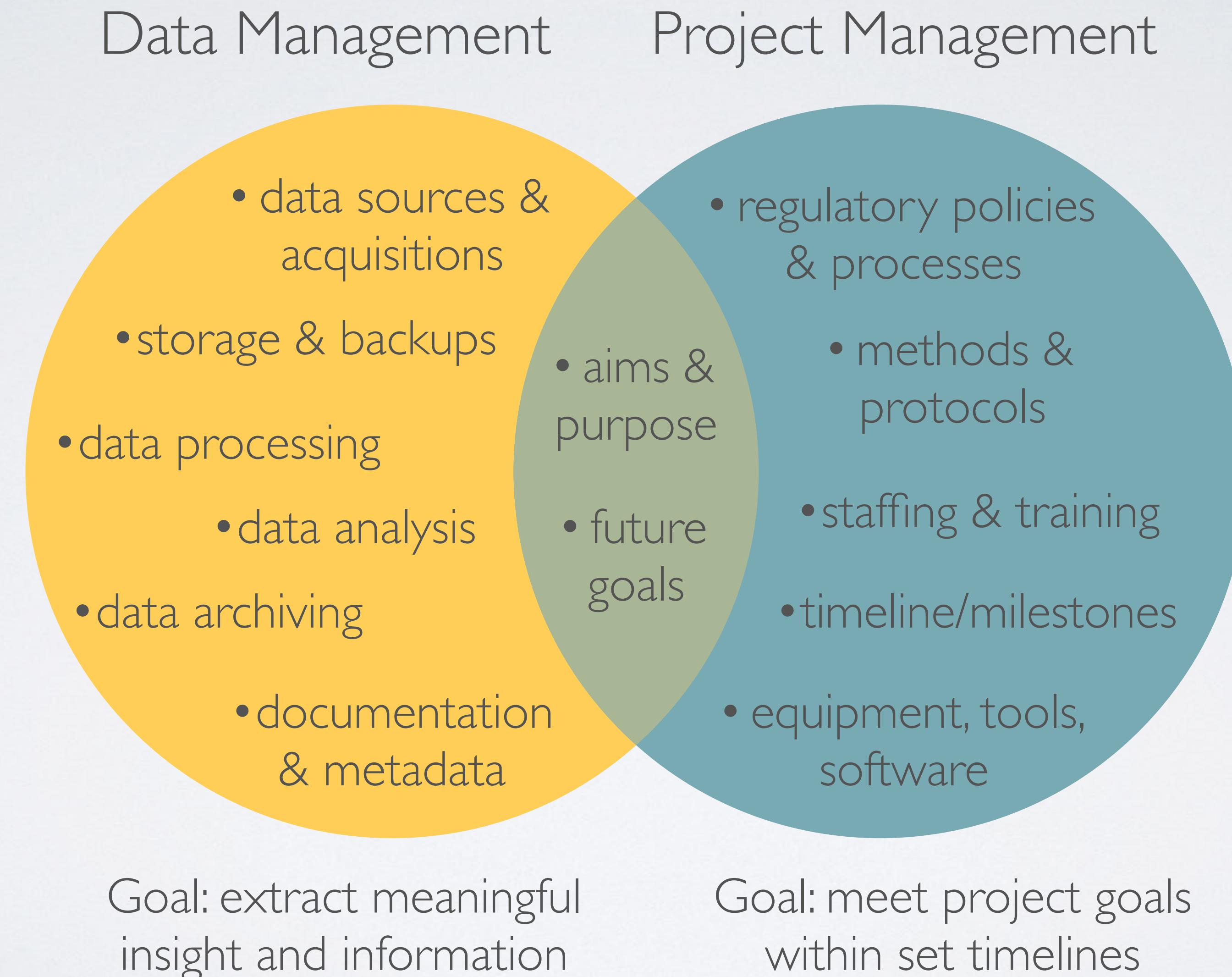
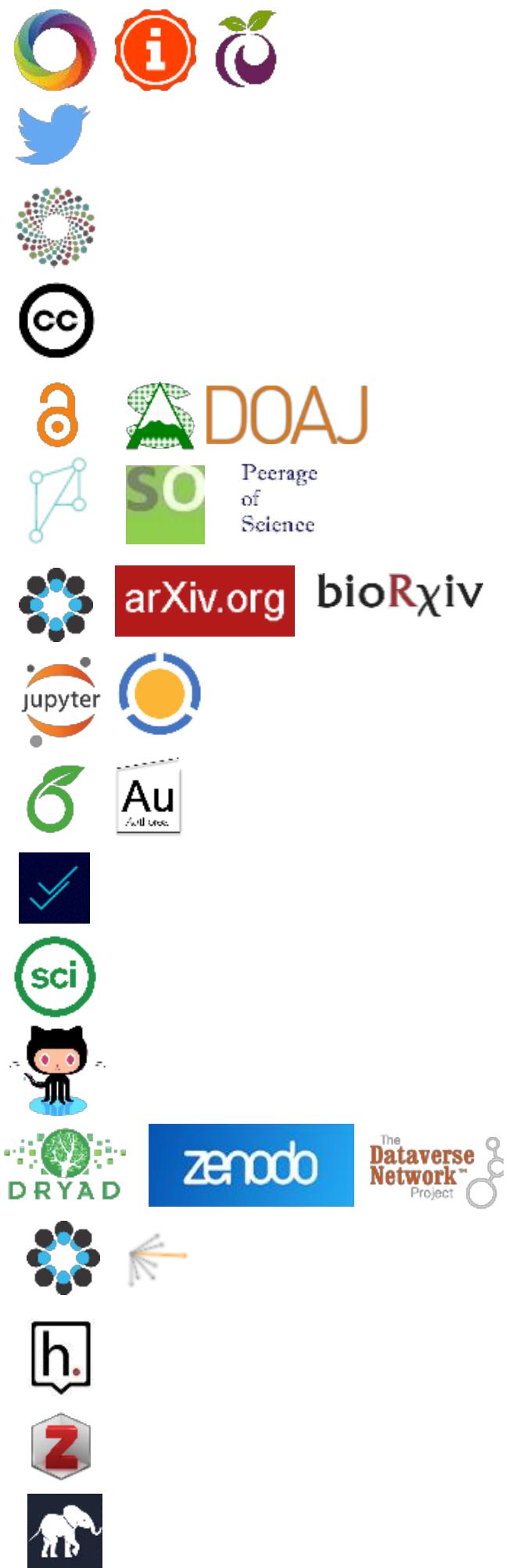
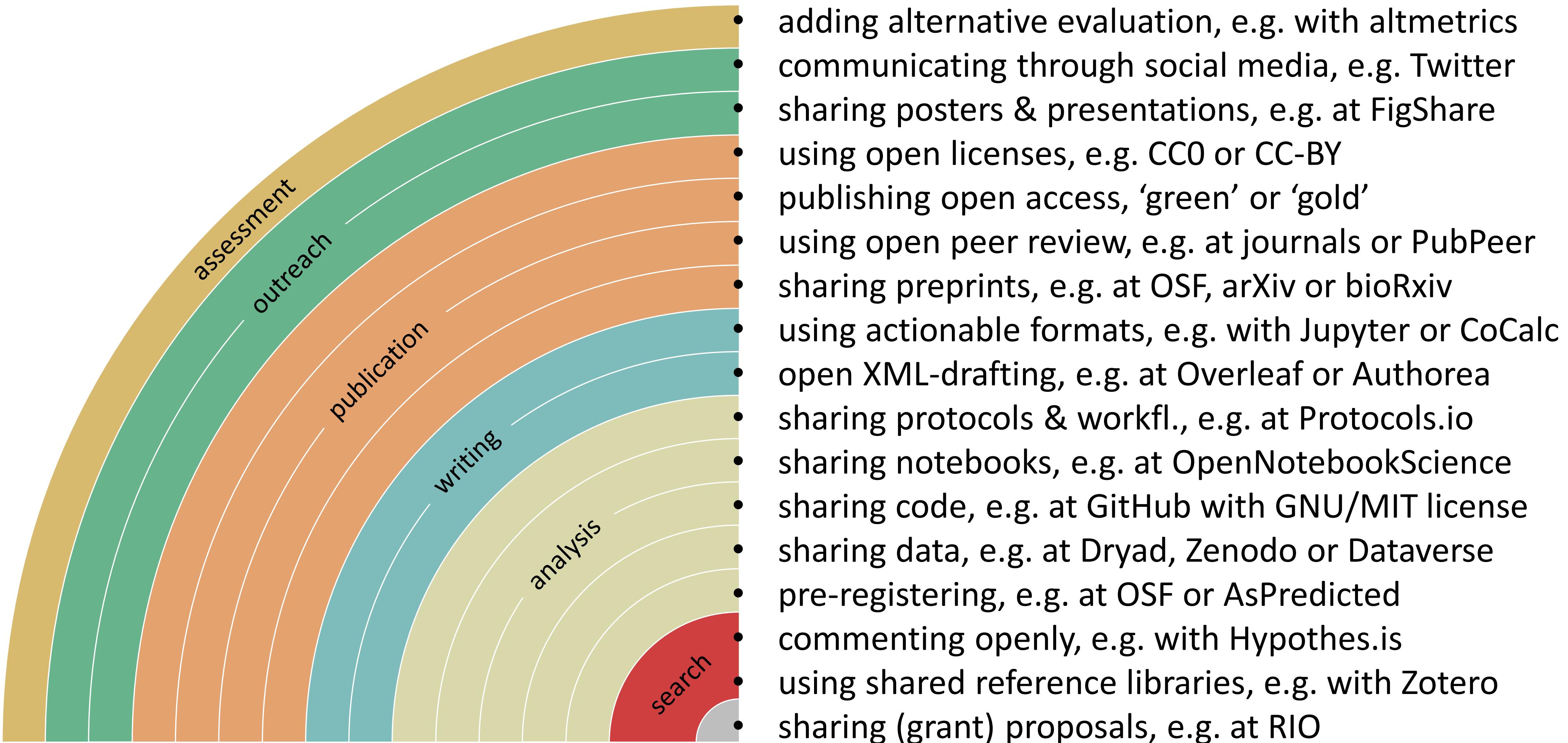


