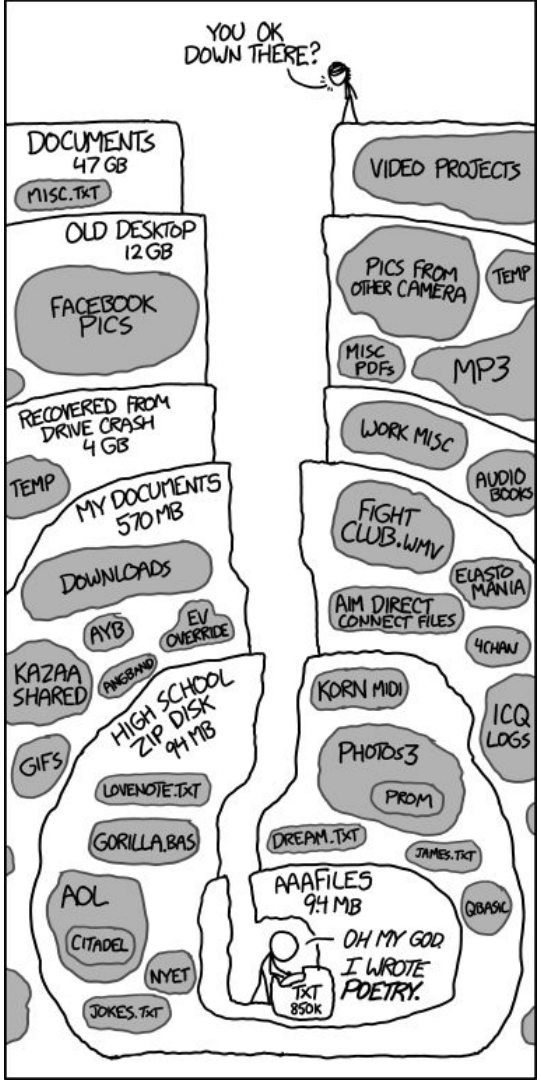
From Chaos to Order

Johannes Algermissen, Hannah Peetz, Eva Poort
Open Science Community Nijmegen


YOU OK DOWN THERE?
DOCUMENTS 47 GB
MISC.TXT
OLD DESKTOP 12 GB
FACEBOOK PICS
RECOVERED FROM DRIVE CRASH 4 GB
TEMP
MY DOCUMENTS 570 MB
DOWNLOADS
AYB
EV OVERRIDE
KAZAA SHARED
PAINGBAND
HIGH SCHOOL ZIP DISK 94 MB
GIFS
LOVENOTE.TXT
GORILLA.BAS
AOL
CITADEL
NYET
JOKES.TXT
VIDEO PROJECTS
PICS FROM OTHER CAMERA
TEMP
MISC PDFS
MP3
WORK MISC
AUDIO BOOKS
FIGHT CLUB.WMV
ELASTOMANIA
AIM DIRECT CONNECT FILES
4CHAN
KORN MIDI
ICQ LOGS
PHOTOS3
PROM
DREAM.TXT
JAMES.TXT
AAAFILES 9.4 MB
QBASIC
OH MY GOD I WROTE POETRY.
TXT 850K

Intro

Overview

1. Who are we?
2. Escape Room (15 min)
3. Naming things (10-15 min)
4. File management (10-15 min)
5. Sharing files (including your future self; 10-15 min)
6. Wrap-up exercise (15 min)
7. Questions

Who are we?

- Eva Poort is a post-doctoral researcher at the Max Planck Institute for Psycholinguistics. She's a fan of organisation.
  - Email: eva.poort@mpi.nl
- Hannah Peetz is a PhD student at the Behavioural Science Institute. She's trying to make up for past sins.
  - Email: h.peetz@psych.ru.nl
- Johannes Algermissen is a PhD student at the Donders Institute. He's German and also a fan of organisation.
  - Email: j.algermissen@donders.ru.nl

Or contact the OSCN directly: contact@openscience-nijmegen.nl

Escape Room

It's 11:45pm. You are sitting at home working on your final assignment for the course Research Project 3 (RP3). You are almost done when...

Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you. (0% complete)

If you'd like to know more, you can search online later for this error: **MEMORY_MANAGEMENT**

The last hour you have been busy fixing the small details but now all that work is lost.

But the way your assignment currently is, you can't hand it in. You have to at least fix a few issues.
The deadline is 11:59pm so you only have 14 minutes left to finish it.

Better get to work!

Luckily, when you open the Final Assignment ('Final Assignment.docx' under Documents > Uni > Bachelor > RP3) you see that your comments are still there.

