Importing Data

Getting Data into RStudio

by Martin Frigaard

Written: October 03 2022

Updated: April 14 2023

Created using the "λέξις" theme

Materials

The slides are in the `slides.pdf` file

The materials for this training are in the `worksheets` folder:

```
worksheets
├── import.Rmd
├── export.Rmd
├── objects.Rmd
├── rmd-basic.Rmd
├── rmd-tables.Rmd
└── rmd-visualizations.Rmd
```

Outline

1. Importing data
2. Common Data Objects
3. R Markdown
4. R Markdown Data Visualizations
5. R Markdown Tables
6. Exporting Data

Import Data

Open `import.Rmd` to follow along

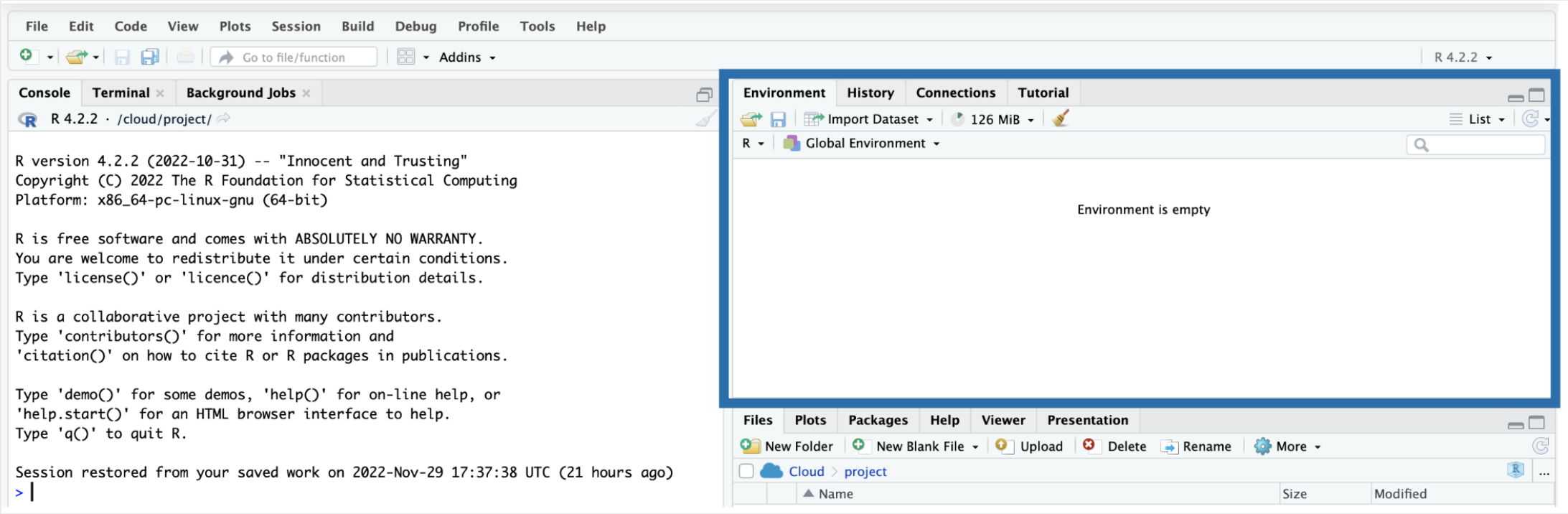
Importing Data

Packages for importing data:

File type	Package
SAS (`.sas7bdat`)	`haven`
Excel (`.xlsx`, `.xls`)	`readxl`, `openxlsx`
Plain Text (`.csv`, `.tsv`, `.txt`)	`readr`, `data.table`

Importing Data (*Environment*)

