$

If all the above questions have affirmative answers, the sample leads to "Accept" the lot it came from; otherwise, it leads to "Reject" it.

Strangely, the Standard does not reveal how to compute the conversion of a $p$ into an $M$, the conversion resulting from an extensive (a dozen pages !) table. Even the most prestigious authors in Acceptance Sampling, Schilling and Neubauer ([15]), only supply the table (their Table T12.7, in pages 735–748). An additional advantage of our webpage is that it incorporates this conversion of $p$ into $M$.

In the Section "Results and discussion", we detail the computation, mainly in its programming on the Internet The procedure is readily accessible and able to be computed at the respective webpage ([16]), as explained further, many other webpages of ours being available for other problems.

## III. WEB APPLICATIONS

The present study shows a web-based application that solves a statistical problem,

Ref. [16]

This constitutes an inherent suggestion for the use by students in "STEM", Science, Technology, Engineering, Mathematics.

As an example of the Web as a real computing medium, we use to mention Ponce ([17]), in Hydraulics, though several websites, of course, permit computations by the user. The site provides a "virtual laboratory", Vlab ([18]), since 2004. His concept is similar to ours, started in 1998. For an arbitrary example of his problems, see Figure 2, "Calculation of the Froude number in open-channel flow", Problem No. 143 A ([19]). Vlab's user inserts the input data and gets the results. We mention this problem because it is perhaps the easiest among some 300 problems in the site: indeed, with flow velocity $u = 1.4$ m/s and depth $y = 5$ m, it is $F = 0.2$ (by simple arithmetic, $g = 9.8$ m/s$^2$, $F = 1.4/\sqrt{(5\times9.8)}$). In our webpages, we always give default data (absent there).

Our webpages (and Ponce's) do not discourage studying. From our more recent experience, Engineering students have quit programming. While the professors (ourselves) do not use real computing, the students fall to Excel, acceptable if well used With webpages to solve the problems, students should do their own calculations, comparing their results with those from a webpage, such as we are presenting ([16] again). Many other cases of ours appeared in previous editions of this Conference, as already mentioned, and in a literature article of ours in Statistics ([20]).


Figure 2 Ponce's Problem No. 143 A (in Hydraulics): Calculation of the Froude number in open-channel flow.

If the students avoid the "new technologies", they are not alone. In the scientific literature, we rarely find a scientific article on numerical topics giving a webpage with their computation where we might use it.

The existence of Computer Science (Informatics) standards facilitates the combination of various languages, with their own characteristics. In this study, we combine Python with gnuplot, under PHP, all of them free. As the present work does not lead to large-scale computing, Python appears to be a reasonable choice. There is also the intention of showing this example as a template, because, accordingly, we always advocate the Web for computing.

For Python, we refer to Python 3, version 3.9.2. For gnuplot, we use version 5.4. These versions (not our choice) are those installed by the administration in the public Linux system in our University.

We recall that, when convenient (not used here), Python is easily connected, namely, to C and to Fortran, by converting the segments of those languages into Python modules, with Python "as glue" ([21], [22]). These modules (in the case of Linux) are dynamically linked files, *i.e.*, shared objects ("so"). For example, while 'f2py' takes a Fortran file named 'Name.f90' to produce 'Name.so', the corresponding 'f2py3' produces (in our Linux installation) a 'Name.cpython-…-x86_64-linux-gnu.so'. This module would be used in the Python script simply as 'import Name'.

A further note about using gnuplot from Python is in order. Python is one of a few languages that accept 'heredoc' literals (e.g., [24], [25]), inside {} in the case of Python. This feature can, of course, be emulated if a language does not contain it, but is quite useful. We used this feature on calling gnuplot, thus, namely, easily permitting to include the sample size ($n$) in a plot title ("Sample of 10 elements"), from the following gnuplot expression, where the Python variable 'nn' is 10:

```
set title 'Sample of {nn} elements' …
```

## IV. RESULTS AND DISCUSSION

As this work is for use on the Web, the computing itself is not visible to the user, and is executable at the mentioned [16]. There, the user has available a set of default data, and just clicks the "Execute" button, or inserts the problem data (input) and gets

*2023 18th Iberian Conference on Information Systems and Technologies (CISTI)*
*20 – 23 June 2023, Aveiro, Portugal*
ISBN: 978-989-33-4792-8

the corresponding results (output). The results are built dynamically in HTML through the scripting language PHP, native for the Web ([26]). The architecture has two components: a web-based front-end, to show the user interface, and a computing back-end to execute the tasks.

