Swiss National Science Foundation

PRODUCING REPRODUCIBLE FIGURES FOR RELIABLE COMMUNICATION: THE SNSF WORKFLOW

Simon Gorin

06 June 2024

AdminR – Spring Meetup

Reference figures at the SNSF: a little tour…

For what are reference figures needed?

Comm. with politicians, research institutions

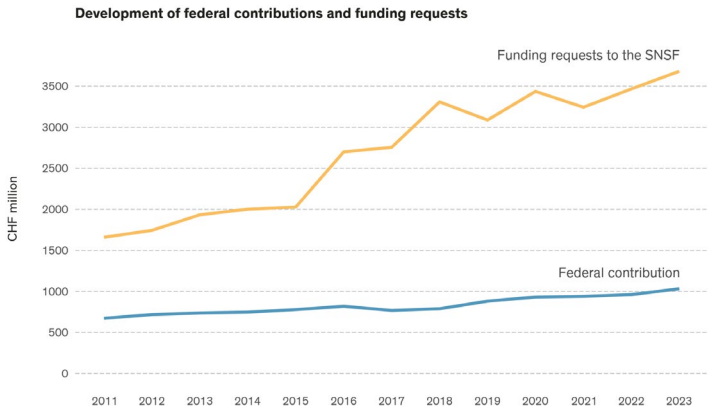
- Funding discussions with the parliament
- Communication with SERI
- Enquiries from researchers/research institutions and partners

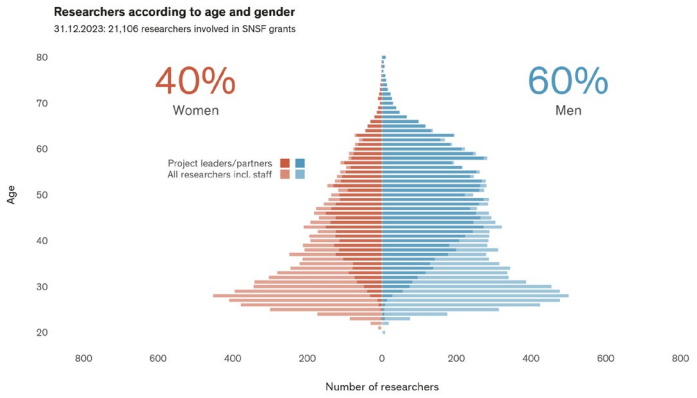
Public communication

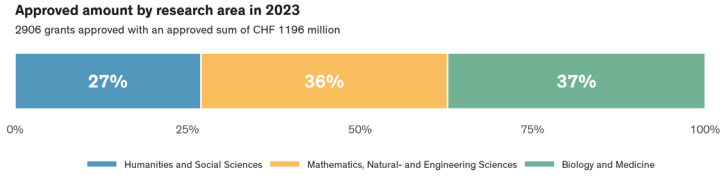
– Annual report on funding activities
– Press release and public documents
– Requests from the media

Internal communication

– Board of Trustees
– Research Council
– Head Office







SNSF

Goals of reference figures: reliable communication and reusability

Using data snapshots for consistency

- Yearly snapshot of internal database
- Ensure consistent figures (always the same for all users)
- Static html page always available (no need to request the figures)

Easy to use

Each figure can be visualized and downloaded in

- 2 formats (pdf, png)
- 3 languages (de, fr, en)
- 3 different font sizes (small, medium, large)

Multilingual

Each figure is available in German, French, and English.

Reusable

- Serves as an example for other analysts to make new visualizations
- Promote corporate design and how to use it

How to make reference figures?

Reference figures: setup

Internal R packages

- `snf.preprocessing`: connection to internal database and utility functions
- `snf.snapshots`: creating and managing database snapshots
- `snf.plot`: ggplot themes and color schemes

Quarto

– Parametrized Quarto document
– Generating different versions of a figure (languages and font sizes) with dynamic `.Rmd` templates and `knitr::knit_child()` / `knitr::knit_expand()`

