Open Collaboration Guide

Daniel Antal

2021-03-27

Contents

Welcome

`I didn't have time to write a short letter, so I wrote a long one instead.` — *Mark Twain*

For collaborators who do not write code, the general Introduction contains a Simple Introduction focusing on documentation tools only. Contributors with no coding experience and ambition will likely work with data curation and documentation, publications which is equally important to developing code. You must be familiar with our developers naming conventions, because we automate research: their programs create data tables, visualizations, maps, blogposts, even books following this vocabulary.

Most of this long-form documentation is intended for collaborators who write code to acquire data, store data, create applications such a statistics, or machine learning applications, and who help us publish data.

Open collaboration is an agile project management method that breaks up the tasks to small, independent, decentralized acts that can be performed by individuals or research groups. It requires a solid shared knowledge base and a very high quality documentation. This collaboration guide serves two goals: to practice this documentation workflow, and to create a continuously improved documentation for our collaboration methods and practices.

1 Introduction

Provided that you use whatever IDE and Git/Github, there is unlikely that this guide will pose any challenge for you. Markdown and YAML are very simple, special-purpose markdown "languages" (the language is a big word here) that you can learn in 1-2 hours. They create marked up text, which we use for a hypertextual reference guide.

To collaboration on our software data products, there are other skills necessary, but for documentation and publication, technically you can get started with a clean text editor and git. We have a Simplified Introduction for you, which shows what can you skip in the rest of the document – almost everything. But do not skip this Introduction.

1.1 Definitions

- `Open Data` is data that is freely available to everyone to use and republish without legal or other restrictions. The most important sources of open data are open science data connected to scientific activities that allow the replication of scientific achievements. In Europe, the re-use of public sector information, in other jurisdictions, freedom of information regulations make various public institutions' and taxpayer funded datasets available for reuse. Open data is a very important source of information for business, scientific and policy uses.

- `Reproducible research`: The quality control of open data is focusing on reviewable, reproducible and confirmable findings. Auditability is a requirement in most high-level business, scientific or policy applications.
- `Open Source`: In most cases, when the data processing code and procedure is not a well-documented, open-source algorithm, reproducibility and confirmability is limited, or impossible.
- `Metadata`: It is a means by which the complexity of an object is represented in a simpler form. For example, the title, the author, and the cover art are metadata about a book. We use the distinction of descriptive, administrative, structural, preservation, and use metadata. See metadata.

1.2 Code of Conduct

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, caste, color, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

*If you work with us, you must adhere to the Contributor Covenant Code of Conduct

1.3 Collaboration Tools

1.3.1 Instant messaging: Keybase

Keybase is a very neat, simple, lightweight team management / chat / social networking application that is extremely focused on privacy, security and encryption.

Keybase Key features

- Secure instant messaging, even with a timed self-destruction feature (e.g. for sharing passwords); Starts a Google Meet or Zoom video call natively with a single command;
- Brings your Whatsapp chat to the more private and secure keybase chat on the fly;
- Team chat rooms in real time. You can filter where you want to be involved, and you can always opt-out;

-K-Drive (similar to OneDrive, Google Drive, Dropbox) – only for our team, and fully encrypted; Works with Github, and it even offers a more private version of Private Github Repos, encrypted gits; An integration with other platforms; It is neat, open source, simple, clean, and usually appreciated more in the open source community than Slack, its big corporation rival.

Practical steps you need to follow to use Keybase

1. Download & install Keybase from https://keybase.io/ on your computer.

An easy procedure. Create yourself a professional login name – similarly to a professional github account, a professional email, etc. (you cannot change the name afterwards)

2. Once you log in to the computer, go to *Devices*, and *Create a paper key*. Write this on paper, or print it, and store it somewhere very safe (not near your computer). This to recover the access in case you lose access to all your devices.
3. You can use Keybase simultaneously on multiple devices – Install Keybase on your smartphone, tablet or any other device. You will be guided through installation & paired with your computer.