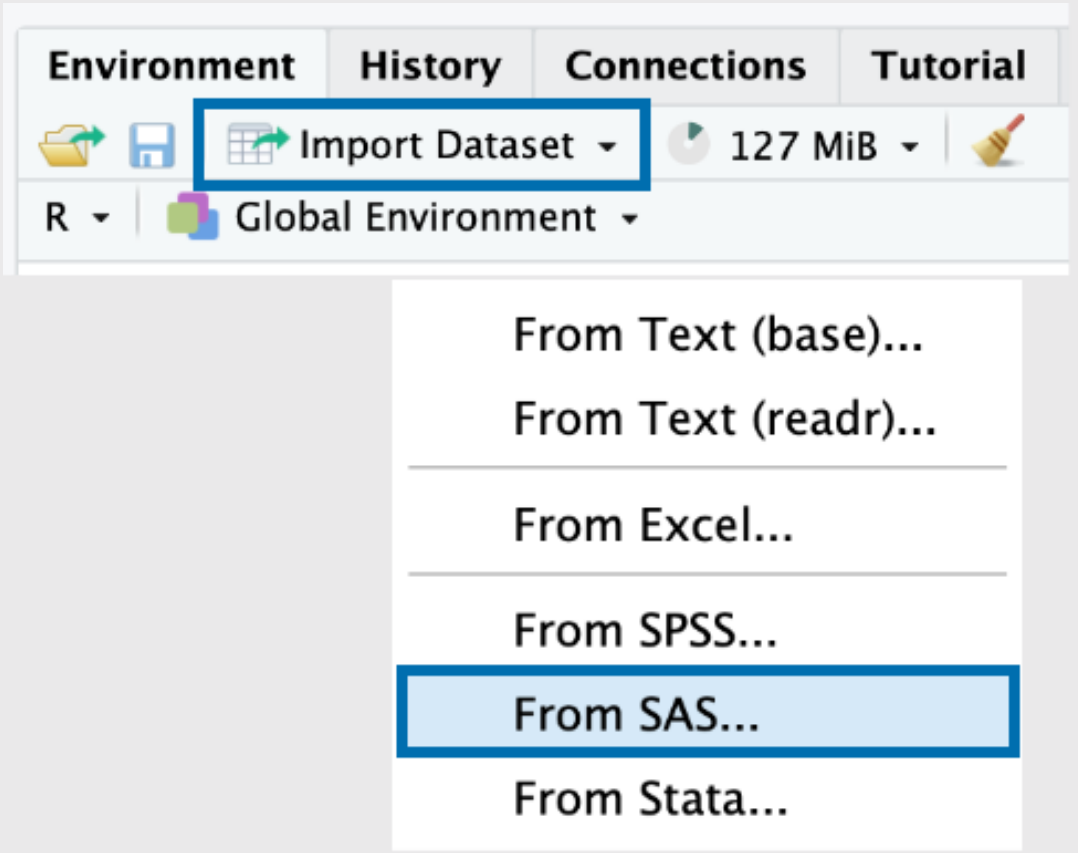
The Environment Pane



Importing Data (*Import Dataset*)