You can find access to your folders here: https://tinyurl.com/ywmf4paa

You can ‘hand in’ your final assignment here: https://tinyurl.com/kyzvet9y

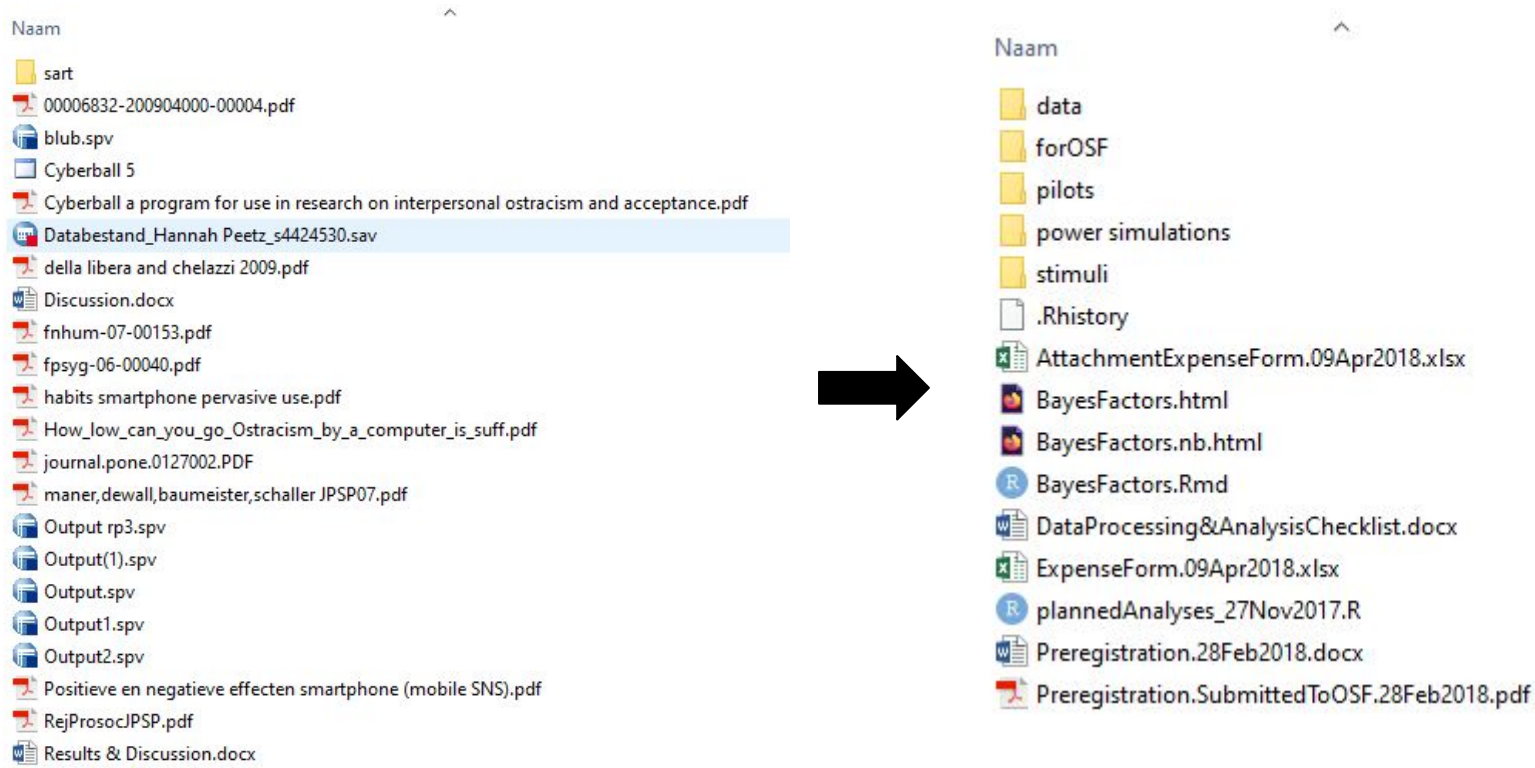
Solutions to the final assignment: https://tinyurl.com/2zmtu7mb

Naming things

Eva Poort

PROTIP: NEVER LOOK IN SOMEONE ELSE'S DOCUMENTS FOLDER.

Naming things: Goal



Naming things: Cardinal rules

1. Be consistent.
2. Use meaningful names.
3. Use names that make it easy to find the right file.

Take-home message: pick a convention and stick to it.

Naming files: Basic tips

- Decide what's important to include in the name and in which order.
  - For assignments: module, type of assignment, title, version.
  - For datafiles: participant number, type of data, session, date.
  - For articles: author's name(s), year of publication, article title.
  - The order should be based on how you think you will search through your files (e.g. by author, by date, by participant).
- Use case or separators to separate words or elements, never spaces.
  - Examples are lowerCamelCase, UpperCamelCase, hyphens (-) and underscores (_).
  - You can use different cases and/or separators for the different elements of your file name (e.g. Author1-et-al_year_ArticleTitle).
  - This helps with readability & batch processing of files.

⇒ Putting it all together:
  - PSYCH101_midTermEssay_Language-development-in-the-congenitally-blind_v01.docx.
  - pp001_raw_session1_20210806.csv
  - Gernsbacher_2018_Rewarding-research-transparency.pdf.