4. Shall you need them, you have *two recovery options*: the paper key and your smartphone.
5. If your smartphone breaks down and needs a replacement, you can add from your computer your new phone and deactivate the old one.
6. Once you are in, look up `antaldaniel`. (Daniel is antaldaniel on Facebook, Twitter, Instagram, github, gmail, yahoo and basically every place where he was an early user, so you can connect with him on whatever channel you want.)
7. After a handshake Daniel will assist your smooth transition, help you find ways to our shared files, your project's files, and set up filters, so you are not flooded with information, while never left out, unless you choose to.
8. Initially, we set up the following "Big teams", as Keybase calls them, and we will send an invitation to join:

- `reprexscience` for our data science teams;
- `reprexdev` for developer(s), which may overlap with business development and science;
- `reprexbd` for business development, which may overlap with science;
- `reprexhumanities` for our creative team and data journalism
- `reprexfriends` for prospective team members, friends, and hoped-for-cooperation partners – partly for people we are discreetly asking to join us, or who want to know more about some of our work and cooperate with us;
- `reprexmanagement` for Istvan (general management) and Daniel (co-founder) - this is a closed team for now;
- `reprexmusic20` for our Music Professionals 2020 team.
- reprexcommunity is an open landing page for anybody, it is a public interface. If you every land there `antaldaniel` will take you to the appropriate, otherwise invisible team room.

Each big team has four special members for a smooth transition: Daniel and Zuzana to assist you with getting familiar with Keybase, zoombot (just type `!zoom` to create a Zoom call with the team members present) and meetbot (that does the same with Google Meet, `!meet`). Daniel will gradually withdraw from some of the teams, once their support is not needed, though each team will have at least one Reprex co-founder present. We invite everybody to at least one team, but you can sign up to as many as you like, shall you find that convenient.

9. Whenever you are in a situation you want to ignore us (e.g. because you sit in your dayjob), just do it. If you have a smartphone, we are there, separated from your Whatsapp friends, work emails, and you can always check on us. We can always send you a secure (and even encrypted) message to get in touch, if needed. However, we will never ever bother you with long emails, Whatsapp messages and other annoying things.

Let's keep things short, give access to the full picture when needed, and let you find out what mix of response time, details and filters works best for you.

1.3.2 Git & Github

Git is a simultaneous collaboration for for any distributed team work - writing, programming, design work. Git is an open source software which makes sure that your teamwork files are always synchronized, clashes are avoided (you modify the same part of a file at the same time with Daniel.) The only hard part to move to Git is to make sure that Git properly works on your computer - it needs to be installed differently on all Linux distros, Mac OSX version all Windows versions. On Windows, you must make sure that Git is on the startup path. Once you are there, you'll life will be much easier.

- Keybase allows the group work on encrypted documents, like business proposals simultaneously using Git synch.

- RStudio allows us to work simultaneously on business proposals, blog posts, templates via Git.
- Github is allows us to use shared folders (`repositories` or simply `repos`) where we can track changes, modify the same thing at the same time, avoid or resolve conflicting edits, assign tasks, and much more.

If you do not have a github account yet, please, sign up now on github.com. Create a very professional profile. It is likely that you will use this profile for future works for decades, as Git is really becoming the norm of digital nomads, freelancers, and tech teams to work together.

Github is not the only service platform that allows distributed, collaborative teamwork. It has many alternatives, for example, GitLab – don't confuse them. We use Github.

- dataobservatory-eu is our private repo collection and private github collaboration platform.

1.3.3 Rstudio IDE & other IDE

Our recommended IDE for documentation purposes is RStudio. You must install R, RStudio at least to make it work. For PDF outputs, you need to add a pdf compiler (recommended: tinytex, a lightweight tex compiler.) If you want to run Python code within RStudio, you need to have Pyhton installed on your computer, too. If you work in documentation and publishing, think about RStudio as a word processor that can save your work in html, pdf, Word, epub, keynote, PowerPoint, or any OpenOffice or Apple formats.

We work with data in the R and Python language, and we are open for C++, too. Our documentation is made in markdown to be able to produce html, pdf, word, powerpoint, reveal.js or any type of output. Our long-form documentation is knit together with the help of the 'bookdown', 'knitr', 'rmarkdown' packages in the R language – they combine YAML headers and enriched markdown that can contain R, Python or C++ executable code 'chunks', and our website use markdown, hugo, and the 'blogdown' connector between Go, R, and markdown.

