

# Reproducible workflows in R

## Infra4NextGen Webinar

Franz Eder

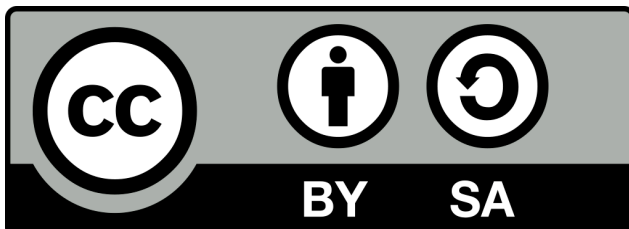
franz.eder@uibk.ac.at

Department of Political Science | University of Innsbruck

# Housekeeping rules

- Please mute your microphone when you are not speaking
- Please turn off your camera during presentations – the session is being recorded
- Post your questions in the chat or raise your hand using the “raise your hand” function in Zoom (“Reactions” button)
- Tell us how we did in the survey (more about that later)
- The presentation and all materials will be shared afterwards (see [GIT](#))

License:



# About me



## Dr. Franz Eder

Assoc. Prof. for International Relations

University of Innsbruck

Research focus: Foreign and Security Policy;  
(Counter-)Terrorism; USA, Europe, Austria; social science  
research methods (v.a. QTA, DNA); academic writing and  
presentation; open and reproducible science

🏠 [www.franz-eder.info](http://www.franz-eder.info) | ✉ [franz.eder@uibk.ac.at](mailto:franz.eder@uibk.ac.at)

📧 [FranzEder](#) | 🦋 [franzeder.bsky.social](#)

🆔 [0000-0003-2625-3631](#) | 📄 [89MVg38AAAAJ](#)

# Program

1 Recap: What is computational reproducibility?

2 Reproducibility iceberg (Rodrigues 2023)

3 Package dependencies with **renv** (Ushey and Wickham 2025)

4 Pipelining with **{targets}** (Landau 2025)

5 Q & A

Franz Johann Eder / I4NG - Reproducible workflows

## I4NG - Reproducible workflows

main I4ng\_reproducible-workflows / + Find file Edit Code

targets v1 Franz Johann Eder authored 7 hours ago 6aa6f472 History

Name	Last commit	Last update
R	targets v1	7 hours ago
_targets	targets v1	7 hours ago
data	CRONOS3 Wave 2 edition 11 data and c...	1 week ago
renv	renv-package installation	1 day ago
.Rprofile	renv-package installation	1 day ago
.gitignore	renv-package installation	1 day ago
README.html	targets v1	7 hours ago
README.md	targets v1	7 hours ago
_targets.R	targets v1	7 hours ago
I4ng_reproducible-workflows.Rproj	RStudio project files	1 week ago
infra4NextGen_reproducible-workflo...	targets v1	7 hours ago
renv.lock	renv-package installation	1 day ago
script_original.R	targets v1	7 hours ago

README.md

### I4NG - Reproducible workflows

#### Abstract

This is the GIT of the webinar *Reproducible workflows in R*. This webinar offers an introduction to efficient, seamless and reproducible data analysis using R.

Project information

- 5 Commits
- 1 Branch
- 0 Tags
- 14.6 MiB Project Storage

README

- + Add LICENSE
- + Add CHANGELOG
- + Add CONTRIBUTING
- + Enable Auto DevOps
- + Add Kubernetes cluster
- + Set up CI/CD
- + Add Wiki
- + Configure Integrations

Created on May 15, 2025

I4NG - Reproducible workflows GIT

# Learning outcomes

After completing this webinar, participants will be able to

- create a reproducible environment for their R projects using the **renv**-package;
- adopt a function-oriented style of programming;
- use the **{targets}**-package as a pipeline tool for their R projects.

# What is computational reproducibility?

## Definition

“the ability of a second researcher to receive a set of files, including data, code, and documentation, and to recreate or recover the outputs of a research project, including figures, tables, and other key quantitative and qualitative results” ([Kitzes 2017, 19](#)).

# Challenges and solutions

- “multiple inconsistent versions of code, data, or both” ([Peikert, Lissa, and Brandmaier 2021, 838](#))
  - solution: Version control ([git](#))
- missing documentation and copy-and-paste errors in final reports
  - solution: dynamic document generation/literate programming ([RStudio Quarto](#))
- software dependencies
  - solution: containerization or [renv-package](#)
- undocumented or ambiguous order of documentation
  - solution: file and folder management; comments; make-files or [targets-package](#)

see [Open Science and Reproducible Research](#)


# Reproducibility iceberg (Rodrigues 2023)


click!



# Package dependencies with **renv**

# What is renv?

renv 1.1.4   Get started   Reference   Articles ▾   Changelog    



## renv

**Overview**

The renv package<sup>1</sup> helps you create reproducible environments for your R projects. Use renv to make your R projects more isolated, portable and reproducible.

- **Isolated:** Installing a new or updated package for one project won't break your other projects, and vice versa. That's because renv gives each project its own private library.
- **Portable:** Easily transport your projects from one computer to another, even across different platforms. renv makes it easy to install the packages your project depends on.
- **Reproducible:** renv records the exact package versions you depend on, and ensures those exact versions are the ones that get installed wherever you go.