It can help to write your chosen convention down (in a README).

Naming files: More basic tips

- Try to avoid changing your mind.
  - The first name/convention tends to stick and subsequent names/conventions are difficult to remember.
- If you use numbers (e.g. for versions, participants), include a leading zero (001).
  - This ensures they are of the same length (so easy to spot the number) and listed sequentially.
- Don't include the folder name in the file name, unless there could be confusion later which folder a file belongs in.
  - This can happen when you have folders for raw data and for processed data, and files named simply with the participant IDs could result in mix-ups.

Naming files: Version control

- Include the date or a version number in the file name (or both, even).
  - If version, use a revision numbering system: indicate major changes with whole numbers (e.g. v01 becomes v02) and minor changes with ‘decimal’ numbers (e.g. v01.01 becomes v01.02).
  - If date, use the YYYYMMDD format. This ensures your files are listed chronologically.
- Before you edit a file, always save it under the new version number or date.
  - This is especially important for version numbers. Don’t rely on the “last opened/edited” column to tell you which is the most recent file.
- If you have many different versions, especially if versions are mixes of different pieces of content, create a ‘version control table’.
- Don’t use “final” in the file name unless it really is the final version.
  - If you need to edit a file with “final” in the name, before you make the edit, save the file again but replace “final” with today’s date or the next version number. Make your edit, then save it again with “final” in the name, overwriting the previous file. Don’t use “_final_v02”, etc.

Naming files: Interacting with your supervisor

- If you send an assignment to a lecturer or supervisor, include the module, assignment type and your name (or student number) in the file.
  - Don’t send Language-development-in-the-congenitally-blind_FINAL.docx.
  - Send PSYCH101_essay_EvaPoort_Language-development-in-the-congenitally-blind.docx
- If you get feedback from your supervisor, resave the file using the same name but including their initials.
  - You sent THESIS_Language-development-in-the-congenitally-blind_20210903.docx
  - Save as THESIS_Language-development-in-the-congenitally-blind_20210903_JD.docx.
  - When you start working on it again, rename it to a new version (date or number) THESIS_language-development-in-the-congenitally-blind_20210915.docx.

Naming folders: Basic tips

- Keep names short (15-20 characters).
- Describe the category of files that the folder will contain.
  - For an experiment: stimuli, experiment_scripts, raw_data, processed_data, merged_data, analysis_scripts, figures, write_up
  - For a module: lectures, assignments, summaries, notes, literature.
- If you have many old versions that make it difficult to spot the correct file, create an “old” folder.
  - Don’t delete these old files until you’re absolutely sure you won’t need them anymore (e.g. in three years time)!

Naming things: Tips for programming

- Use nouns for variables.
  - rawDataset
- Use verbs for functions.
  - processDataset
- Use descriptive names that are long if needed. Better a long name than a comment every time.
  - secondsElapsedSinceStart (not seconds).
- Make the name easy to type, pronounce and remember.
- Avoid abbreviations. Or define them at the start.
  - # Note: ACC = accuracy, # Note: LDT = lexical decision task.
- Use different case types and/or separators for different types of objects.
  - rawDataset, process_dataset, file.name

Naming things: Conclusion

Be consistent. Choose whatever works for you and stick to it.

File management

Hannah Peetz

File management: Goals

1. Easy to file
2. Easy to find
3. Reusable (aka consistency is key)

General message: Find what works for you and stick to it

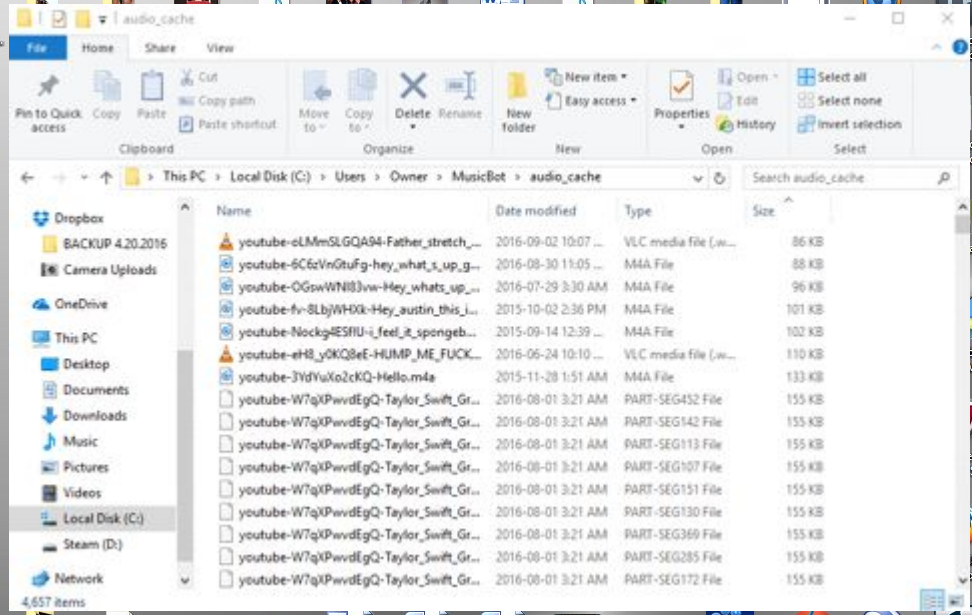
File management: Where

Storing files on:

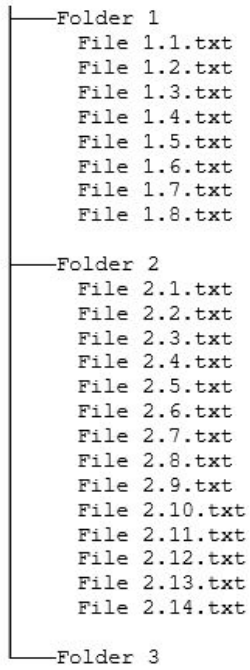
- Desktop?
- Downloads folder?
- To be sorted folder?

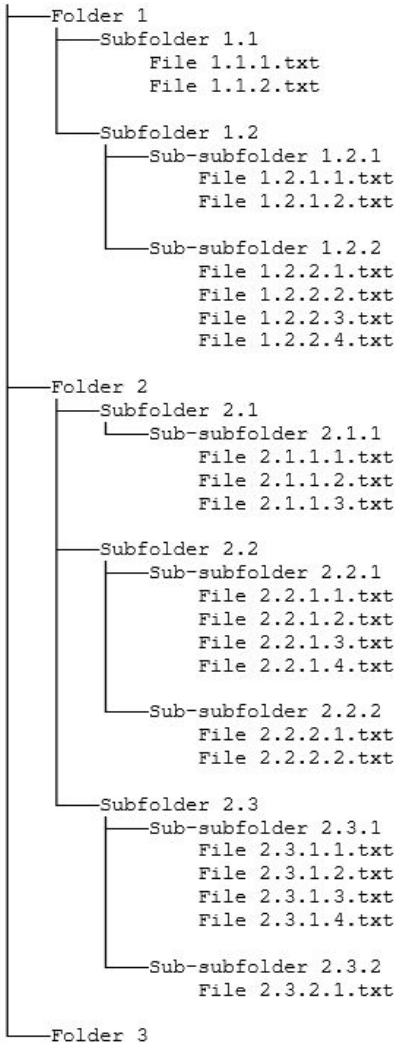
Easy to file 

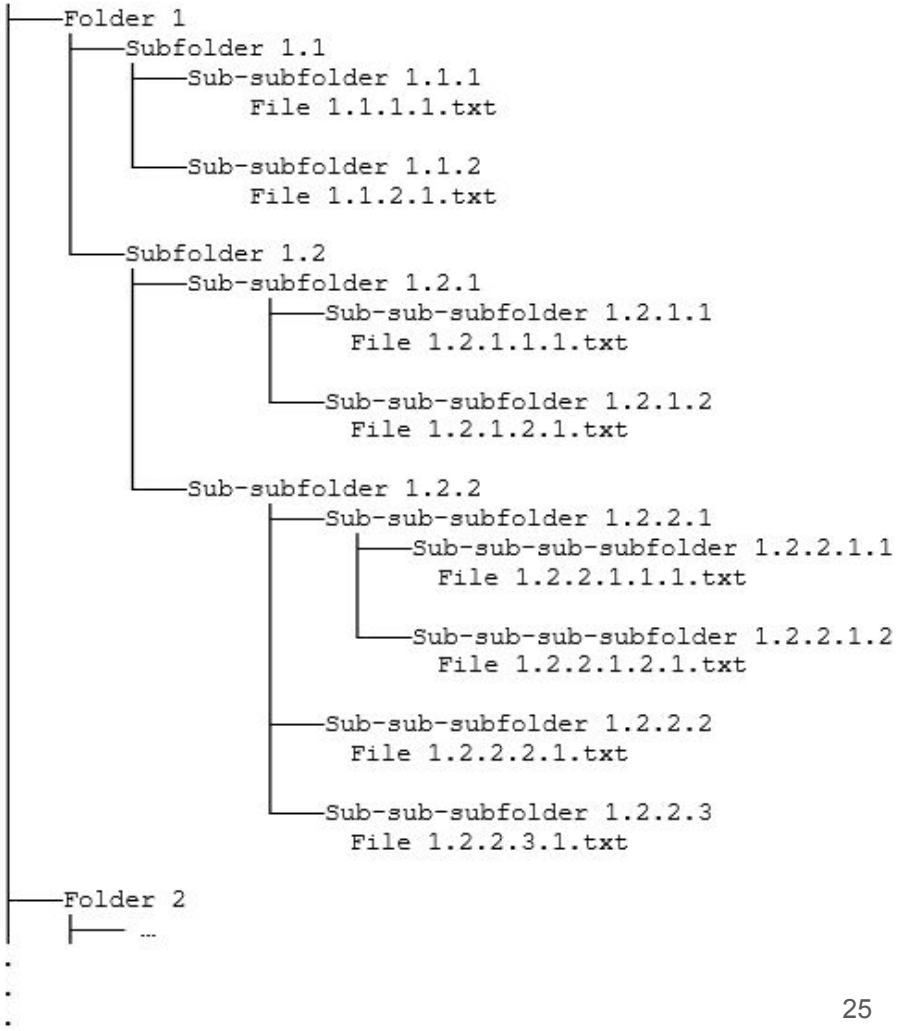
Easy to find 


audio_cache
File
Home
Share
View
Pin to Quick access
Copy
Paste
Cut
Copy path
Paste shortcut
Clipboard
Move to
Copy to
Delete
Rename
Organize
New folder
New item
Easy access
New
Properties
Open
Edit
History
Open
Select all
Select none
Invert selection
Select
This PC › Local Disk (C:) › Users › Owner › MusicBot › audio_cache
Search audio_cache
Dropbox
BACKUP 4.20.2016
Camera Uploads
OneDrive
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Steam (D:)
Network
Name
Date modified
Type
Size
youtube-cLMmSLGQA94-Father_stretch_... 2016-09-02 10:07 ... VLC media file (.w... 86 KB
youtube-6C6zVnGtuFg-hey_what_s_up_g... 2016-08-30 11:05 ... M4A File 88 KB
youtube-OGswWNI83vw-Hey_whats_up_... 2016-07-29 3:30 AM M4A File 96 KB
youtube-fv-8LbjWHXk-Hey_austin_this_i... 2015-10-02 2:36 PM M4A File 101 KB
youtube-NockgAESfIU-i_feel_it_spongeb... 2015-09-14 12:39 ... M4A File 102 KB
youtube-eH8_yOKQ8eE-HUMP_ME_FUCK... 2016-06-24 10:10 ... VLC media file (.w... 110 KB
youtube-3YdYuXo2cKQ-Hello.m4a 2015-11-28 1:51 AM M4A File 133 KB
