7-2017

MatLab vs. Python vs. R

Taylor Colliau
Valparaiso University

Grace Rogers
Valparaiso University

Z. Hughes

Ceyhun Ozgur
Valparaiso University

Follow this and additional works at: https://scholar.valpo.edu/cba_fac_pub

Recommended Citation

Colliau, Taylor; Rogers, Grace; Hughes, Z.; and Ozgur, Ceyhun, "MatLab vs. Python vs. R" (2017). *Business Faculty Publications*. 51.
https://scholar.valpo.edu/cba_fac_pub/51

MatLab vs. Python vs. R

Ceyhun Ozgur[1],Taylor Colliau[2],Grace Rogers[3]
Zachariah Hughes[4],Elyse "Bennie" Myer-Tyson[5]

[12345]*Valparaiso University*

Abstract: Matlab, Python and R have all been used successfully in teaching college students fundamentals of mathematics & statistics. In today's data driven environment, the study of data through big data analytics is very powerful, especially for the purpose of decision making and using data statistically in this data rich environment. MatLab can be used to teach introductory mathematics such as calculus and statistics. Both Python and R can be used to make decisions involving big data. On the one hand, Python is perfect for teaching introductory statistics in a data rich environment. On the other hand, while R is a little more involved, there are many customizable programs that can make somewhat involved decisions in the context of prepackaged, preprogrammed statistical analysis.

Key words: MatLab, Python, R

1. INTRODUCTION

This paper compares the effectiveness of MatLab, Python (Numpy, SciPy) and R in a teaching environment. In this paper we have tried to establish which programming language is best to teach operations research and statistics to students in a college setting. We have also attempted to determine which skill is most desirable to have knowledge about in the workplace.

To begin, Python is a type of programming language. The most common implementation to this programing language is that in C (also known as CPython). Not only is Python a programming language, but it consists of a large standard library. This library is structured to focus on general programming and contain modules for os specific, threading, networking, and databases.

Next, Matlab is most highly regarded as not only a commercial numerical computing environment, but also as a programming language. Matlab similarly has a standard library, but its uses include matrix algebra and a large network for data processing and plotting. It also contains toolkits for the avid learner, but these will cost the user extra.

Lastly, R is a free, open-source statistical software. Colleagues at the University of Auckland in New Zealand, Robert Gentleman and Ross Ihaka, created the software in 1993 because they mutually saw a need for a better software environment for their classes. R has

certainly outgrown its origins, now boasting more than two million users according to an R Community website ("What is R?" 2014).

Although both Python and R are open source programming languages, you do not have to be a programmer to utilize them. While programs such as Excel and SPSS may be simpler and faster to learn, their computational abilities are far inferior to those of Python, R, and Matlab, which require only basic programming knowledge. Between these three programs, when it comes to usability, Python may be a better choice because the syntax it uses compares more similarly to other languages. However, many programmers believe the syntax used by R to be easily learned and understood without explicit instruction. Kevin Markham, a data scientist and teacher, suggested in his article on software learnability that Python and R have comparable learning curves for students without any prior programming experience. Despite this similarity, there is an argument that Python can be easier to learn because its code is read more like regular human language (Markham). There is a tradeoff between the simplicity of closed-source preprogrammed software and the more complicated yet empowering open-source software. If all you require is straightforward, small data analysis, you may not need to look any further than Excel and SPSS. The extent of Python, Matlab, and R, however, reaches numerous additional dimensions of big data analysis capability. Universities may wish to pursue offering instruction on these programs as they are better suited for working with big data and more widely used in the workplace.
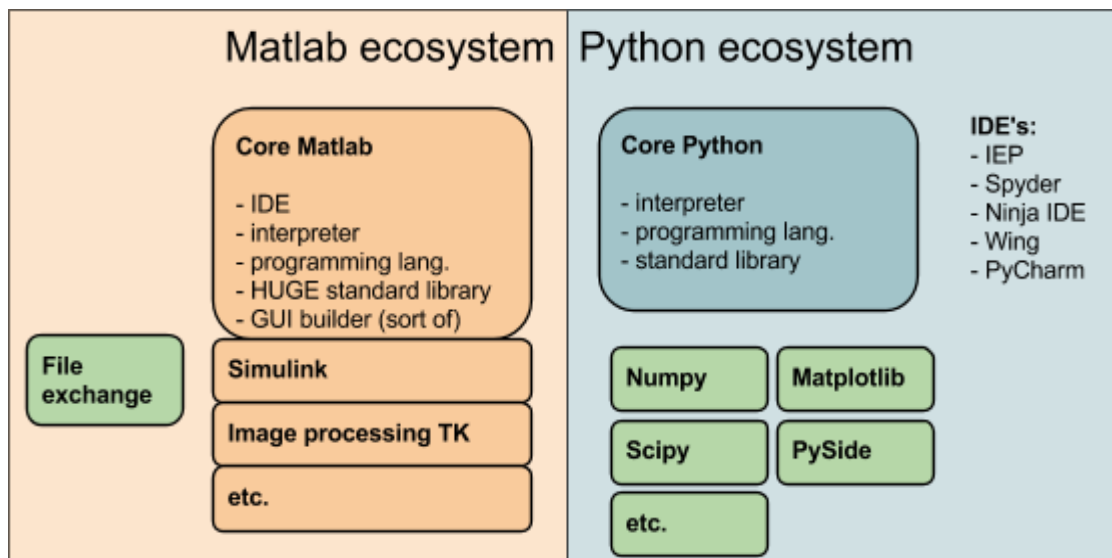
2. MATLAB VS. PYTHON

Figure 1: MatLab vs. Python

3. Basics of MatLab

MatLab is a programming language used mostly by engineers and data analysts for numerical computations. There are a variety of toolboxes available when first purchasing Matlab to further enhance the basic functions that are already available upon purchase. Matlab is available on Unix, Macintosh, and Windows environments, but is also available for student usage on personal computers.

