

Introduction to the Tidyverse

Import, wrangle, model, and
communicate data

2023-08-02



Working with data in R

the tidyverse is a collection of *friendly and consistent* tools for data analysis and visualization.

They live as R packages, each of which does one thing well.

library(tidyverse) will load the core packages:



ggplot2, for data visualisation.

dplyr, for data manipulation.

tidyr, for data tidying.

readr, for data import.

purrr, for functional programming.

tibble, for tibbles, a modern re-imagining of data frames.

stringr, for strings.

forcats, for factors.

lubridate, for dates and times.

This course is hands on!

Each section has an
exercises file:
exercises.qmd

exercises.qmd

The image shows a screenshot of a Quarto document editor. The top bar includes a file explorer showing 'exercises.qmd', a toolbar with navigation and editing icons, and a 'Render' button. Below the toolbar, there are tabs for 'Source' and 'Visual', and an 'Outline' button on the right. The main editor area displays a code chunk starting with a title 'Import Data' and a format of 'html'. The code includes library calls for 'tidyverse' and 'haven'. The chunk content describes learning about importing and exporting files using 'readr' and 'haven' packages. The right sidebar shows an outline of the document structure, listing sections like 'readr', 'Sample data', 'Importing Data', and various 'Your Turn' exercises. The bottom status bar shows the cursor position at line 5, column 1, and the document is titled 'Quarto'.

Code chunks

```
```\{r}  
csv_data <- read_csv(
 "a,b,c,d
 1,2,3,4
 5,6,7,8",
 col_types = ""
)

csv_data
```\
```



Running code chunks

```
```{r}
csv_data <- read_csv(
 "a,b,c,d
1,2,3,4
5,6,7,8",
 col_types = ""
)
```

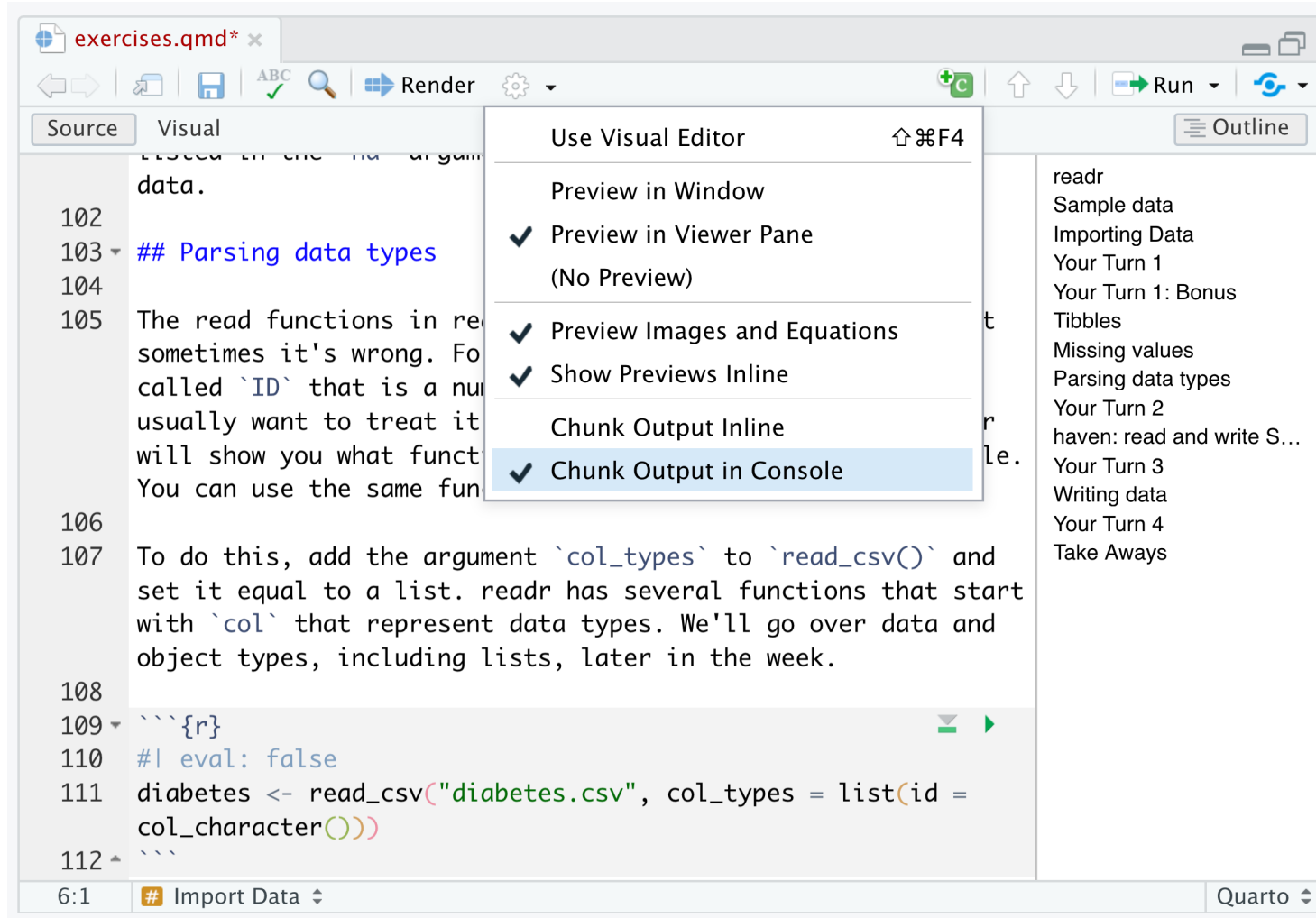
csv\_data  
```



| a | b | c | d |
|----------|----------|----------|----------|
| <dbl> | <dbl> | <dbl> | <dbl> |
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |

2 rows

Outputting to the console



The screenshot shows the Quarto editor interface with a file named `exercises.qmd` open. The 'Source' pane displays R code with line numbers 102 through 112. A context menu is open over the code, showing options for previewing and outputting. The 'Chunk Output in Console' option is selected and highlighted in blue.

Code in the Source pane:

```
102 data.  
103 ## Parsing data types  
104  
105 The read functions in readr sometimes it's wrong. For example, a column called `ID` that is a numeric variable, but we usually want to treat it as a character variable. It will show you what function to use. You can use the same function to read a CSV file.  
106  
107 To do this, add the argument `col_types` to `read_csv()` and set it equal to a list. readr has several functions that start with `col` that represent data types. We'll go over data and object types, including lists, later in the week.  
108  
109 ```{r}  
110 #| eval: false  
111 diabetes <- read_csv("diabetes.csv", col_types = list(id =  
112   col_character()))  
113 ```
```

Context menu options:

- Use Visual Editor (⌘F4)
- Preview in Window
- ✓ Preview in Viewer Pane (No Preview)
- ✓ Preview Images and Equations
- ✓ Show Previews Inline
- Chunk Output Inline
- ✓ Chunk Output in Console

Outline pane on the right:

- readr
- Sample data
- Importing Data
- Your Turn 1
- Your Turn 1: Bonus
- Tibbles
- Missing values
- Parsing data types
- Your Turn 2
- haven: read and write S...
- Your Turn 3
- Writing data
- Your Turn 4
- Take Aways

Bottom status bar: 6:1 # Import Data ↕ Quarto ↕

Project contents

```
|— 01-dplyr_5verbs
|   |— cheatsheet_dplyr_5verbs.pdf
|   |— diabetes.csv
|   |— exercises.qmd
|   |— slides.pdf
```

