DEPARTMENT OF COMPUTER SCIENCE

EDUARDO RAMOS ALEMÃO ATANÁSIO DOS REIS

Bachelor in Computer Science

SURVEYING COMMUNITIES OF USERS OF MATLAB AND CLONE LANGUAGES

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
September, 2021

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

SURVEYING COMMUNITIES OF USERS OF MATLAB AND CLONE LANGUAGES

EDUARDO RAMOS ALEMÃO ATANÁSIO DOS REIS

Bachelor in Computer Science

Adviser: Miguel Jorge Tavares Pessoa Monteiro
Assistant Professor, NOVA University Lisbon

Co-adviser: Ana Catarina Gralha de Almeida
Researcher, NOVA University Lisbon

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
September, 2021

Surveying communities of users of MATLAB and clone languages

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.7.0) [49].

Abstract

MATLAB is a computing environment and programming language known for allowing intricate matrix manipulations and plotting of functions and data. However, *MATLAB* and its clone languages such as *GNU Octave* had, until recently, limitations as regards support to modularity. Only in 2008 did *MATLAB*'s support to object-orientation seem to stabilise. Due to these being relatively recent improvements, the community's adaptation to them has not yet been documented by way of a thorough and conclusive statistical analysis. It was, therefore, unknown to which extent these new capabilities have integrated the community, which is precisely what we proposed to bring to light.

To our knowledge, no previous work distinguished and categorised the different subcommunities that, together, form the community of users of *MATLAB* or any of its clones. These should be distinguishable according to the purpose for which they program, their field of work and their levels of expertise with each feature of these languages, among many other factors.

This thesis contributes with a structured survey analysis resulting in a stratification of the community of users of *MATLAB* and its clone languages. Through an empirical study in the form of an online questionnaire, which received 212 valid responses, the study enables a better grasp on how the community uses these languages and for what purposes. Additionally, it provides an understanding of some of the users' practices and typical behaviours with programming, and more specifically with the languages in question.

During the planning stages of the survey instrument, its validity was thoughtfully considered. Later, after the instrument was administered, the internal consistency was measured. Combined with an adequate sample size and diversity in the participants, this ensures that the study presents statistically significant results and implications.

Keywords: *MATLAB*; *MATLAB* clones; surveys; questionnaires; modularity; object-oriented features; community.

Resumo

O *MATLAB* é um ambiente de computação e uma linguagem de programação que é conhecida por permitir cálculos complexos de matrizes e construção de gráficos. Contudo, o *MATLAB* e os seus clones como o *GNU Octave* tinham, até recentemente, algumas limitações quanto ao seu suporte a modularidade. Apenas em 2008 é que o *MATLAB* conseguiu estabilizar o seu suporte a orientação por objetos. Devido a estas melhorias serem relativamente recentes, a adaptação da comunidade a estes recursos não foi ainda documentada através de uma análise estatística minuciosa e conclusiva. É, desta forma, desconhecido o ponto até que estas novas capacidades foram absorvidas pela comunidade, o que é precisamente o que nós nos propusemos a evidenciar.

Não existem trabalhos realizados anteriormente que, do nosso conhecimento, tenham distinguido e categorizado as diferentes sub-comunidades que, juntas, formam a comunidade de utilizadores de *MATLAB* ou qualquer um dos seus clones. Estas devem ser possíveis de distinguir pela razão por que programam, pelos seus ramos de trabalho e pelos seus níveis de aptidão com cada um destes recursos, entre muitos outros fatores.

Esta tese contribui com uma pesquisa estatística estruturada, da qual resulta uma estratificação da comunidade de utilizadores de *MATLAB* e dos seus clones. Através de um estudo empírico na forma de um questionário online, em que foram obtidas 212 respostas válidas, o estudo proporciona uma melhor compreensão da forma como a comunidade utiliza estas linguagens e com que objetivos. Além disso, proporciona um melhor entendimento dos hábitos e dos comportamentos típicos dos utilizadores, não só com as linguagens em questão, mas com programação de modo geral.

Durante a fase de planeamento do instrumento de pesquisa, a validade do mesmo foi ponderada. Mais tarde, após o instrumento ter sido posto em prática, a consistência interna foi calculada. Em combinação com um número adequado de respostas e com a diversidade dos participantes, isto faz com que o estudo apresente resultados e implicações estatisticamente significativos.

Palavras-chave: *MATLAB*; clones de *MATLAB*; inquéritos; questionários; modularidade; orientação a objetos; comunidade.

Contents

List of Figures

List of Tables

List of Listings

Acronyms

CA Cronbach's Alpha x, 57, 58, 59, 60

FG Focus groups 11, 13, 28, 47

KMO Kaiser-Meyer-Olkin ix, 53, 54, 56

OOP Object-Oriented Programming 3, 19, 22, 25, 65, 66, 77, 78, 79, 81, 83
OSPT One-Sample Proportion Test 67, 68, 69, 70

PCA Principal Component Analysis x, 52, 53, 54, 55, 56, 57, 60
PS Pilot studies 11, 13, 27
PSF Python Software Foundation 30

SC Spearman's Correlation 67, 70, 71
SR Survey research 1, 2, 4, 6, 18

1

Introduction

In this Chapter we present an introduction to this dissertation. We start with an overview on the topic of *MATLAB*, its clone languages and its users (Section 1.1), as well as a summary of the motivation and goals behind this Survey research (SR) on the communities of users of *MATLAB* and its clones (Section 1.2). Afterwards, we discuss the administration of a questionnaire as the approach used to reach the goals of this SR, based on a series of proposed research questions and sub-questions (Section 1.3). Then, we summarise the main contributions brought by this research (Section 1.4). Finally, we conclude the Chapter with an overview of the structure of the remaining part of the document (Section 1.5).

1.1 Background

MATLAB [57] is a computing environment used by millions of engineers, scientists and researchers as well as hundreds of thousands of organisations worldwide [52]. It impacts a vast array of fields: from computational biology, to deep learning, machine learning, internet of things, robotics, and many more.

Throughout the years, several programming languages have emerged and been classified as *MATLAB*'s *clones*. *GNU Octave* [22], *Scilab* [80] and *Rlab* [83] are well known examples. These languages have a similar numerical computation power and typically full compatibility with *MATLAB* files. Moreover, they share many traits and capabilities with *MATLAB*, such as the use of matrices as a primary data type, support for complex numbers and plots, extensive function possibilities and the ease of creating user-defined functions. For these reasons, the communities of users of *MATLAB* and of its clones can be compared and analysed in parallel.

1.2 Motivation and Goals

The intent and feeling of the communities of users of *MATLAB* and its clones is not well documented. Because these programming languages are so broad and offer such a wide

range of capabilities, it is unclear which of those capabilities are, in fact, most used and to which extent they are used. For instance, it was not until 2008 that *MATLAB* improved on some of its limitations regarding object-orientation [54]. Therefore, it is valuable to get to know and characterise the communities, as well as to identify which different issues arise across different portions of the communities, the importance that the communities attach to each issue and how they deal with them. By learning from and improving upon previous research, we are able to gather a more statistically significant number of responses. This way, more accurate and authentic conclusions are drawn by distinctly diversifying and categorising the target communities.

From less experienced to more experienced users, or from data analysts to biologists, they all have different thoughts, practices and opinions concerning their use of these languages. Thus, by categorising these target communities, we are able to understand and distinguish how they interact with *MATLAB* and its clones, as well as able to compare each of them to the entire community as a whole. The results from this could, then, be a valuable resource for the future development of these languages and evidently for anyone considering working with these languages.

Therefore, this thesis aims to better document and more clearly distinguish the different portions of the different communities of users of *MATLAB*, as well as their levels of comfort with many of the distinguished features and capabilities that *MATLAB* and its clones offer. In addition, it brings light to the concerns that these communities have, as well as the obstacles and issues which they face with the languages in question.

There has been research, done in 2017, by Katia Duarte, in which the *MATLAB* and *Octave* programming communities were surveyed concerning the limitations of *MATLAB*'s support to modularity [13]. However, the expectations held for the questionnaire in that work, as regards the grasp of some technical aspects, were possibly unrealistic as they tried to incorporate specific and sophisticated topics which narrowed the spectrum of participants that were able to provide a quality response. Ultimately, they were able to obtain a total of 42 responses with a completion rate of 76.19%, meaning that 23.81% were only partial responses. Due to the small sample size obtained, the validity of the study was threatened and thus, the validity of its conclusions could not be guaranteed.

For our SR, Katia's work was invaluable as it helped us further understand what would work best for our specific case. The limitations they faced were taken into consideration during our own work, letting us optimise it and avoid any unnecessary constraints.

1.3 Approach taken and Research Questions

Following the investigation of the state of the art on SR and *MATLAB*, we constructed and administered a questionnaire in order to accomplish the established goals. With the plan of targeting the communities of users of *MATLAB* and its clone languages, the questionnaire was developed using *Google Sheets* and administered online via e-mail and online community forums specifically related to *MATLAB* or any of its clone languages

(see Chapter 5). It received a total of 215 responses, from which 27 were partial responses due to skip logic in the questionnaire's structure (see Subsection 5.5.3).

To ensure we'd meet the goals, and to guide us during the process of construction and administration of the questionnaire, we proposed a series of research questions and sub-questions. These also help us contextualise and further detail the problem at hand.

1. How is the community of users of *MATLAB* and its clones structured and divided, according to their level of expertise, the application domain in which they program, among other factors?
    a) How is the community divided according to level of expertise?
    b) How is the community divided according to their application's domain?
    c) How is the community divided according to the other languages which they use?
2. How proficient are the users of *MATLAB* and its clones?
    a) Are there users that do not use more than the command window?
    b) Do users spend time keeping their code maintainable?
    c) To what extent do the users make use of *MATLAB*'s modularity capabilities? (e.g. functions, classes)
3. What is the level of users' satisfaction with *MATLAB*'s current support for modularity?
    a) Do users understand and appreciate the benefits that modularity can bring to a *MATLAB* program?
    b) Do users think there are other languages that provide better support for Object-Oriented Programming (OOP)?

Additionally, we formulated a series of hypotheses. The testing of these hypotheses helps us more confidently answer the research questions we established. They are set up in such a way that each hypothesis is, in some form, related to one of the research questions. In Table 1.1 we can observe the null hypotheses formulated, H_0, their respective alternative hypotheses, H_1, and the corresponding research questions.

Table 1.1: Association between the hypotheses and the research questions.

H_0	H_1	Research Question
A user's level of expertise is not correlated to the application domain in which the program.	A user's level of expertise is correlated to the application domain in which they program.	1
A user's level of expertise is not correlated to the usual size of their programs.	A user's level of expertise is correlated to the usual size of their programs.	1
The years of experience a user has is not correlated to the importance they give to their programs' maintainability and reusability.	The years of experience a user has is correlated to the importance they give to their programs' maintainability and reusability.	2
A user's effort to keep a program maintainable is not affected by their expectation of being the sole user of that program.	A user's effort to keep a program maintainable is affected by their expectation of being the sole user of that program.	2
A user's level of expertise does not influence their opinion on *MATLAB*'s support to modularity.	A user's level of expertise directly influences their opinion on *MATLAB*'s support to modularity	3
The importance a user gives to the program's maintainability does not influence their satisfaction with *MATLAB*'s support to modularity.	The importance a user gives to the program's maintainability directly influences their satisfaction with *MATLAB*'s support to modularity.	3

1.4 Main Contributions

Through a survey-based empirical study, including the data analysis and hypothesis testing conducted, we are able to provide the following contributions:

- The stratification and the demographic analysis of the community of users of *MATLAB* and its clone languages;
- An understanding of the types of languages the users typically use, besides *MATLAB* and its clone languages;
- A better understanding of how common a more superficial use of *MATLAB* is;
- An understanding of how much effort the users put into the reusability and maintainability of their code, and in what circumstances;
- An analysis of the different ways users interact with *MATLAB*, such as the size of their programs and for how long they typically maintain and keep their programs operational;
- An understanding of how mindful the users are of *MATLAB*'s modularity, as well as their level of understanding and satisfaction with it.

1.5 Document Structure

Following the current Chapter, the structure of this document is as follows:

- **Chapter** 2 focuses on the state of the art on SR. It presents commonly accepted concepts and methodology for how to construct research questionnaires and how to analyse the resulting data.

- **Chapter 3** presents an overview of the *MATLAB* language, describing its uses and history. Subsequently, it demonstrates the language's syntax and how it functions. Finally, it also analyses *MATLAB*'s most popular clone, GNU Octave, exhibiting how they differ and how it can be used.
- **Chapter 4** includes a rundown on similar work to this study, covering not only surveys done to communities of users of languages similar to MATLAB but also more distant communities. Later in this Chapter is also presented a summary of the different surveys analysed and how they compare to ours on various parameters.
- **Chapter 5** presents an analysis of the target population and the online communities relevant to the study. Then, it introduces the questionnaire specification. Finally, it includes an in-depth analysis of the structure of the questionnaire.
- **Chapter 6** first tackles the process of administering the questionnaire. Then, it includes an analysis of the internal consistency of the questionnaire and the profiling of the participants. Afterwards it tackles the formulation and the testing of hypotheses. Finally, it presents the results and implications of this study.
- **Chapter 7** includes a summary of the conclusions of this study. Additionally, it tackles the future work that could be done in this line of study.

2

Survey Research

In this Chapter we explore the state of the art on SR. We present studies on how a survey should be constructed (Section 2.1) and how its results should be analysed (Section 2.2). This is done because everything discussed in this Chapter is relevant theory to take into consideration during the conduction of this thesis' survey. The information present here is later observable and demonstrated in the survey conducted and in the thesis itself.

2.1 Constructing surveys

A survey consists of, not only the instrument for gathering the information, but a comprehensive method for collecting information to describe, compare or explain certain knowledge, attitudes or behaviours. Therefore, the purpose of SR is to produce statistics by asking questions whose answers will constitute data prepared for analysis. These questions can be asked in person, on paper, by phone, online, among other means. This means gathering data first-hand from its source, the individuals, which is a form of **primary** research [35, 19].

SR is used to gather the opinions and/or feelings of a selected group of individuals, which in the case of this thesis are the users of *MATLAB* and its clone languages. However, it is important to note that, in general, information can only be collected from a fraction of the population. For example in this thesis, it is not realistically possible for us to obtain a response from the entire network of users of *MATLAB* and its clones. Instead, we can only reach a fraction of this population. However this is not a concern, as long as this fraction is large enough to draw relevant conclusions from [19].

There are several key-points to consider when creating a survey instrument, such as wording, response format and question placement. All of these can heavily influence the answers given by the participating individuals.

There are six important stages in survey-based research [35]. Each of these is covered in more detail later in this Chapter:

1. Setting the survey's objectives;
2. Selecting the most appropriate survey design;

3. Constructing the survey instrument (focusing on self-administered questionnaires);
4. Assessing the reliability and validity of the survey instrument;
5. Administering the instrument;
6. Analysing the collected data.

2.1.1 Setting the objectives

The first step to any survey research is setting the survey's objectives [35]. This means defining the survey's expected outcomes or defining the question that the survey intends to answer. The three most common types of objective are [35]:

- To analyse the frequency of a characteristic that occurs in a population;
- To judge the severity of a certain characteristic or condition that occurs in a population;
- To identify factors that may influence a characteristic or condition.

The first two types of objective are descriptive, meaning they describe a condition found in a population in terms of its frequency or impact, whereas the third type analyses the existing relationship between factors and conditions within a population. In the context of this thesis' survey, the objectives are for the most part descriptive, as the main goals include categorising the target communities and analysing the frequency and severity of their characteristics.

Naturally, it is also important to decide if a survey is really the most appropriate method to address the objectives at hand. It is at this point in the thought process that one should question this by asking themselves questions such as:

- What population can answer the survey questions reliably?
- Is there a way to obtain a representative sample of that population?
- Is it clear what variables need to be measured and how to measure them?

If we cannot answer these positively with absolute confidence they should consider other research approaches, different to a survey, to address the proposed objectives.

2.1.2 Survey Design

There are two major types of survey design, **cross sectional** and **longitudinal** [35]. Most surveys in software engineering are built with a **cross sectional** design in mind, in which the participants are asked questions at a particular fixed point in time. However, in **longitudinal** design participants are asked questions at different time periods, and these

can be the same or different participants [35]. This thesis' survey, specifically, has a cross sectional design, as it involves the collection of data at a defined time.

Another important factor that needs to be decided is the means through which the survey will be administered, as this influences the questions that can be addressed. A common option is self-administered questionnaires through the internet, which is a type of survey designed specifically to be completed by a respondent without the intervention of the researchers or an interviewer. This is how this thesis' survey is administered because this makes it easily accessible by anyone worldwide, as opposed to paper questionnaires which are exclusively accessible by a local population. Additionally, because it does not require printing each copy of the questionnaire, this method allows for an easier and faster editing than a paper questionnaire would, and it is also more environmentally friendly.

Additionally, factors such as question ordering and wording will also differ according to the design method chosen.

2.1.3 Developing a Survey Instrument

Just as with most other research studies, we begin by searching what other studies have been previously done on the topic and which methods they used to collect their data. This is done because we do not want to merely duplicate the work and research of someone else. We want to learn from that previous work and improve upon it.

If the previous studies on the topic have developed strategies or questions that worked well, we may choose to adopt them or they may provide ideas about new variables to tackle, facilitating the planning process. On the other hand, if these previous studies faced any issues or unintended outcomes from their instruments we can think of measures that we can adopt to not let it happen again.

Typically, researchers will heavily rely on using existing survey instruments, slightly tailoring them to accommodate their variation. This is because the existing instruments have already been tested for their reliability and validity, letting the new researchers know what to avoid, as well as making it easy to compare results of new studies to the previous ones. Although, this also comes at the cost of some disadvantages, such as potential copyright issues with using the original instrument [35].

When formulating questions for a new survey instrument, we have to decide if they are **open** ended questions, meaning that the respondents are asked to build their own answer, or **close ended**, meaning the respondents are asked to select an answer from the predefined choices provided by the researcher. Even though open ended questions provide the advantage of not imposing any restriction on the respondent, they can be much more difficult to analyse and may leave more room for misinterpretation, provision of an irrelevant answer, or exhaustion of the participants. Thus, close ended questions are typically the preferred approach, although some open ended questions are often unavoidable.

Finally, to later ensure that we are able to differentiate **consistent** respondents from non-consistent respondents, one may add one or more questions that are, essentially, repeated but written in a different manner than before. If the respondent wrote a response earlier in the survey instrument that is not consistent with their response when that same question is repeated with a different structure, this means that their responses might not be valid for that specific section or even for the entirety of the survey instrument. This is because it could be an indication that they are not qualified to answer that section of the survey instrument or that they are not paying enough attention, which would risk yielding results that we may want to invalidate.

2.1.4 Creating (effective) metrics

To ensure we are able to later analyse the data, we must first decide on a set of **metrics**, also known as **measures**. To classify as effective, a metric should be defined very clearly, with usable mathematical properties while being precise and reliable enough to draw a relevant conclusion from. Another requirement for the metrics is that they must provide at least a partial answer to a specific question aimed at one of the research goals. Typically a single metric is not sufficient to adequately answer even a simple question, so the questions and metrics need to be well connected [35].

Metrics can be of a **simple** or **compound** nature. Simple metrics are defined in terms of a measurement unit and they may include counts, rankings, duration, among other similar measures. Whereas compound metrics are defined in terms of two or more measurement units and can be dimensionless. Compound metrics may include percentages, ratios, or other similar measures.

Another element worth mentioning is the limiting of said metrics. Throughout the survey instrument, the bounds of the metrics should be kept consistent (whenever possible) and reasonable. For example, when we're using numerical values for the metric, the lower bound is typically zero because this is intuitive and leaves no margin for misinterpretation. This is not only beneficial for the respondents but it also makes the data analysis process easier. It is also for this reason that data is normalised before being analysed when using most of the common data processing techniques [35].

When a measure never takes on a particular value or range of values, we should look to **truncate** that specific measure in order to keep the results relevant. This same logic also applies when a question is asking for a count and the most common response is "n or more"; in this case the upper bound of the measure should be extended.

An effective measure is one with a good level of **precision, reliability**, **validity**, and a good relationship with other measures, because results can often be misleading if they are handled independently when they are, in fact, related.

There are two different definitions for a measurement's precision, and both of these are valid and worth working to accomplish [35]. The first one is the size of a metric's smallest unit. If the smallest unit of measure is too small it will, most certainly, lead

to more error. The derived measure should not be any greater than that of the original measures, and typically is even less because the arithmetic combination of measures propagates and magnifies the error inherent in the original values. Therefore, the sum of two measures has less precision than either of them do individually. This is a key thought when creating a **compound** metric.

The other definition of precision is also known as **reliability** and it is achieved when the measurements are consistent across different observations in the same circumstances. Rating scales are notoriously culprits of failing to be reliable as they are naturally a highly subjective choice, and deviating from behavioral and/or subjective components is crucial to achieving reliable measures [35].

Furthermore, validity is also an essential characteristic for a measure, such that a measure may be precise and reliable and still not be of use if it lacks validity. It is, however, a multifaceted concept. For instance, there's **content validity** which values how much the metric reflects the domain it is trying to measure, and **criterion validity** which values how much a metric reflects the measured object's relationship to the relevant standards [35].

Finally, a metric's effectiveness can also depend on how it is used in correlation with the other metrics. For example, two metrics can be reliable, precise and valid but still be measuring the same construct thus being completely redundant.

The questions asked need to be precise and unambiguous. Additionally, we have to make sure that the respondents have sufficient knowledge to answer the questions to ensure we do not gather useless data. When designing the answer options, we must ensure they are mutually exclusive and exhaustive though not too long. Common options include numerical values, close ended answers and ordinal scales. Numerical values are straightforward and usually very effective. Close ended answers can, however, be particularly problematic if they are not carefully designed, because they can be inconclusive and unreliable, as frequently a broader scope is needed. Ordinal scales, such as Likert scales [93], are an alternative to close ended answers as they allow for a more accurate and specific response.

In a self-administered questionnaire, it is also important to consider the format and instructions of the questionnaire. There should be some space between questions, and their respective boxes, arrows, among others, to ensure that the questions are clear for the respondents. There should also be a good contrast between text and background, with a font that is easy to read and bold, underlined and in capitalised text for what needs to be emphasised. The use of italics should be avoided as it makes text harder to read [35].

Additionally, the order in which the questions are placed is very significant. We should start with the easy questions first and gradually increase difficulty to avoid discouraging respondents. Lastly, the questionnaire should be accompanied by administrative information, such as [35]:

- The purpose of the study;

- A description of who is sponsoring the study;
- A (realistic) estimate of the time required to complete the survey instrument;
- How the respondents were chosen and why it is relevant to them.

Another important point of consideration is the length of the questionnaire, both in the number of questions and in a real time estimation of completion. The usual tendency is always to add a few extra questions but we have to be cautious because we can easily demotivate the respondents by doing this [35]. If there are too many questions we will need to remove some, which should be done by identifying one or two topics that are addressed by too many questions and remove some of the less essential questions, or by altogether removing a group of questions related to a topic. The focus is on accomplishing a healthy balance between the data that we are interested in and what the respondents are willing to provide.

Lastly, researcher bias should always be minimised. This is the influence we, as the researchers, may (often subconsciously) have on the respondents through the way questions are asked or the instructions are given. We must strive to develop neutral questions that do not influence the way the respondent thinks about the problem, and ask enough questions to cover the topic. Additionally, care should be taken regarding the order of the questions as one question can easily influence the answer to a later question.

2.1.5 Evaluating the instrument

After constructing the survey instrument comes a step just as important, which is evaluating said instrument. This means [35]:

- Making sure the questions are understandable;
- Assessing the likely response rate, effectiveness and coherence of the follow-up procedures;
- Evaluating the **reliability** and **validity** of the instrument;
- Ensuring that the data analysis techniques used match the expected responses.

There are two noteworthy strategies to organize an evaluation: **Focus groups (FG)** and **Pilot studies (PS)** [35].

FG are mediated discussion groups in which the participants are asked to fill in the questionnaire and to identify any potential problems. They help identifying missing, unnecessary or ambiguous questions.

PS involve administering the survey to a smaller sample, using the same procedures as the original survey. This helps in not just identifying any problems with the survey instrument itself but also with the response-rate and follow-up procedures.

However, the most important goal is evaluating the reliability and validity of the instrument. Reliability measures how well we can reproduce the survey data, as well as the extent of measurement error. This means a survey is reliable if we get the same distribution of answers when administering the survey to two similar groups of respondents. On the other hand, validity is concerned with how well the instrument measures what it is supposed to measure [35].

There are four different types of threat to the validity of the results of this thesis' survey, according to Cook and Campbell [99, 65]:

- **Conclusion Validity** is threatened when the ability to draw the correct conclusions from the outcome of the research is, in any way, compromised. For instance, an insufficient number of samples may hinder the reliability or confidence in the results, thus affecting the conclusion validity of the survey.

- **Internal Validity** is threatened by a possible causal relationship between the research instrument and the outcome. In other words, it is related to the sampling and instrumentation stages, and it represents the degree to which conclusions can be drawn, based on the metrics and the research process.

- **Construct Validity** concerns the generalisation of the results of the survey to the theory or concepts behind it, and it is mainly addressed in the instrument evaluation and validation stage. Therefore, it may be affected by the design of the instrument or the metrics created, i.e., if thehttps://www.overleaf.com/project/5f86e8725e76350001548bb7 metrics are not measuring what they are meant to.

- **External Validity** is threatened by conditions that limit the ability to generalise the results obtained to other scenarios, such as different contexts and strata of the population than the one in which the survey was conducted. The survey can be impacted by factors such as generalisability and replicability if, for instance, the participants are for the most part students or beginners in a certain matter and the goal was to conduct research on the population as a whole, across all levels of expertise. In this case, the sample is not representative and heterogeneous to the overall target population, and thus the external validity of the survey is threatened.

