Brief overview of statistical inference

Overview

A brief overview of statistical inference

Hypothesis tests

If there is time

- Confidence intervals
- Reshaping data using tidyr
- Joining data

Homework 4

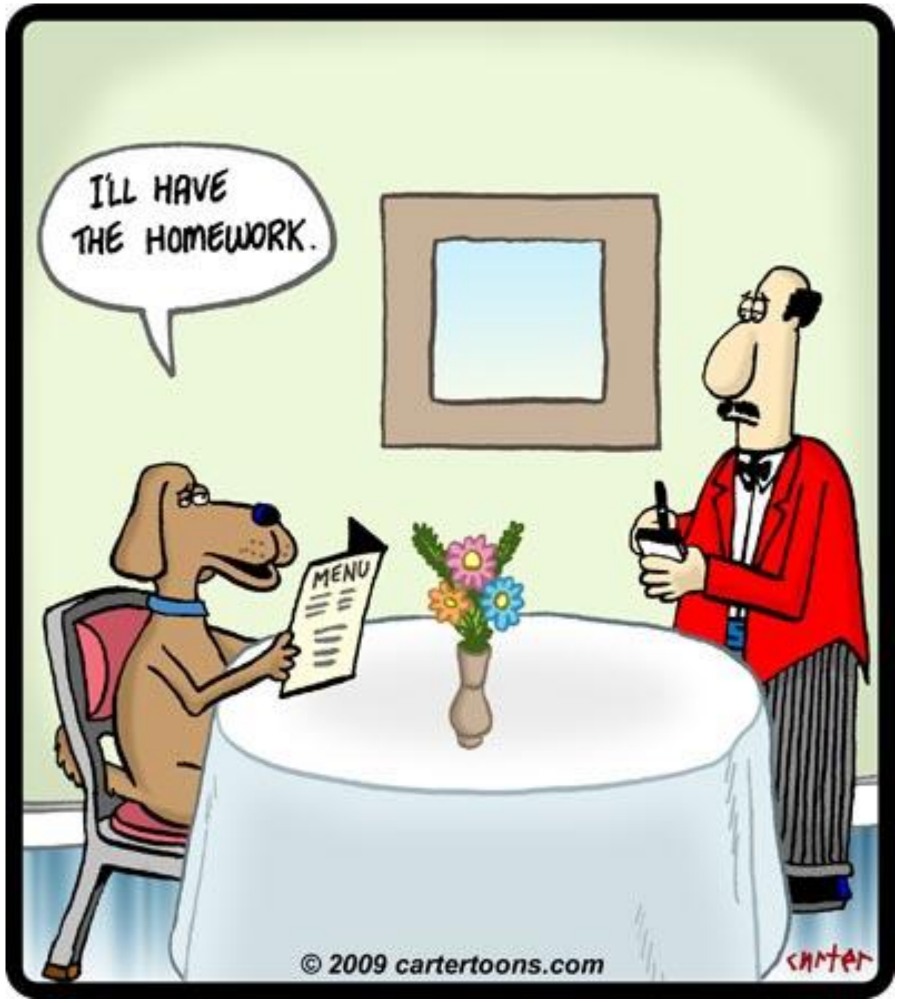
Create a scrollytelling webpage where you analyze a data set of your choosing

Homework 4 plan:

- Draft of homework due by 11pm on Monday July 28th
- Final version is due by 10pm on Wednesday July 30th

You will upload a quarto document and a hmtl document to Canvas

How did homework 3 go?



Class exam

Tuesday July 29th in person during regular class time

- Exam is on paper

If you have accommodations for extra time, stay in the classroom and keep working on the exam

Exam “cheat sheet”

You are allowed an exam “cheat sheet”

One page, single-side, that contains **only code and equations**

- No code comments allowed
- E.g., plot(x, y)