The largest advantage of MatLab is the fact that it makes data visualization so easy for the user. Rather than relying on some foundational knowledge of coding or computer science policies, MatLab instead allows users to get a more intuitive read on their data. The easy-to-read format in which data are presented both pre- and post-analysis allows users to maintain tight control over the way their data are presented. This is perhaps most important for those who may not have advanced statistical backgrounds; one need not understand the finer points of a procedure if the changes are made readily apparent in a table or a graph. This also means that MatLab is able to produce analyses which are friendly to the layperson and may make large-scale data analysis handier for business which deal with it.

The benefit of this visualization is not limited to the two-dimensional nature in which other programs, such as SPSS or SAS, and therefore allows for a better conceptualization of the data in a real-world scenario. Rather than attempting to force multiple factors into several complex models in order to understand how each one interacts with the data, one can look at the rises and falls in the output in a manner that logically follows given how we live our lives in a three-dimensional space. This continues to add to the appeal of MatLab, as this further enhances the readability of analyses and makes businesses more approachable by allowing for output that those without strong statistics backgrounds can interpret.

That said, MatLab does fall into the trap of being somewhat particular in the way that data must be read in and commands must be executed. This is a somewhat expected problem, as software that tends to be more open-code is less layperson-friendly. Therefore, while this is a downfall of directly working with MatLab, the benefits concerning the way data are presented should not be ignored.

4. Basics of Python

Python is another available programming language that can be accessed and used easily by the most experienced programmers, but also by novice students. Python is a programming language that can be used for both major and smaller projects. This is due to its adaptable and being a well-developed programming language. It is a widely used program due to its efficient nature of programming features. Python has also simplified debugging for the programmer due to its built-in debugging feature. Python has ultimately helped programmers become more productive and efficient with their time and has made their developments better.

Python is also one of the top coding languages, as of 2014 (Guo, 2014). This language is required, or at least used, by the overwhelming majority of computer science courses in United States colleges. This ubiquity means that learning Python is almost essential if one wishes to

pursue any degree which requires some fundamental knowledge of coding and/or computer science practices, and especially so for those looking to start a career in data analytics. The prevalence of Python in so many programs nationwide means that those who are concerned about the applicability of their skills need not worry, as it is found in the majority of the top programs and has a basis in the job market post-graduation (Guo, 2014). Python is likely to make a lasting impression on those who learn it as a coding language, either because it is their first or because it is the first one they learned at a higher-level institution. Even if students do not go on to complete their initial major choice, or if they learn other coding languages (or have learned them in the past, even), the fact of the matter is, this language will be the one they come to assign a higher value to due to its weight and influence as a college-level experience.

5. Advantages of Python

Using Python has many advantages to the programmers. The first is that Python is free to the public and to anyone who wants to use the program. This gives an advantage because it allows anyone who has the motivation to learn the program to use it as they please.

It is also an easy program to learn and to read. It is much more generic than Matlab which originally started as a matrix manipulation package that later added a programming language to it. Python is also much easier to make your original ideas into a coding language. With this free program it comes with libraries, lists, and dictionaries that will help the programmer achieve their ultimate goal in a well-organized way. It is used by working with a variety of modules, which allows it to start up very quickly. When using Python it is soon realized that everything is an object, so each object has a namespace itself. This helps give the program structure while keeping it clean and simple. This is why Python excels at introspection. Introspection is what comes from the object nature of Python. Due to Python's easy and clear structure mentioned earlier, introspection is easy to do on this program. This is key in being able to access any part of the program, including Python's internal structures. String manipulation is also simple, easy, and efficient when using Python. Due to Python being virtually available to everyone because of its free of cost nature, it can run on any type of system. These include: Windows, Linux, and OS X. On Python functions and classes can be defined and used wherever the user would like and programmers can design as many as they deem necessary. With Python a user can personally create an application that they think looks good and works well for them. A programmer can choose from a variety of the available GUI (graphical user interface) toolkits.

6. Advantages of Python over Matlab

As one who has become thoroughly familiar with the range of both Matlab's and Python's capabilities through years of use, Phillip Feldman offers the following reasons as to why the qualities of Python are advantageous to those of Matlab despite the their numerous comparable qualities.

(1) Python code is more compact and easier to read than Matlab code.

a. Unlike Matlab, which uses end statement to indicate the end of a block, Python determines block size based on indentation.

b. Python uses square brackets for indexing and parentheses for functions and methods, whereas Matlab uses parentheses for both, making Matlab more difficult to differentiate and understand.

c. Python's better readability leads to fewer bugs and faster debugging.

