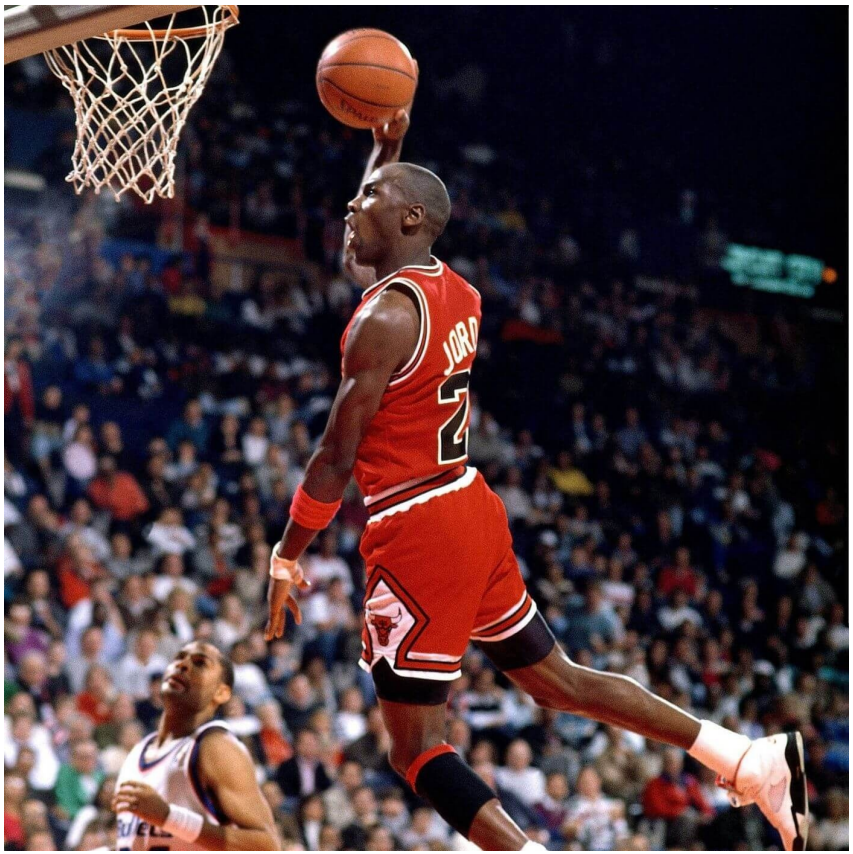
SMGT 430 Workshop:
How to Wrangle Data in R



plyr (2009–2015)

dplyr (2013–present)

Preliminaries

First, download `data.csv`. Then:

If you want to work in R locally on your laptop:

1. You need to have the following packages installed in R:

```
install.packages(
  c("data.table", "dplyr", "tidyr")
)
```

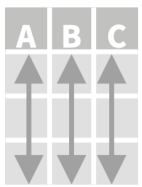
2. Download `exercises.R` move `data.csv` to working directory.

OR if you want to use the Google Colab notebook:

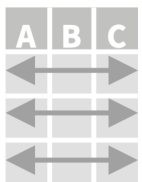
1. Runtime $\rightarrow$ Change runtime type $\rightarrow$ R (not Python 3)
2. Upload `data.csv` to your session.

"Tidy" data

dplyr functions work with pipes and expect **tidy data**. In tidy data:



&



Each **variable** is in its own **column**

Each **observation**, or **case**, is in its own **row**

pipes

x |> f(y)
becomes **f(x, y)**

https://rstudio.github.io/cheatsheets/data-transformation.pdf

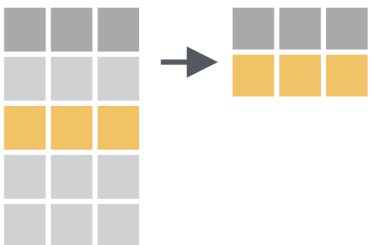
dplyr::select()

select(.data, …**)** Extract columns as a table.

mtcars |> select(mpg, wt)

https://rstudio.github.io/cheatsheets/data-transformation.pdf

dplyr::filter()



filter(.data, …, .preserve = FALSE**)** Extract rows that meet logical criteria.

mtcars |> filter(mpg > 20)

https://rstudio.github.io/cheatsheets/data-transformation.pdf

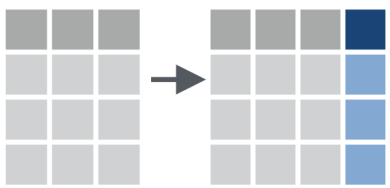
Exercise #1

Select the following columns and rename them using `snake_case`:

- `week`
- `winner_team`
- `winner_pts`
- `loser_team`
- `loser_pts`

dplyr::mutate()

Apply **vectorized functions** to columns. Vectorized functions take vectors as input and return vectors of the same length as output (see back).



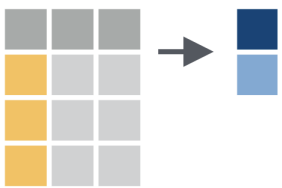
mutate(.data, …, .keep = "all", .before = NULL, .after = NULL**)** Compute new column(s). Also **add_column()**.

```
mtcars |> mutate(gpm = 1 / mpg)
```

https://rstudio.github.io/cheatsheets/data-transformation.pdf

dplyr::summarize()

Apply **summary functions** to columns to create a new table of summary statistics. Summary functions take vectors as input and return one value (see back).



summarize(.data, …**)**
Compute table of summaries.

```
mtcars |> summarize(avg = mean(mpg))
```

https://rstudio.github.io/cheatsheets/data-transformation.pdf

dplyr::group_by()