2.1.6 Document the instrument

After the instrument is completed we should document it. For this, we can write an initial descriptive document, called a "questionnaire specification", which should include:

- The objectives of the study;

- A description of the rationale for each question;

- A description of the evaluation process.

After the instrument is administered, the specification should be updated to include:

- Who the respondents were;
- How the instrument was administered;
- How the follow-up procedure was conducted;
- How the completed questionnaires were processed.

One of the main reasons to document the instrument is because a survey instrument can take a long time until it has enough relevant replies for an analysis. This means that it is easy to forget the details of the instrument creation and administration, which is also why it is good practice to keep an experimental diary or log book for any empirical studies [35].

2.1.7 Obtaining valid data

As previously mentioned in Section 2.1, when we administer a survey we should not survey the entire target population because that is typically not efficient or even possible. Instead, we only survey a sample of the population, but we have to be careful to avoid any bias when choosing a sample. We should select a sample that is truly representative of the larger population and that is not expensive to query [35].

To find a sample, we should first define a target population, which is going to be the group of individuals who are in a position to answer the questions we are proposing and to whom the conclusions and results of the survey apply. A valid sample will be a representative subset of the target population. If it is a non-representative subset we can not claim that the results may be generalised to the entire target population.

We can identify the target population from scrutinising the survey's objectives. The more precise the objectives are, the easier it is to define the target population. We can consider a target population while amidst the questionnaire design and we can also re-assess after any FG or PS are instrumented. We will mainly have to ponder whether the analysis results will address the study objectives and if the target population can answer the research questions.

Once we are confident in the target population, we can use a rigorous sampling method. This can be a **probabilistic sampling method**, **cluster-based sampling** or **non-probabilistic sampling method** [35]. A probabilistic sample is a method in which every member of the target population has a non-null chance of being included in the sample. This process will eliminate any subjectivity and obtain an unbiased and representative sample of the target population. There are many different ways to go about this approach but a common option is taking a simple random sample in which every member of the target population has the same chance of being included in the sample. Afterwards a random number generator is used to assign a number to each person, and then randomly

order the numbers in the list and pick the first n members on the list where n is the desired sample size.

Cluster-based sampling involves surveying individuals that belong to certain defined groups. The members of each cluster will give more similar answers to each other than to the members of other groups. However, this also means the analysis will be more complex than that of a simple random sample.

Non-probabilistic sampling is the process of selecting respondents based on how easily accessible they are or on the belief (by the researchers) that they are representative of the population. This type of sampling runs the risk of being biased so they are harder to draw strong conclusions from. However, sometimes it is an option worth considering if the target population is hard to identify, or if it is highly specific and thus of limited availability.

Additionally, we need to determine the appropriate sample size for the survey. An inadequate sample size may lead to results that are not relevant or significant and it prevents us from being able to compare and contrast different subsets of the population. We should contemplate the analysis plan and ensure that the estimated sample size is sufficient to analyse the smallest important subgroups in the population.

2.2 Analysing survey data

After we have designed and administered the survey with the appropriate metrics and ensured that the amount of data collected is sufficient, it is time to analyse the data. But before we proceed to this analysis, responses should be vetted for completeness and consistency. For example, if most respondents answered all the questions we can discard the responses from the respondents that did not. Alternatively, if some respondents have omitted certain questions we can remove said questions from the analysis. This can only be done if the questions are not set as mandatory. For this reason, if incomplete responses are not satisfactory the questions should be set as mandatory. However, sometimes we are able to use all the responses even if some are incomplete. In that case, we would have different sample sizes for each question but we must account for that actual sample size for each statistic. This approach is suited for analysing sample statistics or comparing mean values, but not when analysing correlation or for regression studies [79].

Afterwards, it may be of interest to partition the responses into more homogeneous sub-groups before analysis. This is done because we may want to compare the responses obtained from different subgroups or to report the results for different subgroups separately, and it makes data easier to interpret.

If the data is of ordinal or nominal nature, its analysis can be more problematic than if it were of a numerical nature. It is common practice to convert the ordinal scale to its numerical equivalent and analyse it as such. This can be a reasonable approach, but it violates the mathematical rules for analysing ordinal data and it includes the risk that the analysis will provide misleading results. If the data is approximately normal, there is

a lower risk of misanalysis in also converting it to numerical values, although we should thoroughly understand the scale type of our data to ensure that a conversion is indeed necessary and appropriate [35].

There are typically three main tasks involved in the statistical analysis of a survey [79]:

- Description;
- Comparison;
- Prediction.

Even though all statistical analyses have something in common, it is important to differentiate the process for analysing dynamic or temporal data from that of static data.

One thing that should be done before starting any statistical analysis is **data cleaning**. This means auditing the data for usable and complete values and removes most, if not all, of the uncertainty about the validity of the results. This process can help make use of scarce data as it makes it possible to extract at least some information, almost regardless of how poor the quality of the data is [79].

2.2.1 Data description

The first step of data description is looking at the data. Two different samples can have the same mean and even standard deviation but be, in fact, vastly different. Mean and standard deviation are more sensitive to extreme values than, for example, a median. Therefore by looking at the data in a graph, for instance, we can detect any outlying details and consider the best statistics in order to not be misled by the results or invalidate them.

Because descriptive statistics are subject to error, however, the precision of the estimate is relevant and important to acknowledge so we should quantify this error. The **standard error** is a common way of representing the precision of an estimate [79]. This is an estimate of the standard deviation of the statistic's sampling distribution and mathematically it is calculated by the division of the standard deviation by the square root of the sample size.

Categorical data can be **binomial**, with two categories, or **multinomial**, with more than two categories, and it is typically described through the proportion or percentage of the total in each category. An example of categorical data is one's gender or favourite operating system. Ordinal data, however, is more difficult to analyse as it contains more information than categories. An example of it is a subjective rating scale. The best description of this type of data is its distribution; by listing the percentage of cases for each value through a **histogram** (or a similar representation) we can get more relevant information than through its standard deviation or range [79].

It is also imperative that we analyse the association between measures. For this we can calculate the **correlation coefficient** which measures the amount by which two measures

covary. A coefficient of 1 indicates a strong positive correlation, meaning that for every positive increase in one variable there is also positive increase in the other variable, and a coefficient of -1 is the opposite, for a strong negative correlation. This is moderately sensitive to the range of variation of each variable [20].

2.2.2 Data comparison

One of the main focuses of a data analysis is the comparison between the new data collected and a real or ideal value. Often, statistical comparison compares the difference in average values while taking into consideration the dispersion in the groups' values. This means that, for example, if the values range from 10 to 20 a difference of 5 units is way more significant than if the values ranged from 1,000 to 2,000.

Categorical data is normally compared through a chi-squared test [25] on a table where the rows represent the samples and the columns represent the categories. However, the description and comparison of categorical data is frequently a straightforward test of whether the proportion of some outcome is the same across two samples [79].

Ordinal data comparison can be done with the same table methods as the categorical data, as well as some rank-based techniques. More specifically for rating scale data, a common practice is to compare the means of two samples taken at different points in time [79]. In other words, comparing repeated survey results from different respondents. Calculating the mean in these cases would be pointless because it is very sensitive to skewed values, as mentioned previously. The median suffers a similar problem in that the scale has such few values that it would often be similar from one sample to another with them being in reality very different samples. Thus, the best ways to analyse such type of data include reducing it to categorical data. It can be done by comparing the entire distribution of responses across every sample in the same table or by just focusing on the few categories of highest interest and comparing the proportion of responses across the different samples [79].

As mentioned at the start of Section 2.2, data prediction is the step that usually follows data comparison. It consists in predicting future observations based on the data available: the more data available, the more accurate the predictions will be. However, in the case of this thesis and as in most survey research, there is no interest in forecasting results, and instead we subsequently analyse the quality of the results in greater detail.

2.2.3 Data quality

As stated before, at the start of Chapter 2, data quality is of utmost importance in any scientific or statistical analysis, which includes surveys and specifically questionnaires. There are many factors that can impact the quality of the data, with some of the most common being [79]:

- Organisational problems;

- Imprecise definitions;
- Lack of data validation;
- Missing data;
- Sampling bias.

For the research to be successful, the researchers involved need to come to an agreement about the meaningfulness of the metrics and each of their roles in the organisation. More specifically, the metrics should be defined by people to whom the metrics apply. If this is not the case then the metrics collected may be irrelevant or inconclusive.

Additionally, the measures must have a very precise definition in order to not create uncertainty for the respondents. The lack of precise definitions means these issues will have to be addressed by the researchers when they arise. If multiple people are responsible for collecting the data, perhaps even over a long period of time, this problem is only worsened and can lead to inaccurate results. When a vague definition is detected during the process of building the instrument, the researchers should also be the ones setting a precise definition.

A precise definition for a metric is, however, not enough to ensure the results recorded for it are accurate. We should also make sure the values themselves are possible as well as clearly distinct from each other. This is an easily fixed issue and unlikely to occur but it is of extreme importance as it can make most of the results useless or biased. Data validation problems are detected by performing extensive assertion and consistency checks of the data-set.

Just as important as the data being valid is there not being missing data. Much like the previous case, missing data can lead to some biased results, although it does not always affect the quality of the results and is somewhat common when working with large data-sets [79].

Finally, pure sampling bias is a difficult factor to identify. More so than all the factors mentioned previously, sampling bias is subtle but destructive as it can turn a well-defined, validated and complete data-set into an unusable clutter. This can be caused by different reasons such as **self-election**, which happens when some units in the population put themselves in the position of being surveyed, so only the individuals who choose to be measured provide data. Similarly, it can also be caused by **non-random sampling** due to the lack of knowledge about the population coupled with a bias related to surveying the individuals that are easier to measure [79].

Detecting sampling bias can be done by thoroughly analysing the data and identifying an absence of certain types of data. Correcting sampling bias, though, is difficult and should not be attempted. Alternatively, a possible approach is often to make it clear what subset of the population or what type of respondent has been surveyed and studied [79].

2.3 Conclusion

This Chapter presents a study of the state of the art on SR. It provides a detailed report on strategies for the construction and administration of a survey, from defining the survey's objectives and design, to its evaluation and documentation. Furthermore, this Chapter covers strategy for the analysis of survey data. This includes a study on data description, data comparison, and the evaluation of the data's quality.

Because this thesis can be considered SR, the goal is for this Chapter to serve as a starting point for the rest of this document. The approaches taken and described in the following Chapters are based on the concepts described here, so thoroughly understanding them is vital.

3

MATLAB

In this thesis, one of the main concerns is understanding how the users of *MATLAB* and its clones interact with these languages. Specifically, we want to understand if and how they make use of the languages' OOP features and how satisfied they are with it. For this reason, it is important to first obtain an understanding of these languages. Therefore in this Chapter we first explore *MATLAB*, detailing what its uses are and how it works (Sections 3.1 and 3.2). Later, we also explore *MATLAB*'s most used clones, such as *GNU Octave*, analysing how they can be used and how they differ from *MATLAB* (Section 3.3).

MATLAB is a dynamic and proprietary programming language developed by *MathWorks* that allows for matrix manipulations, implementation of algorithms and plotting of functions or data. It is extensively used in scientific and engineering domains [52]. It was commercially released in 1984, although its origins date back to the 1960s. Currently, it has more than 4 million users worldwide, with 2,000 existing *MATLAB* based books across 27 different languages. Furthermore, 6,500 colleges and universities around the world are using it to teach and research across many different disciplines [52, 50, 58].

MATLAB is designed for quick application development and fast prototyping, not having any statically-declared types. However this also brings a few disadvantages, including a negative impact on the development of reliable and reusable programs and on performance [26].

3.1 Syntax

3.1.1 Variables

Every variable in *MATLAB* is a multidimensional array, regardless of the type of data it consists of. Additionally, a variable can be either **local** or **global**. Local variables, the most common type, are ones that can only be accessed or referenced inside of the function in which they were defined; whereas global variables can be accessed from any function. Listing 3.1 includes example variable value assignments [7].

```
1  % Defining n and initialising it with the numeric value 5
2  n = 5;
3
4  % Defining s and initialising it with the String value 'Hello
       World'
5  s = 'Hello World';
6
7  % Defining (the array) r and initialising it with 5 elements
8  r = [5 6 7 8 9];
9
10 % Defining and initialising a 3-by-3 matrix
11 m = [1 2 3; 10 20 30; 100 200 300];
12
13 % Defining f and initialising it with the function value
       auxFunction()
14 f = auxFunction();
```

Listing 3.1: Variable value assignment examples

3.1.2 Arrays and matrices

Each cell of an array is indexed and can store any type of data. To initialise an array we can separate the elements with either a comma (,) or a space, as demonstrated in Listing 3.2.

```
1  % Defining (the array) r and initialising it with 5 elements
2  r = [5 6 7 8 9];
```

Listing 3.2: Array declaration example

A matrix is a two-dimensional array, the most common type of array, and is very often used in linear algebra. To initialise a matrix, we simply separate the rows of the array with semicolons, as shown in Listing 3.3.

```
1  % Defining and initialising a 3-by-3 matrix
2  m = [1 2 3; 10 20 30; 100 200 300];
```

Listing 3.3: Matrix declaration example

Additionally, we can declare a matrix full of zeros, ones, or even random values using the methods presented in Listing 3.4.

```
% Defining a 3-by-3 matrix full of zeros
m1 = zeros(3)

% Defining a 3-by-3 matrix full of ones
m2 = ones(3)

% Defining a 3-by-3 matrix full of random numbers between 0 and
    1
m3 = rand(3)
```

Listing 3.4: Additional matrix declaration examples

3.1.3 Operations

Mathematical or logical operations can be done in *MATLAB* through the use of **operators**. These operators are symbols that will tell the compiler which manipulations to perform, and they work on both scalar and non-scalar data. There are five different types of operators [56]:

- Arithmetic Operators;
- Relational Operators;
- Logical Operators;
- Set Operators;
- Bit-wise Operators.

Arithmetic Operators can be applied to both matrices and arrays and they are differentiated by the period (.) symbol whenever necessary (some operations are identical in both structures). In Table 3.1 we present examples of the various arithmetic operators.

	Array Operator	**Matrix Operator**	**Array Function**	**Matrix Function**
Addition	A+B	A+B	plus(A,B)	plus(A,B)
Subtraction	A-B	A-B	minus(A,B)	minus(A,B)
Multiplication	A.*B	A*B	times(A,B)	mtimes(A,B)
Right Division	A./B	A/B	rdivide(A,B)	mrdivide(A,B)
Left Division	*B.\A*	*B\A*	ldivide(B,A)	mldivide(B,A)
Exponentiation	A.^B	A^B	power(A,B)	mpower(A,B)
Transpose	A.'	A'	transpose(A)	ctranspose(A)

Table 3.1: Arithmetic operator examples.

Relational Operators perform element-by-element comparisons between two arrays and return an array of the same size, with elements set to true (1) where the relation is

true and set to false (0) where it is false. In Table 3.2 we present examples of the various relational operators.

	Operator	Function
Equal to	A==B	eq(A,B)
Greater than or equal to	A>=B	ge(A,B)
Greater than	A>B	gt(A,B)
Less than or equal to	A<=B	le(A,B)
Less than	A<B	lt(A,B)
Not equal to	A =B	ne(A,B)
Array equality	-	isequal(A,B)

Table 3.2: Relational operator examples.

Logical Operators also perform element-by-element comparisons and much like with relational operators, logical operations will return an array of logical values (1 for true, 0 for false) to indicate fulfillment of a condition, such as AND, OR, XOR and NOT. In Table 3.3 we present examples of the various logical operators.

	Operator	Function
AND	A&B	and(A,B)
OR	A | B	or(A,B)
XOR	-	xor(A,B)
NOT	Ã	not(A)

Table 3.3: Logical operator examples.

Set operators are mostly used to compare the elements of two sets (arrays of numbers, dates, times, among other types of data) to find commonalities or differences. Additionally, they can be used to perform joins, intersections, unions, and other similar operations between arrays.

Bit-wise Operators are used to set, shift or compare the bit patterns of numbers in an array. These are extremely efficient as they are directly supported by the majority of processing units.

3.2 Object-Oriented Programming

OOP is a popular programming approach based on the concept of "objects", which contain data in the form of attributes or properties and code in the form of methods. It improves the ability to manage software complexity, which is extremely important when developing large applications and/or data structures [61, 55].

OOP enhances *MATLAB*'s support to modularity in its programs, reducing the code maintenance needed and improving its scalability, reliability and reusability. It involves using:

- Class definition files;
- Classes with reference behaviour;
- Events and listeners.

For this, *MATLAB* organises class definition files into five modular blocks, each one delimited by its own specific keyword and the 'end' keyword [55]. Listings 3.5 and 3.6 demonstrate examples of class definition syntax.

```
1  classdef ClassName
2      properties
3          Property1
4          Property2
5          ...
6      end
7      methods
8          function obj = ClassName(arg1, arg2, ...)
9              ...
10     end
11     events
12         EventName
13     end
14 end
```

Listing 3.5: Class definition syntax with example blocks

```
1  classdef ClassName
2      enumeration
3          EnumerationName
4      end
5  end
```

Listing 3.6: Enumeration class syntax

3.2.1 Class definition blocks

Class definition blocks contain the class definition and the specification of its attributes and superclasses. It is initialised with the keyword `classdef` and within it can be included properties, methods, events or enumeration blocks. Listing 3.5 includes an example of a lass definition block [55].

3.2.2 Properties blocks

A properties block includes all the data items that we need to represent a class, although a class may have multiple properties blocks with different settings to better accommodate

the object to the desired scenario. Properties can be defined with constant values, or with values depending on other properties, or even without storing any values. Constant properties do not change so they can be accessed by simply referencing the class's name. Listing 3.5 includes an example of a properties block [55].

3.2.3 Methods blocks

Classes can also contain multiple methods blocks. Much like the properties blocks (Subsection 3.2.2), different methods blocks can specify different attribute settings that only apply to the methods inside that specific block. **Methods** are the operations that can be performed on the object. Essentially, they are what controls the class's properties. Listing 3.5 includes an example of a method block [55].

3.2.4 Events blocks

Events are triggers that activate when something specific happens. For example, we might want to trigger an alert or execute a function when a certain numerical property goes over a threshold. A class can also have multiple events blocks and each one of them can specify different attribute settings. Listing 3.5 includes an example of an events block [55].

3.2.5 Enumeration blocks

Enumeration blocks include enumeration definitions, which are used to represent fixed sets of values (of the same type). These blocks are almost always contained in their own separate class in which a class definition block (Subsection 3.2.1) will have a single enumeration block. We call these classes enumeration classes, as that is their sole purpose, and they can be derived from other classes inheriting the operations of the superclass. Listing 3.6 includes an example of an enumeration block [55, 53].

3.3 MATLAB Clones

Despite *MATLAB*'s flexibility and extensive capabilities, its users have, over the years, built many different alternative languages which we call *MATLAB*'s *clones*. Not only do they try to improve some of the lacking functionalities of *MATLAB*, but they also solve a major concern, and perhaps the most important catalyst: the price. *MATLAB* has a significant monetary cost, and these clones tend to be free of charge.

3.3.1 GNU Octave

GNU Octave is a free open-source clone of *MATLAB* and the one with the most similarities. It started being developed in 1988 as an auxiliary software to an undergraduate-level textbook related to chemical reactor design. Throughout the years, *Octave* evolved to

be much more than that. Today, thousands of people are using *Octave* to teach, learn, research and even for commercial applications [22].

It is extremely compatible with *MATLAB* and can be used to solve linear and nonlinear problems numerically since it provides support to matrix data types and operations, much like *MATLAB*. Because of *Octave*'s extensive similarities with *MATLAB*, users are able to write their code in one language and still maintain the ability to interpret it in the other language. Developers can even use *Octave*'s *traditional mode* which makes it interpret the code in an even more *MATLAB*-compatible setting [22, 84].

However, because *Octave*'s parser allows for specific syntax that *MATLAB* does not, it is possible for programs written for *Octave* to not run in *MATLAB*. And even though both languages have object-oriented capabilities, their implementation is different [84].

3.3.2 Scilab

Scilab is another free open-source clone of *MATLAB*. It was first released in 1994 and still has a growing community, with over 100.000 downloads per month worldwide. *Scilab* is being mainly developed by the *Scilab* team within ESI Group. Users have the ability to alter the software to suit their needs and also share that altered version [80, 82, 15].

An advantage of *Scilab* is its extensive documentation and helpful resources in the form of a wiki, books, tutorials, and mailing lists [81].

It has a very similar syntax to *MATLAB* and is therefore, for the most part, mostly compatible with *MATLAB*. More so, *Scilab* includes a function, *mfile2sci*, which allows a user to translate *MATLAB* files to *Scilab* [81, 78].

3.3.3 Rlab

Rlab is a high level language similar to *MATLAB*, that aims to provide fast prototyping, program development, data-visualisation and data processing. As of the 13th of July, 2001, it is no longer under active development. However, a distinct version called *rlabplus* tries to improve the original *Rlab* and is still under active development at the time of this writing [83, 37].

Although *Rlab* does not try to be a *MATLAB* clone, it does provide an environment in which a user can do matrix math, using concepts and features similar to those in *MATLAB*. The language tries to improve on *MATLAB*'s syntax and semantics.

3.3.4 Conclusion

This Chapter provides an overview of *MATLAB* and its concepts. One of the main concerns of this thesis is the concept of OOP with *MATLAB* and consequently *MATLAB*'s support to modularity. Therefore these are focal points for this thesis' survey. More concretely, the goal is to understand the extent to which *MATLAB*'s OOP features used by the community, as well as how knowledgeable *MATLAB* users are about OOP.

Furthermore, this Chapter also provides insight on languages very similar to *MATLAB*, so-called *MATLAB clones*. This includes languages such as *GNU Octave*, *Scilab* and *Rlab*, and these are the ones covered in this Chapter. The analysis of the compatibility these languages have with *MATLAB* makes us more confident to broaden the target population for our survey, thus also including the users of these languages.

4

Related Work

In this Chapter we will be analysing work similar to this thesis' survey in order to have a better understanding of different techniques used and be able to optimise the way we approach problems encountered. We thereby introduce two types of related work: Surveys done to communities of users of MATLAB or similar languages (Section 4.1) and surveys done to related, but more distant, communities (Section 4.2). We will analyse the methods and strategies used to conduct these surveys as well as the results obtained from them, when possible. In the end, we will summarise and compare all of the different works analysed with this thesis' survey (Section 4.3).

4.1 Surveys done to MATLAB users and adjacent communities

In 2017, a thesis was conducted by Duarte [13] in which the authors surveyed the *MATLAB* and *Octave* programming communities concerning the limitations in the support to modularity offered by *MATLAB*, using *Qualtrics* [69] as their questionnaire administering tool [13]. Their target population, in this case, were developers that had worked in large projects using *MATLAB* in the previous ten years. To reach this target, after executing their PS (see Subsection 2.1.5), they used two different selection methods:

1. Find *MATLAB* (and its clone languages) programming communities online;

2. Create a filter on search engines to search through papers and email the authors.

In the first method they searched for online communities where programmers share their ideas and knowledge, and discuss about *MATLAB* and its clones. These communities were found in a few social networks such as *LinkedIn* [42] and *Facebook* [16], as well as at *MATLAB Central* [59], a forum hosted by *Mathworks*. Across every one of these platforms, however, the *MATLAB* communities seemed to be larger than their Octave counterparts. Some of the communities approached could reach as much as 225,000 members, as was the case for *MATLAB Central*'s community.

In the second method, the authors started by choosing *Google Scholar* as their search engine, in which they then developed a filter using the keywords *"matlab"*, *"software*

engineering" and *"modularity"*. The result would yield every paper that mentioned these specific terms. Using this method they were able to find a total of 2,000 results. Given this high number of results, they further restricted the filter by limiting it to only show results from 2012 or earlier and that had been cited at least 5 times. The end result was 26 papers, giving them a total of 79 available researcher email addresses.

Finally, they obtained 42 responses with a completion rate of 76.19%, meaning that 76.19% of the respondents completed the questionnaire and the remainder were only partial responses. The total number of responses, however, was less than the expected and desirable. The participation invitations sent out to the online communities was expected to carry out the vast majority of the responses. This was not the case, however, and the underwhelming number of responses put the validity of the study at risk. More specifically, it threatened: Conclusion validity, meaning the validity of any conclusion drawn was not guaranteed as a consequence of the smaller sample size; Internal validity, because there is a higher risk in failing to handle the variables properly or in choosing the wrong statistical analysis; External validity, because the smaller sample size may also introduce an unintended bias which can limit the generalisation of the results obtained.

To conclude, the questionnaire and its questions included specific topics which might have prevented it from being easily accessible to a larger number of respondents. Additionally, the emails could have been sent out earlier than they were to allow the recipients more time to respond. Furthermore, the authors could have explored smaller and more active communities as the big communities of the likes of *LinkedIn* and *Facebook* are relatively saturated, meaning that the actual participation from them is not the highest despite these having the highest number of members. These limitations are mentioned in their document and they were taken into account in this thesis, which allowed us to gather a higher number of responses and that in turn results in drawing more accurate conclusions by distinctly diversifying and categorising the target community.

In 2000, an investigation was conducted by Cretchley et al. [6] into the effects of scientific software, specifically *MATLAB*, on first year university mathematics students [6]. Their sample comprised 184 students who completed questionnaires at both the beginning and the end of a semester. Data related to the participants' mathematical skills, feelings, attitudes and beliefs were collected via a variety of different means: a diagnostic test upon entering the course and two examinations at the end of the course; students' responses to tasks on five assignments and to questionnaires attached to them; retrospective views expressed in FG and interviews; and a Likert-scale questionnaire administered at the beginning and end of the course.

The data from the questionnaires were subjected to factor analysis, in which 6 different factors emerged and the Pearson correlation coefficients between them was measured. Afterwards, the responses were analysed by sex, native language, mode of study and degree. The authors found evidence that suggested that the use of technology like *MATLAB* had a strong impact on the learning strategies by particular students, and that almost all students responded positively to using *MATLAB* for ease of computation and graphing [6].