The front-end accepts the task's arguments (data), processes them, schedules the task, and sends these data to the back-end computing system. The back-end finishes the task, and replies to the front-end with the output in a dynamic new page. (The results might appear in the same page, through PHP empty 'action=', but we found it to be less clear.) In the present case and in this type of problems, the output is simply alphanumeric with graphics, as frequently in Engineering and Statistics. This structure is common to our problems, because our intention is to solve them, and make them available.

The numerical segments of our problem are solved in Python, which has its own graphical capabilities. This might lead to use Python for this purpose, but we called gnuplot ([27]), a free graphical package, to which we are more accustomed than the Python plotting libraries, such as 'matplotlib.pyplot'. The use of gnuplot is also preferred for generality, as the call to gnuplot can be done from other languages (such as C, Fortran). For the Web, the results of the problem are formatted with HTML (essentially with the tag 'pre', i.e., pre-formatted text), and shown to the user.

The example is run on the public system of CIIST, the university's Computing Centre ([28]), with a Linux operating system. As of the writing date, it reports 16 GB memory; 'uname' gives amd64, Debian 5.10.140 (2022), x86_64 GNU/Linux; and 'lscpu', 8 CPU's, from Xeon, 2 GHz.

The architecture follows these steps, where the names (bold, italics) follow the particular case described:

***a)*** The user addresses the problem's PHP web page, ***P-acceptSampling.php***, and inserts the data (input) in the fields of an HTML 'form' in the page (data sent to the next file through PHP POST variables) ("P-" recalling a Python script as computing engine).

***b)*** Upon clicking the 'Execute' button, the data are sent to an intermediate PHP file, from the 'form' statement 'action=***AcceptanceSampling.php***'

***c)*** This PHP file prepares the input received and sends it to a previously prepared Python script (next) as command line arguments for execution. The PHP file takes the results from the execution, and creates a (temporary, dynamic) PHP file (web page) with the results.

***d)*** The standalone Python script, ***acceptanceSampling.py***, is then run through a PHP 'shell_exec' statement. This Python script runs the calculations.

***e)*** The Python output (to *stdout*) is sent to a temporary, dynamic page, the Results web page. This page contains an HTML 'pre' tag with the text and one or more graphics ('png'), which were just generated through a call to the (Python) 'base64' module. This way, no files are left on the server, needing no further deleting. This also avoids clashes between several users addressing the web page, a fact to be considered in public access.

***f)*** Auxiliary files — PHP environment files through PHP 'include', i.e., cascading style sheet, and the constant images that characterize the website.

The files necessary to implement the above actions are also mentioned for conciseness in the webpage.

In the above, the emphasized file names obviously vary depending on the particular problem, the other files being the same for other problems, for simplicity and consistency. Schematically, Figure 3 shows the structure of the communication among the various components.

If user-supplied Excel files are necessary, they can easily be read through the Python 'pandas' module. In this problem, however, the most extensive data are just a simple row with the sample data that also can be copied and pasted in the 'textarea'.

The fact that the problem runs "over the Web" adds no difficulty versus the ordinary resolution of a scientific problem, in which, as always, the difficulty is the inherent scientific knowledge of the particular subject.

The language used in this application is Python, and gnuplot, for plotting. The absence of licensing for these choices is also important, considering the presence on the Internet.

Our example to sentence[5] a lot based on its sample is now briefly described. The data for the example can be: (*i*) uploaded as an Excel 'xlsx' or 'xls' file, with a single sheet (then the following data present in the textarea are ignored); or (*ii*) just copied and pasted into the 'textarea' in the web page, in American (decimal mark: point) or European (comma) style. (Note that 'xlsx' files have, of course, only one internal format, irrespective of a user's national version.)

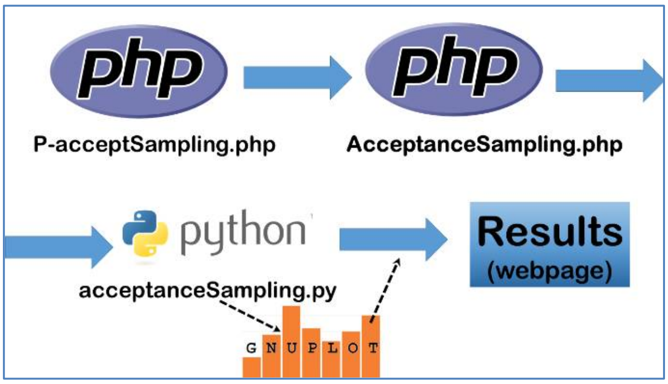


Figure 3 Communication among the various webpage components.

