

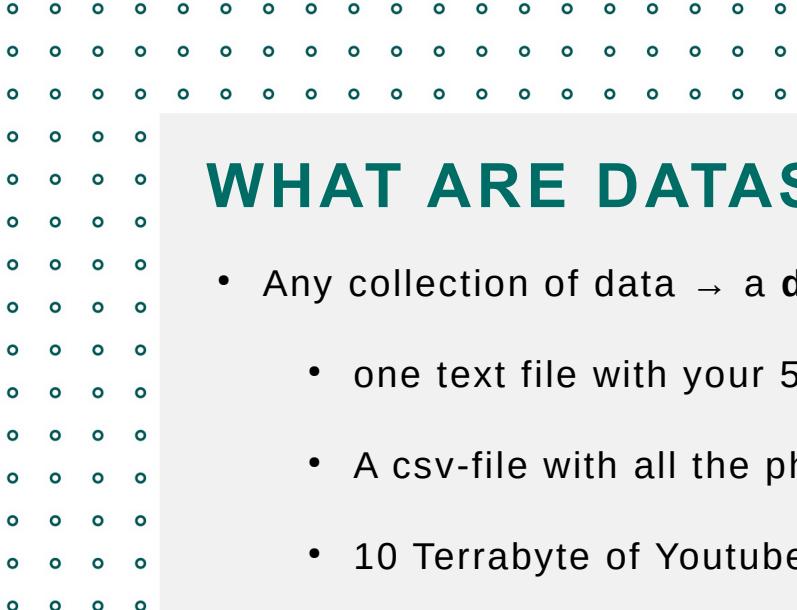


EDSD 2022  
ACCESSING AND ANALYSING WEB AND SOCIAL MEDIA  
DATA

FINDING AND WORKING WITH DATASETS

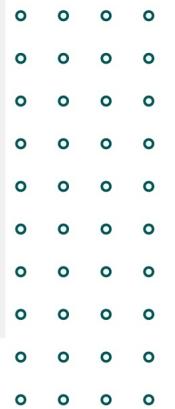
Tom Theile

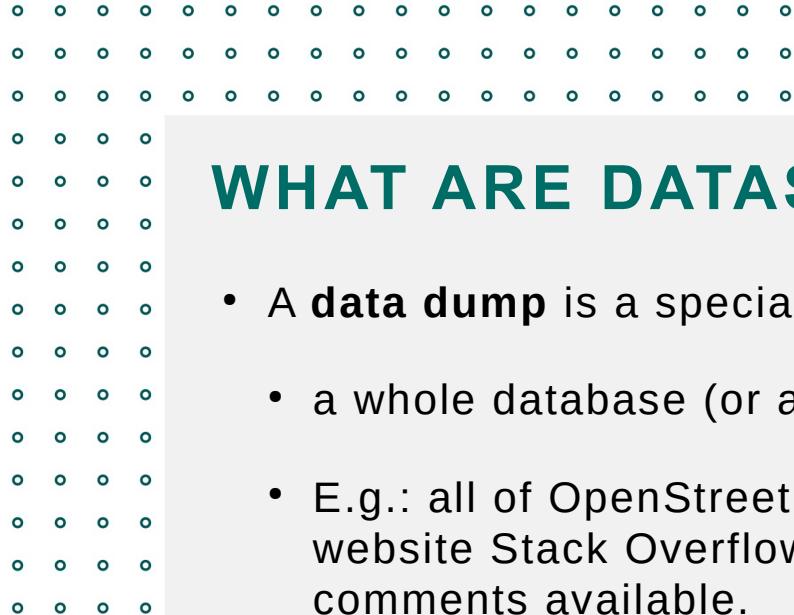
Lab of digital and computational  
demography



## WHAT ARE DATASETS AND DATADUMPS?

- Any collection of data → a **dataset!** Examples:
  - one text file with your 5 favorite prime numbers...
  - A csv-file with all the phone numbers and names of every person and organization in Spain
  - 10 Terrabyte of Youtube videos with their subtitles, collected into 3000 files
  - ... anything
- Often: The data is **structured**
- Sometimes: The data is **labeled** (Humans put a label on every Datum)
  - Important for machine learning, AI





