Paper/Presentation Outline [Stat 159 October 13 2016]

Title: Licenses for Code

Authors: Shannon Chang, Manuel Horta, Nura Kawa, Jared Wilber

---

Introduction

We discuss Licenses in regards to Open Source and Reproducibility. There exists a plethora of licenses to choose from with regards to software development. This paper discusses some of the tenets of licenses with regards given specifically to open-source software. We discusses open-source software, licenses in the context of the former, as well as four specific types of licenses (GNU GPL, BSD, MIT, and APACHE 2.0). Finally, we discuss strategies to choose which licesnse is the best fit for your open-source project.

Open Source

What is Open-Source?

Open-source software is software for which the original source code is made freely available and may be redistributed and modified.
Can Open Source software be used for commercial purposes?
Open Source can be used for commercial purposes. One of the key criterion used by the Open Source Initiative to evaluate licenses states that there shall be "no discrimination against fields of endeavor" (The Open Source Definition (Annotated)"). This guarantees that all Open Source software can be used for commercial purposes. It should be noted, however, that there is a distinction between "commercial" and "proprietary." What this means is that software obtained under an Open Source license may not be further distributed with added restrictions. Specifically, software under a copyleft-style Open Source license may only be further distributed with that same license attached.
###What is "free software" and is it the same as "open source"?

"Free software" and "open source" both refer to "software released under licenses that guarantee a certain, specific set of freedoms" ("Frequently Answered Questions"). Both encapsulate software released under copyleft and permissive (non-copyleft) licenses. As a result, there is a lot of overlap in definition between the two terms. Thus, the Open Source Initiative states in their "Frequently Answered Questions" guide that the two terms can be used interchangeably; the OSI especially encourages use of the phrase "free and open source software" when there is confusion between the two.

Despite the similarities in “free software” and “open source” definition, there are still some key distinctions in history and connotation of each phrase’s wording. Free software is associated with the Free Software Foundation, which was founded in 1985 to protect and promote free software. Open Source, on the other hand, was coined in 1998 by the Open Source Initiative, which shared similar motivations with the FSF, but disagreed on how to promote free software.The OSI’s founders saw software freedom as more of a practical matter than an ideological one. Many of those who have adopted the use of “open source” share the OSI founders’ disagreement with the FSF on how should promote such software, while many also disagree with the FSF on why such software should be promoted. Another major concern amongst “open source” adopters is the ambiguous connotation of the word “free” and whether this refers to freedom or price. The Open Source Definition specifies that “open” means the right to inspect software source code as well as the interminable right to fork and use covered code free of charge. One more key difference between “free software” and “open source” is that the FSF uses a four point definition to evaluate licenses while the OSI uses a ten point definition. Both approaches to the same practical results, so this is yet another matter of connotation.

What is “copyleft”? Is it the same as “open source”?

Copyleft consists of licenses that allow derivative works under the condition that they be placed under the same license as the original. Many copyleft licenses are Open Source, but not all Open Source licenses are copyleft. What this means for software that is Open Source but not copyleft, is that it can be implemented in other software that fall under different licenses; this includes non-open-source/proprietary licenses. This specific type of license is called a “permissive” Open Source license.

Licenses

What is a License? Why do you need one?

A software license tells others what they can and can’t do with your source code. If you want your project to be open-source, you should get a license.

Generally speaking, the absence of a license means that the default copyright laws apply. This means that you retain all rights to your source code and that nobody else may reproduce, distribute, or create derivative works from your work. This might not be what you intend.
Can I call my program “Open Source” even if I don’t use an approved license? No.

Open source software by definition has its source code available, modifiable, and redistributable. By default, any creative works we make (including

code) are under exclusive copyright. Hence, if the owner does not explicitly give permission to modify and redistribute the work (in the form of a license), then it is not open-source material.
###What is a "permissive" Open Source license?

A permissive software license is a "non-copyleft license". Generally they "let people do anything they want with your code as long as they provide attribution back to you and don't hold you liable."

Licenses for Code

What is the difference between source code and compiled code?

Code exists in two forms: source code and object code or compiled code. Source code is any collection of computer instructions written in a programming language readable by humans. Popular languages used for source code include R, Python, C, and Java. This source code is read by a compiler, a set of computer programs that can read source code and translate it into compiled/object code. Object code can be the same as machine code (i.e., binary); if not, it is then translated into a machine language by an assembler.