Posit Connect

– Publication of the html output as a static page

Parametrized Quarto document

```yaml
---
title: "Reference figures `r params$reporting_year`"
subtitle: |
  Page generated on `r stringr::str_replace(lubridate::today(), '(\\d{4})-(\\d{2})-(\\d{2})',
'\\3-\\2-\\1')` | Snapshot from `r stringr::str_replace(params$snap_date, '(\\d{4})-(\\d{2})-(\\d{2})',
'\\3-\\2-\\1')`
format:
  html:
    code-fold: false
    toc: true
    fig-dpi: 300
    fig-format: png
linkcolor: "#6684c1"
include-in-header:
  - text: |
      <style>
      .panel-tabset > .nav-tabs, .panel-tabset > .tab-content {
        border: none;
      }
      .nav-pills .nav-link.active, .nav-pills .show > .nav-link {
        color: #fff;
        background-color: #6684c1;
      }
      .nav-link {
        color: #6684c1;
      }
      figure {
        border: 1px solid #B2B1A7;
        padding: 5px;
      }
      </style>
params:
  reporting_year: 2023
  reporting_day: "-12-31"
  snap_date: "2024-04-11"
---
```

Parametrized Quarto document

Generates a static html page with parametrized:

- reporting year
- date of data snapshot
- page creation date

Managing data snapshots

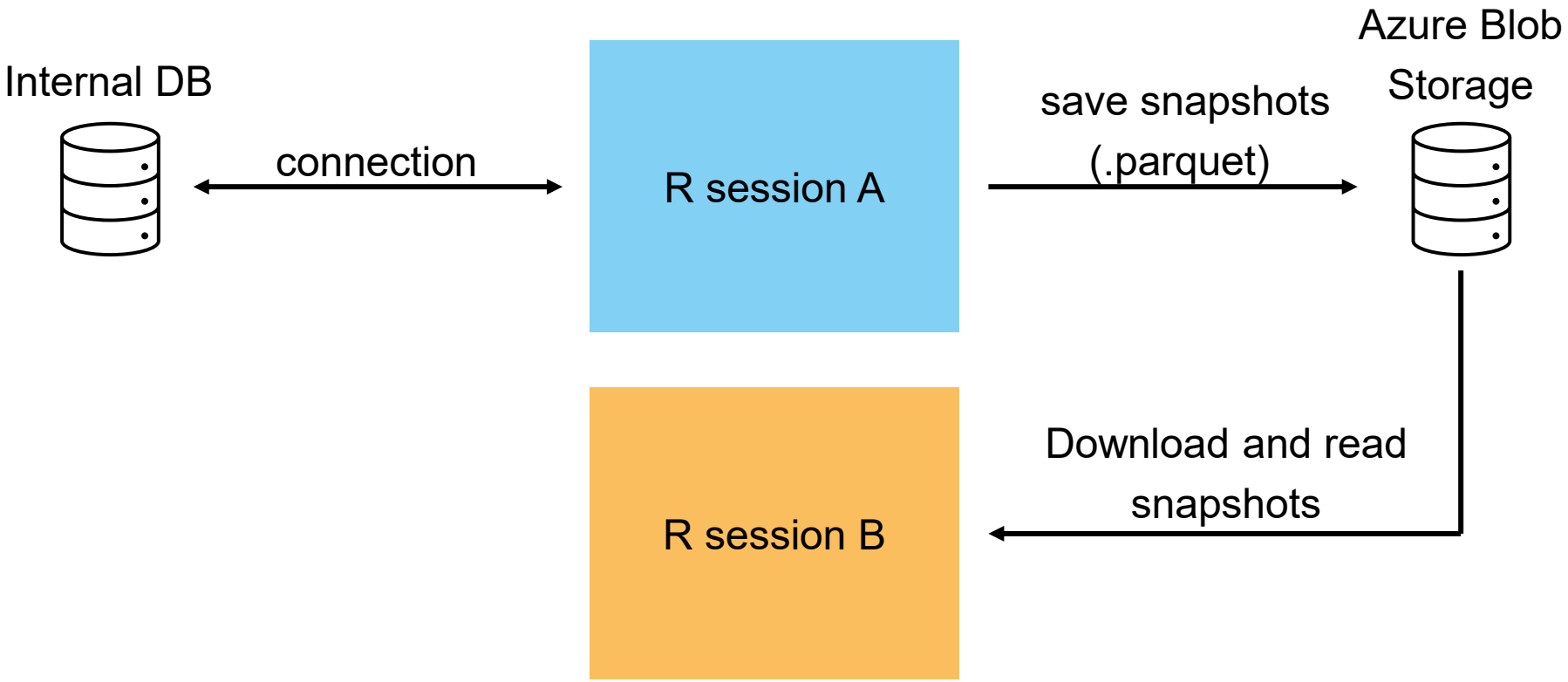
```
# If no snapshots exists for the snapshot date indicated in YAML header, it
# create new ensemble of snapshots.
if (sum(str_detect(list_snapshots(), paste0("datasnapshots/", params$snap_date))) == 0)
  create_snapshot()

# If the snapshots are not available locally, then they are downloaded from ABS
if (!dir.exists(paste0("datasnapshots/", params$snap_date)))
  download_table_snapshot(params$snap_date)

# Read the datasnapshots to the global environment.
read_table_snapshot(params$snap_date, envir = .GlobalEnv)
```

