

# A Short Introduction to Working With Data in R

Jonathan Whiteley

2023-08-28

# Prerequisites

- Access to a copy of the  software
  - ▶ i.e., a “binary executable”
  - ▶ Go to [www.r-project.org](http://www.r-project.org) to get a copy, or ask your system administrator.
- Tidyverse packages installed on the same system as R
  - ▶ Please run this command in R *before* the workshop:

```
install.packages("tidyverse")
```

- Knowledge of common mathematical operations: arithmetic, logarithms, etc.
- Knowledge of basic R concepts, such as *variables*, *objects*, *operators*, *functions*, *packages*, etc.
  - ▶ This is covered in the first workshop: “A Gentle Introduction to R”

# Learning Objectives

- Load tabular data into R
- Explore data to check that it was loaded correctly
- Export data from R to external files
- Data frames
- Clean data
  - ▶ Add & change columns
  - ▶ Edit values systematically
  - ▶ Change data types
- Tidy data
  - ▶ Change the *shape* of a data frame
- Re-use code, reproducible results, automated reports
  - ▶ Scripts
  - ▶ R Markdown, R Notebooks

# Section 1

## Welcome

# Pop Quiz

We will review these *at the end*, so you can see how much you have learned.



Answer in the chat:

What is your favourite emoji? Why do you like to use it so much?

# Introductions

- Name
- Pronouns
- Job title, role
- *optional*: a favourite childhood treat or candy?
- What are you hoping to learn most in today's workshop?

## Section 2

### Loading data into R

## Section 3

### Exploring your data



# Data frames

## Section 4

### Saving data outside R

## Section 5

### Re-using your code: scripts and other files

# Re-using your code: scripts and other files

## Section 6

### The tidyverse collection of packages

```
install.packages("tidyverse")  
help(package="tidyverse")
```

- The `tidyverse` is actually an “opinionated” collection of packages that are designed to work together.
- All packages share an underlying design philosophy, grammar, and data structures.

Today, we will focus on a few of the core packages for loading, cleaning, and manipulating data:

- `tidyr`
- `dplyr`
- `readr`, `readxl`

# A 'pipe' operator



Figure 1: “La Trahison des Images” (“The Treachery of Images”) or “Ceci n'est pas une pipe” (“This is not a pipe”) by René Magritte.



- The `magrittr` package (included with `tidyverse`) provides a “forward-pipe operator”:

```
%>% # ?magrittr::`%>`
```

- The `magrittr` package is automatically loaded when loading most `tidyverse` packages (e.g., `tidyr`, `dplyr`, `ggplot2`), as these packages all use this operator extensively.
  - ▶ It is often unnecessary to load `magrittr` separately, unless you are **not** using these other packages.

## magrittr's 'forward-pipe' operator

- `%>%` allows you to pass results from an expression on the left-hand side (LHS) as the first argument to a *function call* on the right-hand side (RHS). The following are equivalent:

This expression ...	is equivalent to:
<code>x %&gt;% f()</code>	<code>f(x)</code>
<code>x %&gt;% f(y)</code>	<code>f(x, y)</code>
<code>x %&gt;% f(y, z = .)</code>	<code>f(y, z = x)</code>
<code>x %&gt;% f %&gt;% g %&gt;% h</code>	<code>h(g(f(x)))</code>

- This can make code easier to read, as expressions are written and evaluated from *left to right*, rather than from *inside to outside* nested parentheses.



## R now has a ‘native’ pipe operator

- A pipe operator was introduced in base R in v4.1 (May 2021)<sup>1</sup>:

```
|>      # ?pipeOp
```

- It was inspired by the “forward pipe operator” introduced by `magrittr`, but is more streamlined. See these links for details:
  - ▶ Differences between the base R and `magrittr` pipes
  - ▶ “Understanding the native R pipe `|>`”
- Because it is so new, most code examples online still use `%>%` from `magrittr`.
- This document will use `%>%` in the examples, for consistency and because it was designed to work with other tidyverse functions.
- But `|>` might work well for you in simple cases, without having to load any additional packages.

---

<sup>1</sup><https://cran.r-project.org/bin/windows/base/old/4.1.0/NEWS.R-4.1.0.html>

# Pipes: exercise

## Section 7

### Clean data

## Section 8

### Tidy data

# Tidy datasets

*Happy families are all alike; every unhappy family is unhappy in its own way*

— Leo Tolstoy

*Like families, tidy datasets are all alike but every messy dataset is messy in its own way.*

— Hadley Wickham (doi: [10.18637/jss.v059.i10](https://doi.org/10.18637/jss.v059.i10))

- Tidy datasets provide a standardized way to link the *structure* of a dataset (its physical layout) with its *semantics* (its meaning).

▶ `tidyr` vignette

# Section 9

## Review

# Exercise

# Quiz Review



## Section 10

## Backmatter

# References