



Applied Geodata Science I

# Session 5

Prof. Dr. Benjamin Stocker

17.03.2025

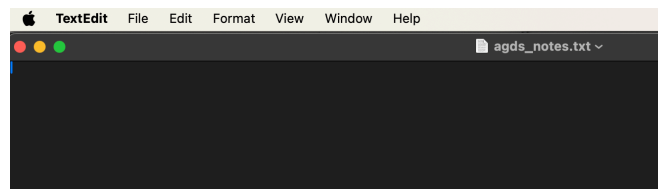


# File name extensions

```
(base) ~/trash >ls -l
ERC_CoG_2025_B2_RESHDIP_20250106.pdf
agds_notes.txt
data_variety.Rmd
demo_data.nc
geco_logo_small.png
ikea.jpg
lithium_circubat.R
styles.css
test.py
testmemory.f90
```

- File name extensions indicate the file format.
- Extensions are interpreted by the operating system to determine the program for opening the respective file.

```
~/trash >open agds_notes.txt
```



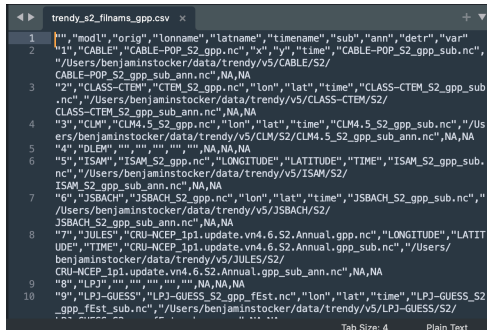
- ```
~/trash >subl trendy_s2_filnams_gpp.csv
```

Tab Size: 4      Plain Text

# Tabular data (text-based)

- Organised by rows and columns.
- Can be read as data frame.

```
~/trash > subl trendy_s2_filnams_gpp.csv
```



```
1 | "modl","orig","lonname","latname","timename","sub","ann","detr","var"  
2 | "1","CABLE","CABLE-POP_S2_gpp.nc","x","y","time","CABLE-POP_S2_gpp_sub.nc",  
  | "/Users/benjaminstocker/data/trendy/v5/CABLE/S2/  
  | CABLE-POP_S2_gpp_sub_ann.nc",NA,NA  
3 | "2","CLASS-CTEM","CTEM_S2_gpp.nc","lon","lat","time","CLASS-CTEM_S2_gpp_sub  
  | ".nc","/Users/benjaminstocker/data/trendy/v5/CLASS-CTEM/S2/  
  | CLASS-CTEM_S2_gpp_sub_ann.nc",NA,NA  
4 | "3","CLM","CLM4.5_S2_gpp.nc","lon","lat","time","CLM4.5_S2_gpp_sub.nc","/Us  
  | ers/benjaminstocker/data/trendy/v5/CLM/S2/CLM4.5_S2_gpp_sub_ann.nc",NA,NA  
5 | "4","DLEM","""","""","""","""","""","""","""","""","""","""","""","""","""","""",  
  | "4","DLEM","""","""","""","""","""","""","""","""","""","""","""","""","""","""",  
  | "5","ISAM","ISAM_S2_gpp.nc","LONGITUDE","LATITUDE","TIME","ISAM_S2_gpp_sub.  
  | nc","/Users/benjaminstocker/data/trendy/v5/ISAM/S2/  
  | ISAM_S2_gpp_sub_ann.nc",NA,NA  
6 | "6","JSBACH","JSBACH_S2_gpp.nc","lon","lat","time","JSBACH_S2_gpp_sub.nc",  
  | "/Users/benjaminstocker/data/trendy/v5/JSBACH/S2/  
  | JSBACH_S2_gpp_sub_ann.nc",NA,NA  
7 | "7","JULES","CRU-NCEP_1p1.update.vn4.6.S2.Annual.gpp.nc","LONGITUDE","LATIT  
  | UDE","TIME","CRU-NCEP_1p1.update.vn4.6.S2.Annual.gpp_sub.nc","/Users/  
  | benjaminstocker/data/trendy/v5/JULES/S2/  
  | CRU-NCEP_1p1.update.vn4.6.S2.Annual.gpp_sub_ann.nc",NA,NA  
8 | "8","LPJ","""","""","""","""","""","""","""","""","""","""","""","""","""","""",  
  | "9","LPJ-GUESS","LPJ-GUESS_S2_gpp_Test.nc","lon","lat","time","LPJ-GUESS_S2  
  | _gpp_Test_sub.nc","/Users/benjaminstocker/data/trendy/v5/LPJ-GUESS/S2/  
  | ... ..  
  | Tab Size: 4 Plain Text
```