## WHAT ARE DATASETS AND DATADUMPS?

- A **data dump** is a special kind of dataset:
  - a whole database (or a significant part of it) of a specific website.
  - E.g.: all of OpenStreetMap; all of Wikidata; all of the english Wikipedia; all of the website Stack Overflow. There is even a collection of all reddit posts and all reddit comments available.
- Why?
  - Service to researchers
  - Prevent people from having to scrape a whole site



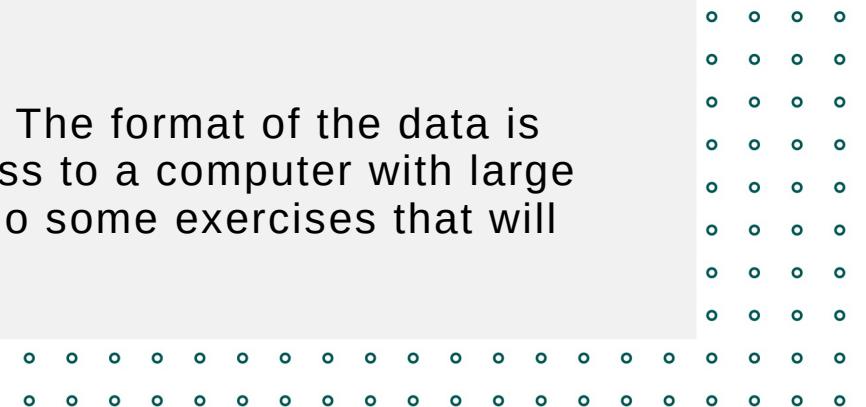


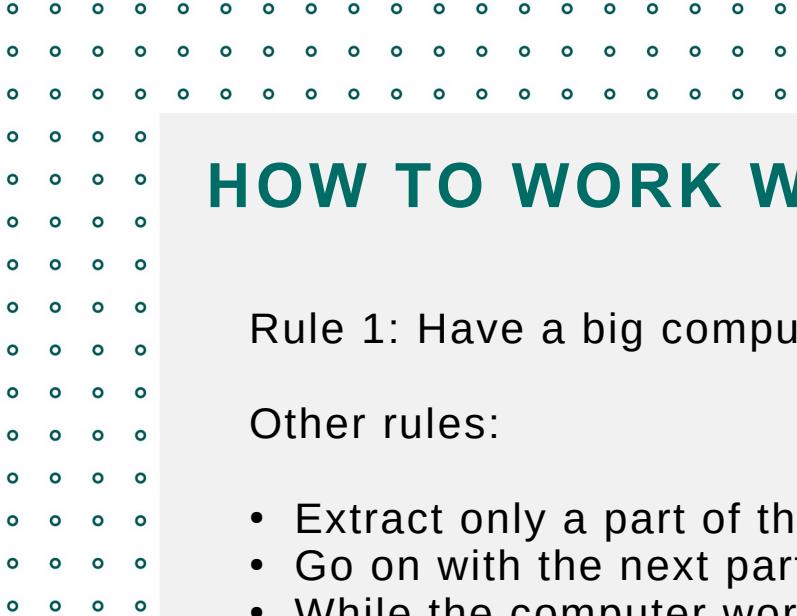
## WHAT ARE DATASETS AND DATADUMPS?

“As of 20 February 2022, there are 6,456,456 articles in the English Wikipedia”, scraping them all with Rvest would take 75 days, if you access 1 page per second (which might get you blocked) ( $1 \text{ page/s} * 60*60*24*365 = 31.5 \text{ Million Pages per year}$ ).

Instead of scraping the whole site, you can download a compressed data dump of the English Wikipedia which is 20.47 GB in size and downloads in a few minutes.

Working with large datasets can sometimes be a challenge. The format of the data is often different and special. Sometimes you don't have access to a computer with large enough RAM to fit the whole dataset into memory. We will do some exercises that will help you find out how to work with large datasets.





