Pacific Northwest
NATIONAL LABORATORY

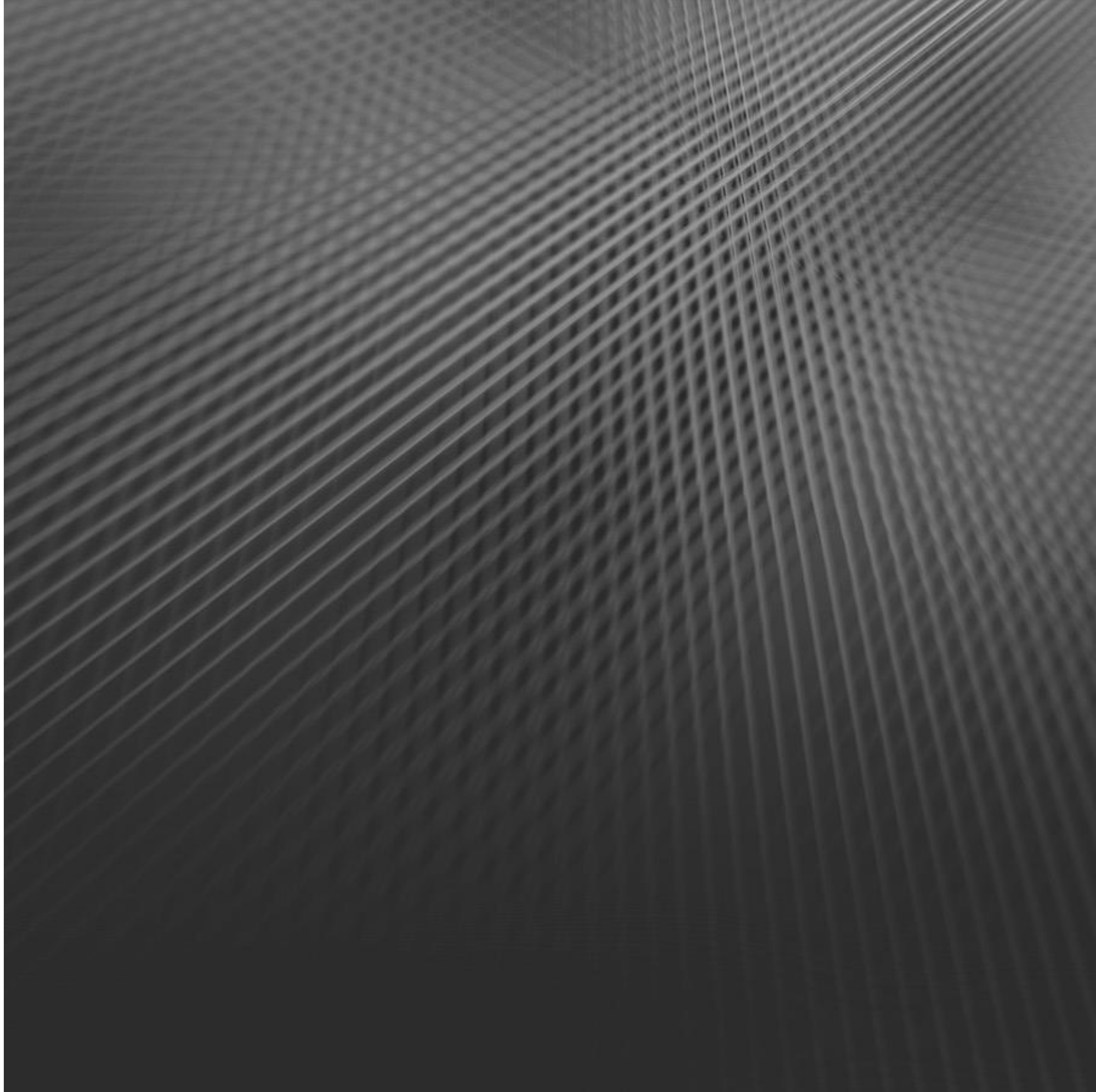
gcamdata
GCAM’s Data System

8 June 2023

Ellie Lochner
Joint Global Change Research Institute

U.S. DEPARTMENT OF ENERGY BATTELLE

PNNL is operated by Battelle for the U.S. Department of Energy



Outline

1. What is *gcamdata*?
2. File structure and naming conventions
3. How to run
4. *Renv* package management
5. Modifying *gcamdata*
6. Debugging
7. Useful functions
8. Common issues

What is *gcamdata*?

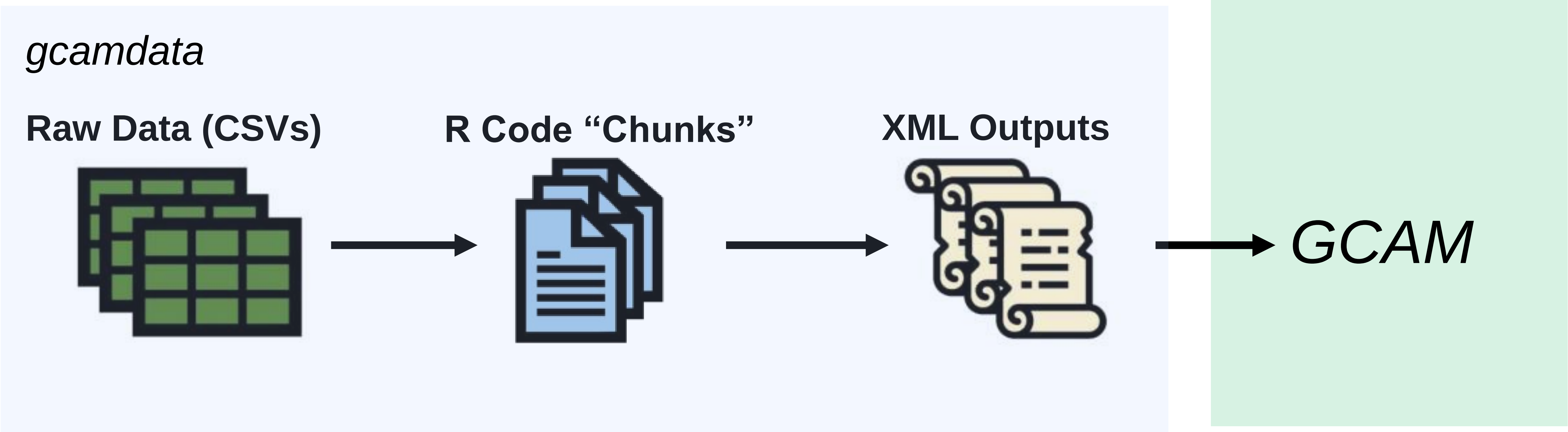
- R package that processes raw inputs to produce the hundreds of XML files needed by GCAM

- GCAM requires a lot of input data (energy, emissions, land-use, water, etc.)
  - Data System History: Spreadsheets → Collection of R scripts → Contained R package

- Developed in response to needs to handle more and more data, better documentation, updated coding practices, reproducibility, framework for new development

- Repository: https://github.com/JGCRI/gcamdata

What is *gcamdata*?

