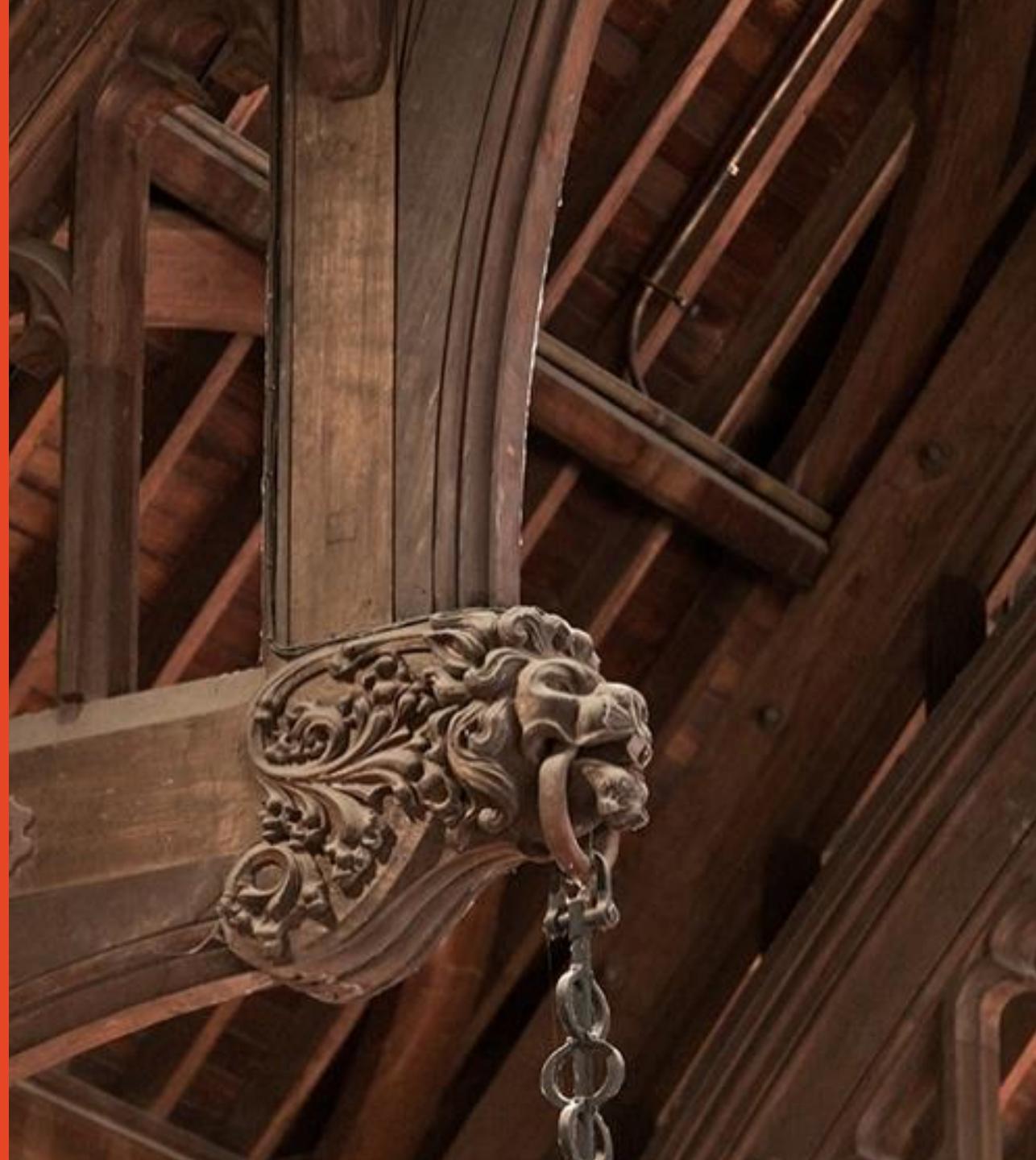


INFO4444 Computing 4 Innovation

Week 6: Open Data, Free and Open-Source Software

**Presented by: Dr. Eman Sayed
School of Computer Science**



Copyright warning

COMMONWEALTH OF AUSTRALIA

Copyright Regulations 1969

WARNING

This material has been reproduced and communicated to
you by or on behalf of the University of Sydney
pursuant to Part VB of the Copyright Act 1968 (**the
Act**).

The material in this communication may be subject
to copyright under the Act. Any further copying or
communication of this material by you may be the
subject of copyright protection under
the Act.

Do not remove this notice.

Agenda

Section One (1st Half)

1.1 Open Data

1.2 Open Collaborative Innovation

Section Two (2nd Half)

2.1 Free and Open Source Software

2.2 Free and Open-Source Software Licenses

Distributed Innovation

Approaches involve external parties own's innovation:

- A. Product platforms
- B. Web APIs
- C. Crowdsourcing innovation / Crowdfunding Innovation
- D. **Releasing data sets “Open data”**
- E. **Free and Open-Source Software**
- F. User innovation
- G. Platform ecosystems
- H. Accelerators, investment and others

Releasing Data Sets “Open Data”

What is ‘Open Data’?

“Open means anyone can freely access, use, modify, and share for any purpose (subject, at most, to requirements that preserve provenance and openness).”

Put most succinctly:

“Open data and content can be freely used, modified, and shared by anyone for any purpose”

<https://opendefinition.org/>

Releasing data sets

- Governments’ “open data” initiatives
 - Static data (e.g. tables of static data)
 - Live data feeds (eg an RSS feed or data service)
 - The Australian federal government (<http://data.gov.au>)
 - Electoral boundaries
 - Crime data, census data
 - NSW Government (<http://data.nsw.gov.au/>)
 - Bus stop data, Electricity consumption data, pollution education etc.