youtube-W7qXPwvdEgQ-Taylor_Swift_Gr... 2016-08-01 3:21 AM PART-SEG452 File 155 KB
youtube-W7qXPwvdEgQ-Taylor_Swift_Gr... 2016-08-01 3:21 AM PART-SEG142 File 155 KB
youtube-W7qXPwvdEgQ-Taylor_Swift_Gr... 2016-08-01 3:21 AM PART-SEG113 File 155 KB
youtube-W7qXPwvdEgQ-Taylor_Swift_Gr... 2016-08-01 3:21 AM PART-SEG107 File 155 KB
youtube-W7qXPwvdEgQ-Taylor_Swift_Gr... 2016-08-01 3:21 AM PART-SEG151 File 155 KB
youtube-W7qXPwvdEgQ-Taylor_Swift_Gr... 2016-08-01 3:21 AM PART-SEG130 File 155 KB
youtube-W7qXPwvdEgQ-Taylor_Swift_Gr... 2016-08-01 3:21 AM PART-SEG369 File 155 KB
youtube-W7qXPwvdEgQ-Taylor_Swift_Gr... 2016-08-01 3:21 AM PART-SEG285 File 155 KB
youtube-W7qXPwvdEgQ-Taylor_Swift_Gr... 2016-08-01 3:21 AM PART-SEG172 File 155 KB
4,657 items

File managem er







File management: Making a folder structure

Many possibilities: Organize by year, topic, project, …

One possibility:

- First split into big overarching categories (e.g. education, work, personal)
- Then organize by time (e.g. year, quarters)
- Then organize by project (e.g. course A, tutoring materials, bills apartment)

Some handy folders for projects:

- Working
- Final
- Archive
    - Overarching archive folder for completed projects (zip to save space)