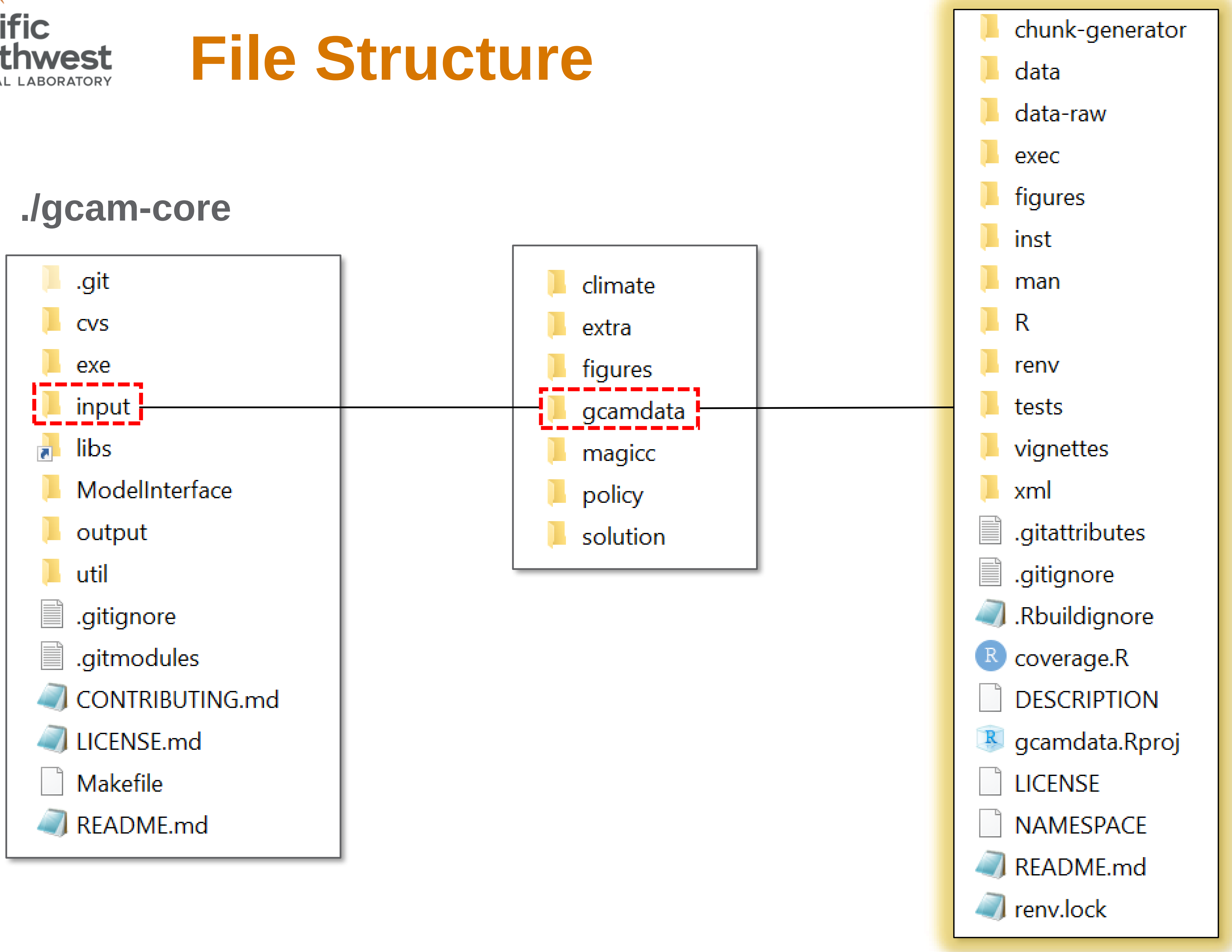


gcamdata

Raw Data (CSVs) → R Code “Chunks” → XML Outputs → *GCAM*

File Structure

./gcam-core



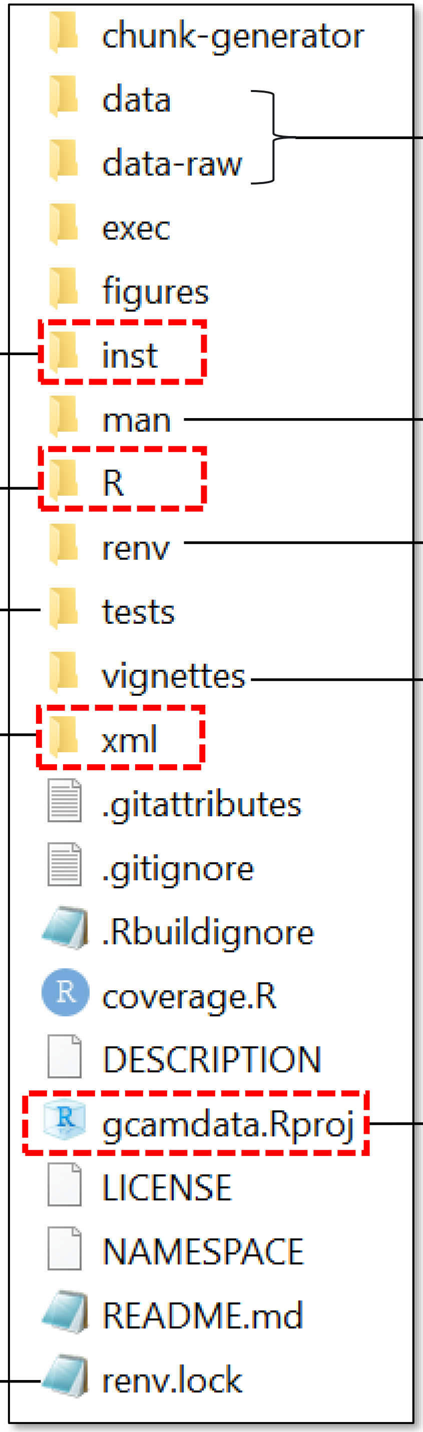
- .git
- cvs
- exe
- input
- libs
- ModelInterface
- output
- util
- .gitignore
- .gitmodules
- CONTRIBUTING.md
- LICENSE.md
- Makefile
- README.md

- climate
- extra
- figures
- gcamdata
- magicc
- policy
- solution

- chunk-generator
- data
- data-raw
- exec
- figures
- inst
- man
- R
- renv
- tests
- vignettes
- xml
- .gitattributes
- .gitignore
- .Rbuildignore
- coverage.R
- DESCRIPTION
- gcamdata.Rproj
- LICENSE
- NAMESPACE
- README.md
- renv.lock

File Structure

./gcam-core/input/gcamdata



chunk-generator
data
data-raw
exec
figures
inst
man
R
renv
tests
vignettes
xml
.gitattributes
.gitignore
.Rbuildignore
coverage.R
DESCRIPTION
gcamdata.Rproj
LICENSE
NAMESPACE
README.md
renv.lock

R data files and script that creates them

Raw input files

Documentation files

Processing code

Symlinks to R packages

Package tests

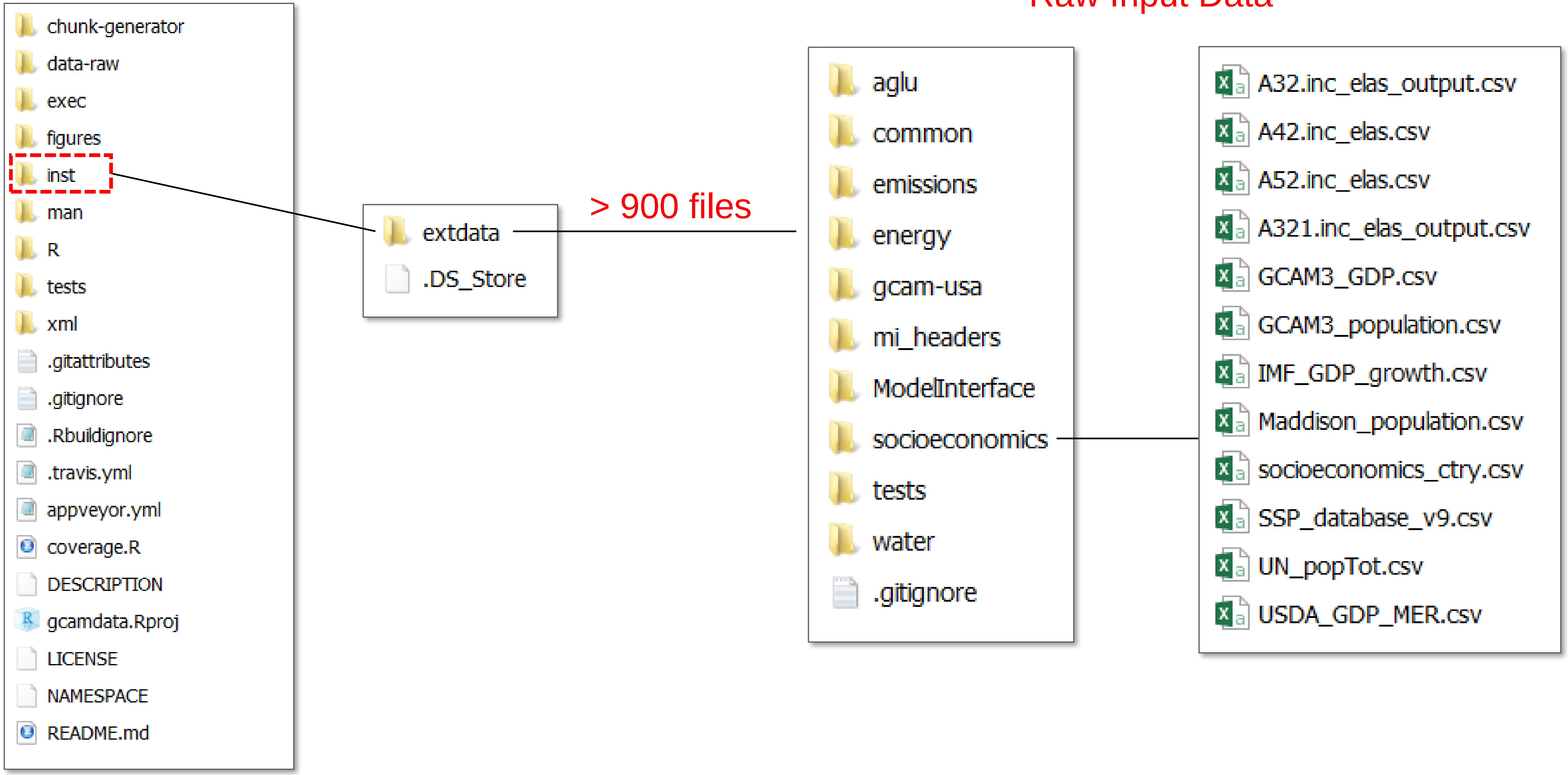
Guides to useful features

XML outputs/ GCAM inputs

R project file

Record of required R packages

File Structure



chunk-generator
data-raw
exec
figures
inst
man
R
tests
xml
.gitattributes
.gitignore
.Rbuildignore
.travis.yml
appveyor.yml
coverage.R
DESCRIPTION
gcamdata.Rproj
LICENSE
NAMESPACE
README.md