(2) While most programming languages, including Python, use zero-based indexing, Matlab uses one-based indexing making it more confusing for users to translate.

(3) The object-oriented programming (OOP) in Python is simple flexibility while Matlab's OOP scheme is complex and confusing.

(4) Python is free and open.

a. While Python is open source programming, much of Matlab is closed

b. The developers of Python encourage users to input suggestions for the software, while the developers of Matlab offer no such interaction.

(5) There is no Matlab counterpart to Python's import statement.

(6) Python offers a wider set of choices in graphics package and toolsets

7. Utilization of Python

Python has been gaining momentum as being the programming language for novice users. Highly ranked Computer Science departments at MIT and UC Berkeley use Python to teach their novice programming language students. The three largest Massive Open Online Course (MOOC) providers (edX, Coursera, and Udacity) all use Python as their programming language for their beginning courses in programming. A variety of professors in other disciplines now utilize the need for novice students to understand Python and its key features.

8. Analysis for Python vs. Matlab

The graph below (Figure 2) accurately shows the top 39 computer science departments that use introductory languages in their curriculum. The seven introductory languages evaluated were Python, Java, Matlab, C, C++, Scheme, and Scratch. The two that we will be concentrating on are Python and Matlab. Sometimes, we hear the comparison between Python and MATLAB. This comparsion can especially be picked up from http://www.pyzo.org/python_vs_matlab.html.

This article shows Python more favorable than MATLAB. However, the following article showsMATLAB more favorable than Python; https://www.mathworks.com/products/matlab/matlab-vs- python.html
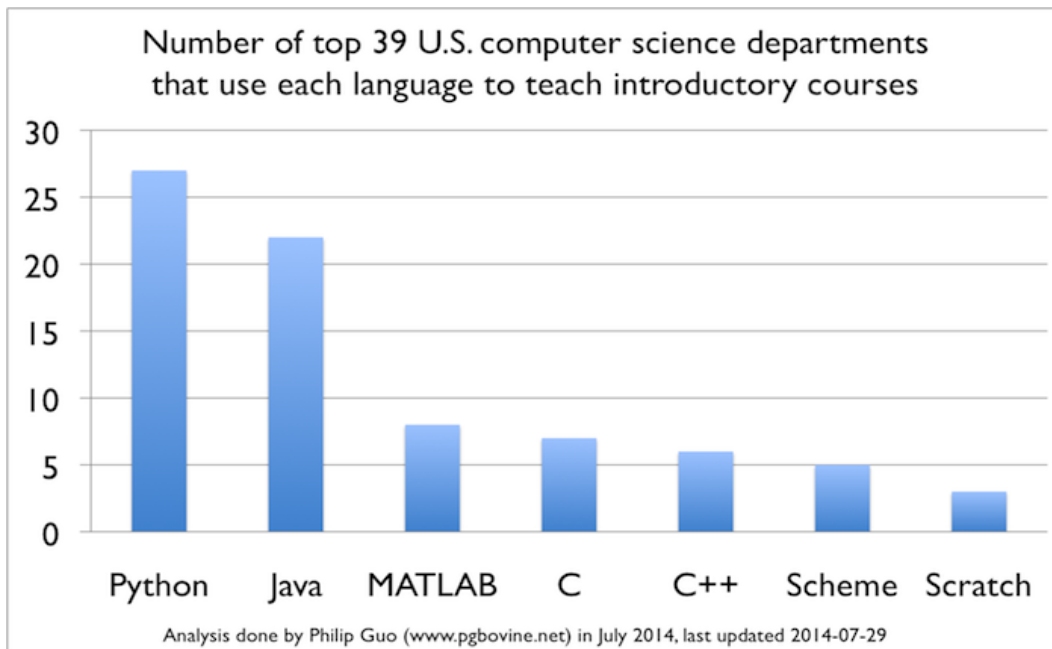
Figure 2:

9. Comparison of Python to other programming languages

Python is clearly the most popular introductory language that was being taught, from the selection on this list. It surpassed Java, that was until recently the most used introductory teaching language over the past decade. Python has been added to most schools teaching curriculum due to its easy to learn and use programs and features. With Python, beginning students do not have to focus their energies on details like types, compilers, and writing boilerplate code, and other algorithms. Python allows the students to easily code the and make the program accomplish the tasks that they want to see achieved.

Matlab was the next most popular programming language after Python and Java. It was mostly entered into the curriculum for science and engineering. This is due to its more advanced features and language characteristics.

10. R VS. PYTHON

The Basics of R

R is software designed to run statistical analyses and output graphics. It can run on virtually any operating system, and is open source (The R Foundation, 2017). This availability across operating systems does put it on par with MatLab for its general flexibility. However, R excels

at having community-driven discussions to find new ways in which to utilize the program as well as to debug potential coding errors. This community support largely stems from the fact that R, like Python, is an open-source and free-to-use software. However, it shares the same issue that Python does; it does not necessarily present data in an easy-to-interpret way for the layperson.