Releasing data sets

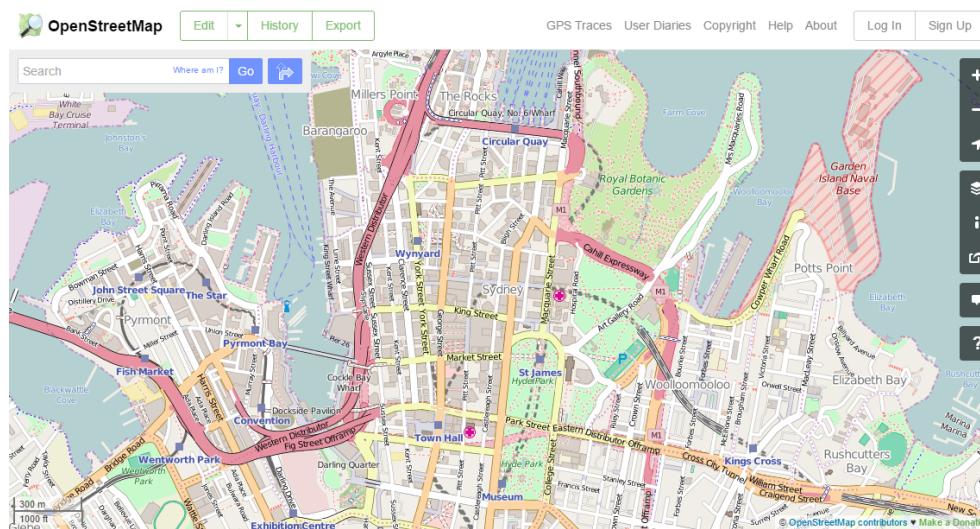
- Many communities are actively building and maintaining open datasets.
- Examples:
 - OpenStreetMap - www.openstreetmap.org/
 - OpenAddresses - openaddresses.io/
- Companies often encourage users to build applications using their data.
- Open data fosters innovation by enabling developers and researchers to create new tools, services, and insights.
- It also promotes transparency, collaboration, and public value by empowering communities to collectively improve and expand shared datasets.

An open source map

– Open Street Map



Steve Coast founded in 2004 for mapping the UK
<http://www.allthepeople.net/steve-coast>



<http://www.openstreetmap.org/#map=15/-33.8680/151.2100>

M Haklay, and P Weber, "Openstreetmap: User-generated street maps", *IEEE Pervasive Computing*, 2008 - ieeexplore.ieee.org

OpenStreetMap Foundation



http://wiki.osmfoundation.org/wiki/Main_Page

- The OpenStreetMap Foundation is an international not-for-profit organization that supports (but does not control) the OpenStreetMap project.
- Established in 2006, its mission is to promote the growth, development, and distribution of free geospatial data.
- The initiative aims to provide free geographic data for anyone to use, contribute to, and share.
- Geospatial data is crowdsourced by volunteers who collect information through manual surveys, GPS devices, aerial imagery, and other tools.

Value in Open Data

- **Economic value:** sectors with the highest potential include Education, Consumer Products, and Transportation.
- **Big impact of data:** lies in enabling transparency, innovation, and data-driven decision-making.
- **Business opportunities:** arise from developing new products, services, and tools using openly available data.

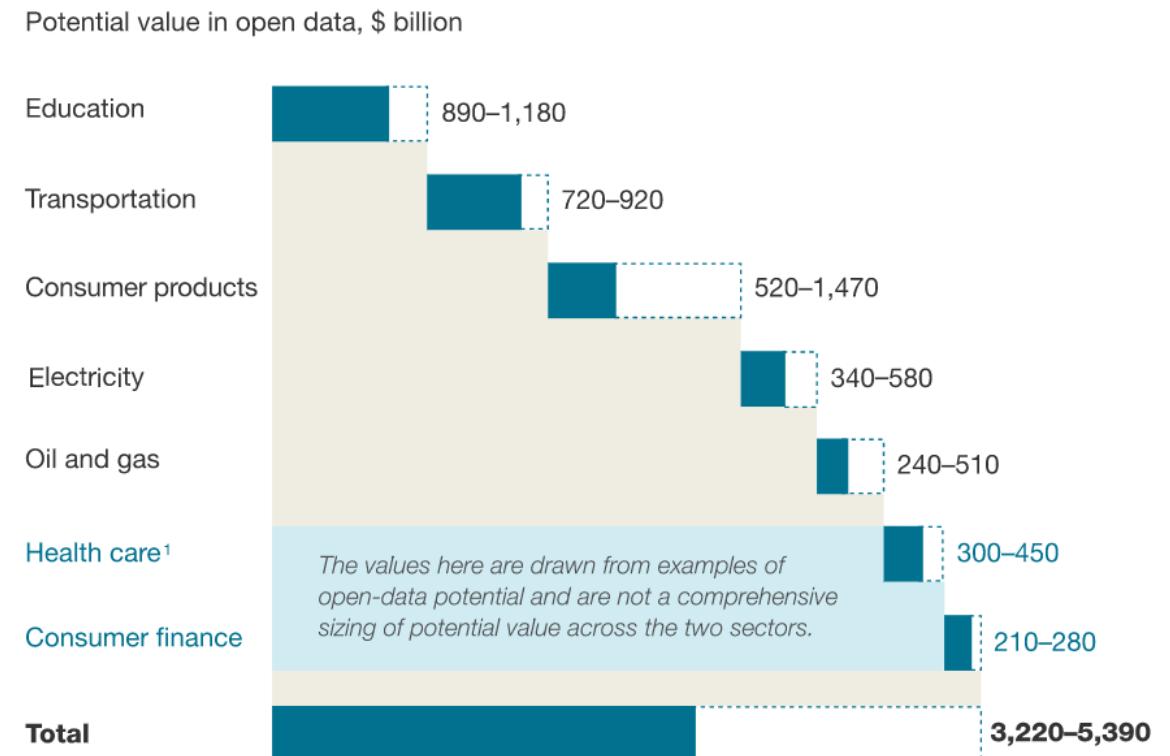
Governments play a central role by:

- Releasing public data sets.
- Setting standards and privacy guidelines.
- Encouraging private-sector innovation through open-data initiatives.