**LINKS**

[View on CRAN](#)

[Browse source code](#)

[Report a bug](#)


**LICENSE**


[MIT](#) + file [LICENSE](#)


**CITATION**

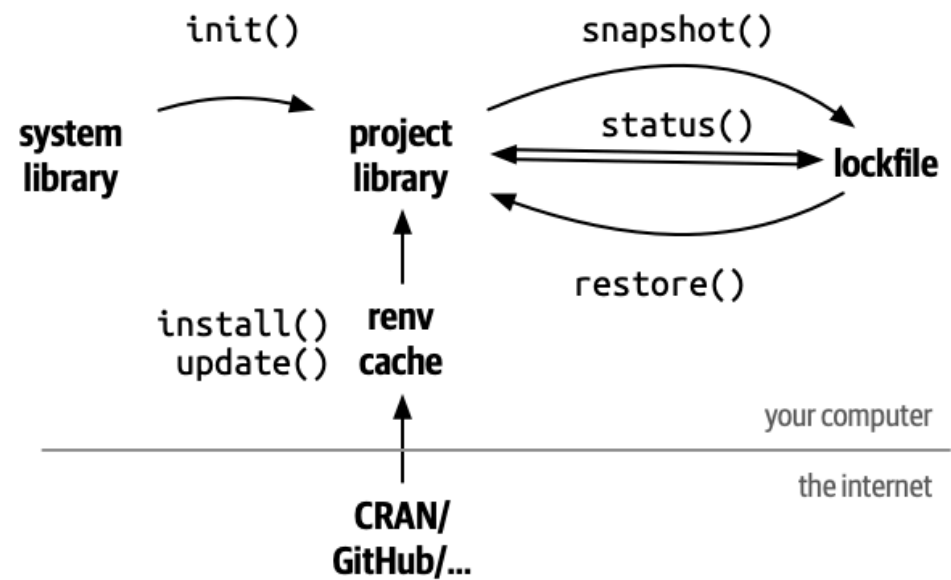
[Citing renv](#)

**DEVELOPERS**

Kevin Ushey  
Author, maintainer 

[Hadley Wickham](#)  
Author 

 **posit**  
Copyright holder, funder



# Install and load

```
1 install.packages("renv")
2
3 # load packages
4 library(renv)
```

```
> install.packages("renv")
Installing package into '/Users/c40298/Library/R/arm64/4.5/library'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/macosx/big-sur-arm64/contrib/4.5/renv_1.1.4.tgz'
Content type 'application/x-gzip' length 2310291 bytes (2.2 MB)
=====
downloaded 2.2 MB

The downloaded binary packages are in
  /var/folders/tv/478_2m316m5gx3jl0j1bq888000gn/T//Rtmp6AipV7/downloaded_packages
> library(renv)

Attaching package: 'renv'

The following objects are masked from 'package:stats':

  embed, update

The following objects are masked from 'package:utils':

  history, upgrade

The following objects are masked from 'package:base':

  autoload, load, remove, use

> |
```

# Initialize

```
1 # initialize renv
2 renv::init()
```

```
> renv::init()
- Linking packages into the project library ... [104/104] Done!
The following package(s) will be updated in the lockfile:

# CRAN -----
- askpass      [* -> 1.2.1]
- backports    [* -> 1.5.0]
- base64enc     [* -> 0.1-3]
- bit          [* -> 4.6.0]
- bit64        [* -> 4.6.0-1]
- blob         [* -> 1.2.4]
- broom        [* -> 1.0.8]
- bslib        [* -> 0.9.0]
- cachem       [* -> 1.1.0]
- callr        [* -> 3.7.6]
- cellranger    [* -> 1.1.0]
- cli          [* -> 3.6.5]
- clipr        [* -> 0.8.0]
- conflicted   [* -> 1.2.0]
- cpp11        [* -> 0.5.2]
- crayon       [* -> 1.5.3]
- curl         [* -> 6.2.2]
- data.table   [* -> 1.17.2]
```

```
The version of R recorded in the lockfile will be updated:
- R          [* -> 4.5.0]

- Lockfile written to "~/Meine Ablage/Forschung/Projekte/Infra4NextGen/i4ng_reproducible-workflows/renv.lock".
Restarting R session...

- Project '~/Meine Ablage/Forschung/Projekte/Infra4NextGen/i4ng_reproducible-workflows' loaded. [renv 1.1.4]
> |
```

# Find packages

```
1 # where are the package libraries?  
2 .libPaths()
```

```
> .libPaths()  
[1] "/Users/c40298/Meine Ablage/Forschung/Projekte/Infra4NextGen/i4ng_reproducible-workflows/renv/library/macos/R-4.5/aarch64-apple-darwin20"  
[2] "/Users/c40298/Library/Caches/org.R-project.R/R/renv/sandbox/macos/R-4.5/aarch64-apple-darwin20/4cd76b74"  
> |
```

```
1 # in which library is the tidyverse package?  
2 find.package("tidyverse")
```

```
> find.package("tidyverse")  
[1] "/Users/c40298/Meine Ablage/Forschung/Projekte/Infra4NextGen/i4ng_reproducible-workflows/renv/library/macos/R-4.5/aarch64-apple-darwin20/tidyverse"  
>
```