This heterogeneous workflow is best served in RStudio. While you can write code in your favorite IDE in Python and/or C++, RStudio is unrivaled in its capability to connect various markup languages and several programming languages. Therefore, for documentation purposes, we use RStudio. You can contribute in any other editor – our longform documentation and our website content is, after all, marked-up text. If you need to execute something for within the documentation, only RStudio will do the trick.

1.4 Simple Introduction

> This part is intendend for collaborators who do not write regularly code, and do not use markdown, LaTeX in their work.

You will likely work with data curation and documentation, publications which is equally important to developing code.

1.4.1 Markdown

Markdown is a simple "language", or rather a writing notation system that lets the word processor know that `*italics*` means *italics*, or `**bold**` means **bold** or `## Writing, blogging` becomes a level 2 heading, and `### Markdown {#markdown}` is a level 3 heading that can be referenced in the table of contents or internal hypertext links with `[Jump to markdown introduction](#markdown)`.

Markdown makes it possible that we can work on documents that will render fine in `html`, `docs`, `pptx`, `pdf`, `google docs`, or `md` for markdown files.

Markdown is a “markup language”. But that is a too big word. It is more of a notation system for writing clear text that later can be automatically formatted.

For example `[Introduction to RMarkdown](https://rmarkdown.rstudio.com/lesson-1.html)` creates the hypertext link Introduction to RMarkdown in `html`, `docs`, `pptx`, `pdf`, `iWork Keynote`, `xlsx` or any file format that we need to create.

Markdown is very important for reproducible research and automation. It makes sure that the content and the form is fully separated.

- It is always the computer’s task to make the formatting work perfectly and beautifully.
- It is sometimes the computer’s task to fill out the document with text and numbers.
- It is a human task to desnbng beautiful documents, like blogposts, business proposals, research reports that are very easy to replicate automatically.

Markdown is not strictly a language, and it has many ‘flavors’ or notation version. The differences are usually related to the programming interface that allows the file conversion. If you are new to markdown, you can start with any flavor, the main text functions are the same.

- StackEdit is a wonderful tool, and we would recommend it, if it would not have been suspended from the Google Drive integration. You can try it out online. It is probably the cleanest interface to get you started.
- Heroku Markdown Editor integrates reasonably easily to your Google Drive. This means that you can edit our blogposts in your Google Documents.
- Docs to Markdown translates your Google Docs to Markdown. However, you must link your own images via a valid path.
- RStudio is our preferred offline application. It integrates seamlessly with Github. It is an integrated programming environment with four panels. If you use it as a markdown text editor, you can just minimize or close the programming tools.

RStudio uses the Rmarkdown, a special version of markdown, where you can insert little programs in `R`, `Python`, `C++`, `SQL`, `D3` or `Bash` scripts. For example, you can write a blogpost that retrieves data with a little program written by our musicology team from Spotify, or embeds a YouTube video, etc. The `code chunks` are visually separated from the proposal or blog post text, and you can ignore it if you do not *yet* write code.

- RMarkdown Cheat Sheet
- Rmarkdown Reference

1.5 Inspiration & Recommended Reading

1.5.1 Books

1.5.1.1 Why: Weapons of Math Destruction Weapons of math destruction, which O’Neil refers to throughout the book as WMDs, are mathematical models or algorithms that claim to quantify important traits: teacher quality, recidivism risk, creditworthiness but have harmful outcomes and often reinforce inequality, keeping the poor poor and the rich rich. They have three things in common: opacity, scale, and damage (**?**).

https://blogs.scientificamerican.com/roots-of-unity/review-weapons-of-math-destruction/

1.5.1.2 How: Metadata Metadata describes all the information that we collect, storeand disseminate in the forms of tables, maps, charts, apps, articles and books. Metadata is not data, it is the organization of data that is ready for use, publication, archiving or other purposes.