<https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/open-data-unlocking-innovation-and-performance-with-liquid-information>

Exhibit

Open data can help unlock \$3 trillion to \$5 trillion in economic value annually across seven sectors.



¹Includes US values only.

Source: McKinsey Global Institute analysis

Value in Open Data – Self-reinforcing cycle

- Open data grows as more people recognize its value and contribute, improving quality and detail.
- Sustainable momentum depends on a strong ecosystem supported by clear policies that protect all stakeholders.
- For companies: Success means investing in the right technologies and data talent.
- For individuals: It requires being informed, responsible contributors and users of open data.
- A thriving open-data cycle fuels innovation, trust, and participation.

Open Data – Examples

	Search
Data Provided	<input type="checkbox"/> Video <input type="checkbox"/> Image <input type="checkbox"/> GPS <input type="checkbox"/> Codes <input type="checkbox"/> Thermal <input type="checkbox"/> LiDAR
Date	2006 <input style="width: 100px; height: 10px; border: 1px solid #ccc; border-radius: 5px;" type="range" value="2006"/> 2019
Diversity	<input type="checkbox"/> Weather <input type="checkbox"/> Season <input type="checkbox"/> Night <input type="checkbox"/> Illumination
Annotation	

Dataset	Date	Location	Data Provided	Annotation	Diversity	Traffic
 A2D2 by Audi Electronics Venture	2019	Germany	LiDAR, Video, Vehicle Data	Bounding Box	Season	Urban
 Apollo Open Platform by Baidu Inc	2018	China	Video, LiDAR, Vehicle Data	Bounding Box	Weather	Urban, Highway, Rural
 ApolloScape by Baidu Inc	2018	China	Video	Semantic Label, Lane Marking	Illumination, Weather	Urban, Highway, Rural
 Argo by Argo	2019	United States	Vehicle Data, LiDAR, Video, Image	Bounding Box, Optical Flow, Behavioral Label, Semantic Label, Lane Marking	Season, Weather, Night	Urban
 Belgium Traffic Sign by ETH Zurich	2011	Belgium	Image	Bounding Box		Urban
 Berkeley DeepDrive by UC Berkeley	2017	San Francisco Bay Area, New York, USA	Video, Image, GPS, Codes	Bounding Box, Semantic Label, Lane Marking	Weather, Night, Illumination	Urban, Rural, Highway
 Bosch Small Traffic Lights by Bosch North America Research	2017	San Francisco Bay Area, USA	Video	Bounding Box	Weather, Illumination	Urban, Rural

Open Data – Example

Pedestrian's safety is one of the main concerns of the innovative companies as well as for the governments.

Caltech Data is one example of the open data used for this concern.

The dataset contains richly annotated video, recorded from a moving vehicle, with challenging images of low resolution and frequently occluded people. <https://data.caltech.edu/records/f6rph-90m20>

Caltech Pedestrian Detection Benchmark



Description

The Caltech Pedestrian Dataset consists of approximately 10 hours of 640x480 30Hz video taken from a vehicle driving through regular traffic in an urban environment. About 250,000 frames (in 137 approximately minute long segments) with a total of 350,000 bounding boxes and 2300 unique pedestrians were annotated. The annotation includes temporal correspondence between bounding boxes and detailed occlusion labels. More information can be found in our [PAMI 2012](#) and [CVPR 2009](#) benchmarking papers.

Download

- **Caltech Pedestrian Dataset.** The training data (set00-set05) consists of six training sets (~1GB each), each with 6-13 one-minute long seq files, along with all annotation information (see the paper for details). The testing data (set06-set10) consists of five sets, again ~1GB each. New: annotations for the entire dataset are now also provided. Output files containing detection results for all evaluated algorithms are also available.
- **Seq video format.** An seq file is a series of concatenated image frames with a fixed size header. Matlab routines for reading/writing/manipulating seq files can be found in [Piotr's Matlab Toolbox](#) (version 3.20 or later required). These routines can also be used to extract an seq file to a directory of images.
- **Matlab evaluation/labeling code (3.2.1).** The annotations use a custom "video bounding box" (vbb) file format. The code also contains utilities to view seq files with annotations overlaid, evaluation routines used to generate all the ROC plots in the paper, and also the vbb labeling tool used to create the dataset (see also this somewhat outdated [video tutorial](#)).
- **Additional datasets in standardized format.** For convenience we are posting full images/annotations in seq/vbb format as well as detection results for all evaluated algorithms on a number of additional datasets. This facilitates training/testing on these additional datasets and exact reproduction of all ROC curves. Full copyright remains with the original authors, please see the respective website for additional information including how to cite evaluation results on these datasets: [INRIA pedestrian dataset \[converted\]](#), [ETH pedestrian dataset \[converted\]](#), [TUD-Brussels pedestrian dataset \[converted\]](#), [Daimler pedestrian dataset \[converted\]](#).

Benchmark Results

[Algorithm Details and References](#) | [Algorithm Runtime vs. Performance](#)

For details on the evaluation scheme please see our [PAMI 2012](#) paper.

Note: The evaluation scheme has evolved since our [CVPR 2009](#) paper.