A few other data (specs and AQL's) have to be supplied that do not justify massive insertion:

- Lower, *L*, and upper, *U*, specifications (specs);
- AQL's for the two specs;
- Indication of the active specs: lower only; upper only; or (most usual) both.
- Boolean showing plot data

[5] This word is typical in the literature on AS.


ISBN: 978-989-33-4792-8

It should be reminded that, in the context of Quality, a product is "good" or "bad" depending on what is expected from it. So, namely, the sample that is proposed as default will lead to a result of Accept or (really) Reject for a favorable or unfavorable set of specifications and AQL's. For example, simply relaxing the AQL for *L* from 0.65 % to 2.5 % inverts the conclusion from Reject to Accept.

As default data for the problem are given to the user in the webpage, results appear with a simple click on the 'Execute' button. An alternative to the default data, besides the user's own data, is to use a self-instructive Excel file also available for simple download and subsequent upload. An image of the input webpage is in Figure 4.

If the Excel file 'sample15.xlsx' ($n = 15$, from another $N$, e.g., $N = 200$, same AQL's) proposed in the webpage is used, the result is an "Accept", the data being shown in in Figure 6.

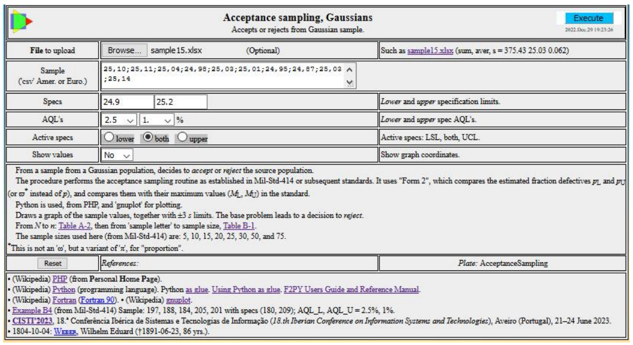


Figure 4 Web page (input) for the Acceptance Sampling procedure. (The default option "both" specifications is selected.)

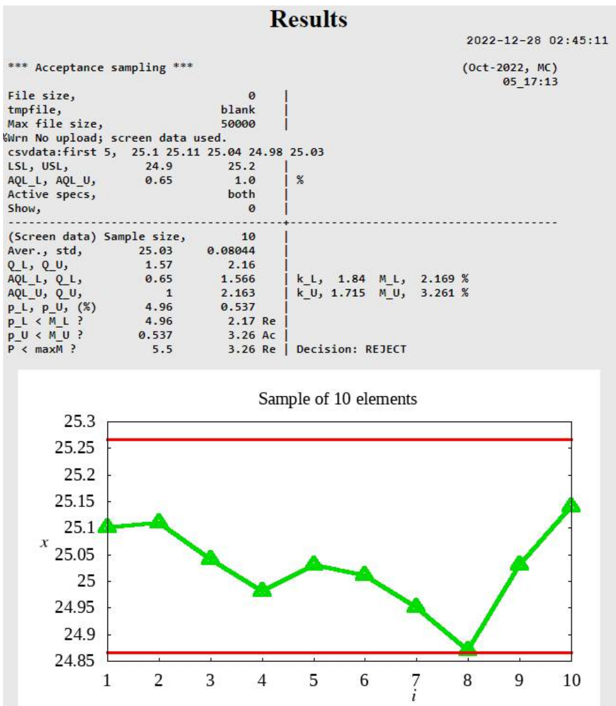


Figure 5 Results from the default data ($n = 10$).