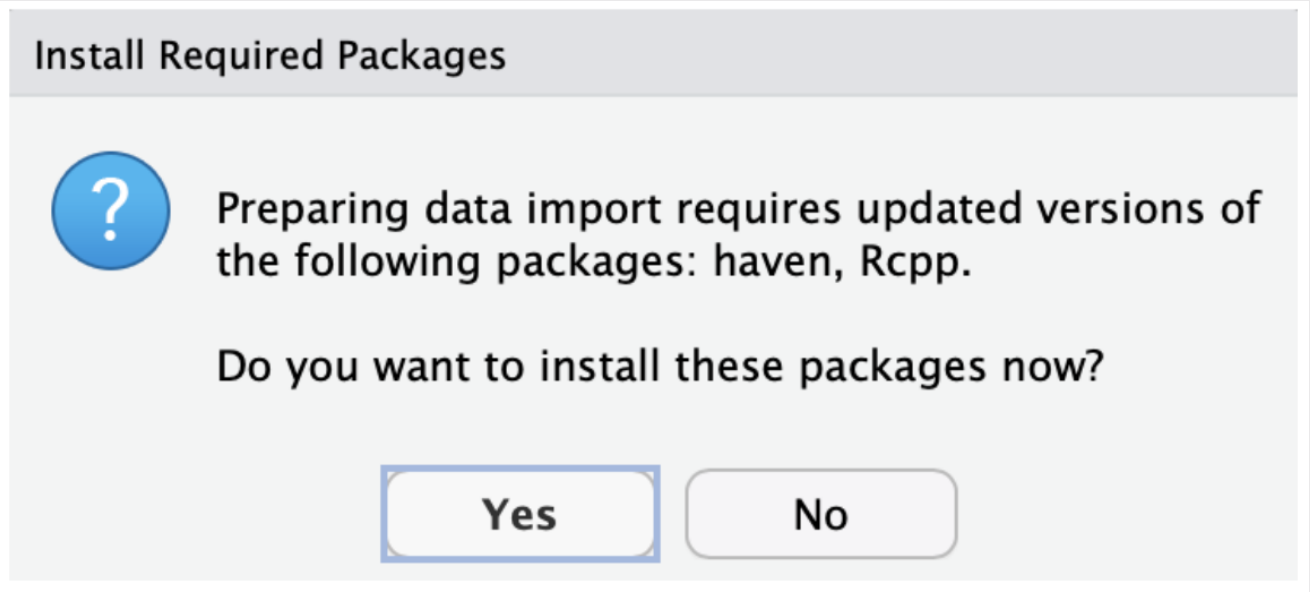
Click Import Dataset

Click From SAS



Importing Data (*Required Packages*)

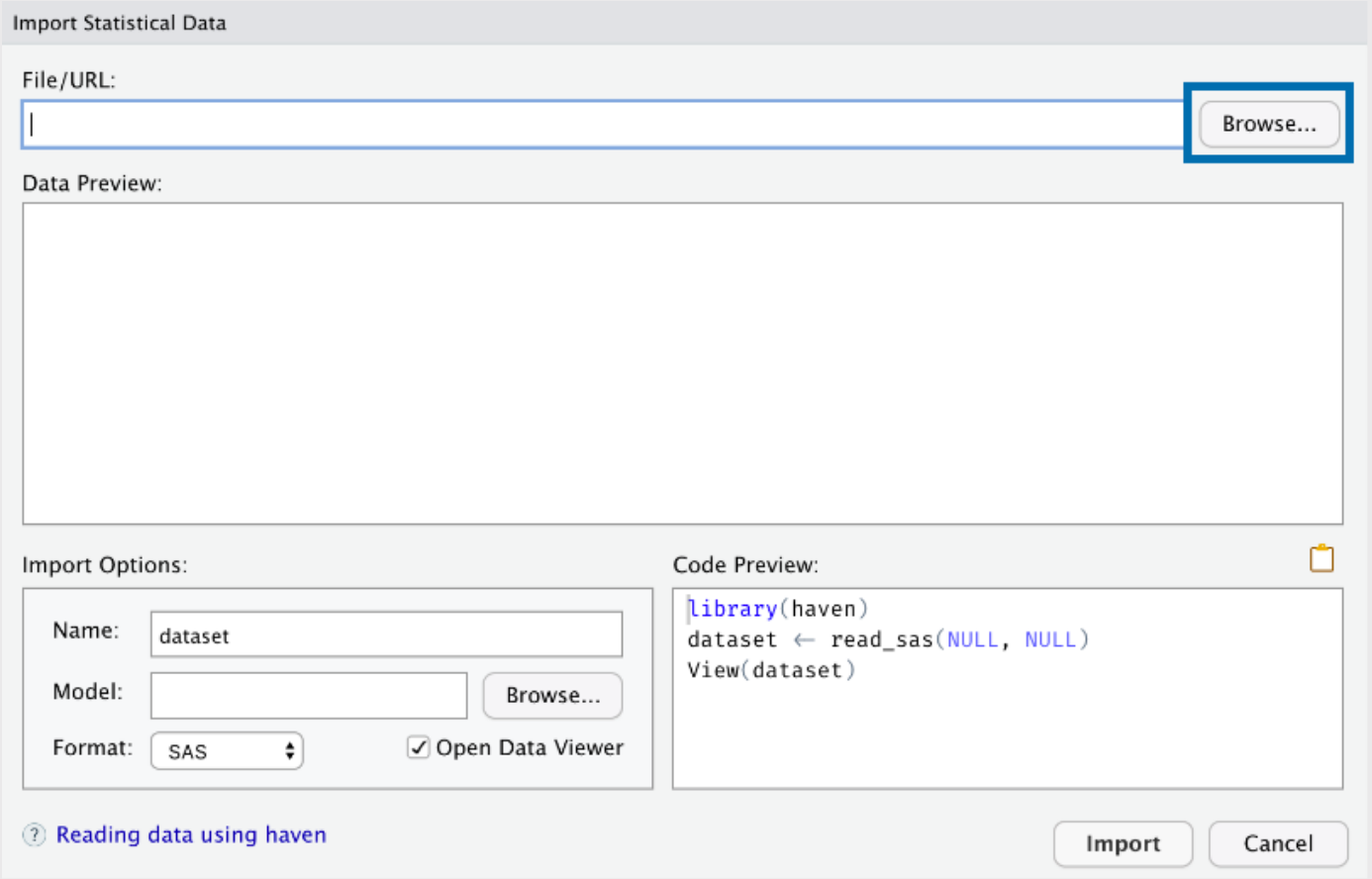
If you see a prompt to install required packages, click Yes



