



Efficient microdata

Efficient programming on the CBS
microdata environment

Erik-Jan van Kesteren

Webpage

<https://is.gd/microdataworkshop>

- Go there now
- Download the materials
- Download the exercise data
- Pray we don't crash uu wifi

Efficient microdata

Materials for a half-day workshop on efficient projects using CBS microdata.

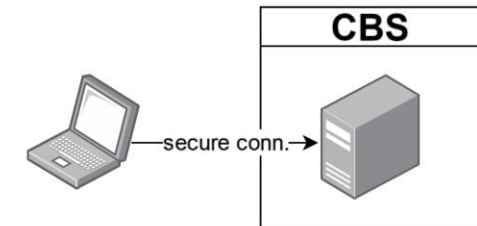
Workshop efficient microdata

Materials for a half-day workshop on efficient projects using CBS microdata. The materials on this page are [CC-BY-4.0](#) licensed.



Schedule

- Interactive instruction (2 hours)
 - Fundamentals of the CBS RA
 - Project structure & reproducibility
 - Code efficiency
- Exercise & consultation in groups (1 hour)



Materials

- Slides (pdf, without fancy gifs 🙄)
- Materials (to follow along with the materials and live coding sessions). After downloading, unzip and double click on the file `workshop_efficient_microdata.Rproj`.
- Exercise data (11.7 GB) to download and put in the folder `fake_cbs_data/Spolis`.

Contact

This project is developed and maintained by the [ODISSEI Social Data Science \(SoDa\)](#) team.



For questions about this course, you can contact us at soda@odissei-data.nl, or you can contact the instructor Erik-Jan (e.vankesteren1@uu.nl) directly.



We help social scientists with data intensive & computational research

Our goal is to enhance the evidence base and impact of social science by bringing the added value of new data sources and new data analysis techniques into social research in the Netherlands

part of  **ODISSEI**

[Contact us](#)[Get more info →](#)

<https://odissei-soda.nl/>

About me

Erik-Jan van Kesteren

Assistant Professor Data Science / Statistics

Team lead ODISSEI SoDa team

Research topics

structural equation modeling, regularization & penalization, Bayesian statistics, multilevel generalized linear models, geospatial data analysis, rank data analysis, visualisation, optimization, federated learning, privacy, fairness, measurement, synthetic data

Today

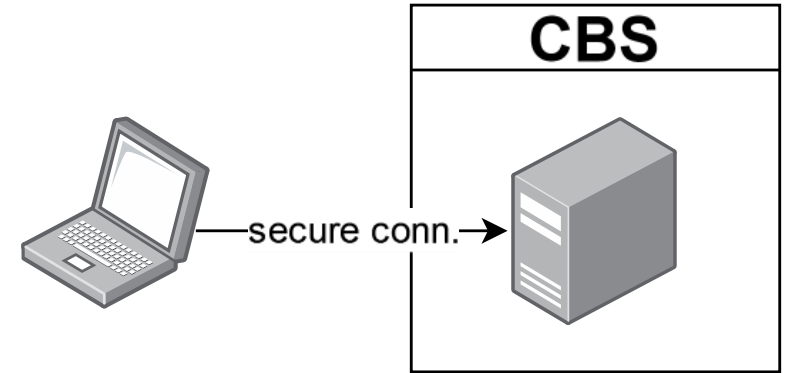
- The CBS RA fundamentals
- Project structure & reproducibility
- Efficient data handling
 - Storage
 - Memory
- Consultation & exercise!



CBS RA fundamentals

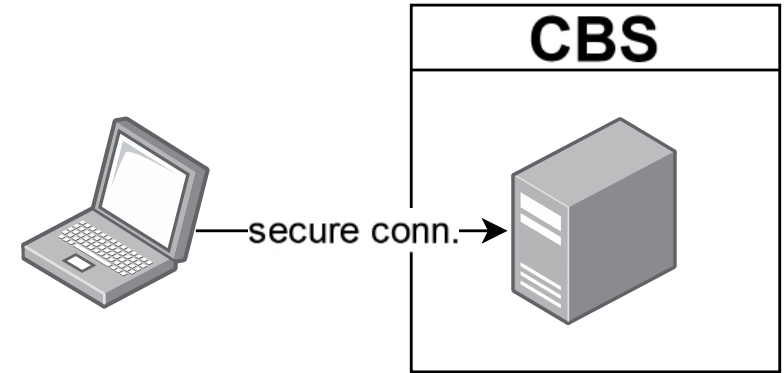
CBS Remote Access environment

- A virtual machine on a big server in the internal network
- Normal windows OS with 100GB storage
- R, python, SPSS, Stata, and more installed by default
- If you want specific libraries, ask the microdata team
- No fancy environment management



