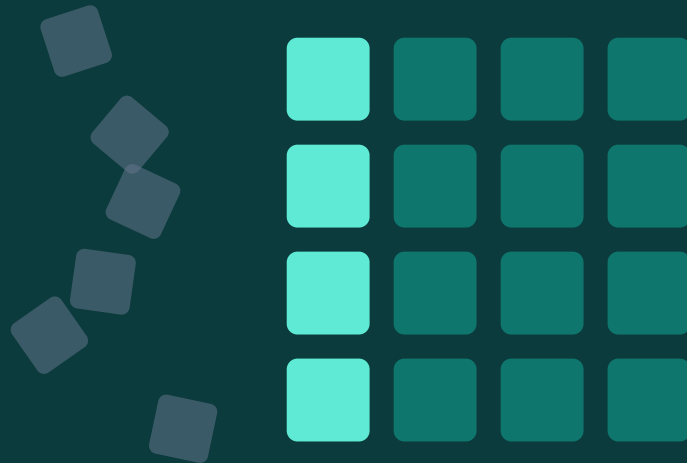


# Tidy Your Data in R

Jane Stephenson

Psych 10

Wednesday, October 28th

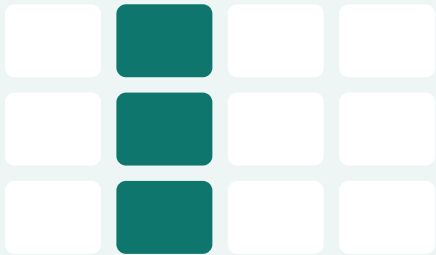


# What is tidy data?

1

## Each variable is a column

Score, group, and week each get their own column.

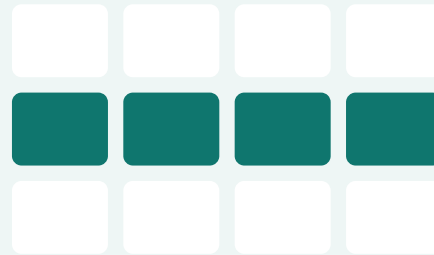


|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

2

## Each observation is a row

One student in one week = one row.

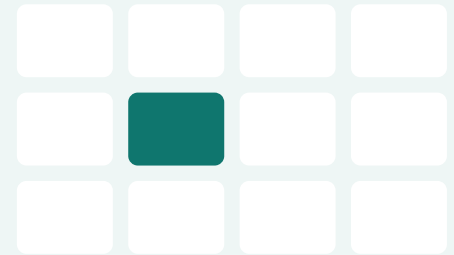


|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

3

## Each value is a cell

One number per cell, no "10/28/2026" stuffed together.



|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Why tidy?

1

Once you learn the rules of tidy data and how to analyze and visualize data in this format, you can apply it to any tidy dataset

2

Organizing variables in columns allows R's vectorized nature to shine because most R functions work with vectorized data.