Cheat sheet must be on a regular 8.5 x 11 piece of paper

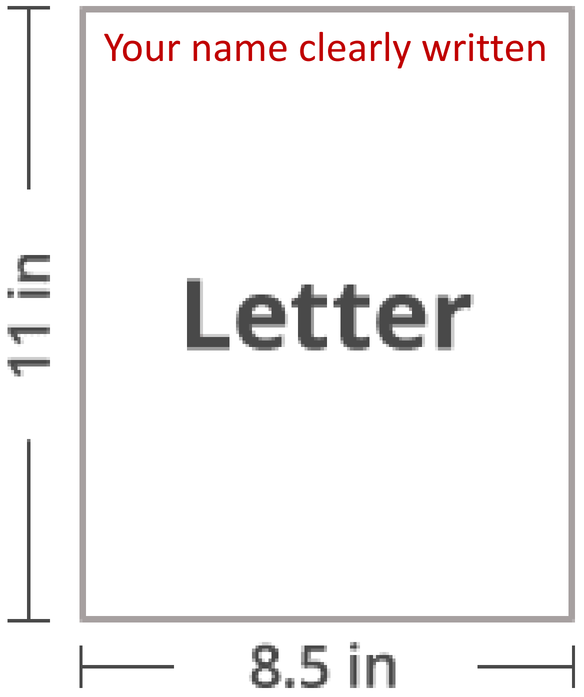
- Your name on the upper left side of the paper

I recommend making a typed list of all functions discussed in class and on the homework

- This will be useful beyond the exam

You must turn in your cheat sheet with the exam

- Failure to do so will result in a 20 point deduction



Questions?

Very quick review of Scrollytelling with Closeread

Review: Closeread components

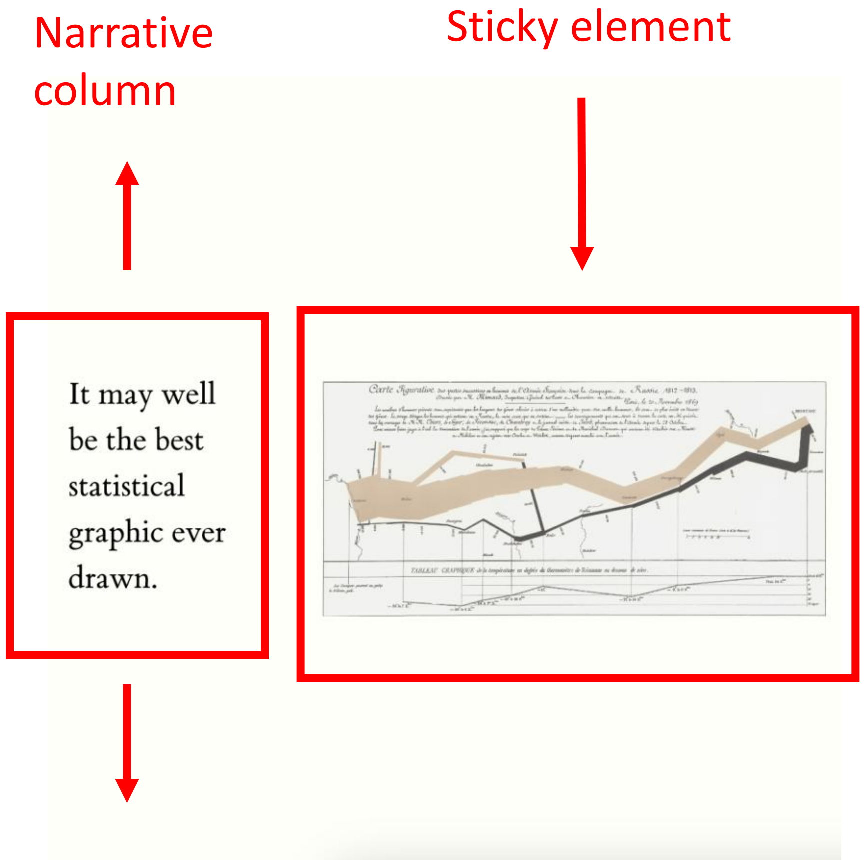
Closeread creates Scollytelilng webpages

Closeread documents are Quarto documents that have Closeread **sections**

Within each Closeread section:

1. "**sticky**" **elements** are tagged to appear in the main column of the Closeread section

2. "**narration"** text appears in the "narrative column" and triggers the appearance of the sticky element



It may well be the best statistical graphic ever drawn.