The analytical power of R is virtually unmatched; one of the strongest competitors, SAS, requires three programming languages to accomplish the same tasks that R can do with one (Ooi, 2016).. This makes it easier to teach as well as learn, as one need not commit as much time to the intricate details of each language. R is a highly flexible software with many additional features that can be downloaded in coding packages (Ooi, 2016).. These packages come as updates both from R itself and as user-created code packages. The feedback process is, therefore, reinforced as one is able to network with others in the same or different fields in order to get software that optimally interacts with data. The user-contributed content can aid in finding the appropriate package. R is also becoming more and more able to handle the strains of large datasets; therefore it will be even more useful in the future than it is now (Ooi, 2016). This is paramount as we move further into the digital age, where one consumer alone can produce vast amounts of data that are usable to many businesses in several different models. Therefore, the ability of R to continue to meet the demands that growing data trends place on it, makes this a highly appealing software for those who want to either acquire or maintain a presence in the large-scale analytics field.

R as a software, and as a program utilized by businesses, focuses on the analytics-side of the equation, rather than on the readability of the data (Ooi, 2016). While this is not necessarily an advantage (if one compares to MatLab), this may work in R's favor as a popular software for analysts. The focus on running the analyses and on keeping the data integrity at the forefront, means that one is able to produce better-quality results and forecasts with R than one might get from other software. Additionally, this also means that those who are less concerned with coding and more concerned with strictly producing data models may find this an easier software with which to work.

11. Comparison of R to Python

When beginning to use R the programmer reads their data into a data frame, used a built-in model by using R's formula language, and then can later look back at the model summary output. When getting started with Python, the programmer has many more choices to make. These can include choosing how they would like to read their data, what kind of structure they should use to store their data in, what machine learning package they should use, and what type of objects does the package even allow to be in the input. Other concerns for the programmer when starting Python could include what shape should the previous talked about objects be in, how does the programmer include categorical variables, and how does the user even access the model's output? There are many beginning questions for Python because it is a general purpose programming language, On the other hand, R specializes in a smaller subset of statistical data and tasks so it is much easier for a programmer to get started. There have been many

comparisons in the literature between R and Python. Interested readers can find most of the important conversation/comparison from

http://www.kdnuggets.com/2015/12/continuum-beautiful-interactive-data-visualizations-webinar.html.

In addition, some of the important conversation/comparison can also be found at https://www.datacamp.com/community/tutorials/r-or-python-for- data-analysis#gs.a5OeRpc.

Guo (2014) makes the case that Python is necessarily superior due to how commonplace it is. However, a 2017 update from Muenchen, notes that R is rapidly gaining a share of the data analytics market. Figure 3 presents a look at the changing landscape of the job postings available for different types of data analytics software (R Bloggers, 2017).

R expanded from being required in 7% of the jobs posted in 2014 to 11% in 2017. While a four-percent shift may not seem as though it is worth noting, the fact that this was accomplished within just three years suggests that this is a trend which not only may continue, but may do so at a highly accelerated rate.
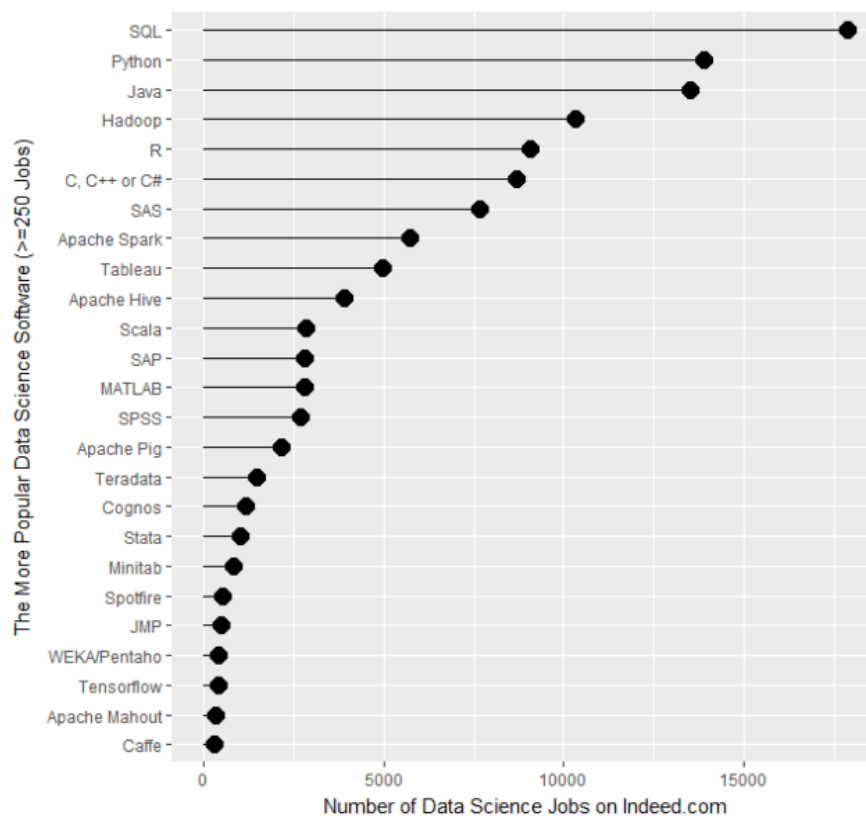

Figure 3: Job Postings for Popular Data Science Software, 2017 (R Bloggers, 2017)

While Python maintains a strong hold on the market (at roughly 15,000 job postings in 2017) (R Bloggers, 2017), it should be noted that R is rapidly decreasing the margin of preference within the data analytics field (with 9,000 in the same time frame) (R Bloggers, 2017). This is likely due to the fact that R resembles SAS strongly and may not strictly rely on a background with computer science or coding. Additionally, the fact that it is open source could continue to drive its popularity, although that would not wholly account for the shift from Python to R (since they are both free-to-use software).