# Install, status, snapshot

```
1 install.packages("tidymodels")
2 renv::status()
3 renv::snapshot()
4 find.package("tidymodels")
```

```
> install.packages("tidymodels")
# Downloading packages -----
- Downloading tidymodels from CRAN ...      OK [81.7 Kb in 0.74s]
- Downloading dials from CRAN ...           OK [424.1 Kb in 0.63s]
- Downloading DiceDesign from CRAN ...      OK [322.4 Kb in 0.63s]
- Downloading hardhat from CRAN ...         OK [843.8 Kb in 0.63s]
- Downloading sparsevctrs from CRAN ...     OK [189.5 Kb in 0.63s]
```

```
- yardstick [1.3.2]
These packages will be installed into "~/Meine Ablage/Forschung/Projekte/Infra4NextGen/i4ng_reproducible-workflows/renv/library/macos/R-4.5/aarch64-apple-darwin20".
```

```
Do you want to proceed? [Y/n]: y
```

```
# Installing packages -----
- Installing DiceDesign ...      OK [installed binary and cached]
- Installing sparsevctrs ...     OK [installed binary and cached]
```

```
- Installing tidymodels ...      OK [installed binary and cached]
```

```
Successfully installed 42 packages in 3 seconds.
```

```
> renv::status()
```

```
No issues found -- the project is in a consistent state.
```

```
> renv::snapshot()
```

```
- The lockfile is already up to date.
```

```
> find.package("tidymodels")
```

```
[1] "/Users/c40298/Meine Ablage/Forschung/Projekte/Infra4NextGen/i4ng_reproducible-workflows/renv/library/macos/R-4.5/aarch64-apple-darwin20/tidymodels"
> |
```

# Update and restore

```
1 # update packages (ATTENTION: run/check code afterwards!)
2 renv::update()
3
4 # download and install all the packages needed for collaborators
5 renv::restore()
```

## .gitignore

```
library/
local/
cellar/
lock/
python/
sandbox/
staging/
```

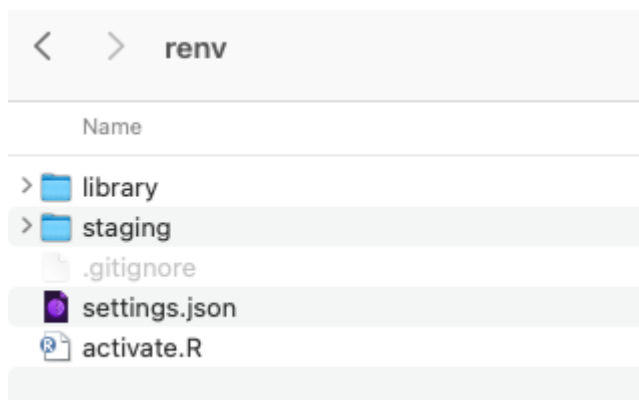
# Re-open RProject

```
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.
```

```
- Project '~/Meine Ablage/Forschung/Projekte/Infra4NextGen/i4ng_reproducible-workflows' loaded. [renv 1.1.4]  
> |
```

## .Rprofile

```
source("renv/activate.R")
```





# Pipelining with **{targets}**

# What is {targets}?

The {targets} R  
package user manual




## 1 Introduction

- How to use {targets} ▾
- 2 A walkthrough to get started
- 3 Help
- 4 Debugging pipelines
- Concepts and best practices ▾
- 5 Functions
- 6 Targets
- 7 Packages
- 8 Projects
- 9 Pseudo-random numbers
- Data and files ▾
- 10 Local data
- 11 Cloud storage
- 12 Literate programming
- Heavy workloads ▾
- 13 Distributed computing
- 14 Performance

## The {targets} R package user manual

AUTHOR  
Will Landau

### 1 Introduction

Pipeline tools coordinate the pieces of computationally demanding analysis projects. The [targets](#) package is a [Make](#)-like pipeline tool for statistics and data science in R. The package skips costly runtime for tasks that are already up to date, orchestrates the necessary computation with implicit parallel computing, and abstracts files as R objects. If all the current output matches the current upstream code and data, then the whole pipeline is up to date, and the results are more trustworthy than otherwise.

#### 1.1 Motivation

Data analysis can be slow. A round of scientific computation can take several minutes, hours, or even days to complete. After it finishes, if you update your code or data, your hard-earned results may no longer be valid. Unchecked, this invalidation creates chronic [Sisyphian](#) loop:

1. Launch the code.
2. Wait while it runs.
3. Discover an issue.
4. Restart from scratch.

#### Table of contents

- 1 Introduction
- 1.1 Motivation
- 1.2 Pipeline tools
- 1.3 The targets package
- 1.4 About this manual
- 1.5 What about drake?