Although this study is quite dated, which means that some of its conclusions may now be outdated due to software and technological improvements, the strategies used in the analysis conducted are still sound.

In 2005, a study was also conducted by Wallin et al. [98] in which, through questionnaires and interviews the authors measured the experience and satisfaction of 77 first-year engineering students with the use of *MATLAB* and other tools during the course [98]. Furthermore, in 2010 there was another survey by Hoole [27] on the experience that graduate students had on using *MATLAB* in their course [27]. In both of these cases, the authors conducted questionnaires at the beginning and end of the courses, and the results helped them understand the students' satisfaction and experience with the different course materials available to them. For instance, the authors found that although 43% of the students leaned towards a negative opinion towards the form of instruction of the course, the majority still regarded the course as meaningful and useful for their future courses.

In this thesis, the questionnaire is the only survey instrument used and it is more focused than the aforementioned investigations, in 2000, as it is strictly focused on the users of *MATLAB* and its clones, and how they interact with these environments. We are not in studying how their programming habits affect their daily lives. There are, however, some similarities in the analysis of the data gathered from the questionnaire, in which a series of factors are extracted and analysed.

During the fall of 2019, a *SciPy* [9] user questionnaire was conducted by Gwóźdź [24] via a *Google Forms* [17] questionnaire. The author promoted the questionnaire through the *SciPy* mailing list and website, several relevant university departments, *Twitter* [94], blogs, physical mailing lists, among others. The goal was to gather some helpful feedback to take into account for future development of *SciPy* and improvement of the documentation. The questionnaire received a total of 185 responses, which the author reckoned was neither too low to be deemed irrelevant, or too high as to be too difficult to analyse [24]. The questionnaire featured a series of multiple choice and 5-point Likert scale questions, and at the end it included three open ended questions in which they asked the respondents if they had positive comments, negative comments, or any ideas for improvement, respectively, for *SciPy*'s documentation. The results were then thoroughly analysed and deemed useful, and they could serve as a road map for future *SciPy* development and for the improvement of the documentation. For instance, the results showed that the majority of respondents were satisfied with the documentation of that time, and that 84.2% of them were able to find the information they were looking for quickly in said documentation. Furthermore, the results provided a clearer understanding of which parts of the documentation were more commonly used and how the documentation was browsed.

The *SciPy* questionnaire comprised of 11 questions, and it was more focused on the user feedback component through its open ended questions. In contrast, this thesis' questionnaire covered a wider range of topics across a higher number of questions and comprised as few open ended questions as possible, as to facilitate the analysis and reduce the risk of misinterpretation, as well as to not exhaust the participants. However, the

sampling strategy and number of responses are similar.

4.2 Surveys done to more distant communities

In this Section, we analyse surveys done to communities of other programming languages, understanding how the communities were reached and what questions were asked. It is useful to obtain an understanding of strategies taken before as they are often applicable to other environments such as this thesis'. For this reason, the surveys analysed in this Chapter serve as inspiration for this thesis' survey.

In 2020, *JetBrains* and the *Python Software Foundation (PSF)* surveyed the *Python* community concerning a wide variety of topics, with the purpose of identifying the latest trends and gathering insight into what *Python* development looked like in 2020 [31]. They have been conducting this questionnaire every year, thus they can compare results to the years prior and analyse the changes and the evolution of *Python* development as a whole.

After the filtering of duplicate and unreliable responses, they managed to gather more than 28,000 responses in the months of October and November of 2020. The respondents were found through the promotion of the questionnaire on numerous platforms, including https://www.python.org [18] (*Python*'s official website, owned by *PSF*), *PSF*'s blog, *Twitter*, *LinkedIn*, *Reddit*, and official mailing lists. In order to prevent bias, channels associated with a specific product or service were not used. The topics covered by the questionnaire were: General usage of the language; Reasons for using the language; Language versions being used; Frameworks and libraries used; Technologies and cloud platforms used; Development tools used; Employment, work and age. With an extensive array of questions they were able to gather extremely specific results when combining responses from different topics. For example, they could tell which version of *Python* was most used for web development or for data analysis, or how the version of *Python* used correlates with the user's age. Therefore, in this case there were some clear upsides of having a lengthy questionnaire. Because the number of participants reached was relatively high, the mortality in the sample did not preclude the authors from reaching significant conclusions, which can occur in a survey of smaller scale.

Similarly, a yearly questionnaire is also conducted in which they analyse the state of the entire developer ecosystem in order to identify the latest trends in tools, languages and technologies being used, among many other factors [30]. In the 2020 edition they gathered responses from almost 20,000 individuals that were reached through *Facebook* ads, *Quora* [28], *Codefund* [12], among some other platforms, as well as through *JetBrains*' own communication channels. This questionnaire was made available in nine different languages to minimise potential bias and make it more accessible.

To further reduce the sampling bias, the results generated went through three weighting stages to display a more realistic picture of the worldwide developer population. In the first stage, they gathered the responses collected while targeting different countries and then applied estimations of the populations of developers in each country to those

responses. In the second stage, they forced the proportion of student or unemployed respondents to be 17% in every country. This 17% figure was an estimate of their populations that they had gathered from the previous year's questionnaire, thus this helps maintain consistency with the previous year's methodology. For the third stage, however, they solved a system of more than 30 equations in which they calculated, for developers from each country, the shares for each of the more than 30 programming languages, as well as the shares for those who said they "currently use *JetBrains* products" and "have never heard of *JetBrains* or its products". These shares then became constants in the equations. The end result allowed for an intricate analysis, covering a wide range of topics concerning the entire developer ecosystem. Despite the efforts to reduce it, there was naturally still some bias left, since users of *JetBrains* products might have been more likely to complete the questionnaire than other developers, but this does not discredit the successful approach taken to reduce most of the sampling bias.

Both of the aforementioned 2020 questionnaires had a high number of responses, higher than could be expected from this thesis' survey. Their reach and analysis capabilities are much greater than this thesis', and for this reason they can afford to be more ambitious in their approach. Conversely, this thesis' survey is more targeted, focusing only on users of *MATLAB* and its clones, and some their habits and opinions, and it does not suffer from the bias that the *JetBrains* surveys suffer from sharing the questionnaire with their own community of users.

In 2011, a questionnaire was conducted by Prabhu et al. [68] on the prevalent programming practices of a community of researchers from diverse scientific disciplines at a doctoral-granting university [68]. This questionnaire was promoted through e-mail to randomly selected researchers from the university's database. The 114 researchers who replied to said e-mail displaying interest in the questionnaire were then led to an interview with the authors. The study then led to results and conclusions indicating, for instance, that new software tools and techniques were necessary to unlock the potential of (what was at the time) high-performance computing and accelerate the pace of scientific advancements as the tools available did not meet the needs of computational scientists. Participants showed to be unsatisfied with the speed of their programs, and stated that performance improvements would drastically accelerate their research. Similarly to some of the works mentioned in Section 4.1, this survey differs slightly from this thesis' survey in that it's not focused strictly on the users of a specific software tool or environment, which is in the context of this thesis is *MATLAB* and its clones. Instead, the survey characterises the scientific computing environment at Princeton University. The results of the survey were split into three different themes, and with each theme the authors posed a set of questions to which they answered through patterns observed in the data collected. For instance, the authors concluded that most scientists were unsatisfied with the speed of their programs and believe that performance improvements would significantly improve their research.

4.3 Related Work Comparison

Surveys can be differentiated by several different variables, such as their target population, the number of responses to the survey instruments, the analysis strategies used, the tools used, among others.

In Table 4.1, we compare this thesis' survey with the surveys mentioned in this Chapter (Sections 4.1 and 4.2) by displaying some of the differentiating elements side by side. Note that for row "Hoole [27]", the number of responses is estimated through the charts presented in the publication, as the exact number is not disclosed. Furthermore, note that some of the studies do not use any survey tool to administer their instrument, or they do not disclose it, which is why some fields are blank in this column.

Firstly, most of the surveys target a different population, as can be seen in the "Community" column. Secondly, in the "Number of responses" column one can observe that, when excluding *Jetbrains'* surveys, most surveys obtained between 100 and 200 responses. This thesis' survey is among the surveys (presented here) with the most responses.

Additionally, the surveys are promoted through different means from each other. However, one can observe that most of these are promoted through various online platforms and community forums as these are typically used by a large number of users, which makes it easier for a survey to gain more exposure when shared in them.

Finally, the questionnaire administering tool, or "Survey tool" used is also different among the various surveys as this is mostly dependent on the authors' intentions and necessities in their survey instrument. For instance, *Jetbrains* hosted their survey instruments directly on their website, which allowed them to fully customise them to accommodate for their preferences and necessities.

With the community of users of *MATLAB* and its clones as its target population, this thesis' survey gathered 215 responses. The survey instrument was created using *Google Forms* and shared through *MATLAB Central*, *ResearchGate*, *Reddit*, *LinkedIn* and *Discourse*.

Table 4.1: Related work comparison.

	Community	Number of responses	Promoted through	Survey tool used	Goals	Conclusions
Duarte [13]	*MATLAB* and *Octave* users	42	*MATLAB Central* *LinkedIn* *Facebook* Researchers	*Qualtrics* paid version	Assessing whether MATLAB developers recognize the limitations to modularity in *MATLAB*.	*MATLAB* developers care about tangling of concerns. There are no evidences that the limitations in the support to modularity in *MATLAB* affect the maintainability (and readability) of the code.
Cretchley et al. [6]	First year university mathematics students	184	Throughout the course	-	Investigate the effect of the use of *MATLAB* for leaning mathematics in a sample of undergraduate mathematics students.	Early difficulties with access and syntax were quickly overcome. There was evidence of heightened interest and enjoyment in students over the duration of the study.
Wallin et al. [98]	First year university engineering students	77	Throughout the course	-	Obtaining feedback that would serve as a basis for modifications to the *MATLAB* course in later years.	The conclusions drawn from a trial instruction of student volunteers can differ tangibly from the experiences that are obtained from a required course. The majority of students had a positive reaction to the form of instruction. However, 43% lean towards a negative opinion.
Hoole [27]	Graduate students	Approximately 100	Throughout the course	-	Confirm the benefits of the addition of programming to the engineering courses	Students affirmed the benefit of the approach taken in future offerings of these courses. Initially poor programmers had difficulties with the programming assignments.
Gwóźdź [24]	*SciPy* users	185	*SciPy* mailing list Universities *Twitter* Blogs	*Google Forms*	Obtain feedback to be taken into consideration for future development of *SciPy*'s documentation.	The results appeared useful and could serve as a helpful roadmap for future development. The majority of users were satisfied with the documentation of that time.
JetBrains [31]	*Python* users	28,000	Python.org Blogs Mailing lists *Twitter* *LinkedIn* *Reddit*	Jetbrains.com	Getting to know the current state of the language and the ecosystem that surrounds it.	*JavaScript* is most popular language for developers to combine with *Python*. The preference for *Python* 3 over *Python* 2 keeps increasing every year. *Linux* is the most used operating system by python developers.
JetBrains [30]	Any software developer	20,000	Social media ads *Quora* *Instagram* *Codefund*	Jetbrains.com	Identifying the latest trends in the tech industry.	*JavaScript* is the most popular language. A majority of the respondents develop for web backend. Only 4% of the respondents had used *MATLAB* in the year prior to the survey. The 5 fastest growing languages are *Python*, *TypeScript*, *Kotlin*, *SQL*, and *Go*.
Prabhu et al. [68]	Scientific researchers	114	E-mail	In-person interviews	Gathering insight into the programming practices of researchers within the university. bbbb	Scientists can benefit from faster computation. Current programming systems and tools do not meet the needs of computational scientists. Scientists tend to want results immediately, despite most tools assuming will invest time and energy in mastering a particular system.
This survey	MATLAB and clones users	215	*MATLAB Central* *ResearchGate* *Reddit* *LinkedIn* *Discourse*	*Google Forms*	Getting to know the community of users of *MATLAB* and clones. Understanding how the community uses these languages. Understanding how satisfied the community is with *MATLAB*'s current support to modularity.	Approximately 13% of *MATLAB* users do not use m-files when working with *MATLAB*, beginners and experts alike. The more experienced *MATLAB* users, as well as the ones who focus on their programs' maintainability and reusability are more satisfied with *MATLAB*'s support to modularity.

4.4 Conclusion

This Chapter presents related work, providing a detailed description of each one's strategies and results, and how this thesis differs from those.

We start by focusing on surveys done to *MATLAB* users and adjacent communities, from which some similarities can be observed, for instance in the sampling strategies, in the number of responses, and in the analysis of the data acquired. Later, we provide insight on surveys done to more distant communities, and these are distinct from this thesis in the way they approach their target population as well as in the way they analyse their acquired data, although some similarities can be found in the way the questionnaires are structured and the questions are formulated. Finally, the Chapter includes a summary and comparison of the different surveys presented as well as this own thesis' survey.

5

Survey Planning

In this Chapter we explore the planning of strategies for the questionnaire. We start by presenting an analysis of the target population and the sampling strategy (Section 5.1) Then, we explore the community feedback gathered during this stage (Section 5.2). Furthermore, we review different online questionnaire administering software (Section 5.3). Afterwards, we analyse the construction of the questionnaire specification (Section 5.4) as well as of the questionnaire itself (Section 5.5). Finally, we examine the threats to the validity of the results of this study (Section 5.6).

5.1 Target population and sampling strategy

The identification and selection of a target population is important, as this directly influences the conclusions we are able to deduct from the results gathered from the questionnaire.

For the study, we wanted to target the population of users of *MATLAB* and any of its clone languages, regardless of their level of experience or background. For this we had to set the boundaries and limits of what is considered a *MATLAB* clone, as this is not officially documented, and thus we decided to include users of *GNU Octave*, *Scilab* and *Rlab* (see Section 3.3) in addition to the *MATLAB* users.

When looking for potential online communities where we could share the research goals and more specifically gather valid responses for the questionnaire, we would search for communities primarily centred around the use of *MATLAB* or the use of any of *MATLAB*'s clones. These communities would, preferably have a high number of members, whose profiles may vary from beginners to experts, from people who program strictly as a hobby to full-time professional programmers, from students to researchers. We analysed multiple online communities of users of *MATLAB* and its clones, and the ones of interest initially were:

- *MATLAB Central* forums;
- *MATLAB Central File Exchange*;

- *ResearchGate* questions;
- *MATLAB*-related subreddits;
- *MATLAB*-related groups on *LinkedIn*.

MATLAB Central [59] is a platform hosted by MathWorks [58]. According to the most recently analysed data by MathWorks, in January 2019, the platform was home to 365,000 contributors accumulating 172,000 visits per day and more than 13 million page views per month. Because this community is strictly focused on *MATLAB*, this was certainly a source of valid responses for us because almost every respondent will have worked with *MATLAB* in the past. Also hosted on *MATLAB Central*, there is the File Exchange [60] platform which allows users to publish and share any data such as custom applications, code examples, functions or scripts.

Similarly, *ResearchGate* [75] is a professional network for researchers and scientists with over 20,000,000 members. However, it is not solely dedicated to *MATLAB* or any of its clones. *ResearchGate* is used by scientists and researchers of diverse scientific disciplines and topics. Thus, when a post is made the user should choose a selection of keywords to associate the post with certain topics and make it stand out. This means the potential respondents will likely be people who searched for one or more of those specific keywords [76]. Our selection of keywords included: "*MATLAB*"; "*Scilab*"; "*Octave*"; "*Rlab*"; "Survey"; "Research"; among others. This means that anyone specifically searching for one of the languages included or for survey research was able to easily find it.

Reddit [73] is a platform that is home to an almost endless number of communities covering virtually any possible topic, including *MATLAB*, any of its clones, programming, education, research, among many others. We promoted the questionnaire in some of these communities, which can consist of anywhere from hundreds of members, for example in the case of the *Octave* subreddit, to millions of members, in the case of less specialised ones such as the programming subreddits [70, 72, 71, 72].

LinkedIn [42] is a professional network with more than 756 million users worldwide, and just like similar platforms it hosts many different groups where users can discuss anything related to that group's topic. This was also a good place to promote the questionnaire in groups related to *MATLAB* and its clone languages languages because the respondents would have to, at the very least, be interested in the topic to even have access to the questionnaire [39]. Most of these groups require an invitation to access, but typically they are quick to accept an invitation request. This simply serves as a way to reduce spam and unproductive discussions. Upon searching for groups through *LinkedIn*'s search feature using keywords such as *MATLAB* and *Octave*, we were able to find multiple groups with a wide range of members. A few groups worthy of mention include "MATLAB Users and Integrators", with approximately 40.000 members [45], "Matlab for beginners and experts", with approximately 8.500 members [44], "GNU Octave users and developers",

with approximately 900 members [40] and 'Scilab Software', with approximately 1.700 members [47].

With the exception of the *MATLAB Central File Exchange*, these platforms act as traditional online forums where users are able to create their own posts and other users are able to comment and discuss that post. On some of the higher traffic platforms, such as the *MATLAB Central* forums and *ResearchGate*, where the post may quickly get lost amid other posts, we would frequently repeat the post to further encourage users to respond to the questionnaire in case they missed the previous posts. Encouraging responses on public forums provides multiple positive effects: not only does it enable a large sample size, but it also allows for a greater amount of participant feedback via the comments.

On the other hand, on *MATLAB Central File Exchange* we are able to sort publication by their latest submission dates and have access to their total number of submissions, downloads, and their reputation on the website. We can then initiate contact with the active *MATLAB* population via email, with those that willingly share their address. A positive effect of this form of contact is the direct, one-on-one conversation where the participant can very easily provide honest feedback in a response email, and that they may be more inclined to respond to that direct message rather than a post on a forum which is directed at the entire user-base of that specific forum.

5.2 Community feedback

During June and July of 2020, we posted on each of the chosen platforms (see Section 5.1). This allowed us to get a sense of what sort of response we could expect from each different source, and also gather some initial feedback concerning our intentions of later sharing the questionnaire with these communities, or if anyone knew of similar work to this study that we could learn from. During this period, we also requested and received invitations to join and permission to post on multiple restricted LinkedIn groups.

Thus, we posted on *Reddit* (specifically the "MATLAB" and "EngineeringStudents" subreddits, as these seemed to be very active communities with *MATLAB* as one of their main topics of discussion), on the *MATLAB Central* forums and on *ResearchGate* [62, 77, 64, 88]. The posts were welcomed and well received on all of the three platforms. We received positive comments on every post, indicating to us that those were good candidates of communities to approach. Additionally, it became clear that *Reddit* was, most definitely, the most responsive platform of the three, which would later be a decisive factor for when choosing a platform in which to first administer the questionnaire (see Section 6.1).

5.3 Questionnaire tools

There are many different online tools that can be used to construct and administer a questionnaire. The selection of this questionnaire administration software is based on

different factors such as the desired number of questions, expected number of respondents and the monetary budget available. The following Table 5.1 includes a comparison of the current benefits offered by some of the candidates, by both their free and paid versions [69, 92, 17, 96, 3, 89]. Some relevant properties to analyse are the limit of questions supported, the limit of respondents and support for a full data export. In their free versions, most of these tools are limited in one or more of these properties. However, in their paid versions most of the restrictions are eliminated and some advanced options are enabled, such as more question types or better customisation.

Table 5.1: Questionnaire administration tool comparison. Note that *Google Forms* does not currently have a paid version.

	Free version	**Paid version**
***Survey Monkey** [91]*	10 Questions limit 100 Respondents limit Very few question types No data export Limited result analysis	Unlimited questions 7500 Responses/month Many more question types Data export Better look customisation
***Typeform** [95]*	10 Questions limit 100 Responses/month No logic jumps Good customisation Basic results export	Unlimited questions 10000 Responses/month Skip logic More question types Branch logic
***Google Forms** [17]*	Unlimited questions Unlimited respondents Skip and branching logic Limited visual customisation Data is automatically collected in *Google Sheets*	Not applicable
***Qualtrics** [69]*	15 Questions limit 100 Respondents limit 8 Question types No visual customisation Raw data export	Unlimited questions Up to 22 question types Advanced questionnaire logic Advanced results reports
***Alchemer** [2]*	Unlimited questions 100 Respondents limit No questionnaire logic Raw data export Few question types	Unlimited respondents Questionnaire logic More question types API integrations Data encryption
***SurveyLegend** [90]*	Unlimited questions Unlimited respondents No data export Skip logic No branching	Data export Advanced branching

For this thesis' survey, we use *Google Forms* since it provides all the conditions needed while being completely free of cost: Unlimited questions and responses, sufficient questionnaire logic (including branching logic) and it allows for a full data export. None of the other free options analysed manage to satisfy all these three important criteria simultaneously.

Zenodo is a platform specifically designed for users to share research publications, datasets, software, among many other things [100]. Through *Zenodo*, users are be able to upload the dataset along with a title and description, as well as keywords to help

other users find it quicker. To allow the data collected to be used in the future for any research or scientific interest that might occur, we have published the raw responses on *Zenodo* [101].

Additionally, after the research is complete, we intend to publish the raw responses on *Zenodo* [100] to allow that data to be used in the future for any research or interest that might occur.

5.4 Questionnaire specification

To help us through this planning phase, we used *Google Sheets* [23]. On this platform, a user is able to create a file containing as many spreadsheets as needed. It allows for the creation of countless tables and graphs of all shapes and sizes, with many different options of formatting, or for a concise organisation and manipulation of data. This spreadsheet functioned, for us, as a documenting tool, also known as a "questionnaire specification", during the creation and construction of the form. Initially, it contained:

- A list of the research questions and the corresponding sub-questions;
- a list of the null hypotheses of the research;
- a list of potential threats to validity;
- a table including the form's questions, including each question's type, description, extracted factors, among other data;
- miscellaneous "brainstorming" notes.

Later on, during the analysis stages of the research, this spreadsheet also served as a tool where we could construct presentable result tables that would later be presented in Chapter 6.

This tool allowed us to be more organised with the data and notes in a way that would have been very hard to maintain using paper or another software. Additionally, it allowed us to very easily and quickly calculate some simple descriptive statistics of the data using its mathematical capabilities.

5.5 Building the questionnaire

5.5.1 Questionnaire's Structure

After analysing the target population, gathering initial feedback and creating the questionnaire specification, it was time to start building the questionnaire. For this, we had to take a close look at the initial research questions and sub-questions (referenced in Section 1.3), and from those build a coherent and focused questionnaire that would meet all of the criteria. Using the questionnaire specification, we designed the layout for the

questionnaire, dividing it in different sections with each containing a series of questions in a deliberate order and pattern.

The instrument comprises 3 main sections with questions associated to the research questions, as well as 2 smaller subsidiary sections, one in the beginning and one in the end of the questionnaire:

- Section 1: The introductory section, where we make the participant acquainted with the study in a short, concise text. Additionally, we ask the participants where they learnt about this survey;
- Section 2: In this section we ask the participant about their programming experience;
- Section 3: This section is focused on the importance the respondents give to the reusability of their programs;
- Section 4: In this section, the questions concern the respondents' use of the languages as well as their level of satisfaction with it;
- Section 5: The closing section, in which we thank the participants and ask for their contact information, at their discretion.

The initial section, **section 1**, is where the respondent is introduced to the research study by describing the purpose of the survey. Because we must also account for the principles of research ethics (e.g. data confidentiality and informed consent), this text is clear that participation is voluntary and that the participant may withdraw from it at any given point [99]. Furthermore, we clarify that the participant's information will be kept confidential and that the results of the study will not contain any information that could potentially identify the respondents, and that those results will only be used for scholarly purposes. Finally in this text, we leave an email address that the participants may contact, in case they have any questions concerning the survey. Following the introductory text, we ask the respondent where they learnt about the survey (e.g. *LinkedIn*, *MATLAB Central*, *ResearchGate*, among others). These data are useful for later, when we profile the participants and analyse the descriptive statistics of the questionnaire.

Then, **section 2** begins the core segment of the questionnaire. In this section, we ask the participants questions concerning their programming background, habits and experience for the most part. This includes questions such as "How many years of programming experience do you have?" and "What programming language do you use the most?". This section is extremely important as it helps us stratify the community, which was one of the main goals of this study. This means that, with this section, we are able to structure and categorise the community in different segments, divided by their level of expertise, the application domain in which they program, the programming languages they use, among other factors.

Subsequently, in **section 3**, the questions focus on the importance and dedication given by the participants to the maintenance and reusability of their *MATLAB* programs. Thus, this section comprises questions concerning the scale of their programs as well as the participants' expectations on their programs. This section helps us understand what the typical level of focus on reusability and maintainability is, which was another one of the main goals. Together with section 2, we are already able to identify potential patterns just from these two sections.

Afterwards, in **section 4**, the last portion of the core segment of the questionnaire, we present questions concerning the participants' use of the languages, in addition to their level of satisfaction with the *MATLAB*'s capabilities. More specifically, we try to determine if and how the participants make use of *MATLAB*'s support to modularity, as well as what is their level of satisfaction with said support to modularity and with the programs' maintenance capabilities. Additionally, we ask the participants if they use object-oriented features in other languages, and which language they consider to be *MATLAB*'s strongest competitor concerning typical uses of the language. Therefore in this section we are able to get a sense of how the participants actually write their code, if they take advantage of *MATLAB*'s modularity capabilities and if they are satisfied with it. This allows us to potentially identify even more data patterns, in combination with the data collected from the second and third sections of the questionnaire, allowing for an elaborate and deep analysis.

Finally, **section 5** is the closing section, where we thank the respondents for their contribution. In addition, section 5 includes text fields where respondents may leave their email addresses in case they would like to receive the aggregated results of this study and in case they are willing to participate in a future questionnaire concerning this same study. This is, evidently, completely voluntary taking into account the common confidentiality and consent principles of research ethics [99], and it may help any future work done within the spectrum of this research.

Table 5.2 shows the structure of the questionnaire, containing 26 total questions distributed among 5 sections. Furthermore, appendix A contains the questionnaire itself, including the questionnaire's introductory text, questions, question descriptions and the response options to each question.