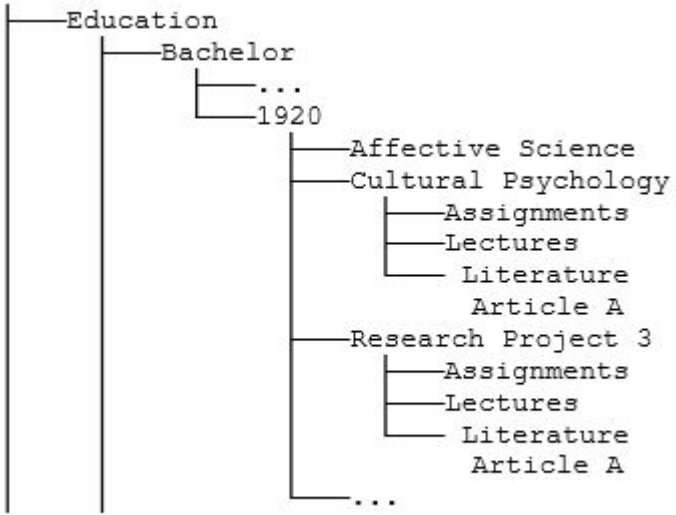
File management: Templates

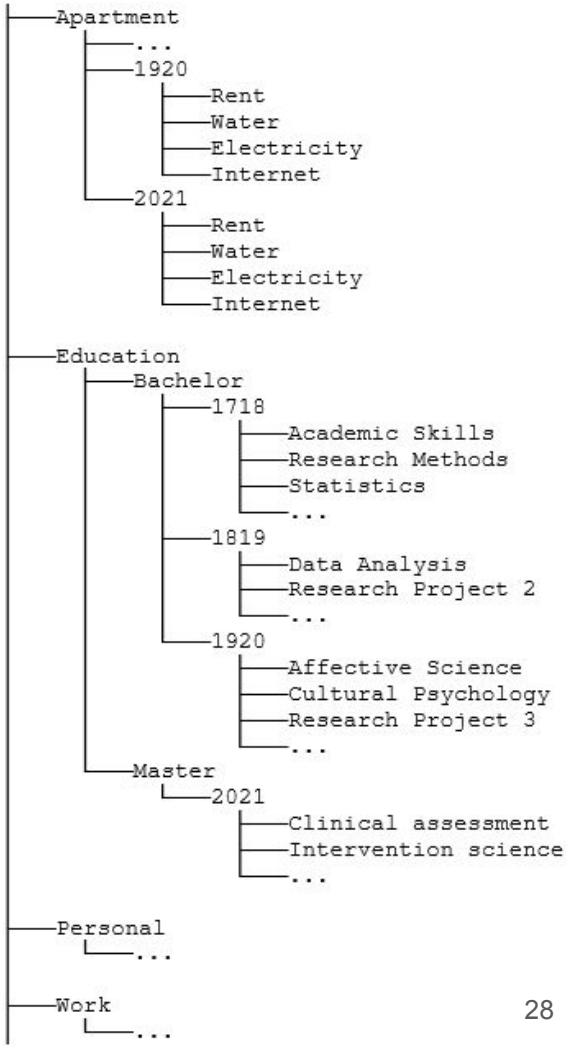
Don’t want to invent a new system for each new project…

… use templates

Reusable & Consistent

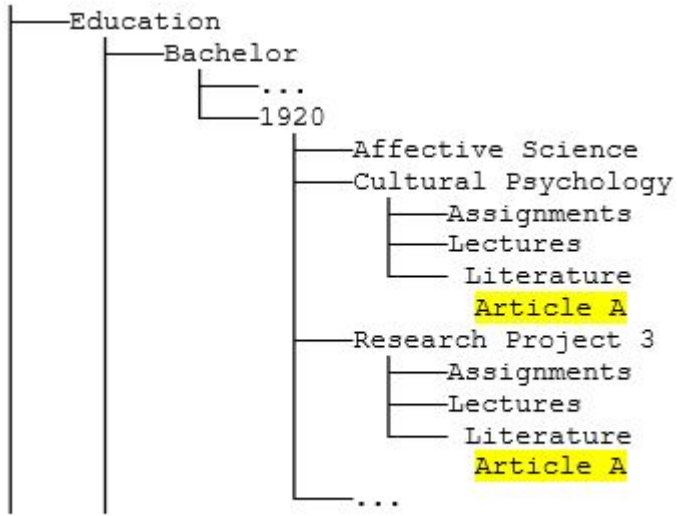
File management: Example





File management: Duplicates

Everything needs *one* place

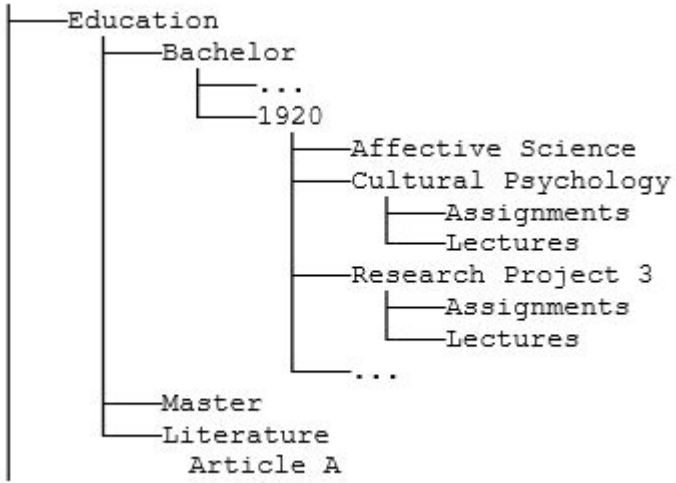


File management: Duplicates

Everything needs *one* place

Multiple places?