Importing Data (*Dialogue Box*)

You will see the Import Statistical Data Dialogue Box

Click Browse and navigate to the `data/medical.sas7bdat` file



Importing Data (*Dialogue Box*)

You will see the path in File/URL

A preview of the data will appear in Data Preview

File/URL:

/cloud/project/data/medical.sas7bdat

Data Preview:

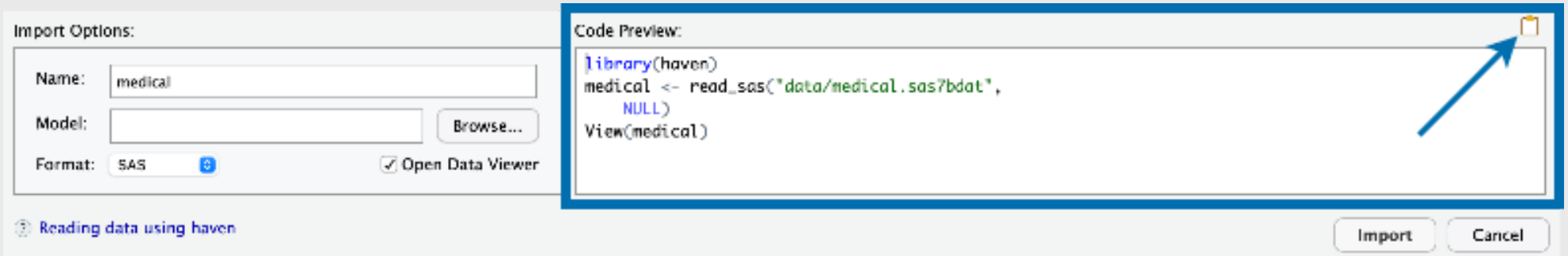
ID = person identifier	YEAR = year index	MEDEXP = annual medical expenditure in hundreds of dollars	INC = annual income in thousands of dollars	AGE = age in years	INSUR = 1 if individual i has private health insurance in year t and...
1	1	9	49	51	1
1	2	9	51	52	1
1	3	9	55	53	1
1	4	10	58	54	1
1	5	11	61	55	1
2	1	6	48	62	1
2	2	7	48	63	1
2	3	7	58	64	1
2	4	7	59	65	1
2	5	7	63	66	1
3	1	4	46	57	0
3	2	3	51	58	0
3	3	5	55	59	0
3	4	4	58	60	0
3	5	4	63	61	0
4	1	5	68	48	1
4	2	3	70	49	1
4	3	6	75	50	1

Previewing first 50 entries.

