

María José Lee & Fabian Metz

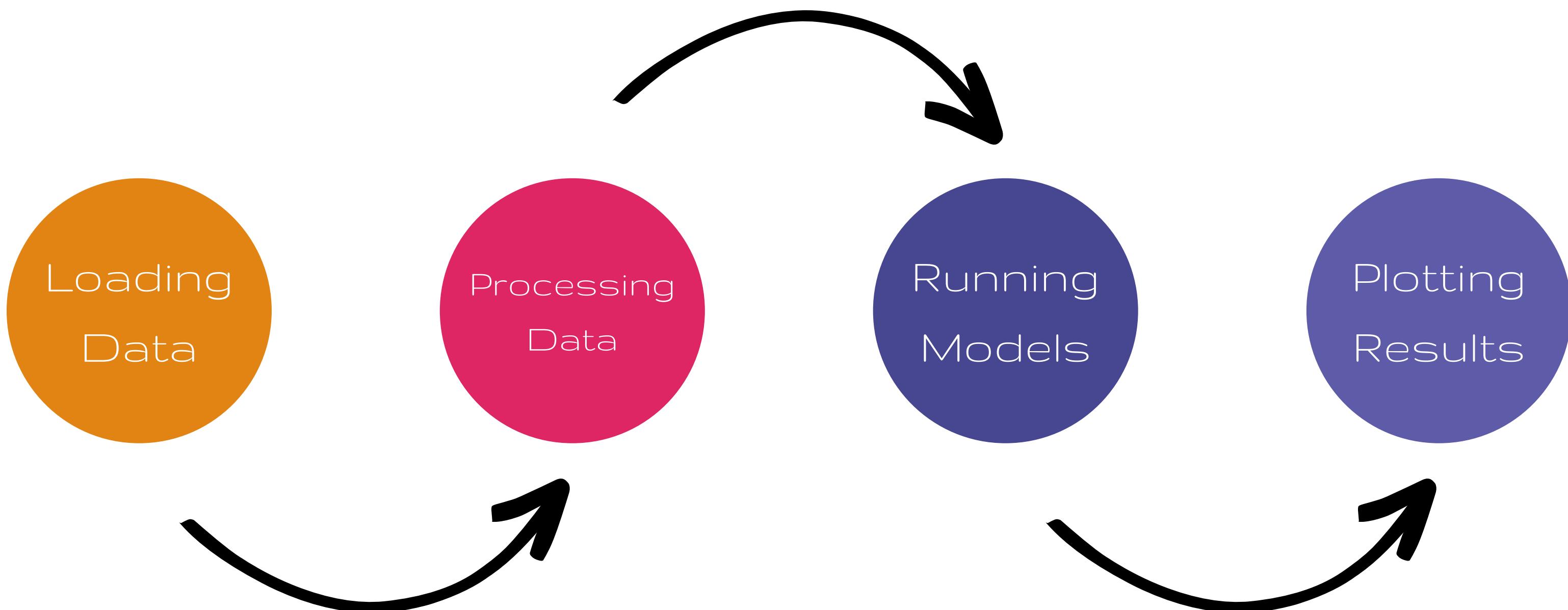
# TARGETS R PACKAGE

Ensuring reproducible and  
trustworthy code



THE BASICS

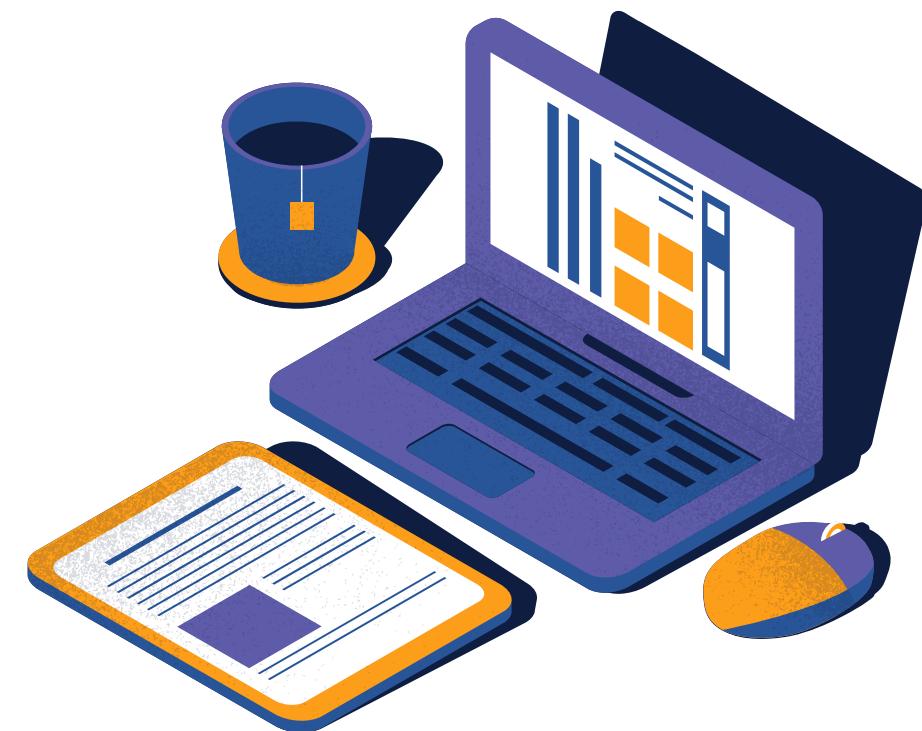
# WHAT IS A TARGET?