## Step 1: Look at your data file

| combined_sales.xlsx - LibreOffice Calc                           |       |     |             |          |             |          |        |          |            |          |       |
|------------------------------------------------------------------|-------|-----|-------------|----------|-------------|----------|--------|----------|------------|----------|-------|
| File Edit View Insert Format Styles Sheet Data Tools Window Help |       |     |             |          |             |          |        |          |            |          |       |
|                                                                  | A     | B   | C           | D        | E           | F        | G      | H        | I          | J        |       |
| 1                                                                |       |     | Noblesville |          | Bloomington |          | Muncie |          | South Bend |          |       |
| 2                                                                | month | day | price       | quantity | price       | quantity | price  | quantity | price      | quantity |       |
| 3                                                                |       | 1   | 1           | 18.50    | 2,764       | 18.50    | 2,631  | 18.50    | 2,928      |          |       |
| 4                                                                |       | 1   | 8           | 11.51    | 2,751       | 11.51    | 2,788  | 11.51    | 3,162      |          |       |
| 5                                                                |       | 1   | 15          | 14.78    | 2,612       | 14.78    | 2,989  | 14.78    | 2,986      |          |       |
| 6                                                                |       | 1   | 22          | 13.73    | 2,723       | 13.73    | 2,827  | 13.73    | 2,978      |          |       |
| 7                                                                |       | 1   | 29          | 15.80    | 2,764       | 15.80    | 2,804  | 15.80    | 2,885      |          |       |
| 8                                                                |       | 2   | 5           | 14.41    | 2,369       | 14.41    | 2,932  | 14.41    | 3,134      |          |       |
| 9                                                                |       | 2   | 12          | 14.56    | 2,931       | 14.56    | 2,706  | 14.56    | 3,020      |          |       |
| 10                                                               |       | 2   | 19          | 15.10    | 2,317       | 15.10    | 2,948  | 15.10    | 2,766      |          |       |
| 11                                                               |       | 2   | 26          | 13.34    | 2,479       | 13.34    | 2,683  | 13.34    | 3,063      |          |       |
| 12                                                               |       | 3   | 4           | 14.15    | 2,967       | 14.15    | 3,017  | 14.15    | 3,050      |          |       |
| 13                                                               |       | 3   | 11          | 14.55    | 2,600       | 14.55    | 2,839  | 14.55    | 2,651      |          |       |
| 14                                                               |       | 3   | 18          | 18.46    | 2,485       | 18.46    | 2,601  | 18.46    | 2,816      |          |       |
| 15                                                               |       | 3   | 25          | 18.73    | 2,285       | 18.73    | 2,880  | 18.73    | 2,808      |          |       |
| 16                                                               |       | 4   | 1           | 13.10    | 2,671       | 13.10    | 2,772  | 13.10    | 3,027      |          |       |
| 17                                                               |       | 4   | 8           | 14.48    | 2,902       | 14.48    | 3,168  | 14.48    | 2,661      |          |       |
| 18                                                               |       | 4   | 15          | 14.09    | 2,390       | 14.09    | 2,993  | 14.09    | 3,339      |          |       |
| 19                                                               |       | 4   | 22          | 17.60    | 2,053       | 17.60    | 2,916  | 17.60    | 3,498      |          |       |
| 20                                                               |       | 4   | 29          | 19.90    | 2,346       | 19.90    | 2,573  | 19.90    | 2,910      |          |       |
| 21                                                               |       | 5   | 6           | 15.77    | 2,771       | 15.77    | 2,920  | 15.77    | 2,479      | 15.77    | 3,143 |
| 22                                                               |       | 5   | 13          | 15.99    | 2,685       | 15.99    | 2,587  | 15.99    | 2,985      | 15.99    | 2,489 |
| 23                                                               |       | 5   | 20          | 18.20    | 2,560       | 18.20    | 2,326  | 18.20    | 3,127      | 18.20    | 3,073 |
| 24                                                               |       | 5   | 27          | 14.94    | 2,579       | 14.94    | 2,585  | 14.94    | 2,842      | 14.94    | 3,453 |
| 25                                                               |       | 6   | 3           | 17.37    | 2,290       | 17.37    | 2,770  | 17.37    | 2,966      | 17.37    | 2,660 |
| 26                                                               |       | 6   | 10          | 12.88    | 2,787       | 12.88    | 2,936  | 12.88    | 3,039      | 12.88    | 3,095 |
| 27                                                               |       | 6   | 17          | 19.38    | 2,300       | 19.38    | 2,680  | 19.38    | 2,607      | 19.38    | 3,077 |
| 28                                                               |       | 6   | 24          | 11.43    | 2,400       | 11.43    | 2,946  | 11.43    | 3,587      | 11.43    | 3,619 |
| 29                                                               |       | 7   | 1           | 16.16    | 2,422       | 16.16    | 2,628  | 16.16    | 2,815      | 16.16    | 2,051 |

# Reading in your data

```
library(tidyverse)
read_csv(
  file = "data/my_file.csv",      # specify the name and location of your file
  skip = 2,                       # skip the first two rows
  comment = "#",                  # skip all rows that start with "#"
  na = c("N/A", ""),             # change cells with "N/A" or blank to R's NA
  col_names = FALSE),            # tell R that there are no column names
  col_names = c("x", "y", "z"))  # tell R that this is what you want to call
                                # the column names
```

```
library(readxl)
read_excel("my_file.xlsx")
```

```
library(haven)
read_sav("my_file.sav")
```

# Your tidy toolkit: the pipe

The pipe `|>` means “then”

## No pipe

```
summarize(
  group_by(
    filter(
      mutate(students, total_hours =
hours_wk1 + hours_wk2 + hours_wk3),
      housing == "off"
    ),
    year
  ),
  mean_total = mean(total_hours),
  .groups = "drop"
)
```

## Pipe

```
students |>
  mutate(total_hours = hours_wk1 +
    hours_wk2 + hours_wk3) |>
  filter(housing == "off") |>
  group_by(year) |>
  summarize(mean_total =
    mean(total_hours),
    .groups = "drop")
```

install once: `install.packages("tidyverse")`    load each session: `library(tidyverse)`

# Your tidy toolkit: the pipe

The pipe `|>` means “then”

## interim names

```
step1 <- mutate(students,  
  total_hours = hours_wk1 + hours_wk2  
                + hours_wk3)  
step2 <- filter(step1,  
  housing == "off")  
step3 <- group_by(step2, year)  
summarize(step3, mean_total =  
  mean(total_hours), .groups = "drop")
```

## Pipe

```
students |>  
  mutate(total_hours = hours_wk1 +  
            hours_wk2 + hours_wk3) |>  
  filter(housing == "off") |>  
  group_by(year) |>  
  summarize(mean_total =  
            mean(total_hours),  
            .groups = "drop")
```

install once: `install.packages("tidyverse")`    load each session: `library(tidyverse)`