extdata
.DS_Store

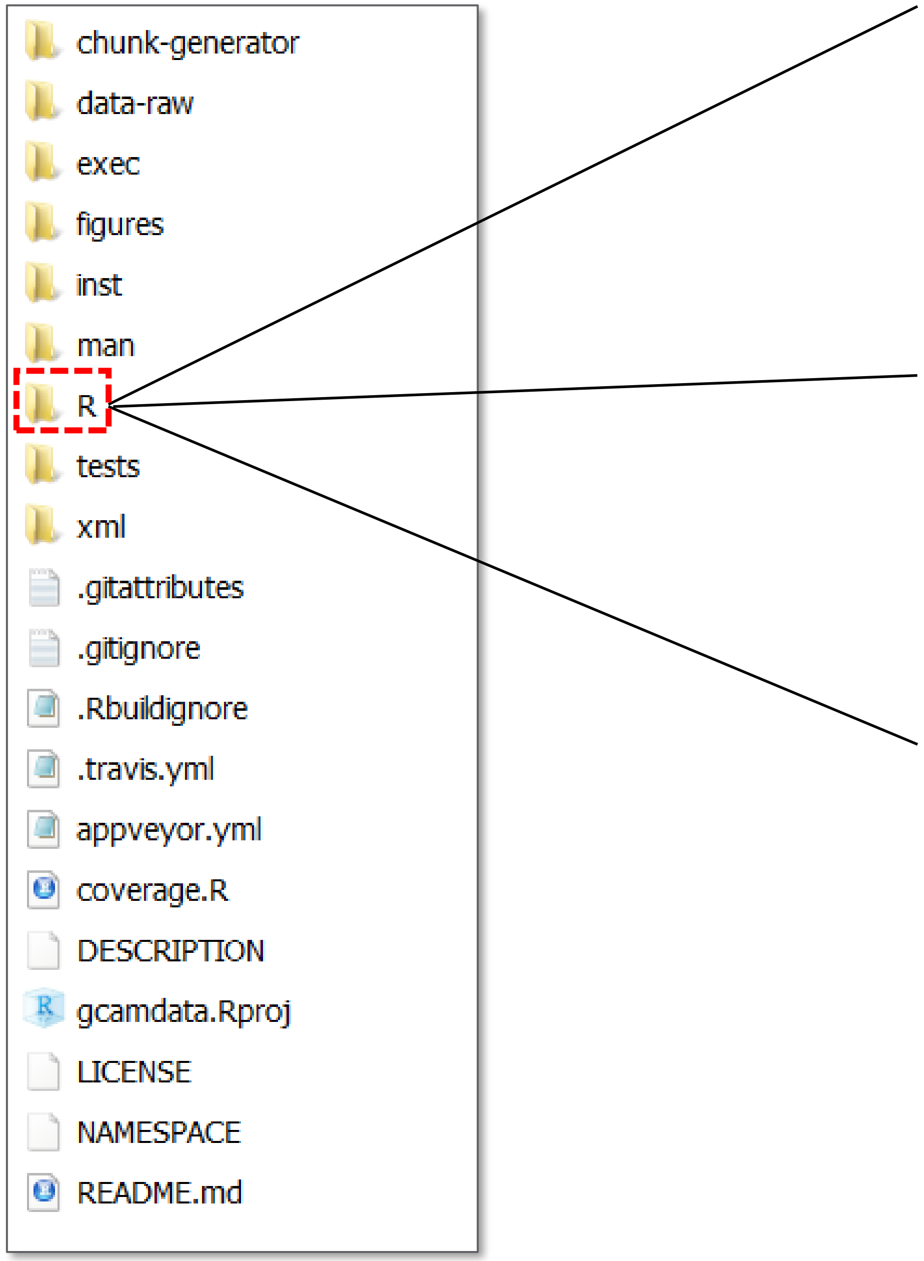
> 900 files

Raw Input Data

aglu
common
emissions
energy
gcam-usa
mi_headers
ModelInterface
socioeconomics
tests
water
.gitignore

A32.inc_elas_output.csv
A42.inc_elas.csv
A52.inc_elas.csv
A321.inc_elas_output.csv
GCAM3_GDP.csv
GCAM3_population.csv
IMF_GDP_growth.csv
Maddison_population.csv
socioeconomics_ctry.csv
SSP_database_v9.csv
UN_popTot.csv
USDA_GDP_MER.csv

Types of R Scripts



chunk-generator
data-raw
exec
figures
inst
man
R
tests
xml
.gitattributes
.gitignore
.Rbuildignore
.travis.yml
appveyor.yml
coverage.R
DESCRIPTION
gcamdata.Rproj
LICENSE
NAMESPACE
README.md

1. **Processing Scripts**
   - Processes the raw input CSVs and outputs of other chunks.

     zsocio_L101.Population.R
     zenergy_L261.Cstorage.R

2. **XML Creation Scripts**
   - Processes data frames from (1) into XMLs

     zwater_xml_electricity_water.R
     zgcamusa_xml_en_prices.R

3. **Other**
   - Contains constants, functions, information needed for the data system to function, etc.

     constants.R
     module_helpers.R

Module Name Structure

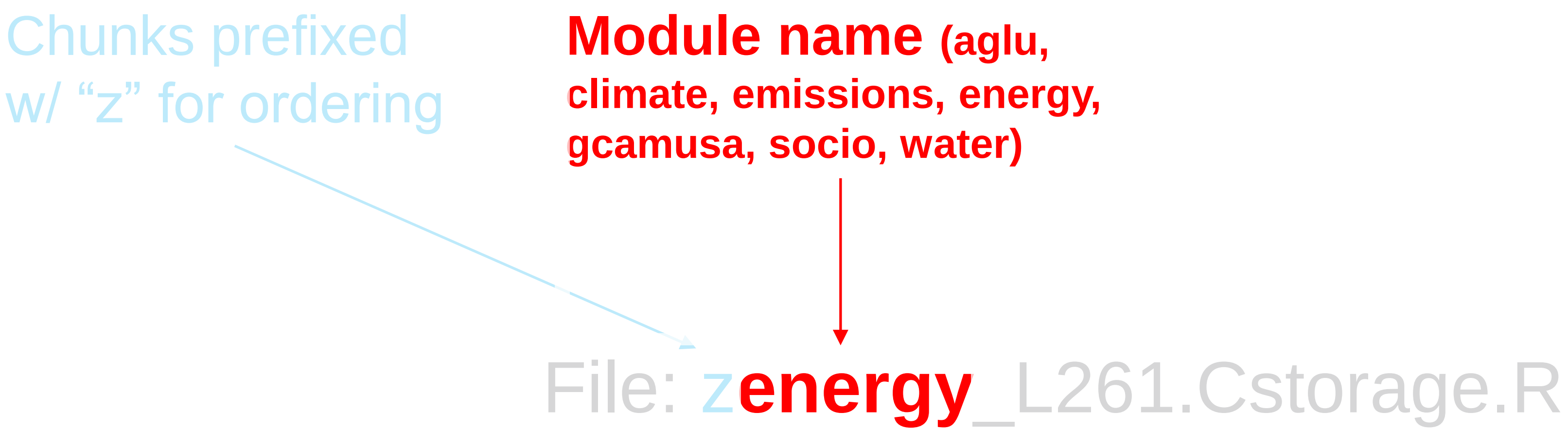
File: zenergy_L261.Cstorage.R

Module Name Structure

Chunks prefixed w/ “z” for ordering

File: **z**energy_L261.Cstorage.R

Module Name Structure

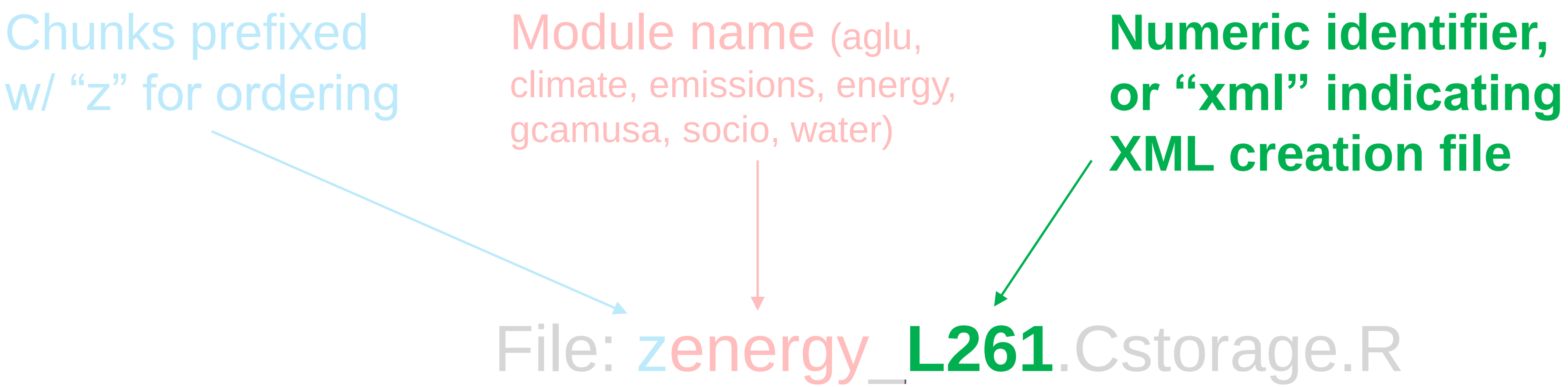


Chunks prefixed w/ “z” for ordering

Module name (aglu, climate, emissions, energy, gcamusa, socio, water)

