Spreadsheet vs Database

Avoiding Data Disasters

Anne Pajon, CRUK-CI

Spreadsheet

The good...

- It's easy to **browse** data.
- It's easy to **manually enter and edit** data.
- It's easy to **share** copies of files.
- You have fine control over **visual presentation**.
- It has a very **flexible structure**.
- Formulas make it a **living document**.
- It has a **built-in suite of helpers** for charts, comments, spell checking, etc.
- It's relatively **easy to learn**.

The not so good...

- It **lacks data integrity**. Because every cell is unique, things can get very **inconsistent**. What you see doesn't necessarily represent the underlying data. A number is not necessarily a number. Data is not necessarily data.
- It's not very good for **working with multiple datasets** in combination.
- It's not very good for answering **detailed questions** with your data.
- It **doesn't scale**. As the amount of data increases, performance suffers, and the visual interface becomes a liability instead of a benefit. It also has fixed limits on how big a spreadsheet and its cells can be.
- **Collaborating is hard**. It's hard to control versions and have a "master" set of data, especially when many people are working on the same project.

Enter relational databases

What is relational database?

It consists of a “**server**” that stores all your data (think of a huge library) and a **mechanism for querying** it (think of a reference librarian).

The querying is where **SQL** comes in, SQL stands for **Structured Query Language**, and it is a syntax for requesting things from the database. It’s the language the reference librarian speaks.

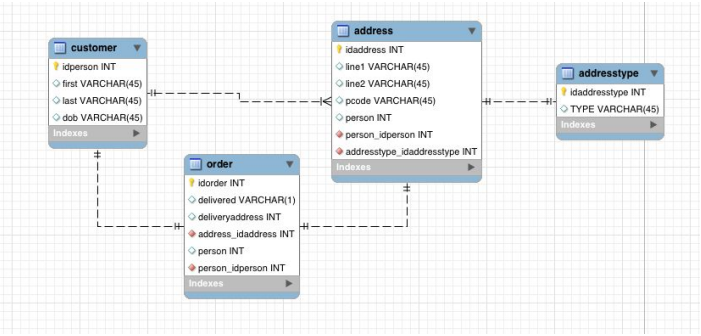
The “relational” part is a hint that these databases care about **relationships between data**.

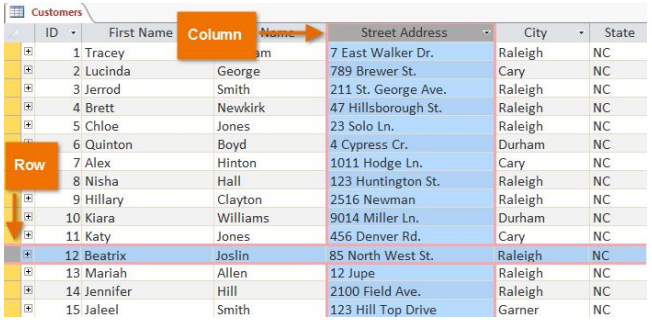
The database mantra

Everything in its proper place.

A database encourages you to **store things logically**. Sometimes it forces you to.

Every database consists of **tables and relationships** between them. Think of a table like a single worksheet in an Excel file, except with more ground rules. A database table consists of **columns** and **rows**.





Columns

Every column is given a **name** (like ‘Address’) and a defined **column type** (like ‘Integer,’ ‘Date’, ‘Date+Time’, or ‘Text’).

Columns define the structure of your data.