We follow the metadata definition and concepts of Pomerantz: Metadata from the MIT Press Essential Knowledge series (?). If you are a data scientist or an engineer, you will understand this book well. If you help us with data journalism, documentation, publications, this small book can help you understand the challenges we face when we want to make sure that our data will be easy to find, easy to use, transparent with the "small print." Read this short book by just skipping whatever you find technical. The first chapters of the book will save you countless of hours of misunderstandings.

It is not, Pomerantz tell us, just "data about data." It is a means by which the complexity of an object is represented in a simpler form. For example, the title, the author, and the cover art are metadata about a book. When metadata does its job well, it fades into the background; everyone (except perhaps the NSA) takes it for granted.

1.5.1.3 Critically: Data Feminism Critical attitude to working with with big data and AI: Data Feminism. This is a much celebrated book, and with a good reason. It views AI and data problems with a feminist point of view, but the examples and the toolbox can be easily imagined for small-country biases, racial, ethnic, or small enterprise problems. A very good introduction to the injustice of big data and the fight for a more fair use of data (?).

1.5.2 Blog posts & Podcasts

"big data increases inequality and threatens democracy." With Facebook's new trending topics algorithm and data-driven policing in the news, the book is certainly timely.

Any single episode of the now discontinued What's the Point Podcast: https://fivethirtyeight.com/features/introducing-fivethirtyeight-newest-podcast-whats-the-point/

- How a bad algorithm re-organized fire stations that hurt the black Bronx the most when the fire broke out: Why The Bronx Really Burned
- Who's Accountable When An Algorithm Makes A Bad Decision? Cathy O'Neil is most interested in meet three criteria: They are widespread, secret and unfair — the scoring methods that she says, for example, generate credit scores, keep someone in prison, or deny someone a job. On this week's What's The Point, O'Neil discusses her new book, "Weapons of Math Destruction"
- In Algorithms of Oppression, Safiya Umoja Noble challenges the idea that search engines like Google offer an equal playing field for all forms of ideas, identities, and activities. Data discrimination is a real social problem; Noble argues that the combination of private interests in promoting certain sites, along with the monopoly status of a relatively small number of Internet search engines, leads to a biased set of search algorithms that privilege whiteness and discriminate against people of color, specifically women of color. - submitted by ajgmolina

2 Data Curation

We are constantly looking for new data sources that may be interesting to our partners or for our own R&D activities.

3 Data Acquistion

Whatever is the source of the data we use it, we never trust it fully. We need check its strength and weaknesses, and bring it to a complete, documented and tidy form.

3.1 Metadata

Metadata plays an important role to find whatever we acquired and use it properly. It plays an important role during the storage of data in our databases and in documentation and publication, too. We placed it into a separate chapter.

3.2 Eurostat

3.3 Harmonized Survey Programs

3.4 Music APIs

3.4.1 Spotify API

3.4.2 Bandcamp

4 Data Storage & Databases

We may pursue several data storage philosophies.

Sometimes it is best to download data as it is, without processing, and saving it. These are the raw data assets. The raw data can be stored in a file system, or a very simple database.

Often it is not desirable to save the data in raw form, and we pre-process it to limit redundancy. For example, when downloading data from the Spotify API, it may be desirable to filter out redundant information, and just save marginal, new data. This data is best stored in some searchable, amendable database.

When we work with machine learning applications, we usually need tidy and processed individual data. When we work with business, policy, scientific analysis, we often need statistically aggregated (or disaggregated) data. Similarly to a machine learning application, the individual data goes through a complicated and error-prone software code until it gains its final form. The best view for a non-statistical data scientist to view our indicator creation process as an app itself. When we release a regional GDP dataset, we modify the input data in a very complicated process, and the published dataset is not an input (like a well-designed feature set) but a finished output.

The finished statistical output is a product, not a raw ingredient, and it is not desirable to store it in a database format. There are so many things that can go wrong with an indicator. Our statistical indicators are created by a code that is never finished. Whenever a new issue comes up, we add more processing code, new unit-tests. We constantly improve the statistically processed datasets, they need versioning, and often the best solution if the raw data is processed on the spot with the latest code.

We resisted databases for long, particularly centralized databases, because statistical end-products are at best created instantaneously with the best available data input, processing code and unit-testing.