Importing Data (*Dialogue Box*)

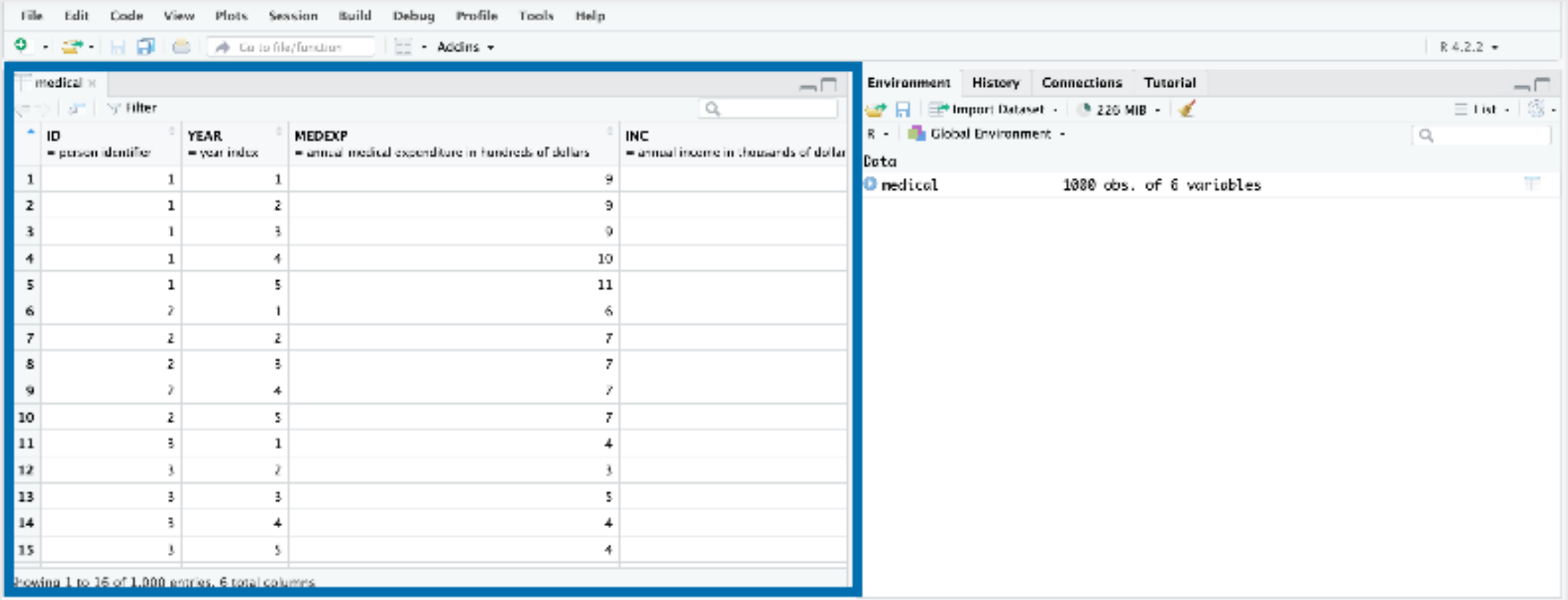
You see we have additional Import Options