Table 5.2: Structure of the questionnaire.

Section	**Basis**	**Number of questions**
1	Introduction	1
2	Background, experience	13
3	Importance given to reusability and maintainability	5
4	Use of the language, satisfaction and opinion	7
5	Closing	-

5.5.2 Question types

Throughout its core segments (sections 2, 3 and 4), the questionnaire presents questions in various different types:

- 10 Likert scale questions;
- 6 drop-down questions;
- 4 checkbox questions;
- 3 multiple choice questions;
- 2 open ended questions.

The 5-point Likert scale questions present an ordinal scale ranging between two opposite extremes. These extremes are commonly "Strongly disagree" and "Strongly agree" as a way to measure the participants agreement with a provided statement, but may also be used to measure many different aspects such as satisfaction or the frequency of a habit. Not only are these questions effective at keeping the respondent engaged, but they also allow for an straightforward description and analysis of the results [29]. For this reason, this is the question type most often used in the questionnaire, as we tried to shape questions into this type whenever it made sense to do so.

Drop-down questions in which the respondent is presented a list of different response options with which they can provide an answer, and they may only pick one of those options. This list of response options is hidden until the respondent clicks on its corresponding interface element, granting a cleaner and more discrete presentation to the questionnaire and preventing the question from looking too big and complicated at first sight, and instead making it more approachable and the overall questionnaire less extensive.

Checkbox questions present a list of response options to the respondent, from which they may pick as many or as few as they wish. Additionally, it allows us to make one of those response options an open text field in which the respondent may insert a text containing his own response option. Although that obviously makes for a more difficult analysis, this type of question provides the participant with the necessary flexibility in their response.

Similarly to checkbox questions, multiple choice questions present a list of response options in which there may also be an open text field as a response option. However in this case, the respondent is only allowed to select one option from the list, meaning the options are mutually exclusive, which is necessary for some questions. Multiple choice questions are also similar to the drop-down questions, however these two are visually different to the respondent. In multiple choice questions, the list of options is never hidden, as opposed to a drop-down question in which the options are listed in a drop-down element and thus hidden until the respondent opens that element. Additionally,

multiple choice questions allow the respondent to write in an open text field to provide their own response option, whereas drop-down questions do not.

Open ended questions are questions where the respondent is given only a blank text field and they are to fill it with whatever they wish to say. This, unmistakably, gives a lot of freedom to the respondent. However, it makes for a more difficult data analysis. Although we tried to avoid this type of question as much as possible, it occasionally proved necessary.

5.5.3 Dividing question

The goals with this study included stratifying the community of users of *MATLAB* and its clone languages, and also understanding if these users take advantage of the languages' support to modularity. This is one of the reasons for why the questionnaire is divided into sections. The questions of each section vary in complexity and accessibility, with the questions in Section 2 being more focused on the respondent's background and programming experience and thus, more accessible. Any participant is able to provide valuable data in responding to the questions in this section of the questionnaire. On the other hand, in sections 3 and 4 the questions are more focused on the way users program and it refers to concepts that are not as easily understood by every participant in the questionnaire.

For this reason, at the end of section 2 is included a question through which we can apply branch logic to the questionnaire. Depending on the answer provided to this specific question the users are either redirected to the the more advanced sections of the questionnaire or are redirected straight to section 5, the final section of the questionnaire, if they are deemed unfit to answer the more advanced questions. This way, we are still able to make use of these participants' valid data with their responses to section 2 while avoiding unwanted data from them responding to questions whose topics they are not familiar with, and potentially compromising the results of the survey.

Thus, the last question of section 2 is the multiple-choice question: "I use only the command window when working with *MATLAB*.". The list of possible response options is:

- "True, I use only the command window.";
- "False, I use the command window to solve small problems or to complement my coding (e.g. inspecting variables, testing functions, etc.).";
- "False, I never use the command window.".

With this question, we can separate the respondents who only use *MATLAB* through the command window (the users we deemed to be more unfamiliar and inexperienced) from the ones who write *MATLAB* programs using m-files and either never use the command window, or use it only to complement their coding. The former do not advance to section 3 of the questionnaire, and instead are redirected to section 5. The latter follow the standard path, where they are taken to the following section, section 3.

5.5.4 Consistency-measuring questions

Respondent consistency is something we strive to ensure and maintain as much as possible, because inconsistent responses could be a severe threat to the validity of the research results. Therefore, to ensure that we are capable of distinguishing and identifying inconsistent respondents, the questionnaire includes a consistency test. In other words, the questionnaire includes two questions that, in fact, measure the same factor. These questions are not contiguous, but instead they are in two different sections in order to not influence, in any way, the respondent's behaviour (see Subsection 2.1.3).

If a participant's responses are not consistent throughout the questionnaire, this could indicate that none of their responses are trustworthy. Whether this is due to them not being qualified to answer to those sections of the questionnaire or due to them being distracted, it could represent irrelevant or misleading results to the survey. Consequently, the consistency test in the questionnaire is comprised of these two questions:

- "When I develop a program in *MATLAB* I always try to make it easily reusable and maintainable.";
- "I try to find and minimise the use of duplicated code across the various m-files.".

Although they are worded differently, these two questions are measuring the same factors and should, thus, be interpreted in a similar way to one another. Both of these questions measure the dedication and effort given by the respondent to the reusability and maintainability of their programs. Additionally, they are both of a 5-point Likert scale type, ranging from "Strongly disagree" to "Strongly agree". The first question is introduced in the start of Section 2, and the second question in Section 3. During the result analysis stage of this research, we measure the difference between the data collected from each of these two questions. This, in turn, allows us to discard the more inconsistent data points or, in other words, the responses of the more inconsistent respondents (see Section 6.2).

5.6 Threats to Validity

For the threats to validity of this survey we follow guidelines based on Cook and Campbell [99] (see Subsection 2.1.5), and they are presented below:

5.6.1 Conclusion Validity

Low statistical power. Due to a limited sample size, the validity of some more intricate inferences is affected. For instance, 93.87% of the respondents have stated that they use *MATLAB*, as opposed to all 100% of respondents. This means that any inference related only to *MATLAB* has its validity affected.

Fishing. Searching or *fishing* for a specific outcome or conclusion is a threat to the validity of the inferences and conclusions drawn from this study. Interpretations from a single or few individuals are always subject to a potential bias, even if only slightly, and that is the case with this study. There may be different interpretations to be drawn from the results obtained that are not thought of or presented here.

Reliability of measures. Due to a less than ideal question wording or instrument layout, there is possibly a lack of reliability in the measures used. This threatens the validity of the results of the study, as the measures should be able to be repeated with the same outcome.

5.6.2 Internal Validity

Instrumentation. If the layout of the different sections is illogical, or if some questions are perceived as ambiguous by the participants, the validity of the experiment is threatened.

5.6.3 Construct Validity

Inadequate preoperational explication of constructs. The study's constructs may not be properly and clearly defined before they were translated into measures in the construction of the questionnaire. This means that the theory and intention behind the study may be incoherent, and the validity of the results may be affected.

Hypothesis guessing. If the respondents try to guess an intended result or a hidden expectation of the questionnaire, then this could influence their responses and they could lean towards or against these guesses. This also threatens the validity of the instrument.

Evaluation apprehension. It is a human tendency to try to look better when being evaluated. And during the questionnaire, the respondents may have felt like they were being evaluated and thus, they may have provided false data in order to seem better. For example, they could have said that they have more experience than they actually have. This undermines the validity of the experiment.

5.6.4 External Validity

Interaction of selection and treatment. If the subject population is different than the population that was initially planned to generalise the results to, the validity of the results is threatened. For instance, programmers that have never dealt with *MATLAB* or any similar language may have decided to respond to the questionnaire. With the sample size obtained, we consider the results to be safe to generalise to the whole target population. However, some of the results and inferences obtained are only extendable to a portion of the population (i.e. the participants who use *MATLAB* beyond just the command window.

5.7 Conclusion

This Chapter describes the identification of the target population as well as the sampling strategy to reach said population. For this research study, the target population is the users of *MATLAB* and its clone languages. This population is reached through posts on online forums such as the *MATLAB Central* forums or *Reddit*, as well as via e-mail. Community feedback was also gathered at an early stage in order to help us better understand how each forum works and the type of response we could expect from each one.

Furthermore, for the construction and administration of the questionnaire, we use *Google Forms* in this study, as it showed to be the best option following the analysis and comparison of many of the different tools available for this task. In addition, this Chapter provides a detailed description of the questionnaire specification that was built to help organise the data and notes for this study.

This Chapter also incorporates a comprehensive overview of the construction and design of the questionnaire, including an analysis of the questionnaire's structure, of the types of question, of the branch logic used in the form of a dividing question, and the questions designed specifically to measure the respondents' consistency.

Finally, this Chapter also provides a rundown on the threats to the validity of this research study and its results.

6

Survey Execution and Data Analysis

In this Chapter we cover the administration of the questionnaire (Section 6.1) and the verification of the internal consistency of the responses obtained (Section 6.2). Then, we analyse the profile of the participants (Section 6.3). Following that, we present the hypothesis formulation (Section 6.4) and testing (Section 6.5). Finally, we analyse the results and implications of the survey (Section 6.6).

6.1 Administering the questionnaire

The execution stage of the survey starts with administering the questionnaire to only a portion of the initial target population in order to evaluate different metrics, such as the number and frequency of responses, logistics such as the interaction in the selected forum and the comments section of the post. This method allows for an improvement of not just the questionnaire by acting as FG (see Subsection 2.1.5), but also of the approach and strategy of the publications. Through the analysis and consideration of said feedback given directly by the respondents themselves, this improvement can be made before administering the questionnaire at full scale.

As we had previously analysed the target communities (see Section 5.1), the only necessary step to start administering the questionnaire would be to pick a community out of all the ones analysed where it would make the most sense to publish the questionnaire first. Therefore, we chose to start with *Reddit*, specifically with the *MATLAB* "subreddit", as it seemed to be the most populous and interactive community, which we expected to provide us with the largest amount of feedback [63].

Immediately after publishing the questionnaire for the first time on October 22nd, 2020, we started getting feedback via the comments. For instance, a user warned us about the fact that one of the open text fields in section 5 of the questionnaire (in which we invite respondents to leave their email address) was marked as a required field. This was, evidently, a mistake that we were able to quickly fix, thanks to the quick feedback. Additionally, we had users commenting that one of the multiple-choice questions was somewhat confusing and did not have enough options to cover all possible scenarios. This

led us to change the wording slightly on the response options, as well as to add one more option to cover a wider variety of scenarios and to add a description to the question in order to better clarify what was meant.

By the time we neared a number of 100 responses, the frequency of the responses as well as the interaction in the comments section was starting to cool down, and we felt satisfied with this first publication of the questionnaire. It allowed us to improve the instrument and minimise risks before sharing it with a larger audience. For all purposes, the responses from this first publication are valued as equivalent to the responses from the following publications, as the resulting changes to the questionnaire did not have enough impact to invalidate any of the results or conclusions drawn in the slightest.

Afterwards, it was time to administer the questionnaire to a wider audience on November 2nd, 2020. Thus, we reached out to the rest of the communities we had previously analysed (see Section 5.1). We published the questionnaire via a post on the *MATLAB Central* forums [51], on *ResearchGate* [74], and on 4 different *LinkedIn* groups ("MATLAB Users and Integrators" [46], "Matlab beginners and experts" [43], "Scilab Software" [48] and "GNU Octave users and developers" [41]).

Much like *Reddit*, these platforms also allow users to comment on a post which allows for a discussion on the topic of the post and in the context of this thesis, allows the respondents to provide further feedback on the questionnaire itself. They are however, and as we expected, not as active and responsive as *Reddit*.

In the case of *MATLAB Central* and *ResearchGate*, we replicated this same post multiple times because these platforms have a high frequency of posts, and the posts would fade out quicker than on the other platforms, and users of these platforms could more easily miss it. This turned out to increase the number of responses from these sources with each post, therefore it showed to have a significant, positive impact in increasing the diversity of the participants. In total, we posted the questionnaire two times on *ResearchGate* and three times on *MATLAB Central*. Participants from *ResearchGate* showed to have some interaction to provide some feedback in the comments section, whereas participants from *MATLAB Central* had nearly no interaction in the comments section, despite the similar number of respondents from each of these two platforms (see Section 6.3).

On the other hand, on *LinkedIn* we only posted once in each of the groups mentioned. Even though these groups have a high number of users, there are not as many posts as in the other platforms and thus we did not feel that more than 1 post was necessary. Participants from this platform nonetheless still accounted for a significant portion of the respondents (see Section 6.3).

Because the questionnaire was constructed and hosted on *Google Forms*, data from the responses is automatically collected to a *Google Sheets* spreadsheet. Using this platform, we could quickly monitor the data by looking at every response to every question while the questionnaire was still accepting replies. By doing this, we started to notice a predominance of *MATLAB* users, as opposed to users of *MATLAB*'s clone languages. Therefore,

we decided to also post the questionnaire on a *GNU Octave Discourse* forum [11].

Discourse [10] is an open-source forum software, and on it part of the *GNU Octave* community hosts a forum where users can interact with each other and discuss everything *Octave*. In this platform we did not receive a lot of feedback or interaction via the comments section, although we were still capable of gathering responses (see Section 6.3).

The total number of responses was 215. In Table 6.1 we can observe the number of participants from each source as well as the respective approximate percentage values. The total number of participants in this Table is 212 as opposed to 215 because the responses of 3 participants who showed to be inconsistent were discarded during the process of verification of internal consistency (see Section 6.2).

Table 6.1: Source of the responses - *Where did you hear about this survey?*

Source	Number of participants	Percentage
Reddit	101	48%
LinkedIn	56	26%
ResearchGate	17	8%
MATLAB Central	13	6%
GNU Octave Discourse	9	4%
E-mail	10	5%
Word of mouth	6	3%
Total	212	100%

6.2 Verifying the Internal Consistency

The first step in the analysis of the data is to verify the internal consistency of the responses. A total of 215 responses were gathered, with 188 of those being complete responses and the remaining 27 being partial responses. During this process we analyse the continuous variables (i.e. interval variables) and the ordinal variables (i.e. Likert scales), as the others are not easily quantifiable. The goal is to reduce the set of variables into a smaller set of "artificial" variables (also known as the principal components) that account for most of the variance of the original set of variables [87]. But before we are able to do that, we analyse the consistency of the respondents and depending on the results consider discarding any participants' responses. Through this method, we can detect false or random responses, making us more confident that the answers that we do not discard are real and valuable data. To do this, we use the Kendall rank correlation coefficient.

6.2.1 Consistency test

In the questionnaire, we included a pair of questions that were very similar in order to be able to measure the consistency of the respondents based on the answers to these

two questions (questions 15 and 21, see appendix A). Kendall's tau distance, also called *bubble-sort distance*, is a metric with which one can calculate the number of and the degree of disagreement between two lists [34]. This involves pairwise comparison of all data points. In the context of this questionnaire, it is used to measure the discrepancy in the answers given to these two similar questions. In Figure 6.1 we can observe the frequency of each distance value in the responses.

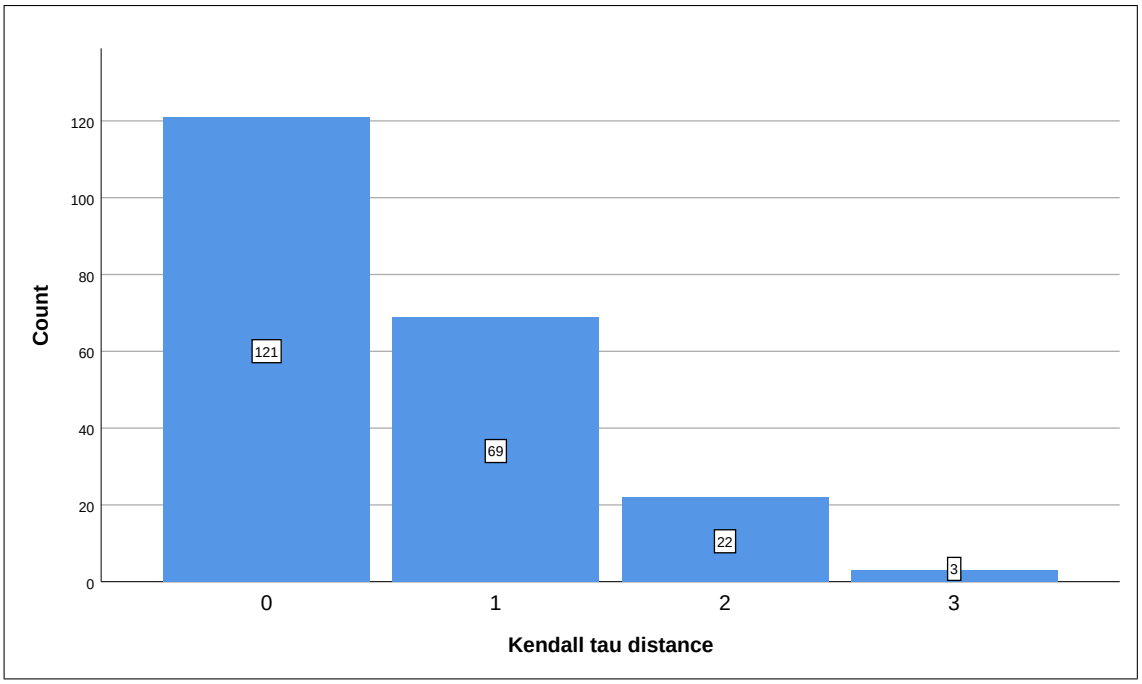

Figure 6.1: Kendall tau distance between answers to questions 15 and 21.

Both questions are presented in the form of a statement, where the response values are in a Likert scale format, ranging from "1" (Strongly disagree) to "5" (Strongly agree). A Kendall's tau distance of 0 means a participant answered the same to both questions. In other words, a distance of 0 represents maximum consistency. In contrast, a distance of 4 is the highest possible, and it means a participant answered "Strongly disagree" to one of the questions, and "Strongly agree" to the other, and therefore he was inconsistent. Thus, from the graph we can conclude that 121 participants demonstrated a distance of 0, 69 participants demonstrated a distance of 1, 22 participants demonstrated a distance of 2 and 3 participants demonstrated a distance of 3. The participants that demonstrate a distance of 3 are the most inconsistent of the data set, as no participant demonstrated a distance of 4. The mean distance recorded is, approximately, 0.57.

Additionally, we can calculate the Kendall's tau-b correlation coefficient [86]. This is a nonparametric measure of strength and direction of an association between two variables. In this case, a Kendall's tau-b correlation was run to determine the relationship between the answers to question 15 and the answers to question 21 amongst the 188 participants.

However, before we can analyse the data using Kendall's tau-b we must first make

sure that the data "passes" the three assumptions required to obtain a valid result:

1. The two variables must be measured on at least an ordinal scale. Both variables are obtained from responses to a 5-point Likert scale question, so they fit this requirement.

2. The two variables must represent paired observations. This is true in this context, as the number of paired observations is equal to the number of participants that answered the two questions associated with these variables.

3. There must be a monotonic relationship between the two variables.

Kendall's tau-b bases its analysis on **concordant** and **discordant** pairs, assessing the degree to which there is a monotonic relationship. So every pair of observations (e.g., participants) can be:

- **Concordant:** When the value of both variables is larger or smaller for one participant than the other participant;

- **Discordant:** When the value of one of the two variables is larger in one observation compared to the other observation, but the other variable is smaller;

- **Tied:** When the value of one or both of the two variables is the same across both observations.

All of these three cases are taken into account, as expressed in the equation below [38]:

$$Kendall's\tau_b = \frac{C - D}{\sqrt{(C + D + T_X) \times (C + D + T_Y)}}$$

In the equation, **C** represents the number of concordant pairs and **D** represents the number of discordant pairs based on all distinct pairs of participants. $\mathbf{T_X}$ represents the number of pairs with a tie only on the first variable and $\mathbf{T_Y}$ represents the number of pairs with a tie only on the second variable. Kendall's tau-b ranges from -1 to +1. A -1 coefficient indicates a perfect negative relationship where all pairs of observations are **discordant**, whereas a +1 coefficient indicates a perfect positive relationship where all the pairs of observations are **concordant**.

From the correlation test results we are able to observe in Table 6.2, there was a strong, positive correlation between both questions' answers ($\tau_b = 0.450$, $p = 0$) [97].

Based on the mean Kendall's tau distance (0.57) and Kendall's tau-b correlation coefficient observed, and since there are not *rules of thumb* or concrete guidelines to follow for this procedure in this specific case, we have decided to discard the answers from the 3 participants whose Kendall's tau distance had a value of 3. This is done to preserve the integrity of the results and to facilitate further data analysis.

Table 6.2: Kendall's tau-b correlation coefficient.

			Question 15	Question 21
Kendall's tau_b	**Question 15**	**Correlation Coefficient**	1.000	.450**
		Sig. (2-tailed)	.	.000
		N	188	188
	Question 21	**Correlation Coefficient**	.450**	1.000
		Sig. (2-tailed)	.000	.
		N	188	188
** Correlation is significant at the 0.01 level (2-tailed).				

6.2.2 Principal component analysis

PCA is a variable-reduction technique that allows us to reduce the larger set of variables into a smaller set of "artificial variables" that account for most of the variance of the original variables. But before we can analyse the data using PCA, we must first make sure the data can be analysed with this technique. PCA is appropriate only if the data "passes" the four assumptions required for it to provide a valid result [87]:

1. The variables should be continuous or ordinal (i.e., can be a 5-point Likert scale);
2. There should be a linear relationship between the variables. In practice, this assumption is relaxed with the use of ordinal data for variables;
3. There should be sampling adequacy, meaning a large enough sample size to produce reliable results;
4. There should not be significant outliers that could have a disproportionate influence on the results.

The first relevant part of the analysis is the **Correlation Matrix** Table (Table 6.3), where we can observe the correlation values between all the variables in the PCA and thus test the linearity between all variables. We must examine this Table in search of any variables that are not strongly correlated with any other variable. The level of correlation is considered strong if $r \geq 0.3$.

Table 6.3: Correlation matrix.

	Q2	Q4	Q5	Q7	Q9	Q12	Q15	Q16	Q17	Q18	Q19	Q21	Q22	Q23	Q24
Q2	**1.000**	0.294	0.307	0.443	0.445	-0.014	0.235	-0.312	0.267	0.274	0.087	0.235	0.269	-0.012	0.224
Q4	0.294	**1.000**	0.880	0.557	0.551	0.073	0.083	-0.241	0.431	0.294	0.030	0.127	0.207	-0.059	0.165
Q5	0.307	0.880	**1.000**	0.635	0.534	-0.097	0.081	-0.277	0.486	0.333	-0.012	0.128	0.263	-0.014	0.211
Q7	0.443	0.557	0.635	**1.000**	0.774	-0.110	0.289	-0.287	0.565	0.375	0.087	0.293	0.416	0.155	0.390
Q9	0.445	0.551	0.534	0.774	**1.000**	0.004	0.203	-0.265	0.406	0.305	0.086	0.211	0.262	0.155	0.382
Q12	-0.014	0.073	-0.097	-0.110	0.004	**1.000**	-0.011	0.082	-0.019	-0.097	0.020	-0.051	-0.068	-0.045	-0.184
Q15	0.235	0.083	0.081	0.289	0.203	-0.011	**1.000**	-0.276	0.247	0.242	0.095	0.550	0.510	0.219	0.295
Q16	-0.312	-0.241	-0.277	-0.287	-0.265	0.082	-0.276	**1.000**	-0.319	-0.399	-0.085	-0.278	-0.299	0.005	-0.159
Q17	0.267	0.431	0.486	0.565	0.406	-0.019	0.247	-0.319	**1.000**	0.417	0.179	0.302	0.383	0.033	0.240
Q18	0.274	0.294	0.333	0.375	0.305	-0.097	0.242	-0.399	0.417	**1.000**	0.190	0.326	0.372	0.050	0.082
Q19	0.087	0.030	-0.012	0.087	0.086	0.020	0.095	-0.085	0.179	0.190	**1.000**	0.073	0.064	-0.061	-0.065
Q21	0.235	0.127	0.128	0.293	0.211	-0.051	0.550	-0.278	0.302	0.326	0.073	**1.000**	0.614	0.275	0.303
Q22	0.269	0.207	0.263	0.416	0.262	-0.068	0.510	-0.299	0.383	0.372	0.064	0.614	**1.000**	0.285	0.320
Q23	-0.012	-0.059	-0.014	0.155	0.155	-0.045	0.219	0.005	0.033	0.050	-0.061	0.275	0.285	**1.000**	0.374
Q24	0.224	0.165	0.211	0.390	0.382	-0.184	0.295	-0.159	0.240	0.082	-0.065	0.303	0.320	0.374	**1.000**

Thus, from the Table we can see that Q12 and Q19 do not have a strong correlation with any of the other variables. Therefore we are, for now, keeping an eye on these two variables in the upcoming steps of this analysis. They correspond to questions 12 ("The last time I programmed in *MATLAB* or a similar language was...") and 19 ("The m-files I deal with tend to have...") (see Appendix A).

Next, we can test the sampling adequacy of the data through three different methods:

- The KMO measure of sampling adequacy for the overall data set;
- The KMO measure for each individual variable;
- Bartlett's Test of Sphericity.

Using the KMO measure we can detect if there are linear relationships between the variables and therefore decide if it is appropriate to execute a PCA. It can range from 0 to 1, with values above 0.6 being the minimum required for sampling adequacy. Table 6.4 contains the KMO measure for the overall data set: 0.795, which is a "middling" value meaning that it approves the overall data set for a PCA [32].

Table 6.4: KMO and Bartlett's Test

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		0.795
Bartlett's Test of Sphericity	**Approx. Chi-Square**	1122.356
	df	105
	Sig.	0

The KMO measures for the individual variables can be found on Table 6.5, which is the Anti-image Correlation Table. These are the values in the main diagonal whose cells are highlighted in gray background. Once again, we want the KMO values to be as close to 1 as possible, with 0.5 being the minimum value accepted for us to involve that variable in the PCA. In this case, we can observe that Q12 has a KMO measure of 0.258.