# Meet the data: study habits

| ID | Yr        | Hsg | GPA | Sleep | wk1_hrs | wk2_hrs | wk3_hrs |
|----|-----------|-----|-----|-------|---------|---------|---------|
| 1  | Freshman  | on  | 3.2 | 7.5   | 10      | 12      | 11      |
| 2  | Freshman  | off | 3.6 | 8.0   | 14      | 15      | 16      |
| 3  | Freshman  | on  | 2.9 | 6.5   | 8       | 9       | 10      |
| 4  | Sophomore | on  | 3.4 | 7.0   | 12      | 13      | 14      |
| 5  | Sophomore | off | 3.8 | 7.5   | 16      | 18      | 17      |
| 6  | Sophomore | on  | 3.0 | 6.0   | 9       | 10      | 11      |
| 7  | Junior    | off | 3.5 | 7.0   | 15      | 14      | 16      |
| 8  | Junior    | on  | 3.1 | 6.5   | 11      | 12      | 12      |
| 9  | Junior    | off | 3.9 | 8.0   | 18      | 17      | 19      |
| 10 | Senior    | on  | 3.3 | 7.0   | 13      | 14      | 15      |
| 11 | Senior    | off | 3.7 | 7.5   | 17      | 16      | 18      |
| 12 | Senior    | on  | 3.0 | 6.5   | 10      | 11      | 12      |

# Meet the data: study habits

| ID | Yr        | Hsg | GPA | Sleep | wk1_hrs | wk2_hrs | wk3_hrs |
|----|-----------|-----|-----|-------|---------|---------|---------|
| 1  | Freshman  | on  | 3.2 | 7.5   | 10      | 12      | 11      |
| 2  | Freshman  | off | 3.6 | 8.0   | 14      | 15      | 16      |
| 3  | Freshman  | on  | 2.9 | 6.5   | 8       | 9       | 10      |
| 4  | Sophomore | on  | 3.4 | 7.0   | 12      | 13      | 14      |
| 5  | Sophomore | off | 3.8 | 7.5   | 16      | 18      | 17      |
| 6  | Sophomore | on  | 3.0 | 6.0   | 9       | 10      | 11      |
| 7  | Junior    | off | 3.5 | 7.0   | 15      | 14      | 16      |
| 8  | Junior    | on  | 3.1 | 6.5   | 11      | 12      | 12      |
| 9  | Junior    | off | 3.9 | 8.0   | 18      | 17      | 19      |
| 10 | Senior    | on  | 3.3 | 7.0   | 13      | 14      | 15      |
| 11 | Senior    | off | 3.7 | 7.5   | 17      | 16      | 18      |
| 12 | Senior    | on  | 3.0 | 6.5   | 10      | 11      | 12      |

## Cryptic names

→ `rename()`

## Hours in three columns

→ `pivot_longer()`

# The plan

1

## Clean

Give columns clear names

```
rename()
```

2

## Subset

Pick columns and rows

```
select()
```

```
filter()
```

```
slice()
```

3

## Create

Make new columns

```
mutate()
```

4

## Reshape

Wide to long

```
pivot_longer()
```

5

## Summarize

Stats by group

```
group_by()
```

```
summarize()
```

```
mutate()
```

```
ungroup()
```

# rename()

```
students <- raw |>
  rename(id = ID, year = Yr, housing = Hsg, gpa = GPA, sleep_hours = Sleep,
         study_hours_wk1 = wk1_hrs, study_hours_wk2 = wk2_hrs, study_hours_wk3 = wk3_hrs)
```

| id<br>(was ID) | year<br>(was Yr) | housing<br>(was Hsg) | gpa<br>(was GPA) | sleep_hours<br>(was Sleep) | study_hours_wk1<br>(was wk1_hrs) | study_hours_wk2<br>(was wk2_hrs) | study_hours_wk3<br>(was wk3_hrs) |
|----------------|------------------|----------------------|------------------|----------------------------|----------------------------------|----------------------------------|----------------------------------|
| 1              | Freshman         | on                   | 3.2              | 7.5                        | 10                               | 12                               | 11                               |
| 2              | Freshman         | off                  | 3.6              | 8.0                        | 14                               | 15                               | 16                               |
| 3              | Freshman         | on                   | 2.9              | 6.5                        | 8                                | 9                                | 10                               |
| 4              | Sophomore        | on                   | 3.4              | 7.0                        | 12                               | 13                               | 14                               |
| 5              | Sophomore        | off                  | 3.8              | 7.5                        | 16                               | 18                               | 17                               |
| 6              | Sophomore        | on                   | 3.0              | 6.0                        | 9                                | 10                               | 11                               |

Amber headers were renamed; the rows underneath are unchanged (first six rows shown).