[Edit this page](#)

## Introduction to the **targets** R package



Will Landau

# When (not) to use **{targets}**

- pipeline tool for statistics and data science in R (see GNU Make)
- to coordinate the pieces of computationally demanding analysis projects
- when steps of analyses have to be repeated multiple times
- to make these analyses reproducible
- **DO NOT HAVE TO USE IT FOR** simple analyses

# Function-oriented style of programming

- **{targets}** expects users to adopt a function-oriented style of programming (see Functions)
- functions should be organized according to three key steps of an analysis
  - data generation
  - data analysis
  - reporting results
- save functions in subfolder **R** in the Rproject directory

# Original script

```
1 ###
2 # 00. configuration----
3 ###
4
5 # install packages if needed
6 install.packages("tidyverse")    # collection of packages for data science
7
8 # load packages
9 library(tidyverse)
10
11 ###
12 # 10. loading data----
13 ###
14
15 # read Cronos3, wave 2 dataset from data-folder
16 df <- read_csv("data/CRON3W2e01.1.csv")
17
18 # view head of the dataframe
19 head(df)
20
21 # select variables:
22 ## w2gq1: How worried about climate change
23 ## w2gq6: Humans meant to rule over nature
24
25 df <- df %>% select(w2gq1, w2gq6, agegroup35) %>%
```

# Step 1: getting the data

From...

```

1  ###
2  # 10. loading data----
3  ###
4
5  # read Cronos3, wave 2 dataset from data-folder
6  df <- read_csv("data/CRON3W2e01.1.csv")
7
8  # select variables:
9  ## w2gq1: How worried about climate change
10 ## w2gq6: Humans meant to rule over nature
11
12 df <- df |> select(w2gq1, w2gq6, agegroup35) |>
13   rename("climate" = w2gq1, "rule" = w2gq6) |>
14   filter(climate != 9) |> # remove "No answer"
15   filter(rule != 9) |> # remove "No answer"
16   filter(agegroup35 != 9) |>
17   mutate(rule = case_when( # recode answers i
18     rule == 1 ~ 5, # Agree strongly becomes
19     rule == 2 ~ 4,
20     rule == 3 ~ 3,
21     rule == 4 ~ 2,
22     rule == 5 ~ 1) # Disagree strongly beco
23   ) |>
24   mutate(agegroup35 = as_factor(agegroup35))
25   mutate(agegroup35 = fct_recode(agegroup35

```

...to 01\_data.R

```

1  get_data <- function(file) {
2    read_csv(file) |>
3    select(w2gq1, w2gq6, agegroup35) |>
4    rename("climate" = w2gq1, "rule" = w2gq6) |>
5    filter(climate != 9) |> # remove "No answer"
6    filter(rule != 9) |> # remove "No answer"
7    filter(agegroup35 != 9) |>
8    mutate(rule = case_when( # recode answers i
9      rule == 1 ~ 5, # Agree strongly becomes
10     rule == 2 ~ 4,
11     rule == 3 ~ 3,
12     rule == 4 ~ 2,
13     rule == 5 ~ 1) # Disagree strongly beco
14   ) |>
15   mutate(agegroup35 = as_factor(agegroup35))
16   mutate(agegroup35 = fct_recode(agegroup35,
17     "35 and older" = 9)
18 }

```

## Step 2: analyzing the data

From...

```
1 fit_model <- lm(climate ~ rule, df)
```

...to `02_analyze.R`

```
1 fit_model <- function(data) {  
2   lm(climate ~ rule, data)  
3 }
```

# Step 3: reporting results

From...

```
1 # plotting the model
2 ggplot(df, aes(x = rule, y = climate)) +
3   geom_jitter(color = "#66B32F", alpha = 0.3)
4   geom_abline(intercept = fit_model$coefficie
5               color = "#E72E6B", linetype = 2)
6   labs(x = "Humans meant to rule over nature"
7   labs(y = "How worried about climate change"
8   theme_minimal()
9
10 # violin plot of agegroup vs "How worried about
11 ggplot(df, aes(agegroup35, climate)) +
12   geom_violin() +
13   geom_jitter(color = "#2A4B9B", alpha = 0.3)
14   labs(x = "") +
15   labs(y = "How worried about climate change"
16   theme_minimal()
```

...to 03\_plot.R

```
1 plot_model <- function(data, model) {
2   ggplot(data, aes(x = rule, y = climate)) +
3     geom_jitter(color = "#66B32F", alpha = 0.3)
4     geom_abline(intercept = model$coefficie
5                 color = "#E72E6B", linetype = 2)
6     labs(x = "Humans meant to rule over nature"
7     labs(y = "How worried about climate change"
8     theme_minimal()
9   }
10
11 plot_violin <- function(data) {
12   ggplot(data, aes(agegroup35, climate)) +
13     geom_violin() +
14     geom_jitter(color = "#2A4B9B", alpha = 0.3)
15     labs(x = "") +
16     labs(y = "How worried about climate change"
17     theme_minimal()
18   }
```



# How to use {targets}

## Step 1: Install packages

```
1 install.packages("targets")
2 install.packages("tarchetypes")
```

```
> install.packages("tarchetypes")
# Downloading packages -----
- Downloading tarchetypes from CRAN ...      OK [219.1 Kb in 1.2s]
Successfully downloaded 1 package in 2.2 seconds.

The following package(s) will be installed:
- tarchetypes [0.13.1]
These packages will be installed into "~/Meine Ablage/Forschung/Projekte/Infra4NextGen/i4ng_reproducible-workflows/renv/library/macos/R-4.5/aarch64-apple-darwin20".

Do you want to proceed? [Y/n]: y

# Installing packages -----
- Installing tarchetypes ...      OK [built from source and cached in 2.5s]
Successfully installed 1 package in 2.5 seconds.
>
```