Placing data in a database, particularly a not very well documented database goes against our core reproducible research principles. Strictly speaking, we must reveal our entire data process, from downloading a data point from the Spotify API to releasing a visualization, both input and processing code. A database adds a lot of complexity and renders the full process oversight unreadable. While technically reproducability may still be there, it is not inviting. (We can also reverse-engineer a lot of data processes, but we do not think that this is a real reproducible practice.)

We always must keep the right balance between the advantages of using a database and the advantages of a fully transparent and readable data workflow. We must balanace the documentation burden of a constantly updated database with keeping the process as a documentation and not hiding immediate results in a database.

4.1 Metadata

Metadata plays an important role in our databases and in documentation, too. We placed it into a separate chapter.

4.2 Raw data assets

4.3 Processed, individual data

4.4 Indicators - statistically processed data

4.5 Periodic data releases

4.6 Interactive data releases

4.7 Continous data releases in API

5 Applications

We process and release data to be used in various business, policy and scientific applications. In this collaboration guideline we do not want to document these applications - they are very specific to the problem, domain or client. The aim of this guideline to make the data collection, processing and dissemination workflow open for collaborators. The aim of this collaboration is to create high-quality, timely and well-documented data assets for various applications. We add application-specific information here to the extent that it helps data collaboration.

5.1 File, Variable Names, Value Labels

> There are only two hard things in Computer Science: cache invalidation and naming things. – Phil Karlton

We are mainly naming things when we name a path to a file (subdirectory and filename), create a database schema, and write code in any language. Of course, computers can translate between names, but the worst use of our time is to write code to rename things in a database to our code, or rename file names.

In recent reproducible science practice there have been many attempts to standardize variable naming. Because different (programming) languages have different well-established cultures, and naming things is a cognitive process, a full harmonization is not possible, but a high-level of harmonization already can save many-many hours in code writing, analysis and debugging.

5.1.1 Path

Because our final output is often an automatically created website (in the go language, using the hugo server application), we must be very aware of the entire path. A hugo website contains that website metadata in the path - the relative position of a file tells the server what is the aim of the file.

Our repos follow the naming conventions of peer-reviewed R packages. They can be extended to allow Python-specific elements.

Always present: `data-raw`: subfolder that contains data to be further processed `data`: contains well-processed, well-documented, re-usable, or publishable, releasable data. `R`: R scripts `plots`: saved static plots, maps `Python`: Python scripts `not_included`: your scrap files, environment files, *not to be synchronized on github.*

In R software releases only: `man` : not always present, `manual` code documentation files. `vignette`: only in R software releases, documentation file

In longform documentation only: `_book`: Only present in long-form bookdown publication, the current version of the books in different folders. (It is accompanied with a `_bookdown_files` folder) s.

In websites only: Hugo has its own relative path system.

5.1.1.1 R language

Consider the two identical meaning R code:

```
read.csv("data/my-csv.csv")

read.csv(file.path("data-raw", "my-csv.csv"))
```

On CRAN, only the latter is permissible for publication? Why? Because the various Linux distributions, Windows and Mac uses paths differently. You can never be sure that a file path will mean the same thing with 100% certainty, especially in the case of a full path. The full path is file-system dependent, and the three families of operational system use different ones.

The function `file.path()` in R makes sure that on any R-supported platforms your code will thy to read `my-csv.csv` from the relative `data-raw` path.

5.1.2 Variable nameing styles

We prefer the `snake_case_naming_convention` with lowercase letters. The names should omit stopwords, whenever possible, remain as short as possible. The organization of the name should help simple variable name selection rules, such as `starts_with()` or `ends_with()` in the tidyverse - this helps programming a great deal!

Consider the following scenario:

`spotify_artist_id` `spotify_song_id` `spotify_artist_name` `deezer_song_id` `deezer_artist_name` `bandcamp_song_id` `bandcamp_artist_name` `bandcamp_song_title`

Do you work with bandcamp data only:

`select ( starts_with("bandcamp"))`

Do you want to check how ID's match

`select ( ends_with("id"))`

Do you want to work with artist (person) data or song (sound recording) data as a unit?