If none existing remotely for the indicated date, create the snapshots

If none existing locally for the indicated date, download locally the snapshots

Read the local snapshots for the indicated date

Managing data snapshots



Creating figures programmatically

````
### Overview over all funding instruments

```{r}
#| echo: false
#| results: asis

src ←
  prep_chunks_all_languages(
    "funding_portfolio_5",
    width = 8,
    height = 4,
    fold_text = "'overall funding by instruments'"
  )

res ← knitr::knit_child(text = src, quiet = TRUE)

cat(res, sep = '\n')

```
````

Generic function generating the chunks for the 9 versions of a figure

Which template to use

Dimensions of the figure

Knit and write the 9 chunks generated with `prep_chunks_all_languages()`

Calling a template with different languages (step 1)

```r
prep_chunks_all_languages <- function(name, width, height, fold_text,
                                      incl_it = FALSE) {

  long_lang <- c("English", "Deutsch", "Français")
  short_lang <- c("en", "de", "fr")
  len <- rep(name, 3)

  if (incl_it) {
    long_lang <- c(long_lang, "Italiano")
    short_lang <- c(short_lang, "it")
    len <- rep(name, 4)
  }

  src <-
    pmap(
      list(
        long_lang,
        short_lang,
        len,
        width,
        height
      ),
      function(lang, lang_short, filename, width, height) {
        paste0(
          "#### ", lang, "\n\n::: {.panel-tabset .nav-pills}\n\n",
          knitr::knit_expand(
            here("templates", paste0(name, "_template.Rmd"))
          ),
          "\n\n:::"
        )
      }
    )
```

Set the languages to use

- Map over the 3 languages (and other parameters)
- Call the figure template
- Knit the `.Rmd` template with dynamic parts using `knitr::knit_expand()`

Dynamic template with 3 font sizes

```
#### Small

```{r fund-portfolio-5-fi-categories-1-{{lang_short}}, fig.align='center', fig.width={{width}},
fig.height={{height}}, dev='png', echo=FALSE}

plot_m ←
  draw_funding_portfolio_5(lang = "{{lang_short}}", font_size = "normal")

plot_m

```

```{r, echo=FALSE}

add_pdf_download(filename = "{{filename}}",
                 plot = plot_m, width = {{width}},
                 height = {{height}}, lang = "{{lang_short}}")
HTML(" ")
add_png_download(filename = "{{filename}}",
                 plot = plot_m, width = {{width}},
                 height = {{height}}, lang = "{{lang_short}}")

```

Medium

```{r fund-portfolio-5-fi-categories-2-{{lang_short}}, fig.align='center', fig.width={{width}},
fig.height={{height}}, dev='png', echo=FALSE}
```

Font size

Language

The plotting function (specific for each figure)

Download options below the figure

...

Chunks for “Medium” and “Large” font sizes

SNSF
```

Calling a template with different languages (step 2)

```
prep_chunks_all_languages ← function(name, width, height, fold_text,
                                     incl_it = FALSE) {
```

...

```
  src_vec ← paste0(unlist(src), collapse = "\n\n")

  src_vec_tab ←
    paste0(
      knitr::knit_child(
        text = fun_chunk,
        quiet = TRUE
      ),
      "\n\n::: {.panel-tabset .nav-pills}\n\n",
      src_vec,
      "\n\n:::\n\n"
    )

  return(src_vec_tab)
}
```

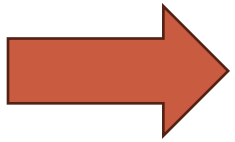
Knitting the 9 chunks:

- Language [de, fr, en] × Size [small, medium, large]

Making a figure: summary

```
src ←
  prep_chunks_all_languages(
    "funding_portfolio_5",
    width = 8,
    height = 4,
    fold_text = "'overall funding by instruments'"
  )
```