Review: Closeread example

```
::::{.cr-section}
```

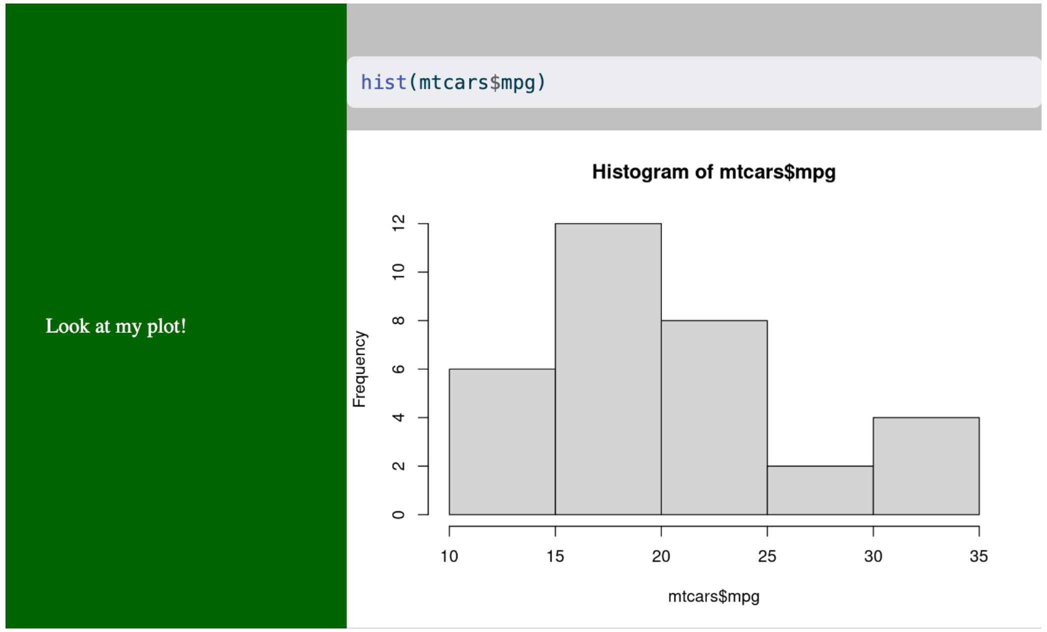
Closeread sections must have at least 3 colons.
They can have more colons to make sections stand out.

Look at my plot! @cr-myplot

Narration text refers to stickies

Stickies name must start with #cr-

~~~
:::{#cr-myplot, echo = TRUE}
    ```{r}
        hist(mtcars$mpg)
    ```
:::


::::
~~~


Look at my plot!

```
hist(mtcars$mpg)
```

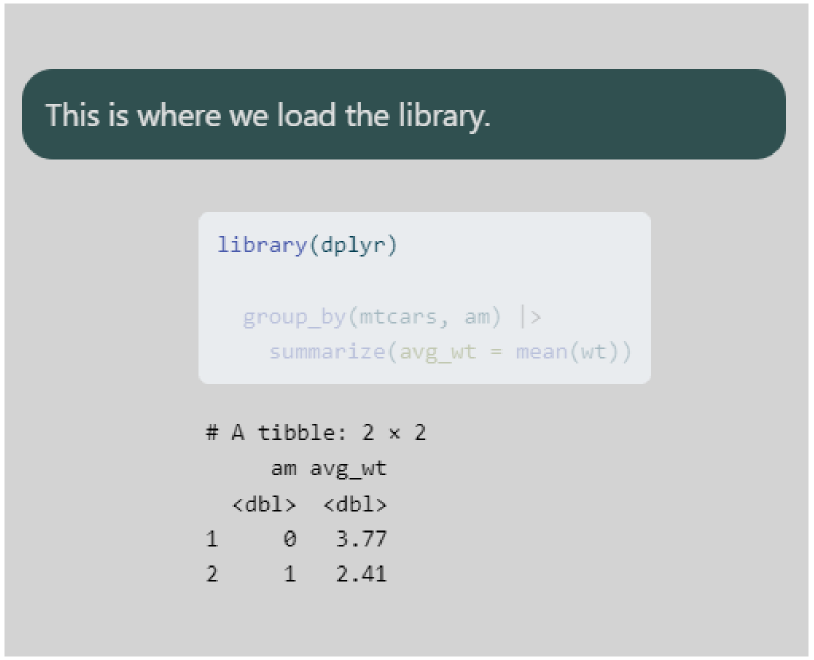
Review: Focus effects

We can add “focus effects” which can do the following:

- Scale, pan, or zoom in on images (or other elements)
- Highlight lines of text

Example:

~~~
This is where we load the library. [@cr-dplyr]{highlight="1"}
This calculates summary statistics. [@cr-dplyr]{highlight="2-3"}

:::{#cr-dplyr}
    ```{r}
    library(dplyr)
    group_by(mtcars, am) |>
        summarize(avg_wt = mean(wt))
    ```
:::
~~~