**Your turn:** from raw, rename only Hsg to housing\_type (Rmd exercise 1.1).

# select()

```
students |> select(id, year, gpa)           # by name (highlighted below)
students |> select(id, starts_with("hours")) # by pattern
students |> select(-housing)                # drop a column with a minus sign
students |> select(id, gpa:hours_wk1)       # a range of neighboring columns
```

| id | year      | housing | gpa | sleep_hours | study_hours_wk1 | study_hours_wk2 | study_hours_wk3 |
|----|-----------|---------|-----|-------------|-----------------|-----------------|-----------------|
| 1  | Freshman  | on      | 3.2 | 7.5         | 10              | 12              | 11              |
| 2  | Freshman  | off     | 3.6 | 8.0         | 14              | 15              | 16              |
| 3  | Freshman  | on      | 2.9 | 6.5         | 8               | 9               | 10              |
| 4  | Sophomore | on      | 3.4 | 7.0         | 12              | 13              | 14              |
| 5  | Sophomore | off     | 3.8 | 7.5         | 16              | 18              | 17              |
| 6  | Sophomore | on      | 3.0 | 6.0         | 9               | 10              | 11              |

☒ kept by select(id, year, gpa)      ☐ not selected

**Your turn:** select id, year, and housing; then every column except sleep\_hours.  
**Challenge:** use another function to select all columns that contain the word “hours”

# filter()

```
students |> filter(gpa >= 3.5)
students |> filter(year == "Senior", housing == "off")
students |> filter(sleep_hours < 7 | gpa > 3.8)
students |> filter(year %in% c("Junior", "Senior"))
```

filter() keeps rows where the test is TRUE.

A comma means AND

| means OR.

Use == (two equals signs) to test equality, and put text in quotes.

**Your turn:** keep students who sleep at least 7.5 hours; then on-campus students with gpa below 3.2; then students whose GPA is between 3.0 and 3.5.

| id | year      | housing | gpa | sleep_hours |
|----|-----------|---------|-----|-------------|
| 1  | Freshman  | on      | 3.2 | 7.5         |
| 2  | Freshman  | off     | 3.6 | 8.0         |
| 3  | Freshman  | on      | 2.9 | 6.5         |
| 4  | Sophomore | on      | 3.4 | 7.0         |
| 5  | Sophomore | off     | 3.8 | 7.5         |
| 6  | Sophomore | on      | 3.0 | 6.0         |
| 7  | Junior    | off     | 3.5 | 7.0         |
| 8  | Junior    | on      | 3.1 | 6.5         |
| 9  | Junior    | off     | 3.9 | 8.0         |
| 10 | Senior    | on      | 3.3 | 7.0         |
| 11 | Senior    | off     | 3.7 | 7.5         |
| 12 | Senior    | on      | 3.0 | 6.5         |

 kept

 removed

# slice()

```
students |> slice_head(n = 3)      # the first 3 rows
students |> slice_tail(n = 2)     # the last 2 rows
students |> slice_max(gpa, n = 3)  # the 3 highest GPAs
students |> slice_min(gpa, n = 3) # the 3 lowest GPAs
students |> slice_sample(n = 3)   # 3 random rows (changes each run)
```

`slice_head(n = 3)`

| id | year     | housing | gpa |
|----|----------|---------|-----|
| 1  | Freshman | on      | 3.2 |
| 2  | Freshman | off     | 3.6 |
| 3  | Freshman | on      | 2.9 |

`slice_max(gpa, n = 3)`

| id | year      | housing | gpa |
|----|-----------|---------|-----|
| 9  | Junior    | off     | 3.9 |
| 5  | Sophomore | off     | 3.8 |
| 11 | Senior    | off     | 3.7 |

`slice_min(gpa, n = 3)`

| id | year      | housing | gpa |
|----|-----------|---------|-----|
| 3  | Freshman  | on      | 2.9 |
| 6  | Sophomore | on      | 3.0 |
| 12 | Senior    | on      | 3.0 |

`slice()` works by position or rank; `filter()` works by a test.

**Your turn:** show the first 5 rows; find the 2 students who sleep the most; find the one who sleeps the least (Rmd 4.1 to 4.3).

# mutate()

```
students <- students |>
  mutate(total_hours = study_hours_wk1 +
         study_hours_wk2 + study_hours_wk3,
         avg_hours = total_hours / 3
  )
```

`mutate(new_column = calculation)` adds a column and keeps every row. You can use a column you just created (`total_hours`) in the very next line.

**Your turn:** create `sleep_gap` (8 minus `sleep_hours`), then `gpa_pct` (`gpa / 4 * 100`).

| id | study_hours_wk1 | study_hours_wk2 | study_hours_wk3 | total_hours | avg_hours |
|----|-----------------|-----------------|-----------------|-------------|-----------|
| 1  | 10              | 12              | 11              | 33          | 11.0      |
| 2  | 14              | 15              | 16              | 45          | 15.0      |
| 3  | 8               | 9               | 10              | 27          | 9.0       |
| 4  | 12              | 13              | 14              | 39          | 13.0      |
| 5  | 16              | 18              | 17              | 51          | 17.0      |
| 6  | 9               | 10              | 11              | 30          | 10.0      |
| 7  | 15              | 14              | 16              | 45          | 15.0      |
| 8  | 11              | 12              | 12              | 35          | 11.7      |
| 9  | 18              | 17              | 19              | 54          | 18.0      |
| 10 | 13              | 14              | 15              | 42          | 14.0      |
| 11 | 17              | 16              | 18              | 51          | 17.0      |
| 12 | 10              | 11              | 12              | 33          | 11.0      |