Multiple solutions

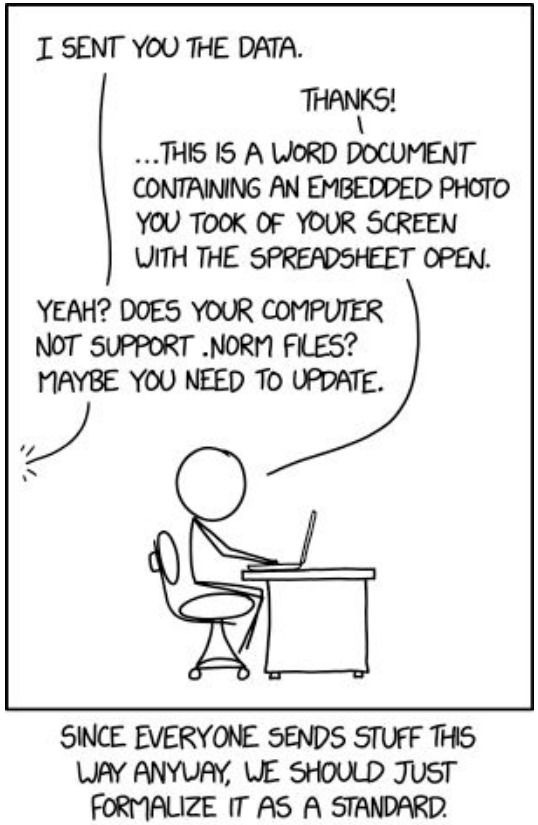


File management: Additional tips

- Pin it
- Tag it

Other tools & software:

- DropIt or Zotero
- Literature managers (e.g. Mendeley)



Sharing things with others (including your future self)

Johannes Algermissen

Rule no. 1: ***Comment*** what you do!

- There’s hardly ever too many comments…
- Focus less on technical terms
  (“*use function x with input y to get z*”),
  but rather what potential problem you want to solve
  (“*check for RTs that are too long, mark those events in new variable*”)

Different working stations - what can go wrong?

- Setting a **root directory**
  - `rootdir = "E://mytask/"`
  - Initialize all other directory ***relative*** to that directory
  - In R: `datadir = paste0(rootdir,"data/")`
- Load supporting **packages/libraries**
  - Check first if packages are actually installed
  - In R: `if(!require(somepackage)){install.packages("somepackage");library(somepackage)}`
  - Next level: load exact version needed… see groundhog, e.g. http://datacolada.org/95
- Check what steps have already been performed
  - First create **name of output file** of a certain step
  - Check if file already exists; only perform step if it does not exist
  - In R: `outputfile = paste0(datadir,"RTs_aggr_sub_cond.csv")); if(!(file.exists(outputfile)){...}`
  - Requires deliberate step to overwrite old files!

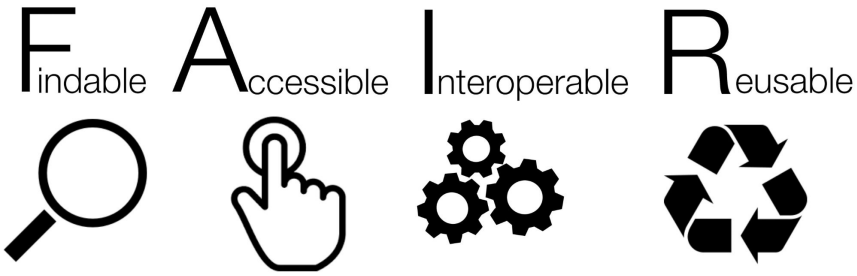
Human-readable scripts

- A) At the beginning:
  - Load **packages/ libraries** at the beginning
    - Check whether installed, if not install
    - Pay attention to the order in which you load them
  - **Global variables**: directories, plotting defaults (font size, line width), mathematical constants, ...
    - (participant number, participant peculiarities, variant of analysis, a.k.a. “suffix”)
    - Do “hard-coding” of information that can change (subject number) only **once** at the beginning (“Don’t repeat yourself” a.k.a. DRY principle)
- B) Outsource repeated steps into **functions** in an external script
  - Packages, directories, global variables,... “**configuration** script” (config)
  - Run config script at the beginning of each file
    - Change default at one spot → use it in every other relevant script
  - General structure of a function: Load data, transform data, save data

Documenting data/ scripts

- README files
  - Data dictionaries
  - Codebox
  - Codebooks
- No common standards/conventions…
- Minimum:
  - 1. Annotate **variables**, what they mean, what their values mean (units!)
  - 2. Annotate **extraordinary events** (incomplete data, data to be excluded)
  - 3. Explain **order** in which scripts have to be run
    - Or use a master script…

Open Data



Findable

Make your data available on repository with a persistent identifier (DOI, handle) and metadata

Accessible

Be explicit about data usage terms (agreement with downloader)