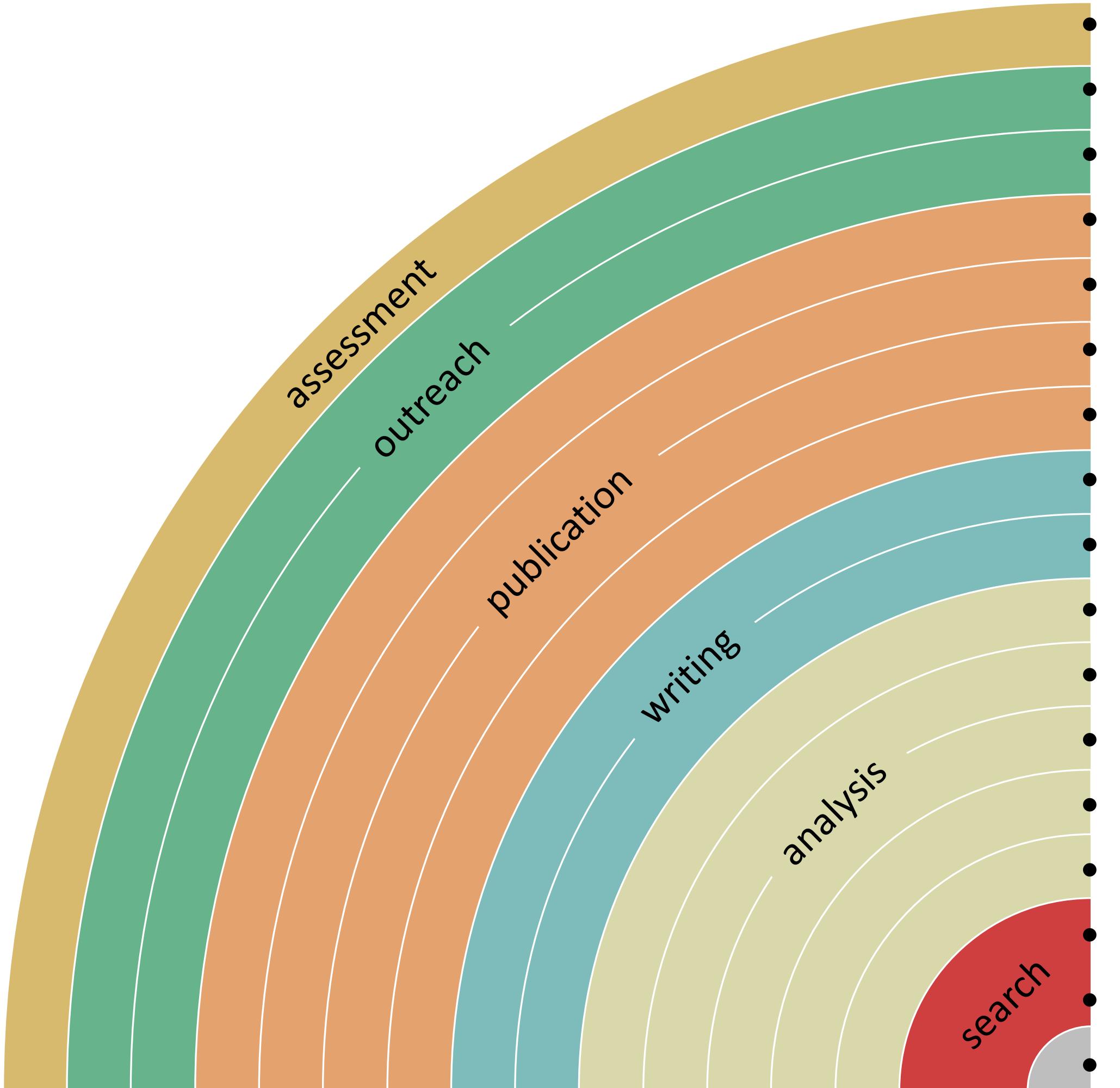
Good Enough Practices for Data and Project Management

Alaina Pearce

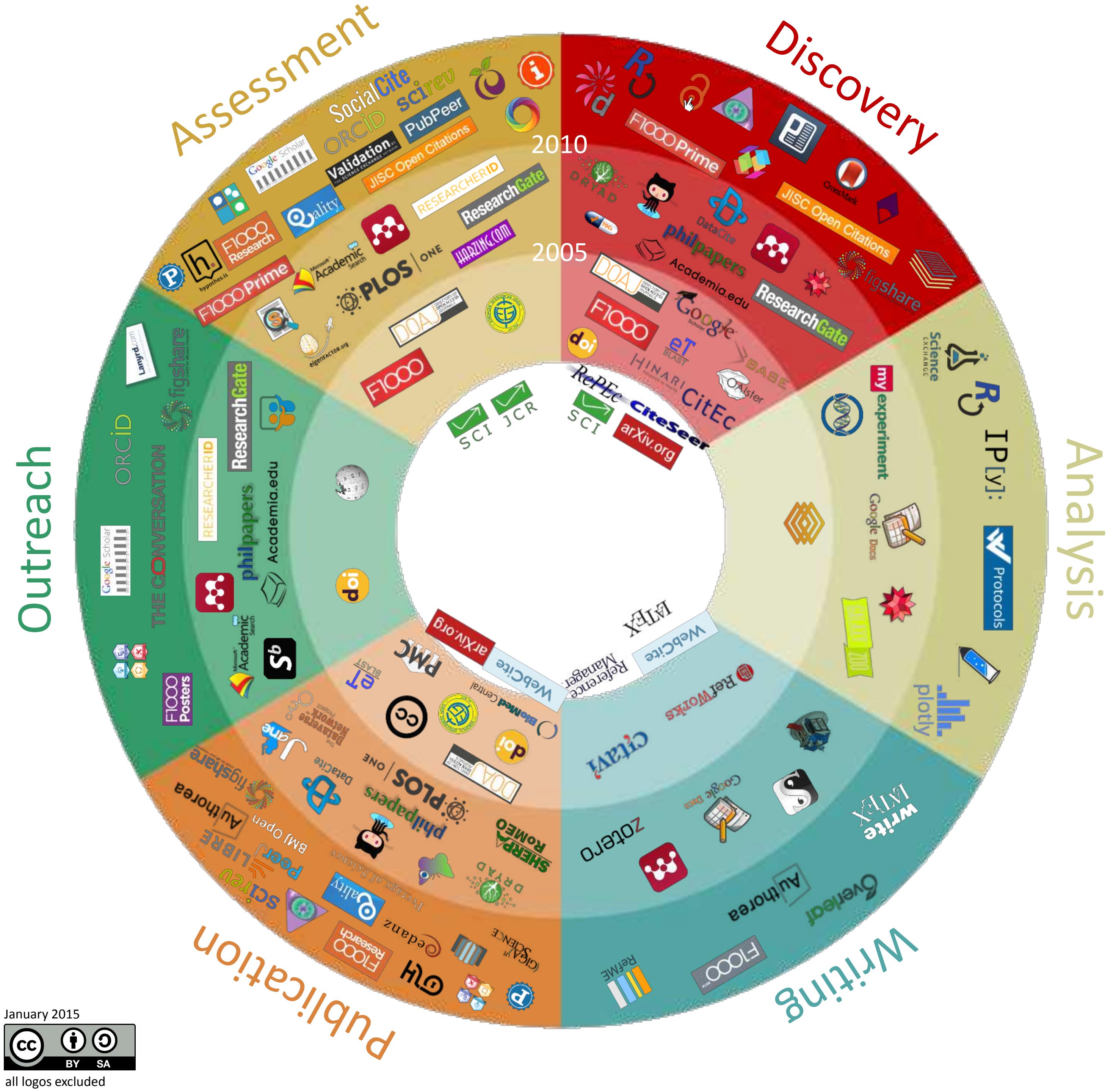
Project vs Data Management







Bianca Kramer & Jeroen Bosman <https://101innovations.wordpress.com>



January 2015



all logos excluded



Bianca Kramer

With What Time???