# WHY USE TARGETS?

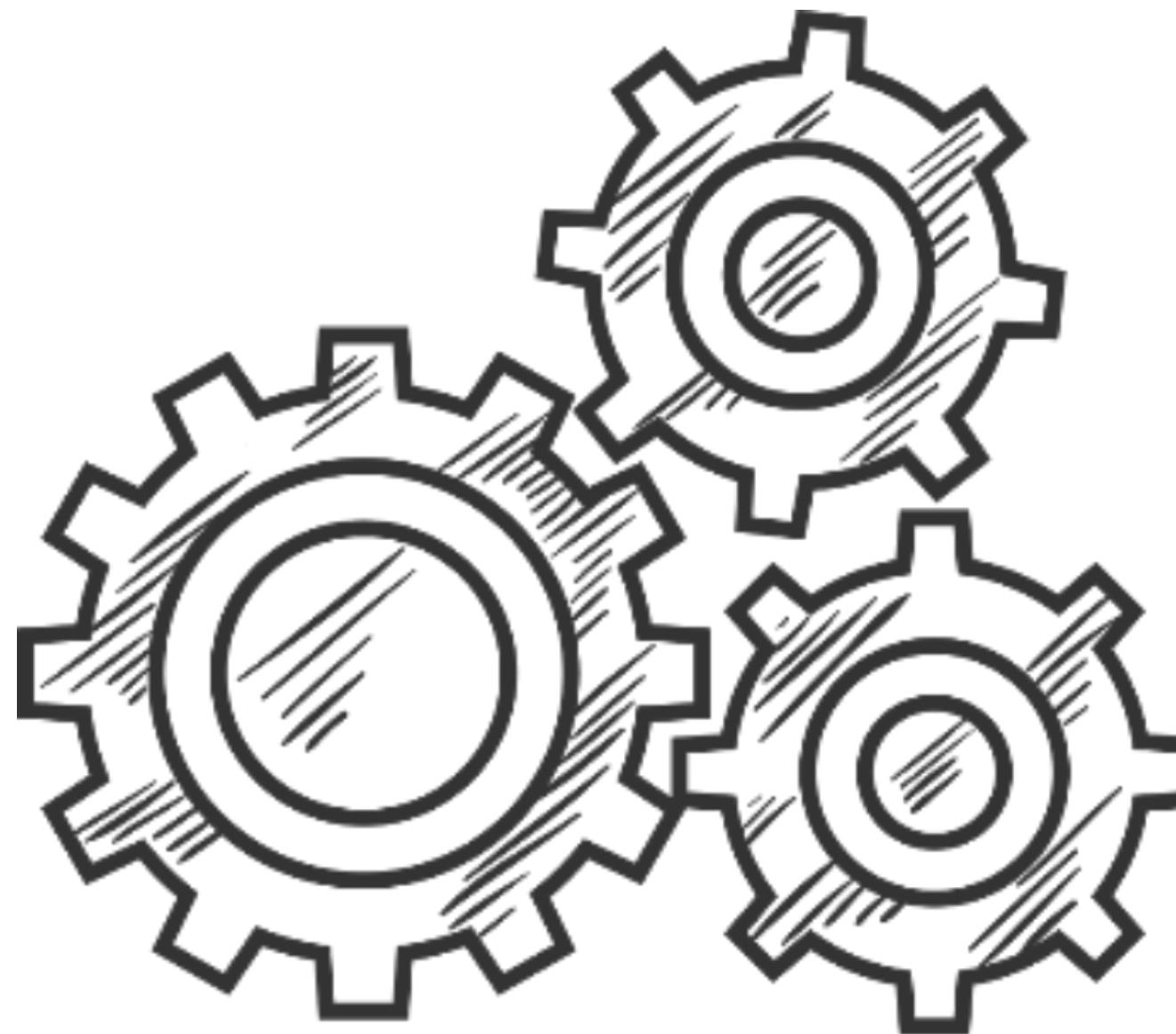
## MOTIVATION

To cut down on unnecessary long runtime. It understands the user's set up, and skips unnecessary steps while running it.



## PHILOSOPHY

Explicitly tailored to R, this is a function oriented package, intended to help users better organize and interact with the tasks (or targets) of their pipeline, so that the end result is efficient, reproducible and trustworthy.



# HOW IT WORKS

Define the steps needed to run your pipeline as functions.

Input the functions to the `–targets.R` file.

Use the Targets functions to run your pipeline.

# functions

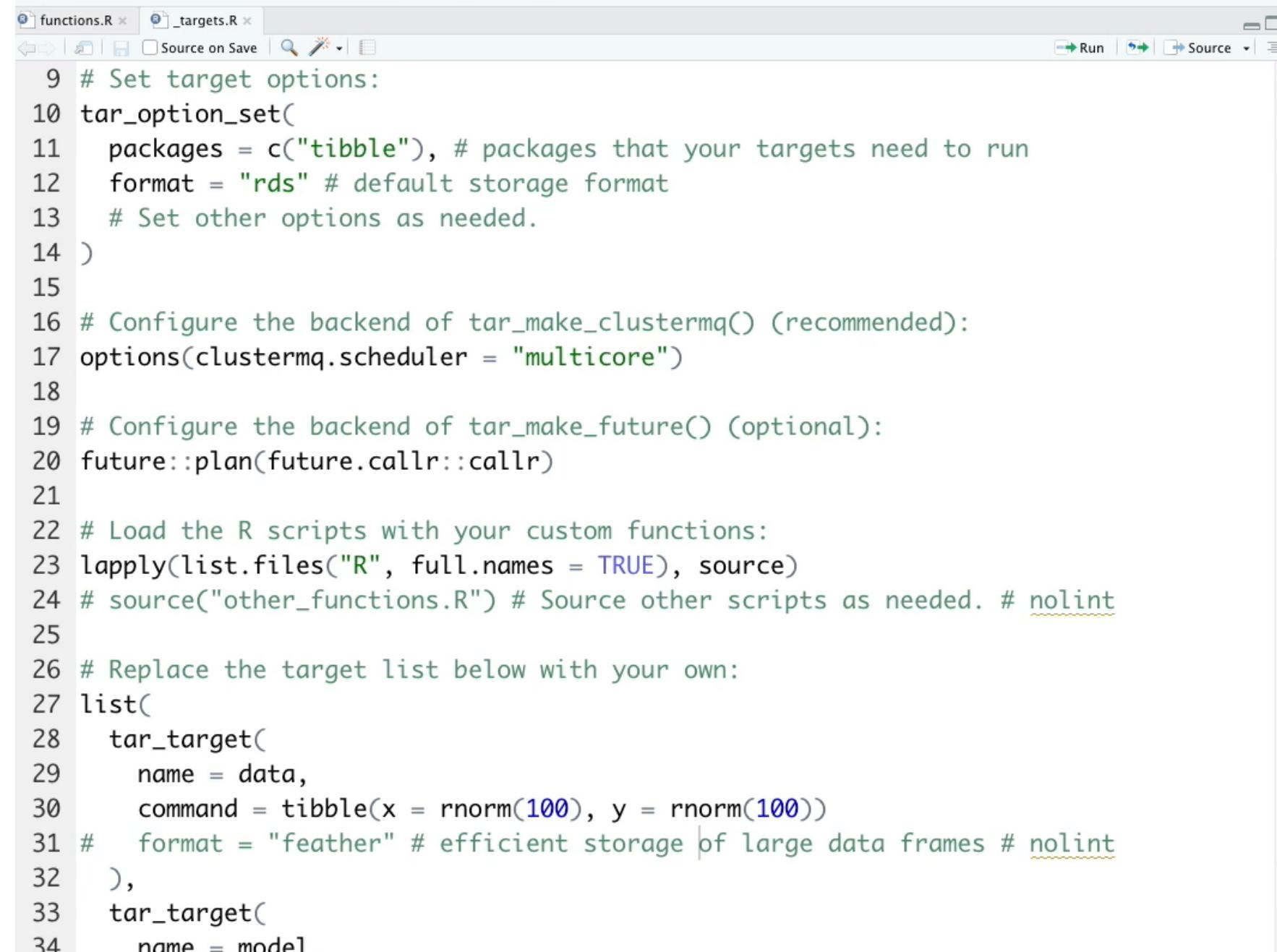
# USE\_TARGETS()