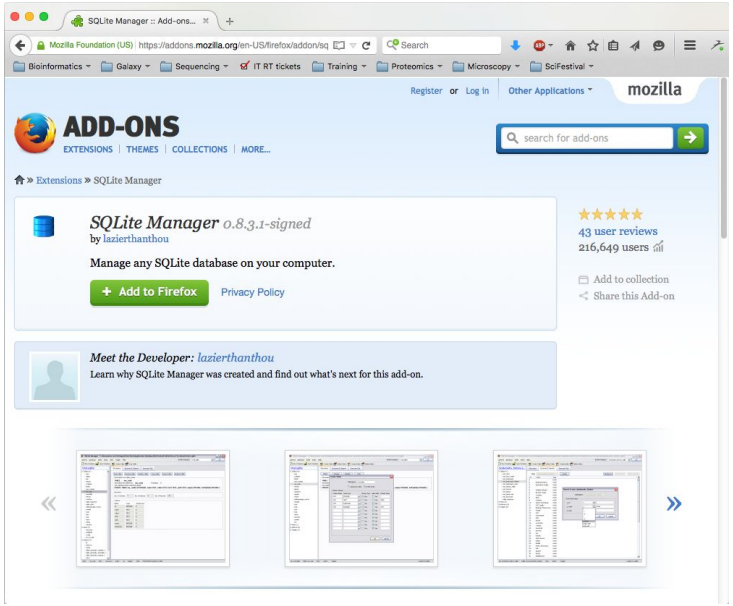
Rows

Rows are the actual data in the table. Once you establish the column structure, you can add in as many rows as you like.

Every row has a value for every column.

Where to start?

SQLite is a good way to get started. You can install the “**SQLite Manager**” addon for Firefox and do everything within the browser.



File Management

Avoiding Data Disasters

Anne Pajon, CRUK-CI

Use descriptive and informative **file names**

File Names ... Best Practices

- Do not name all your data files '**`data.xls`**'
- Include any information that will allow you to distinguish your files from one another
  - Project or experiment name or acronym
  - Location/spatial coordinates
  - Researcher name/initials
  - Date or date range of experiment
  - Type of data
  - Conditions
  - Version number of file
  - Three-letter file extension for application-specific files
- Choose a consistent naming scheme across all your files
- Include in the directory a '**`README.txt`**' file that explains your naming format along with any abbreviations or codes you have used

File Names ... Other Tips (1)

- **Avoid special characters** such as ~ ! @ # $ % ^ & * ( ) ` ; < > ? , [ ] { } ' " and |
- **Use short file names**, long ones do not work well with all types of software
- A **good format for date** designations is **`YYYYMMDD or YYMMDD`**
  - All of your files stay in chronological order, even over the span of many years
- Use **leading zeros** for clarity and to make sure files sort in sequential order
  - For example, use "001, 002, ...010, 011 ... 100, 101, etc." instead of "1, 2, ...10, 11 ... 100, 101, etc."

File Names ... Other Tips (2)

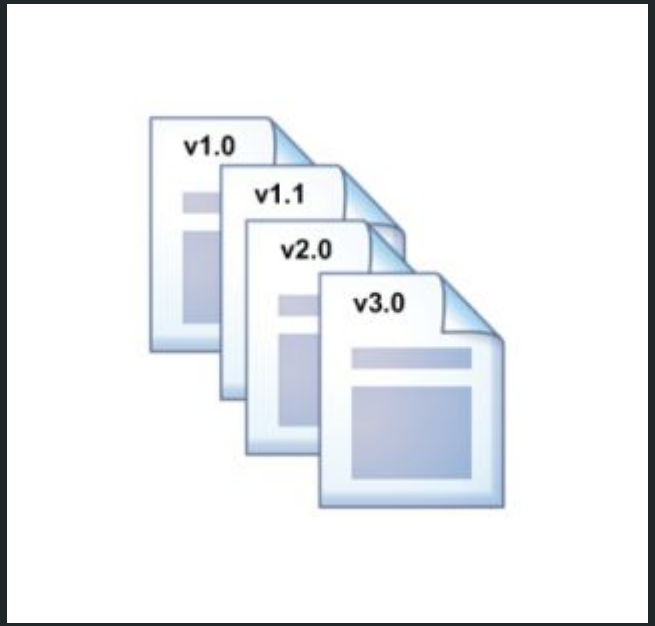
- **Do not use spaces**. Some software will not recognize file names with spaces, and file names with spaces must be enclosed in quotes when using the command line. Other options include:
  - Underscores, e.g. `file_name.xxx`
  - Dashes, e.g. `file-name.xxx`
  - No separation, e.g. `filename.xxx`
  - Camel case, where the first letter of each section of text is capitalized, e.g. `FileName.xxx`