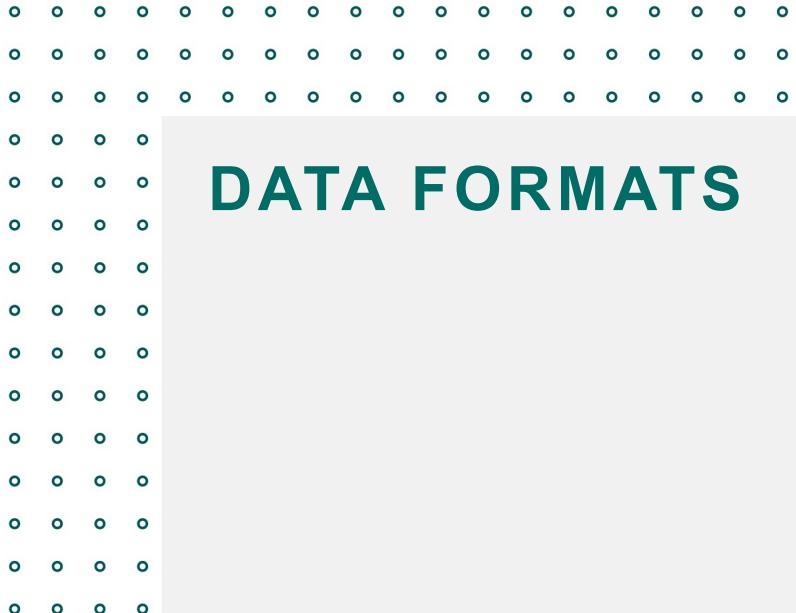
## HOW TO WORK WITH BIG DATA?

Rule 1: Have a big computer to work with big data! Lots of RAM will be helpful.

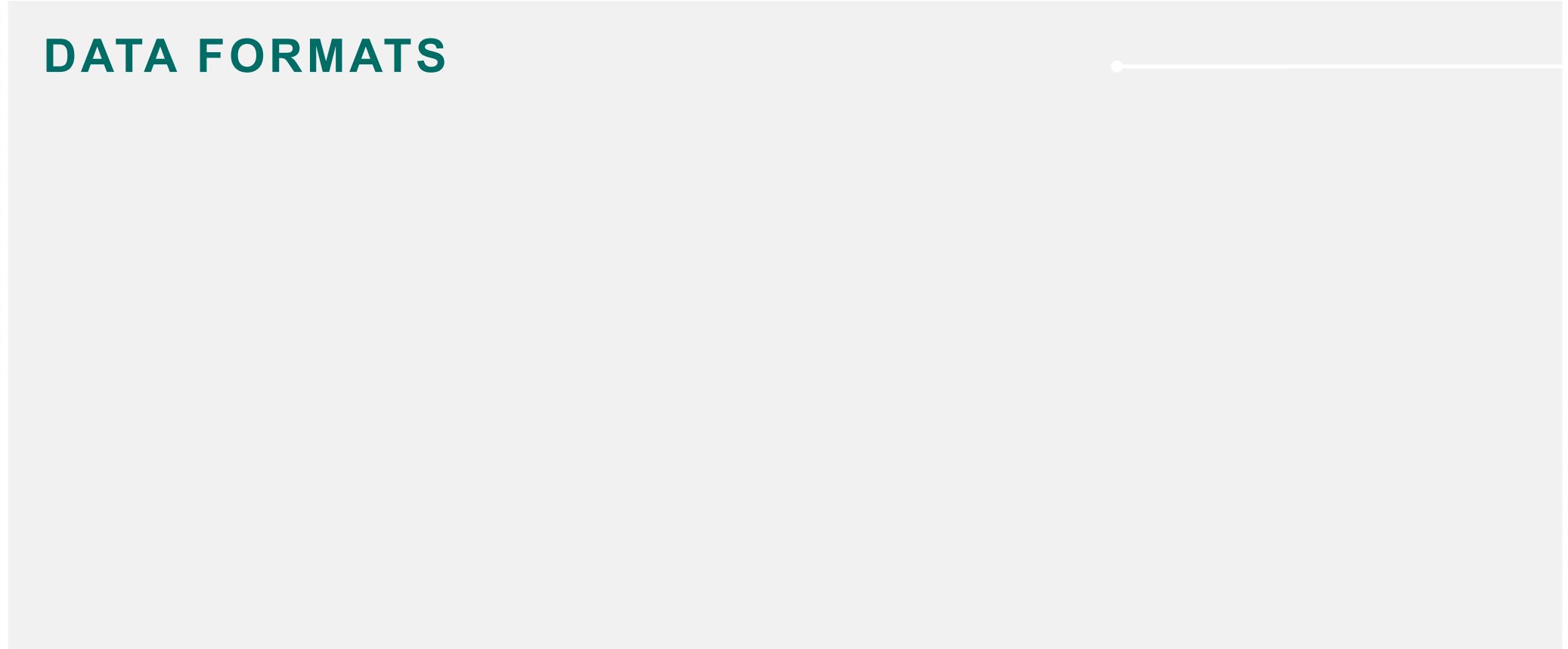
Other rules:

- Extract only a part of the data, filter and save what you need. Close that part
- Go on with the next part, until you went through the whole data.
- While the computer works: do something else. Wait for a few days.
- → Now you have much smaller data!