Table 6.5: Anti-image Correlation matrix.

	Q2	Q4	Q5	Q7	Q9	Q12	Q15	Q16	Q17	Q18	Q19	Q21	Q22	Q23	Q24
Q2	.916[a]	0.017	-0.016	-0.104	-0.165	-0.040	-0.049	0.142	0.056	-0.053	-0.026	-0.044	-0.046	0.144	-0.060
Q4	0.017	.684[a]	-0.829	0.133	-0.240	-0.301	-0.027	-0.028	0.014	-0.005	-0.062	-0.080	0.046	0.115	0.000
Q5	-0.016	-0.829	.697[a]	-0.278	0.138	0.277	0.092	0.069	-0.120	-0.042	0.131	0.103	-0.073	-0.036	0.011
Q7	-0.104	0.133	-0.278	.814[a]	-0.604	0.094	-0.093	-0.081	-0.268	-0.019	-0.011	0.022	-0.144	-0.018	-0.044
Q9	-0.165	-0.240	0.138	-0.604	.789[a]	-0.095	0.041	0.065	0.089	-0.049	-0.046	0.010	0.124	-0.099	-0.166
Q12	-0.040	-0.301	0.277	0.094	-0.095	.258[a]	-0.049	-0.045	-0.104	0.089	0.025	0.031	-0.027	-0.051	0.188
Q15	-0.049	-0.027	0.092	-0.093	0.041	-0.049	.858[a]	0.114	0.003	0.004	-0.050	-0.300	-0.202	-0.025	-0.099
Q16	0.142	-0.028	0.069	-0.081	0.065	-0.045	0.114	.884[a]	0.088	0.225	-0.006	0.046	0.043	-0.085	0.029
Q17	0.056	0.014	-0.120	-0.268	0.089	-0.104	0.003	0.088	.896[a]	-0.163	-0.147	-0.068	-0.077	0.089	-0.085
Q18	-0.053	-0.005	-0.042	-0.019	-0.049	0.089	0.004	0.225	-0.163	.884[a]	-0.122	-0.102	-0.116	-0.027	0.157
Q19	-0.026	-0.062	0.131	-0.011	-0.046	0.025	-0.050	-0.006	-0.147	-0.122	.623[a]	0.008	0.012	0.044	0.100
Q21	-0.044	-0.080	0.103	0.022	0.010	0.031	-0.300	0.046	-0.068	-0.102	0.008	.826[a]	-0.375	-0.106	-0.064
Q22	-0.046	0.046	-0.073	-0.144	0.124	-0.027	-0.202	0.043	-0.077	-0.116	0.012	-0.375	.857[a]	-0.138	-0.050
Q23	0.144	0.115	-0.036	-0.018	-0.099	-0.051	-0.025	-0.085	0.089	-0.027	0.044	-0.106	-0.138	.699[a]	-0.280
Q24	-0.060	0.000	0.011	-0.044	-0.166	0.188	-0.099	0.029	-0.085	0.157	0.100	-0.064	-0.050	-0.280	.825[a]

a. Measures of Sampling Adequacy(MSA)

Earlier in Subsection 6.2.2, we were already alerted to the variable Q12 due to the values it obtained in the correlation matrix (Table 6.3). Because this variable also demonstrates a relatively low KMO measure (Table 6.5), we have decided to exclude this variable for the remainder of this PCA phase. Consequently we are able to execute a new KMO and Bartlett's Test and analyse the new results after excluding Q12 from the analysis, which we can observe in Table 6.6.

Table 6.6: Second KMO and Bartlett's Test

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		0.809
Bartlett's Test of Sphericity	**Approx. Chi-Square**	1091.228
	df	91
	Sig.	0

From Table 6.6, we can see that the overall KMO measure has increased as expected and it is now 0.809.

Additionally, we can now analyse the results of the Bartlett's Test of Sphericity and we can observe that the test is statistically significant. This is demonstrated by the "Sig." row which states a value of ".000", thus a $p < 0.0005$. This further indicates that a PCA may be useful in this case, as the data is likely "factorisable".

The **communality** is the proportion of each variable's variance that is preserved and accounted for in the PCA, and it can be expressed as a percentage. We can observe the data's communalities in Table 6.7.

Table 6.7: Communalities Table.

	Initial	**Extraction**
Q2	1	0.327
Q4	1	0.761
Q5	1	0.799
Q7	1	0.763
Q9	1	0.674
Q15	1	0.588
Q16	1	0.405
Q17	1	0.515
Q18	1	0.529
Q19	1	0.34
Q21	1	0.662
Q22	1	0.641
Q23	1	0.571
Q24	1	0.591

A PCA produces as many components as variables. For that reason, the initial communalities (displayed in the "Initial" column) are all of value 1, as these are taking into account all the components produced. However, the goal is to retain only some of the components. Therefore, the communalities reported in the "Extraction" column are the proportion of each variable's variance that is preserved when only the components being

retained are taken into account. As expected, these communalities have values of less than 1.

Because we are analysing 14 variables, we are presented with 14 components. If we were to hold on to all 14 components we would be able to account for all the variance in the data set. That is not the goal of this analysis, however. The goal is to retain as much of the variance as possible using as few components as possible. The first component is the one that retains the most amount of variance, and each subsequent component preserves less and less variance than the previous one. Therefore, we only need to hold on to the first few components as those account for the majority of the variance.

An **eigenvalue** is a measure of the variance that is retained by a component. In Table 6.8, we can observe the 14 components ordered by their eigenvalues. Furthermore, the table includes the percentage of variance each component is accounting for and the cumulative values of those percentages through the addition of each component.

Table 6.8: Total variance retained.

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	4.853	34.661	34.661	4.853	34.661	34.661	3.718	26.559	26.559
2	1.954	13.96	48.62	1.954	13.96	48.62	2.676	19.117	45.676
3	1.359	9.71	58.33	1.359	9.71	58.33	1.772	12.654	58.33
4	0.939	6.709	65.04						
5	0.877	6.268	71.307						
6	0.736	5.255	76.562						
7	0.63	4.502	81.064						
8	0.582	4.155	85.22						
9	0.496	3.546	88.765						
10	0.472	3.373	92.139						
11	0.425	3.039	95.177						
12	0.379	2.71	97.887						
13	0.201	1.437	99.324						
14	0.095	0.676	100						

With a total of 14 variables, the 14 components account for a total of 25 eigenvalues of variance. Additionally, if we observe the first component we find that it retains 4.853 eigenvalues of variance, which is 34.661% of the total variance as reported in the "% of Variance columns".

The Kaiser criterion, also known as the eigenvalue-one criterion, is a popular method to establish how many components we should retain in the PCA [33]. This method states that an eigenvalue of less than 1 indicates that the component retains less variance than a variable would and therefore should not be retained. This means that we want retain components 1, 2 and 3 and that all the components 4 and above are discarded. We deem this to be an effective method for this specific case, as the deciding components 3 and 4 are fairly distant in eigenvalue and they are also both distant from the eigenvalue threshold of 1. If components 3 and 4 had respective eigenvalues of, for example, 1.02 and 0.98, we would have to take a different approach as these values would be quite close to the threshold. But it is not the case with these results and therefore we opt to keep the first 3 components as the Kaiser criterion suggests.

In addition, a visual inspection of the scree plot generated further indicates that three

components should be retained, as evidenced by the fact that the fourth component is distinctively the inflection point (see Figure 6.2) [5]. The inflection point is represented by the point where the graph begins to level out and where subsequent points add little to the total variance.

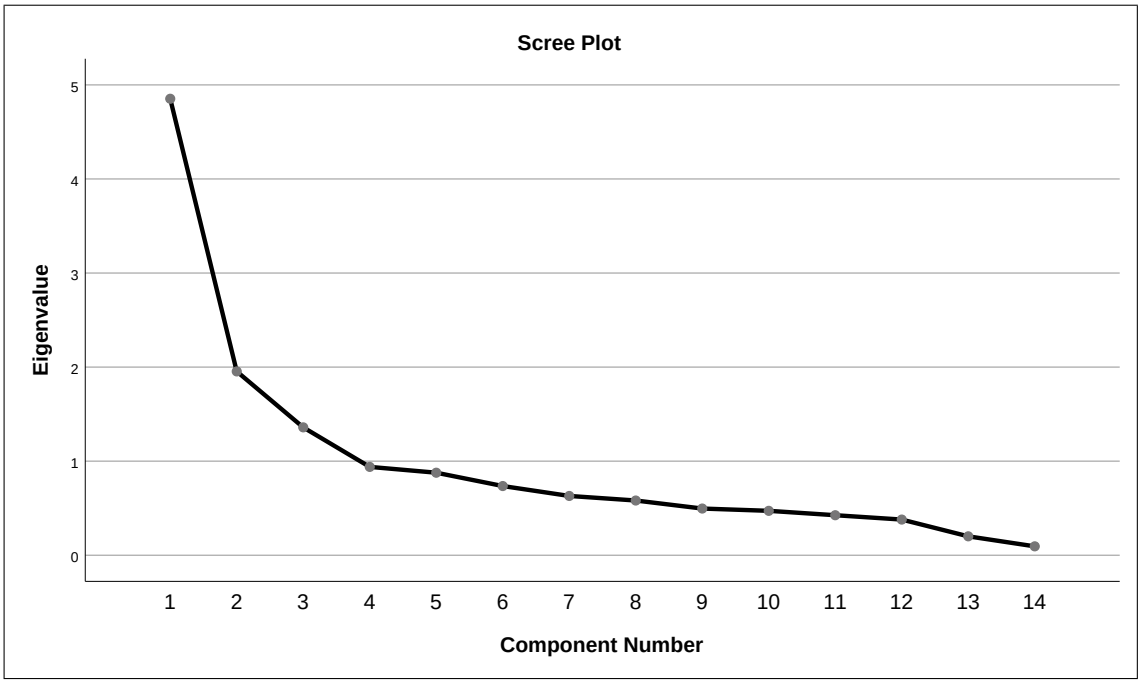

Figure 6.2: Scree plot.

Therefore, we can now analyse the Rotated Component Matrix in Table 6.9, by employing a Varimax orthogonal rotation to make it easier to interpret. In this Table, which is sorted by coefficient size, each variable has one component that loads strongly on it.

In summary, a PCA was run on 15 variables, each one corresponding to one of the questions in the questionnaire which fit the required criteria (continuous and ordinal variables). Inspection of the correlation matrix revealed that all variables had at least one correlation coefficient greater than 0.3, with the exceptions of Q12 and Q19. The overall KMO measure was 0.795. However, the individual KMO measure for Q12 showed to be insufficient, and therefore we removed this variable from the analysis in order to avoid misleading results. We then obtained a new overall KMO measure of 0.809, with individual KMO measures all greater than 0.5. Furthermore, Barlett's test of sphericity was statistically significant, indicating that the data was "factorisable".

The PCA revealed three components that had eigenvalues greater than one and which accounted for 34.661%, 13.96% and 9.71% of the total variance, respectively. Visual inspection of the scree plot also indicated that three components should be retained [5]. This three-component solution retains approximately 58.330% of the total variance of the data set.

In the questionnaire, questions 2 to 14 were part of Section 2 which mainly assessed

Table 6.9: Rotated component matrix and communalities. Coefficients higher than 0.3 are highlighted in bold and with a dark coloured cell.

	Component			
	1	2	3	Communalities
Q5	**0.889**	-0.04	0.082	0.327
Q4	**0.863**	-0.089	0.091	0.761
Q7	**0.811**	**0.308**	0.105	0.799
Q9	**0.788**	0.23	0.004	0.763
Q17	**0.564**	0.212	**0.39**	0.674
Q2	**0.462**	0.202	0.27	0.588
Q21	0.079	**0.737**	**0.336**	0.405
Q22	0.222	**0.7**	**0.319**	0.515
Q15	0.045	**0.695**	**0.32**	0.529
Q23	0.008	**0.664**	**-0.361**	0.34
Q24	**0.349**	**0.617**	-0.298	0.662
Q18	**0.335**	0.209	**0.611**	0.641
Q19	-0.033	-0.055	**0.58**	0.571
Q16	-0.284	-0.22	**-0.525**	0.591

the participant's background and general programming experience. These are what Component 1 corresponds to, for the most part, in the matrix obtained. Similarly, questions 15 to 21 in the questionnaire were meaning to assess the level of proficiency of the participants with *MATLAB*, and how these participants interact with this programming environment. This is once again demonstrated in the component matrix obtained in the form of Component 3 - these questions are more heavily loaded on Component 3. Lastly, the goal with questions 22 to 26 was to assess the participants' familiarity with *MATLAB*'s current support to modularity, and what their opinion of it is. As expected, these questions have a stronger influence on Component 2.

Therefore, the results obtained are consistent with the intended designed and structure of the questionnaire, as a clear distinction between the components is reflected in their respective questions.

6.2.3 Cronbach's Alpha

CA is a measure of reliability and internal consistency that is used in conjunction with a data reduction technique like PCA. It is used to deduce how much the variables on a (continuous or ordinal) scale (or PCA component, in this survey's context) are measuring the same underlying construct or dimension. Since the variables are on an ordinal scale, we can use this technique to calculate their consistency. But because CA determines how well a set of questions are grouped together, we must run multiple CA tests, one for each of the components resultant from the PCA.

In Table 6.10 we can see, in the "Cronbach's Alpha" column, that $\alpha = 0.835$ for this

Table 6.10: Component 1 - Cronbach's Alpha.

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.835	.843	8

scale, which includes every question that had a heavy load (coefficient of ≥ 0.3) in Component 1. This resulted in a total of 8 items (questions) being included in the scale, as demonstrated in the "N of items" column.

A CA of 0.835 indicates a high level of internal consistency for this scale, as any value of 0.7 or higher is recommended [36, 8]. We have to ensure that these are valid results. One way to do this is to observe the Item-Total Statistics Table obtained, particularly the Corrected Item-Total Correlation (Table 6.11).

Table 6.11: Component 1 - Item-Total Statistics.

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	**Corrected Item-Total Correlation**	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
Q4	23.69	36.532	**.664**	.794	.802
Q5	23.98	35.472	**.716**	.814	.795
Q7	23.31	36.884	**.782**	.727	.791
Q9	23.17	39.350	**.693**	.655	.806
Q17	23.68	34.998	**.582**	.396	.816
Q2	23.77	40.390	**.440**	.237	.831
Q24	23.27	43.760	**.317**	.189	.842
Q18	24.17	38.460	**.426**	.228	.838

The "Correlated Item-Total Correlation" is the Pearson correlation between the specific question and the sum of all the other questions in the scale [4]. Therefore, if all the items in this scale were perfectly measuring the same underlying construct, we would expect a high correlation coefficient here. In this case, we encounter no items with a coefficient of ≤ 0.3 so there is not any alarming concern in deeming this CA analysis valid for Component 1.

Table 6.12: Component 2 - Cronbach's Alpha.

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.762	.762	6

Similarly, in Table 6.12 we can observe that the CA for the second component is $\alpha = 0.762$ which is indicative of a high level of consistency since it is greater than the threshold of 0.7. Because of Table 6.13 we can deem this to be a valid result because, just like with Component 1, there are not any items that demonstrate a correlation coefficient lower than 0.3 in the "Correlated Item-Total Correlation" column.

Table 6.13: Component 2 - Item-Total Statistics.

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	**Corrected Item-Total Correlation**	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
Q7	19.32	12.534	**.437**	.250	.747
Q21	19.01	12.185	**.589**	.454	.706
Q22	19.15	11.114	**.635**	.476	.689
Q15	18.96	12.689	**.546**	.359	.718
Q23	19.37	13.755	**.354**	.187	.764
Q24	19.30	12.919	**.478**	.282	.734

Table 6.14: Component 3 - Cronbach's Alpha.

Cronbach's Alpha	**Cronbach's Alpha Based on Standardized Items**	**N of Items**
.496	.543	8

Table 6.15: Component 3 - Item-Total Statistics.

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	**Corrected Item-Total Correlation**	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
Q17	23.55	13.857	**.366**	.261	.391
Q21	22.86	15.698	**.536**	.462	.361
Q15	22.81	16.611	**.442**	.363	.398
Q16	23.94	24.670	**-.418**	.224	.695
Q18	24.04	14.580	**.348**	.297	.403
Q19	24.69	18.673	**.139**	.059	.490
Q22	22.99	14.625	**.569**	.483	.327
Q23	23.22	18.355	**.191**	.114	.475

Finally, in Table 6.14 we can see that, at first, Component 3's scale seems to show a low level of internal consistency with a CA value of only 0.496. However, we must also take a look at the Item-Total Statistics Table (Table 6.15). In this Table we can notice that there is a negative value in the "Correlated Item-Total Correlation" column, corresponding to question 16 ("*I expect to be the sole user of my MATLAB programs.*"). This was expected, however, as this particular question could be distinguished by its wording and format, when compared to the rest of the questions on this scale. This means that we have to "reverse recode" this specific variable and re-run this CA test with the new reversed variable instead. "Reverse recoding" is, in other words, flipping the response values. For instance, a response of "Strongly disagree" would have a numerical value of 1, but after reverse coding this variable that response would turn into a 5. Likewise, a response of "Strongly agree" that would have a numerical value of 5 would now have a value of 1, and the same thing would happen for each of the 5 different possible responses to this question (with the exception of the responses of numerical value 3, which remain the same value before and after the "reverse recoding").

After "reverse recoding" question 16, we can observe the new CA value (in Table 6.16) of 0.723 for Component 3, which this time indicates a strong level of consistency

Table 6.16: Component 3 - Cronbach's Alpha - after "reverse recoding".

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.723	.725	8

Table 6.17: Component 3 - Item-Total Statistics - after "reverse recoding".

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	**Corrected Item-Total Correlation**	Squared Multiple Correlation	Cronbach's Alpha if Item Deleted
Q17	23.6811	22.305	**.474**	.261	.685
Q21	22.9892	25.500	**.568**	.462	.671
Q15	22.9405	26.448	**.499**	.363	.684
Q16R	23.9351	24.670	**.418**	.224	.695
Q18	24.1676	22.619	**.505**	.297	.675
Q19	24.8162	29.542	**.148**	.059	.739
Q22	23.1243	24.044	**.611**	.483	.656
Q23	23.3459	29.608	**.153**	.114	.738
Q16R - Q16 after "reverse recoding"					

in this scale. In Table 6.17 we can notice that two items have a correlation coefficient of ≤ 0.3, so this scale is not as strong as previous two scales corresponding to Components 1 and 2, but that is to be expected because, as we have seen in Subsection 6.2.2, the first component retains the most amount of variance and each subsequent component preserves less variance than the previous one.

To conclude, in Table 6.18 we can observe a summary of each component and their CA values.

Table 6.18: Summary of PCA and CA results.

Component	Predominant questions (≥ 0.3)	Eigenvalue	% of Variance	Cronbach's Alpha
1	Q2, Q4, Q5, Q7, Q9, Q17, Q18, Q24	4.853	34.661%	.835
2	Q7, Q15, Q21, Q22, Q23, Q24	1.954	13.96%	.762
3	Q17, Q15, Q16, Q18, Q19, Q21, Q22, Q23	1.359	9.71%	.723

All 3 components have a CA value of more than 0.7, which is considered to be the recommended [36, 8]. This means that the components have a high level of internal consistency, as they accurately measure what is intended.

6.3 Profiling the participants

When the questionnaire was concluded, there was a total of 215 respondents. However, because we discarded the responses from 3 participants due to their inconsistency (see Section 6.2) we are left with a total of 212 responses. For this reason, all of the following percentages displayed and discussed in this Section are based on the 212 responses. Upon

observation of the responses to question 1 of the questionnaire, we can see that approximately 47.64% of the respondents heard about the survey on *Reddit* and 26.42% heard about it on *LinkedIn*. The remainders heard about it on *MATLAB Central*, *ResearchGate*, E-mail, *GNU Octave Discourse* or through word of mouth (see Figure 6.3).

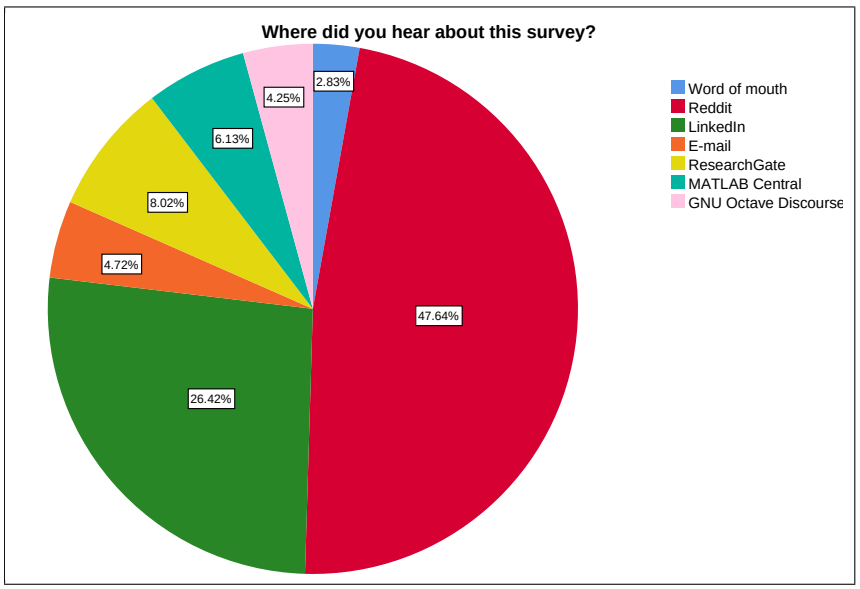

Figure 6.3: *Where did you hear about this survey?*

In Figure 6.4, we can observe that the participants comprised 68 students (32.08%), 15 teachers (7.08%), 51 researchers (24.06%), 74 employed programmers (34.91%, employed by a company, freelancers or self-employed) and 3 retirees (1.42%), according to the responses to question 3 of the questionnaire.

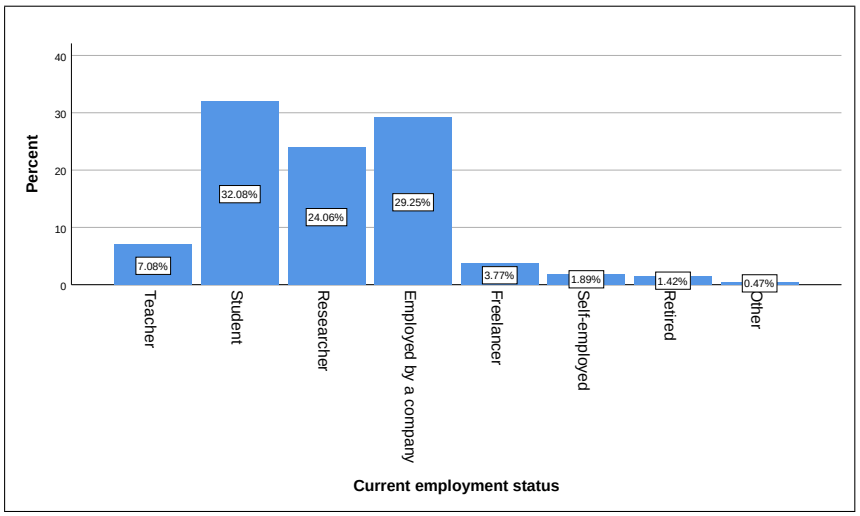

Figure 6.4: Participants' employment status.

We also asked the participants the application domain for which they used *MATLAB* and similar languages, in question 11 of the questionnaire, allowing participants to

state multiple different application domains. In Table 6.19, we can observe that 61.32% reported that they use it for *Data Analytics*. 45.75% have reported that they use these languages for *Signal Processing*, 35.38% use it for *Control Systems*, 28.77% for *Image and Video Processing* and 26.42% for *Machine Learning*. Alongside these specialisations, some participants also stated that they use *MATLAB* and similar languages for other application domains such as *System Modelling* (8.02%), *Wireless Communications* (5.66%), *Simulations* (5.66%), *Computational Finance* (5.19%), *Computational Biology* (5.19%), among other miscellaneous purposes. These options are non-exclusive, meaning that a respondent may pick more than one option.

Table 6.19: Application domain for which the participants use *MATLAB* and similar languages.

Domain	**Percentage**
Data Analytics	61.32%
Signal Processing	45.75%
Control Systems	35.38%
Image and Video Processing	28.77%
Machine Learning	26.42%
System Modelling	8.02%
Wireless Communications	5.66%
Simulations	5.66%
Computational Finance	5.19%
Computational Biology	5.19%

In Table 6.20, we can observe that, from the responses to question 6 of the questionnaire, approximately 94%, or 199 of the 212 participants, declared that *MATLAB* is one of the languages they use and 23.58% or 50 of the 212 say that *Octave* is one of the languages they use. Additionally, 10.38% say that they use *Scilab* and 3.77% say that they use *Rlab*. Finally, 3.30% of the participants stated that they use *Julia*. These options are non-exclusive, meaning that a respondent may pick more than one option indicating that they use more than one programming language. The mention of *Julia* was not anticipated and it is a language that was not included as a response option, but that participants wrote, on their initiative, in an empty text-field response option.

Table 6.20: *Which of the following programming languages (MATLAB and similar languages) do you use?*

MATLAB-like languages used	Percentage	Value
MATLAB	93.87%	199
Octave	23.58%	50
Scilab	10.38%	22
Rlab	3.77%	8
Julia	3.30%	7

In addition, in question 8 of the questionnaire we asked the participants what is the

language they use the most. In Table 6.21 we can observe that 62.74%, or 133 of the 212 participants, stated that *MATLAB* is the language they use most. Secondly, 11.79%, or 25 participants, stated that *C* is the language they use most. Then, *Python* with 8.96% and of the participants stating that it is the language they use most. 6.60% of the participants stated that *C++* is the language they use most. Other languages were also mentioned, such as *R*, *Octave*, *C#*, *Julia*, and *Scilab*.