Choose **file formats** that will ensure long-term access

File Formats ... Best Practices

- Save data in a **non-proprietary** (open) file format when possible
  - Usable on diverse platforms and by multiple applications
  - Export your data as tab separated file
- Unencrypted
- Uncompressed
- In common usage by the research community
- Preferred formats
  - Tabular data: CSV, TXT

Track different **versions** of your documents



Data versioning

Versioning refers to saving new copies of your files when you make changes allowing you to reverse or roll back those changes or retrieve specific versions of your files later

- Simple file versioning
- Simple software options
- Advanced software options

Simple File versioning

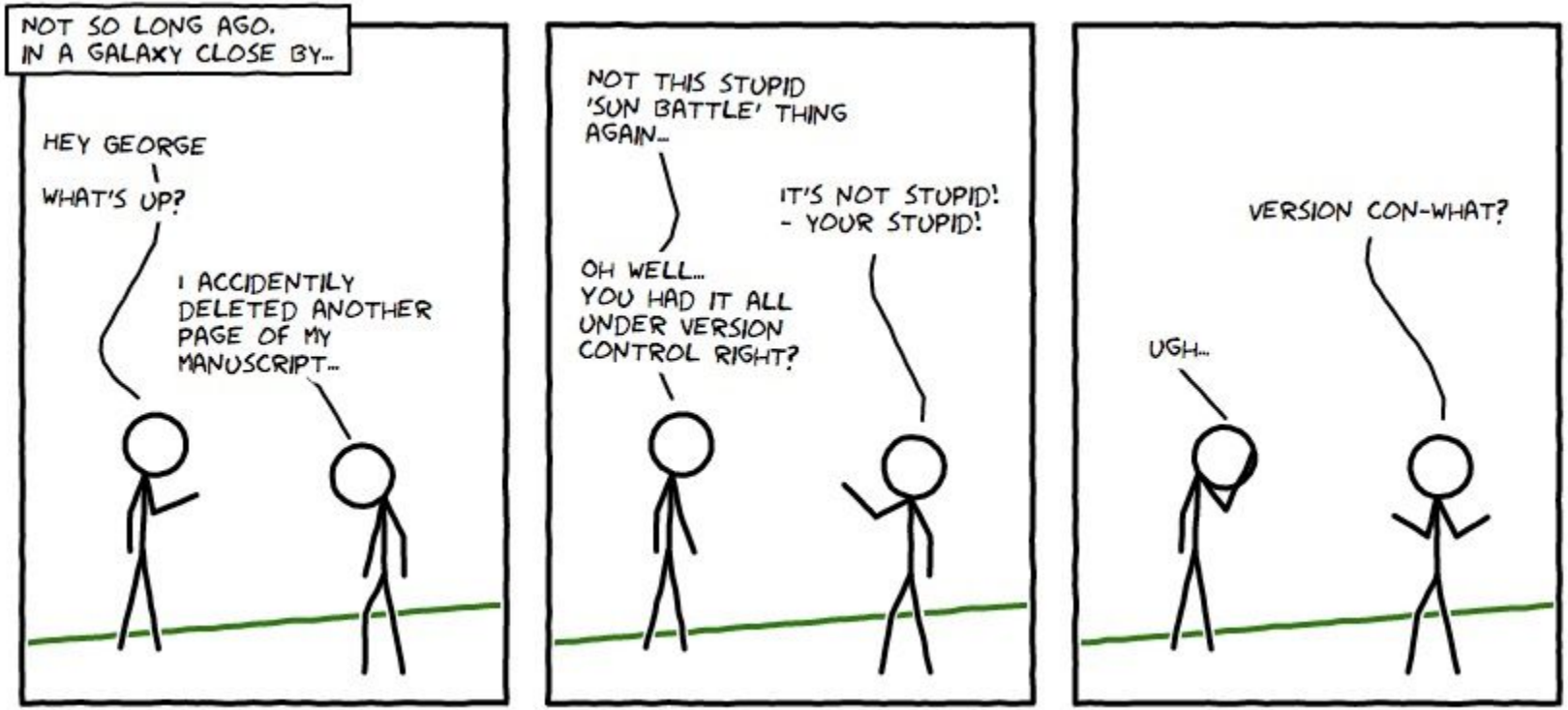
- Manually save new versions when you make significant changes
  - Include a version number, e.g. "v1," "v2," or "v2.1" into file names
- This works well, only if...
  - No need to keep lots of different versions
  - Only one person working on these files
  - Always access these files from one location