File: z**energy**_L261.Cstorage.R

Module Name Structure



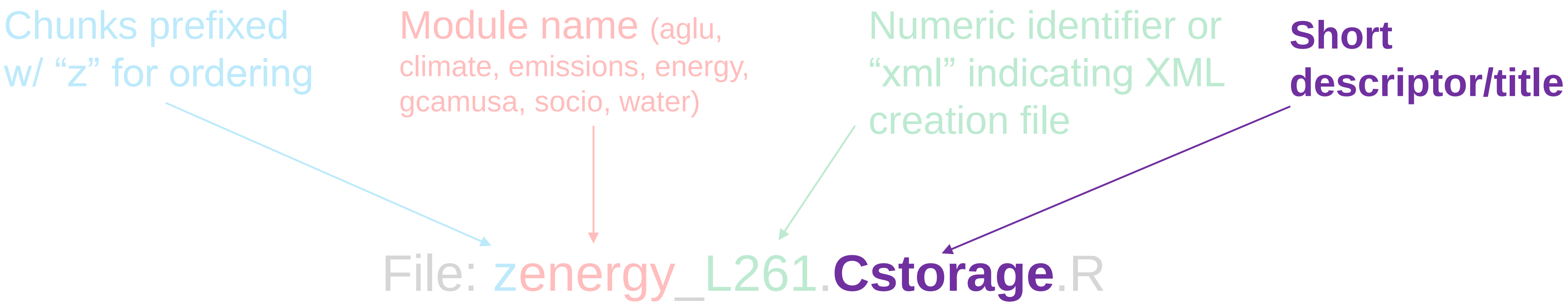
Chunks prefixed w/ “z” for ordering

Module name (aglu, climate, emissions, energy, gcamusa, socio, water)

Numeric identifier, or “xml” indicating XML creation file

File: zenergy_**L261**.Cstorage.R

Module Name Structure



Chunks prefixed w/ “z” for ordering

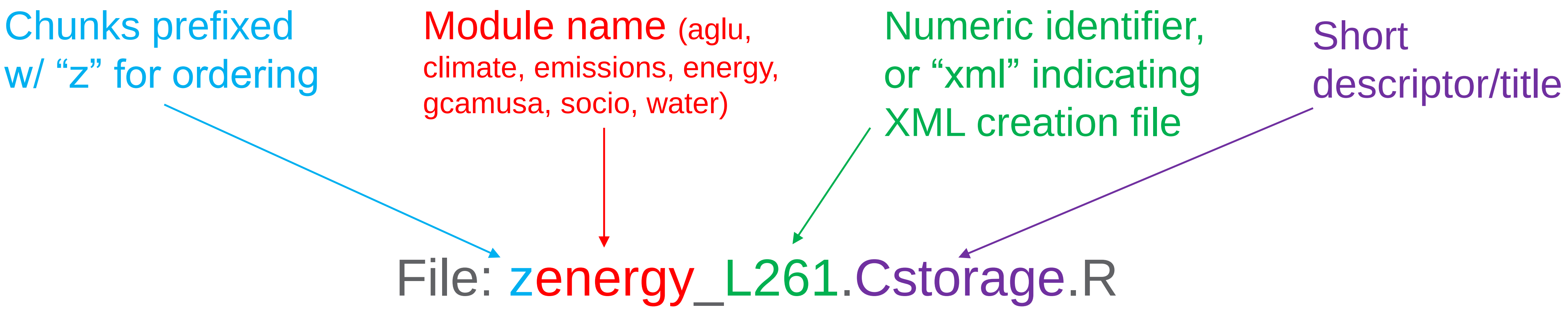
Module name (aglu, climate, emissions, energy, gcamusa, socio, water)

Numeric identifier or “xml” indicating XML creation file

Short descriptor/title

File: zenergy_L261.**Cstorage**.R

Module Name Structure



Chunks prefixed w/ “z” for ordering

Module name (aglu, climate, emissions, energy, gcamusa, socio, water)

Numeric identifier, or “xml” indicating XML creation file

Short descriptor/title

File: zenergy_L261.Cstorage.R

Function: module_energy_L261.Cstorage()

Examples:

- zaglu_xml_ag_trade.R / module_aglu_an_input_xml()
- zgcamusa_L1321.cement / module_gcamusa_L1321.cement()

How to run *gcamdata*

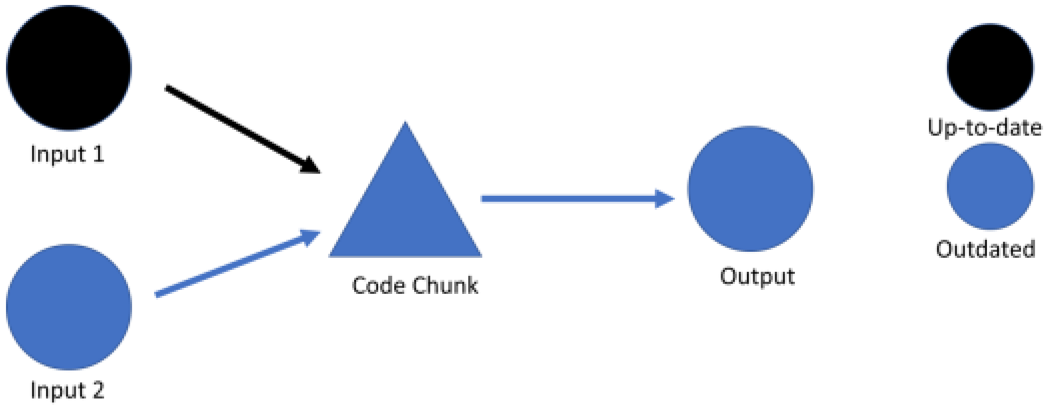
How to run *gcamdata*

- Step 1: Install R and RStudio
  - R: https://cran.r-project.org/
    - R v4.1.0 has been tested and works with GCAM v7.0
  - RStudio: https://posit.co/download/rstudio-desktop/
- Step 2: Download or clone GCAM: https://github.com/JGCRI/gcam-core
- Step 3: Install R packages – use *renv* (details later)
- Step 4: Open *input/gcamdata/gcamdata.Rproj*
- Step 5: Load gcamdata: `devtools::load_all()`
- Step 6: Run the driver: `driver_drake()`

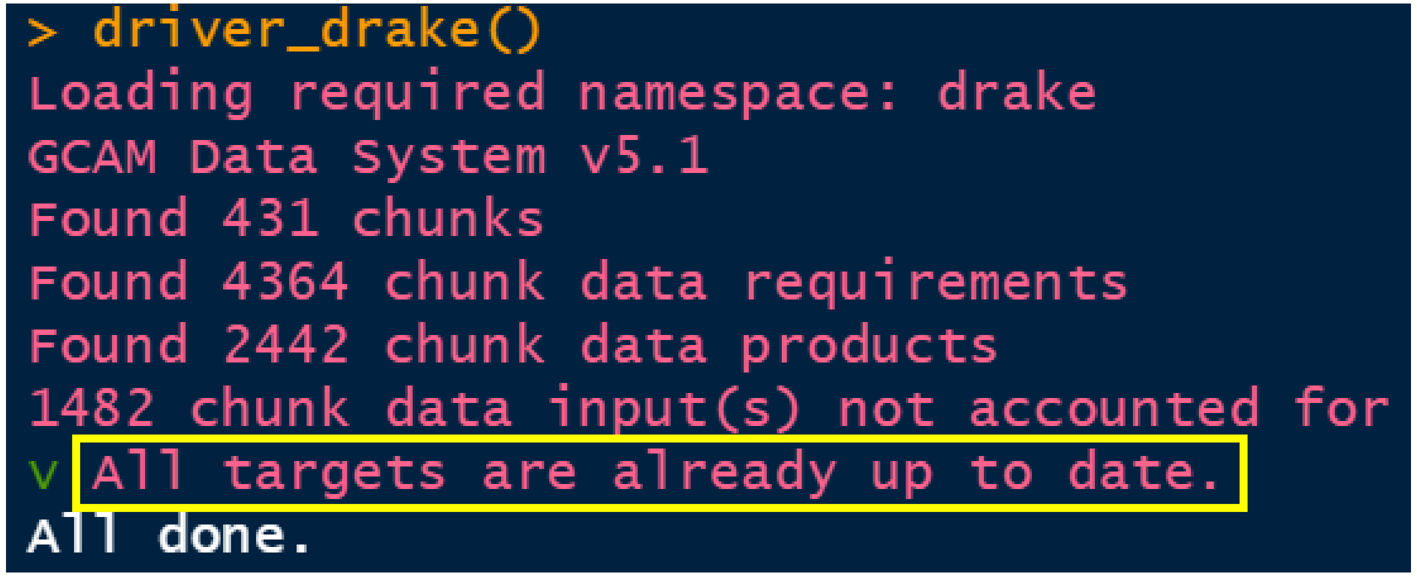
Note: You DO NOT need to do this if you just want to run the release version, since the release version comes with the XMLs pre-built.

