Cleaning and tidying data using `janitor`

Emily Malcolm-White

`janitor` helps clean data quickly and consistently.

```
library(tidyverse)
library(janitor)
```

Here is some messy data to start:

```
data <- read.csv("data/messy_student_survey.csv")
```

Clean column names with `clean_names()`

clean_names():

- Converts all column names to snake_case
- Removes special characters and extra spaces

```
data <- data %>%
  clean_names()

names(data)
```

```
[1] "student_name"            "year_level"
[3] "major"                   "gpa"
[5] "favorite_color"          "likes_r"
[7] "hometown"                "time_spent_studying_hrs"
[9] "x"
```

Remove empty rows or columns with remove_empty()

Excel files often include blank rows or columns that sneak into your dataset. Cleaning them helps avoid weird bugs. This doesn't happen so often with .csv files.

```
data <- data %>%
  remove_empty("rows") %>%
  remove_empty("cols")
```

Missing data NA

Be explicit about your choices with NAs Don't just drop them silently—use `drop_na()` or `replace_na()`. If you choose to drop missing rows, you need to explain why.

Identify duplicate rows with `get_dupes()`

```
data %>%
  get_dupes()
```

```
   student_name  year_level       major  gpa favorite_color   likes_r hometown
1                                        NA
2                                        NA
3                                        NA
4                                        NA
5                                        NA
6   Student_105 Fourth Year     Biology 3.93            Red        No    Other
7   Student_105 Fourth Year     Biology 3.93            Red        No    Other
8   Student_125 Fourth Year  Statistics 2.33           None Sometimes       LA
9   Student_125 Fourth Year  Statistics 2.33           None Sometimes       LA
10  Student_156  Third Year Undeclared  2.39           Blue       Yes New York
11  Student_156  Third Year Undeclared  2.39           Blue       Yes New York
12  Student_362  First Year     Biology 3.23           Blue Sometimes       LA
13  Student_362  First Year     Biology 3.23           Blue Sometimes       LA
14  Student_375  Third Year     Biology 3.35           None Sometimes       LA
15  Student_375  Third Year     Biology 3.35           None Sometimes       LA
16  Student_378  First Year  Statistics 2.58            Red        No  Chicago
17  Student_378  First Year  Statistics 2.58            Red        No  Chicago
18  Student_395 Second Year Undeclared  2.13            Red Sometimes New York
```

```
19  Student_395 Second Year Undeclared  2.13           Red Sometimes New York
20  Student_451  First Year     Biology 3.37         Green       Yes New York
21  Student_451  First Year     Biology 3.37         Green       Yes New York
22   Student_69 Fourth Year     Biology 2.37        Purple Sometimes  Chicago
23   Student_69 Fourth Year     Biology 2.37        Purple Sometimes  Chicago
24   Student_74 Fourth Year  Statistics 3.70         Green        No  Chicago
25   Student_74 Fourth Year  Statistics 3.70         Green        No  Chicago
   time_spent_studying_hrs dupe_count
1                       NA          5
2                       NA          5
3                       NA          5
4                       NA          5
5                       NA          5
6                       10          2
7                       10          2
8                       11          2
9                       11          2
10                       9          2
11                       9          2
12                      11          2
13                      11          2
14                       8          2
15                       8          2
16                      10          2
17                      10          2
18                      16          2
19                      16          2
20                      10          2
21                      10          2
22                      13          2
23                      13          2
24                       8          2
25                       8          2
```

Note

Function	Purpose	Output
`get_dupes()`	Find and display duplicated rows	Only the duplicated rows (plus a `dupe_count`)

`distinct()`	Remove duplicates (keep unique rows only)	A cleaned dataset with no duplicates

In this case, it looks appropriate to drop all the duplicated rows (since the student_id is being repeated)

```
data <- data %>%
  distinct()
```

tably

```
data %>%
  tabyl(year_level) %>%
  adorn_totals("row") %>%
  adorn_percentages("col") %>%
  adorn_pct_formatting()
```

```
 year_level   n percent
            0.0    0.2%
 First Year 0.3   26.3%
Fourth Year 0.2   24.0%
Second Year 0.2   24.8%
 Third Year 0.2   24.8%
      Total 1.0  100.0%
```

Tips janitor would approve of

- Clean first → Analyze second (clean_names(), remove_empty(), get_dupes() first)