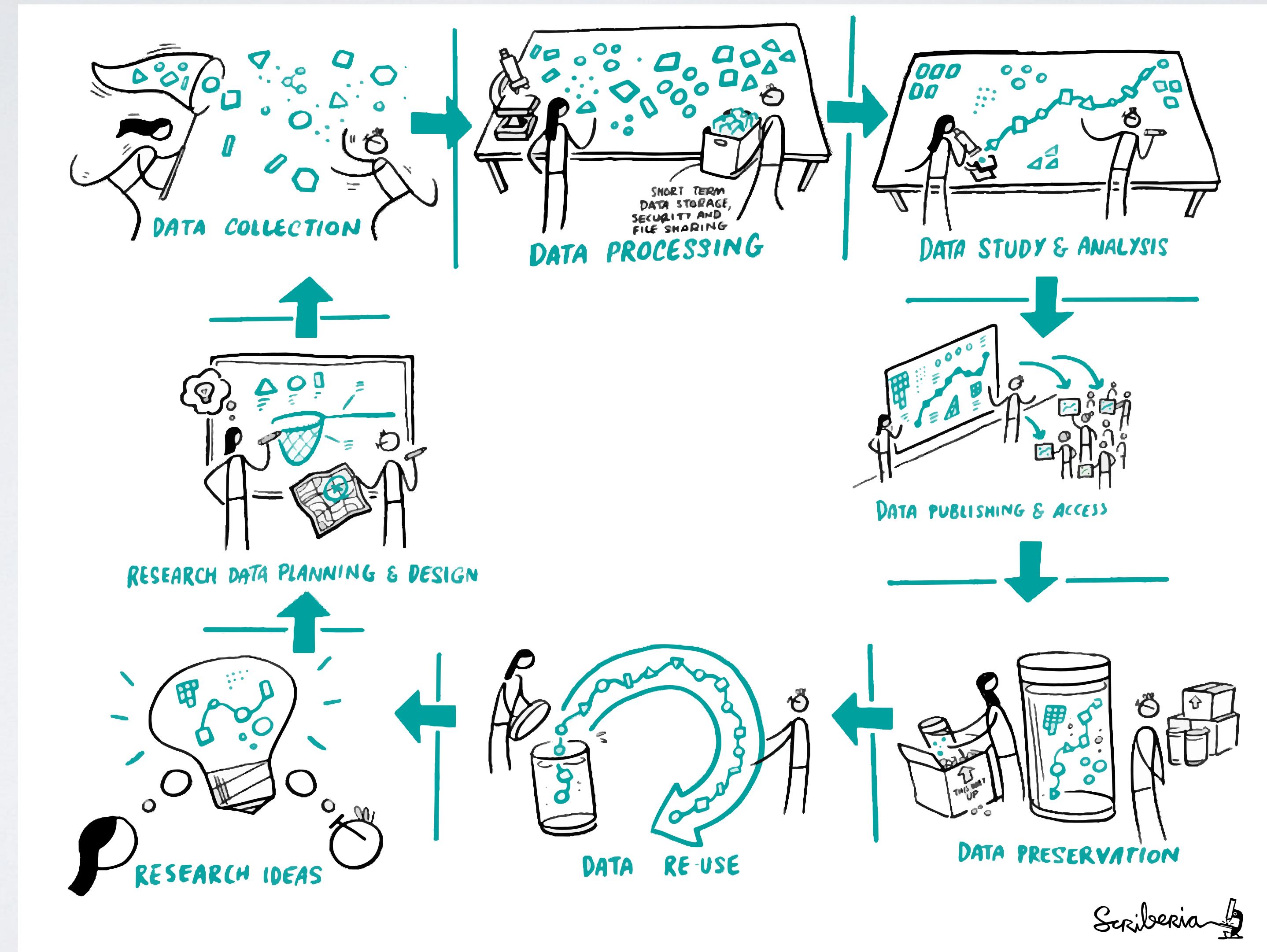


'Good Enough'

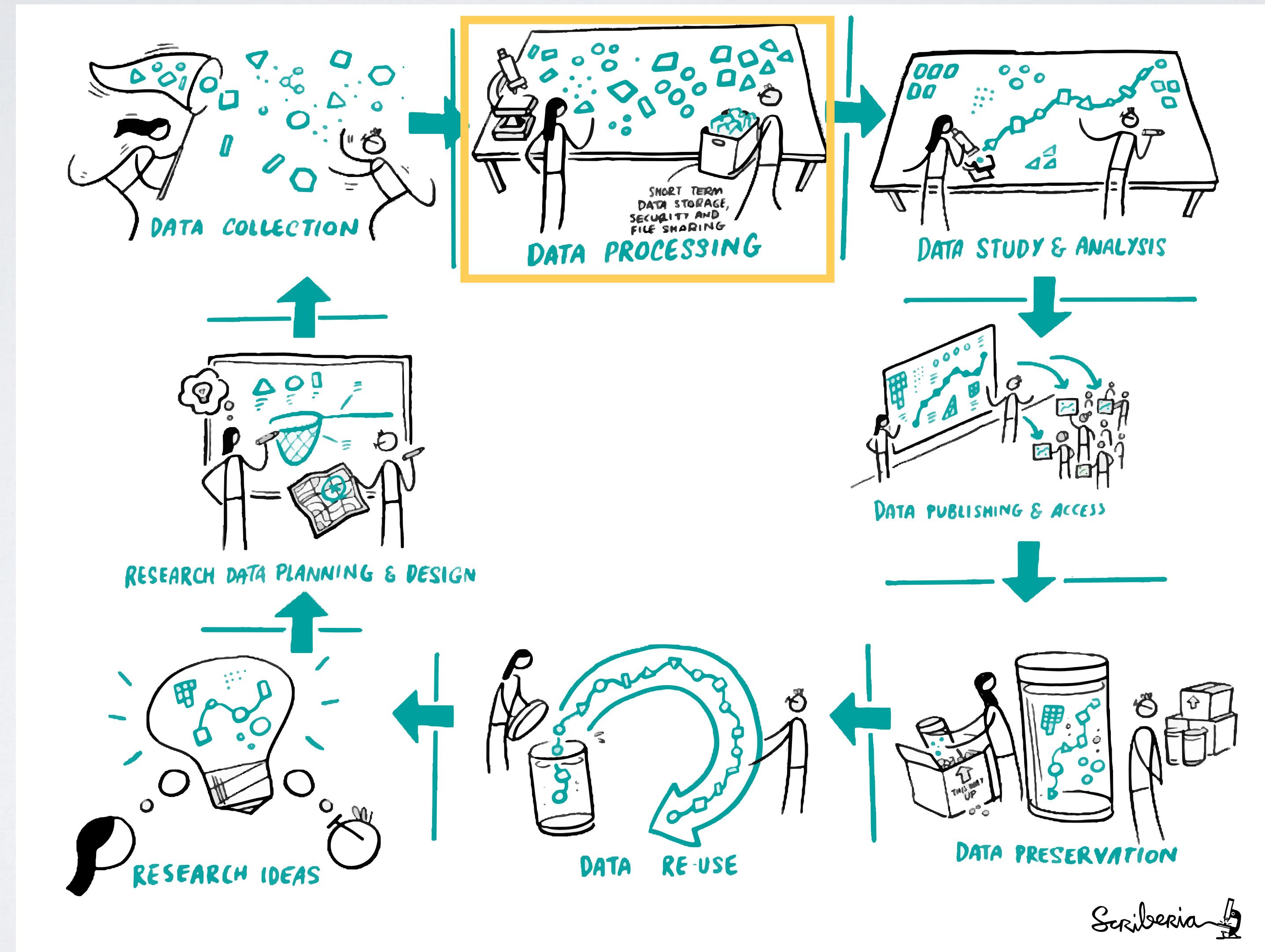
- (relatively) low effort
- shallow learning curve
- beneficial to current and future you
- increases 'openness' of research



Project Lifecycle



Project Lifecycle

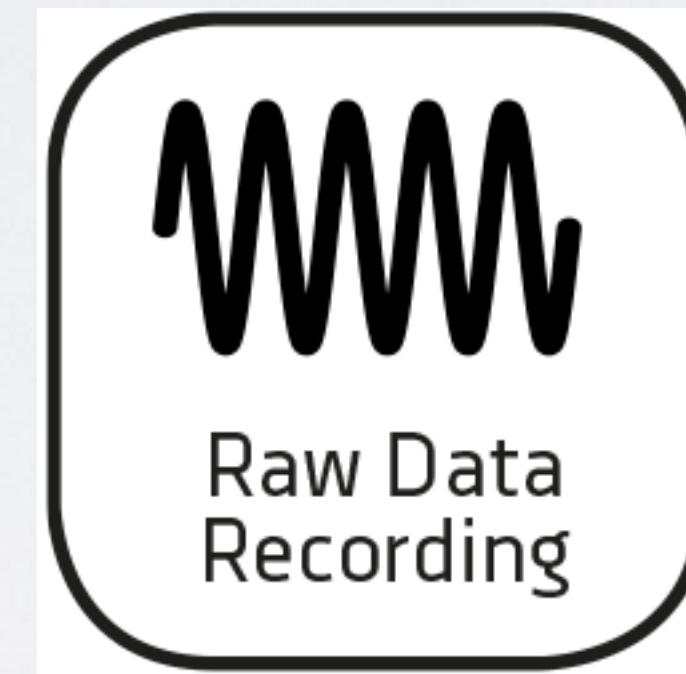
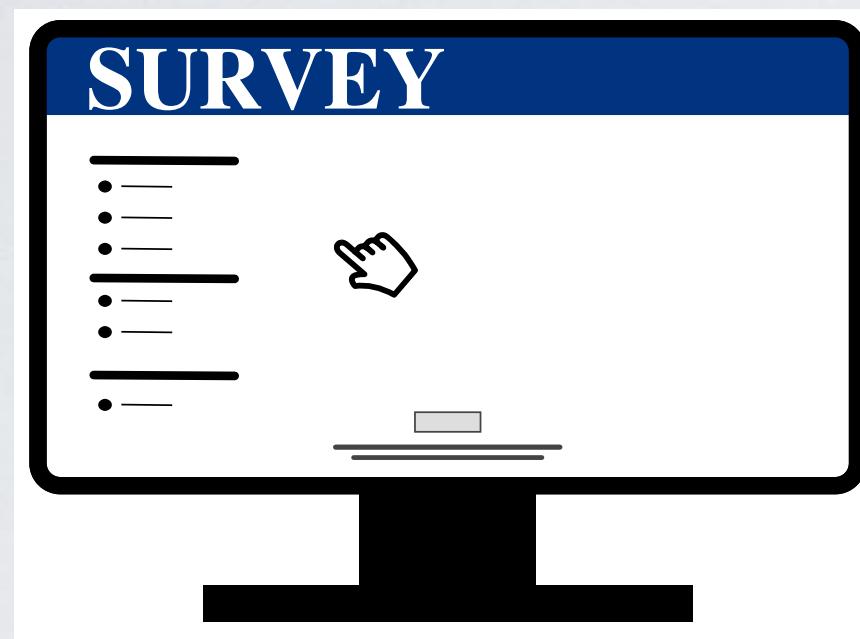