Pacific Northwest
NATIONAL LABORATORY

driver_drake()

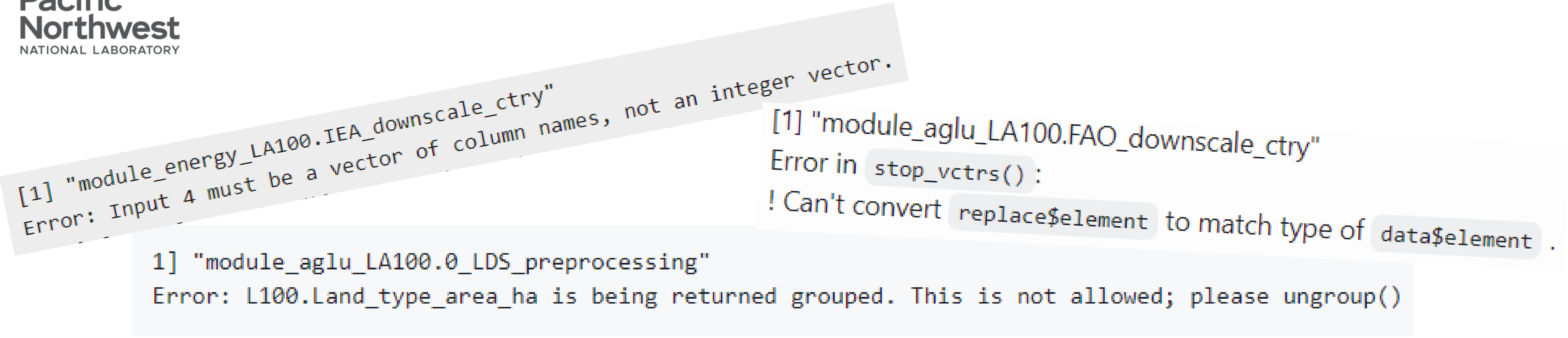


- Function that runs the data system
- Stores data and functions in cache
- Only runs what is out-of-date
- Major timesaver because it prevents repeated building of identical outputs



```
> driver_drake()
Loading required namespace: drake
GCAM Data System v5.1
Found 431 chunks
Found 4364 chunk data requirements
Found 2442 chunk data products
1482 chunk data input(s) not accounted for
v All targets are already up to date.
All done.
```

Common Issue: Package Issues

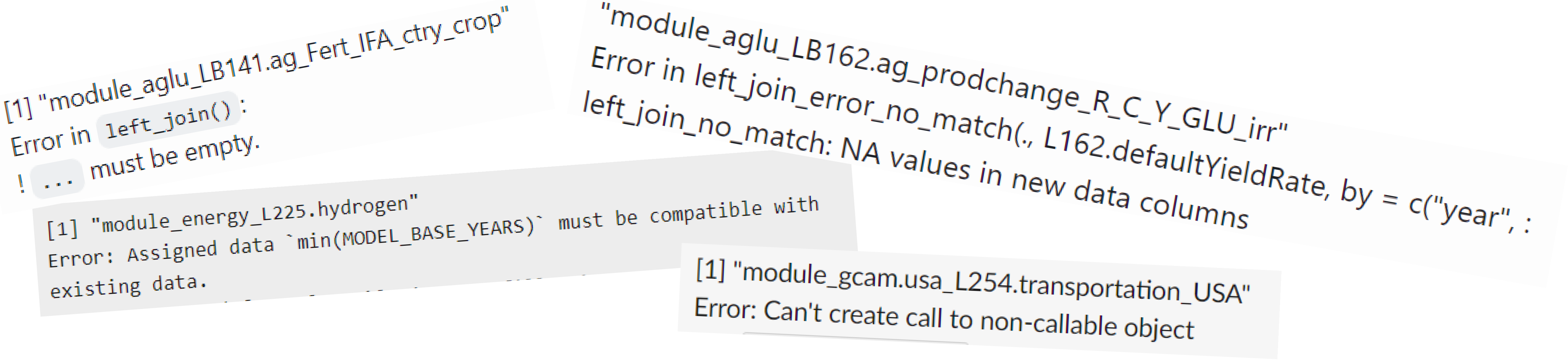


```
[1] "module_energy_LA100.IEA_downscale_ctry"
Error: Input 4 must be a vector of column names, not an integer vector.
```

```
[1] "module_aglu_LA100.FAO_downscale_ctry"
Error in stop_vctrs() :
! Can't convert replace$element to match type of data$element .
```

```
1] "module_aglu_LA100.0_LDS_preprocessing"
Error: L100.Land_type_area_ha is being returned grouped. This is not allowed; please ungroup()
```

Common Issue: Package Issues



```
[1] "module_aglu_LB141.ag_Fert_IFA_ctry_crop"
Error in left_join() :
! ... must be empty.
```

```
"module_aglu_LB162.ag_prodchange_R_C_Y_GLU_irr"
Error in left_join_error_no_match(., L162.defaultYieldRate, by = c("year", :
left_join_no_match: NA values in new data columns
```

```
[1] "module_energy_L225.hydrogen"
Error: Assigned data `min(MODEL_BASE_YEARS)` must be compatible with
existing data.
```

```
[1] "module_gcam.usa_L254.transportation_USA"
Error: Can't create call to non-callable object
```

Solution: *renv* – R package management

- “renv” = <u>R</u>eproducible <u>Env</u>ironment
- Gives each project its own package library
- Key files:
  - Renv directory: holds symbolic links to the package cache
  - Renv lock file: specifies which R packages and versions are used
- Renv automates *R package* version control, not R version control
  - R version specified in lock file, but not enforced
  - *gcamdata* in GCAM 7.0 works with R 4.1.0

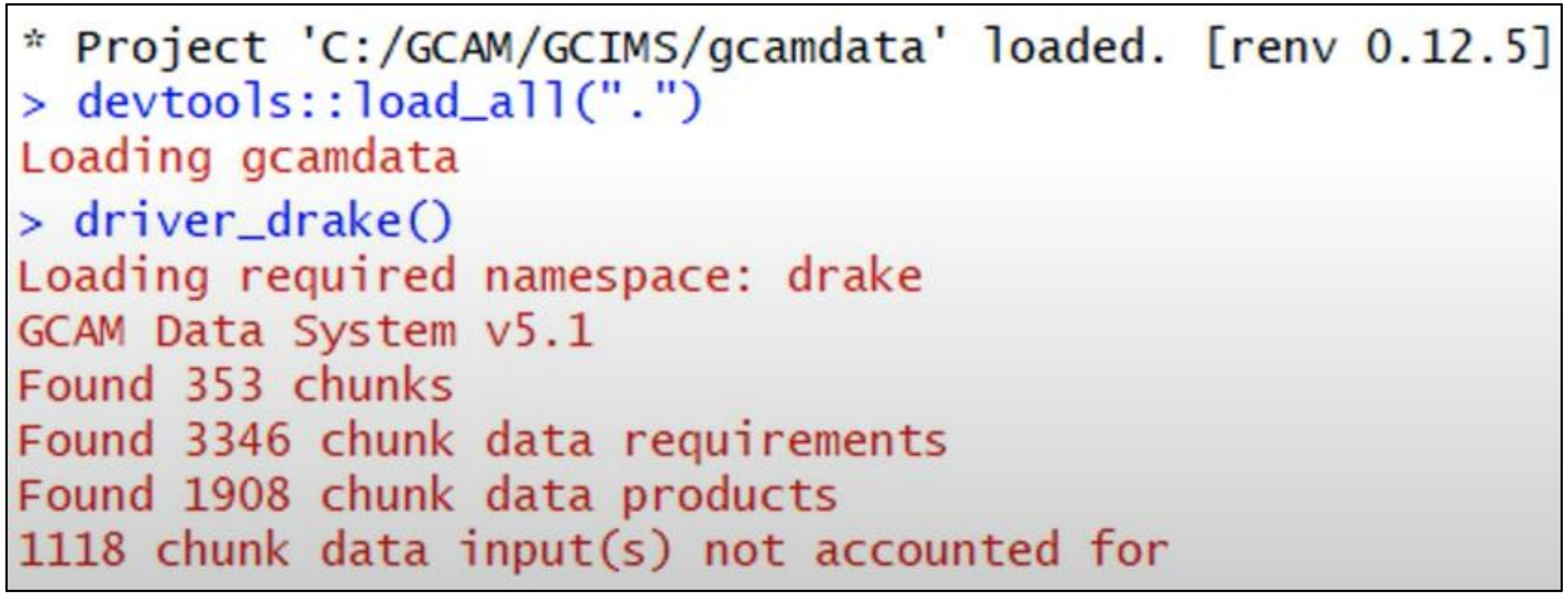
How to active *renv* in gcamdata workspace