Table 6.21: *What programming language do you use the most?*

Most used language	Percentage	Count
MATLAB	62.74%	133
C	11.79%	25
Python	8.96%	19
C++	6.60%	14
R	2.36%	5
Octave	2.36%	5
C#	1.42%	3
Julia	1.42%	3
Scilab	1.42%	3

Regarding experience with the language, according to the responses to question 5 of the questionnaire, 14.62% of the respondents have less than 1 year of experience with *MATLAB* or its clone languages, 28.30% have between 1 and 4 years of experience, 22.64% have between 4 and 7 years of experience and 11.79% have between 7 and 10 years of experience. The remaining 22.64% have more than 10 years of experience with *MATLAB* or its clone languages. In Figures 6.5, 6.6 and 6.7 we can observe more information, including the correlation between years of experience with *MATLAB* or its clone languages and the respondents' self-assessed level of expertise with *MATLAB*.

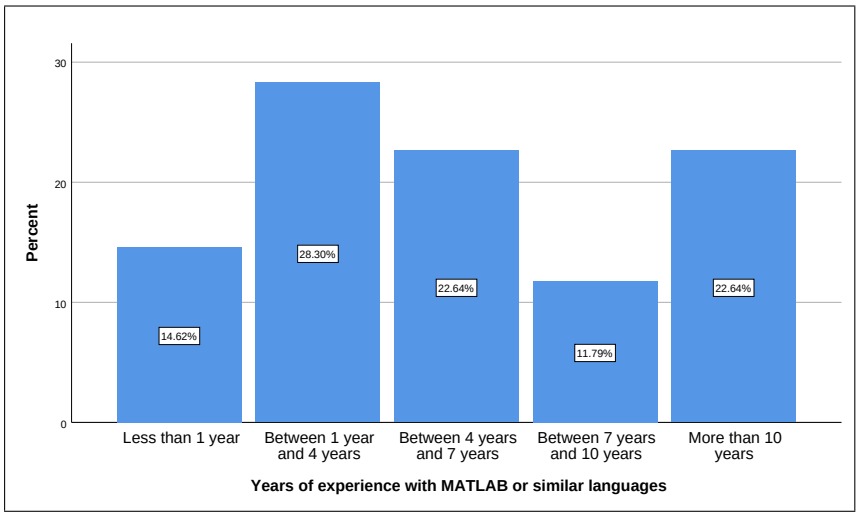

Figure 6.5: Participants' years of experience.

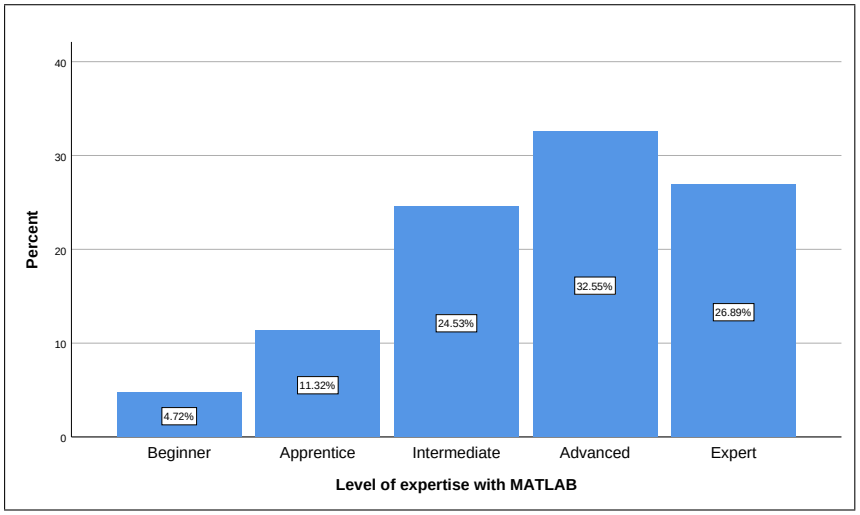

Figure 6.6: Participants' level of expertise.

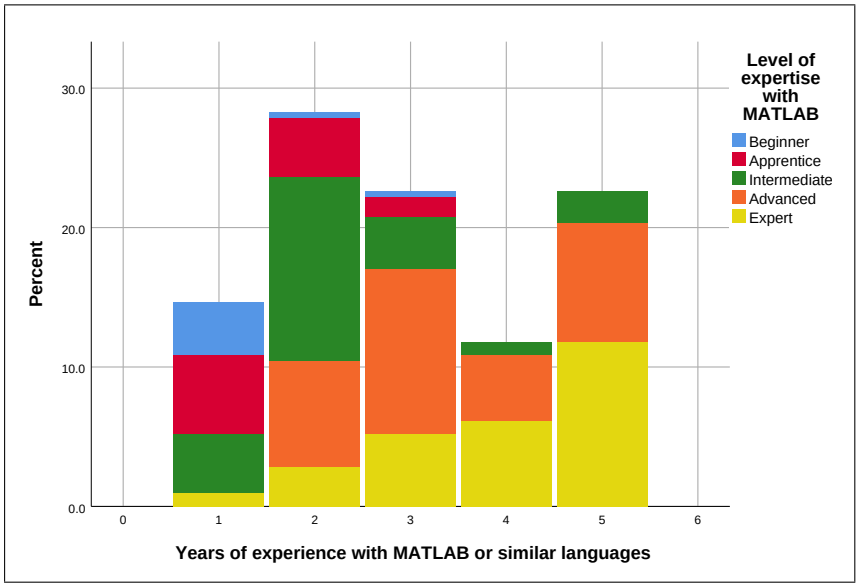

Figure 6.7: Years of experience with *MATLAB* vs level of expertise with *MATLAB*.

Furthermore, in Table 6.22 we can observe that 84.91%, or 180 of the 212 participants, have stated development environments are on a *Windows* operating system (question 13 of the questionnaire). Moreover, 33.96% declare to use Unix or Linux and 20.75% declare to use macOS for their development environments. These options are non-exclusive, meaning that a respondent may use one, two, or all of the three options.

Table 6.22: *On which operating systems are your development environments?*

OS	Percentage	Count
Windows	84.91%	180
Linux	33.96%	72
macOS	20.75%	44

In Figure 6.8 we can observe that, according to the responses to question 14 of the questionnaire, approximately 87%, or 185 of the 212 participants, stated that they do not exclusively use the command window when working with *MATLAB*. In other words, these participants typically write their *MATLAB* code in m-files, and they either never use the command window, or they use it to solve small problems or to complement their coding (i.e., inspecting variables, testing functions). Thus, the remaining 27 participants did not advance to the subsequent section in the questionnaire and were, instead, redirected to the end of the questionnaire, leaving us with 185 full responses.

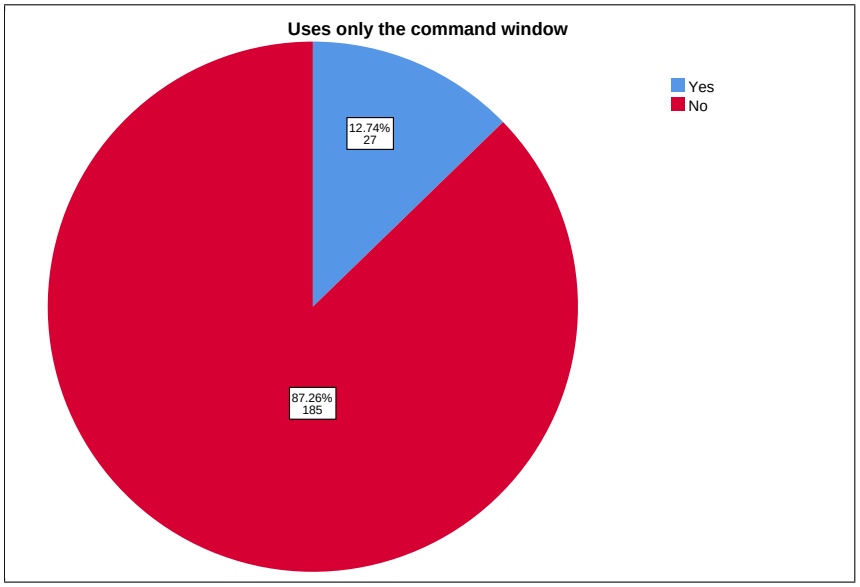

Figure 6.8: *Do you use only the command window?* (As opposed to writing in m-files)

To calculate the percentage of participants that utilise OOP in *MATLAB*, we analyse the answers to question 20 in the questionnaire, "Regarding *MATLAB*'s modules, in my programs I use..." (see appendix A). If we consider the participants that use both Classes and Objects in their programs as those that use OOP, we can declare that approximately 22% of participants utilise OOP in *MATLAB*, or 47 out of 212. Furthermore,

approximately 61% of the respondents stated that they use OOP in other programming languages.

Regarding typical uses of *MATLAB*, in question 25 of the questionnaire, the participants considered *MATLAB*'s strongest competitors to be: *Python*, with approximately 60% of the responses, *Octave*, with approximately 8% of the responses, and *R* with approximately 6% of the responses. Other languages mentioned include: *Scilab*, *Julia*, *Wolfram Mathematica* and *C++*, but each of these accounted for approximately 5% or less of the number of valid responses. Since the question corresponding to these statistics had an open-answer option, it is worth mentioning that we have excluded any unclear or invalid responses from the total percentages.

6.4 Hypothesis formulation

A hypothesis is used to explain or predict a phenomenon in a designated environment. It must, therefore, be testable and falsifiable, and it must state an expected relationship between different variables [99].

A null hypothesis, H_0, states that there is not a pattern in the environment that is being tested. This is the hypothesis that an individual running an experiment wants to be able to reject using the data collected [99].

An alternative hypothesis, H_1, is the hypothesis that is accepted in the case that the null hypothesis is rejected [99]. Table 6.23 presents the pairs of hypotheses formulated in the context of this thesis.

Table 6.23: Pairs of null hypotheses, H_0, and alternative hypotheses, H_1.

	H_0	H_1
1	A user's level of expertise is not correlated to the application domain in which they program.	A user's level of expertise is correlated to the application domain in which they program.
2	A user's level of expertise is not correlated to the usual size of their programs.	A user's level of expertise is correlated to the usual size of their programs.
3	The years of experience a user has is not correlated to the importance they give to their programs' maintainability and reusability.	The years of experience a user has is correlated to the importance they give to their programs' maintainability and reusability.
4	A user's effort to keep a program maintainable is not affected by their expectation of being the sole user of that program.	A user's effort to keep a program maintainable is affected by their expectation of being the sole user of that program.
5	A user's level of expertise does not influence their opinion on *MATLAB*'s support to modularity.	A user's level of expertise directly influences their opinion on *MATLAB*'s support to modularity.
6	The importance a user gives to the program's maintainability does not influence their satisfaction with *MATLAB*'s support to modularity.	The importance a user gives to the program's maintainability directly influences their satisfaction with *MATLAB*'s support to modularity.

6.5 Hypothesis testing

In this Section we test the null hypotheses formulated (see Section 6.4) based on the answers provided to the questions present in the questionnaire (see appendix A). Because most of the variables are ordinal with a discrete distribution, corresponding to the 5-point

Likert scale questions in the questionnaire, we use non-parametric tests because with this type of variable we are not able to meet the assumption required for parametric tests that states that the data should be normally distributed. In addition, we assume a statistical significance level α of 5%, or in other words, $\alpha = 0.05$.

Spearman's rank-order correlation, otherwise known as **Spearman's Correlation (SC)**, is a non-parametric measure of the strength and direction of association between two continuous or ordinal variables. In other words, it calculates a coefficient, ρ, which accurately determines whether there is a monotonic association between the two variables being analysed. However, in order to obtain a valid result, it is only appropriate to use SC if the data being analysed is verified by three assumptions:

- There are two variables that are measured on a continuous and/or ordinal level;
- The two variables represent paired observations;
- There is a monotonic relationship between the two variables.

The coefficient, ρ can range from -1 to +1, in which the sign (positive or negative) indicates the direction of the relationship, and the absolute value indicates the strength of the relationship. Thus, a ρ of +1 suggests a perfect positive association and a ρ of -1 suggests a perfect negative association of ranks. A value of 0 indicates that there is no association between the ranks [21, 14].

One-Sample Proportion Test (OSPT) is a non-parametric test used to assess whether a proportion of a population is different than its hypothesized proportion in the population from which the sample data are drawn [67]. In order to obtain valid results using OSPT, the data should be verified by four assumptions:

- The variable is binary;
- The variable is observed a known number of times;
- The probability of the outcome of interest is constant in every trial;
- The trials are independent.

Additionally, with this test we will also assume a statistical significance level of 0.05.

6.5.1 Hypothesis 1 - A user's level of expertise is not correlated to the application domain in which they program.

In order to test this hypothesis, we measure the correlation between the users' level of experience with *MATLAB* and the application domain in which they program, for each of the different domains shown in the data:

- *Data Analytics*;
- *Signal Processing*;
- *Control Systems*;
- *Image and Video Processing*;
- *Machine Learning*;
- *System Modelling*;
- *Wireless Communications*;
- *Simulations*;
- *Computational Finance*;
- *Computational Biology*;
- other miscellaneous responses.

To measure this correlation, we apply a OSPT using the answers to question 7 (*Rate your level of experience with MATLAB.*) and the answers to question 11 (*For what do you use MATLAB or similar languages?*). With each of the different response options to question 11 representing a dichotomous variable with a positive ('Yes') and a negative ('No') value, we firstly calculate the percentage of participants that have a positive response in each of the dichotomous variables. Then, for each level of experience (1-'Beginner', 2-'Apprentice', 3-'Intermediate', 4-'Advanced', 5-'Expert') we calculate how many participants have a positive response in each of the dichotomous variables from question 11 (e.g. what percentage of experts use *MATLAB* for *Data Analytics*).

Table 6.24: One-Sample Proportion Test - Hypothesis 1 - Experts that use *MATLAB* for *Machine Learning*.

Null hypothesis	**Test**	**Sig.**	**Decision**
The categories defined by *Machine Learning* in experts = (Yes) and (No) occur with probabilities 0.264 and 0.736.	One-Sample Binomial Test	.013	Reject the null hypothesis.

Approximately 26.4% of the participants use *MATLAB* for *Machine Learning*. However, approximately 40.3% of the 'Expert' participants use *MATLAB* for the same purpose. From the Table 6.24, we can see that the corresponding OSPT for this group of participants is statistically significant as demonstrated by $p - value \leq 0.05$.

Approximately 5.7% of the participants use *MATLAB* for *Wireless Communications*. However, approximately 12.3% of the 'Expert' participants use *MATLAB* for the same purpose. From the Table 6.25, we can see that the corresponding OSPT for this group of participants is statistically significant as demonstrated by $p - value \leq 0.05$.

Table 6.25: One-Sample Proportion Test - Hypothesis 1 - Experts that use *MATLAB* for *Wireless Communications*.

Null hypothesis	Test	Sig.	Decision
The categories defined by *Wireless Communications* in experts = (Yes) and (No) occur with probabilities 0.057 and 0.943.	One-Sample Binomial Test	.032	Reject the null hypothesis.

Table 6.26: One-Sample Proportion Test - Hypothesis 1 - Advanced participants that use *MATLAB* for *Signal Processing*.

Null hypothesis	Test	Sig.	Decision
The categories defined by *Signal Processing* in advanced participants = (Yes) and (No) occur with probabilities 0.458 and 0.542.	One-Sample Binomial Test	.021	Reject the null hypothesis.

Approximately 45.8% of the participants use *MATLAB* for *Signal Processing*. However, approximately 60.9% of the 'Advanced' participants use *MATLAB* for the same purpose. From the Table 6.26, we can see that the corresponding OSPT for this group of participants is statistically significant as demonstrated by $p - value \leq 0.05$.

Table 6.27: One-Sample Proportion Test - Hypothesis 1 - Advanced participants that use *MATLAB* for *Control Systems*.

Null hypothesis	Test	Sig.	Decision
The categories defined by *Control Systems* in advanced participants = (Yes) and (No) occur with probabilities 0.354 and 0.646.	One-Sample Binomial Test	.021	Reject the null hypothesis.

Approximately 35.4% of the participants use *MATLAB* for *Control Systems*. However, approximately 47.8% of the 'Advanced' participants use *MATLAB* for the same purpose. From the Table 6.27, we can see that the corresponding OSPT for this group of participants is statistically significant as demonstrated by $p - value \leq 0.05$.

Table 6.28: One-Sample Proportion Test - Hypothesis 1 - Intermediate participants that use *MATLAB* for *Signal Processing*.

Null hypothesis	Test	Sig.	Decision
The categories defined by *Signal Processing* in intermediate participants = (Yes) and (No) occur with probabilities 0.458 and 0.542.	One-Sample Binomial Test	.021	Reject the null hypothesis.

Approximately 45.8% of the participants use *MATLAB* for *Signal Processing*. However, approximately 28.8% of the 'Intermediate' participants use *MATLAB* for the same purpose. From the Table 6.28, we can see that the corresponding OSPT for this group of participants is statistically significant as demonstrated by $p - value \leq 0.05$.

Approximately 35.4% of the participants use *MATLAB* for *Control Systems*. However,

Table 6.29: One-Sample Proportion Test - Hypothesis 1 - Intermediate participants that use *MATLAB* for *Control Systems*.

Null hypothesis	**Test**	**Sig.**	**Decision**
The categories defined by *Control Systems* in intermediate participants = (Yes) and (No) occur with probabilities 0.354 and 0.646.	One-Sample Binomial Test	.021	Reject the null hypothesis.

approximately 21.2% of the 'Intermediate' participants use *MATLAB* for the same purpose. From the Table 6.29, we can see that the corresponding OSPT for this group of participants is statistically significant as demonstrated by $p - value \leq 0.05$.

Therefore, we are able to reject Hypothesis 1, as there is a correlation between the level of experience in *MATLAB* of the participants and whether or not they use it for *Machine Learning*, *Wireless Communications*, *Signal Processing* and *Control Systems*. Furthermore, the rest of the combinations of application domains and experience levels were also tested but they did not result in statistically significant results.

6.5.2 Hypothesis 2 - A user's level of expertise does not influence the usual size of their programs.

To test this hypothesis, we measure the correlation between the users' level of experience with *MATLAB* and the number of m-files their programs tend to have. For this, we use the variables corresponding to the answers to questions 7 (*Rate your level of experience with MATLAB.*) and 18 (*My MATLAB programs tend to have...*). Q7 corresponds to the answers to question 7 and Q18 corresponds to the answers to question 18.

In Table 6.30 we present the results of the SC test. In this Table, we can see that the correlation coefficient, ρ, between Q7 and Q18 is 0.365, as is shown in the "Correlation Coefficient" row, in the highlighted quarter of the Table. This indicates a moderate correlation [1]. Additionally, we can observe a statistical significance value of 0 and therefore $p \leq 0.05$.

Table 6.30: Spearman's Correlation - Hypothesis 2 - First test.

			Level of expertise with *MATLAB*	Number of m-files my programs tend to have...
Spearman's rho	Level of expertise with *MATLAB*	Correlation Coefficient	1.000	.365**
		Sig. (2-tailed)	.	.000
		N	212	185
	Number of m-files my programs tend to have...	**Correlation Coefficient**	**.365****	1.000
		Sig. (2-tailed)	**.000**	.
		N	**185**	185
** Correlation is significant at the 0.01 level (2-tailed).				

Thus in conclusion, there was a **moderate, positive correlation** between the users' level of experience with *MATLAB* and the number of m-files their programs tend to have ($\rho = 0.365, p = 0$). Thus, we are able to reject Hypothesis 2.

However, we also want to measure the correlation between the users' level of experience with *MATLAB* and the number of functions their m-files tend to have. We do

not consider the result of this experiment relevant for the case of Hypothesis 2, but it is nonetheless a relevant statistic to analyse. In this case we use the variables corresponding to the answers to questions 7 (*Rate your level of experience with MATLAB.*) and 19 (*The m-files I deal with tend to have...*). Q7 corresponds to the answers to question 7 and Q19 corresponds to the answers to question 19.

In Table 6.31 we present the results of the SC test. In this Table, we can see that the correlation coefficient, ρ, between Q7 and Q19 is 0.061, as is shown in the "Correlation Coefficient" row, in the highlighted quarter of the Table. This would indicate a weak correlation. However, we can observe a statistical significance value of 0.41 and therefore $p \geq 0.05$.

Table 6.31: Spearman's Correlation - Hypothesis 2 - Second test.

			Level of expertise with *MATLAB*	The m-files I deal with tend to have...
Spearman's rho	Level of expertise with *MATLAB*	Correlation Coefficient	1.000	.061
		Sig. (2-tailed)	.	.410
		N	212	185
	The m-files I deal with tend to have...	**Correlation Coefficient**	**.061**	1.000
		Sig. (2-tailed)	**.410**	.
		N	**185**	185

Therefore, we cannot conclude that there was a statistically significant correlation between the users' level of experience with *MATLAB* and the number of functions their m-files tend to have ($\rho = 0.061, p = 0.41$).

6.5.3 Hypothesis 3 - The years of experience a user has with *MATLAB* is not correlated to the importance they give to their programs' maintainability and reusability.

In order to test this hypothesis, we use the variables corresponding to questions 5 (*How many years of experience do you have with MATLAB or a similar language?*) and 15 (*When I develop a program in MATLAB I always try to make it easily reusable and maintainable.*). Q5 corresponds to the answers to question 5, whereas Q15 corresponds to the answers to question 15.

In Table 6.32 we present the results of the SC test. In this Table, we can see that the correlation coefficient, ρ, between Q5 and Q15 is 0.06, as is shown in the "Correlation Coefficient" row, in the highlighted quarter of the Table. This would indicate a weak correlation. However, we can observe a statistical significance value of 0.414 and therefore $p \geq 0.05$.

Table 6.32: Spearman's Correlation - Hypothesis 3.

			Years of experience with *MATLAB* or similar languages	When I develop a program in *MATLAB*, I always try to make it easily reusable and maintainable
Spearman's rho	Years of experience with *MATLAB* or similar languages	Correlation Coefficient	1.000	.060
		Sig. (2-tailed)	.	.414
		N	212	185
	When I develop a program in *MATLAB*, I always try to make it easily reusable and maintainable	**Correlation Coefficient**	**.060**	1.000
		Sig. (2-tailed)	**.414**	.
		N	**185**	185

Therefore, we are not able to reject Hypothesis 3. We can instead conclude, by accepting the null hypothesis, that **there is not a statistically significant correlation** between the years of experience a user has with *MATLAB* and the importance they give to their programs' maintainability and reusability ($\rho = 0.06$, $p = 0.414$).

6.5.4 Hypothesis 4 - A user's effort to keep a program maintainable is not affected by their expectation of being the sole user of that program.

Somers' delta, otherwise known as **Somers' *d***, is a nonparametric measure of association between an ordinal dependent variable and an ordinal independent variable [85, 66]. It can also be interpreted as the number of concordant pairs minus the number of discordant pairs among pairs not tied on the independent variable. While there are other tests to analyse the association between two ordinal variables, Somers' d is specifically appropriate when the distinction between a dependent and independent variable is relevant. So to run Somers' d we need to consider its two assumptions:

- There is a dependent variable and an independent variable, and they are both measured on an ordinal scale;
- There is a monotonic relationship between the dependent and the independent variable.

Somers' delta value can range from -1 to +1. The sign (positive or negative) indicates the direction of the relationship, and the absolute value indicates the strength of the relationship. Thus, a value of -1 indicates that all of the observed pairs are discordant and a value of +1 indicates that all of the observed pairs are concordant.

To test Hypothesis 4, we use the variables corresponding to the answers to questions 15 (*When I develop a program in MATLAB I always try to make it easily reusable and maintainable.*) and 16 (*I expect to be the sole user of my MATLAB programs.*) from the questionnaire (see appendix A).

Q15, the variable corresponding to the answers to question 15, is the dependent variable and it has five categories, ranging from "Stronghly disagree" to "Strongly agree". And Q16, the variable corresponding to the answers to question 16, is the independent variable and it has the same five categories as Q15.

In the Directional Measures Table 6.33, we present the results of the Somers' d test. From this Table, we can observe that the value of Somers' d, for the case of Q15 being the dependent variable, is -0.226 as demonstrated in the "Value" column. Additionally, in the "Approximate Significance" column we can notice that the statistical significance value (p-value) of this test is, approximately, 0. Therefore, $p \leq 0.05$ which indicates that we have a statistically significant result.

Additionally, in Figure 6.9 we can observe a clustered bar chart where we can visually interpret the differences in proportions and weight between the different categories of the dependent variable (Q15) for the different values of the independent variable (Q16).

Table 6.33: Somers' *d* - Hypothesis 4.

			Value	Asymptotic Standard *Error*[a]	Approximate *T*[b]	Approximate Significance
Ordinal by Ordinal	Somers' d	Symmetric	-.244	0.064	-3.800	.000
		I expect to be the sole user of my *MATLAB* programs Dependent	-.265	.070	-3.800	.000
		When I develop a program in *MATLAB*, I always try to make it easily reusable and maintainable Dependent	-.226	.060	-3.800	.000
a. Not assuming the null hypothesis.						
b. Using the asymptotic standard error assuming the null hypothesis.						