## DATA FORMATS





# DATA FORMATS - CSV

## Comma Separated Value

- A table, stored in a text file
  - one row of the table → one line in the text file
  - Values in the table are separated by commas

(Show example)

## Advantages:

- \* easy to read with any tool (Text editor, R, Python, Excel)
  - \* easy to write, share, store
  - \* future proof

## Disadvantages

- \* large, if left uncompressed
  - \* loses type information. A date has to be converted to a string when writing, and back to a date when reading. This can lead to errors.
  - \* Slower than some alternatives to read.

```
1 numa,ida,numb,idb,kmdist,midist
2 2,USA,20,CAN,731,456
3 2,USA,31,BHM,1623,1012
4 2,USA,40,CUB,1813,1130
5 2,USA,41,HAI,2286,1425
6 2,USA,42,DOM,2358,1471
7 2,USA,51,JAM,2315,1444
8 2,USA,52,TRI,3494,2179
9 2,USA,53,BAR,3330,2076
10 2,USA,54,DMA,3208,2001
11 2,USA,55,GRN,3332,2078
12 2,USA,56,SLU,3180,1983
13 2,USA,57,SVG,3240,2020
14 2,USA,58,AAB,2834,1767
15 2,USA,60,SKN,2760,1721
16 2,USA,80,BLZ,2606,1625
17 2,USA,70,MEX,3024,1885
18 2,USA,90,GUA,2993,1866
19 2,USA,91,HON,2922,1822
20 2,USA,92,SAL,3024,1886
```



## DATA FORMATS - JSON

(Show example)

Advantages:

- \* Widely used
- \* easy to read and write
- \* Hierarchical structure can be preserved

Disadvantages:

- \* verbose and large files
- \* slow to parse for very large files

<https://www.json.org/example.html>

```
{  
  "type": "node",  
  "id": 253073176,  
  "lat": 54.1527184,  
  "lon": 12.0648400,  
  "tags": {  
    "addr:city": "Rostock",  
    "addr:country": "DE",  
    "addr:housenumber": "6a",  
    "addr:postcode": "18109",  
    "addr:street": "Güstrower Straße",  
    "addr:suburb": "Lichtenhagen",  
    "brand": "Rewe",  
    "brand:wikidata": "Q16968817",  
    "brand:wikipedia": "en:REWE",  
    "name": "Rewe",  
    "old_name": "sky",  
    "opening_hours": "Mo-Sa 07:00-22:00",  
    "operator": "Supermärkte Nord Vertr",  
    "organic": "yes",  
    "shop": "supermarket",  
    "wheelchair": "yes"  
  }  
}
```





# DATA FORMATS - XML

Like a cross between json and HTML (but mostly the bad parts)

Some old systems still use it... so maybe you have to use it as well!

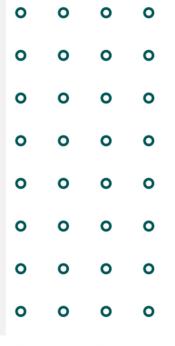
## Advantages:

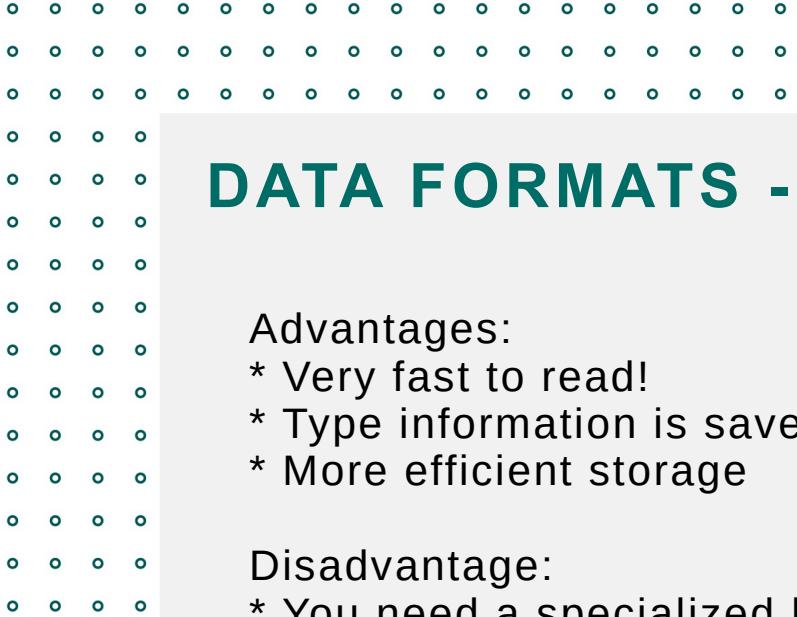
- \* Well structured
  - \* readable
  - \* Libraries for every language available

## Disadvantages:

- \* very verbose

```
<--<note>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
</note>
```





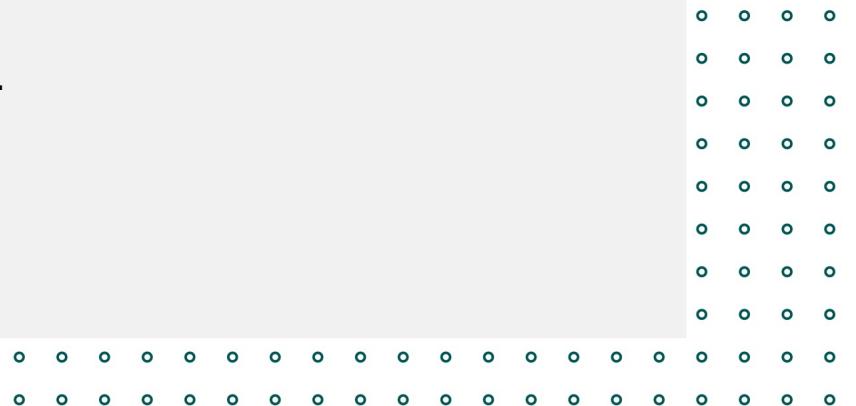
## DATA FORMATS - PARQUET

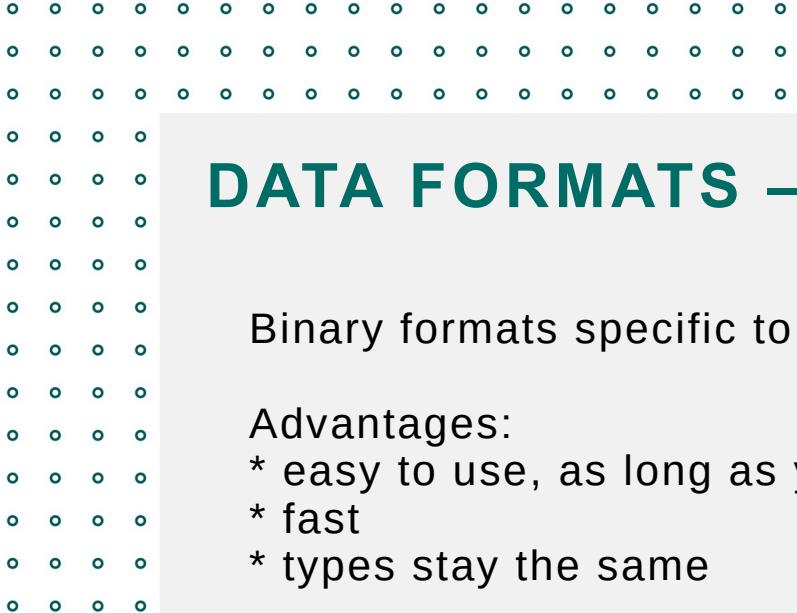
Advantages:

- \* Very fast to read!
- \* Type information is saved. A date will stay a date.
- \* More efficient storage

Disadvantage:

- \* You need a specialized library to write and read it (But it is a standard and can be shared between languages)
- \* Binary format – you can't use a text editor to look inside it.