It should also be noted that R itself focuses heavily on the data analytics applications of the software. These jobs, therefore, are more likely to appeal to those who want to use statistical analyses within their work. Python-based jobs are more likely to focus on the coding aspects of the language, and therefore may not strictly appeal to those in statistics programs or to businesses that are involved in large-data analytics (R Bloggers, 2017).

12. R VS. MATLAB

We often hear people complain how expensive MATLAB licenses are. Then we wonder why they don't just use R. We can use R to replace MATLAB.

13. Advantages of Matlab

MatLab has a large number of committed users which include many universities and a few companies who have the budget to buy a license for the program. Even though it is used in many universities, MatLab is easy for beginners who are just starting to learn about programming language because the package, when purchased, includes all that you will need. When using Python you are required to install extra packages. One part of MatLab is a product called Simulink, which is a core part of the MatLab package for which there does not yet exist a good alternative in other programming languages.

14. Advantages of R

R is a statistical package that tries to solve problems of statistics in nature. There are many prepackaged programs in R that attempt to solve various analytics problems. However, MatLab is used to teach various aspects of mathematics, such as calculus or graphing equations. As soon below in Figure 3, R is currently widely used for data mining today. Out of those polled, 47% indicated that they used R on a daily basis as opposed to only 23% that said that they used MatLab. This shows that in the analytics field, R is preferred over MatLab when it comes to performing statistical analysis. This claim can be supported below in Figure 4 where approximately 1,600 jobs on Indeed.com required the knowledge of R, and the program MatLab was never a requirement for employment.
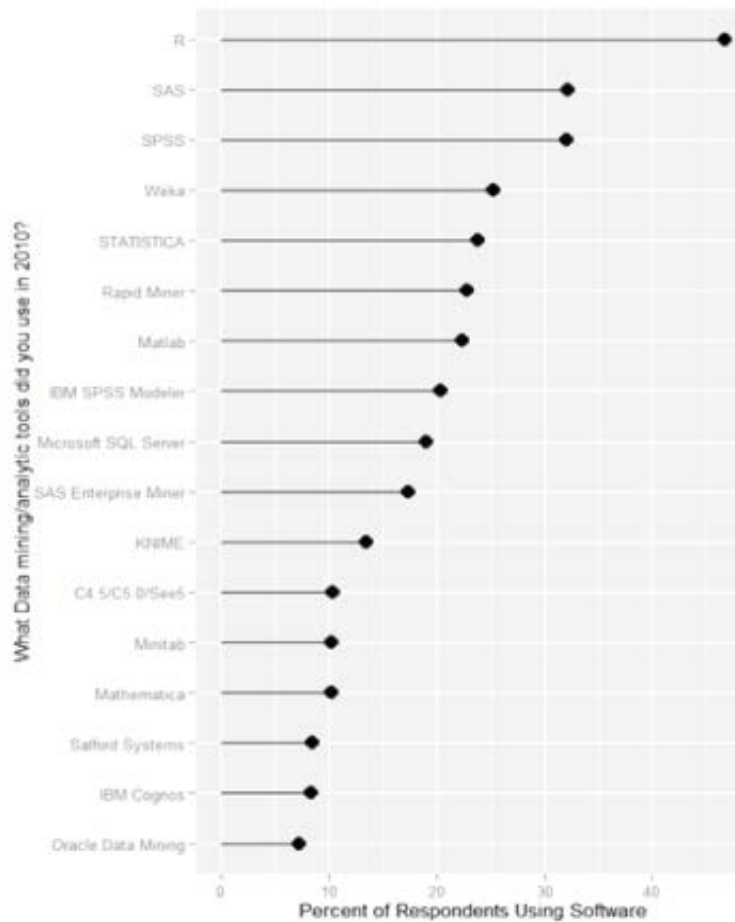
15. R VS. SAS VS. SPSS

Figure 4: 2010 Analytics Survey Results of Analytic Tools (Muenchen, 2014)

As seen above, data miners use R, SAS, and SPSS the most. Because 47%, 32%, and 32% percent of respondents use R, SAS, and SPSS, respectively, it can be inferred that these are the software skills that the greatest proportion of employers will continue to look for (Ozgur, 2015). Surprisingly, schools do not teach students the same software that businesses look for. In his article that measures the popularity of many data analysis software, Robert Muenchen notes that discovering the software skills that employers are seeking would "require a time consuming content analysis of job descriptions" (Muenchen, 2014). However, he finds other ways to figure out the statistical software skills that employers seek. One of these methods is to examine which software they currently use. Muenchen includes a survey conducted by Rexer