- Open gcamdata project file: input/R/gcamdata.Rproj
- Load renv: `library(renv)`
  - If renv hasn’t been installed yet, run `install.packages(“renv”)`
- Initialize the local R library with
  - `renv::activate()` → activates use of renv
  - `renv::restore()` → synchronizes library with lockfile
- Note, this may take awhile on initial set-up

Only have to be executed ONCE per gcamdata workspace

How to active *renv* in gcamdata workspace

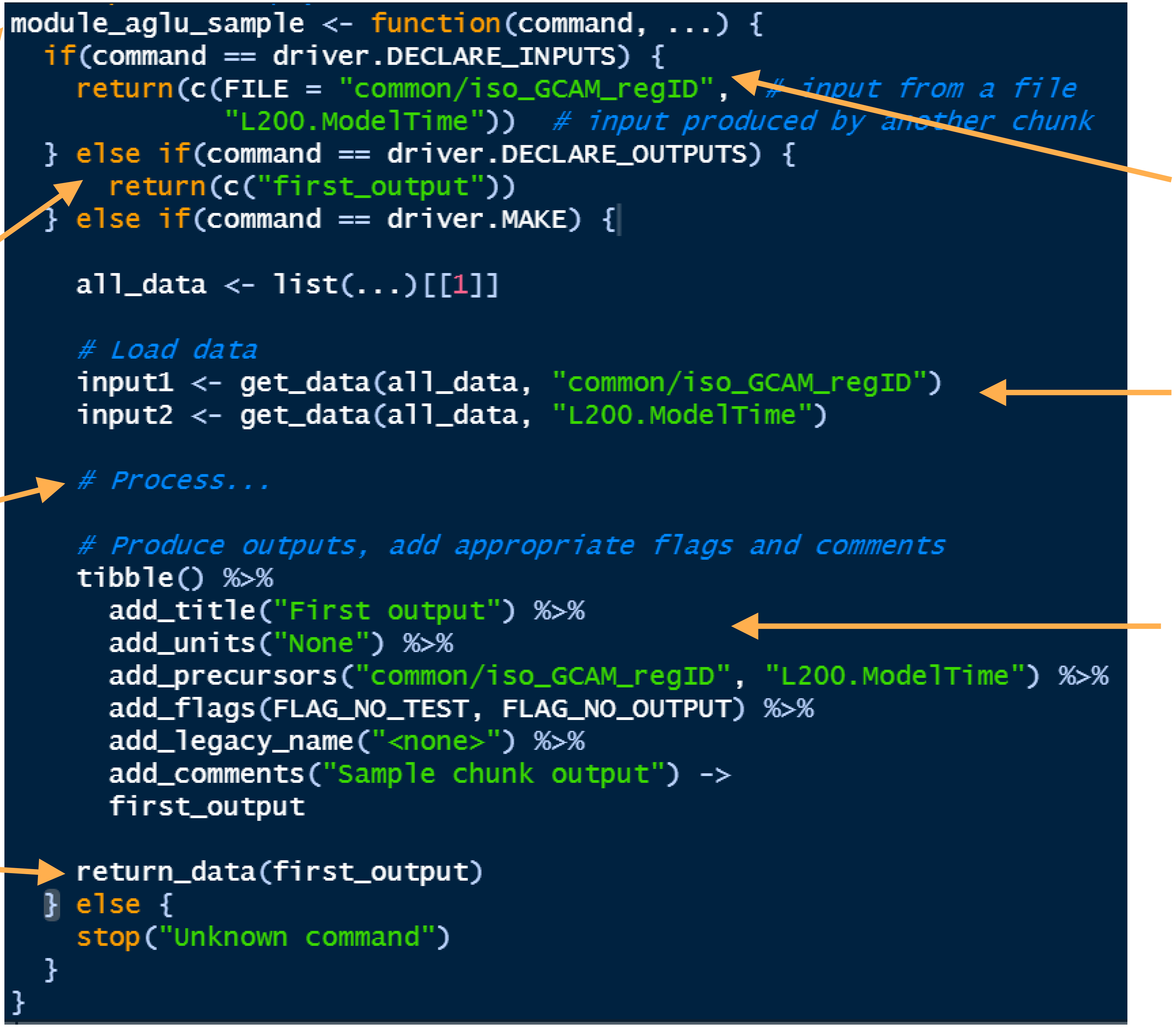
- After initial set-up, a message from renv will be printed to the console when the gcamdata.Rproj is opened
- Proceed to load package and run driver_drake



```
* Project 'C:/GCAM/GCIMS/gcamdata' loaded. [renv 0.12.5]
> devtools::load_all(".")
Loading gcamdata
> driver_drake()
Loading required namespace: drake
GCAM Data System v5.1
Found 353 chunks
Found 3346 chunk data requirements
Found 1908 chunk data products
1118 chunk data input(s) not accounted for
```

Adding to/Modifying the Data System

Anatomy of a *gcamdata* chunk



```
module_aglu_sample <- function(command, ...) {
  if(command == driver.DECLARE_INPUTS) {
    return(c(FILE = "common/iso_GCAM_regID",  # input from a file
             "L200.ModelTime"))  # input produced by another chunk
  } else if(command == driver.DECLARE_OUTPUTS) {
      return(c("first_output"))
  } else if(command == driver.MAKE) {

    all_data <- list(...)[[1]]

    # Load data
    input1 <- get_data(all_data, "common/iso_GCAM_regID")
    input2 <- get_data(all_data, "L200.ModelTime")

    # Process...

    # Produce outputs, add appropriate flags and comments
    tibble() %>%
      add_title("First output") %>%
      add_units("None") %>%
      add_precursors("common/iso_GCAM_regID", "L200.ModelTime") %>%
      add_flags(FLAG_NO_TEST, FLAG_NO_OUTPUT) %>%
      add_legacy_name("<none>") %>%
      add_comments("Sample chunk output") ->
      first_output

    return_data(first_output)
  } else {
    stop("Unknown command")
  }
}
```