Note: We render at most 15 top results per plot (but always include the VJ and HOG baselines).

1. Caltech Pedestrian Detection Dataset. We have made it available on GitHub under a standard MIT license.

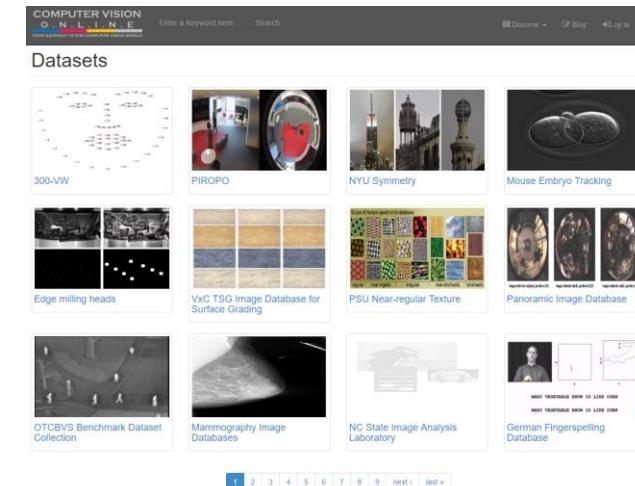
<https://pdollar.github.io/files/papers/DollarCVPR09peds.pdf>

Example: Computer Vision / Medical Imaging Community

- In Research community, open source and open data is important for
 - Benchmarking / Dissemination
 - Replication / Citations
 - Reputation / State of the Art
 - Building a community



<http://www.visceral.eu/>



<https://computervisiononline.com/datasets>
<http://www.cvpapers.com/cvpr2017.html>
<http://www.cvpapers.com/cvpr2014.html>

Open Data – Australian Government

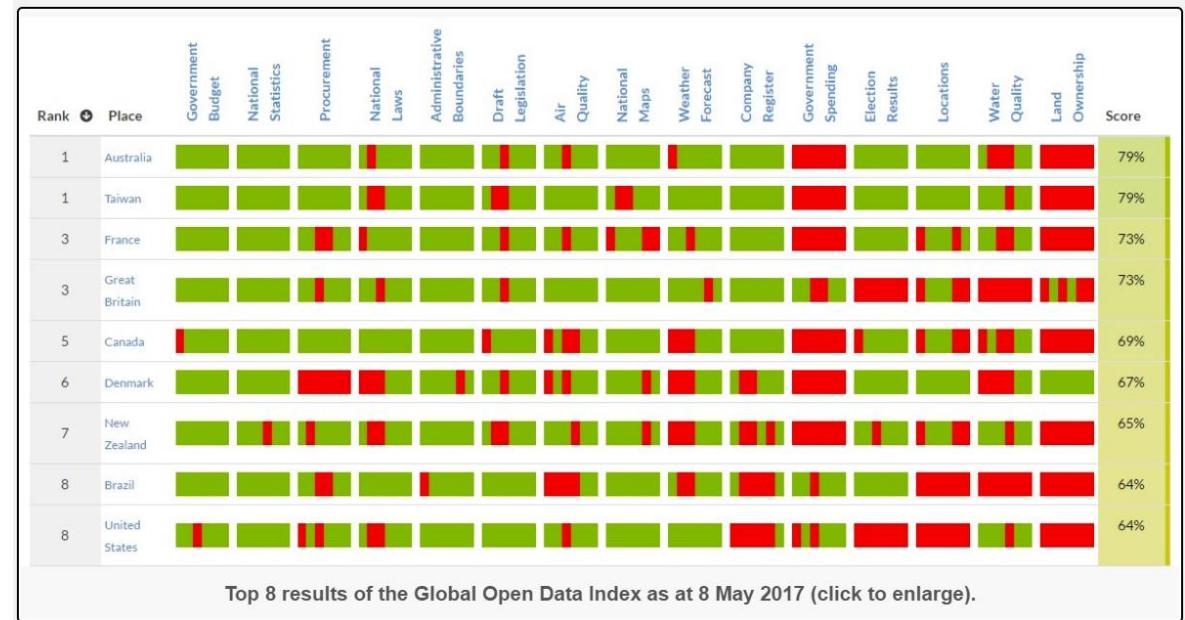
Why Open Data?

"Data is a game-changer for government. Open data provides the intelligence for insight, invention and exploration that translate into better products and services that improve everyday life and encourage business growth."

The Hon. Victor Dominello, MP, Minister for Innovation and Better Regulation, launching the 2016 Open Data Policy



NSW Government Open Data Policy



<http://data.nsw.gov.au/>

<https://www.ipc.nsw.gov.au/open-data-infographic>

<https://www.spatialsource.com.au/gis-data/australia-leads-world-open-data>

ANZICS Datathon 2018 - Sydney Uni 2024 - Queensland University of Technology

- Australia New Zealand Intensive Care Society (ANZICS)
- To explore large critical care datasets to answer a clinical question.
- Partnership between Society, Industry, and the University.
- Freely available medical data for research. <https://mimic.mit.edu/>

Premium Sponsor

PHILIPS

Co – hosts



Supporters



Alfred Health

Google Cloud



The University of Sydney

<https://mimic.physionet.org/>



Open Collaborative Innovation

Free and open-source software

Software – Proprietary, Free and Open-source

- Most software was proprietary
 - Built by or for a specific person, organization or group of organizations:
 - Own intellectual property rights; and
 - Full control over the software and how it is used.
- In free and open-source software:
 - Source code is made available.
 - Source code can be changed and redistributed by others.
 - (more precise definitions coming later)

Proprietary, Free and Open-source Software - Examples

Proprietary software



IBM DB2

Oracle Database



Internet Explorer

(Many of these also use open source within them)