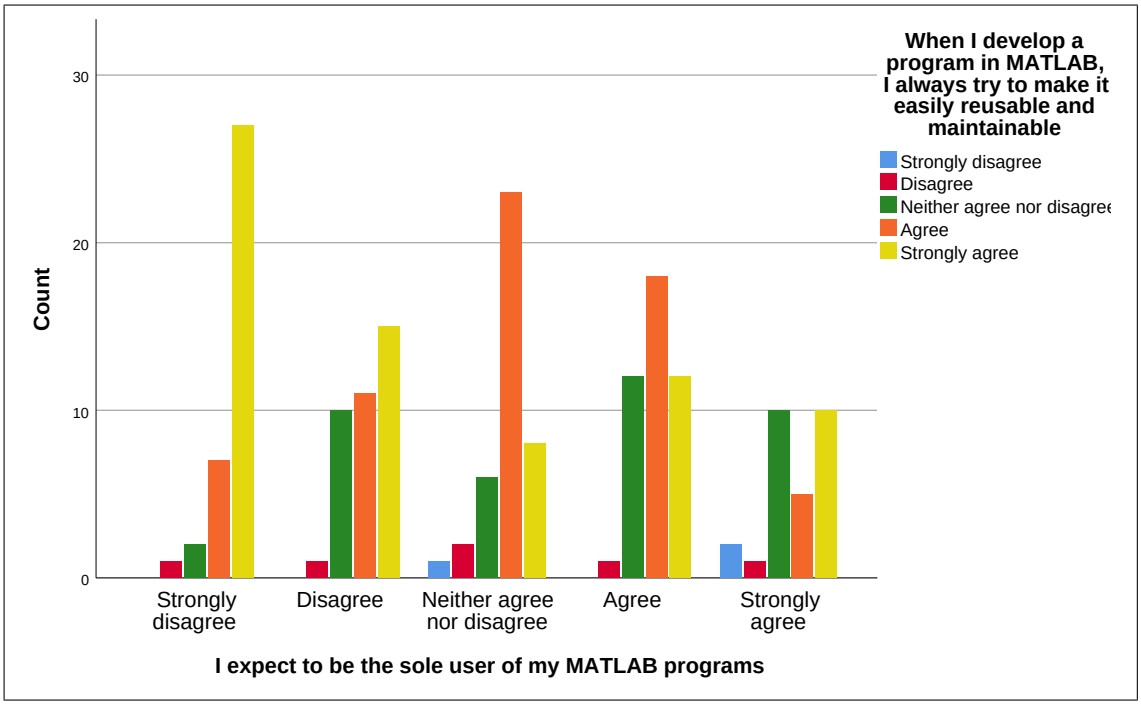

Figure 6.9: Correlation between the variables - Hypothesis 4.

Therefore we can conclude that there was a **negative correlation** (with absolute value ≥ 0.2) between the participants' expectation to be the sole user of their programs and the participants' effort to keep a program maintainable ($d = -0.226$, $p = 0$). Thus, we can reject Hypothesis 4.

6.5.5 Hypothesis 5 - A user's level of expertise does not influence their opinion on *MATLAB*'s support to modularity.

To test this hypothesis, we use the variables corresponding to the answers to questions 7 (*Rate your level of experience with MATLAB.*) and 23 (*I am satisfied with MATLAB's current support to modularity.*) from the questionnaire (see appendix A).

Q7, the variable corresponding to the answers to question 7, is the independent variable and it has five categories ranging from "Beginner" to "Expert". And Q23, the variable corresponding to the answers to question 23, is the dependent variable and it also has five categories, ranging from "Strongly disagree" to "Strongly agree".

In the Directional Measures Table 6.34 we present the results of the Somers' *d* test.

From this Table, we can notice that the value of Somers' d, for the case of Q23 being the dependent variable, is 0.154 as demonstrated by the "Value" column. Additionally, in the "Approximate Significance" column we can see that the statistical significance value (p-value) of this test is, approximately, 0.014. Therefore, $p \leq 0.05$ which indicates that we have a statistically significant result.

Table 6.34: Somers' d - Hypothesis 5.

			Value	Asymptotic Standard $Error^a$	Approximate T^b	Approximate Significance
Ordinal by Ordinal	Somers' d	Symmetric	.157	.064	2.453	.014
		Level of expertise with *MATLAB* Dependent	.159	.065	2.453	.014
		I am satisfied with *MATLAB*'s current support to modularity Dependent	.154	.063	2.453	.014
a. Not assuming the null hypothesis.						
b. Using the asymptotic standard error assuming the null hypothesis.						

Additionally, in Figure 6.10 we can observe a clustered bar chart where we can visually interpret the differences in proportions and weight between the different categories of the dependent variable (Q23) for the different values of the independent variable (Q7).

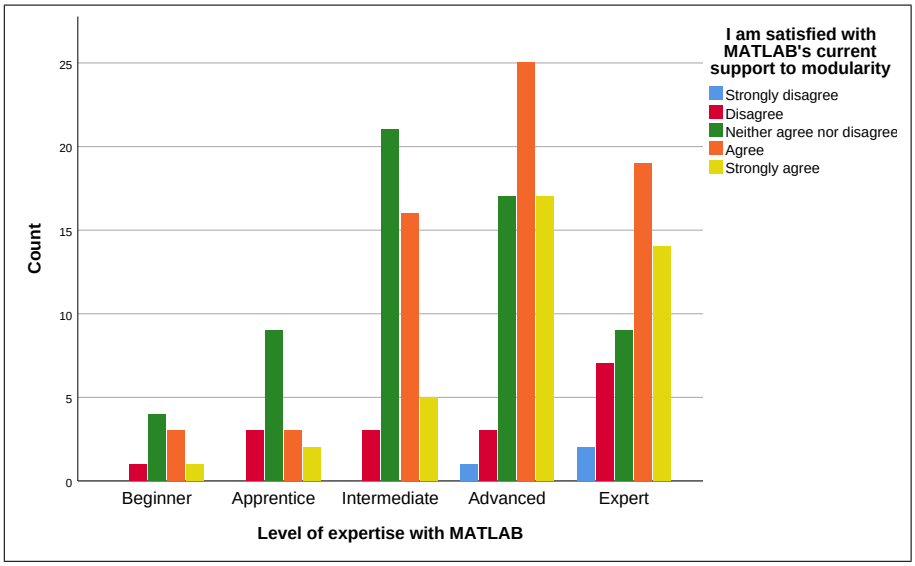

Figure 6.10: Correlation between the variables - Hypothesis 5.

Therefore we can conclude that there was a **weak, positive correlation** (with absolute value ≤ 0.2) between participants' satisfaction with *MATLAB*'s support to modularity and their level of experience with *MATLAB*. Therefore we reject Hypothesis 5 ($d = 0.154$, $p = 0.014$).

6.5.6 Hypothesis 6 - The importance a user gives to the program's maintainability does not influence their satisfaction with *MATLAB*'s support to modularity.

To test this hypothesis, we use the variables corresponding to the answers to questions 15 (*When I develop a program in MATLAB I always try to make it easily reusable and maintainable.*) and 23 (*I am satisfied with MATLAB's current support to modularity.*) from the questionnaire (see appendix A).

Q15, the variable corresponding to the answers to question 15, is the independent variable and it has five categories, ranging from "Stronghly disagree" to "Strongly agree". And Q23, the variable corresponding to the answers to question 23, is the dependent variable and it has the same five categories as Q15.

In the Directional Measures Table 6.35 we present the results of the Somers' d test. From this Table, we can observe that the value of Somers' d, for the case of Q23 being the dependent variable, is 0.230 as demonstrated in the "Value" column. Additionally, in the "Approximate Significance" column we can notice that the statistical significance value (p-value) of this test is, approximately, 0. Therefore, $p \leq 0.05$ which indicates that we have a statistically significant result.

Table 6.35: Somers' d - Hypothesis 6.

			Value	Asymptotic Standard $Error^a$	Approximate T^b	Approximate Significance
Ordinal by Ordinal	Somers' d	Symmetric	.225	.059	3.775	.000
		When I develop a program in *MATLAB*, I always try to make it easily reusable and maintainable Dependent	.219	.058	3.775	.000
		I am satisfied with *MATLAB*'s current support to modularity Dependent	.230	.061	3.775	.000
a. Not assuming the null hypothesis.						
b. Using the asymptotic standard error assuming the null hypothesis.						

Additionally, in Figure 6.11 we can observe a clustered bar chart where we can visually interpret the differences in proportions and weight between the different categories of the dependent variable (Q23) for the different values of the independent variable (Q15).

Therefore we can conclude that there was a **positive correlation** (with absolute value ≥ 0.2) between the importance a respondent gives to their programs' maintainability and their satisfaction with *MATLAB*'s current support to modularity. ($d = 0.230$, $p = 0$). In other words, a user that considers a program's maintainability to be important is more likely to demonstrate satisfaction with *MATLAB*'s current support to modularity. Thus, we reject Hypothesis 6.

In conclusion, we rejected 5 of the 6 null hypotheses formulated. In appendix C can be found a summary in the form of a Table, containing the verdict of the test of each null hypothesis as well as the types of tests used.

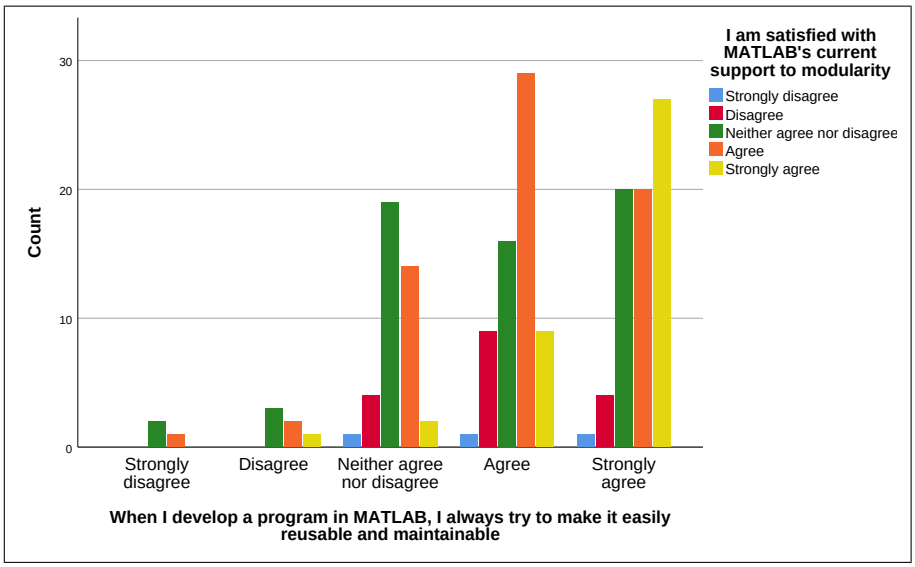

Figure 6.11: Correlation between the variables - Hypothesis 6.

6.6 Results and implications

6.6.1 Answering the research questions

1. How is the community of users of *MATLAB* and its clones structured and divided, according to their level of expertise, the application domain in which they program, among other factors?

With this survey research, we wanted to stratify the community into different levels of expertise, different domains, different languages used, and multiple other factors. We were able to do just that (see Section 6.3), and with the testing of Hypotheses 1 and 2 we were able to understand the correlation between the participants' level of expertise and their applications' domain and usual program size.

From the testing of these hypotheses, we are able to conclude that a *MATLAB* user's level of expertise is directly correlated with their application's domain. More specifically, we can deduce that the users who consider themselves 'Experts' are more likely to be working with *MATLAB* or its clone languages for domains such as *Machine Learning* and *Wireless Communications*. Additionally, we can deduce that the users who consider themselves 'Advanced' in terms of their level of expertise are more likely to be working with *MATLAB* or its clone languages for domains such as *Signal Processing* and *Control Systems*. On the contrary, the users who consider themselves 'Intermediate' are less likely to be using *MATLAB* or its clone languages for *Signal Processing* and *Control Systems*. This suggests that domains such as *Machine Learning*, *Wireless Communications*, *Signal Processing* and *Control Systems* may have a steeper learning curve.

Additionally, we were also able to deduce that a *MATLAB* user's level of expertise is also directly correlated with the size of their programs. In other words, we deduced that

the more experienced users of *MATLAB* are more likely to work on programs involving a larger number of m-files. A metric for which the mean is, approximately, 6 to 10 m-files, users with above-average experience tend to have more than that. This suggests that the more experienced users have a better grasp of the techniques that allow for a better organisation of a *MATLAB* program.

To summarise, we can conclude that some fields, such as *Machine Learning* or *Signal Processing*, involve a steeper learning curve than others, as they may require more complex programming techniques. Additionally, we can deduce that with more experience with *MATLAB* a user becomes more capable of building larger and more scalable programs, with a higher number of m-files. In turn, this suggests that *MATLAB* provides the necessary tools and support (namely strong support to modularity) to build a large scale project, which the more experienced *MATLAB* users are more able to take advantage of.

2. How proficient are the users of *MATLAB* and its clones?

Through this research, we also got a better understanding of the level of proficiency of the users of *MATLAB* and its clones. For instance, we now have a better estimate of how much these users focus on their programs' maintainability, and which *MATLAB* modules are used the most.

From testing the hypotheses formulated, and more specifically Hypothesis 4, we are able to conclude that there is an inverse correlation with *MATLAB* users' effort put into the maintainability of their programs and their expectation of being the sole user of their programs. In other words, the more users expect other people to use their programs, the more effort they will put into making sure the code is easily understandable, maintainable and reusable.

This suggests that users of *MATLAB* and its clones are more worried about how well others are able to perceive and understand their code than how they themselves will understand their own code in the future. That is to say, users that work on programs solely by themselves do not care as much about the reusability and maintainability of their programs as they may be confident in their ability to revisit their own code in the future and easily understanding it. On the other hand, users that work on programs alongside other colleagues may feel a bigger necessity to keep the code presentable and reusable so that their peers have an easier time understanding and building upon it.

3. What is the level of users' satisfaction with *MATLAB*'s current support for modularity?

Furthermore, we also wanted to reach a better grasp of how satisfied the users are with *MATLAB*'s current support for modularity. With this research, we were able to better understand how satisfied the users are with the current modularity capabilities of *MATLAB*, and what they think of as good alternatives to *MATLAB* in regards to OOP (see Section 6.3).

Through the testing of Hypotheses 5 and 6, we are able to identify a direct correlation between the users' level of expertise and their opinion on *MATLAB*'s current support to modularity. In other words, the more experienced users have shown to be more satisfied with *MATLAB*'s support to modularity than the less experienced users. As the mean satisfaction on a scale of 1 to 5 was shown to be 3.65, the more experienced users are likely to be above that level of satisfaction. This suggests that more experienced users are able to more easily overcome the disadvantages of this environment, or that they think that *MATLAB* offers more than what the alternatives are able to.

In addition, we also identified a direct correlation between the importance *MATLAB* users give to their programs' maintainability and their satisfaction with *MATLAB*'s support to modularity. The users that are generally more aware about and put more effort into their programs' maintainability and reusability also show to be the ones more satisfied with *MATLAB*'s current support to modularity.

In conclusion, we can safely say that more experience and more practice with *MATLAB* lead to a greater satisfaction with *MATLAB*'s current support to modularity, which in itself might indicate that OOP in *MATLAB* is not very beginner-friendly. The more experienced, as well as the more dedicated to their programs' maintainability, are generally more satisfied with *MATLAB*'s current modular capabilities. That barrier seems to be, however, hard to break. Thus beginners or less dedicated users may be more inclined to pick alternatives to *MATLAB*.

6.6.2 Inferences

There is a balanced distribution across different levels of experience in the community.

Through this empirical study, we are able to infer some more general conclusions. For instance, we noticed a healthy mix of all levels of experience in the community. We believe that there is not much bias from the specific platforms that we chose to reach out to, as these were quite varied in their responses to this matter. For example, the most common answer for "Years of experience with programming" from the *Reddit* participants was "between 1 and 4 years", where as from the *ResearchGate* and *MATLAB Central* participants it was "more than 10 years". Thus, we believe this healthy combination of less experienced and more experienced users of *MATLAB* and its clones is true across this community as a whole, meaning that new users are joining the ecosystem as much as more experienced users also remain in the ecosystem.

Across all levels of experience, there are users interacting with *MATLAB* strictly through the command window.

Additionally, we are able to infer a few conclusions concerning how the participants use these languages. Specifically, we found that, contrary to what we expected, students

are not the only demographic group that uses just the command window component of *MATLAB*, as opposed to writing in m-files. In fact, out of the 27 participants that stated they exclusively use the command window when working with *MATLAB*, only 8 (approximately 30%) of those were students. Similarly, only 8 out of those 27 participants stated that they have less than 1 year of programming experience. Furthermore, 9 of those 27 participants said to have more than 10 years of experience with programming. This data suggests that there are widely different expertise levels among the users of *MATLAB* that use it on a superficial level, through just the command window functionality. These users do not write their programs on m-files and most likely use *MATLAB* only for simpler purposes.

The use of OOP with *MATLAB* is uncommon.

Approximately 22%, 47 of the 212 participants, use OOP with *MATLAB*, as noted in Section 6.3. However, 61% of the participants stated that they use OOP with other programming languages. These results suggest that when they wish to use OOP, the majority of the respondents opts to use a programming language other than *MATLAB*. This could be because they view it as a better environment for an OOP approach, or due to a lack of awareness of the current state of *MATLAB*'s OOP capabilities.

The majority of *MATLAB* users are satisfied with its support to modularity.

Furthermore, out of the 185 participants who answered to Sections 3 and 4 of the questionnaire (the sections concerning *MATLAB* program reusability, respondents' level of satisfaction with *MATLAB* and their general use of the language, to which the respondents who use *MATLAB* strictly through the command window did not have access to), approximately 56.8% claimed to be satisfied with *MATLAB*'s current support to modularity. Approximately 32.4% were indifferent to it, and the remainder 10.8% showed to be unsatisfied with *MATLAB*'s support to modularity. This suggests that a minority of *MATLAB* users is unsatisfied by *MATLAB*'s modularity and that the majority thinks *MATLAB* programs are sufficiently easy to maintain and that they are scalable, reliable and easily reusable.

***Python* is largely considered to be *MATLAB*'s strongest competitor.**

Approximately 69%, or 127, of the 185 participants that use more than the command window when working with *MATLAB* consider *MATLAB*'s strongest competitor to be *Python*. However, 41% of these 127 participants have stated that they do not use OOP in other languages. This suggests that although many view *Python* as a good alternative to *MATLAB*, considering *MATLAB*'s typical uses, this is not strictly due to better OOP capabilities, but possibly different reasons such as the difference in price and accessibility, or the different tools available for each of these environments, as some participants

highlighted in the comment sections of the posts in which the survey was announced and published.

***MATLAB* is considered to be efficiently scalable, but also easily understood when containing only 1 m-file.**

Additionally, with the data collected we are able to observe that the participants whose programs tend to have between 2 and 10 m-files are also the participants who have the most trouble understanding the code or its structure when maintaining that program. This means that participants whose programs tend to have only 1 m-file or more than 10 m-files are more likely to quickly understand the code and how it is structured. This suggests that the smaller *MATLAB* programs, with only 1 m-file, are easily understood and that *MATLAB* programs are, for the most part, efficiently scalable, as is evidenced by the fact that programs with more than 10 m-files are better understood than the programs with 2 to 10 m-files.

6.7 Conclusion

The process of administration of the study's questionnaire, detailed at the beginning of this Chapter, starts with a post on *Reddit* which includes an introduction to the study and a web address linking to the questionnaire itself. This is followed by a post on multiple other online platforms, such as the *MATLAB Central* forums and *LinkedIn*.

Following the closure of the questionnaire is the verification of the internal consistency of the data collected, which is comprehensively detailed in this Chapter. From this we were led to discard the data from 3 respondents whose answers were inconsistent. Furthermore, we were able to conclude that the general results obtained were consistent with the intended design and structure of the questionnaire, as expected. In addition, a profiling of the participants is also included in this Chapter.

This Chapter also includes the formulation of the 6 null hypotheses tested in this study. Subsequently, it includes the testing of said hypotheses, from which 5 were rejected and 1 was accepted using various different statistical tests (see Appendix C).

Lastly, the results obtained and their implications are also presented in this Chapter. From these results we are able to answer the research questions initially proposed (see Section 1.3).

7

Conclusions

In this Chapter, we first summarise this thesis and its survey research (Section 7.1). Then, we examine the results and contributions brought with this thesis (Section 7.2). Finally, we explore the future work that is possible within this line of study (Section 7.3).

7.1 Summary

One of the main motivations for this research was a way to stratify the community of users of *MATLAB* and its clone languages. Furthermore, we wanted to analyse how users interact with these languages, and more specifically if and how they use OOP in *MATLAB*. Additionally, we wanted to measure the level of satisfaction users feel with *MATLAB*, and more specifically its modularity. There was similar research done on this topic, by Katia Duarte in 2017 for her master thesis, but the results were inconclusive [13] (see Chapter 4). However, a significant portion of the users of *MATLAB* is likely to be unfamiliar with concepts such as code tangling and code scattering, which may have impacted the results of the questionnaire used by Katia Duarte, which was based on those terms. This implication, potentially compounded by the fact that the questionnaire was relatively extensive might have led to the insufficiency of full responses.

Thus, with this thesis we decided to conduct a survey research with an instrument that was as accessible and understandable as possible, for even beginners to be able to provide valuable data. We achieved this through a careful and thorough planning of the structure of questionnaire and its questions (see Chapter 5). Additionally, through branch logic we were able to exclude the less knowledgeable users from the more advanced questions, where these respondents would not be capable of providing valuable data. This allowed us to achieve a satisfactory number of responses as well as a high confidence in the validity of the results, by trying to address each of these concerns. In addition, we were able to obtain valuable responses from users of the clone languages, most notably Octave, further widening the spectrum of communities reached and thus strengthening the resulting stratification of the general community.

Through publications on various online platforms, such as *MATLAB Central*, *Reddit*

and *LinkedIn*, we reached a total of 215 full responses, 3 of which we deemed invalid due to showing inconsistency in their responses (see Section 6.2). In addition, 64 respondents shared their e-mail address in order to receive the results of the survey, and 34 respondents expressed that they would be willing to participate in a future questionnaire and also shared their e-mail address for such.

7.2 Results and Contributions

With this thesis' survey, we were able to conduct an in-depth stratification and demographic analysis of the community of users of *MATLAB* and its clone languages. For instance, 32.08% of the respondents stated that they are students, while 29.25% stated that they're employed by a company and 24.06% stated that they're a researcher. Additionally, we found that *Data Analytics* is the application domain for which the participants use *MATLAB* and its clone languages the most, while domains like *Signal Processing*, *Control Systems*, *Image and Video Processing*, and *Machine Learning* are also prevalent. Additionally, the years of experience with *MATLAB* of the respondents were spread rather evenly. 14.6% stated having less than 1 year of experience, 28.3% between 1 and 4 years, 22.6% between 4 and 7 years, 11.8% between 7 and 10 years, and the remaining 22.6% with over 10 years of experience with *MATLAB*. Furthermore, 25% of the students stated that programming is their primary activity. In contrast, 59.7% of the respondents that are employed by a company consider programming as their primary activity.

The survey also enabled us to provide a better understanding of the other languages this sample of the target population typically uses. In fact, we asked the 212 respondents what language they use the most and they stated that it is *MATLAB*, by 62.74% of the sample, *C*, by 11.79% of the sample, *Python*, by 8.96% of the sample, and *C++*, by 6.60% of the sample. Furthermore, in addition to the language they use the most, an additional 37.74% stated that they use *Python*, 18.40% stated that they use *C++*, 15.09% stated that they use *C*, 7.55% stated that they use *R*, 6.60% stated that they use *C#*, 6.60% stated that they use *Java* and 5.66% stated that they use *Javascript*.

Following this thesis, we also have a better understanding of how common the more superficial use of *MATLAB* is, as in using only the command window. We found that approximately 13% of *MATLAB* users work only with the command window, as opposed to programming using m-files. And this type of usage of the language is not done exclusively by the least experienced. In fact, approximately 33% of those users have more than 10 years of experience with programming.

Additionally, this thesis contributes with a better understanding of how much effort the users put into the reusability of their programs. We found, specifically, that users generally try to make their programs easily reusable and maintainable, and that the users who do not expect to be the sole users of their programs do this to a greater extent. Over half of the respondents stated that they have had to maintain a *MATLAB* program for

a duration of over a year, so this is, and should be, a matter worth considering when working with *MATLAB*.

The thesis also provides an analysis of the different ways users interact with *MATLAB*. For instance, we found that out of the users that work with m-files, and not just the command window, 14.6% tend to develop programs with only 1 m-file, 40.5% develop programs with 2 to 5 m-files, 20% develop programs with 6 to 10 m-files, and the remaining 24.9% develop programs that tend to have more than 10 m-files. Additionally, we found that in each m-file, 24.9% of users tend to write only 1 function, 49.2% tend to write 2 to 5 functions, 8.6% tend to write 6 to 10 functions and 17.3% tend to write more than 10 functions. Furthermore, concerning this population we also found that, in their *MATLAB* programs, 37.3% of programmers use Classes, 20% use Enumerations and 38.4% use Objects. However, we also found a disparity between the users who use Classes and Objects: only 25% use both of these types of module in their *MATLAB* programs, while 11.9% use Classes and not Objects, and another 12.9% use Objects and not Classes.

Ultimately, this thesis also contributes with a better understanding of how mindful the population is of *MATLAB*'s support to modularity. The majority of users, 67%, state that they often think about modularity when they're working with *MATLAB*. 56.8% state that they are satisfied with *MATLAB*'s current support to modularity. However, it is the more experienced *MATLAB* users and the ones who are most mindful of their programs' maintainability and reusability that show the highest level of satisfaction with *MATLAB*'s current modular capabilities.

7.3 Future Work

Although we were capable of drawing conclusions and achieving results with a healthy sample of the initial target population, there is room for further analysis in this line of study. For instance, some smaller communities such as the *Scilab* and *Rlab* user-base were difficult to reach, from whom it would have been beneficial to have gathered more input from. Additionally, we initially didn't consider some communities such as the Julia user-base which turned out to be strongly present in the dataset, despite the lack of specific targeting towards them. This language's user-base could be worth considering for future work on this topic, as it revealed to have some overlap with the community of users of *MATLAB* and its clones.