CBS Remote Access environment

- Data is made available via a drive on a per-project basis
G:/microdata
- You can see all tables, but you have access only to those you requested
- Additional metadata is also available (for everyone)



Microdata at CBS

- Register data and questionnaires
- You can (subject to restrictions and costs) also upload your own data & link
- All these tables can be combined to do your research!

<https://www.cbs.nl/nl-nl/onze-diensten/maatwerk-en-microdata/microdata-zelf-onderzoek-doen/catalogus-microdata>

Catalogus microdata

Onder strikte voorwaarden kunnen instanties microdata gebruiken om [zelf onderzoek](#) te doen. Hieronder ziet u per thema de recentste documentatierapporten van de beschikbare microdatabestanden:

- [Arbeid en sociale zekerheid](#)
- [Bedrijven](#)
- [Bevolking](#)
- [Bouwen en wonen](#)
- [Financiële en zakelijke diensten](#)
- [Gezondheid en welzijn](#)
- [Handel en horeca](#)
- [Inkomen en bestedingen](#)
- [Internationale handel](#)
- [Industrie en energie](#)
- [Landbouw](#)
- [Macro-economie](#)
- [Natuur en milieu](#)
- [Onderwijs](#)
- [Overheid en politiek](#)
- [Prijzen](#)
- [Veiligheid en recht](#)
- [Verkeer en vervoer](#)
- [Vrije tijd en cultuur](#)

Microdata at CBS

- The tables are made by humans / different departments: manual work
- They are (mostly) SPSS .sav files
- Some files are huge! (SPOLISBUS)
- Their names / versions can change without warning

<https://www.cbs.nl/nl-nl/onze-diensten/maatwerk-en-microdata/microdata-zelf-onderzoek-doen/catalogus-microdata>

Catalogus microdata

Onder strikte voorwaarden kunnen instanties microdata gebruiken om [zelf onderzoek](#) te doen. Hieronder ziet u per thema de recentste documentatierapporten van de beschikbare microdatabestanden:

- [Arbeid en sociale zekerheid](#)
- [Bedrijven](#)
- [Bevolking](#)
- [Bouwen en wonen](#)
- [Financiële en zakelijke diensten](#)
- [Gezondheid en welzijn](#)
- [Handel en horeca](#)
- [Inkomen en bestedingen](#)
- [Internationale handel](#)
- [Industrie en energie](#)
- [Landbouw](#)
- [Macro-economie](#)
- [Natuur en milieu](#)
- [Onderwijs](#)
- [Overheid en politiek](#)
- [Prijzen](#)
- [Veiligheid en recht](#)
- [Verkeer en vervoer](#)
- [Vrije tijd en cultuur](#)

Additional data

- There are additional files to help with analysis
- Metadata & supplementary data. Translation files, key/value files, lists of existing postal codes, and more.
- These reside in a different location (not G:/)
- This location has also changed in the past & could change in the future too

Imports/exports

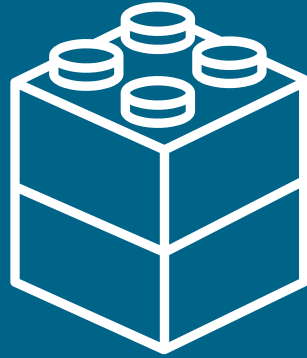
Exporting analysis results is subject to output check

- Ensures our privacy
- This is manual labour, done by microdata team member
- Each output costs time and money

You can also import and export code files

- This does not cost money!!
- More on this later

Any questions?



