



Efficient microdata

Efficient programming on the CBS
microdata environment

Erik-Jan van Kesteren

Today

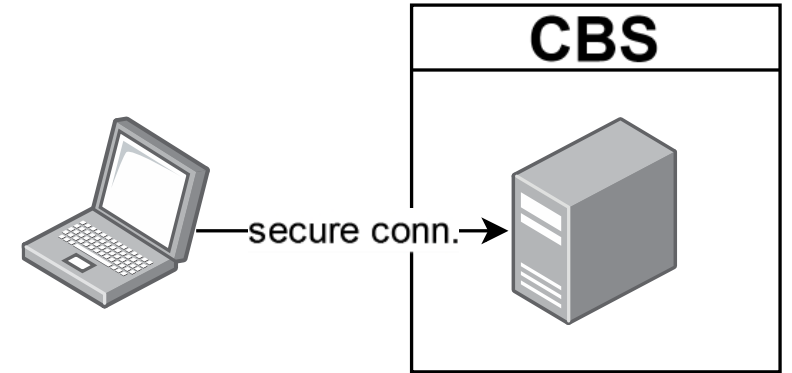
- The CBS RA fundamentals
- Project structure & reproducibility
- Efficient data handling
 - Storage
 - Memory
- Consultation & exercise!



CBS RA fundamentals

CBS Remote Access environment

- A virtual machine on a big server in the internal network
- “Normal” windows environment
- Data is made available via a drive on a per-project basis
G:/microdata
- Additional metadata is also available



Microdata at CBS

- Register data and questionnaires
- You can (subject to restrictions and costs) also upload your own data
- 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 (meta)data 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

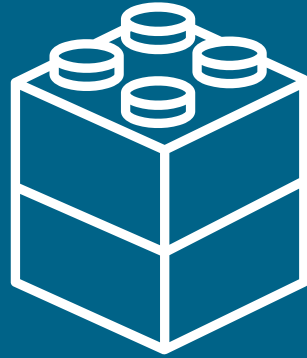
Exporting analysis results is subject to output check

- Ensures our privacy
- This is manual labour, done by a member of microdata team
- 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 & reproducibility

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	Make these objects efficiently stored Depends on your application
└─ 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	Make these objects efficiently stored Depends on your application
└ img/ └ plot.png	
└ 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
└ 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	

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)

Today

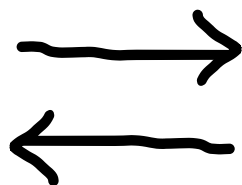
- The CBS RA fundamentals
- Project structure & reproducibility
- Efficient data handling
 - Storage
 - Memory
- Consultation & exercise!



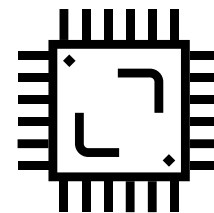
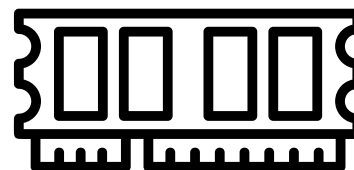
Efficient data handling



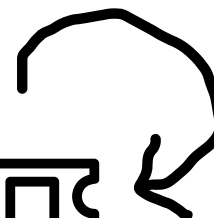
loading
importing



saving
storing



processing





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

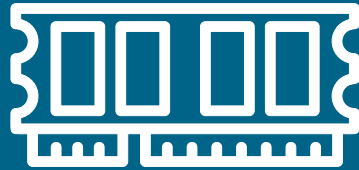
CBS | Henri Faasdreef 312 | Postbus 24500 | 2490 HA
Den Haag

Email: microdata@cbs.nl

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

Efficiently storing large R datasets

Live coding



Memory

Top tip #4

**Read your program's error messages!
They give a lot of diagnostic info**

In R:

Error: cannot allocate vector of size 745.1 Gb

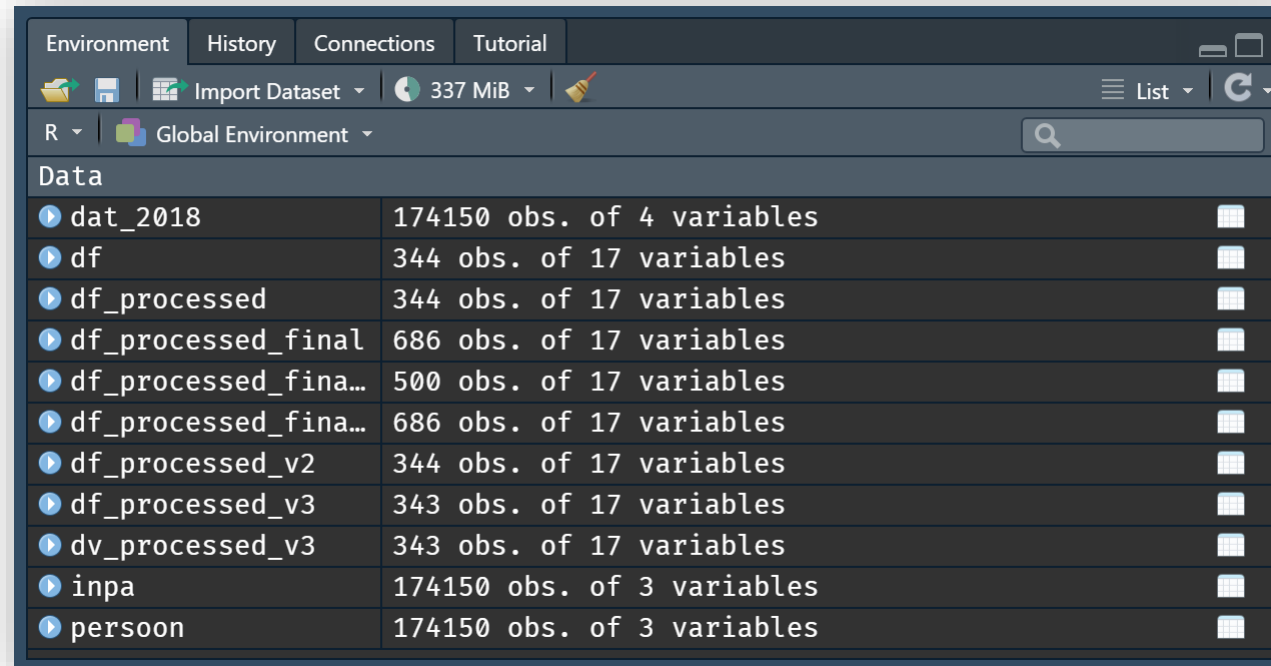
In Python (numpy)

```
numpy.core._exceptions._ArrayMemoryError:  
Unable to allocate 745. GiB for an array with  
shape (100000000000000,) 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. Each row has a blue arrow icon on the left and a grid icon on the right.

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:

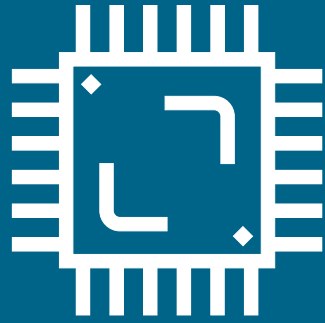
- Create chunked data objects
- Create a proper database

Top tip #5

Investigate whether the “heavy” RA machine will solve your memory issues

Working with larger-than-memory data

Live coding 3



Compute

Compute-heavy applications

- Large simulations, e.g.,
 - agent-based models
 - computational models
 - complex systems stuff
 - statistical simulations (large power analyses)
- Many different conditions
 - Perform some computation for each neighbourhood in NL
- Bayesian estimation with large models (many parameters, many posterior samples)

Speeding up a function with C++

Live coding 4

Embarrassingly parallel

Many independent computations, little or no effort is needed to separate the problem into a number of parallel tasks

- Simulations
- Applying a function to many conditions
- Running a piece of code with many different settings
- Bootstrapping

Top tip #6

Is your problem parallelizable? Look into the ODISSEI Secure Supercomputer

Supercomputing for Social Scientists with R

Would you like to understand how to work with a supercomputer and translate your R workflow from a graphical-user-interface (GUI) on your desktop to a scripting/automated workflow that leverages the resources of a supercomputer?



[Sign up](#)

 **24 May 2022**

 **9.00-17.00**

 **Online**

[Sign up](#) 

Event type

Training

Prerequisite knowledge

No prior knowledge required

Costs

Free

Top tips, collected

- Run your heavy tasks during low-intensity hours on the RA environment
- If you can afford it, just buy extra storage space for your project 😊
- Create a clear code folder, export your code from the RA, and publish it!
- Read your program's error messages! They give a lot of diagnostic info
- Investigate whether the “heavy” RA machine will solve your memory issues
- Is your problem parallelizable? Look into the ODISSEI Secure Supercomputer
- Want to know more? Join the workshop.

Thank you!



<https://odissei-data.nl>

<https://www.surf.nl/en/agenda/supercomputing-for-social-scientists-with-r>

https://github.com/sodascience/cbs_microdata_computing

[@SoDa_NL](#)

Questions?

Default light slide

Default subheading

This is the body of the text

Default subheading

Note that the text is not black, but “black, text 1, lighter 25%”

Default subheading

This makes things easier on the eyes

Default subheading

This is the body of the text

Default dark slide

Default subheading

The dark slide brings some variation

Default subheading

It can highlight important aspects of the presentation.

Default subheading

This is the body of the text

Default subheading

This is the body of the text

Is this an impact slide?

Here is an impactful slide with a sentence on it.

Here is a topic related to the aforementioned question.