Why Creative Commons discourages use of the CC BY license for code?

With respect to licensing, it is important to choose a license that differentiations between source and compiled code. Creative Commons therefore strongly discourages the use of the CC-BY license for code, because "The license does not make a distinction between source or compiled forms of the work whereas licenses intended for use on software generally refer to source code as 'the preferred form for making modifications.' This explicit reference to source code recognizes the fact that software code can exist in two forms, source and compiled, and for modification transmission of the binary form alone is not sufficient. (Stodden 36)"

Describe the Licenses for code: Why get a license?

It is not imperative that a user who uploads his code to a Github repository obtain a license. However, any user that wants their work to be open-source - essentially, be open to being reproduced - should get a license. Without a license, a developer's code is subject to copyright laws. "Components of the research compendium have different features that necessitate different licensing approaches... Licenses do not remove or rescind copyright protection but allow the creator to specify the conditions under which use of the work takes place. ...even if a license fails to be recognized as a valid contract by a court, use of the work will remain subject to injunction and other remedies associated with copyright violation (Stodden 13). Thus, licenses allow a developer to retain copyright protection over his work while keeping the work open-source.

What is the GNU General Public License?

The GNU GPL is a copyleft license (derivative work must be distributed under the same license) that requires anybody who distributes either the original code or some derivative of it to make the source available under the same terms as the original. It also provides an express grant of patent rights from contributors to users. It was created by Richard Stallman and is one of the most popular software licenses. Bash, GIMP, and Privacy Badger use it.

What is the BSD license?

The BSD license was originally designed (in 1988) for its namesake, the Unix-like operating system called the Berkeley Software Distribution. Since then it has gone through multiple versions and is the sixth most popular license on GitHub. It is a permissive license and has three main versions: “BSD Old”, “BSD New” and “FreeBSD”.

What is the MIT license?

The MIT license is the most popular software license. It was developed at the Massachusetts Institute of Technology(MIT). It is a very short license. As opposed to copyleft, it is permissive; i.e. it allows people to do anything they want with the code so as long as they provide attribution back to the code’s author and don’t hold them liable for anything. It is used by j.Query, .NET Core, and Rails.

What is the Apache 2.0 license?

The Apache 2.0 license is, as with the MIT License, permissive. It provides essentially everything the MIT license does with the exception that it provides an express grant of patent rights from contributors to users.

Which Open Source license should I choose to release my software under?

Clearly there are many licenses to choose from. And maybe you just don’t know for certain which to choose. For example, If you’re concerned about patent

rights you may want to use the Apache 2.0 license. Or if you want something permissive, the MIT license may be for you.
Ultimately, to properly choose a license that suits you, you need to understand what a license provides you.

A great tool and starting point for this choice is the website choosealicense.com.
This website allows you to see a quick description of some common open source licenses. Clicking them allows you to see the permissions, limitations, and conditions for each.

Reproducible Research

What is the Reproducible Research Standard (RRS) framework proposed by Stodden?

The Reproducible Research Standard "is a licensing structure that makes media, code, data, and data arrangements – the research compendium – available for reuse, possibly with attribution." If a researcher wishes to release his or her research compendium entirely to the public domain, this also complies with the Reproducible Research Standard.

The goal is to make research compendia available so it is possible for results to be verified, and to promote further research upon previous works.

Conclusion

Open-Source software is a very important part of repoducible and collaborative data science. One of the most important features within this domain is the software license. The software license comes in many forms, each with their own caveats. Selecting which license to use is highly dependent on the project at hand and a very important step in producing reproducility.

References

"No License." Web. 12 Oct. 2016. http://choosealicense.com/no-license/

"Frequently Answered Questions." News. Open Source Initiative, n.d. Web. 12 Oct. 2016. https://opensource.org/faq#free-software.

"The Open Source Definition (Annotated)." News. Open Source Initiative, n.d. Web. 12 Oct. 2016. https://opensource.org/osd-annotated.

Stodden, Victoria. "Enabling Reproducible Research: Licensing Scientific Innovation". International Journal of Communications Law and Policy, Volume 13, Winter 2009. 0Web. http://web.stanford.edu/~vcs/papers/ijclp-STODDEN-2009.pdf