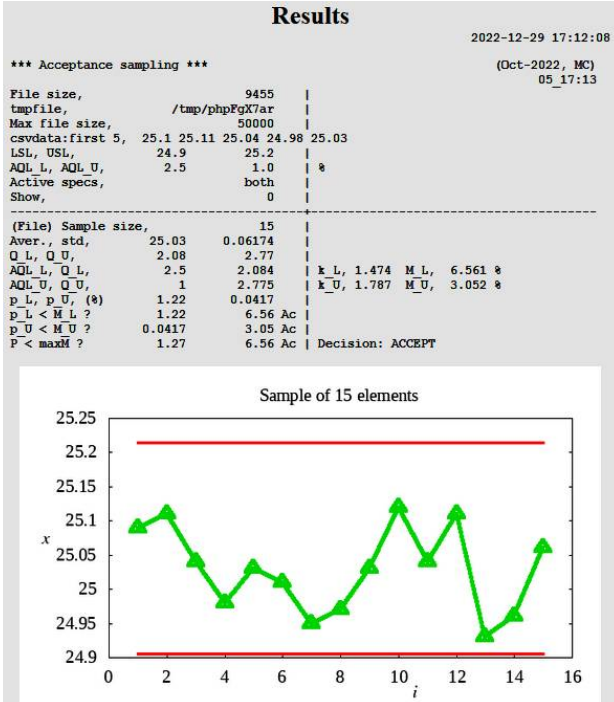


Figure 6 Results from the proposed alternative data ($n = 15$).

## V. CONCLUSIONS

Regarding scientific computing, the Internet medium has been, in our opinion, a little explored field. We mentioned one noteworthy comparable example of such computing by another author, while countless numerical publications offer no web-based verification. The present application just needs a browser, requiring no software installation by the user, who easily supplies the problem data. The program webpage solves a problem in "acceptance sampling", a major component of Statistical Quality Control. It is programmed in PHP and Python, with gnuplot for graphing, and is available for execution on the first author's website. Our objectives were: to propose the use of this type of computing based on the client-server model, which permits access without any need from the user's terminal; to draw attention to the worrying issue of students quitting computing; and to show the usefulness of combining several languages and utilities, with Python as a suitable manager.

The architecture proposed serves an indefinite number of problems, for which the only difficulty is their intrinsic mathematical nature, and the language used can be any other, provided that it is compatible through due standards, and that speed, security and licensing are adequate.

The programs and scripts written are essentially the same as would be out of the Web. Along the years in our academic and industrial practice, we have convinced ourselves that Web computing is recommendable for both purposes, promoting the much-praised connection between academia and industry.



ISBN: 978-989-33-4792-8

## Acknowledgements

The author MC does research at the Department of Chemical Engineering, IST, University of Lisbon, Lisbon, Portugal, and CERENA, *"Centro de Recursos Naturais e Ambiente"* (Centre for Natural Resources and the Environment), under Project UID/04028/2020, funded by FCT, *"Fundação para a Ciência e a Tecnologia"* (Portuguese 'National Science Foundation'); EAC teaches and does research at H&TRC-Health & Technology Research Center, ESTeSL (*Escola Superior de Tecnologia da Saúde*), *Instituto Politécnico de Lisboa* (Polytechnic Institute of Lisbon); AES is MSc student at the Dept. of Computer Science and Engineering, IST; JLM teaches and does research at the Dept. of Technologies, ESTG, Polytechnic Institute of Portalegre, Portalegre, Portugal, and CERENA; and JB does research at the Dept. of Chemical Engineering and CERENA. All thank Project BetterPlastics. CIIST, *"Centro de Informática do IST"* (Informatics Centre of IST) supplies the computing system.


## References


[1] E. Carolino, M. Casquilho, M. Rosário Ramos, I. Barão (2016), "Applied scientific computing over the Web: robust methods in Acceptance Sampling for Weibull variables", ISBIS 2016 Meeting on Statistics in Business and Industry, 08–10 Jun., Barcelona (Spain).

[2] ASQ, "Variables sampling and Mil-Std-414", https://asq.org/quality-progress/articles/variables-sampling-and-mil-std-414?id=a45a676baf6141c49cd18d68b2be78ca, accessed 30-Dec-2022.

[3] ISO, ISO 3951-1:2022, "Sampling procedures for inspection by variables — Part 2: General specification for single sampling plans indexed by acceptance quality limit (AQL) for lot-by-lot inspection of independent quality characteristics", https://www.iso.org/standard/74706.html , accessed 30-Dec-2022.

[4] Informatics, Online Etymological Dictionary, https://www.etymonline.com/search?q=informatics , accessed 30-Dec-2022.

[5] B. Franco, and M. Casquilho (2011), "A Web application for scientific computing: combining several tools and languages to solve a statistical problem", CISTI'2011, *6.ª Conf. Ibérica de Sistemas e Tecnologias de Informação* (6.th Iberian Conference on Information Systems and Technologies), Chaves (Vila Real, Portugal.