This is where we load the library.

```
library(dplyr)

group_by(mtcars, am) |>
  summarize(avg_wt = mean(wt))
```

```
# A tibble: 2 × 2
     am avg_wt
  <dbl>  <dbl>
1     0   3.77
2     1   2.41
```

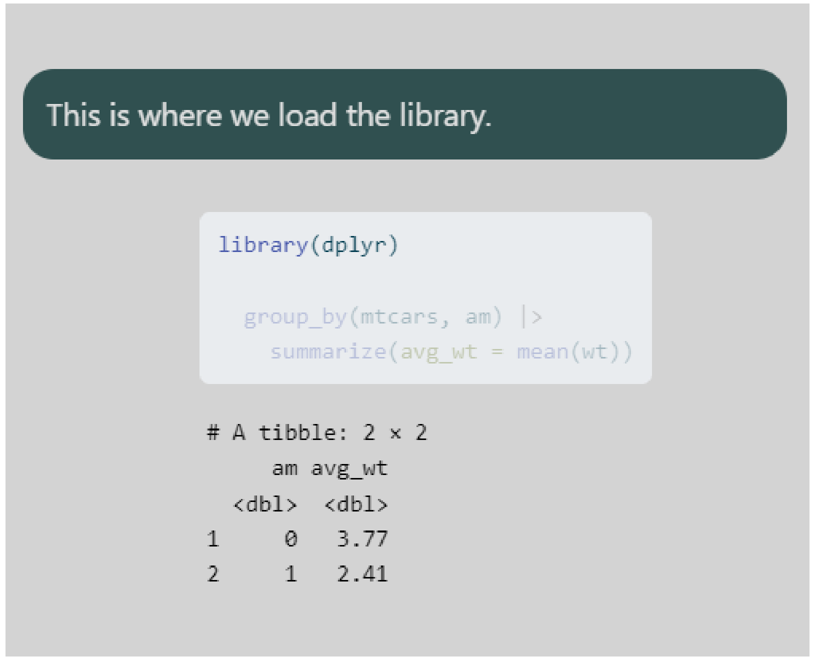
Review: Layouts

We can also change the layout of where the narration text and stickies appear by modifying the Quarto header

```
---
  format:
        closeread-html:
                cr-section:
                        layout: "overlay-center"
---
```

Options are:

- sidebar-left   sidebar-right
- overlay-right   overlay-right   overlay-center


This is where we load the library.

```
library(dplyr)

  group_by(mtcars, am) |>
    summarize(avg_wt = mean(wt))
```

```
# A tibble: 2 × 2
     am avg_wt
  <dbl>  <dbl>
1     0   3.77
2     1   2.41
```

Questions?

Also see [this app](#) which can help create Closeread documents

A brief introduction to Statistical inference

Statistical Inference

In **statistical inference** we use a sample of data to make claims about a larger population (or process)

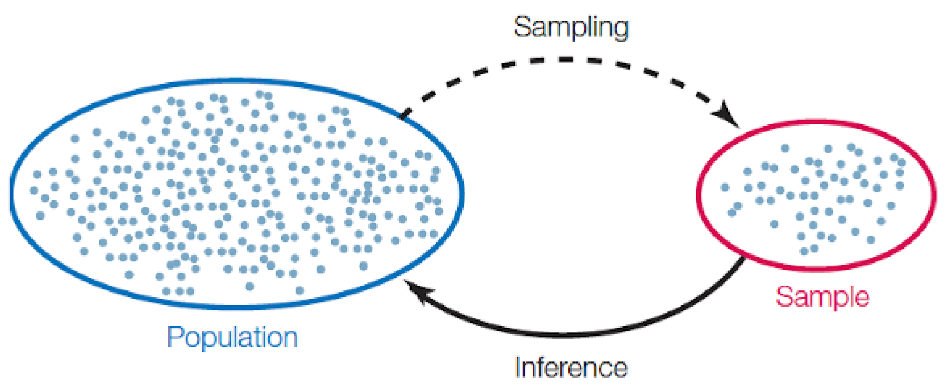
Examples:

<u>OkCupid data</u>

- **Sample**: 59,000 OkCuipd users from San Francisco
- **Population**: What is the population of interest?