## Step 2: use\_targets()

```
1 library(targets)
2 use_targets()
```

```
> library(targets)
> use_targets()

✓ Wrote _targets.R

i The package "usethis" is required.
✗ Would you like to install it?

1: Yes
2: No

Selection: 1

Installing package into '/Users/c40298/Meine Ablage/Forschung/Projekte/Infra4NextGen/i4ng_reproducible-workflows/renv/library/macos/R-4.5/aarch64-apple-darwin20'
(as 'lib' is unspecified)
also installing the dependencies 'credentials', 'zip', 'gitcreds', 'httr2', 'ini', 'desc', 'gert', 'gh', 'rprojroot', 'whisker'

trying URL 'https://packagemanager.posit.co/cran/latest/bin/macosx/big-sur-arm64/contrib/4.5/credentials_2.0.2.tgz'

trying URL 'https://packagemanager.posit.co/cran/latest/bin/macosx/big-sur-arm64/contrib/4.5/usethis_3.1.0.tgz'

The downloaded binary packages are in
  /var/folders/tv/478_2m316m5gx3jl0j1bq8880000gn/T//Rtmp6NLwR6/downloaded_packages

□ Modify _targets.R.

> |
```

## Step 3: modify **\_targets.R**

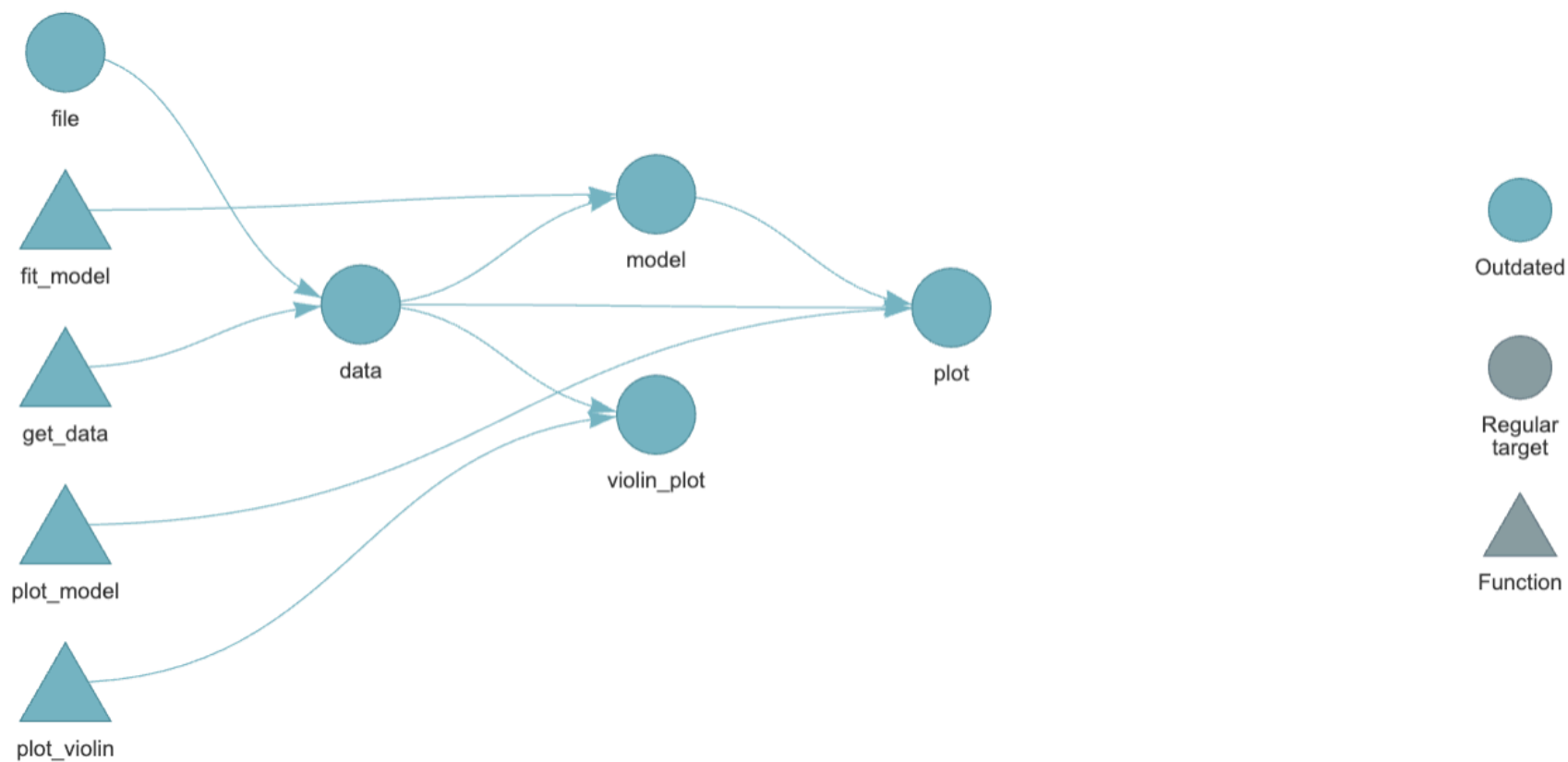
```
_targets.R x
Source on Save

1 # Created by use_targets().
2 # Follow the comments below to fill in this target script.
3 # Then follow the manual to check and run the pipeline:
4 # https://books.ropensci.org/targets/walkthrough.html#inspect-the-pipeline
5
6 # Load packages required to define the pipeline:
7 library(targets)
8 # library(tarchetypes) # Load other packages as needed.
9
10 # Set target options:
11 tar_option_set(
12   packages = c("tibble") # Packages that your targets need for their tasks.
13   # format = "qs", # Optionally set the default storage format. qs is fast.
14   #
15   # Pipelines that take a long time to run may benefit from
16   # optional distributed computing. To use this capability
17   # in tar_make(), supply a {crew} controller
18   # as discussed at https://books.ropensci.org/targets/crew.html.
19   # Choose a controller that suits your needs. For example, the following
20   # sets a controller that scales up to a maximum of two workers
21   # which run as local R processes. Each worker launches when there is work
22   # to do and exits if 60 seconds pass with no tasks to run.
23   #
24   # controller = crew::crew_controller_local(workers = 2, seconds_idle = 60)
25   #
26   # Alternatively, if you want workers to run on a high-performance computing
27   # cluster, select a controller from the {crew.cluster} package.
```

```
1 # Load packages required to define the pipeline:
2 library(targets)
3 library(tarchetypes)
4
5 # Set target options:
6 tar_option_set(
7   packages = c("tidyverse") # Packages that your targets need for their tasks.
8 )
9
10 # Run the R scripts in the R/ folder with your custom functions:
11 tar_source()
12
13 # Replace the target list below with your own:
14 list(
15   tar_target(file, "data/CRON3W2e01.1.csv", format = "file"),
16   tar_target(data, get_data(file)),
17   tar_target(model, fit_model(data)),
18   tar_target(plot, plot_model(model, data)),
19   tar_target(violin_plot, plot_violin(data))
20 )
```