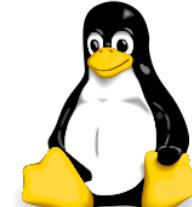
Free and open source software



Android



Chrome



Linux



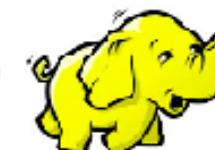
Firefox



Apache
HTTP Server



Alfresco



Hadoop



Tomcat



openstack™
CLOUD SOFTWARE



node.js

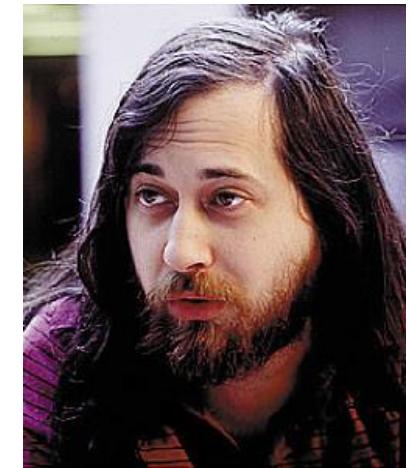


Apache
OpenOffice™



Free and open source software – Example GNU

- Programmer at MIT AI Lab
- Many manufacturers supplied source code (eg drivers)
- Stallman (and others) modified software to meet their needs
- Companies became more protective of their source code
- Stallman (and others) were refused access to the source code for the software of the first laser printer (from Xerox)
- Stallman was not able to modify the software
- Inspired Stallman that people should be free to modify all software



Richard Stallman

*"technical means
to a social end"*

What is Free Software? (using Free Software Foundation definition)

- Free Software Foundation (FSF)
- Founded by Richard Stallman in 1983
- "Free" refers to freedom, not price
- Focuses on ethical and social values of technology use

What is Free Software? (using Free Software Foundation definition)

- “Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. (Source: <http://www.gnu.org/philosophy/free-sw.html>)
- Four essential freedoms of a Software
 - Run the program, for any purpose (*freedom 0*)
 - Study how the program works, and change it to make it do what you wish (*freedom 1*)
 - Redistribute copies of it (*freedom 2*)
 - Distribute copies of your modified versions to others (*freedom 3*)

“Copyleft”

- A general method for making a program (or other work) free, and requiring all modified and extended versions of the program to be free as well.” (Free Software Foundation)
- Example: GNU Public License (GPL)

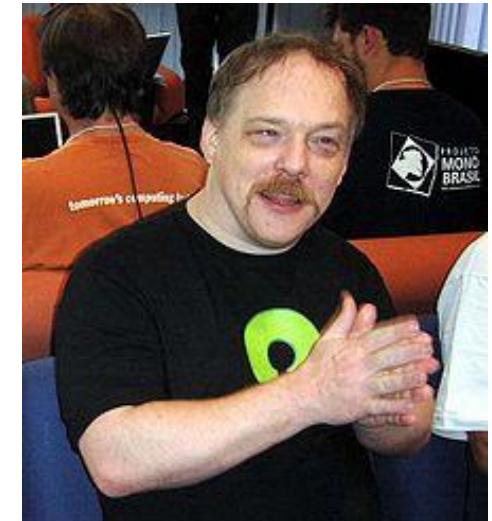


Copyleft symbol

<http://www.gnu.org/copyleft/>

Open Source Software

- By Eric Raymond (1998)
- “Free software” was:
 - Ambiguous
 - Intimidating to companies
- Pragmatic aspects of providing access to source code, rather than the “rights” of users



Eric Raymond

Linus's law - "Given enough eyeballs, all bugs are shallow"

What is Open Source Software? (using Open Source Initiative definition)

- To be classified as OSS, the software must be (according to its licence):
 - Freely redistributable
 - Source code must be available for free or at reasonable reproduction cost
 - Modifications and derived works must be allowed and be distributable under same terms
 - Can protect integrity of author's source code as long as allow source code patches
 - No discrimination against people/groups
 - No discrimination against fields of endeavour
 - Must not be restricted to use with a specific product
 - Must not place restrictions on other software distributed with it
 - Must be technology-neutral

Source: <http://www.opensource.org/docs/osd>

Free software and open-source software - Examples

- **OSS with copyleft**
 - Linux kernel – Core of many operating systems
 - Blender – 3D creation suite
 - GNU Octave – Scientific computing (similar to MATLAB)
 - Nextcloud – Self-hosted collaboration platform
 - MariaDB – MySQL fork with GPL licensing
- **OSS without copyleft**
 - Apache HTTP Server – Popular web server (Apache License)
 - TensorFlow – Google's ML framework (Apache License 2.0)
 - Chromium – Open-source base of Google Chrome (BSD license)
 - OpenCV – Computer vision library (Apache License)
 - React.js – Meta's JavaScript library (MIT license)

Note: It's a bit more complicated than this as some of this software is available under multiple licences. More later on OSS licences.

Open source hosting platforms

Provide hosting, version control, issue tracking, documentation (wikis), and release/download management

Many also support code reviews, Continuous Integration/Continuous Delivery (CI/CD integration), and community collaboration

- Popular Platforms:

- GitHub – >330 million repositories (2024), largest OSS community
- GitLab – Open-core platform, offers private and public repo hosting
- Bitbucket – Popular with teams using Atlassian tools (e.g. Jira)
- SourceForge – >500k projects, popular for legacy and niche tools
- Codeberg – >200k - Privacy-focused Git hosting powered by Gitea (EU-based)

Github

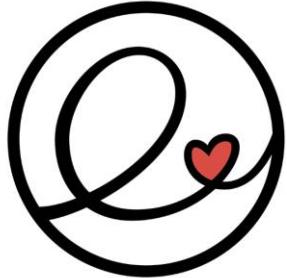
- Hosts over 330 million repositories (as of 2024)
- Popular projects: React, TensorFlow, Kubernetes, Next.js
- Free for public and private repositories with basic features
- Work together across unlimited private repositories with a paid plan.