<u>Billionaire data</u>:

- **Sample**: 2,781 billionaires in 2024
- **Population**: What is the population of interest?



Population: all individuals/objects of interest

Sample: A subset of the population

Terminology

A parameter is number associated with the <u>population/process</u>

- e.g., population mean height of all men μ

A **statistic** is number calculated from the <u>sample</u>

- e.g., the height of a sample of n = 59,000 men $\bar{x}$

A statistic can be used as an estimate of a parameter

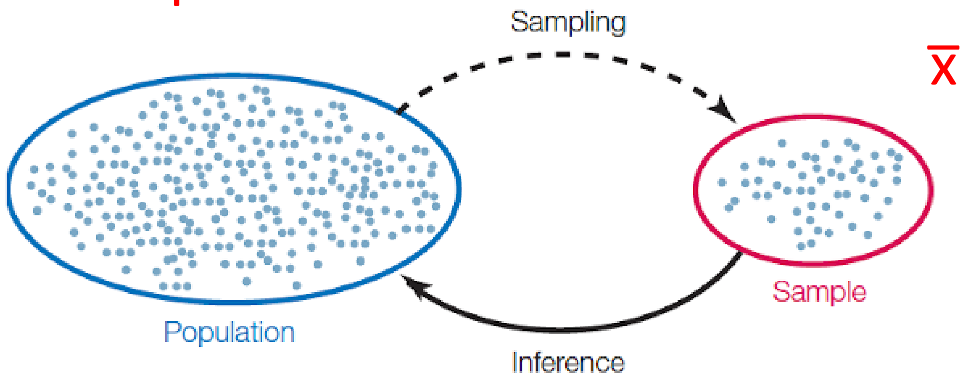
- A parameter is a single fixed value
- Statistics tend to vary from sample to sample

parameter

μ

statistic

$\bar{x}$



Example:

- Using the mean height of 59,000 men ($\bar{x}$) to estimate the mean height of all men (μ)

Examples of parameters and statistics

	Sample statistic	**Population parameter**	**Bechdel example**
Mean	$\bar{x}$	μ	`mean(profiles$height)` $\bar{x} = 68.3$
Proportion	$\hat{p}$	π	`prop.table(table(profiles$drinks))[6]` $\hat{p} = 0.697$
Correlation	r	ρ	`cor(min_temp, max_temp)` $r = 0.967$

Sampling

Simple random sample: each member in the population is equally likely to be in the sample

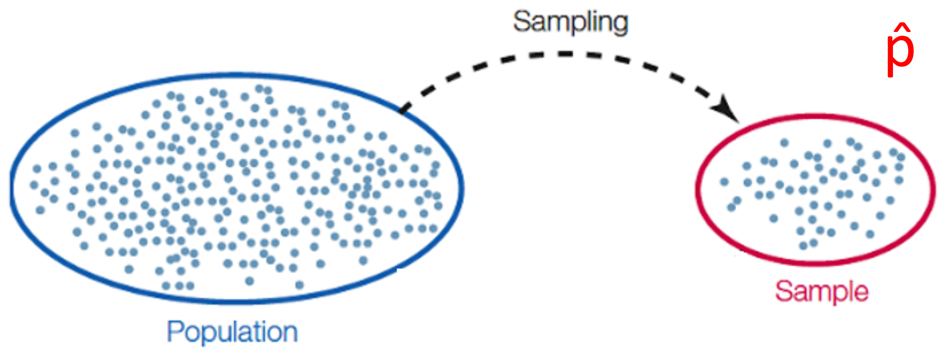
- Allows for generalizations to the population

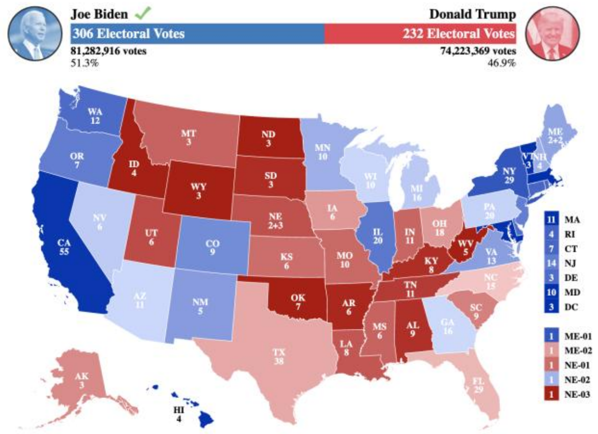
parameter

π

statistic

$\hat{p}$





Polls of 1,000 voters: $\hat{p}_{\text{Trump}}$

Vote on election day: π_{Trump}

Hypothesis tests

Statistical tests (hypothesis test)

A **statistical test** uses data from a sample to assess a claim about a population (parameter)

<u>Example 1</u>: The average body temperature of humans is 98.6˚

How can we write this using symbols?