Structure

Efficient project folder structure

```
my_project/  
├── raw_data/  
│   ├── questionnaire_data.csv  
├── processed_data/  
│   ├── questionnaire_processed.rds  
│   └── analysis_object.rds  
├── img/  
│   └── plot.png  
├── 01_load_and_process_data.R  
├── 02_create_visualisations.R  
├── 03_main_analysis.R  
├── 04_output_results.R  
├── my_project.Rproj  
└── readme.md
```

Efficient project folder structure

my_project/

└ raw_data/	At CBS, this is on a different disk (G:/)! Does not count towards your 100GB quote
└─ questionnaire_data.csv	
└ processed_data/	
└─ questionnaire_processed.rds	
└─ analysis_object.rds	
└ img/	
└─ plot.png	
└ 01_load_and_process_data.R	
└ 02_create_visualisations.R	
└ 03_main_analysis.R	
└ 04_output_results.R	
└ my_project.Rproj	
└ readme.md	

Efficient project folder structure

my_project/

└ raw_data/ └ questionnaire_data.csv	At CBS, this is on a different disk (G:/)! Does not count towards your 100GB quote
└ processed_data/ └ questionnaire_processed.rds └ analysis_object.rds └ img/ └ plot.png	Make these objects efficiently stored Binary formats are better later we'll also see database files
└ 01_load_and_process_data.R	
└ 02_create_visualisations.R	
└ 03_main_analysis.R	
└ 04_output_results.R	
└ my_project.Rproj	
└ readme.md	

Efficient project folder structure

my_project/

<ul style="list-style-type: none">└ raw_data/<ul style="list-style-type: none">└ questionnaire_data.csv	At CBS, this is on a different disk (G:/)! Does not count towards your 100GB quote
<ul style="list-style-type: none">└ processed_data/<ul style="list-style-type: none">└ questionnaire_processed.rds└ analysis_object.rds└ img/<ul style="list-style-type: none">└ plot.png	Make these objects efficiently stored Binary formats are better later we'll also see database files
<ul style="list-style-type: none">└ 01_load_and_process_data.R└ 02_create_visualisations.R└ 03_main_analysis.R└ 04_output_results.R	Clear ordering & separation of tasks Separating preprocessing & analysis
<ul style="list-style-type: none">└ my_project.Rproj└ readme.md	

Efficient project folder structure

my_project/

<ul style="list-style-type: none">└ raw_data/<ul style="list-style-type: none">└ questionnaire_data.csv	At CBS, this is on a different disk (G:/)! Does not count towards your 100GB quote
<ul style="list-style-type: none">└ processed_data/<ul style="list-style-type: none">└ questionnaire_processed.rds└ analysis_object.rds└ img/<ul style="list-style-type: none">└ plot.png	Make these objects efficiently stored Binary formats are better later we'll also see database files
<ul style="list-style-type: none">└ 01_load_and_process_data.R└ 02_create_visualisations.R└ 03_main_analysis.R└ 04_output_results.R	Clear ordering & separation of tasks Separating preprocessing & analysis
<ul style="list-style-type: none">└ my_project.Rproj	Use .Rproj file for portability
<ul style="list-style-type: none">└ readme.md	

Live coding 1: example project

DOI [10.5281/zenodo.6504837](https://doi.org/10.5281/zenodo.6504837)



Reproducibility & offline workflow

Reproducibility

- After your project, export your structured code folder!
- Double check your documentation
- Create a DOI / archived version (OSF, Zenodo)
- <https://odissei-soda.nl/tutorials/post-1/>
- This way, others can benefit from your work
- Let ODISSEI know 😊
- <https://odissei-data.github.io/ODISSEI-code-library/>

Reproducibility

You cannot export data; create synthetic data

- Create “fake” version of input data outside the RA
- You can do this with a script
- Don’t do this for all input tables (too much work!)
- Do this for the intermediate processed table