`select ( contains("song"))`

`select ( contains("artist"))`

In R, we use `dbplyr` and the `tidyverse`, particularly `dplyr`, a non-standard evaluation of the language which makes R and SQL as interchangeable as possible. The `dbplyr` package allows the combination of queries to a database in SQL, R-dpylr, or even both in the same query.

Again, when updating/querying a database, the best if database columns need not to be renamed in the updating/processing code.

The `lower_snake_case` is a partly subjective choice. The lowercase is usually a very natural compromise between `UPPERCASE`, or `camelCase` naming conventions, and the renaming can happen with simple programmatic ways. In the R language we use the package `snakecase` for this, which translates a text or caption to any case you want - title case, camel case, snake case.

5.1.3 Character coding

Character coding is an unresolved problem of the world. We use `UTF-8` whatever it means, and we pray that our products will be readable on all platforms.

5.2 Statistical Processing & Indicators

All our critical processing code goes through anonymous peer-review on CRAN. We take pride in the fact that our software goes through dozens of unit-tests and a human review and it can be tested by anybody.

5.2.1 retroharmonize

The aim of retroharmonize is to provide tools for reproducible retrospective (ex-post) harmonization of datasets that contain variables measuring the same concepts but coded in different ways. Ex-post data harmonization enables better use of existing data and creates new research opportunities. For example, harmonizing data from different countries enables cross-national comparisons, while merging data from different time points makes it possible to track changes over time. (**?**)

It has two peer-reviewed releases.

5.2.2 regions

The regions package is an offspring of the eurostat package on rOpenGov. It started as a tool to validate and re-code regional Eurostat statistics, but it aims to be a general solution for all sub-national statistics. It will be developed parallel with other rOpenGov packages. (**?**)

It has one peer-reviewed release.

5.2.3 iotables

iotables processes all the symmetric input-output tables of the EU member states, and calculates direct, indirect and induced effects, multipliers for GVA, employment, taxation. These are important inputs into policy evaluation, business forecasting, or granting/development indicator design. iotables is used by about 800 experts around the world. (**?**)

It has several peer-reviewed releases.

5.3 Machine Learning Applications

5.3.1 Listen Local app

5.3.2 Bandcamp Librarian app

6 Documentation And Publications

6.1 Citations, Bibliography

For any external literature, data source, please, store the citation information in a well-formatted, thematic `.bib` files.

We create programatically `.bib` citation information files for our data products, software releases, and published documents.

```
@Manual{R-regions,
  title = {regions: Processing Regional Statistics},
  author = {Daniel Antal},
  note = {R package version 0.1.6},
  url = {https://regions.danielantal.eu/},
  year = {2021},
}

@Manual{R-retroharmonize,
  title = {retroharmonize: Ex Post Survey Data Harmonization},
  author = {Daniel Antal},
  note = {R package version 0.1.15},
  url = {https://retroharmonize.dataobservatory.eu/},
  year = {2021},
}
```

Marking up a reference to (?) with `[@R-retroharmonize]` will place the citation in your selected format (here APA) to the text, and to the references at the end of this documentation.

6.2 Software Releases

We have so far three released software products.

- The aim of retroharmonize is to provide tools for reproducible retrospective (ex-post) harmonization of datasets that contain variables measuring the same concepts but coded in different ways.
- The regions package is an offspring of the eurostat package on rOpenGov. It started as a tool to validate and re-code regional Eurostat statistics, but it aims to be a general solution for all sub-national statistics. It will be developed parallel with other rOpenGov packages.
- iotables processes all the symmetric input-output tables of the EU member states, and calculates direct, indirect and induced effects, multipliers for GVA, employment, taxation. These are important inputs into policy evaluation, business forecasting, or granting/development indicator design. iotables is used by about 800 experts around the world.

They are released on CRAN, and they follow the CRAN guidelines for unit-testing and human review. As CRAN relies on extensive automated testing, the formatting standard is *very strict* both for the software code and its documentation. The slightest deviation results in rejection.