Function name includes sector

List of chunk outputs

Process data

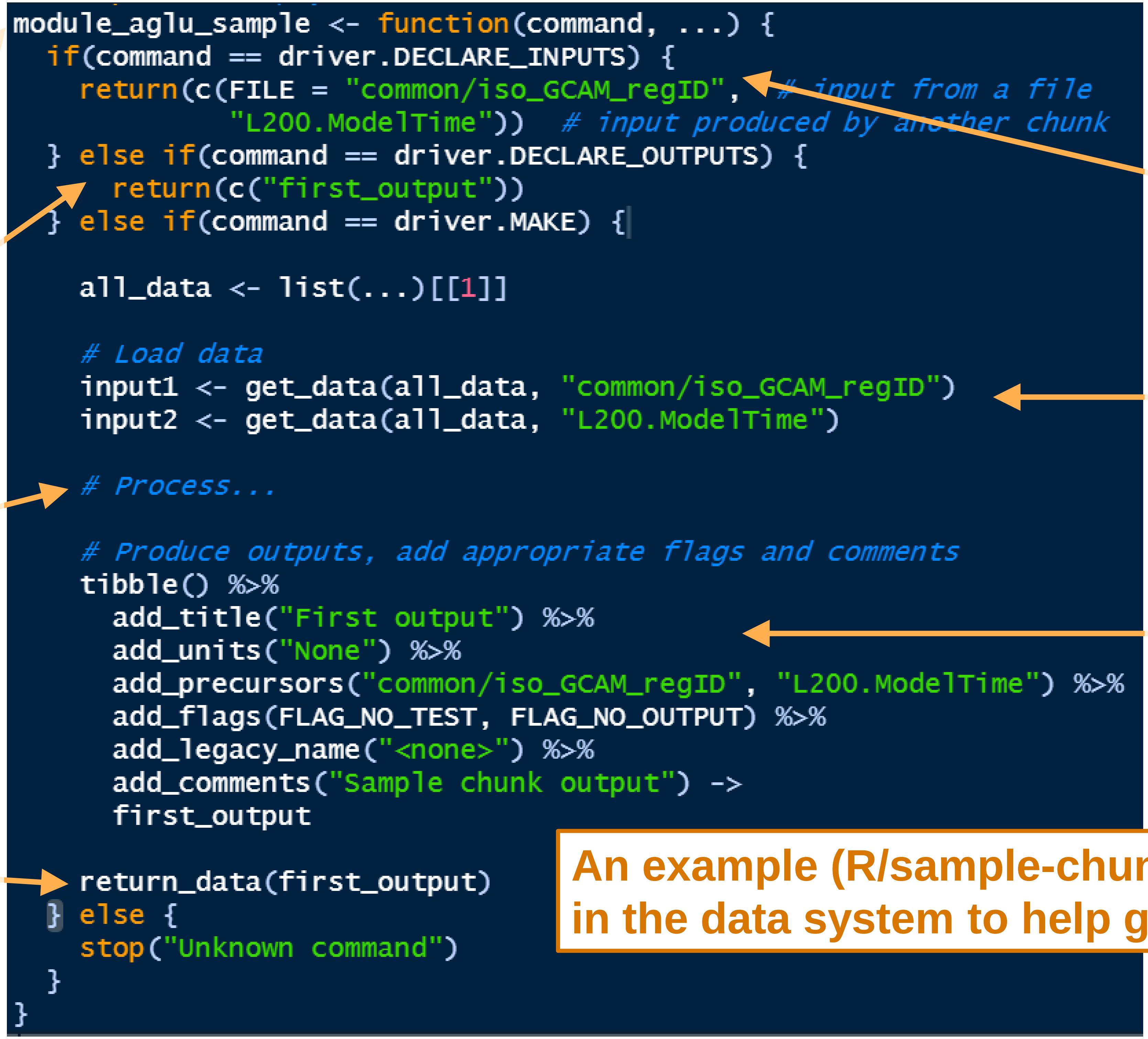
Return data back to driver

List of inputs, usually CSVs or R data frames from other chunks

Load inputs

Produce output and add metadata

Anatomy of a *gcamdata* chunk



```
module_aglu_sample <- function(command, ...) {
  if(command == driver.DECLARE_INPUTS) {
    return(c(FILE = "common/iso_GCAM_regID",  # input from a file
             "L200.ModelTime"))  # input produced by another chunk
  } else if(command == driver.DECLARE_OUTPUTS) {
     return(c("first_output"))
  } else if(command == driver.MAKE) {

    all_data <- list(...)[[1]]

    # Load data
    input1 <- get_data(all_data, "common/iso_GCAM_regID")
    input2 <- get_data(all_data, "L200.ModelTime")

    # Process...

    # Produce outputs, add appropriate flags and comments
    tibble() %>%
      add_title("First output") %>%
      add_units("None") %>%
      add_precursors("common/iso_GCAM_regID", "L200.ModelTime") %>%
      add_flags(FLAG_NO_TEST, FLAG_NO_OUTPUT) %>%
      add_legacy_name("<none>") %>%
      add_comments("Sample chunk output") ->
      first_output

    return_data(first_output)
  } else {
    stop("Unknown command")
  }
}
```

Function name includes sector

List of chunk outputs

Process data

Return data back to driver

List of inputs, usually CSVs or R data frames from other chunks

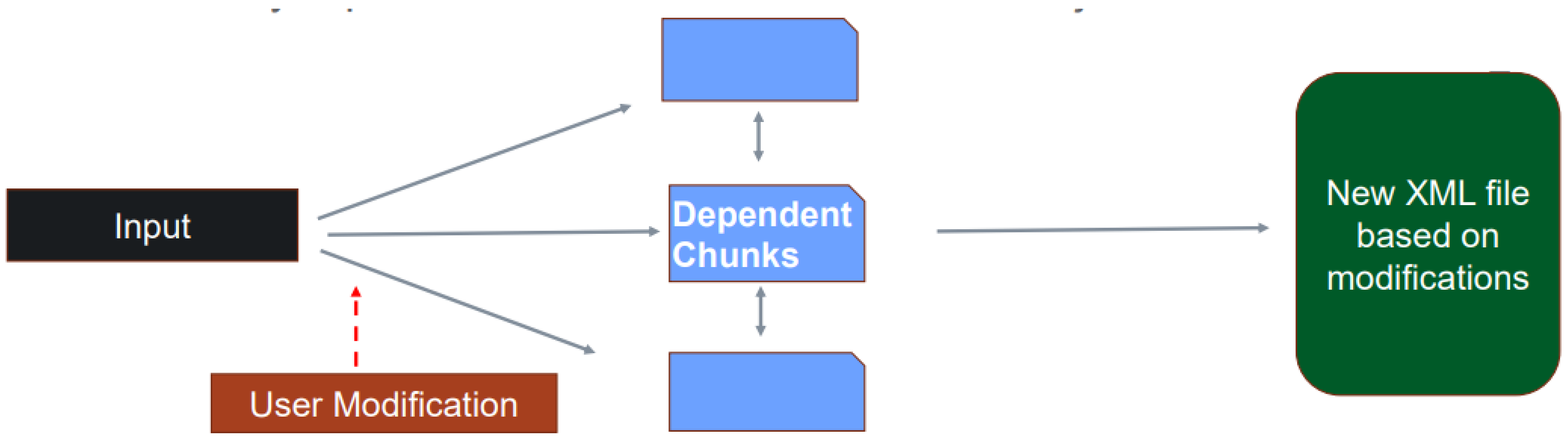
Load inputs

Produce output and add metadata

An example (R/sample-chunk.R) is included in the data system to help get you started

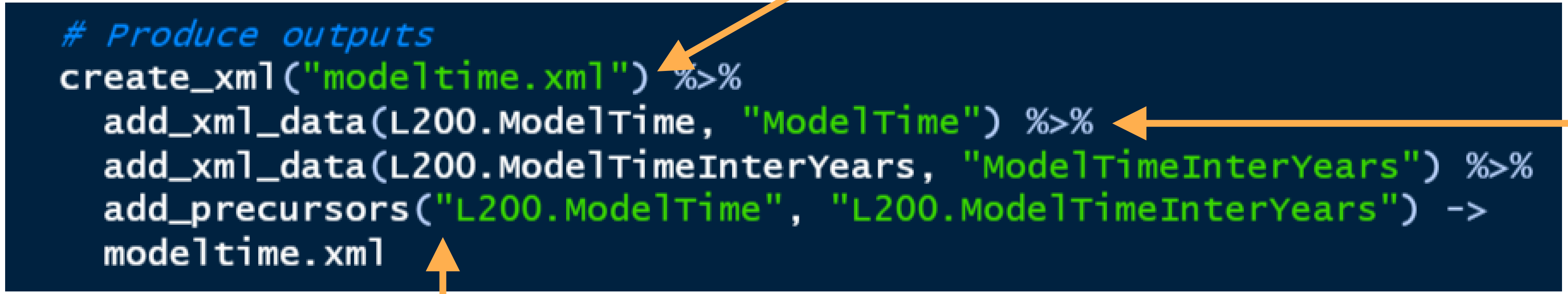
User-Modification Functions (Preview)

- Chunk that can be “plugged” into *gcamdata*
- New chunk can modify any objects that are used or created in *gcamdata* and pass the modified object to all dependent chunks.



Go to “*Creating XMLs and using user-modification functions*” in the next session to learn more!

XML Creation (Preview)



```
# Produce outputs
create_xml("modeltime.xml") %>%
  add_xml_data(L200.ModelTime, "ModelTime") %>%
  add_xml_data(L200.ModelTimeInterYears, "ModelTimeInterYears") %>%
  add_precursors("L200.ModelTime", "L200.ModelTimeInterYears") ->
  modeltime.xml
```

create_xml: Sets up the creation of the XML object

add_xml_data: Tells which data frames to include in the XML

“ModelTime” is a header

- Headers tell the XML how to format the table columns.

add_precursors: All inputs used to create the XML

From ModelInterface_headers.txt (in inst/extdata/mi_headers)

```
ModelTime, modeltime/+{time-step}start-year, modeltime/+start-year,
modeltime/+final-calibration-year, modeltime/+end-year, modeltime/
+carbon-model-start-year, scenario, scenario/modeltime
```

Go to “*Creating XMLs and using user-modification functions*” in the next session to learn more!

Debugging/Useful Functions

Debugging a Chunk

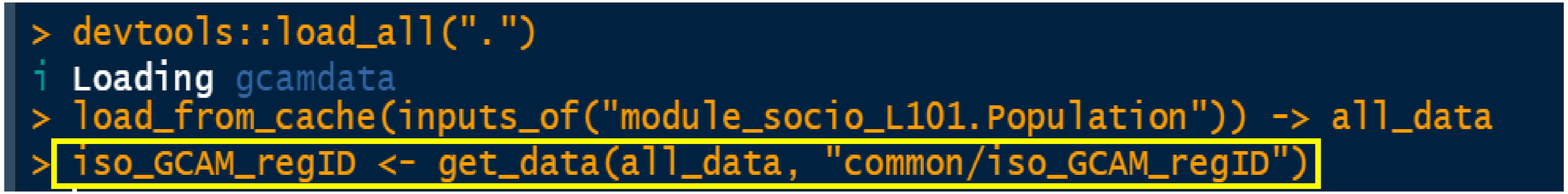
To run a chunk line by line:

```
> devtools::load_all()
> load_from_cache(inputs_of("module_socio_L101.Population")) -> all_data
```

OR

```
> driver_drake(stop_before = " module_socio_L101.Population ") -> all_data
```

- “stop_before” returns chunk inputs
- “stop_after” returns chunk outputs



```
> devtools::load_all(".")
i Loading gcamdata
> load_from_cache(inputs_of("module_socio_L101.Population")) -> all_data
> iso_GCAM_regID <- get_data(all_data, "common/iso_GCAM_regID")
```

Useful Functions: `load_from_cache()`

Loads objects from drake cache ← Only works if you've previously run *driver_drake*