Interoperable

Make your data human and machine readable, e.g. BIDS

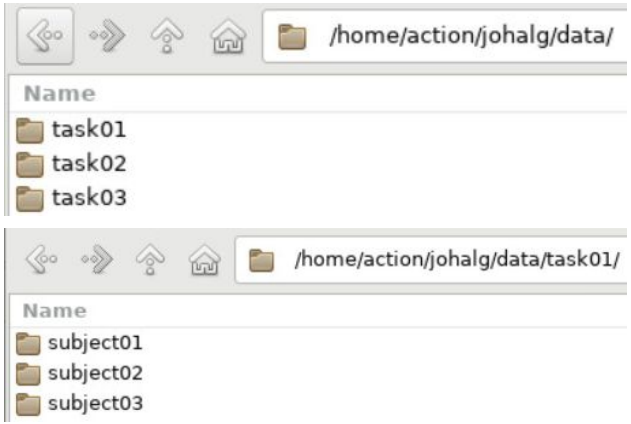
Reusable

Make sure you document enough details, e.g. “data descriptor” paper

this can be cited, along with citing our data -> measurable impact!

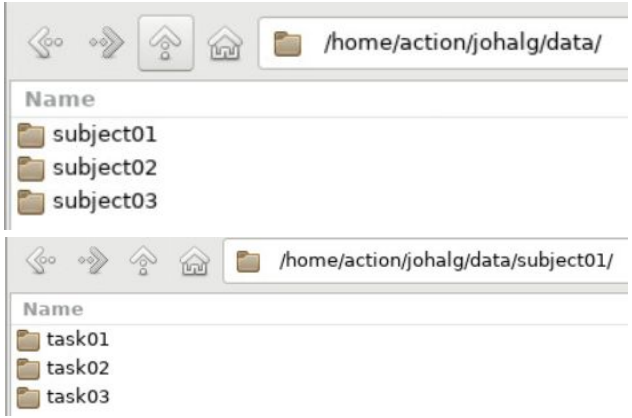
Machine-readable data: BIDS

- Collecting data from multiple participants from multiple tasks



OR

?



- BIDS (neuroimaging): https://bids.neuroimaging.io/
- Conventions for file/ folder **naming** and folder **organization**
- "**Sidecar**" files:
  - .json file: relevant information on data acquisition, e.g. MR scanner, computer OS, ...
  - .tsv file: demographics such as participant age, sex, … , unusual events
- Several *data2bids* and *bidsvalidator* apps available

Example structure in BIDS format

```
ds001
├── dataset_description.json
├── participants.tsv
├── sub-01
│   ├── anat
│   │   ├── sub-01_inplaneT2.nii.gz
│   │   └── sub-01_T1w.nii.gz
│   └── func
│       ├── sub-01_task-balloonanalogrisktask_run-01_bold.nii.gz
│       ├── sub-01_task-balloonanalogrisktask_run-01_events.tsv
│       ├── sub-01_task-balloonanalogrisktask_run-02_bold.nii.gz
│       ├── sub-01_task-balloonanalogrisktask_run-02_events.tsv
│       ├── sub-01_task-balloonanalogrisktask_run-03_bold.nii.gz
│       └── sub-01_task-balloonanalogrisktask_run-03_events.tsv
├── sub-02
│   ├── anat
│   │   ├── sub-02_inplaneT2.nii.gz
│   │   └── sub-02_T1w.nii.gz
│   └── func
│       ├── sub-02_task-balloonanalogrisktask_run-01_bold.nii.gz
│       ├── sub-02_task-balloonanalogrisktask_run-01_events.tsv
│       ├── sub-02_task-balloonanalogrisktask_run-02_bold.nii.gz
│       ├── sub-02_task-balloonanalogrisktask_run-02_events.tsv
│       ├── sub-02_task-balloonanalogrisktask_run-03_bold.nii.gz
│       └── sub-02_task-balloonanalogrisktask_run-03_events.tsv
...
...
└── task-balloonanalogrisktask_bold.json
```

Now it’s your turn!

Now it’s your turn!

In break-out rooms, discuss:

- What have you struggled with the most so far?
- What’s the biggest change you can make to improve your own file management?

Q & A

More resources

- Eglen, S. J., Marwick, B., Halchenko, Y. O., Hanke, M., Sufi, S., Gleeson, P., ... & Poline, J. B. (2017). Toward standard practices for sharing computer code and programs in neuroscience. *Nature Neuroscience*, 20(6), 770-773. https://www.nature.com/articles/nn.4550
- van Vliet, M. (2020). Seven quick tips for analysis scripts in neuroimaging. *PLOS Computational Biology*, 16(3), e1007358. https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007358
- Slide deck by Danielle Navarro on project structure: https://slides.djnavarro.net/project-structure/#1
- Slide deck by Jenny Bryan on “naming things”: https://speakerdeck.com/jennybc/how-to-name-files
- Harvard tips for naming files: https://datamanagement.hms.harvard.edu/collect/file-naming-conventions.
- BIDS: https://bids.neuroimaging.io/ and https://bids-specification.readthedocs.io/en/stable/
- Fieldtrip tutorial on “Converting an example behavioral dataset for sharing in BIDS”: https://www.fieldtriptoolbox.org/example/bids_behavioral/