# Step 4: Inspect the pipeline

```
1 tar_visnetwork()
```



# Step 5: Run the pipeline

```
1 tar_make()
```

```
> tar_make()
- The project is out-of-sync -- use `renv::status()` for details.
+ data dispatched
Rows: 10442 Columns: 175
— Column specification —
Delimiter: ","
chr      (7): w2name, w2proddate, cntry, w2survlng, w2lastansw, w2uagent, region
dbl      (160): w2edition, idno, age, agegroup35, w2weight, w2mode, essround, w2...
dtm      (8): w2inwds, w2ginws, w2ginwe, w2einwe, w2sinwe, w2hinwe, w2dinwe, w...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
✓ data completed [217ms, 13.49 kB]
+ violin_plot dispatched
✓ violin_plot completed [9ms, 162.81 kB]
+ model dispatched
✓ model completed [5ms, 24.72 kB]
+ plot dispatched
✓ plot completed [4ms, 178.59 kB]
✓ ended pipeline [689ms, 4 completed, 1 skipped]
> |
```

# Step 6: Read results

```
1 tar_read(model)
```

```
> tar_read(model)

Call:
lm(formula = climate ~ rule, data = data)

Residuals:
    Min       1Q   Median       3Q      Max
-2.64467 -0.41419  0.04676  0.58581  2.27724

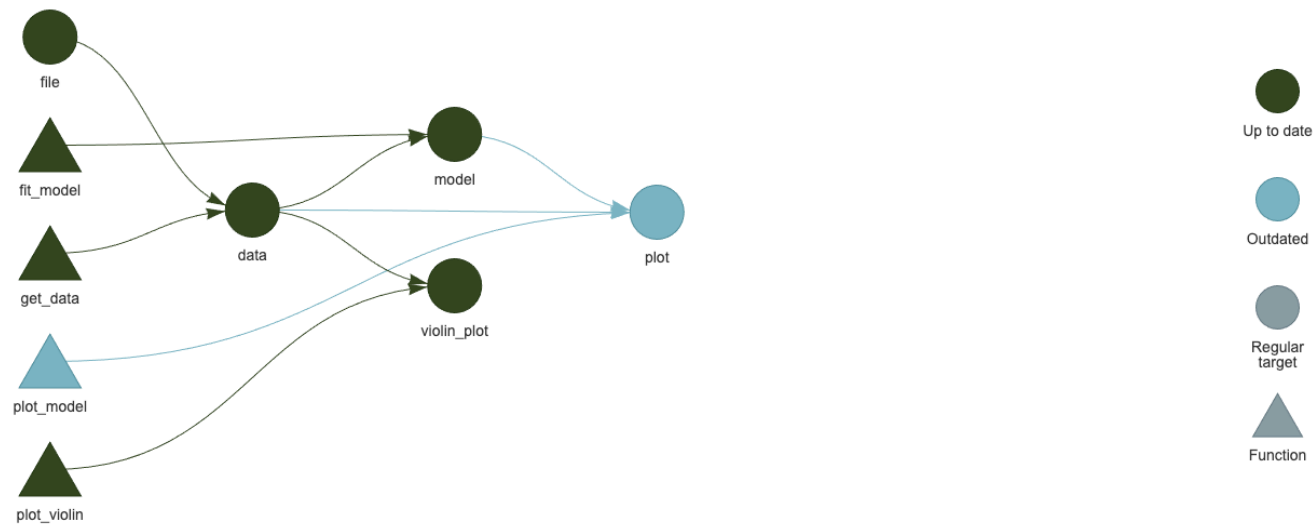
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.875143   0.021118   183.5  <2e-16 ***
rule        -0.230476   0.008695   -26.5  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9092 on 10272 degrees of freedom
Multiple R-squared:  0.06401,    Adjusted R-squared:  0.06392
F-statistic: 702.5 on 1 and 10272 DF,  p-value: < 2.2e-16
```