 new columns

# pivot\_longer()

| id | year     | hours_wk1 | hours_wk2 | hours_wk3 |
|----|----------|-----------|-----------|-----------|
| 1  | Freshman | 10        | 12        | 11        |
| 2  | Freshman | 14        | 15        | 16        |
| 3  | Freshman | 8         | 9         | 10        |



| id | year     | week | hours |
|----|----------|------|-------|
| 1  | Freshman | 1    | 10    |
| 1  | Freshman | 2    | 12    |
| 1  | Freshman | 3    | 11    |
| 2  | Freshman | 1    | 14    |
| 2  | Freshman | 2    | 15    |
| 2  | Freshman | 3    | 16    |
| 3  | Freshman | 1    | 8     |
| 3  | Freshman | 2    | 9     |
| 3  | Freshman | 3    | 10    |

```
hours_long <- students |>
  select(id, year, starts_with("hours_wk")) |>
  pivot_longer(
    cols           = starts_with("hours_wk"),
    names_to       = "week",
    names_prefix   = "hours_wk",
    names_transform = as.integer,
    values_to      = "hours"
  )
```

Three hours columns become two (week and hours): 12 students × 3 weeks = 36 rows. The first 9 are shown.

**Your turn:** re-run it without names\_prefix and see what week\_label looks like (Rmd 6.1).

 reshaped columns       same color = same student

# group\_by() and summarize()

| id | year      | total_hours |
|----|-----------|-------------|
| 1  | Freshman  | 33          |
| 2  | Freshman  | 45          |
| 3  | Freshman  | 27          |
| 4  | Sophomore | 39          |
| 5  | Sophomore | 51          |
| 6  | Sophomore | 30          |
| 7  | Junior    | 45          |
| 8  | Junior    | 35          |
| 9  | Junior    | 54          |
| 10 | Senior    | 42          |
| 11 | Senior    | 51          |
| 12 | Senior    | 33          |

 Freshman     Sophomore  
 Junior     Senior



| year      | n | mean_total | mean_gpa |
|-----------|---|------------|----------|
| Freshman  | 3 | 35.0       | 3.23     |
| Junior    | 3 | 44.7       | 3.50     |
| Senior    | 3 | 42.0       | 3.33     |
| Sophomore | 3 | 40.0       | 3.40     |

```
students |>
  group_by(year) |>
  summarize(
    n          = n(),
    mean_total = mean(total_hours),
    mean_gpa   = mean(gpa)
  )
```

**Your turn:** mean GPA for on- vs. off-campus students; number of students in each year who live in off vs on campus; max sleep in each year.

# group by two variables

```
hours_long |>
  group_by(year, week) |>
  summarize(
    mean_hours = mean(hours),
    .groups     = "drop"
  )
```

Two grouping variables give one row per combination: 4 years × 3 weeks = 12 rows. `.groups = "drop"` removes the grouping from the result. Grouping by week is possible because week is now a column.

**Your turn:** using `hours_long`, find the mean hours for each week, all students together (Rmd 7.4).

| year      | week | mean_hours |
|-----------|------|------------|
| Freshman  | 1    | 10.7       |
| Freshman  | 2    | 12.0       |
| Freshman  | 3    | 12.3       |
| Junior    | 1    | 14.7       |
| Junior    | 2    | 14.3       |
| Junior    | 3    | 15.7       |
| Senior    | 1    | 13.3       |
| Senior    | 2    | 13.7       |
| Senior    | 3    | 15.0       |
| Sophomore | 1    | 12.3       |
| Sophomore | 2    | 13.7       |
| Sophomore | 3    | 14.0       |

 Freshman    Sophomore    Junior    Senior

# Putting it together

```
hours_summary <- raw |>
  rename(id = ID, year = Yr, housing = Hsg,
         gpa = GPA, sleep_hours = Sleep,
         hours_wk1 = wk1_hrs, hours_wk2 = wk2_hrs,
         hours_wk3 = wk3_hrs) |>
  mutate(total_hours = hours_wk1 + hours_wk2 +
         hours_wk3) |>
  filter(total_hours >= 40) |>
  select(id, year, starts_with("hours_wk")) |>
  pivot_longer(
    cols           = starts_with("hours_wk"),
    names_to       = "week",
    names_prefix   = "hours_wk",
    names_transform = as.integer,
    values_to      = "hours"
  ) |>
  group_by(year, week) |>
  summarize(mean_hours = mean(hours),
            .groups = "drop")
```

**hours\_summary**

| year      | week | mean_hours |
|-----------|------|------------|
| Freshman  | 1    | 14.0       |
| Freshman  | 2    | 15.0       |
| Freshman  | 3    | 16.0       |
| Junior    | 1    | 16.5       |
| Junior    | 2    | 15.5       |
| Junior    | 3    | 17.5       |
| Senior    | 1    | 15.0       |
| Senior    | 2    | 15.0       |
| Senior    | 3    | 16.5       |
| Sophomore | 1    | 16.0       |
| Sophomore | 2    | 18.0       |
| Sophomore | 3    | 17.0       |

Mean weekly hours for the six students who studied 40+ hours in total.