Data Processing Pipelines



1. Preserve Raw Data

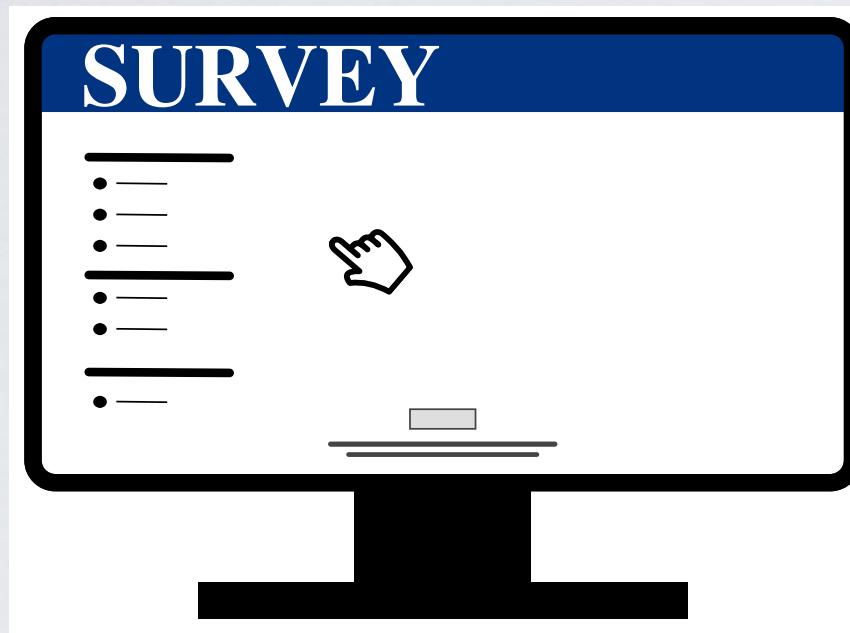
Raw Data: data as it was originally collected



Save data in its original form and DO
NOT alter or 'improve' it

1. Preserve Raw Data

Raw Data: data as it was originally collected

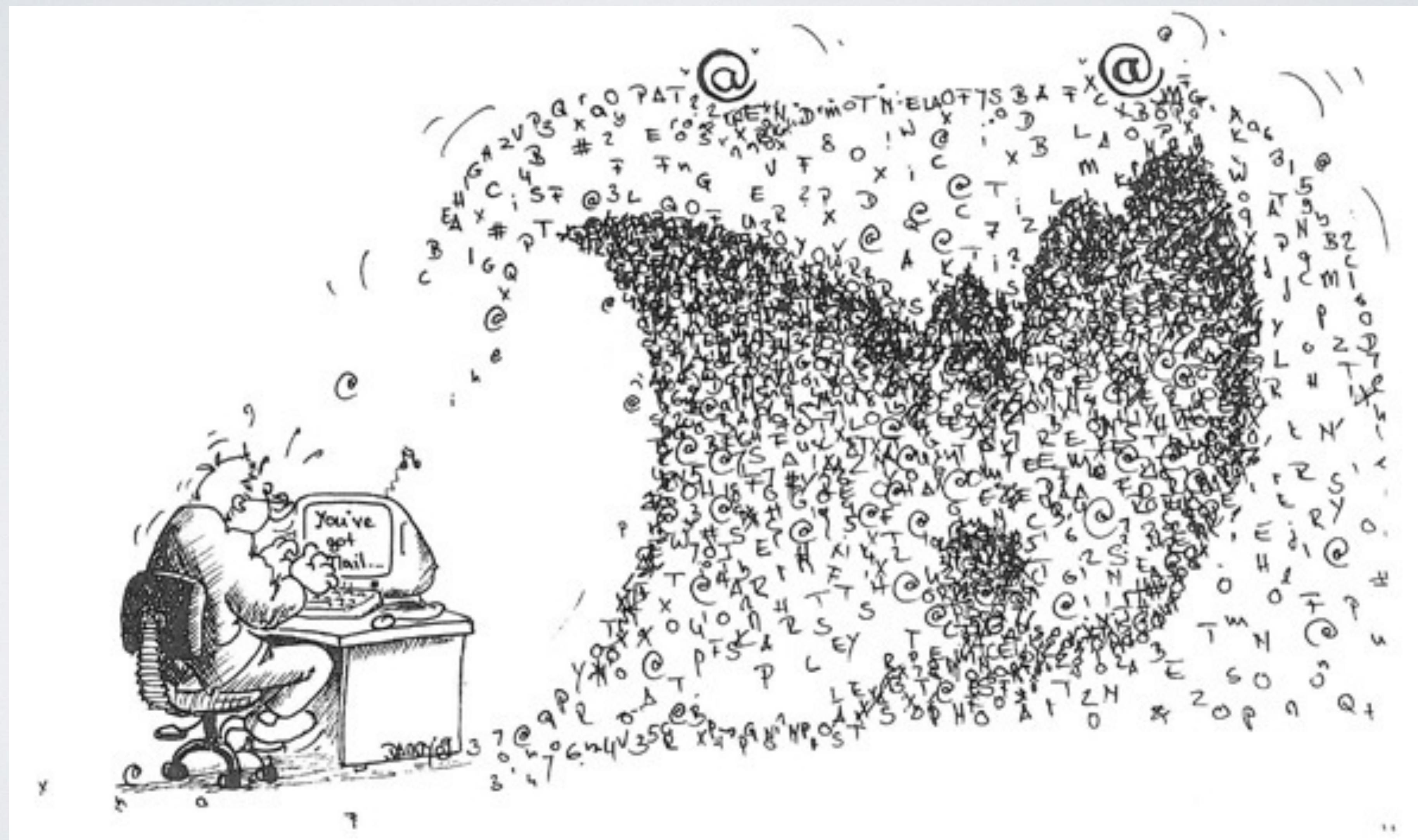


Save in data in its original form and DO
NOT alter or 'improve' it

What makes this 'Open'?

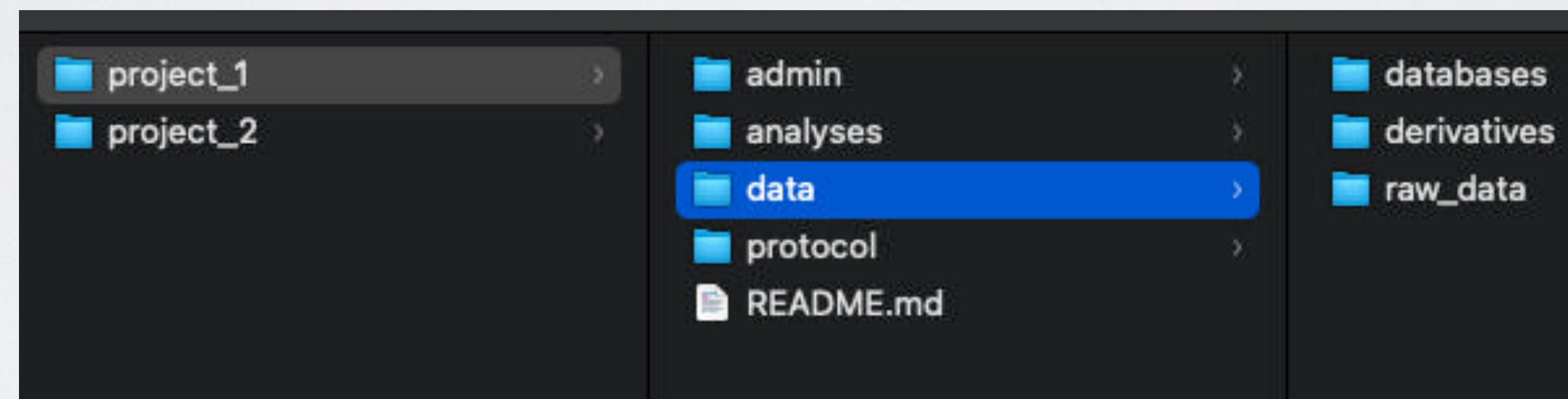
- Stable starting point
- Test reproducibility of pipeline
- Recover from mishaps
- Experiment without fear

Data Tsunami



2. Create a Central Hub

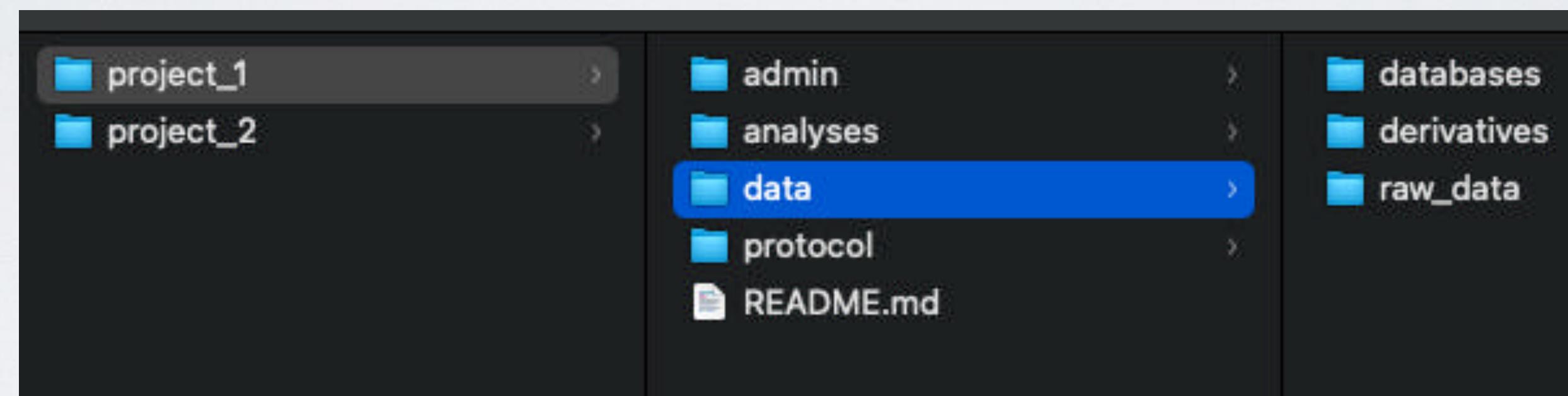
Directory Structures: organization of files into a hierarchical structure



- Create a directory for each project
- Use a consistent structure
- Separate data management from project management