Computational reproducibility

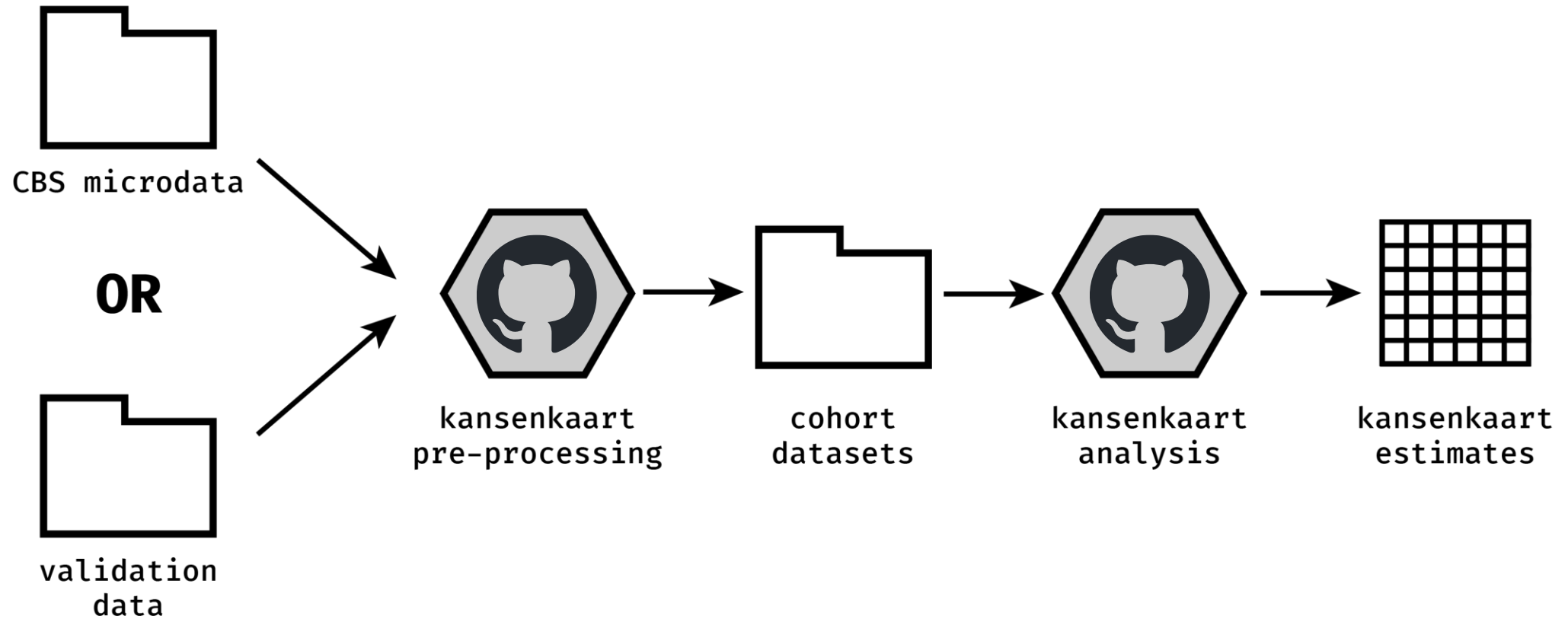
Offline workflow example

You might want to go a step further with version control

1. Develop preprocessing & analysis on GitHub
2. Upload to CBS RA
3. Do some small code fixes there to make it run
4. Run once, export results
5. Export code from CBS, update GitHub repository