Analytics, a data mining consulting firm, about the relative popularity of various data analysis software in 2010. The results of the survey are pictured in Figure 1. As seen, data miners use R, SAS, and SPSS the most. Because 47%, 32%, and 32% percent of respondents use R, SAS, and SPSS, respectively, it can be inferred that these are the software skills that the greatest proportion of employers will continue to look for. However, this method only examines the software that employers might seek if they are hiring, so it does not accurately measure the software that they currently look for. Muenchen's other method does this, studying software skills that employers currently seek as they try to fill open positions. In this approach, Muenchen puts together a rough sketch of statistical software capabilities sought by employers by perusing the job advertising site, Indeed.com, a search site the comprises the major job boards—Monster, Careerbuilder, Hotjobs, Craigslist—as well as many newspapers, associations, and company websites (Muenchen, 2014). He summarized his discovery in Figure 2.
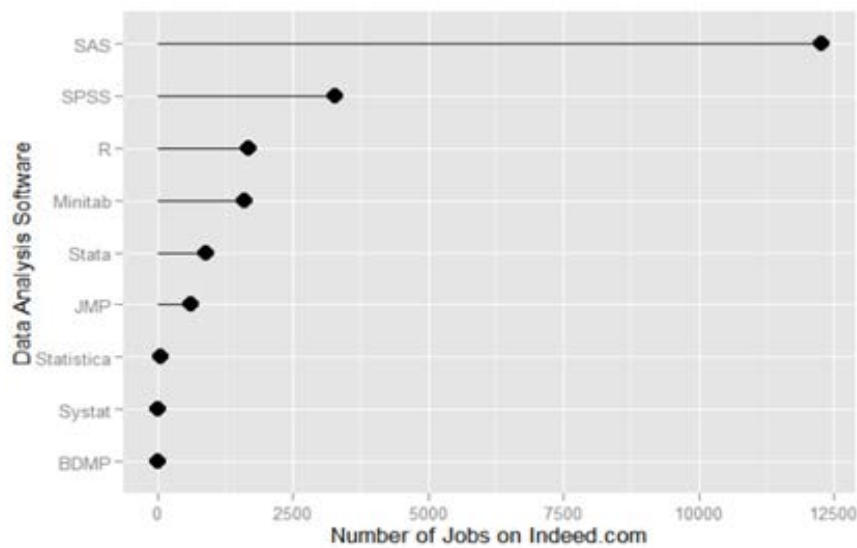

Figure 5: Jobs requiring various software (Muenchen, 2014)

As seen—in contrast to R's greater usage by companies over SAS, illustrated in Figure 3—job openings in SAS substantially lead open positions that require any other data analysis software. For employers, SPSS and R skills finish in second and third place. This second estimation method of Muenchen measures the software skill deficits in the job market. It seems that the demand for people with SAS skills outweighs the number of individuals with this capability. One reason for this disconnect could be that colleges and universities are not teaching SAS skills in proportion to the demand for these skills.

16. PERSONAL EXPERIENCE

Two of the authors have had experience with each program (MatLab, Python, and R) within a business class setting for one author and statistics setting for the other. In the next few paragraphs they will be talking about their experiences in each of these programs in addition to a brief discussion of SPSS, Stata, SAS, and Excel. The pros and cons of all the various applications will be discussed from a student's perspective including a description of how the programs are being used in today's classrooms to enhance the overall educational experience. Although SPSS, SAS, and Excel are not the major software applications being discussed in this paper, it is necessary to briefly discuss them since they are also major competitors that students may encounter after graduation.

Microsoft Excel was probably the most commonplace software that was used in all of my business classes to prepare students for performing everyday analytics in their future career. Excel specifically can be used by small businesses to perform data mining for smaller data sets that consist of up to a few thousand rows. Excel is extremely easy to use and since students are often times introduced to the software in elementary school, it becomes second nature to go to the program for everyday needs. Excel is so widely used that during an internship with a Fortune 50 company, I used Excel daily to help me with basic analytics. Another pro of Excel is that in later versions, you can use add-ins such as MegaStat that help with data analytics. Microsoft has since decided to incorporate many analytical tools such as regression analysis, time-series, and descriptive statistics into its Microsoft 2016 software. The major con of Excel is that it cannot be used with big datasets and therefore is not a viable option when working with big data.

The commonality of Excel may be one of the larger drawbacks to its usage in statistical analyses; the knowledge of Excel was also something one was expected to have rather than taught. The other author never learned how to use Excel in a formal setting. Rather, it was scattered lessons and large gaps. The fact that Excel is considered to be common means that one is simply to acquire the knowledge of it on one's own time. This may lead to frustrating course gaps where one is unable to keep up with the workload or where one must put more time into learning the software than the class dictates solely because the understanding was assumed. This is a persistent problem with commonplace software and could be remedied by having an introductory-level course for those who may not have focused on the use of Excel before a collegiate experience.