1. Load gcamdata object:
   `load_from_cache("common/GCAM_region_names")`

2. Load all *inputs* from chunk:
   `load_from_cache(inputs_of(“module_energy_L1323.iron_steel”))`

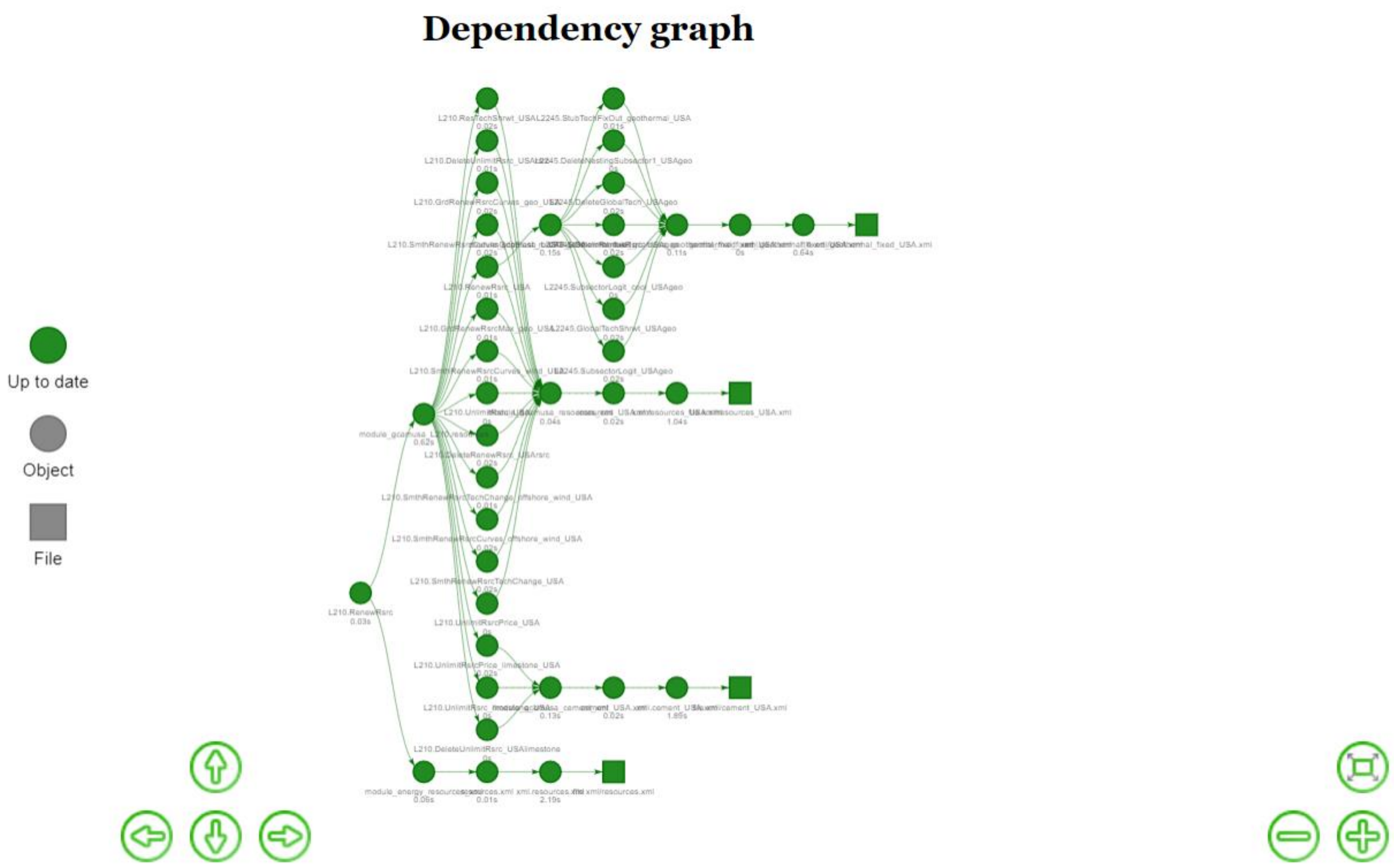
3. Load all *outputs* from chunk:
   `load_from_cache(outputs_of(“module_energy_L1323.iron_steel”))`

Useful Functions: Tracing

vis_drake_graph()

- Visualize targets and dependency relationships
- Need visNetwork package installed
- Get the plan
  - `plan <- driver_drake(return_plan_only = TRUE)`
- Display the dependency graph downstream from L210.RenewRscr
  - `vis_drake_graph(plan, from = make.names("L210.RenewRsrc"))`

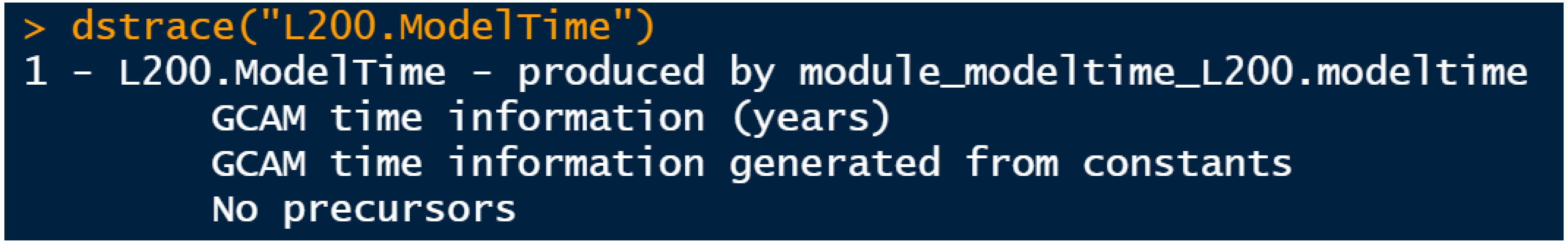
Useful Functions: Tracing


Dependency graph
Up to date
Object
File
L210.ResTechShrwt_USA
0.02s
L2245.StubTechFixOut_geothermal_USA
0.01s
L210.DeleteUnlimitRsrc_USA
0.01s
L2245.DeleteNestingSubsector1_USAgeo
0s
L210.GrdRenewRsrcCurves_geo_USA
0.02s
L2245.DeleteGlobalTech_USAgeo
0.02s
L210.SmthRenewRsrcCurves
0.02s
0.15s
0.02s
0.11s
0s
0.64s
geothermal_fixed_USA.xml
L210.RenewRsrc_USA
0.01s
L2245.SubsectorLogit_cool_USAgeo
0s
L210.GrdRenewRsrcMax_geo_USA
0.01s
L2245.GlobalTechShrwt_USAgeo
0.02s
L210.SmthRenewRsrcCurves_wind_USA
0.01s
L2245.SubsectorLogit_USAgeo
0.02s
L210.UnlimitRsrc_USA
0s
0.04s
0.02s
1.04s
resources_USA.xml
module_gcamusa_L210.resources
0.62s
L210.DeleteRenewRsrc_USArsrc
0.02s
L210.SmthRenewRsrcTechChange_offshore_wind_USA
0.01s
L210.SmthRenewRsrcCurves_offshore_wind_USA
0.02s
L210.SmthRenewRsrcTechChange_USA
0.02s
L210.RenewRsrc
0.03s
L210.UnlimitRsrcPrice_USA
0s
L210.UnlimitRsrcPrice_limestone_USA
0.02s
L210.UnlimitRsrc_limestone_USA
0s
0.13s
0.02s
1.85s
cement_USA.xml
L210.DeleteUnlimitRsrc_USAlimestone
0s
module_energy_resources
0.06s
resources.xml
0.01s
2.19s
resources.xml

Useful Functions: Tracing

dstrace()

- Function to trace data files through the data system
- Tells you what data objects feed into other data objects
- Utilizes GCAM_DATA_MAP, an R data file that stores the information of all input files and R chunks
- `dstrace(`**`object_name, direction = "upstream",`**` graph = FALSE, gcam_data_map = GCAM_DATA_MAP, previous_tracelist = NULL, recurse = TRUE, ...)`



```
> dstrace("L200.ModelTime")
1 - L200.ModelTime - produced by module_modeltime_L200.modeltime
        GCAM time information (years)
        GCAM time information generated from constants
        No precursors
```

Common Issues

1. Package issues – Use renv!

1.
   ```
   Error in left_join_error_no_match(df_left, df_right) :
   left_join_no_match: NA values in new data columns
   ```

ID	X1
1	A1
2	A2

↔

ID	X2
2	B1
3	B2

=

ID	X1	X2
1	A1	NA
2	A2	B1

↑ **FAILS**

2. When running driver_drake …

   ```
   Error in file.rename(tmp, filename) : expanded 'to' name too long
   ```

   1. Windows imposes a maximum file path length that is relatively small
   2. Solution: Shorten path to workspace

Resources

- GitHub Repository: https://github.com/JGCRI/gcamdata
- Wiki: https://github.com/JGCRI/gcamdata/wiki
- Issues? Use GitHub Issues: https://github.com/JGCRI/gcamdata/issues
- Questions/Ideas? Use GitHub Discussions! https://github.com/JGCRI/gcamdata/discussions

Pacific Northwest
NATIONAL LABORATORY

Thank you