- $\mu = 98.6$

Statistical tests (hypothesis test)

A **statistical test** uses data from a sample to assess a claim about a population (parameter)

<u>Example 2</u>: More than half of voters disapprove of Trump's job performance

How can we write this using symbols?

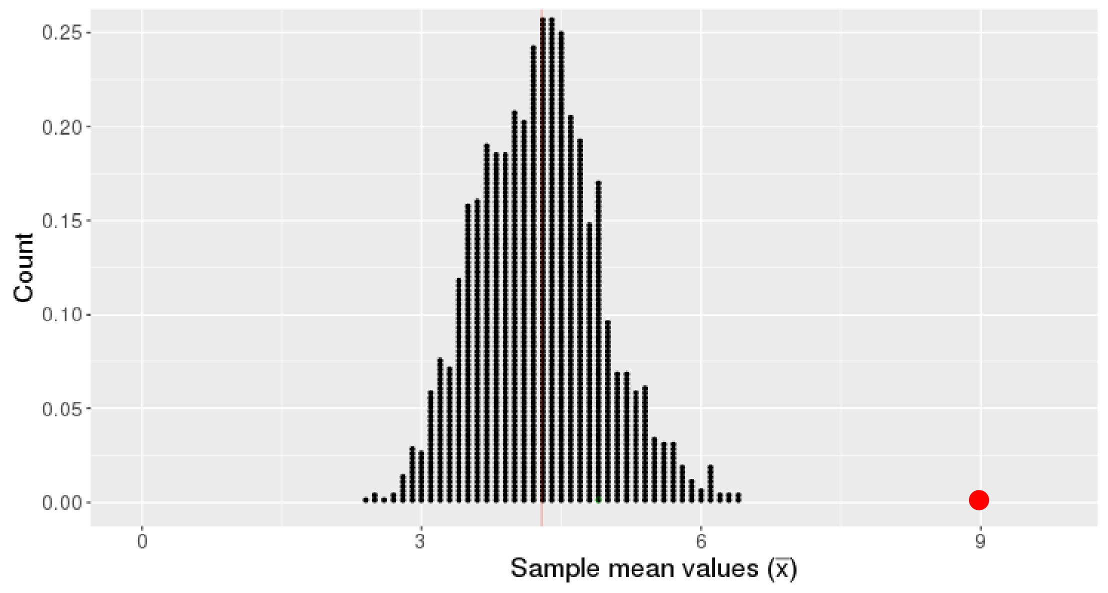
- $\pi_{\text{Disapprove}} > 0.5$

Basic hypothesis test logic

We start with a claim about a population parameter

- E.g., $\mu = 4$

This claim implies we should get a certain distribution of statistics



If our observed statistic is highly unlikely, we reject the claim

Example claims (hypotheses)

Let’s see if we can write the following claims (hypotheses) using symbols

Claim: 90% of Yale students graduate within four years

Claim: The average age of a Yale undergraduate is 20

Claim: 70% of Yale classrooms have fewer than 20 students in attendance

Testing claims (hypotheses)

Claim: 90% of Yale students graduate within four years

- H: $\pi = 0.90$
- To test this claim, we could randomly selected $n = 100$ Yale graduates
- If we found the proportion that graduated in 4 years is $\hat{p} = .85$, would we believe the claim?