We also see a Code Preview. Click on the small copy icon, then click Import



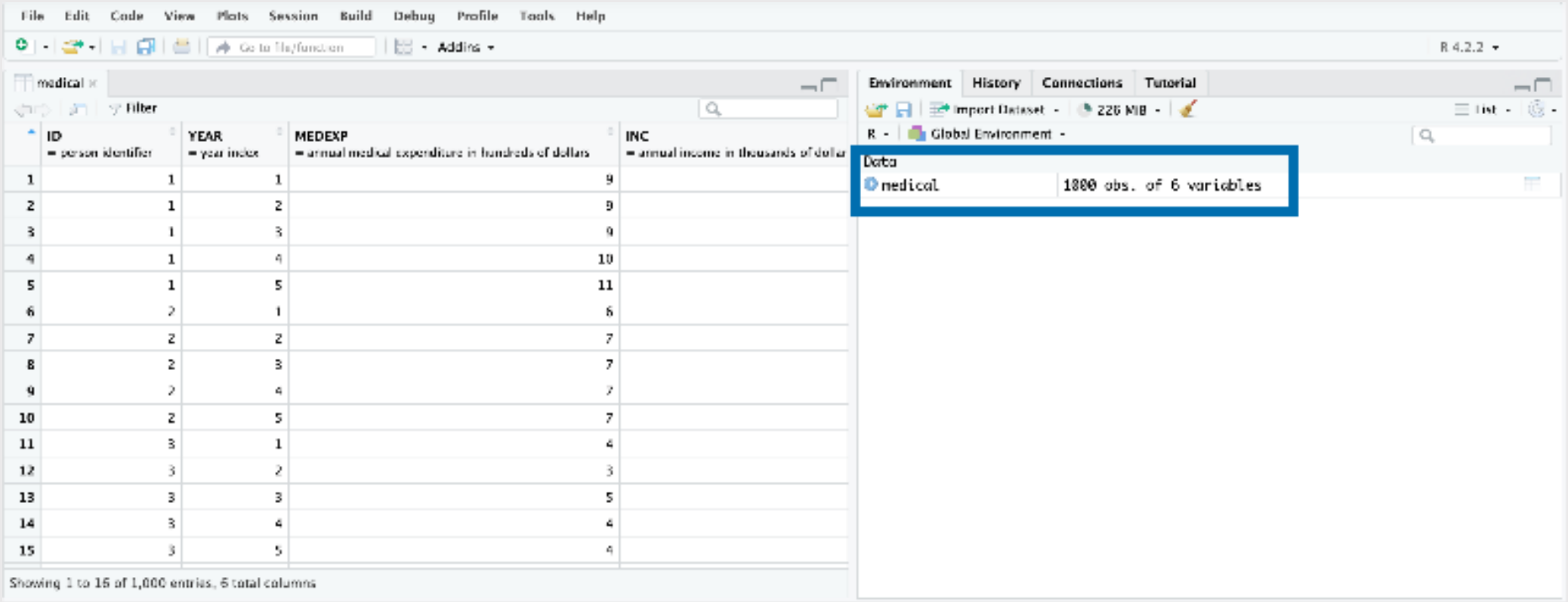
Importing Data (*Data Viewer*)

RStudio imports the data and opens it in the Data Viewer



Importing Data (*Data Viewer*)

We can also see `medical` has been added to our Environment pane



Importing Data

Is what we did reproducible?