Use **group_by(**.data, …, .add = FALSE, .drop = TRUE**)** to create a "grouped" copy of a table grouped by columns in ... dplyr functions will manipulate each "group" separately and combine the results.

```
mtcars |>
  group_by(cyl) |>
  summarize(avg = mean(mpg))
```

https://rstudio.github.io/cheatsheets/data-transformation.pdf

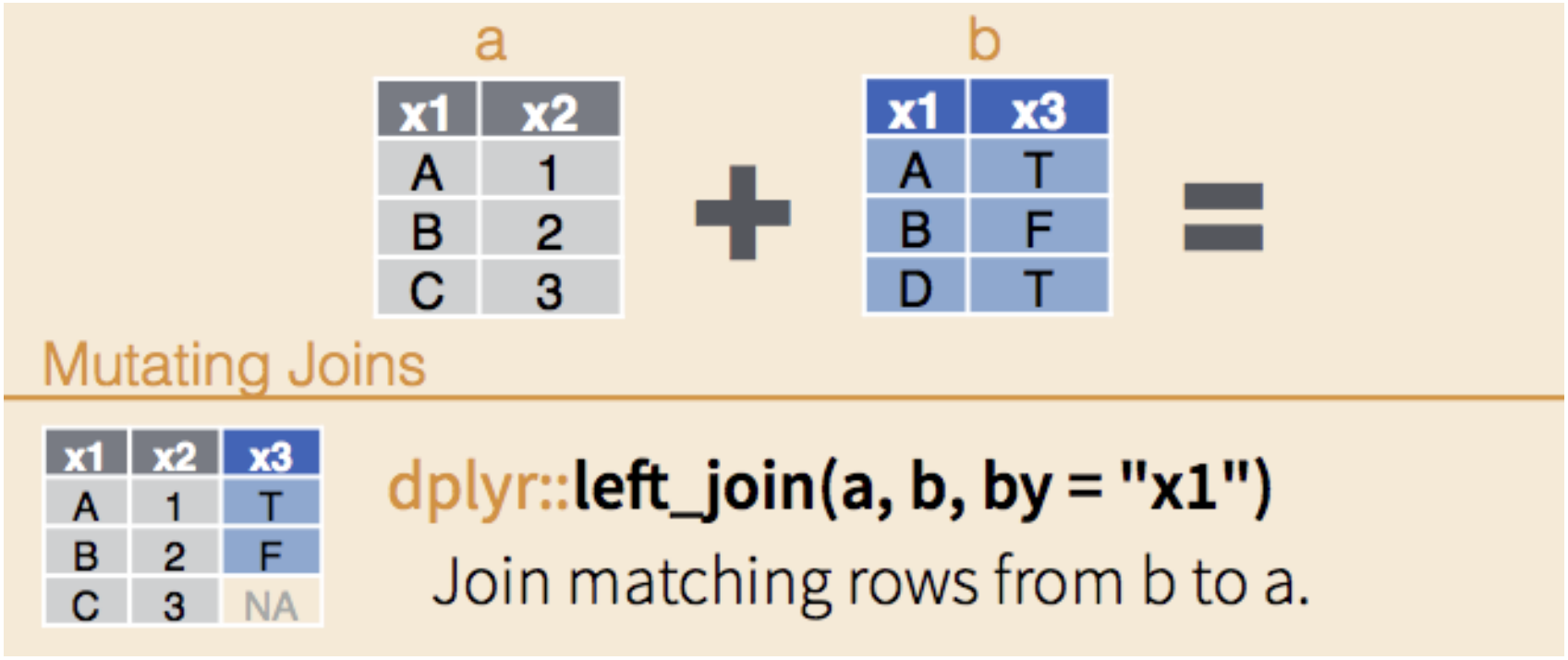
Exercise #2

Calculate total wins and point differential for the winners.

Exercise #3

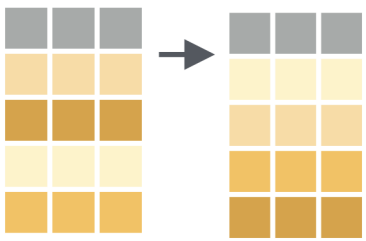
Calculate total wins and point differential for the losers.

dplyr::left_join()



https://bookdown.org/ansellbr/WEHI_tidyR_course_book/week3.html

dplyr::arrange()



arrange(.data, …, .by_group = FALSE**)** Order rows by values of a column or columns (low to high), use with **desc()** to order from high to low.

```
mtcars |> arrange(mpg)
mtcars |> arrange(desc(mpg))
```

https://rstudio.github.io/cheatsheets/data-transformation.pdf

Exercise #4

Create a sorted standings table by joining the winners table and the losers table.

tidyr::pivot_longer()

table4a

country	1999	2000
A	0.7K	2K
B	37K	80K
C	212K	213K

country	year	cases
A	1999	0.7K
B	1999	37K
C	1999	212K
A	2000	2K
B	2000	80K
C	2000	213K

pivot_longer(data, cols, names_to = "name", values_to = "value", values_drop_na = FALSE**)**

"Lengthen" data by collapsing several columns into two. Column names move to a new names_to column and values to a new values_to column.

pivot_longer(table4a, cols = 2:3, names_to ="year", values_to = "cases")

```
df |>
  tidyr::pivot_longer(
    cols = c("X1", "X2", "X3"),
    names_to = "V1",
    values_to = "V2",
  )
```

https://rstudio.github.io/cheatsheets/tidyr.pdf

Exercise #5

Try redoing Exercises #2–4 in a single pipe change starting with `tidyr::pivot_longer`.

Additional resources

R for Data Science:

`https://r4ds.hadley.nz/`

Tidyverse Cookbook:

`https://rstudio-education.github.io/tidyverse-cookbook/transform-tables.html`

dplyr cheatsheet:

`https://rstudio.github.io/cheatsheets/data-transformation.pdf`