The screenshot shows the GitHub repository page for 'twitter/bootstrap'. At the top, there's a navigation bar with links for 'Explore GitHub', 'Search', 'Features', and 'Blog'. On the right, there are buttons for 'Sign up for free' and 'Sign in'. Below the navigation, the repository name 'twitter/bootstrap' is displayed, along with a star icon (48,009 stars), a fork icon (14,257 forks), and a code icon. A summary card below the repository name states: 'Sleek, intuitive, and powerful front-end framework for faster and easier web development.' with a 'Read more' link and a link to 'http://twitter.github.com/bootstrap'. The main content area shows tabs for 'Code', 'Network', 'Pull Requests (68)', 'Issues (156)', 'Wiki', and 'Graphs'. The 'Code' tab is selected. It displays a list of files and their commit history. A merge pull request from 'waynn/patch-1' is shown at the top. The commit list includes:

- docs authored by mdo 23 days ago
- img authored by shiiit [fat] a month ago
- js authored by shiiit [fat] a month ago
- less authored by 2.3.1 [fat] a month ago
- .gitignore authored by rilian 8 months ago
- .travis.yml authored by conzett 8 months ago
- .dist directory added to .gitignore by fat 1 year ago

At the bottom right, it says 'latest commit 37d0a30589' and '1000+ commits'.

New Model for FOSS?



elementary OS community logo ELEMENTARY



- '**pay what you want**' model: Users can choose how much to pay, even \$0 (free)
- Championed by Foré and his team, it led to a massive increase in the company's income.
- Open-source projects remain free, but offer value that encourages support
- Challenges traditional assumptions in the FOSS community around monetization
- Encourages sustainability without restricting user access

<https://www.forbes.com/sites/jasonevangelho/2019/09/03/paying-for-linux-distros-and-foss-software-successful-elementary-os/#3ce63a5f3b3a>

Importance Of FOSS For Innovation

Growing importance of free and open-source software (FOSS)

- Widely adopted by companies for:
 - Internal IT infrastructure (e.g., Linux, Kubernetes)
 - Running web services (e.g., Apache, Nginx, Node.js, JBoss)
 - Developing cloud-native applications (e.g., Docker, Prometheus, Terraform)
 - Accelerating software development and innovation
- FOSS enables:
 - Faster prototyping
 - Community-driven improvements
 - Cost efficiency
 - Vendor neutrality

Open-Source usage in Enterprises 2024

- Widespread Adoption:
 - Over 95% of organizations have increased or maintained their use of open-source software in 2024.
- Active Participation:
 - A significant number of companies actively contribute to open-source projects, fostering innovation and collaboration.
- Primary Consideration:
 - Many enterprises prioritize open-source solutions over proprietary software during the decision-making process.

<https://www.statista.com/statistics/1419477/open-source-projects-adopted/>

Open source benefits for innovation

Industry aspects that benefit most from open source

1. INDUSTRY STANDARDS AND INTEROPERABILITY 62%

Clear potential for OSS to become the unifying force across complex, multi-vendor environments.

2. INNOVATION 59%

OSS provides a unique environment where diverse contributors can experiment and advance cutting-edge technologies.

3. REDUCED PRODUCT DEVELOPMENT COSTS 53%

OSS allows companies to build on existing, community-tested solutions rather than reinventing the wheel.

2024 WORLD OF OPEN SOURCE SURVEY, Q23, SAMPLE SIZE = 381

Source: Page 10 in https://www.linuxfoundation.org/hubfs/LF%20Research/2024GlobalSpotlight_121224.pdf?hsLang=en

Importance of FOSS in R&D and startups

- Most infrastructure used in R&D and startups relies on FOSS:
 - Operating systems (e.g. Linux)
 - Containers and orchestration (e.g. Docker, Kubernetes)
 - Configuration and provisioning (e.g. Ansible, Terraform)
- Most new software is built using FOSS tools and platforms:
 - Languages & runtimes (e.g. Python, Java, Node.js, Go, Rust)
 - Frameworks & libraries (e.g. Spring, Django, React, TensorFlow)
 - Build and CI/CD (e.g. Jenkins, GitHub Actions, GitLab CI)
- FOSS accelerates development and reduces cost:
 - Speeds up time-to-market
 - Enhances interoperability and compatibility
 - Enables teams to focus on innovation, not infrastructure

Some open-source business models

- Sell support and services
 - Example: Canonical (Ubuntu), Red Hat (RHEL)
- Sell certified version with enterprise tools
 - Example: Red Hat OpenShift, HashiCorp (Terraform Enterprise)
- Sell “enterprise edition” (proprietary features on top)
 - Example: MySQL Enterprise, Elastic (ElasticSearch)
- Dual licensing (copyleft triggers commercial license)
 - Example: The Qt Company (Qt framework), MariaDB
- Strategic platform control and ecosystem growth
 - Example: Google (Android), Meta (React.js)

Meta Open Source - Example



- Meta has open-sourced major tools like React, PyTorch, and LLaMA (Large Language Model Meta AI).
- These projects empower developers and researchers worldwide to build advanced AI and web applications faster.
- Open-sourcing fosters global collaboration, leading to rapid innovation, feedback, and community-driven enhancements.
- Meta gains from shared maintenance, broader adoption, and establishing leadership in AI and developer ecosystems.