## DATA FORMATS – PICKLE AND RDATA

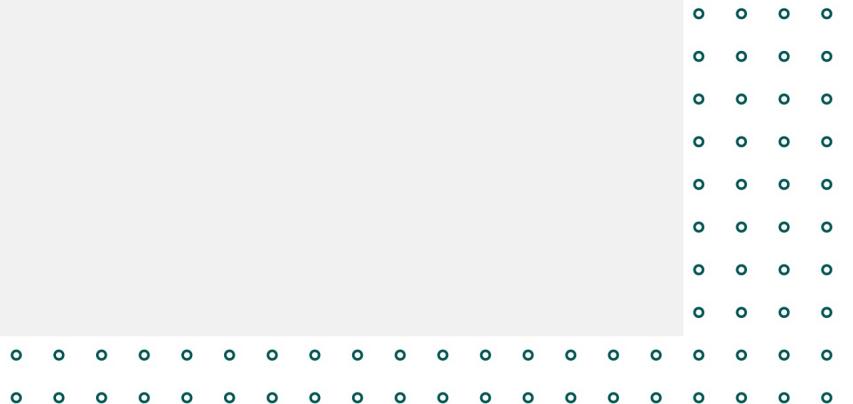
Binary formats specific to a programming language

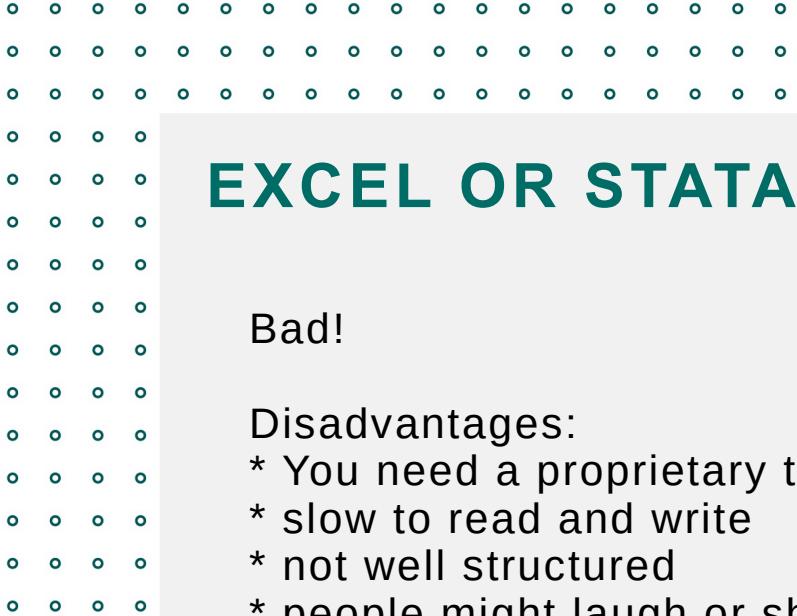
Advantages:

- \* easy to use, as long as your environment stays the same
- \* fast
- \* types stay the same

Disadvantages:

- \* No compatibility between languages
- \* → not good for sharing or long-time storage
- \* no metadata
- \* Binary format – you can't use a text editor to look inside it.





## EXCEL OR STATA

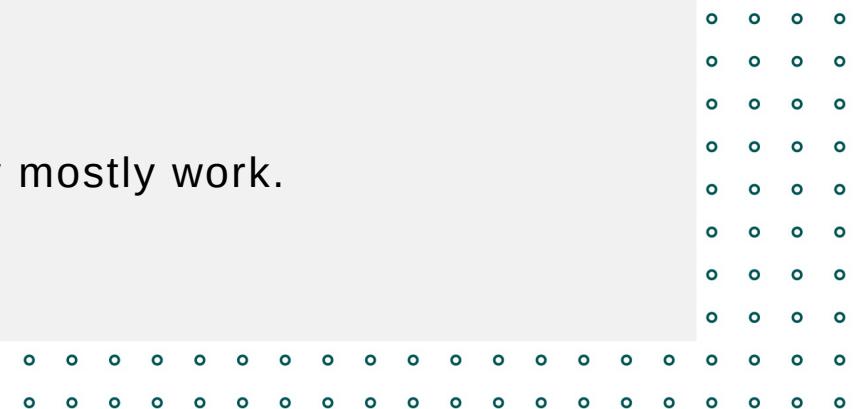
Bad!