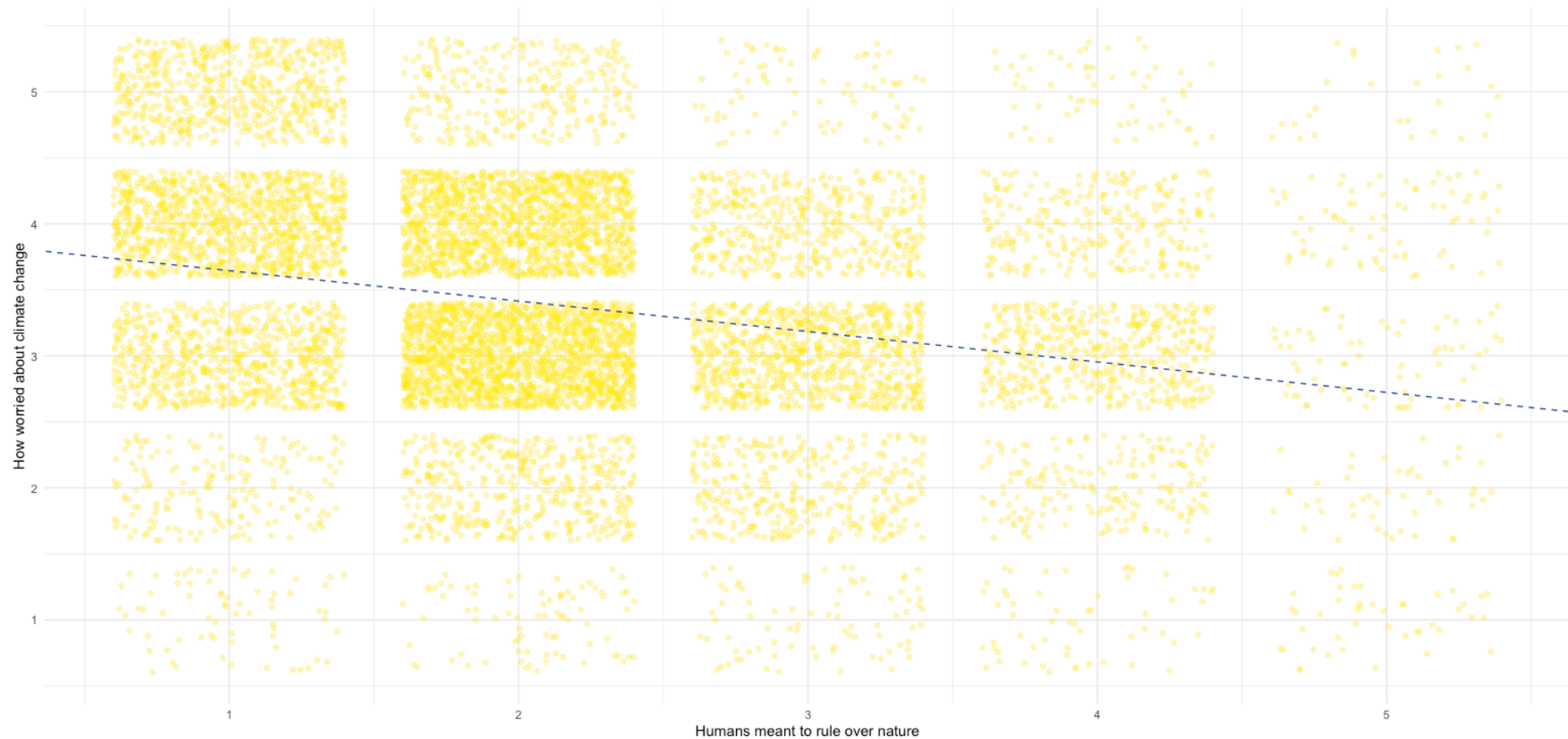
# Change functions?

```
1 # change function in 03_plot.R
2
3 plot_model <- function(data, model) {
4   ggplot(data, aes(x = rule, y = climate)) +
5     #geom_jitter(color = "#66B32F", alpha = 0.3) +
6     geom_jitter(color = "#ffed00", alpha = 0.3) +
7     geom_abline(intercept = model$coefficients[1],
8                 slope = model$coefficients[2],
9                 #color = "#E72E6B", linetype = 2) +
10                color = "#2A4B9B", linetype = 2) +
11     labs(x = "Humans meant to rule over nature") +
12     labs(y = "How worried about climate change") +
13     theme_minimal()
14 }
```





```
> tar_make()  
+ plot dispatched  
✓ plot completed [29ms, 178.66 kB]  
✓ ended pipeline [1.2s, 1 completed, 4 skipped]  
> tar_read(plot)
```