Concerning the usage of OOP in *MATLAB*, the participants revealed widely different methods that are worthy of further analysis. Specifically, some participants reported that they make use of Classes and not Objects in their *MATLAB* programs, while others reported the complete opposite in which they use Objects but not Classes. This is intriguing and something we did not expect to observe from the data collected, but is something that could be analysed, in future work, whether it is an anomaly in this questionnaire or whether there is actually a relevant portion of the community that uses only one of these two modules, and the reasoning behind it. Although we were capable of analysing

which of these modules *MATLAB* programmers use, we do not know the frequency or the purpose with which they use each one. This would be an interesting aspect to analyse in order to better understand the strengths and weaknesses of each of these modules, as perceived by the community. It could not only provide a stronger understanding of how the users interact with this environment as a whole, one of the goals of this research, but also lead to further questions after analysing those results.

A more focused analysis of the users' opinions on *MATLAB* and its competitors should lead to deeper results in that regard. While we were capable of measuring user satisfaction and analysing which other languages the community deems as strong competitors, we were not able to explore the principles behind those answers. This could potentially lead to further questions concerning possible limitations and improvements for *MATLAB* that weren't possible to explore before due to a lack of awareness to them.

Bibliography

[1] H. Akoglu. "User's guide to correlation coefficients". In: *Turkish journal of emergency medicine* 18.3 (2018), pp. 91–93 (cit. on p. 70).

[2] Alchemer. *Alchemer - Formerly SurveyGizmo*. Last visited on September, 2021. URL: https://www.alchemer.com/ (cit. on p. 38).

[3] Alchemer. *Plans and Pricing - Alchemer*. Last visited on September, 2021. URL: https://www.alchemer.com/plans-pricing/ (cit. on p. 38).

[4] K. G. Calkins. *Correlation Coefficients*. Last visited on September, 2021. URL: https://www.andrews.edu/~calkins/math/edrm611/edrm05.htm (cit. on p. 58).

[5] R. B. Cattell. "The scree test for the number of factors". In: *Multivariate behavioral research* 1.2 (1966), pp. 245–276 (cit. on p. 56).

[6] P. Cretchley et al. "MATLAB in early undergraduate mathematics: An investigation into the effects of scientific software on learning". In: *Mathematics Education Research Journal* 12.3 (2000), pp. 219–233 (cit. on pp. 28, 33).

[7] T. A. Davis. *MATLAB primer*. CRC press, 2010 (cit. on p. 19).

[8] R. F. DeVellis. *Scale development: Theory and applications*. Vol. 26. Sage publications, 2016 (cit. on pp. 58, 60).

[9] S. developers. *SciPy*. Last visited on September, 2021. URL: https://www.scipy.org/ (cit. on p. 29).

[10] Discourse. *Discourse*. Last visited on September, 2021. URL: https://www.discourse.org/ (cit. on p. 49).

[11] Discourse. *GNU Octave Discourse Group*. Last visited on September, 2021. URL: https://octave.discourse.group/t/a-study-on-the-users-of-matlab-and-similar-languages/467 (cit. on p. 49).

[12] R. the Docs Inc. *EthicalAds*. Last visited on September, 2021. URL: https://www.ethicalads.io/?ref=codefund (cit. on p. 30).

[13] K. I. P. Duarte. "Limitations in the support to Modularity in MATLAB: a Survey-based Empirical Study". MSc thesis. Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, 2017 (cit. on pp. 2, 27, 33, 81).

[14] A. G. D. of Education and Training. *StatHand*. Last visited on September, 2021. URL: https://stathand.net/ (cit. on p. 67).

[15] ESI. *ESI group*. Last visited on September, 2021. URL: https://www.esi-group.com/ (cit. on p. 25).

[16] Facebook. *Facebook*. Last visited on September, 2021. URL: https://www.facebook.com/ (cit. on p. 27).

[17] G. Forms. *Google Forms: Free Online Surveys for Personal Use*. Last visited on September, 2021. URL: https://www.google.com/forms/about/ (cit. on pp. 29, 38).

[18] P. S. Foundation. *Welcome to Python.org*. Last visited on September, 2021. URL: https://www.python.org/ (cit. on p. 30).

[19] F. J. Fowler Jr. *Survey research methods*. Sage publications, 2013 (cit. on p. 6).

[20] A. Ganti. *Correlation Coefficient*. Last visited on September, 2021. URL: https://www.investopedia.com/terms/c/correlationcoefficient.asp (cit. on p. 16).

[21] G. V. Glass. "A ranking variable analogue of biserial correlation: Implications for short-cut item analysis". In: *Journal of Educational Measurement* 2.1 (1965), pp. 91–95 (cit. on p. 67).

[22] GNU. *GNU Octave - About*. Last visited on September, 2021. URL: https://www.gnu.org/software/octave/about.html (cit. on pp. 1, 25).

[23] Google. *Google Sheets: Free Online Spreadsheets for Personal Use*. Last visited on September, 2021. URL: https://www.google.com/sheets/about/ (cit. on p. 39).

[24] M. Gwóźdź. *Github - SciPy user survey results*. Last visited on September, 2021. URL: https://github.com/mkg33/GSoD/blob/master/user_survey_summary.pdf (cit. on pp. 29, 33).

[25] A. Hayes. *Chi-Square (2) Statistic Definition*. Last visited on September, 2021. URL: https://www.investopedia.com/terms/c/chi-square-statistic.asp (cit. on p. 16).

[26] L. Hendren. "Typing aspects for MATLAB". In: *Proceedings of the sixth annual workshop on Domain-specific aspect languages*. 2011, pp. 13–18 (cit. on p. 19).

[27] S. R. H. Hoole. "Programming skills in graduate engineering classes: Students from disparate disciplines and eras". In: *The International Journal of Engineering Education* 26.3 (2010) (cit. on pp. 29, 32, 33).

[28] Q. Inc. *Quora*. Last visited on September, 2021. URL: https://quora.com/ (cit. on p. 30).

[29] S. Jamieson. "Likert scales: How to (ab) use them?" In: *Medical education* 38.12 (2004), pp. 1217–1218 (cit. on p. 42).

[30] JetBrains. *Methodology - The State of Developer Ecosystem 2020*. Last visited on September, 2021. URL: https://www.jetbrains.com/lp/devecosystem-2021/methodology/ (cit. on pp. 30, 33).

[31] JetBrains. *Python Developers Survey 2020 - JetBrains*. Last visited on September, 2021. URL: https://www.jetbrains.com/lp/python-developers-survey-2020/ (cit. on pp. 30, 33).

[32] H. F. Kaiser. "An index of factorial simplicity". In: *Psychometrika* 39.1 (1974), pp. 31–36 (cit. on p. 53).

[33] H. F. Kaiser. "The application of electronic computers to factor analysis". In: *Educational and psychological measurement* 20.1 (1960), pp. 141–151 (cit. on p. 55).

[34] M. G. Kendall. "A new measure of rank correlation". In: *Biometrika* 30.1/2 (1938), pp. 81–93 (cit. on p. 50).

[35] B. A. Kitchenham and S. L. Pfleeger. "Personal opinion surveys". In: *Guide to Advanced Empirical Software Engineering*. Springer, 2008, pp. 63–92 (cit. on pp. 6–13, 15).

[36] R. B. Kline. *Principles and practice of structural equation modeling*. Guilford publications, 2015 (cit. on pp. 58, 60).

[37] M. Kostrun. *Rlabplus*. Last visited on September, 2021. URL: http://rlabplus.sourceforge.net/ (cit. on p. 25).

[38] A. M. Liebetrau. *Measures of association*. Vol. 32. Sage, 1983 (cit. on p. 51).

[39] LinkedIn. *About LinkedIn*. Last visited on September, 2021. URL: https://about.linkedin.com/ (cit. on p. 36).

[40] LinkedIn. *GNU Octave users and developers*. Last visited on September, 2021. URL: https://www.linkedin.com/groups/4044339/ (cit. on p. 37).

[41] LinkedIn. *GNU Octave users and developers*. Last visited on September, 2021. URL: https://www.linkedin.com/feed/update/urn:li:activity:6729013268859367424/ (cit. on p. 48).

[42] LinkedIn. *LinkedIn*. Last visited on September, 2021. URL: https://linkedin.com/ (cit. on pp. 27, 36).

[43] LinkedIn. *Matlab beginners and experts*. Last visited on September, 2021. URL: https://www.linkedin.com/feed/update/urn:li:activity:6729011554890588160/ (cit. on p. 48).

[44] LinkedIn. *Matlab for beginners and experts*. Last visited on September, 2021. URL: https://www.linkedin.com/groups/1843503/ (cit. on p. 36).

[45] LinkedIn. *MATLAB Users and Integrators*. Last visited on September, 2021. URL: https://www.linkedin.com/groups/134533/ (cit. on p. 36).

[46] LinkedIn. *MATLAB Users and Integrators*. Last visited on September, 2021. URL: https://www.linkedin.com/feed/update/urn:li:activity:6729011031722496000/ (cit. on p. 48).

[47] LinkedIn. *Scilab Software*. Last visited on September, 2021. URL: https://www.linkedin.com/groups/3688414/ (cit. on p. 37).

[48] LinkedIn. *Scilab Software*. Last visited on September, 2021. URL: https://www.linkedin.com/feed/update/urn:li:activity:6729012464039559168/ (cit. on p. 48).

[49] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/master/template.pdf (cit. on p. ii).

[50] MathWorks. *A Brief History of MATLAB*. Last visited on September, 2021. URL: https://www.mathworks.com/company/newsletters/articles/a-brief-history-of-matlab.html (cit. on p. 19).

[51] MathWorks. *A study on the users of MATLAB and similar languages*. Last visited on September, 2021. URL: https://www.mathworks.com/matlabcentral/answers/633854-a-study-on-the-users-of-matlab-and-similar-languages (cit. on p. 48).

[52] MathWorks. *About Us*. Last visited on September, 2021. URL: https://www.mathworks.com/company/aboutus.html (cit. on pp. 1, 19).

[53] MathWorks. *Define Enumeration Classes*. Last visited on September, 2021. URL: https://www.mathworks.com/help/matlab/matlab_oop/enumerations.html (cit. on p. 24).

[54] MathWorks. *Inside MATLAB Objects in R2008a*. Last visited on September, 2021. URL: https://www.mathworks.com/company/newsletters/articles/inside-matlab-objects-in-r2008a.html (cit. on p. 2).

[55] MathWorks. *Introduction to Object-Oriented Programming in MATLAB*. Last visited on September, 2021. URL: https://www.mathworks.com/company/newsletters/articles/introduction-to-object-oriented-programming-in-matlab.html (cit. on pp. 22–24).

[56] MathWorks. *Language Fundamentals*. Last visited on September, 2021. URL: https://www.mathworks.com/help/matlab/language-fundamentals.html (cit. on p. 21).

[57] MathWorks. *Math. Graphics. Programming*. Last visited on September, 2021. URL: https://www.mathworks.com/products/matlab.html?s_tid=hp_products_matlab (cit. on p. 1).

[58] MathWorks. *MathWorks - Makers of MATLAB and Simulink*. Last visited on September, 2021. URL: https://www.mathworks.com/ (cit. on pp. 19, 36).

[59] MathWorks. *MATLAB Central - About*. Last visited on September, 2021. URL: https://www.mathworks.com/matlabcentral/about.html?s_tid=gn_mlc_about (cit. on pp. 27, 36).

[60] MathWorks. *MATLAB Central - About File Exchange*. Last visited on September, 2021. URL: https://www.mathworks.com/matlabcentral/about/fx/ (cit. on p. 36).

[61] MathWorks. *Object-Oriented Programming in MATLAB*. Last visited on September, 2021. URL: https://www.mathworks.com/discovery/object-oriented-programming.html (cit. on p. 22).

[62] MathWorks. *Surveying the MATLAB community*. Last visited on September, 2021. URL: https://www.mathworks.com/matlabcentral/answers/541361-surveying-the-matlab-community?s_tid=prof_contriblnk (cit. on p. 37).

[63] R. .-. MATLAB. *A study on the users of MATLAB and similar languages*. Last visited on September, 2021. URL: https://www.reddit.com/r/matlab/comments/jfy9vn/a_study_on_the_users_of_matlab_and_similar/ (cit. on p. 47).

[64] R. .-. MATLAB. *Surveying the MATLAB community*. Last visited on September, 2021. URL: https://www.reddit.com/r/matlab/comments/hj91dx/surveying_the_matlab_community/ (cit. on p. 37).

[65] J. S. Molléri, K. Petersen, and E. Mendes. "An empirically evaluated checklist for surveys in software engineering". In: *Information and Software Technology* 119 (2020), p. 106240 (cit. on p. 12).

[66] R. Newson. "Confidence intervals for rank statistics: Somers' D and extensions". In: *The Stata Journal* 6.3 (2006), pp. 309–334 (cit. on p. 72).

[67] V. Nijs. *Compare a single proportion to the population proportion*. Last visited on November, 2021. URL: https://radiant-rstats.github.io/docs/basics/single_prop.html (cit. on p. 67).

[68] P. Prabhu et al. "A survey of the practice of computational science". In: *SC'11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE. 2011, pp. 1–12 (cit. on pp. 31, 33).

[69] Qualtrics. *Online Survey Software - Qualtrics*. Last visited on September, 2021. URL: https://www.qualtrics.com/uk/core-xm/survey-software/ (cit. on pp. 27, 38).

[70] Reddit. *Homepage - Reddit*. Last visited on September, 2021. URL: https://www.redditinc.com/ (cit. on p. 36).

[71] Reddit. *Reddit - EngineeringStudents*. Last visited on September, 2021. URL: https://www.reddit.com/r/EngineeringStudents/ (cit. on p. 36).

[72] Reddit. *Reddit - The GNU Octave Subreddit*. Last visited on September, 2021. URL: https://www.reddit.com/r/octave/ (cit. on p. 36).

[73] Reddit. *reddit: the front page of the internet*. Last visited on September, 2021. URL: https://www.reddit.com/ (cit. on p. 36).

[74] ResearchGate. *A study on the users of MATLAB and similar languages*. Last visited on September, 2021. URL: https://www.researchgate.net/post/A_study_on_the_users_of_MATLAB_and_similar_languages (cit. on p. 48).

[75] ResearchGate. *ResearchGate*. Last visited on September, 2021. URL: https://www.researchgate.net (cit. on p. 36).

[76] ResearchGate. *ResearchGate - About*. Last visited on September, 2021. URL: https://www.researchgate.net/about (cit. on p. 36).

[77] ResearchGate. *Surveying the MATLAB community*. Last visited on September, 2021. URL: https://www.researchgate.net/post/Surveying_the_MATLAB_community (cit. on p. 37).

[78] E. Rietsch. *Scilab - from a Matlab User's Point of View*. Last visited on September, 2021. URL: https://wiki.scilab.org/Tutorials?action=AttachFile&do=get&target=Scilab4Matlab.pdf (cit. on p. 25).

[79] J. Rosenberg. "Statistical methods and measurement". In: *Guide to Advanced Empirical Software Engineering*. Springer, 2008, pp. 155–184 (cit. on pp. 14–17).

[80] Scilab. *Scilab - About*. Last visited on September, 2021. URL: https://www.scilab.org/about/scilab-open-source-software (cit. on pp. 1, 25).

[81] Scilab. *Scilab - Community*. Last visited on September, 2021. URL: https://www.scilab.org/about/community (cit. on p. 25).

[82] Scilab. *Scilab - Company*. Last visited on September, 2021. URL: https://www.scilab.org/about/company (cit. on p. 25).

[83] I. Searle. *Rlab Web Site*. Last visited on September, 2021. URL: http://rlab.sourceforge.net/ (cit. on pp. 1, 25).

[84] N. Sharma and M. K. Gobbert. "A comparative evaluation of Matlab, Octave, FreeMat, and Scilab for research and teaching". In: *UMBC Faculty Collection* (2010) (cit. on p. 25).

[85] R. H. Somers. "A new asymmetric measure of association for ordinal variables". In: *American sociological review* (1962), pp. 799–811 (cit. on p. 72).

[86] L. Statistics. *Kendall's Tau-b using SPSS Statistics*. Last visited on September, 2021. URL: https://statistics.laerd.com/spss-tutorials/kendalls-tau-b-using-spss-statistics.php (cit. on p. 50).

[87] L. Statistics. *Principal Components Analysis (PCA) using SPSS Statistics*. Last visited on September, 2021. URL: https://statistics.laerd.com/spss-tutorials/principal-components-analysis-pca-using-spss-statistics.php (cit. on pp. 49, 52).

[88] R. .-. E. Students. *Surveying the MATLAB community*. Last visited on September, 2021. URL: https://www.reddit.com/r/EngineeringStudents/comments/hj9n6t/surveying_the_matlab_community/ (cit. on p. 37).

[89] SurveyLegend. *Plans for Surveys, Forms and Polls - SurveLegend*. Last visited on September, 2021. URL: https://www.surveylegend.com/pricing/ (cit. on p. 38).

[90] SurveyLegend. *SurveyLegend*. Last visited on September, 2021. URL: https://www.surveylegend.com/ (cit. on p. 38).

[91] SurveyMonkey. *SurveyMonkey*. Last visited on September, 2021. URL: https://surveymonkey.com/ (cit. on p. 38).

[92] SurveyMonkey. *SurveyMonkey Plans and Pricing*. Last visited on September, 2021. URL: https://surveymonkey.com/pricing/individual/details/ (cit. on p. 38).

[93] SurveyMonkey. *What is a Likert scale?* Last visited on September, 2021. URL: https://www.surveymonkey.com/mp/likert-scale/ (cit. on p. 10).

[94] Twitter. *Twitter*. Last visited on September, 2021. URL: https://twitter.com/ (cit. on p. 29).

[95] Typeform. *Typeform*. Last visited on September, 2021. URL: https://www.typeform.com/ (cit. on p. 38).

[96] Typeform. *Typeform - Free Plan*. Last visited on September, 2021. URL: https://help.typeform.com/hc/en-us/articles/360032972852-Free-plan (cit. on p. 38).

[97] D. A. Walker. "JMASM9: converting Kendall's tau for correlational or meta-analytic analyses". In: *Journal of Modern Applied Statistical Methods* 2.2 (2003), p. 26 (cit. on p. 51).

[98] H. P. Wallin et al. "Learning MATLAB: Evaluation of methods and materials for first-year engineering students". In: *International journal of engineering education* 21.4 (2005), p. 692 (cit. on pp. 29, 33).

[99] C. Wohlin et al. *Experimentation in software engineering*. Springer Science & Business Media, 2012 (cit. on pp. 12, 40, 41, 44, 66).

[100] Zenodo. *Zenodo - Research. Shared.* Last visited on September, 2021. URL: https://zenodo.org/ (cit. on pp. 38, 39).

[101] Zenodo. *Zenodo - Surveying the communities of users of MATLAB and similar languages (Responses).* Last visited on September, 2021. URL: https://zenodo.org/record/5006428 (cit. on p. 39).

A

Surveying the communities of users of MATLAB and similar languages

This appendix includes the entirety of the questionnaire used in this research study, including its introductory text, the questions, the question descriptions, and the response options to each question.

MATLAB usage and practices

The purpose of this survey is to collect feedback from users of MATLAB and similar languages (Octave, Scilab, Rlab, etc.) regarding how these languages are being used.

Your participation in this research study is voluntary. If you decide to participate, you may withdraw at any time.

We will keep your information confidential, with all data being stored in a password protected electronic format. Additionally, the surveys will not contain any information that could personally identify you. The results of this study will only be used for scholarly purposes and may be shared in papers of the specialty.

For any questions you might have concerning this survey, please do not hesitate to contact us via email at er.reis@campus.fct.unl.pt.
*Required

1. Where did you hear about this survey? *

Mark only one oval.

- E-mail
- LinkedIn
- MATLAB Central
- Reddit
- ResearchGate
- GNU Octave Discourse
- Word of mouth

This section concerns your programming experience.

2. Programming is my primary professional activity. *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree						Strongly agree

3. What status do you currently identify with the most? *

Mark only one oval.

- ◯ Fully employed by a company / organisation
- ◯ Partially employed by a company / organisation
- ◯ Self-employed (earning income directly from their own business or trade)
- ◯ Freelancer (pursuing a profession without a long-term commitment to an employer)
- ◯ Teacher
- ◯ Student
- ◯ Researcher
- ◯ Retired
- ◯ Other

4. How many years of experience do you have with programming in general? *

Mark only one oval.

- ◯ Less than 1 year
- ◯ Between 1 and 4 years
- ◯ Between 4 and 7 years
- ◯ Between 7 and 10 years
- ◯ More than 10 years

5. How many years of experience do you have with MATLAB or a similar language? *

Mark only one oval.

- ◯ Less than 1 year
- ◯ Between 1 and 4 years
- ◯ Between 4 and 7 years
- ◯ Between 7 and 10 years
- ◯ More than 10 years

6. Which of the following programming languages (MATLAB and similar languages) do you use? *

Select one or more options.

Tick all that apply.

- [ ] MATLAB
- [ ] Octave
- [ ] Scilab
- [ ] RLab
- [ ] Other: _____

7. Rate your level of experience with MATLAB. *

Mark only one oval.

	1	2	3	4	5	
Trainee	◯	◯	◯	◯	◯	Expert

8. What programming language do you use the most? *

_____

9. Rate your level of experience with the programming language you use the most. *

Mark only one oval.

	1	2	3	4	5	
Trainee	◯	◯	◯	◯	◯	Expert

10. What other programming languages do you use, if any?

_____

11. For what do you use MATLAB or similar languages? *

Select one or more options.

Tick all that apply.

- [ ] Data Analytics
- [ ] Machine Learning
- [ ] Signal Processing
- [ ] Wireless Communications
- [ ] Image and Video Processing
- [ ] Control Systems
- [ ] Computational Finance
- [ ] Computational Biology
- [ ] Other: _____

12. The last time I programmed in MATLAB or a similar language was... *

Mark only one oval.

- ◯ less than 1 year ago.
- ◯ between 1 to 4 years ago.
- ◯ over 4 years ago.

13. On which operating systems are your development environments? *

Select one or more options.

Tick all that apply.

- [ ] Windows
- [ ] Unix / Linux
- [ ] macOS
- [ ] Other: _____

14. I use only the command window when working with MATLAB. *

As opposed to working with executable text files containing code.

Mark only one oval.

- ◯ True, I use only the command window. *Skip to question 27*
- ◯ False, I use the command window to solve small problems or to complement my coding (e.g. inspecting variables, testing functions, etc.).
- ◯ False, I never use the command window.

This section focuses on the importance you give to the reusability of your programs.

15. When I develop a program in MATLAB, I always try to make it easily reusable and maintainable. *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

16. I expect to be the sole user of my MATLAB programs. *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

17. In the past, I've had to maintain a MATLAB program for over a year. *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

18. My MATLAB programs tend to have... *

Mark only one oval.

- only 1 m-file.
- 2 to 5 m-files.
- 6 to 10 m-files.
- 11 to 20 m-files.
- 21 to 50 m-files.
- more than 50 m-files.

19. The m-files I deal with tend to have... *

Mark only one oval.

- only 1 function.
- 2 to 5 functions.
- 6 to 10 functions.
- more than 10 functions.

This final section concerns your use of the language, as well as your level of satisfaction with that use.

20. Regarding MATLAB's modules, in my programs I use...

Modules are the distinct units that enclose the code related to a specific functionality.

Tick all that apply.

- [ ] Classes
- [ ] Enumerations
- [ ] Functions
- [ ] M-files
- [ ] Objects

21. I try to find and minimise the use of duplicated code across the various m-files. *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

22. I often think about modularity when I'm working with MATLAB. *

Modularity is the approach of organizing a program into multiple modules (e.g., m-files, functions, classes or objects).

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

23. I am satisfied with MATLAB's current support to modularity. *

Modularity is the approach of organizing a program into multiple modules (e.g., m-files, functions, classes or objects).

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

24. When maintaining a MATLAB program, I don't have any trouble understanding the code or its structure. *

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

25. Regarding typical uses of MATLAB, I consider MATLAB's strongest competitor to be... *

Mark only one oval.

- ◯ Octave
- ◯ SciLab
- ◯ RLab
- ◯ Python
- ◯ R
- ◯ Other: _____

26. I use object-orientation features when programming in other languages. *

Languages that are not MATLAB, Rlab, Scilab or Octave.

Mark only one oval.

- ◯ True
- ◯ False

Thank you for your contribution!

27. If you wish to receive the results of the study via e-mail, feel free to leave your email address in the text field below.

_____

28. If you wish to participate in a future questionnaire surrounding this research, feel free to leave your email address in the text field below.

_____

B

Variable labels

This appendix contains a Table which presents the variables created for and used in the Data Analysis stage of this research study (Chapter 6), as well as the questions they correspond to in the questionnaire (see Table B.1).

Table B.1: Data variables and corresponding questions.

Variable	Question	Question number
Q2	Programming is my primary professional activity.	2
Q4	Years of experience with programming.	4
Q5	Years of experience with MATLAB or similar languages.	5
Q7	Level of expertise with MATLAB.	7
Q9	Level of expertise with language I use the most.	9
Q12	Last time I programmed in MATLAB or a similar language.	12
Q15	When I develop a program in MATLAB, I always try to make it easily reusable and maintainable.	15
Q16	I expect to be the sole user of my MATLAB programs.	16
Q17	In the past, I've had to maintain a MATLAB program for over a year.	17
Q18	Number of m-files my programs tend to have...	18
Q19	The m-files I deal with tend to have...	19
Q21	I try to find and minimise the use of duplicated code across the various m-files.	21
Q22	I often think about modularity when I'm working with MATLAB.	22
Q23	I am satisfied with MATLAB's current support to modularity.	23
Q24	When maintaining a MATLAB program, I don't have any trouble understanding the code or its structure.	24

C

Statistical tests

This appendix contains a Table which presents the tests used to test the null hypotheses in the Data Analysis stage of this research study (Chapter 6), as well as the verdict of each of these tests (see Table C.1).

Table C.1: Statistical tests summary.

Null Hypothesis	**Test used**	**Verdict**
1	Rank-biserial Correlation	Rejected
2	Spearman's Correlation	Rejected
3	Spearman's Correlation	Accepted
4	Somers' d	Rejected
5	Somers' d	Rejected
6	Somers' d	Rejected