**Why do I need to know how to tidy when <insert favorite AI> can do it for me?**

Tidying data requires a lot of decisions that you should be in charge of making!

Examples:

- Which participants/rows do I exclude?
- Should this composite be summed or averaged?
- Does this variable need to be centered?

# Tips & Prompts to use AI as a collaborative tool

## **Tips/Guidelines:**

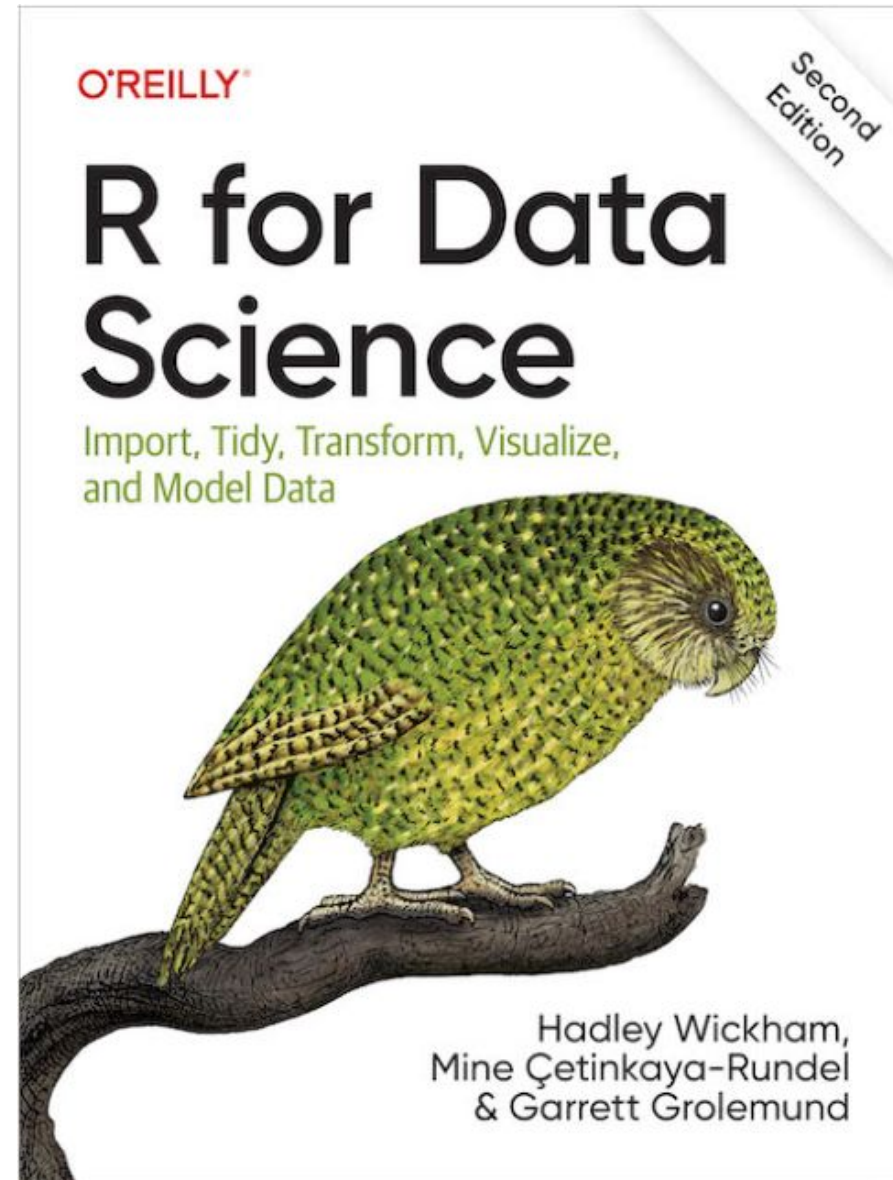
- Give clear and detailed instructions
- Verify everything
- Be specific in your prompts
- Specify hypotheses, analytic strategies, model content
- Ask follow-up questions, ask for sources

## **Example prompts:**

- *I want to tidy this dataset so that my main variables of interest (X, Y, Z) are easy to analyze. Using tidyverse functions, what should I do?*

# Appendix

**For more on tidying  
data in R, I highly  
recommend this  
book!**



# Function cheat sheet

**rename()**

change column names

**distinct()**

drop duplicate rows

**drop\_na()**

remove rows with NA

**select()**

pick columns

**filter()**

pick rows by condition

**slice()**

pick rows by position or rank

**if\_else()**

two-way recode in mutate()