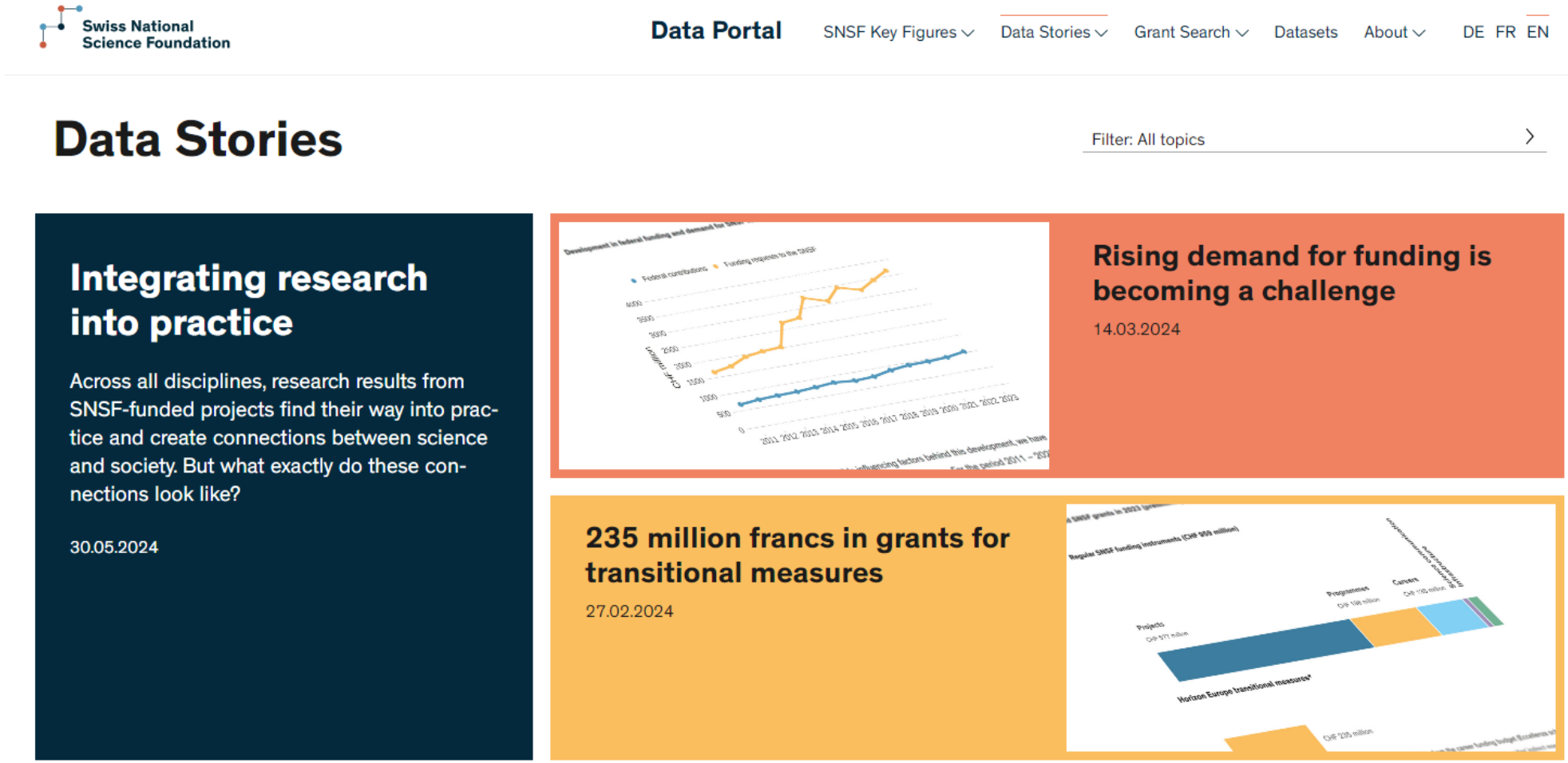
- Knit the corresponding template for each language
- Each template call the corresponding plotting function three times (small, medium, and large font sizes)
- Knit and write the 9 chunks (3 languages ×3 font sizes)



Another example of reproducible figures for communication

SNSF Data Stories

https://data.snf.ch/stories



SNSF Data Stories

https://data.snf.ch/stories

- Insights into research funding in the form of data-driven articles made with:
  - Quarto
  - `snf.datastory` package (https://github.com/snsf-data/snf.datastory)
  - SNSF data story R project template (https://github.com/snsf-data/datastory_template)

- Data and analysis code are made openly available on Github when possible

- The data story SNSF open data: Who gets funding? How to calculate figures yourself shows how access and analyse yourself open SNSF data

Thank you!

Merci !

Vielen Dank!