Simple Software Options

- Use Google Drive's word processing, spreadsheet and presentation
  - Any time you edit files, new versions are saved as you go
  - Version information includes who was editing the file and when the new version was created

- Dropbox free version keeps track of all of the changes you make for 30 days
  - The paid Packrat version keeps track of every change you ever make to the files in your Dropbox

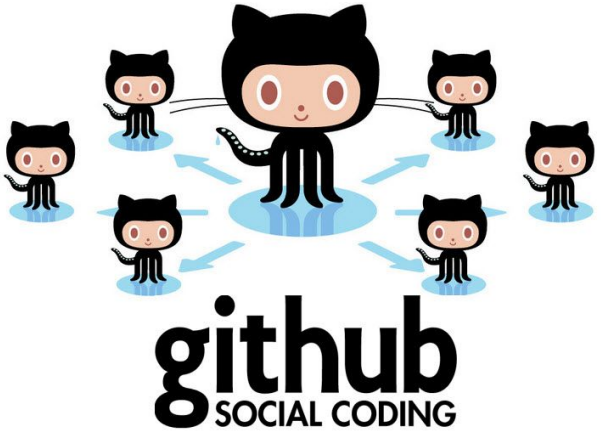
Advanced Software Options



Advanced Software Options

Version control systems like subversion and git are frequently used for groups writing software and code, but can be used for any kind of files or projects. Many people share their git repositories on GitHub.

Version control is the management of changes to documents, computer programs, and other collections of information. Changes are usually identified by a number named the "**revision number**".
Each revision is associated with a timestamp and the person making the change. Revisions can be compared, restored, and with some types of files, merged.



GitHub Desktop

Overview | Release Notes | Help

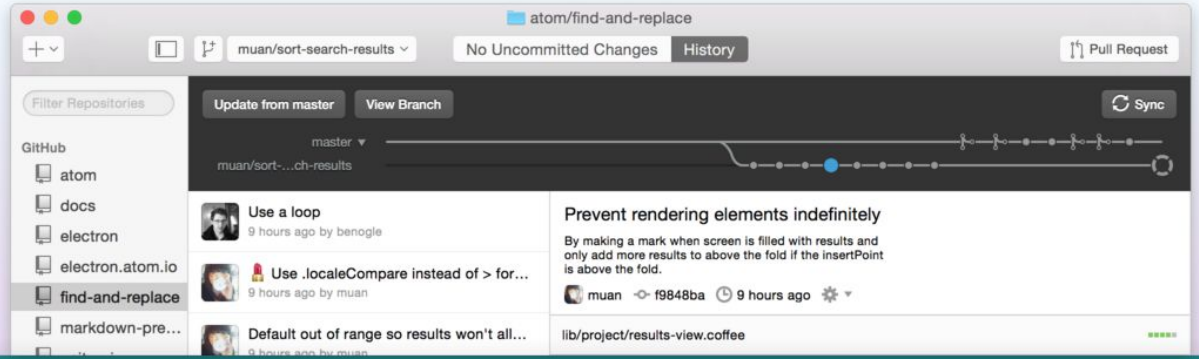
Simple collaboration from your desktop

GitHub Desktop is a seamless way to contribute to projects on **GitHub** and **GitHub Enterprise**.