**case\_match()**

recode specific values

**rowSums(pick())**

add across columns

**mdy(), year()**

parse and split dates

**group\_by()**

compute within groups

**summarize()**

one row per group

**ungroup()**

remove grouping

**mutate()**

create or change columns

**pivot\_longer()**

wide to long

**pivot\_wider()**

long to wide

# Setup code for advanced data wrangling

```
library(tidyverse)

raw <- data.frame(
  ID          = c(1, 2, 3, 4, 5, 6, 6, 7, 8, 9, 10, 11),
  Cond        = c(1, 2, 3, 1, 2, 3, 3, 1, 2, 3, 1, 2),
  Age         = c(19, 24, 31, 22, NA, 27, 27, 35, 29, 20, 41, 23),
  Survey_Date = c("01/12/2026", "01/12/2026", "02/03/2026", "02/09/2026",
                  "02/10/2026", "03/02/2026", "03/02/2026", "03/02/2026",
                  "03/16/2026", "04/06/2026", "04/20/2026", "04/21/2026"),
  q1          = c(2, 4, 3, 5, 2, 4, 4, 3, 1, 4, 2, 3),
  q2          = c(3, 4, 2, 4, 2, 5, 5, 3, 2, 3, 2, 4),
  q3          = c(2, 5, 3, 4, 1, 4, 4, 2, 2, 4, 3, 3),
  q4          = c(3, 4, NA, 5, 2, 4, 4, 3, 1, 4, 2, 3),
  mood_wk1    = c(5, 4, 6, 3, 5, 4, 4, 6, 5, 3, 7, 4),
  mood_wk2    = c(6, 6, 7, NA, 6, 5, 5, 6, 7, 5, 7, 6),
  mood_wk3    = c(6, 7, 8, 4, 7, 6, 6, NA, 8, 6, 7, 8)
)
```

# distinct()

```
nrow(dat)           # 12 rows
dat <- dat |> distinct()
nrow(dat)           # 11 rows

dat |> distinct(condition_code)
```

`distinct()` keeps the first copy of each identical row (the rose row is dropped). With column names, it returns the unique values of those columns instead.

| id | condition_code | age | survey_date | q1 | q2 | q3 | q4 |
|----|----------------|-----|-------------|----|----|----|----|
| 1  | 1              | 19  | 01/12/2026  | 2  | 3  | 2  | 3  |
| 2  | 2              | 24  | 01/12/2026  | 4  | 4  | 5  | 4  |
| 3  | 3              | 31  | 02/03/2026  | 3  | 2  | 3  | NA |
| 4  | 1              | 22  | 02/09/2026  | 5  | 4  | 4  | 5  |
| 5  | 2              | NA  | 02/10/2026  | 2  | 2  | 1  | 2  |
| 6  | 3              | 27  | 03/02/2026  | 4  | 5  | 4  | 4  |
| 6  | 3              | 27  | 03/02/2026  | 4  | 5  | 4  | 4  |
| 7  | 1              | 35  | 03/02/2026  | 3  | 3  | 2  | 3  |
| 8  | 2              | 29  | 03/16/2026  | 1  | 2  | 2  | 1  |
| 9  | 3              | 20  | 04/06/2026  | 4  | 3  | 4  | 4  |
| 10 | 1              | 41  | 04/20/2026  | 2  | 2  | 3  | 2  |
| 11 | 2              | 23  | 04/21/2026  | 3  | 4  | 3  | 3  |

 duplicate removed by `distinct()`

# drop\_na()

```
dat |> drop_na(age)
# NAs in age only
```

```
dat |> drop_na()
# NAs in any column
```

`drop_na(age)` removes only id 5.

`drop_na()` removes every row with an NA anywhere: ids 3, 4, 5, and 7.



Before unilaterally omitting all missing data, be sure to think about what the missingness tells you

| id | age | q4 | mood_wk1 | mood_wk2 | mood_wk3 |
|----|-----|----|----------|----------|----------|
| 1  | 19  | 3  | 5        | 6        | 6        |
| 2  | 24  | 4  | 4        | 6        | 7        |
| 3  | 31  | NA | 6        | 7        | 8        |
| 4  | 22  | 5  | 3        | NA       | 4        |
| 5  | NA  | 2  | 5        | 6        | 7        |
| 6  | 27  | 4  | 4        | 5        | 6        |
| 7  | 35  | 3  | 6        | 6        | NA       |
| 8  | 29  | 1  | 5        | 7        | 8        |
| 9  | 20  | 4  | 3        | 5        | 6        |
| 10 | 41  | 2  | 7        | 7        | 7        |
| 11 | 23  | 3  | 4        | 6        | 8        |

 missing value (NA)

 removed by `drop_na()`

# split a date with lubridate

```
dat <- dat |>
  mutate(
    survey_date = mdy(survey_date),
    survey_year = year(survey_date),
    survey_month = month(survey_date, label = TRUE),
    survey_day = day(survey_date)
  )
```

mdy() turns text in month/day/year order into a real date. year(), month(), and day() then pull out each part. label = TRUE gives month names (Jan, Feb, ...).