2. Create a Central Hub

Directory Structures: organization of files into a hierarchical structure



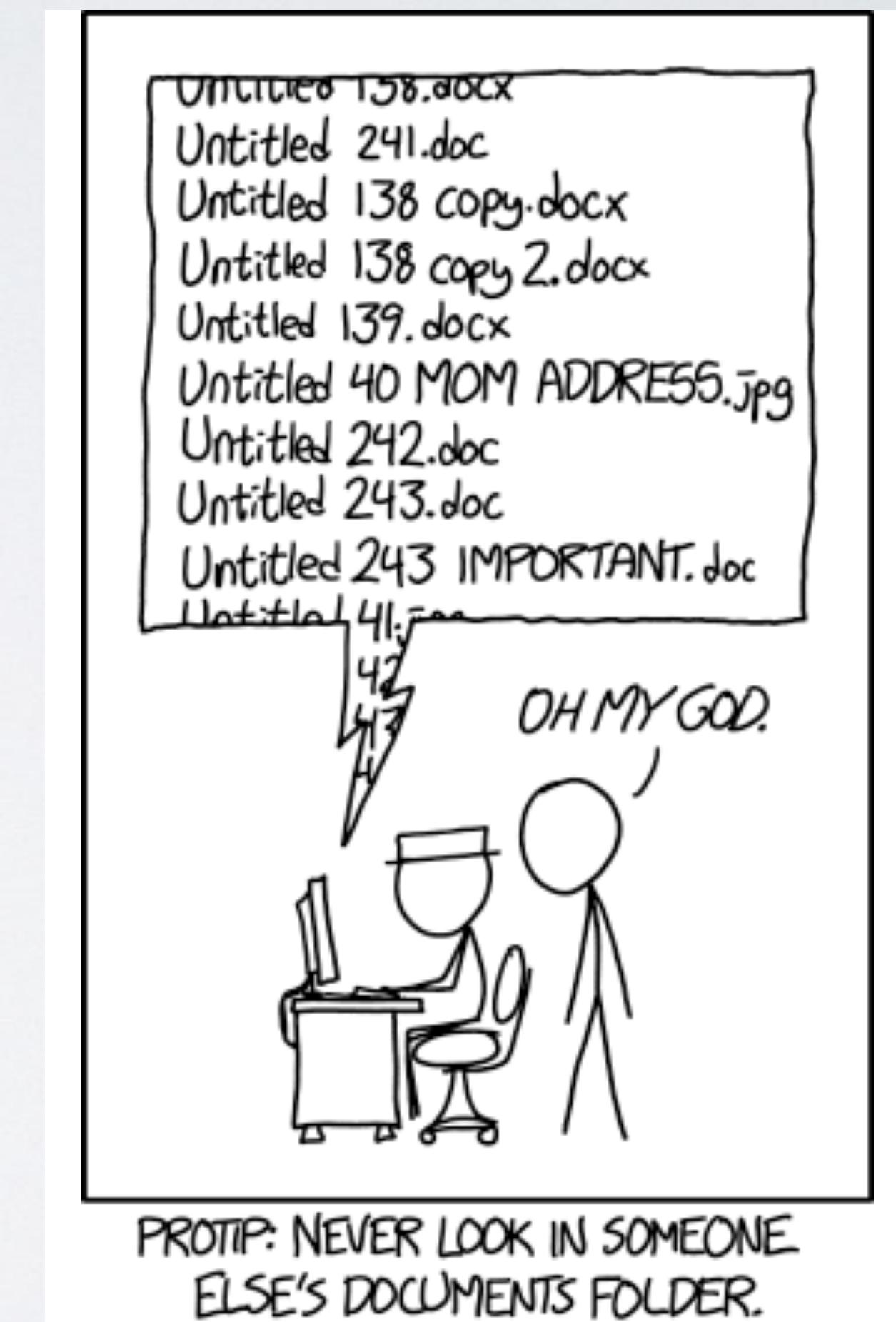
- Create a directory for each project
- Use a consistent structure
- Separate data management from project management

What makes this ‘Open’?

- Easy to find data, code, protocol
- Consistent (at least within lab)
- Bigger Lift: match field standards (e.g., BIDS, MlxS)

3. Use Meaningful Names

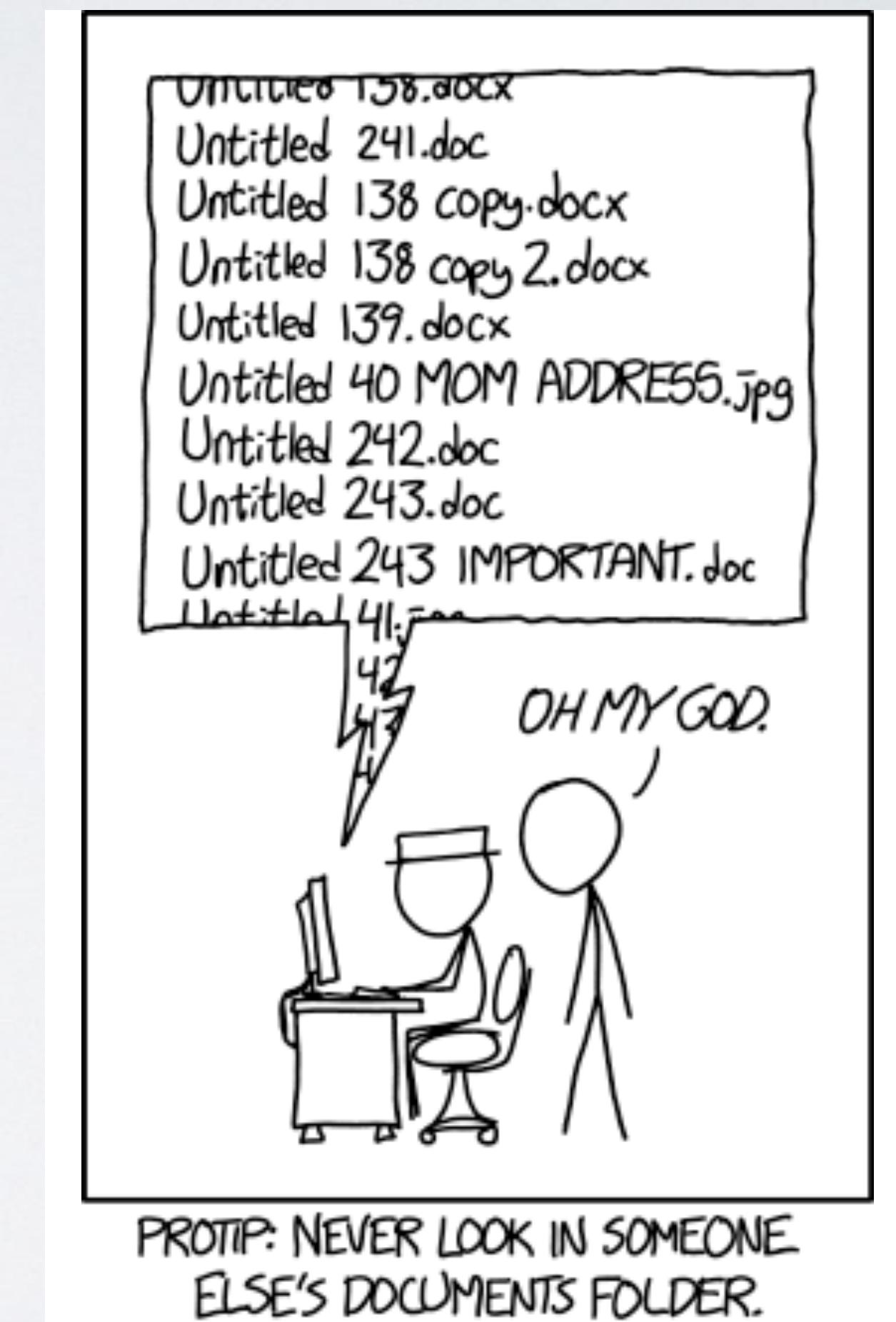
Leverage filenames to help you manage complex projects



3. Use Meaningful Names

Leverage filenames to help you manage complex project

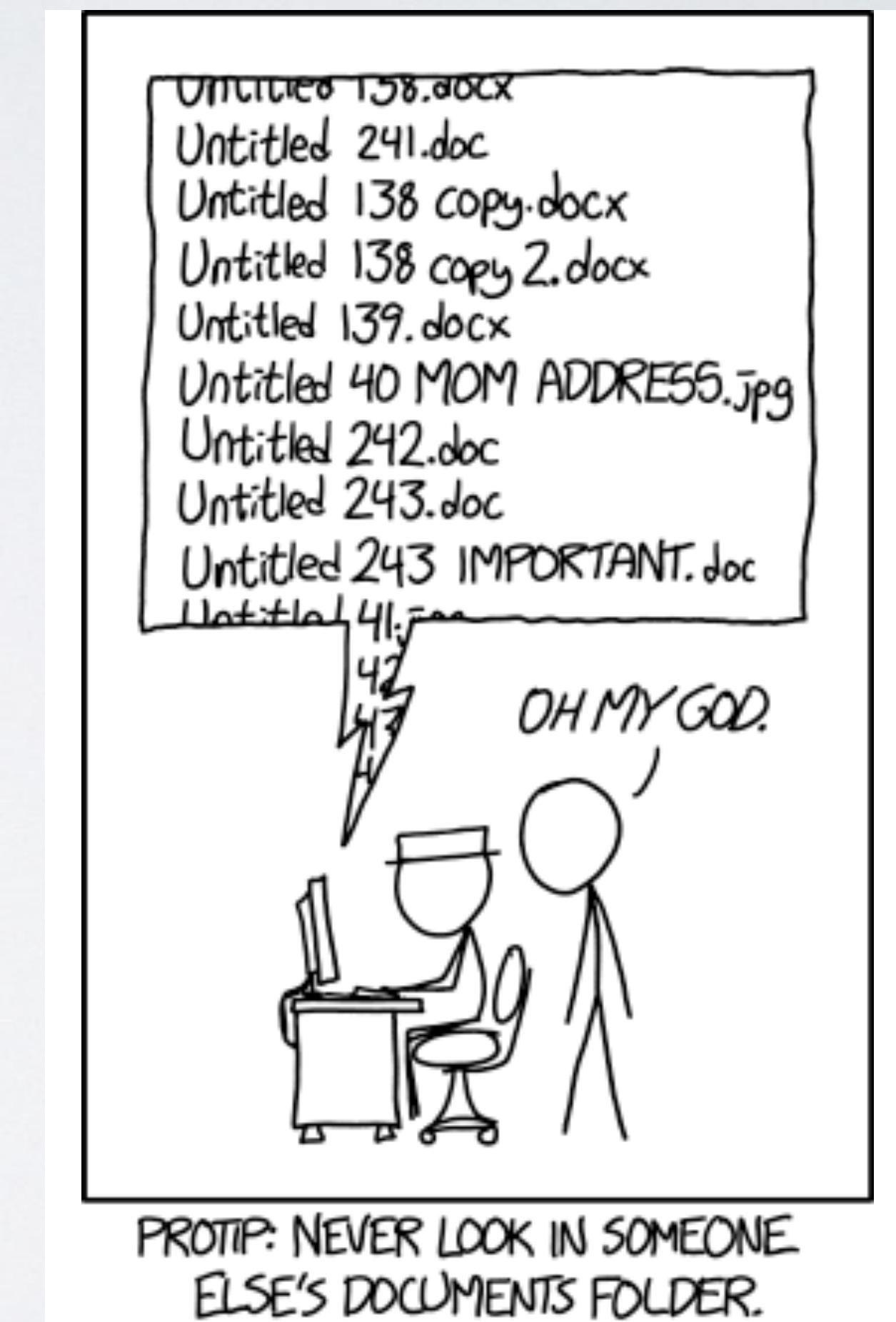
- Human Readable: names should clearly describe content in the simplest way possible (e.g., 'code', 'data')