In R:

```
> df <- read_csv("~/trash/trendy_s2_filnams_gpp.csv")
```

```
> df  
# A tibble: 17 × 10  
  ...1 modl orig lonname latname timename sub ann  
  <dbl> <chr> <chr> <chr> <chr> <chr> <chr>  
1 1 CABLE CABLE-POP_S2_... x y time CABL... /Use...  
2 2 CLASS-CTEM CTEM_S2_gpp.nc lon lat time CLAS... /Use...  
3 3 CLM CLM4.5_S2_gpp... lon lat time CLM4... /Use...  
4 4 DLEM NA NA NA NA NA NA  
5 5 ISAM ISAM_S2_gpp.nc LONGIT... LATITU... TIME ISAM... /Use...  
6 6 JSBACH JSBACH_S2_gpp... lon lat time JSBA... /Use...  
7 7 JULES CRU-NCEP_1p1... LONGIT... LATITU... TIME CRU-... /Use...  
8 8 LPJ NA NA NA NA NA NA  
9 9 LPJ-GUESS LPJ-GUESS_S2_... lon lat time LPJ-... /Use...  
10 10 OCN NA NA NA NA NA NA  
11 11 LPX-Bern LPX_S2_gpp.nc LONGIT... LATITU... TIME LPX_... /Use...  
12 12 ORCHIDEE orchidee_S2_g... LONGIT... LATITU... TIME orch... /Use...  
13 13 ORCHIDEE-MICT NA NA NA NA NA NA  
14 14 SDGVM SDGVM_S2_gpp... LONGIT... LATITU... TIME SDGV... /Use...  
15 15 SURFEX NA NA NA NA NA NA  
16 16 VEGAS VEGAS_S2_gpp... lon lat time VEGA... /Use...  
17 17 VISIT VISIT_S2_gpp... lon lat time VISI... /Use...
```

# Structured data (text-based)

- Attribute-value pairs
- Hierarchy of attributes
- Common formats: JSON, XML

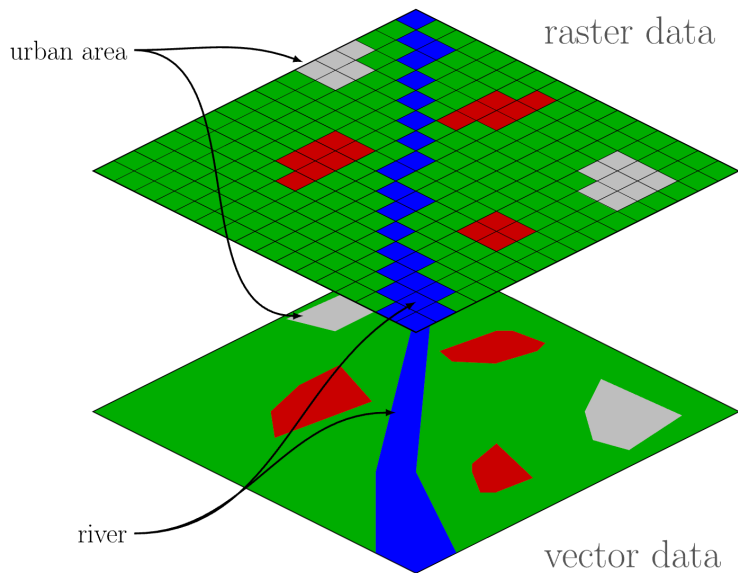
```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [
    "Catherine",
    "Thomas",
    "Trevor"
  ],
  "spouse": null
}
```

- Not represented as text characters.
- Common file types:
  - Executable (application)
  - Image (.png, .jpg., tiff), audio (.mp3), video files (.mp4)
  - (Layered) raster data with geographic reference (GeoTIFF: .tiff, .geotiff)
  - Array-oriented data, typically with geographic (and temporal) reference (NetCDF: .nc)
  - Shapefile: structured data with geographic reference (.shp + .shx + .dbf)
  - Structured binary data (HDF5: .hdf)

- Provides information for reading and interpreting data.
- In text-based files often contained in the header.
- In binary data, need special commands for writing/reading meta data.

```
# This is meta-data associated with the tabular CSV file
# for which the data is listed below.
#
# In addition to some meta-data, the first row of the data
# contains the column header data
column_one, column_two, column_three
1, 2, 3
1, 2, 3
1, 2, 3
```

[https://geco-bern.github.io/handfull\\_of\\_pixels/geospatial\\_R.html](https://geco-bern.github.io/handfull_of_pixels/geospatial_R.html)



- (Layered) raster data with geographic reference (GeoTIFF: .tiff, .geotiff)
- Array-oriented data, typically with geographic (and temporal) reference (NetCDF: .nc)
- Shapefile: structured data with geographic reference (.shp + .shx + .dbf)



# Online data sources: known URL of a file



```
# define a URL with data of interest
# in this case annual mean CO2 levels at Mauna Loa
url <- "https://gml.noaa.gov/webdata/ccgg/trends/co2/co2_annmean_mlo.csv"

# read in the data directly from URL
df <- read.table(
  url,
  header = TRUE,
  sep = ",",
)
```

- Application Programming Interface (API) defines how to access online data.
- Dedicated API libraries facilitate online data access for specific sources.

```
# load the library
library("MODISTools")

# list all available products
products <- MODISTools::mt_products()

# print the first few lines
# of available products
print(head(products))
```

# Online data sources: general-purpose API functions with httr



- GET accesses data, given a source-specific query.
- The query specifies attribute-value pairs for a given API.
- Read the documentation of a given API for required attributes.

```
# formulate a named list query to pass to httr
query <- list(
  "argument" = "2",
  "another_argument" = "3"
)

# The URL of the API (varies per product / param)
url <- "https://your.service.endpoint.com"

# download data using the
# API endpoint and query data
# status variable will include if
# the download was successful or not
# the write_disk() function captures
# data if available and writes it to
# disk
status <- httr::GET(
  url = url,
  query = query,
  httr::write_disk(
    path = "/where/to/store/data/filename.ext",
    overwrite = TRUE
  )
)
```