Sets up separate `_targets.R` file in which you:

- Load the needed libraries
- Source the relevant functions for your pipeline to work
- Define a list of your targets

## TAR\_EDIT()

Allows you to access your `_targets.R` file to edit it.



```
functions.R x _targets.R x
Source on Save Run Source

9 # Set target options:
10 tar_option_set(
11   packages = c("tibble"), # packages that your targets need to run
12   format = "rds" # default storage format
13   # Set other options as needed.
14 )
15
16 # Configure the backend of tar_make_clustermq() (recommended):
17 options(clustermq.scheduler = "multicore")
18
19 # Configure the backend of tar_make_future() (optional):
20 future::plan(future.callr::callr)
21
22 # Load the R scripts with your custom functions:
23 lapply(list.files("R", full.names = TRUE), source)
24 # source("other_functions.R") # Source other scripts as needed. # nolint
25
26 # Replace the target list below with your own:
27 list(
28   tar_target(
29     name = data,
30     command = tibble(x = rnorm(100), y = rnorm(100))
31     # format = "feather" # efficient storage of large data frames # nolint
32   ),
33   tar_target(
34     name = model,
```

# TAR\_TARGET()

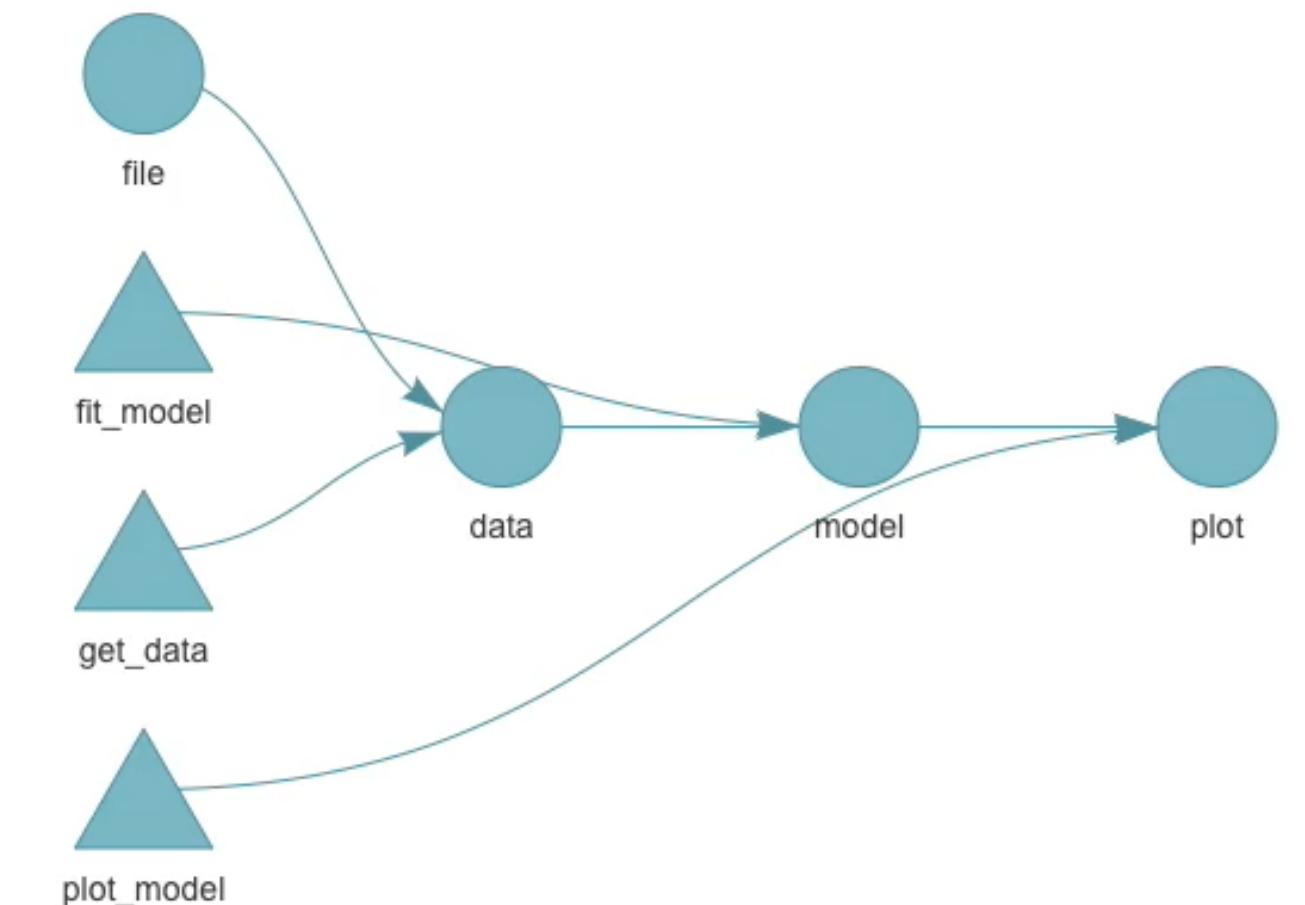
```
list(  
  tar_target(flights_file, "Data/nyc_flights.csv", format = "file"), # loading the data  
  tar_target(flights_data, get_data(flights_file)), # importing it to the R environment  
  tar_target(run_lm, run_lmmodel(flights_data$arr_delay, flights_data$dep_delay,  
    data = flights_data, n_var = 1)), # running a linear model  
  tar_target(metrics, mod_metrics(run_lm)) # calculating the metrics for the linear model  
)
```

Within the `_targets.r` file, this function requires you to supply the name of the function, as well as the input it is supposed to work on.

