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:
Abigail Atchison, Haley Anderson, Christina Berardi, Natalie Best, Cristiano Firmani, Rene German, Erik Linstead. 2018. Poster: A Topic Analysis of the R Programming Language. In *ICSE '18 Companion: 40th International Conference on Software Engineering Companion, May 27-June 3, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, Article 4, 2 pages. https://doi.org/10.1145/3183440.3195087

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.

ICSE '18 Companion, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5663-3/18/05.
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 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

#	Topic
5	fit family coef model weights
11	cluster dist means result cores
12	ddalpha points patterns depths classifier
13	env gui envir container exists
14	url query api json status
16	species train prediction data lda
26	par lower upper dist distribution
28	gene data verbose ontology flag
32	spec wave frequency seq matrix
34	file read write header append
55	lambda kernel density kerncentres gaussian
59	age sex data summary mortality
65	seed nmf rng set random
75	amelia tcltk priors frame state
84	msm states population transition covariates
85	time date year period series

Table 1: A sampling of the 100 topic models created from R source code files.

seeding techniques through random number generation and non-negative matrix factorization, while topic 75 alludes to the practice of imputing data. Various statistically-grounded algorithms can be specifically identified in topics as well. For example, topics 84 and 85 demonstrate the use of multi-state Markov models and time series analysis, respectively. Topic 5, 11, and 16 all outline clustering and classification techniques.

We also consider co-occurrence in our investigation of the R topic space. Co-occurrence can be defined in the context of this study as the appearance of the same word in multiple topics. This helps us to identify topics that overlap and, through the surfacing of topics that overlap with many other topics, identify topics that center on key applications of R. In the topics generated,those centered on scientific application commonly overlapped with each other. This comes as no surprise as there are many general key words intended for scientific computing that are used in multiple R packages regardless of the 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

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[4] Nicoleta Caragea, Antoniade-Ciprian Alexandru, Ana Maria Dobre, et al. 2014. R–a Global Sensation in Data Science. *Revista Română de Statistică nr* (2014), 7.

[5] Stephen Cass. 2017. The 2017 Top Programming Languages. (2017). https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages

[6] Tse-Hsun Chen, Stephen W Thomas, and Ahmed E Hassan. 2016. A survey on the use of topic models when mining software repositories. *Empirical Software Engineering* 21, 5 (2016), 1843–1919.

[7] Steven Andrew Culpepper and Herman Aguinis. 2011. R is for revolution: A cutting-edge, free, open source statistical package. *Organizational Research Methods* 14, 4 (2011), 735–740.

[8] Kurt Hornik. 2016. Are there too many R packages? *Austrian Journal of Statistics* 41, 1 (2016), 59–66.

[9] Kurt Hornik and Friedrich Leisch. 2002. *Vienna and R: Love, marriage and the future*. na.

[10] Erik Linstead, Lindsey Hughes, Cristina Lopes, and Pierre Baldi. 2009. Exploring Java software vocabulary: A search and mining perspective. In *Proceedings of the 2009 ICSE Workshop on Search-Driven Development-Users, Infrastructure, Tools and Evaluation*. IEEE Computer Society, 29–32.

[11] Erik Linstead, Paul Rigor, Sushil Bajracharya, Cristina Lopes, and Pierre Baldi. 2007. Mining concepts from code with probabilistic topic models. In *Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering*. ACM, 461–464.

[12] Erik Linstead, Paul Rigor, Sushil Bajracharya, Cristina Lopes, and Pierre F Baldi. 2008. Mining internet-scale software repositories. In *Advances in neural information processing systems*. 929–936.

[13] R Development Core Team. 2008. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org ISBN 3-900051-07-0.

[14] Stephen W Thomas, Bram Adams, Ahmed E Hassan, and Dorothea Blostein. 2010. Validating the use of topic models for software evolution. In *Source Code Analysis and Manipulation (SCAM), 2010 10th IEEE Working Conference on*. IEEE, 55–64.