# {targets} and Quarto

## Step 1: create Quarto document

```
---  
title: "Report"  
author: "Franz Eder"  
format: html  
---  
  
We can also use {targets} in Quarto by just reading the R objects into the document with tar_read().
```

Switch markdown editing mode (⇧⌘F4)

Here, for example, is the summary of the model:

```
```{r}  
#| label: "quarto-model"  
  
library(targets)  
tar_read(model)  
```
```

And now we plot the model

```
```{r}  
#| label: "quarto-plot"  
  
tar_read(plot)  
```
```

## Step 2: modify **\_targets.R**

```
1 # Load packages required to define the pipeline:
2 library(targets)
3 library(tarchetypes)
4
5 # Set target options:
6 tar_option_set(
7   packages = c("tidyverse", "tibble") # Packages that your targets need for their tasks.
8 )
9
10 # Run the R scripts in the R/ folder with your custom functions:
11 tar_source()
12
13 # Replace the target list below with your own:
14 list(
15   tar_target(file, "data/CRON3W2e01.1.csv", format = "file"),
16   tar_target(data, get_data(file)),
17   tar_target(model, fit_model(data)),
18   tar_target(plot, plot_model(data, model)),
19   tar_target(violin_plot, plot_violin(data)),
20   tar_quarto(report, "report.qmd")
21 )
```

## Step 3: tar\_make()

```
> tar_make()
+ report dispatched
✓ report completed [15.7s, 3.62 MB]
✓ ended pipeline [17.1s, 1 completed, 5 skipped]
> |
```

### Report

AUTHOR  
Franz Eder

We can also use `{targets}` in Quarto by just reading the R objects into the document with `tar_read()`.

Here, for example, is the summary of the model:

```
library(targets)
tar_read(model)
```

Call:

```
lm(formula = climate ~ rule, data = data)
```

Residuals:

|  | Min      | 1Q       | Median  | 3Q      | Max     |
|--|----------|----------|---------|---------|---------|
|  | -2.64467 | -0.41419 | 0.04676 | 0.58581 | 2.27724 |

Coefficients:

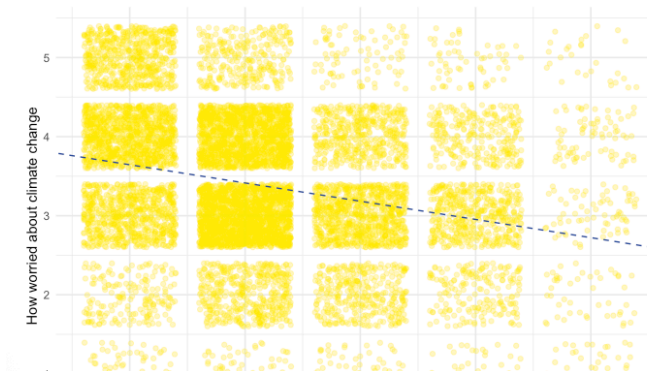
|             | Estimate  | Std. Error | t value | Pr(> t )   |
|-------------|-----------|------------|---------|------------|
| (Intercept) | 3.875143  | 0.021118   | 183.5   | <2e-16 *** |
| rule        | -0.230476 | 0.008695   | -26.5   | <2e-16 *** |

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9092 on 10272 degrees of freedom  
Multiple R-squared: 0.06401, Adjusted R-squared: 0.06392  
F-statistic: 702.5 on 1 and 10272 DF, p-value: < 2.2e-16

And now we plot the model

```
tar_read(plot)
```



# Bibliography

- Kitzes, Justin. 2017. “The Basic Reproducible Workflow Template.” In *The Practice of Reproducible Research: Case Studies and Lessons from the Data-Intensive Sciences*, edited by Justin Kitzes, Daniel Turek, and Fatma Deniz, 19–30. Oakland, CA: University of California Press.
- Landau, Will. 2025. *The targets r Package User Manual*. ropensci.org. <https://books.ropensci.org/targets/>.
- Peikert, Aaron, Caspar J. van Lissa, and Andreas M. Brandmaier. 2021. “Reproducible Research in r: A Tutorial on How to Do the Same Thing More Than Once.” *Psych* 3 (4): 836–67. <https://doi.org/10.3390/psych3040053>.
- Rodrigues, Bruno. 2023. “Building Reproducible Analytical Pipelines with r.” <https://raps-with-r.dev/>.
- Ushey, Kevin, and Hadley Wickham. 2025. *Renv: Project Environments*. R package version 1.1.4,. <https://rstudio.github.io/renv/>.

# Upcoming Events

## Hackathon: Analysing CRONOS data and communicating key insights to policy makers

11 June 2025 - 16 June 2025

Venue: Online

9am-3pm (CEST)

[Register via Zoom](#)



[Register](#)

## Webinar: Data-driven insights for a green, digital, healthy, strong and equal Europe

1 July 2025

Venue: Online

10-11am (CEST)

[Register via Zoom](#)



*Speakers: Lisa Hirsch and Julia Geistberger (University of Vienna)*

[Register](#)

# Feedback



# Q & A