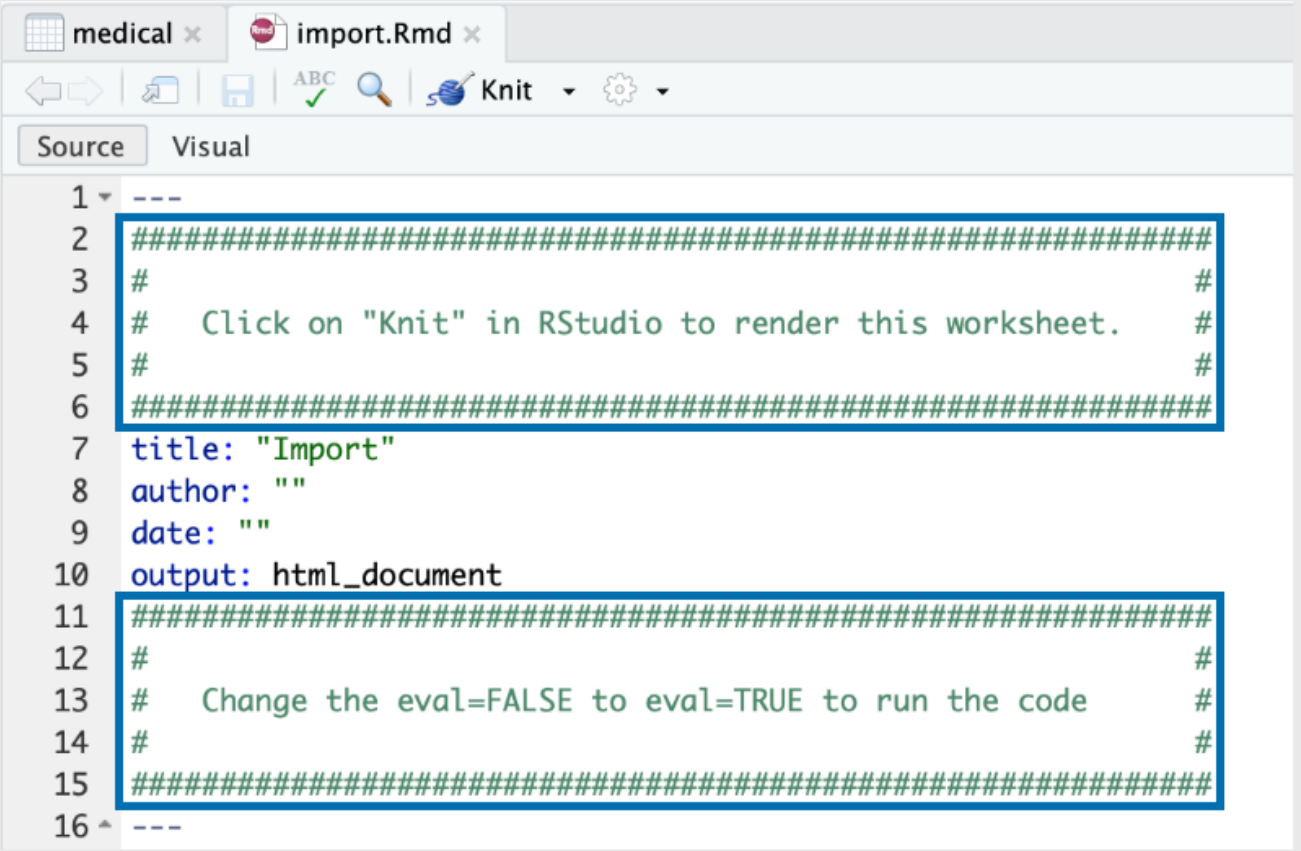
No, but it can be!

Open `import.Rmd` from the `worksheets` folder

Importing Data

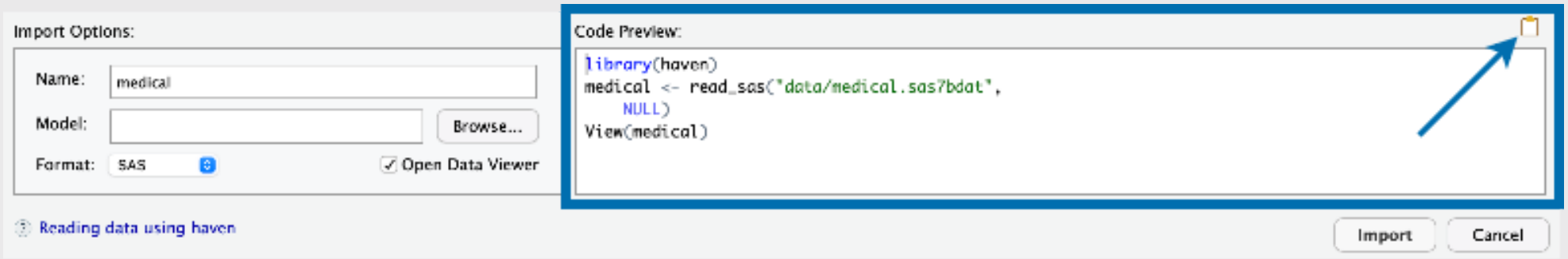
In `Import.Rmd`

- Instructions inside `#` boxes won't run
- Fill in `author` and `date` (inside quotes)



Importing Data (from local)