[6] M. Casquilho, F. C. Rosa (2012), "Scientific computing on the Internet: direct resolution of several engineering problems", CISTI'2012, 7.th Iberian Conf. on Information Systems and Technologies, Madrid (Spain).

[7] M. Casquilho (2013), *"Computação científica, Internet, Indústria"* (Scientific computing, Internet, Industry), 1.st Portuguese Meeting on Mathematics for Industry, FCUP, University of Porto, Porto (Portugal).

[8] M. Barros, and M. Casquilho (2019). Linear Programming with CPLEX: an illustrative application over the Internet", CISTI'2019, Coimbra (Portugal).

[9] M. Casquilho, A. Paredes, F. C. Rosa, J. L. Miranda (2021), "A web-based cooling tower application", CISTI'2021, Chaves (Vila Real), online, Portugal.

[10] M. Casquilho, A. Pires, Fátima C. Rosa, J. L. Miranda, J. Bordado (2022), "A web-based superabsorbent polymer solver (in simple science)", CISTI'2022, Madrid (Spain).

[11] M. Casquilho, J. Buescu (2022), "From unequal size sample sums to estimating the standard deviation", ICMASC 2022, International Conf. on Mathematical Analysis and Applications in Science and Engineering, Porto (Portugal).

[12] M. Casquilho, J. Buescu (2022), "Mean of Exponential distribution: estimation from sums of unequal size samples", COMPSTAT 2022, 24.th Intern'l Conf. on Computational Statistics, Bologna (Italy), 23–26 Aug..

[13] M. D. Wilkinson (2016), "Comment: The FAIR Guiding Principles for scientific data management and stewardship", *Scientific Data,* 3:160018, 9 pp.

[14] B. M. Colosimo, ed. (2019), Message, *J. of Quality Technology*, 51 (1), 1–2 or https://www.tandfonline.com/doi/full/10.1080/00224065.2019.1569896., accessed 30-Dec-2022.

[15] E. Schilling, D. Neubauer (2017), "Acceptance Sampling in Quality Control", 3.rd ed., CRC Press, Taylor & Francis Group, Chapman & Hall, 841 pp.

[16] M. Casquilho (2022), "Acceptance sampling by variables (Gaussian)", http://web.tecnico.ulisboa.pt/~mcasquilho/compute/CISTI_2023/Acceptance/P-acceptSampling.php

[17] V. M. Ponce, San Diego State University, http://ponce.sdsu.edu/ , accessed 30-Dec-2022.

[18] V. M. Ponce (2017), Vlab, http://ponce.sdsu.edu/onlinefallvelocity.php , accessed 30-Dec-2022.

[19] V. M. Ponce, "Calculation of the Froude number in open-channel flow", http://ponce.sdsu.edu/onlinefroude.php , accessed 30-Dec-2022.

[20] M. Casquilho, J. Buescu (2022), "Standard deviation estimation from sums of unequal size samples", *Monte Carlo Methods and Applications* (doi: 10.1515/mcma-2022-2118).

[21] The SciPy community, "Using Python as glue", https://numpy.org/doc/stable/user/c-info.python-as-glue.html , accessed 30-Dec-2022.

[22] T. E. Oliphant (2007). Python for scientific computing. *Computing in Science & Engineering*, *9*(3), 10–20, in Kermode (2020) [23].

[23] J. R. Kermode (2020), "f90wrap: an automated tool for constructing deep Python interfaces to modern Fortran code". *Journal of Physics: Condensed Matter*, 32, 30, 305901 (open access), https://iopscience.iop.org/article/10.1088/1361-648X/ab82d2 , accessed 30-Dec-2022.

[24] Wikipedia, "Here document", https://en.wikipedia.org/wiki/Here_document , accessed 30-Dec-2022.

[25] Bash heredoc example, Indiana State University, http://mama.indstate.edu/users/ice/bash/heredoc

[26] PHP, The PHP Group, https://www.php.net/ , accessed 30-Dec-2022.

[27] Gnuplot, http://www.gnuplot.info/ , accessed 30-Dec-2022.

[28] CIIST, "Centro de Informática do IST" (Informatics Centre of Instituto Superior Técnico), https://si.tecnico.ulisboa.pt/ , accessed 30-Dec-2022.


CERENA
Centro de Recursos Naturais e Ambiente

FCT
Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR



ISBN: 978-989-33-4792-8