SPSS is considered a medium sized analytical tool since it can be used with bigger datasets up to 2 billion cases. Although, SPSS is used for larger projects, it is still very easy to use since it is menu driven. These menu options make SPSS a software that is quick to learn and since it has many similarities to Excel, there is hardly any learning curve. This makes it a very good option for business analytics classes since professors will not be required to spend copious amounts of time acquainting students to a new program. I used SPSS in an Econometrics course while handling an Enterprise Survey Dataset that contained approximately 12 million cases. A con of SPSS that might not make it extremely attractive to be taught is that it can be

difficult to perform data cleansing. Unlike using a programming language like R, SAS, or Python, the user has to manually clean the dataset.

The similarities of SPSS to Excel end at the data-entry step. SPSS is designed primarily for statistical analyses, whereas Excel has a multitude of uses. This means that one is required to download additional updates to Excel in order to utilize it for statistics. SPSS, on the other hand, comes with these packages pre-installed. SPSS is a menu-driven software, which makes it somewhat unique as a statistical software. Most rely on some level of coding in order to input and/or manipulate the data (though SPSS does have that capability as well). This makes it a strong candidate as introductory analytics software. The ability of SPSS to handle large numbers of cases means that students are able to work with a range of data sets in order to understand the nuances as well as the general ideas of analytics. This software, being menu-driven, is intuitive for the layperson to use and is a handy stepping-stone into other software as students are able to being to learn to input code once they have mastered both the menu system and the analysis outputs. Therefore, SPSS does have a place, but it should not be considered one of the better software options for those who are looking to do serious statistical analyses or to make a career as an analyst, because it functions so differently from the rest of the standard software.

Stata is another software used for data analysis, primarily in the soft sciences such as political science. The software is a nice balance between the user-friendliness of SPSS and the coding applications of the other software. Stata still has menus available for those who are just learning the program (or those who do better with a visualization of what the coded commands are doing), but it primarily relies upon some level of code execution in order to work with the data. This coding process is more strict than the process for SPSS, but not as strict as the purely code-based software (such as SAS, R, or Python) are. This means that Stata could be considered the next step between learning from menus and learning to code, especially if one is interested in analytics. The largest drawback to Stata is that it may not necessarily be applicable in certain scenarios. The smallest package of Stata is the most cost-friendly, but can only handle between 1,000-3,000 cases depending on the complexity of the data. This means that it is neither a tool for large data analysis or for learning the basics of analytics software at the smaller levels, but that it can be used to understand the finer points of code without a coding background. This means that students who are more interested in the analyses than the code, and who might not understand the finer points of why a code has bugs or how to debug a code, are able to learn without the vast learning curve that can make SAS a frustrating software to start with.

SAS is an extremely popular analytics software that has been around for numerous years (first limited release was in 1971). My experience with SAS in the classroom environment was in an introduction to data analytics course and during a SAS Shootout Competition with other schools nationwide. The biggest pro of SAS is that it can handle as many cases as your computer has memory to process. This makes it an extremely useful analytical tool because essentially no data set can be too big. I once asked a SAS representative how many cases SAS could handle and their response was to ask me how many I needed. If your computer cannot handle the billions of cases in a dataset then you can use SAS Cloud Analytics and have near

unlimited amounts of space. SAS was also the major analytical tool that my Fortune 50 employer used during my internship, and certification in the program was greatly desired. As a student the biggest con of SAS is that you need to understand the programming language. This creates a learning curve and unless a student is committed to the software it can take several weeks to begin to understand how to even import a dataset and perform basic analytics or data cleaning on a dataset.

SAS as a statistical software logically follows Excel, SPSS, and Stata. This software is primarily code-based and functions much more like pure computer code than any of the other aforementioned software packages. This means that the software requires more precise commands and a better understanding of the underlying code principles in order for the user to make the most of the software. SAS does have the advantage of being able to be run locally (as in on one's own machine) or over the internet, without requiring a download. This necessarily makes it as available as Python and R, as there are no operating system restrictions. The largest challenge comes with reading the log in case of an error and learning how a code can still execute incorrectly even without producing an error message. For example, if one forgets the semicolon at the end of a line, SAS will likely not execute the code at all. However, if one omits a @ or @@ at the end of an input step, the code will still execute, but will miss many of the data points. This means that SAS requires at least some mastery over the basic principles of coding as well as understanding why and how errors are made. This makes the process a frustrating one if students do not come in with knowledge of computer programming or other software which relies, at least in some part, on the coding side of operation. However, this can be mitigated by stepping them through the software in the suggested order, or by giving them access to reading materials that cover the common mistakes made with coding and what an error actually means.

R is a major analytical tool that I believe directly competes with SAS. My experience using R was during my internship for the Fortune 50 corporation, where R was the second largest widely used analytical tool for mining big data. I notice that the most difficult part of using R is the natural programming syntax. When teaching to students this would need to be kept in mind because it can be difficult to learn and use. On the contrary, R has a massive amount of open source coding available online that can help users get started. This is useful for those who have difficulty understanding the language because it offers a stepping stone into the use of the program. Although R might be difficult to understand, once the user has a grasp on the software, the computing capabilities allow the program to process billions of cases quickly.