3. Use Meaningful Names

Leverage filenames to help you manage complex project

- Human Readable: names should clearly describe content in the simplest way possible (e.g., 'code', 'data')
- Computer Readable: ability of a computer to parse a name
 - Use '-' or '_' in place of spaces
 - No special characters (e.g., '&', '#', '^', etc)

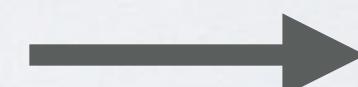


3. Use Meaningful Names

Leverage filenames to help you manage complex project

- Human Readable: names should clearly describe content in the simplest way possible (e.g., 'code', 'data')
- Computer Readable: ability of a computer to parse a name
 - Use '-' or '_' in place of spaces
 - No special characters (e.g., '&', '#', '^', etc)
- Sortable: help you find what you need in the future
 - Dates: YYYY-MM-DD
 - Pad with zeros (subject IDs, versions, etc)

```
fig_1.pdf  
fig_10.pdf  
fig_11.pdf  
fig_12.pdf  
fig_2.pdf  
fig_3.pdf  
fig_4.pdf  
fig_5.pdf  
fig_6.pdf  
fig_7.pdf  
fig_8.pdf  
fig_9.pdf
```



```
fig_01.pdf  
fig_02.pdf  
fig_03.pdf  
fig_04.pdf  
fig_05.pdf  
fig_06.pdf  
fig_07.pdf  
fig_08.pdf  
fig_09.pdf  
fig_10.pdf  
fig_11.pdf  
fig_12.pdf
```

3. Use Meaningful Names

Leverage filenames to help you manage complex project

- Human Readable: names should clearly describe content in the simplest way possible (e.g., 'code', 'data')
- Computer Readable: ability of a computer to parse a name
 - Use '-' or '_' in place of spaces
 - No special characters (e.g., '&', '#', '^', etc)
- Sortable: help you find what you need in the future
 - Dates: YYYY-MM-DD
 - Study IDs: Pad with zeros

What makes this 'Open'?

- Makes data more findable
- Can be a form of metadata
- Bigger Lift: adopt field standards

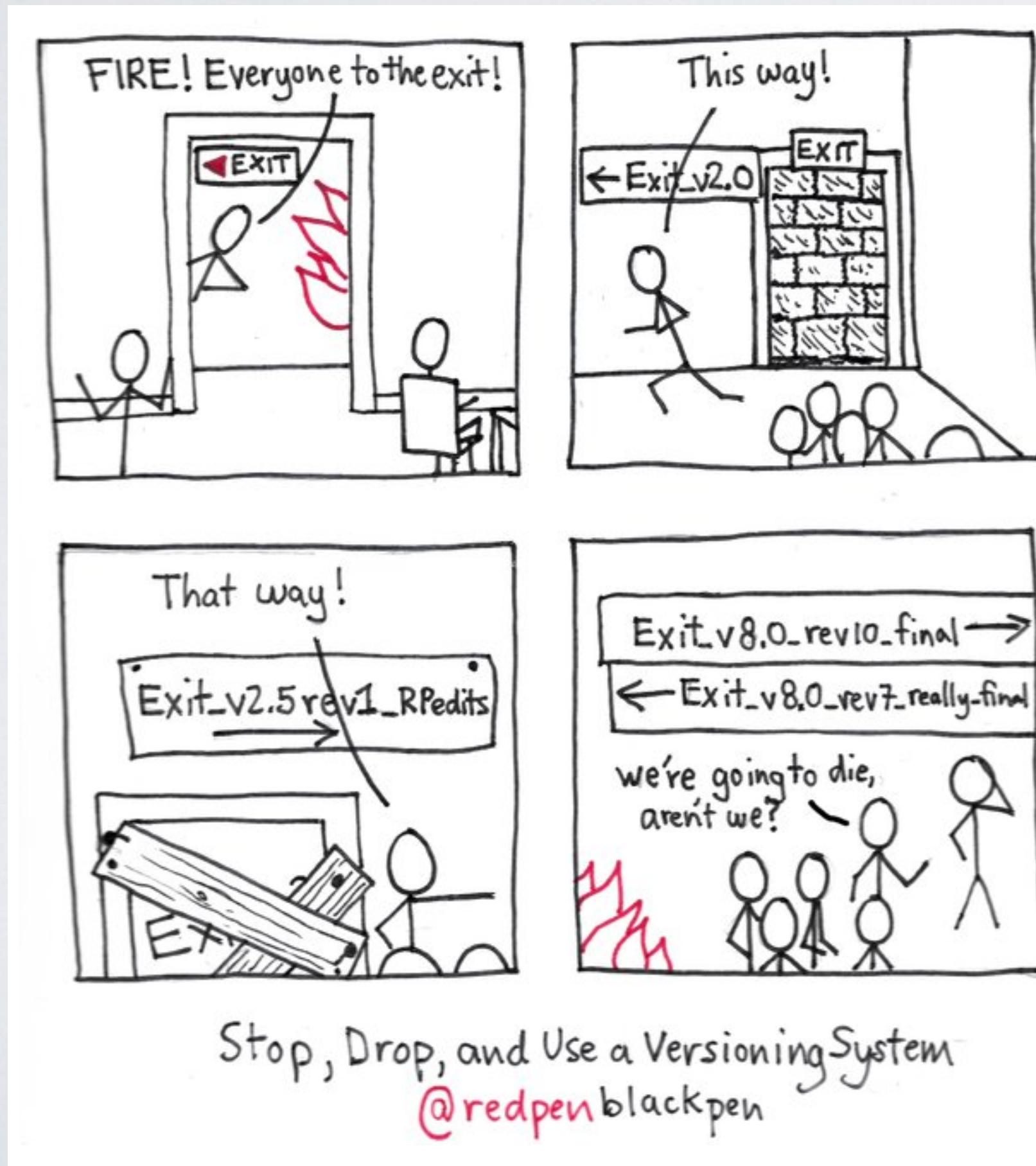
4. Preserve the Journey

Version control: tracking and managing changes to documents or code



4. Preserve the Journey

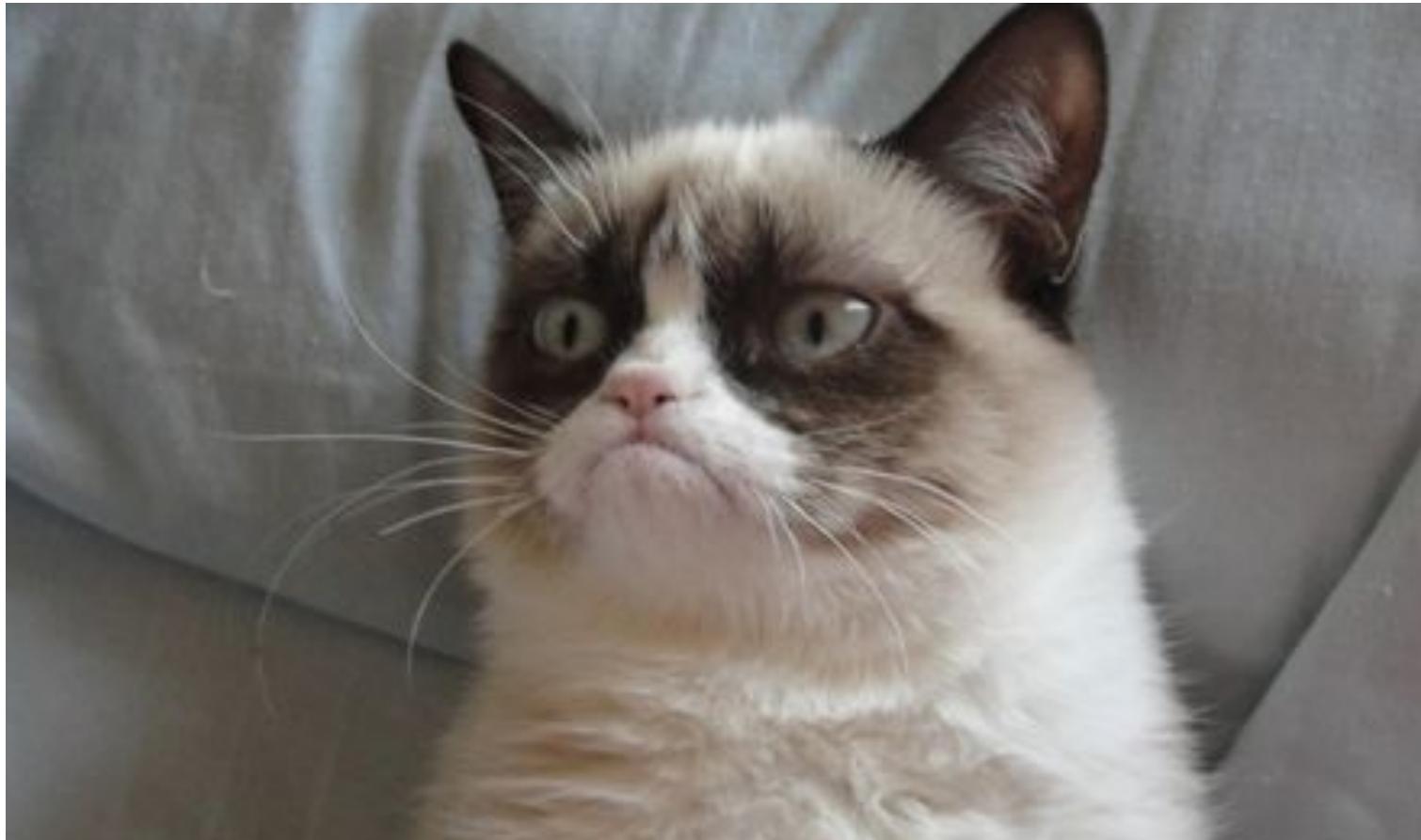
Version control: tracking and managing changes to documents or code