Disadvantages:

- \* You need a proprietary tool to work properly with it
- \* slow to read and write
- \* not well structured
- \* people might laugh or shout at you

Advantages:

- \* nothing?
- \* There are libraries in R to read and write Excel files. They mostly work.
- \* Sometimes Excel might be „the right tool for the job“
- \* Good when working together with non-programmers

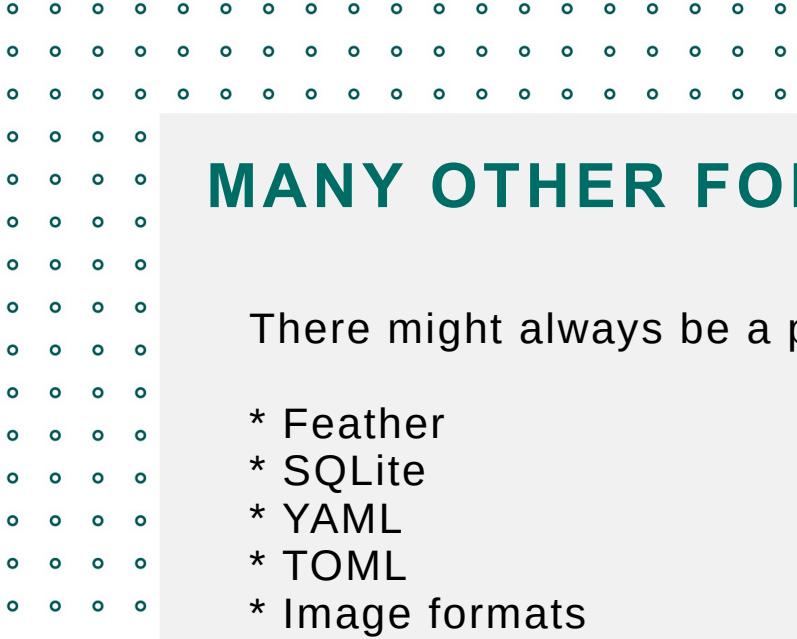




## PDF

Great for sharing documents. Bad for sharing data. If you get data in pdf-format...  
good luck!



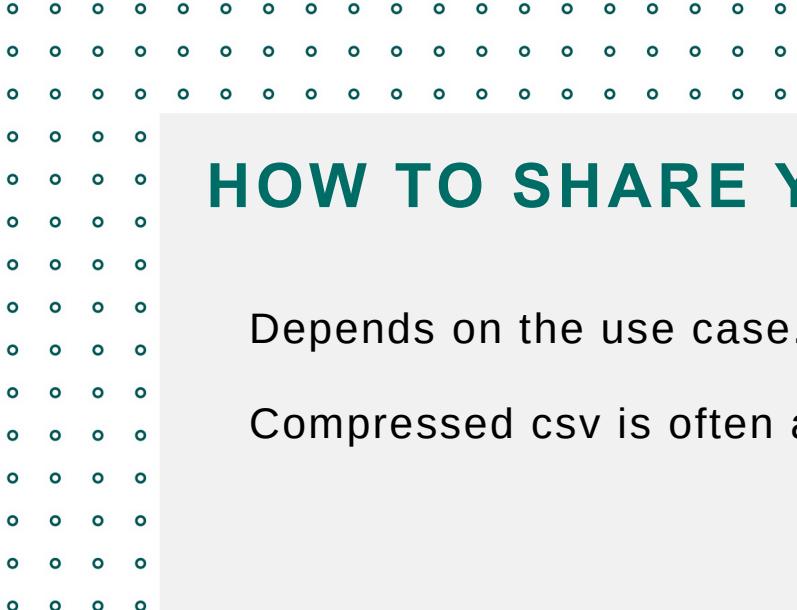


## MANY OTHER FORMATS

There might always be a package in R to read it.

- \* Feather
- \* SQLite
- \* YAML
- \* TOML
- \* Image formats
- \* Video formats





## HOW TO SHARE YOUR DATA

Depends on the use case.

Compressed csv is often a good choice.



THANK YOU FOR  
YOUR ATTENTION!



**Tom Theile**

Software developer at the lab of digital and computational demography

[theile@demogr.mpg.de](mailto:theile@demogr.mpg.de)



**THANK YOU!**