Apple Open Source – Example



Swift

- Apple open-sourced Swift, its modern programming language, to expand its adoption beyond Apple platforms.
- Goal: Make Swift the best general-purpose programming language.
- By open-sourcing Swift, Apple encourages global developer contributions and accelerates language improvement.
- A wider developer base helps attract top talent and drives innovation in app and desktop development for Apple's ecosystem.

<https://swift.org/>

Microsoft Open Source – Example



Microsoft | Open Source

- .NET Core is open source and supports Windows, Linux, and macOS
 - Enables cross-platform development with a unified stack
- Over 2,000 active open-source projects maintained or supported by Microsoft
 - Includes VS Code, TypeScript, PowerShell, and more
- Strategic advantage for Microsoft Azure
 - Open-source friendliness attracts developers and enterprises
- More than 60% of Azure virtual machines now run Linux (up from 20% in earlier years)
 - Reflects strong support for diverse workloads

https://github.com/microsoftopensource.microsoft.com?utm_source=chatgpt.com

Open-source lab model



- UC Berkeley AMPLab: A multi-year collaborative research initiative (2011–2016)
 - It operated from 2011 to 2016, yet in just five years, it launched open-source tools like Apache Spark and helped seed major tech companies, demonstrating how time-limited academic initiatives can leave lasting industry impact.
- Industry Sponsors: Amazon, Google, IBM, SAP, Microsoft + 15 others
- Open-Source Innovations:
 - Apache Spark (cluster computing), Mesos (cluster manager), Tachyon (now Alluxio – distributed storage), GraphX (graph computation), MLBase (machine learning abstraction)
- Companies formed:
 - Mesosphere (\$122.25M invested), Databricks (\$47M invested), Tachyon Networks (\$17M invested)

Benefits of Industry–University Collaboration



- Companies benefit through:
 - Early access to cutting-edge technology
 - Influence over tech direction and outcomes
 - Reduced risk of being disrupted
 - Access to top student talent for recruitment
 - Strategic insights into emerging markets
- UC Berkeley benefits through:
 - Funding for advanced computer science research
 - Real-world problem alignment
 - Stronger ties with industry
 - Incubation of successful startups (e.g., Databricks)
 - Enhanced innovation reputation and ecosystem impact

Challenges – FOSS in products and services

- License Compliance
 - Ensuring the obligations of FOSS licenses (e.g. GPL, Apache) are fully understood and respected.
- Contamination Risk
 - Risk of accidentally including GPL-licensed code in proprietary software, which may require source code disclosure.
- Product Quality Assurance
 - Open-source code varies in quality; integrating it without rigorous testing may reduce product reliability.
- Security Vulnerabilities
 - Known or unpatched security issues in dependencies can expose products to risk.

[Open source software challenges predicted to continue in 2025](#)

Challenges – FOSS in products and services

- Need for Clear Open Source Policies
 - Companies must establish internal policies and controls to govern the use of FOSS responsibly.
- Governance Gaps
 - Gartner (2014) predicted that 50% of Global 2000 IT companies would implement open source governance programs, yet only 1 in 3 companies surveyed at the time had such a policy in place.
- Risks Without Governance
 - Lack of oversight can lead to license violations, security risks, and inconsistent usage practices.

Source: <http://blog.blackducksoftware.com/2011/07/29/policy-precedes-controls/>

Managing open source software

Organisations use specialised tools to track OSS usage, ensure license compliance, and align with internal OSS policies:

- Revenera (formerly Palamida)
 - Open source compliance and software composition analysis
 - <https://www.revenera.com>
- Synopsys Black Duck
 - Scans codebases for security and license risks
 - Offers a comprehensive knowledge base of OSS components
 - <https://www.synopsys.com/software-integrity/security-testing/software-composition-analysis.html>
- FOSSology
 - Open-source license compliance tool developed by HP
 - Continues as a community-driven project
 - <https://www.fossology.org>
- For a comprehensive list of tools - <http://fossology.org/>

FOSS Licenses

Usage and Licenses

Open Source Software – Obligations

- Obligations vary by license type, and may include:
 - No obligations at all
 - (e.g., public domain or very permissive licenses like MIT)
 - If redistributing the OSS as part of your product, you may need to:
 - Acknowledge the use of the open-source component
 - Share the source code of any modifications made
 - Include the original license text with your distribution
 - Refrain from enforcing patents on the OSS features (e.g., Apache 2.0's patent clause)
- Understanding license terms is crucial to avoid legal or compliance issues.

Public domain

- Work in the public domain does not have intellectual property rights.
- Examples
 - Shakespeare's works, Beethoven's music, old photos (expired copyright)
- Not commonly used for software :
 - Most recent activity.
 - some authors have started releasing software into the public domain.
Via CC0 license (Creative Commons Zero) which is a public domain dedication tool designed to allow creators to waive all rights to their work globally.
 - Legal limitations: Author can't fully disclaim rights, making true public domain dedication complex.

Open-source Software – License Types

- Open-source licenses approved by OSI (Open-source Initiative)
- Common licenses (widely used)
 - [Apache License 2.0 \(Apache-2.0\)](#)
 - [3-clause BSD license \(BSD-3-Clause\)](#)
 - [2-clause BSD license \(BSD-2-Clause\)](#)
 - [GNU General Public License \(GPL\)](#)
 - [GNU Lesser General Public License \(LGPL\)](#)
 - [MIT license \(MIT\)](#)
 - [Mozilla Public License 2.0 \(MPL-2.0\)](#)
 - [Common Development and Distribution License 1.0 \(CDDL-1.0\)](#)
 - [Eclipse Public License 2.0 \(EPL-2.0\)](#)



<https://opensource.org/licenses/category>

MIT License

- A short and simple permissive license
- Require preservation of copyright and license notices
- License works, modifications, and larger works may be distributed under different terms and without source code.