Before :

|    | Survey_Date |
|----|-------------|
| 1  | 01/12/2026  |
| 2  | 01/12/2026  |
| 3  | 02/03/2026  |
| 4  | 02/09/2026  |
| 5  | 02/10/2026  |
| 6  | 03/02/2026  |
| 7  | 03/02/2026  |
| 8  | 03/02/2026  |
| 9  | 03/16/2026  |
| 10 | 04/06/2026  |
| 11 | 04/20/2026  |

# split a date with lubridate

```
dat <- dat |>
  mutate(
    survey_date = mdy(survey_date),
    survey_year = year(survey_date),
    survey_month = month(survey_date, label = TRUE),
    survey_day = day(survey_date)
  )
```

mdy() turns text in month/day/year order into a real date. year(), month(), and day() then pull out each part. label = TRUE gives month names (Jan, Feb, ...).

After:

| id | survey_date<br>(was text) | year | month | day |
|----|---------------------------|------|-------|-----|
| 1  | 2026-01-12                | 2026 | Jan   | 12  |
| 2  | 2026-01-12                | 2026 | Jan   | 12  |
| 3  | 2026-02-03                | 2026 | Feb   | 3   |
| 4  | 2026-02-09                | 2026 | Feb   | 9   |
| 5  | 2026-02-10                | 2026 | Feb   | 10  |
| 6  | 2026-03-02                | 2026 | Mar   | 2   |
| 7  | 2026-03-02                | 2026 | Mar   | 2   |
| 8  | 2026-03-16                | 2026 | Mar   | 16  |
| 9  | 2026-04-06                | 2026 | Apr   | 6   |
| 10 | 2026-04-20                | 2026 | Apr   | 20  |
| 11 | 2026-04-21                | 2026 | Apr   | 21  |

 parsed date     new column

# mutate() with if\_else()

```
dat <- dat |>
  mutate(
    age_group = if_else(
      age >= 30,
      "30 and over",
      "under 30",
      missing = "unknown"
    )
  )
```

if\_else(test, value if TRUE, value if FALSE). Without missing =, id 5 would get NA because its age is missing.

| id | age | age_group   |
|----|-----|-------------|
| 1  | 19  | under 30    |
| 2  | 24  | under 30    |
| 3  | 31  | 30 and over |
| 4  | 22  | under 30    |
| 5  | NA  | unknown     |
| 6  | 27  | under 30    |
| 7  | 35  | 30 and over |
| 8  | 29  | under 30    |
| 9  | 20  | under 30    |
| 10 | 41  | 30 and over |
| 11 | 23  | under 30    |

 new column     from missing =     NA input

# mutate() with case\_match()

```
dat <- dat |>
  mutate(
    condition = case_match(
      condition_code,
      1 ~ "control",
      2 ~ "mindfulness",
      3 ~ "exercise"
    )
  )
```

`case_match(column, old_value ~ new_value, ...)` recodes specific values. Add `.default =` to say what happens to any other value.

| id | condition_code | condition   |
|----|----------------|-------------|
| 1  | 1              | control     |
| 2  | 2              | mindfulness |
| 3  | 3              | exercise    |
| 4  | 1              | control     |
| 5  | 2              | mindfulness |
| 6  | 3              | exercise    |
| 7  | 1              | control     |
| 8  | 2              | mindfulness |
| 9  | 3              | exercise    |
| 10 | 1              | control     |
| 11 | 2              | mindfulness |

 new column

# mutate() with rowSums(pick())

```
dat <- dat |>
  mutate(
    anxiety_total = rowSums(pick(q1:q4))
  )
```

```
# ignore NAs instead:
rowSums(pick(q1:q4), na.rm = TRUE)
```

pick() selects columns inside mutate(); rowSums() adds across each row. Id 3 has a missing q4, so the total is NA. With na.rm = TRUE it would be 8.

| id | q1 | q2 | q3 | q4 | anxiety_total |
|----|----|----|----|----|---------------|
| 1  | 2  | 3  | 2  | 3  | 10            |
| 2  | 4  | 4  | 5  | 4  | 17            |
| 3  | 3  | 2  | 3  | NA | NA            |
| 4  | 5  | 4  | 4  | 5  | 18            |
| 5  | 2  | 2  | 1  | 2  | 7             |
| 6  | 4  | 5  | 4  | 4  | 17            |
| 7  | 3  | 3  | 2  | 3  | 11            |
| 8  | 1  | 2  | 2  | 1  | 6             |
| 9  | 4  | 3  | 4  | 4  | 15            |
| 10 | 2  | 2  | 3  | 2  | 9             |
| 11 | 3  | 4  | 3  | 3  | 13            |

 new column

 NA