Available for Mac and Windows

Download GitHub Desktop
OS X 10.9 or later

By clicking the Download button you agree to the End-User License Agreement



Your GitHub workflow in one native app

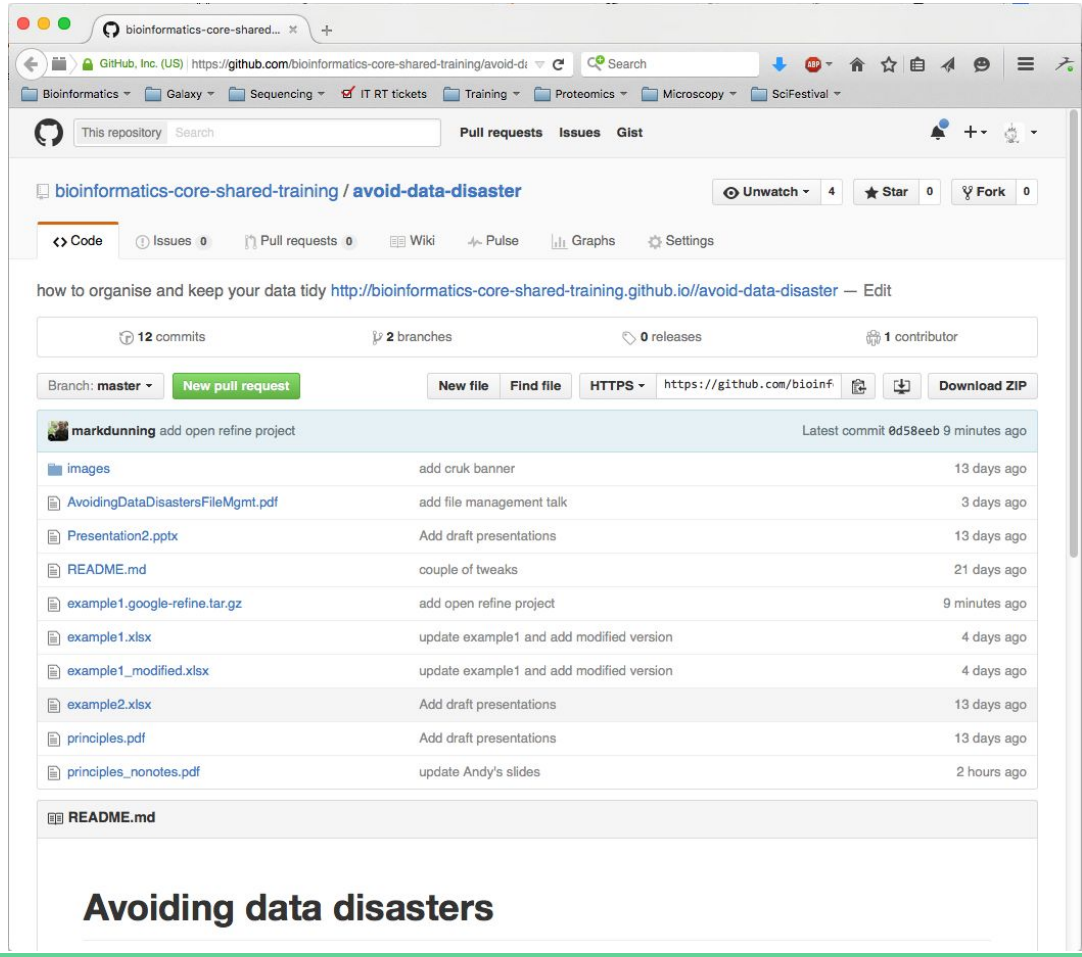
Clone repositories → Create branches → Commit changes → Share code

Training material on github

https://github.com/

Search for repository:

avoid-data-disaster
or bioinformatics-core-shared-training



Reference

- Data best practices STANFORD UNIVERSITY LIBRARIES
  - http://library.stanford.edu/research/data-management-services/data-best-practices

- Excel vs Databases SCHOOL OF DATA
  - http://schoolofdata.org/2013/11/07/sql-databases-vs-excel/