- Manual: use file naming to document drafts (e.g., dates, version numbers)
- Software: git, GitHub, subversion
- Allows you to trace your steps

4. Preserve the Journey

Version control: tracking and managing changes to documents or code



99 little bugs in the code
99 little bugs
Take one down and compile it
117 little bugs in the code...

- Manual: use file naming to document drafts (e.g., dates, version numbers)
- Software: git, GitHub, subversion
- Allows you to trace your steps

4. Preserve the Journey

Version control: tracking and managing changes to documents or code

What makes this ‘Open’?

- Documents project and data history
- Can reproduce process if needed
- Bigger Lift: use a version control software (e.g., git)

- Manual: use file naming to document drafts (e.g., dates, version numbers)
- Software: git, GitHub, subversion
- Allows you to trace your steps

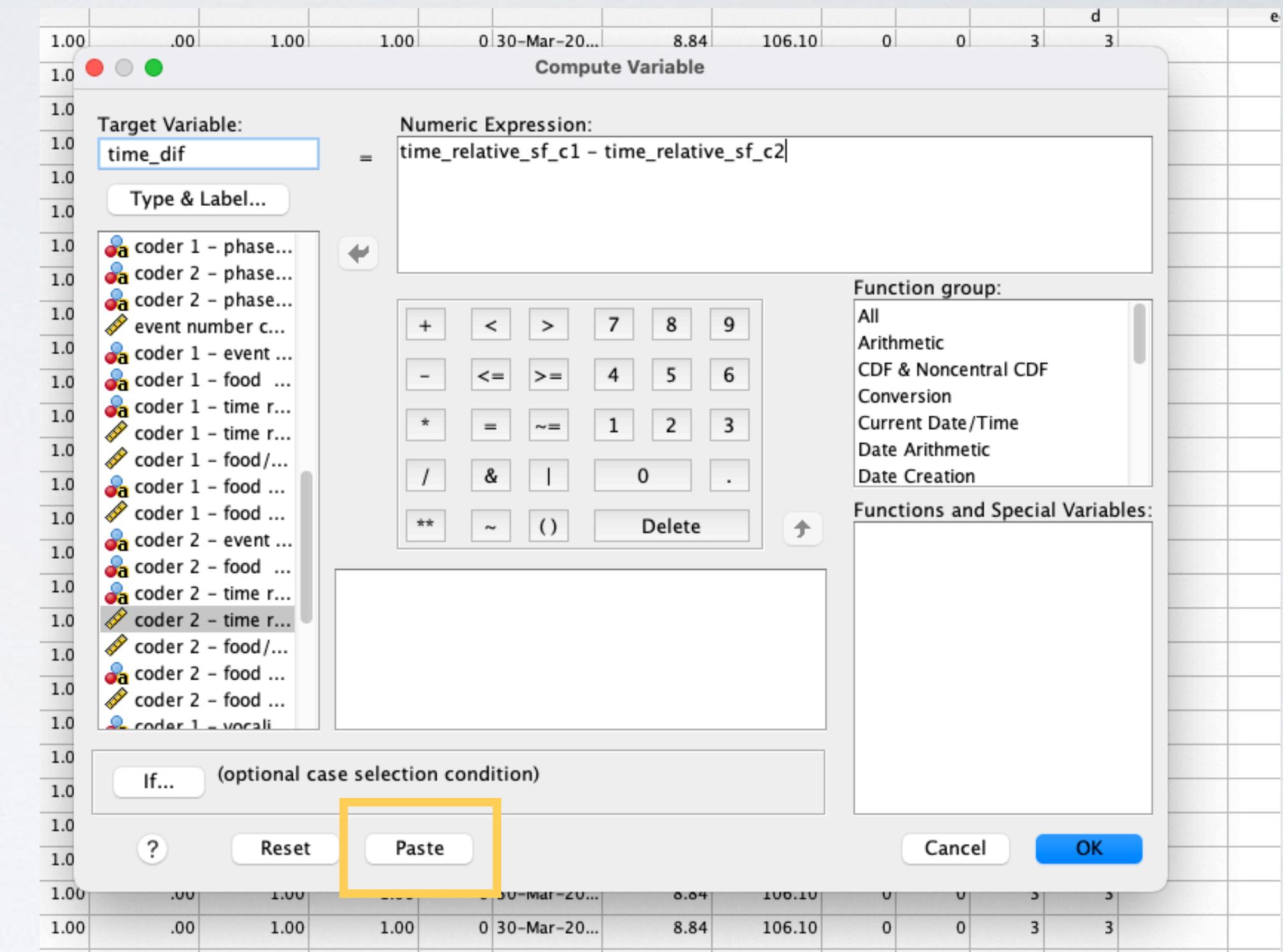
5. Avoid Manual Manipulations

- Manual data manipulations leave no trace
 - Hard to reproduce
 - Error prone
- Alternatives:
 - Save Syntax in SPSS
 - Include calculations in variable descriptions
 - Script data cleaning



5. Avoid Manual Manipulations

- Manual data manipulations leave no trace
 - Hard to reproduce
 - Error prone
- Alternatives:
 - Save Syntax in SPSS
 - Include calculations in variable descriptions
 - Script data cleaning



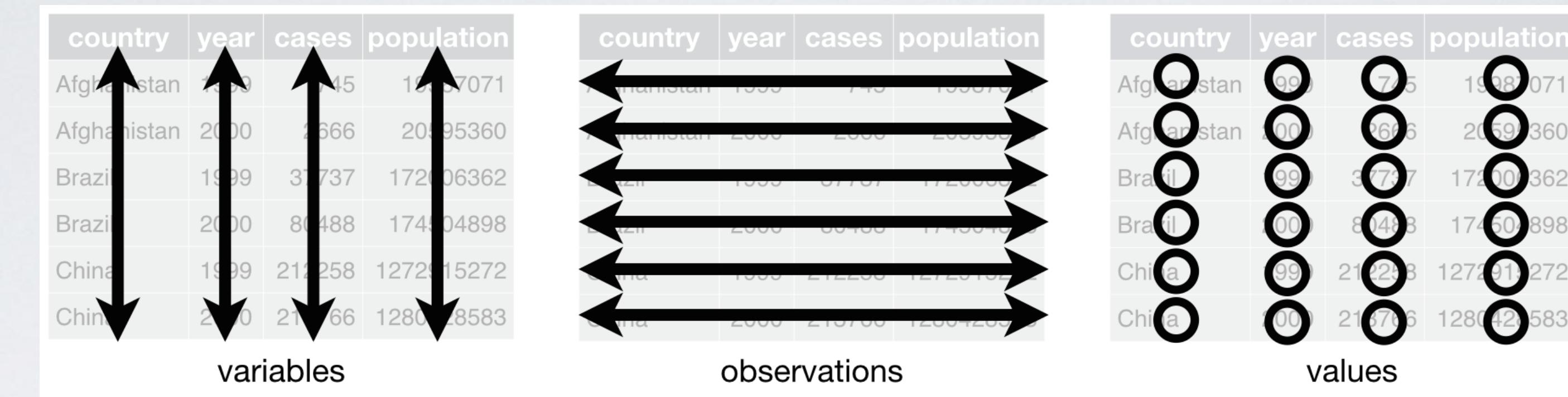
5. Avoid Manual Manipulations

- Manual data manipulations leave no trace
 - Hard to reproduce
 - Error prone
- Alternatives:
 - Save Syntax in SPSS
 - Include calculations in variable descriptions
 - Script data cleaning

What makes this ‘Open’?