Permissions	Conditions	Limitations
<ul style="list-style-type: none">• Commercial use• Distribution• Modification• Private use	<ul style="list-style-type: none">• License and copyright notice	<ul style="list-style-type: none">• Liability• Warranty

<https://choosealicense.com/licenses/mit/>

Apache License 2.0

- Same as MIT license but
 - Additional “patent use” permission
 - Additional “state changes” condition

Permissions	Conditions	Limitations
<ul style="list-style-type: none">• Commercial use• Distribution• Modification• <u>Patent use</u>• Private use	<ul style="list-style-type: none">• License and copyright notice• <u>State changes</u>	<ul style="list-style-type: none">• Liability• <u>Trademark use</u>• Warranty

- Example: PDF.js

<https://choosealicense.com/licenses/apache-2.0/>

Mozilla Public License 2.0

- Same as Apache 2.0 but with the “same license (file)” condition

Permissions	Conditions	Limitations
<ul style="list-style-type: none">• Commercial use• Distribution• Modification• Patent use• Private use	<ul style="list-style-type: none">• <u>Disclose source</u>• License and copyright notice• <u>Same license (file)</u>	<ul style="list-style-type: none">• Liability• Trademark use• Warranty

<https://choosealicense.com/licenses/lgpl-3.0/>

GNU Affero General Public License(AGPL) v3.0

- Same as Apache 2.0 but
 - Additional conditions: “disclose code” and “network use is distribution”

Permissions	Conditions	Limitations
<ul style="list-style-type: none">• Commercial use• Distribution• Modification• Patent use• Private use	<ul style="list-style-type: none">• Disclose source• License and copyright notice• <u>Network use is distribution</u>• Same license• State changes	<ul style="list-style-type: none">• Liability• Warranty

<https://choosealicense.com/licenses/agpl-3.0/>

GNU General Public License GPLv3.0

- Same as AGPLv3.0 but
 - No “network use is distribution” condition

Permissions	Conditions	Limitations
<ul style="list-style-type: none">• Commercial use• Distribution• Modification• Patent use• Private use	<ul style="list-style-type: none">• Disclose source• License and copyright notice• Same license• State changes	<ul style="list-style-type: none">• Liability• Warranty

<https://choosealicense.com/licenses/gpl-3.0/>

GNU Lesser General Public License (GNU LGPLv3)

- Same as GPLv3.0 but
 - Additional “same license (library) condition

Permissions	Conditions	Limitations
<ul style="list-style-type: none">• Commercial use• Distribution• Modification• Patent use• Private use	<ul style="list-style-type: none">• Disclose source• License and copyright notice• <u>Same license (library)</u>• State changes	<ul style="list-style-type: none">• Liability• Warranty

<https://choosealicense.com/licenses/lgpl-3.0/>

Open Source Licenses – Summary

Permissive licences:

Changes need not be made available

Public domain MIT BSD Apache Software
License

Restrictive (copyleft) licences:

Changes must be made available

GPLv2 GPLv3 AGPL



- More restrictions/conditions on users
- More assurances of software staying free

Which license to use?

Understanding license types is critical for ensuring legal compliance, especially in commercial products.

Permissive licences:

Changes need not be made available

Public domain MIT BSD Apache Software
License

Restrictive (copyleft) licences:
Changes must be made available

GPLv2 GPLv3 AGPL



More adoption of your software
Not strong on sharing changes made by others
(deep expertise and knowledge)
=> use a permissive licence (eg BSD, Apache)

Users make their changes available of
your software:
=>use a restrictive licence (eg GPLv3)

Top Open Source licenses in 2024

- The most popular licenses include the MIT license, BSD licenses (3-clause and 2-clause), Apache 2.0 license, and GNU General Public license (2.0 and 3.0).
- These licenses continue to lead the way as the go-to choices for countless Open Source projects worldwide, reflecting their widespread adoption and versatility.

License	Pageviews	Visitors
mit	1.7M	1M
bsd-3-clause	247.9K	207K
apache-2-0	244.6K	184K
bsd-2-clause	115.8K	98.2K
gpl-2-0	83.8K	71.7K
gpl-3-0	72.4K	61.5K
isc-license-txt	48.7K	34.9K
lGPL-2.1	21.6K	19K
lGPL-3-0	17.8K	15.4K
OFL-1.1	16.9K	14.1K
mpl-2-0	14.8K	12.8K
postgresql	14.6K	12K
ms-pl-html	13.8K	9.9K

References

- Baldwin, Carliss, and Eric Von Hippel. "Modeling a paradigm shift: From producer innovation to user and open collaborative innovation." *Organization Science*, Vol 22, No 6, 2011.
- C. DiBona (ed) *Open Sources: Voices from the Open Source Revolution*, O'Reilly & Associates, 1999. Available online: <http://oreilly.com/openbook/opensources/book/>
- Karl Fogel, *Producing Open Source Software: How to run a successful free software project*, O'Reilly Media, 2005. Available free online: <http://producingoss.com/>
- R. Goldman and R. Gabriel, *Innovation Happens Elsewhere: Open Source as Business Strategy*, Morgan Kaufmann Publishers, 2005. Available free online: <http://www.dreamsongs.com/IHE/>
- Melissa A. Schilling, *Strategic Management of Technological Innovation*, 4th edition, McGraw-Hill, 2013.
- R. Stallman, The Cathedral and the Bazaar, <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>, 1999.
- E. von Hippel, *Democratizing Innovation*, MIT Press, 2005. . Available free online: <http://web.mit.edu/evhippel/www/democ1.htm>

Lecture: Distributed Innovation III – User Innovation, Platform Ecosystem & Intellectual Property Protection