Poster: A Topic Analysis of the R Programming Language

Abigail Atchison, Haley Anderson, Christina Berardi, Natalie Best, Cristiano Firmani, Rene German, Erik Linstead
Schmid College of Science and Technology
Chapman University
{atchi102,ander427,berar105,best120,firma103}@mail.chapman.edu,{german,linstead}@chapman.edu

ABSTRACT
We leverage Latent Dirichlet Allocation to analyze R source code from 10,051 R packages in order to better understand the topic space of scientific computing. Our method is able to identify several generic programming concepts and, more importantly, identify concepts that are highly specific to scientific and high performance computing applications.

CCS CONCEPTS
• Software and its engineering → Software libraries and repositories;

KEYWORDS
R, topic modeling, machine learning

ACM Reference Format:

1 INTRODUCTION
Despite its increasing popularity [5] and the availability of large repositories of open source code, the R programming language [13] has received relatively little attention from the empirical software engineering community. While this is ironic in the sense that many of the tools used to mine software repositories are implemented in this language, including those used in this study, this also creates opportunities for new research directions. In this poster we apply topic modeling, in particular Latent Dirichlet Allocation (LDA) [3], to identify functional concepts from over 10,000 R packages. The topics generated and discussed in this study represent traditional programming topics, as well as a divergence from these generic concepts through the generation of a wide range of topics centered on scientific and high performance computing. This study represents the first of its kind, focused on statistical computing software, and provides a foundation for future work that looks to compare and contrast this rapidly growing codebase with other software domains and repositories.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
ICSE ’18 Companion, May 27-June 3, 2018, Gothenburg, Sweden © 2018 Copyright held by the owner/author(s).
https://doi.org/10.1145/3183440.3195087

2 DATA
Data for this study was taken from the Comprehensive R Archive Network (CRAN) [1]. CRAN consists of 10,051 unique R packages. The requirements for submitting R packages are documented on CRAN, as well as the extensive review process each package undergoes before it is added to the repository. Each package is composed of R files, with varying amounts of C and C++ files for performance optimizations. In our analysis here we focus only on topic models generated solely from R source code but, in future analyses, we will include C and C++ code bundled with the packages.

3 METHODS
In this study we apply LDA to a curated repository of R source code files in order to extract topic models that will give insight into the functionality implemented in R. LDA is a statistical topic modeling algorithm capable of learning the underlying document-topic and topic-word distributions from a text corpus. LDA-based approaches to modeling software repositories have been shown to be more effective than non-statistical techniques [12]. Since its first application in [11], LDA and similar topic models have been used for a wide variety of software analysis applications [6, 14].

To prepare our corpus for LDA, we parsed the source code files, filtering stop-words and applying standard naming heuristics (camel case, underscore, etc) to split identifiers. This was then processed using the LDA implementation provided by the R package, Mallet [2]. For this study we found 100 topics to be sufficient by generating topic models of various sizes and assessing the results for human interpretability.

4 RESULTS
Figure 1 provides a sample of topics extracted by our model, which includes functions both non-specific and specific to R. Topic 26, for example, illustrates the use of R to produce summary statistics on existing data sets while, topics 14 and 34 demonstrate R functionality associated with api usage and I/O operations, respectively. Extending beyond these fundamental concepts are topics that demonstrate the specific domains in which R is commonly applied. Topic 28 alludes to DNA sequencing while topic 32 demonstrates processing light spectrum and topic 59 shows a study of gender mortality rates.

Most importantly, though, is the multitude of topics in which R is clearly being leveraged for its statistical computing capabilities, demonstrating functionality only lightly supported, if at all, in other widely-used programming languages. Topic 12 demonstrates functions for depth-based classification. Topic 55 presents diagnostics for univariate stationary extreme value mixture models such as kernel density estimation. Topic 65 shows the use of
An interesting trend is that other topics not centered on scientific computing languages. The way in which a specific algorithm such as “sample”, “summary”, and “data.” An interesting trend is that other topics not centered on scientific computing, like the general and domain specific topics discussed previously, do not relate to topics within their own category as commonly. For example, topics describing general utilities often do not share words with other topics describing general utilities. Though many of these non-scientific topics do not relate to each other they do often relate to the scientific topics. This is significant as it once again affirms that R is centered on its scientific, high-performance use.

The emerging themes presented by the topic models generated in this study begin to illustrate an important aspect of the study of statistical and scientific computing languages. The way in which a language is commonly leveraged, examined through a study of the topic space of source code files existing in the language’s ecosystem, can give great insight into the strengths of that language and serves as a baseline for comparison to other languages whose properties have already been widely studied.

5 RELATED AND FUTURE WORK

While we believe we are the first to take a machine learning approach to analyzing R with topic models, others have analyzed the strengths and weaknesses of R, such as Caragea et al. in their SWOT (Strengths, Weaknesses, Opportunities, Threats) analysis [4], and Culpepper et al. in their review of the limitations and benefits of a statistical computing tool that is continuously being updated [7]. Hornick, 10 years after his original analysis of R in [9], which largely introduced the language to the statistical community, returned to inspect the evolution of R [8] noting the rise of the R programming language and what it could mean for statistical computing.

In the future, we intend to expand the initial work described here to include more facets of contemporary scientific computing software packages. To start, we will expand our corpus by incorporating R packages from repositories other than CRAN, and also by parsing projects implemented in other scientific computing languages such Matlab. Further, as outlined earlier, C and C++ code is often included in R and Matlab packages. We are curious if this code consists primarily of performance optimizations, and what those optimizations contribute to the R ecosystem. Additionally, we would like to explore the naming conventions of R and Matlab in order to understand how they compare with what has already been observed in languages such as Java[10]. In this way, we can provide a robust and empirically-based model of the fundamental differences from the procedural and object-oriented technologies that have been the emphasis of research in mining software repositories.

REFERENCES

[1] [n. d.]. The Comprehensive R Archive Network. ([n. d.]). http://CRAN.R-project.org/

[2] [n. d.]. CRAN - Package mallet. ([n. d.]). https://cran.r-project.org/web/packages/mallet