MatLab is an interesting program that I used in both calculus and differential equations courses. My main usage of MatLab was for basic computations and for graphing equations in three dimensions. These are probably the biggest pros of MatLab in my opinion. It can be used as a sophisticated calculator while also offering the user aesthetically pleasing graphical representations of data. As a student I can say that the biggest downfall of MatLab is the lack of open source code online for the program. Since the usage of the software requires a license to operate the amount of code online is scarce. This means that you have to learn the coding for every use of MatLab on your own without being able to use others preexisting code. Although

this could be seen as a pro since it forces the users to immerse themselves in learning syntax, it can become cumbersome if a specific command is not working.

Python was the first programming language that I ever learned and the program was actually taught to me during a computer software course in high school. I later used the program in an Economic Development Council Internship to perform basic analytics. This brings up one of the biggest pros of Python, in my opinion, and that is that it is fairly easy to learn and offers many add-on programs. For example, I have used Pandas, which is an open source analytical tool that runs through Python. When you combine this ease of use with wide-sweeping applications, that makes Python extremely attractive both in classrooms and in work environments. You do not have to be a programmer to program in python. The commands are extremely simple and you can run commands in Python to read datasets in other statistical software such as SAS. Cons of the Python software include speed and the program's inability to identify and fix semantic errors which could be extremely frustrating when dealing with large quantities of code to perform numerous actions. From a classroom perspective, the major con of Python would be that although it is considered an introductory programming language, this might turn away students who struggle to grasp syntax. However, I would like to note that learning Python is much easier than learning SAS or R, but more difficult to learn than SPSS or Excel.

17. CONCLUSION

In this paper we have discussed the pros and cons of Python, MatLab, and R. We have compared and contrasted each of the languages to one another, while also talking about the educational value of each program in a teaching environment. After reviewing all three of the programs in depth, we have reached the conclusion that Python is the best language to be taught in a classroom environment. This is because it is easy to use and will allow students access to open source coding that can be found online when performing more difficult analysis. However, we would like to note that R might be a better program to teach to students since it is widely used in corporations around the nation for data mining. Having knowledge of a program like R could provide students with a competitive advantage while looking for a job upon graduation. In addition to R, SPSS or SAS could also be viable alternatives which should be considered when teaching big data analytics to students.

REFERENCES

[1]. C. Ozgur, M. Kleckner & Y. Li "Selection of Software for University Courses & for Firms that Use Business Analytics" SAGE Open Journal April-June 2015 pp. 1-12.

[2]. Eglen, S. J. (2009, August 28). A Quick Guide to Teaching R Programming to Computational Biology Students. Retrieved February 20, 2016, from http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000482

[3]. Feldman, Phillip M. (2015, Sept. 18). Eight Advantages of Python Over Matlab. Retrieved from http://phillipmfeldman.org/Python/Advantages_of_Python_Over_Matlab.html

[4]. Guo, Philip. (2014, July 7). Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities. Retrieved from: http://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-us-universities/fulltext

[5]. Guo, Philip. (2007, May). Why Python is a great language for teaching beginners in introductory programming classes. Retrieved from: http://pgbovine.net/python-teaching.htm

[6]. Hughes, Zachariah. (2015, March). Personal Experience Section in MatLab vs. Python vs. R.

[7]. Markham, Kevin. (2016, February 2). Should you teach Python or R for data science? Retrieved February 18, 2016, from: http://www.dataschool.io/python-or-r-for-data-science/

[8]. MATLAB Basic Tutorials. Retrieved February 18, 2016, from: http://ctms.engin.umich.edu/CTMS/index.php?aux=Basics_Matlab

[9]. M. Kleckner, C.Ozgur & C. Wilder "Choice of Software for Business Analytics Courses" 2014 Annual Meeting of the Midwest Decision Sciences Institute, April 2014 pp.69-87

[10]. Muenchen, R. A. (2014, February). "The Popularity of Data Analysis Software." Retrieved from http://r4stats.com/articles/popularity/.

[11]. Ooi, H. 2016. Experiences with using R in credit risk. Retrieved from: http://files.meetup.com/1685538/R%20and%20SAS%20in%20Banking.pdf

[12]. Python for Beginners - Python Training Course - Udemy. Retrieved from:
https://www.udemy.com/python-for-beginners/?siteID=oCUR7eOwwME-
8mj81nbpWjfGzuuaYiVpTg&LSNPUBID=oCUR7eOwwME

[13]. Pyzo. Python vs. Matlab. Retrieved February 21, 2016 from:
http://www.pyzo.org/python_vs_matlab.html

[14]. R Bloggers (2017). Data Science Job Report 2017. Retrieved from:
https://www.r-bloggers.com/

[15]. Revolution Analytics. (2014, February 20). What is R? Retreived February 21, 2016, from:
http://www.inside-r.org/what-is-r, February 20, 2014.

[16]. The R Foundation. (2017). The R Project for Statistical Computing. Retrieved from:
https://www.r-project.org/

[1]Ceyhun Ozgur
Valparaiso University
ceyhun.ozgur@valpo.edu

[2]Taylor Colliau
Valparaiso University
taylor.colliau@valpo.edu

[3]Grace Rogers
Valparaiso University
grace.rogers@valpo.edu

[4]Zachariah Hughes
Valparaiso University
zachariah.hughes@valpo.edu

[5]Elyse "Bennie" Myer-Tyson
Valparaiso University
bennie.myertyson@valpo.edu