# TAR\_VISNETWORK()

Gives you a visual representation of the relations within your pipeline, as well as the state of the objects that make it up.

```
```{r, message = FALSE, warning=FALSE, error = FALSE}
tar_visnetwork()
```



```
```{r}
tar_outdated()
```

```
[1] "flights_file" "metrics"      "run_lm"        "flights_data"
```

**or**

```
```{r}
tar_outdated()
```

```
character(0)
```

## TAR\_OUTDATED()

Provides you with a list vector of the outdated targets within your model.

If nothing is outdated, the output will return "character (0)".

# TAR\_MANIFEST()

Lets the user retrieve metadata of your Targets, such as the r command that run when the target is run.

`{r, message = FALSE} tar_manifest() `	
<b>name</b> <chr>	<b>command</b> <chr>
flights_file	"Data/nyc_flights.csv"
flights_data	get_data(flights_file)
run_lm	run_lmmodel(flights_data\$arr_delay, flights_data\$dep_delay, \n data = flights_data, n_var = 1)
metrics	mod_metrics(run_lm)

# TAR\_MAKE()

```
```{r}
tar_make()
```

- start target flights\_file
- built target flights\_file [0.58 seconds]
- start target flights\_data
- built target flights\_data [2.57 seconds]
- start target run\_lm
- built target run\_lm [0.6 seconds]
- start target metrics
- built target metrics [25.13 seconds]
- end pipeline [46.98 seconds]

Runs all outdated targets in the correct order, and saves them as files.

# TAR\_READ()

Reads a targets "return value" from its file. Used to assign its return value to an object.

```
```{r}
linear_model <- tar_read(run_lm)
linear_model
```

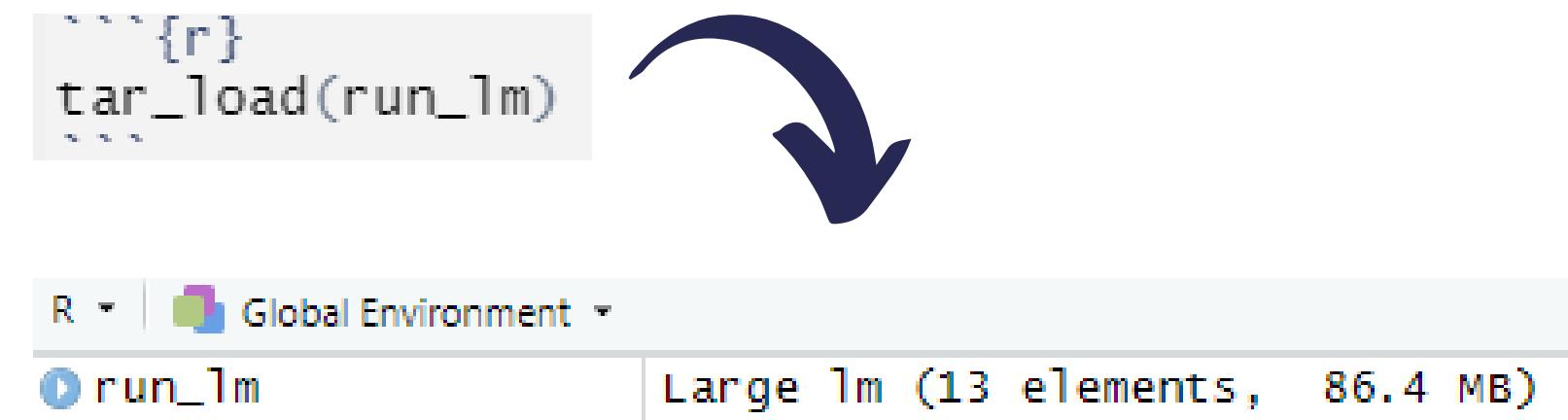
```
call:
lm(formula = y ~ x1, data = data)
Coefficients:
(Intercept)      x1
-5.90          1.02
```



# TAR\_LOAD()

Loads the "return value" of a given target from its storage in –targets/objects to the environment.

```
```{r}
tar_load(run_lm)
```



**THANK YOU  
FOR  
LISTENING!**



# REFERENCES

- <https://docs.ropensci.org/targets/>
- <https://books.ropensci.org/targets/walkthrough.html>
- <https://books.ropensci.org/targets/>
- <https://github.com/wlandau/targets-tutorial>
- <https://www.youtube.com/watch?v=pbC6NX1n01Q&list=PLvgdJdJDL-APJqHy5CXs6m4N7hUVp5rb4>