- Data processing will be reproducible
- Can reverse to original data if needed
- Bigger Lift: move away from GUI-based analysis software to open code/syntax based programs (e.g., R, python)

6. 'Tidy' Your Data



- Every variable is in its own column
- Each participant/sample is in its own row
- Each value is in its own cell

6. 'Tidy' Your Data

- Use open file formats — csv, html, txt, jpeg

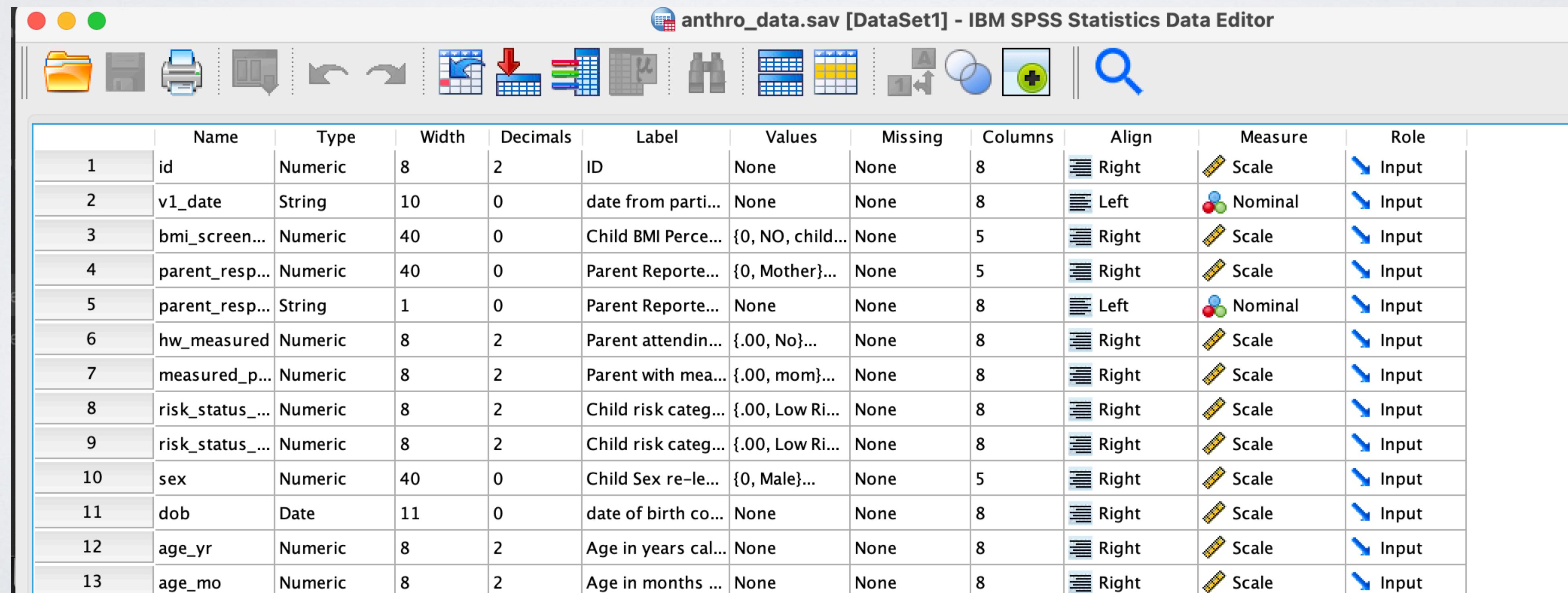
6. 'Tidy' Your Data

- Use open file formats — csv, html, txt, jpeg
- Create a data dictionary

column	variable	label	value_labels	type	n_na	range
1	id	ID	NULL	double	0	c(1, 133)
2	v1_date	date from participant contacts databases ('verified_visit_da	NULL	character	0	c("2018-01-31", "2022-05-07")
3	bmi_screenout	Child BMI Percentile Screen Out	c('YES, child is overweight, sc	double	0	c(0, 1)
4	parent_respondent	Parent Reported: Parent relationship to child re-leveled in R	c(Mother = 0, Father = 1, Oth	double	0	c(0, 1)
5	parent_respondent_o	Parent Reported: Parent specify relationship to child if other	NULL	character	0	c("", "")
6	hw_measured	Parent attending Visit 1 had measured height and weight	c(No = 0, Yes = 1)	double	0	c(1, 1)
7	measured_parent	Parent with measured BMI at Visit 1	c(mom = 0, dad = 1)	double	0	c(0, 1)
8	risk_status_mom	Child risk categor: Low risk: Mom BMI < 26, High Risk: Mom	c('Low Risk' = 0, 'High Risk' =	double	0	c(0, 1)
9	risk_status_both	Child risk category: Low Risk: Mom and Dad BMI < 25, High	c('Low Risk' = 0, 'High Risk' =	double	0	c(0, 2)
10	sex	Child Sex re-leveled in R to start with 0	c(Male = 0, Female = 1)	double	0	c(0, 1)
11	dob	date of birth converted to format yyyy-mm-dd in R	NULL	double	0	c(14333, 16391)
12	age_yr	Age in years calculated from dob and start_date	NULL	double	0	c(7, 8.99)
13	age_mo	Age in months calculated from dob and start_date	NULL	double	0	c(84, 107.9)
14	ethnicity	Parent Reported: Child ethnicity	c('NOT Hispanic or Latino' = 0	double	0	c(0, 0)
15	race	Parent Reported: Child race -- Note: prefer not to answer (p	c('White/Caucasian' = 0, 'Am	double	0	c(0, 2)
16	income	Parent Reported: Yearly household income -- Note: prefer n	c('Less than \$20,000' = 0, '\$20	double	3	c(0, 5)
17	parent_ed	Parent Reported: Parent education re-leveled in R to start w	c('High School or GED (12 yea	double	0	c(0, 5)

6. 'Tidy' Your Data

- Use open file formats — csv, html, txt, jpeg
- Create a data dictionary

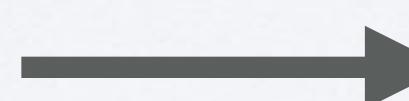


The screenshot shows the IBM SPSS Statistics Data Editor interface. The title bar reads "anthro_data.sav [DataSet1] - IBM SPSS Statistics Data Editor". The toolbar contains various icons for file operations like Open, Save, Print, and Data manipulation. The main window displays a data dictionary (variable view) with 13 rows of variables and their properties.

	Name	Type	Width	Decimals	Label	Values	Missing	Columns	Align	Measure	Role
1	id	Numeric	8	2	ID	None	None	8	Right	Scale	Input
2	v1_date	String	10	0	date from parti...	None	None	8	Left	Nominal	Input
3	bmi_screen...	Numeric	40	0	Child BMI Perce...	{0, NO, child...	None	5	Right	Scale	Input
4	parent_resp...	Numeric	40	0	Parent Reporte...	{0, Mother}...	None	5	Right	Scale	Input
5	parent_resp...	String	1	0	Parent Reporte...	None	None	8	Left	Nominal	Input
6	hw_measured	Numeric	8	2	Parent attendin...	{.00, No}...	None	8	Right	Scale	Input
7	measured_p...	Numeric	8	2	Parent with mea...	{.00, mom}...	None	8	Right	Scale	Input
8	risk_status_...	Numeric	8	2	Child risk categ...	{.00, Low Ri...	None	8	Right	Scale	Input
9	risk_status_...	Numeric	8	2	Child risk categ...	{.00, Low Ri...	None	8	Right	Scale	Input
10	sex	Numeric	40	0	Child Sex re-le...	{0, Male}...	None	5	Right	Scale	Input
11	dob	Date	11	0	date of birth co...	None	None	8	Right	Scale	Input
12	age_yr	Numeric	8	2	Age in years cal...	None	None	8	Right	Scale	Input
13	age_mo	Numeric	8	2	Age in months ...	None	None	8	Right	Scale	Input

6. 'Tidy' Your Data

- Use open file formats — csv, html, txt, jpeg
- Create a data dictionary
- One piece of information per cell



height	height_ft	height_in
5 ft 6 in	5	6
5 ft 2 in	5	2
7 ft	7	0
5 ft 11 in	5	11

6. 'Tidy' Your Data

- Use open file formats — csv, html, txt, jpeg
- Create a data dictionary
- One piece of information per cell
- Do not use highlighting/font color as data

height
5 ft 6 in
5 ft 2 in
7 ft
5 ft 11 in

→

height_ft	height_in	check_height
5	6	0
5	2	0
7	0	1
5	11	0

6. 'Tidy' Your Data

- Use open file formats — csv, html, txt, jpeg
- Create a data dictionary
- One piece of information per cell
- Do not use highlighting/font color as data

What makes this 'Open'?

- Open formats are accessible
- All data are computer readable
- Data are documented
- Makes data re-use and sharing easier

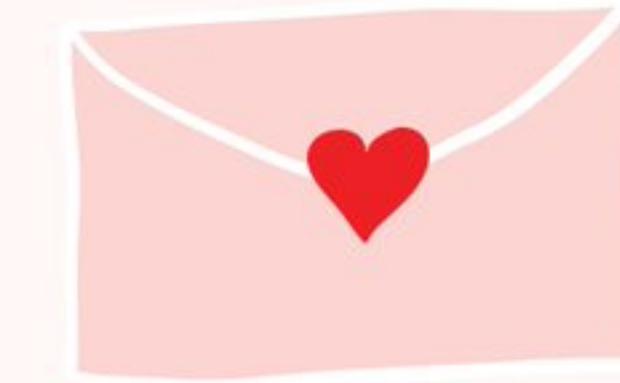
7. Metadata Magic

Metadata: the who, what, when, where, and why of your data

Easiest: when in doubt, document

- Data dictionaries
- Standard operating procedures manuals
- Lab notebooks
- changelog file (document versions)
- README
 - Description of folders/files
 - Can provide instructions on use of code/
data
 - License information

METADATA IS A
LOVE NOTE TO
THE FUTURE!



7. Metadata Magic

Metadata: the who, what, when, where, and why of your data

Medium Effort: Data Manual

- Larger
- More verbose and detailed
- Can include science/rational/citations
- Like a user manual for data

METADATA IS A
LOVE NOTE TO
THE FUTURE!



7. Metadata Magic

Metadata: the who, what, when, where, and why of your data

Bigger Lift: Structured Metadata

- Often laid out in fields
- Can require use of shared vocabularies
- Often field/data type specific

METADATA IS A
LOVE NOTE TO
THE FUTURE!



7. Metadata Magic

Metadata: the who, what, when, where, and why of your data

Bigger Lift: Structured Metadata

- Often laid out in fields
- Can require use of shared vocabularies
- Often field/data type specific

What makes this ‘Open’?

- Makes data more findable
- Helps others (and future you) understand the data
- Shared vocabularies help to harmonize data within a field

'Good Enough' Practices

1. Preserve Raw Data
2. Create a Central Hub
3. Use Meaningful Names
4. Preserve the Journey
5. Avoid Manual Manipulations
6. 'Tidy' Your Data
7. Metadata Magic