We already have the code to import `medical.sas7bdat` from local



We need to adjust the file path to `../data/medical.sas7bdat`

```
. # importing with dialogue
└── data/
    └── medical.sas7bdat
```

```
. # importing from file
├── data/
│   └── medical.sas7bdat
└── worksheets/
    └── import.Rmd
```

Importing Data (download and import)

We can also download the file from a `url`

```
download.file(
    url = "http://www.principlesofeconometrics.com/sas/medical.sas7bdat",
    )
```

And save this to a local `destfile`

```
download.file(
    url = "http://www.principlesofeconometrics.com/sas/medical.sas7bdat",
    destfile = "../data/downloads/medical.sas7bdat")
```

Importing Data (download and import)

Now we can import the file from our `downloads/` folder

```
. # importing from downloads folder
├── data/
│   ├── medical.sas7bdat
│   └── downloads/
│           └── medical.sas7bdat
└── worksheets/
    └── import.Rmd
```

```
medical <- read_sas("../data/downloads/medical.sas7bdat")
```

Importing Data (parameters)

For a more permanent solution, we can use parameters in our R Markdown file to store file location (or other metadata)

```
title: "May Report"
author: "Joe Smith"
date: "2022-11-30"
output: html_document

params:
  sas_data_url: !r file.path("http://www.principlesofeconometrics.com/sas/medical.sas7bdat")
  sas_data_dir: !r c("../data/sas/")
```

```
download.file(url = params$sas_data_url,
    )
```

```
download.file(url = params$sas_data_url,
    destfile = params$sas_data_dir)
```

Importing Data (multiple files)

If we have a folder with multiple files, we can reduce duplicated code with iteration.

```
. # importing multiple files
├── data/sas/
│   ├── elemapi-2000.sas7bdat
│   ├── elemapi2-2000.sas7bdat
│   ├── hsb2.sas7bdat
│   └── nations.sas7bdat
└── worksheets/
    └── import.Rmd
```

```
# create vector of files
sas_filenames <- list.files(
            path = "../data/sas",
            full.names = TRUE)
all_sas_data <- sas_filenames |>
    # give this vector names
    purrr::set_names() |>
    # use read_sas() on all files
    purrr::map(.x = , .f = read_sas)
```

`all_sas_data` is a list of datasets

Importing Data (multiple files)

Each named according to their path in `data/sas/`

```
str(all_sas_data)
# $ ../data/sas/elemapi-2000.sas7bdat : tibble [400 × 21] (S3: tbl_df/tbl/data.frame)
#   ..$ snum    : num [1:400] 906 889 887 876 888 ...
#   .. ..- attr(*, "label")= chr "school number"
#   ..$ dnum    : num [1:400] 41 41 41 41 41 98 98 108 108 108 ...
#   .. ..- attr(*, "label")= chr "district number"
#   .. [list output truncated]
#  $ ../data/sas/elemapi2-2000.sas7bdat: tibble [400 × 22] (S3: tbl_df/tbl/data.frame)
#   ..$ snum    : num [1:400] 906 889 887 876 888 ...
#   .. ..- attr(*, "label")= chr "school number"
#   ..$ dnum    : num [1:400] 41 41 41 41 41 98 98 108 108 108 ...
#   .. ..- attr(*, "label")= chr "district number"
#   .. [list output truncated]
#  $ ../data/sas/hsb2.sas7bdat         : tibble [200 × 11] (S3: tbl_df/tbl/data.frame)
#   ..$ id     : num [1:200] 3 5 16 35 8 19 6 1 4 22 ...
#   ..$ female : num [1:200] 0 0 0 1 1 1 1 1 1 0 ...
#   .. [list output truncated]
#  $ ../data/sas/nations.sas7bdat      : tibble [109 × 15] (S3: tbl_df/tbl/data.frame)
#   ..$ country : chr [1:109] "Algeria" "Argentin" "Australi" "Austria" ...
#   .. ..- attr(*, "label")= chr "Country"
#   ..$ pop     : num [1:109] 21.9 30.5 15.8 7.6 100.6 ...
#   .. ..- attr(*, "label")= chr "1985 population in millions"
#   .. [list output truncated]
```