6.3 Metadata

Metadata plays an important role in our databases and in documentation, too. We placed it into a separate chapter.

6.4 Data releases

We currently work with two platforms, and we must maintain compatibility with data repositories that our clients use. Both provide some validation, version control, and give a standard **doi** to our releases.

6.4.1 Dataverse

dataverse.org/ has a well-supported API, and this is the choice of our first academic partner, IViR.

The Dataverse Project is being developed at Harvard's Institute for Quantitative Social Science (IQSS), along with many collaborators and contributors worldwide. The Dataverse Project was built on our experience with our earlier Virtual Data Center (VDC) project, which spanned 1997-2006 as a collaboration between the Harvard-MIT Data Center (now part of IQSS) and the Harvard University Library. Precursors to the VDC date to 1987, comprising such entities as pre-web software to automatically transfer cataloging information by FTP to other sites across campus automatically at designated times, and before that to a stand-alone software guide to local data.

6.4.2 Zenodo

zenodo is the choice of the European Union, and it is likely that in the future all EU-financed research data must be published here.

The OpenAIRE project, in the vanguard of the open access and open data movements in Europe was commissioned by the EC to support their nascent Open Data policy by providing a catch-all repository for EC funded research. CERN, an OpenAIRE partner and pioneer in open source, open access and open data, provided this capability and Zenodo was launched in May 2013.

In support of its research programme CERN has developed tools for Big Data management and extended Digital Library capabilities for Open Data. Through Zenodo these Big Science tools could be effectively shared with the long-tail of research.

6.5 Publications

Our work is often embedded in publications. We want to make the application of various formatting guidelines as painless as possible. This is one of the main motivations to use markdown and Rmarkdown to document our work. There are more and more conversion tools that automatically convert our longform documentation from markdown or Rmarkdown to the formatting standards of almost any scientific publisher.

7 Metadata

We follow the metadata definition and concepts of Metadata from the MIT Press Essential Knowledge series.

It is not, Pomerantz tell us, just "data about data." It is a means by which the complexity of an object is represented in a simpler form. For example, the title, the author, and the cover art are metadata about a book. When metadata does its job well, it fades into the background; everyone (except perhaps the NSA) takes it for granted.

Pomerantz explains what metadata is, and why it exists. He distinguishes among different types of metadata—descriptive, administrative, structural, preservation, and use—and examines different users and uses of each type. He discusses the technologies that make modern metadata possible, and he speculates about metadata's future. By the end of the book, readers will see metadata everywhere. Because, Pomerantz warns us, it's metadata's world, and we are just living in it.

8 Contributor Covenant Code of Conduct

8.1 Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, caste, color, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

8.2 Our Standards

Examples of behavior that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people
- Being respectful of differing opinions, viewpoints, and experiences
- Giving and gracefully accepting constructive feedback
- Accepting responsibility and apologizing to those affected by our mistakes, and learning from the experience
- Focusing on what is best not just for us as individuals, but for the overall community

Examples of unacceptable behavior include:

- The use of sexualized language or imagery, and sexual attention or advances of any kind
- Trolling, insulting or derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or email address, without their explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

8.3 Enforcement Responsibilities

Community leaders are responsible for clarifying and enforcing our standards of acceptable behavior and will take appropriate and fair corrective action in response to any behavior that they deem inappropriate, threatening, offensive, or harmful.

Community leaders have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

8.4 Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is officially representing the community in public spaces. Examples of representing our community include using an official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

8.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported to the community leaders responsible for enforcement at [INSERT CONTACT METHOD]. All complaints will be reviewed and investigated promptly and fairly.

All community leaders are obligated to respect the privacy and security of the reporter of any incident.

8.6 Enforcement Guidelines

Community leaders will follow these Community Impact Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

8.6.1 1. Correction

Community Impact: Use of inappropriate language or other behavior deemed unprofessional or unwelcome in the community.

Consequence: A private, written warning from community leaders, providing clarity around the nature of the violation and an explanation of why the behavior was inappropriate. A public apology may be requested.

8.6.2 2. Warning

Community Impact: A violation through a single incident or series of actions.

Consequence: A warning with consequences for continued behavior. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.