Recommended for complex / multi-collaborator projects

Offline workflow example

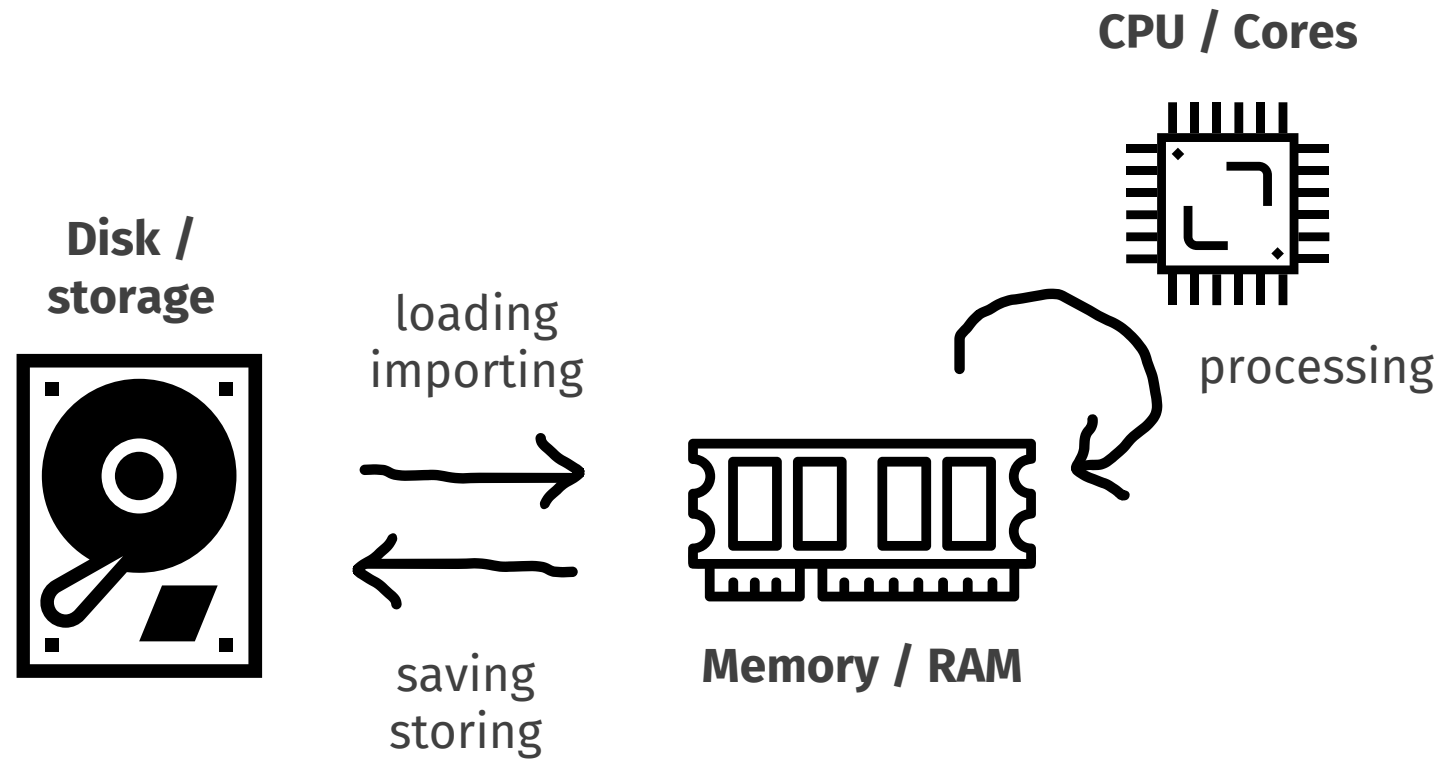


Offline workflow example

How far you should go depends a lot on the project

- Contact us in consultation time later or schedule a meeting with us

Efficient data handling





Storage

Geachte relatie,

Uit een meting op maandag 4 april 2022 blijkt dat project 0000, Titel van het project, een ruimtebeslag kent van **133** GB. De limiet voor het project is **100** GB.

Als u de extra capaciteit daadwerkelijk nodig heeft, dan kunt u een verzoek indienen om extra capaciteit bij te kopen. De kosten hiervoor bedragen 25 euro per 50 GB per maand.

Met vriendelijke groet,

Firstname Lastname

DBD Team Dataservices

CBS | Henri Faasdreef 312 | Postbus 24500 | 2490 HA
Den Haag

Email: microdata@cbs.nl

Volg [statistiekcb](#)s op twitter | facebook | instagram

Efficiently storing large R datasets

Live coding

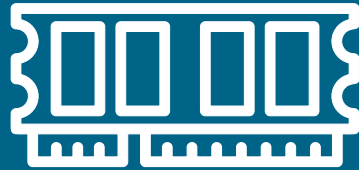
Try to create tidy data

Intermediate data should be **tidy**

- Each variable is a column; each column is a variable.
- Each observation is a row; each row is an observation.
- Each value is a cell; each cell is a single value

<https://tidyr.tidyverse.org/articles/tidy-data.html>

Make column names legible for humans and machines
`janitor::clean_names()`



Memory

In R:

Error: cannot allocate vector of size 745.1 Gb

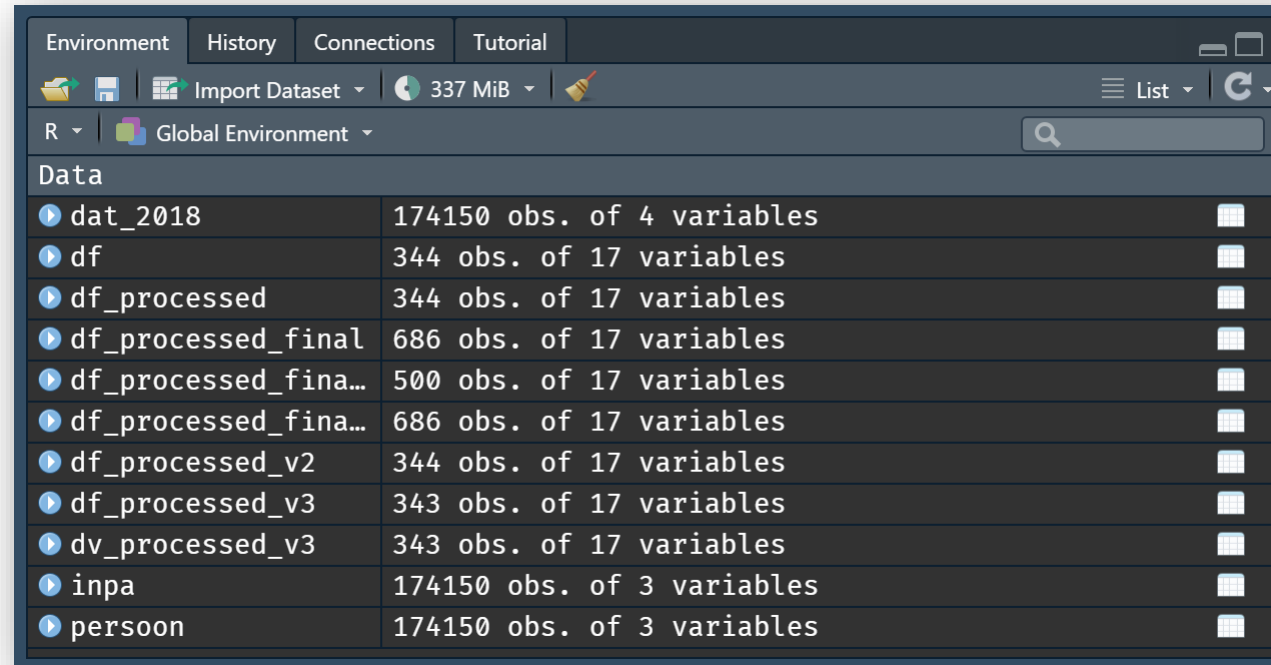
In Python (numpy)

```
numpy.core._exceptions._ArrayMemoryError:  
Unable to allocate 745. GiB for an array with  
shape (1000000000000000,) and data type float64
```












In Stata

(no clue, I really don't use Stata??)

Clean your session / environment



The screenshot shows the RStudio Environment pane. At the top, there are tabs for 'Environment', 'History', 'Connections', and 'Tutorial'. Below the tabs is a toolbar with icons for file operations and a memory usage indicator showing '337 MiB'. The main area displays the 'Global Environment' with a search bar. A table titled 'Data' lists the objects in the environment, including their names, sizes, and variable counts.

Data		
▶ dat_2018	174150 obs. of 4 variables	
▶ df	344 obs. of 17 variables	
▶ df_processed	344 obs. of 17 variables	
▶ df_processed_final	686 obs. of 17 variables	
▶ df_processed_fina...	500 obs. of 17 variables	
▶ df_processed_fina...	686 obs. of 17 variables	
▶ df_processed_v2	344 obs. of 17 variables	
▶ df_processed_v3	343 obs. of 17 variables	
▶ dv_processed_v3	343 obs. of 17 variables	
▶ inpa	174150 obs. of 3 variables	
▶ persoon	174150 obs. of 3 variables	

Efficiently processing large datasets

Live coding 2

Larger-than-memory data

- Sometimes, your data really is larger-than-memory
- It is possible to do analyses on datasets which are on-disk

Two options:

- Process data in chunks
- Create a proper database

Working with larger-than-memory data

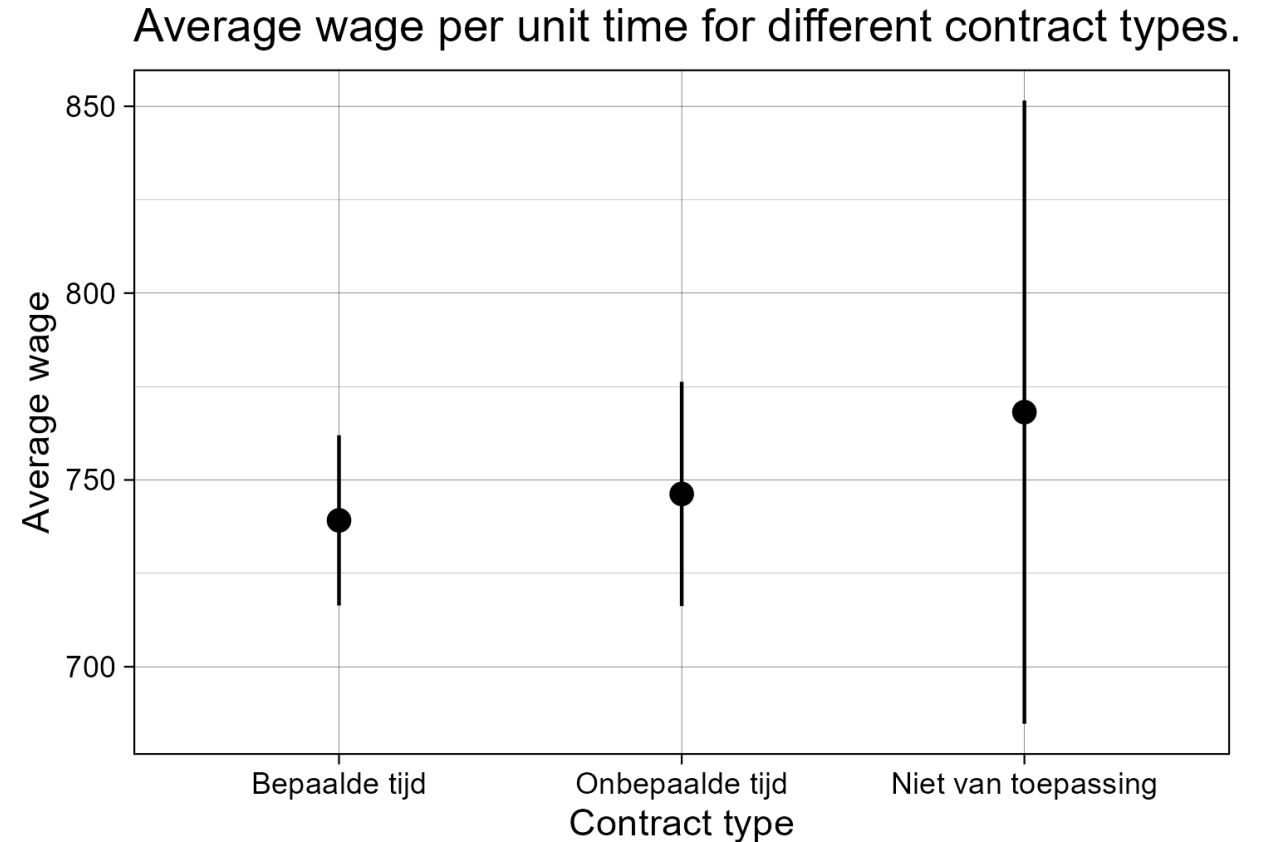
Live coding 3

Questions?

Group consultations / exercise

Exercise

- Fake SPOLISBUS
- Create a “simple” plot
- Use the skills from this workshop!
- Discuss with your group members!



Groups

CBS RA data, data handling, exercise, reproducibility, big tables, advanced computation, version control

- Discuss your questions; were they answered? Do you have more questions?
- Help each other!
- Do the exercise 😊

Members of the SoDa team will walk around

First name	Last name	Group
Henry	Abbink	Data handling
Nathalie	Aerts	CBS RA data
Nadya	Ali	Exercise
Stanislav	Avdeev	Data handling
Özgün	Balaban	CBS RA data
Dorien	Beeres	CBS RA data
Cindy	Biesenbeek	Exercise
Clémentine	Cottineau	Advanced computation
Ernst-Jan	de Bruijn	Reproducibility
Niek	de Meijier	CBS RA data
Sreeparna	Deb	CBS RA data
Bob	Driessen	Reproducibility
Federica	Giardina	Data handling
Kyri	Janssen	Big tables
Yuxuan	Jin	Data handling
Steven	Kema	Data handling
Megha	Khosla	Advanced computation
Maša	Krajnc	Big tables
Marie	Labussiere	Reproducibility
Huyen	Nguyen	CBS RA data
Marjanne	Piena	Big tables
Friso	Scheepstra	Version control
Raoul	Schram	CBS RA data
Elizaveta	Sivak	Big tables
Florian	Sniekers	Version control
Pascalie	Spaan	CBS RA data
Lisa	Timm	Version control
Ignacio	Urria	Advanced computation
Mark	van der Meijden	Exercise
Kars	van Oosterhout	Data handling
Michiel	van Rijn	Exercise
Alexandra	Voce	CBS RA data

Thank you!



<https://odissei-soda.nl>