Testing claims (hypotheses)

Claim: The average age of a Yale undergraduate is 20

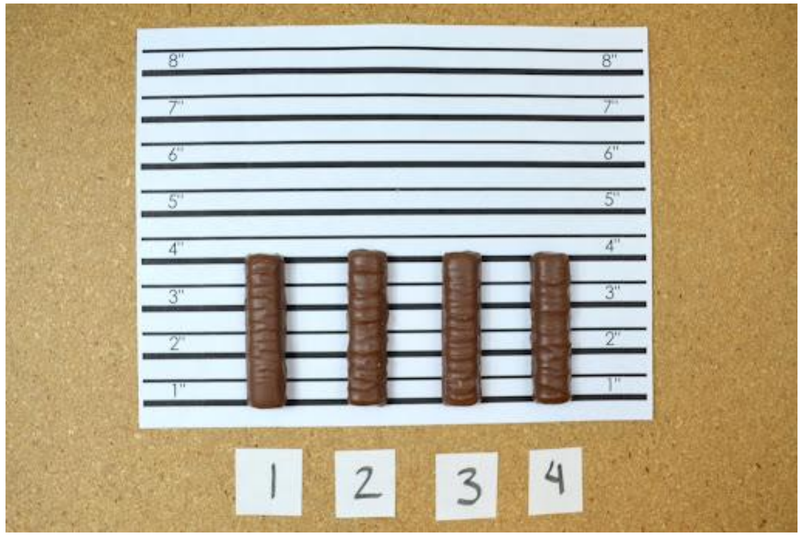
- H: $\mu = 20$
- To test this claim, we could randomly selected n = 50 Yale graduates
- If we found the average age of in our sample of students was $\bar{x} = 20.2$, would we believe the claim?

Visual hypothesis test

Visual hypothesis test

In visual hypothesis tests, we create data visualizations to try to assess whether particular relationships exist in our data.

- One way this is done through a visual lineup



[Which candy bar is 5th Avenue bar?]

Visual hypothesis test

Which plot shows the true relationship between a car's weight and the number of miles per gallon a car gets?

Let’s try it in R!

Hypothesis tests for correlation

Hypothesis tests for correlation

Is there a positive correlation between the number of pages in a book and the price of the book?

What is the population parameter and the statistic of interest?

Hypothesis testing for correlation

1. Write down the null and alternative in symbols and words

Null hypothesis:

- There is no correlation between book price and the number of pages

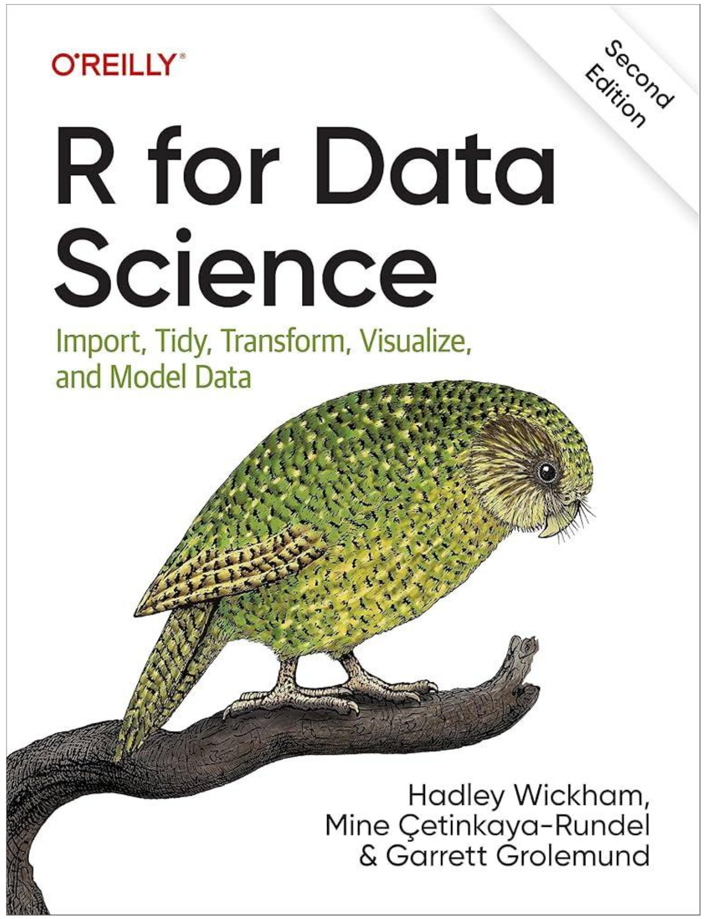
Alternative hypothesis:

- There is a positive correlation between book price and the number of pages

In symbols:

$H_0: \rho = 0$

$H_A: \rho > 0$


O'REILLY
Second Edition
R for Data Science
Import, Tidy, Transform, Visualize, and Model Data
Hadley Wickham, Mine Çetinkaya-Rundel & Garrett Grolemund

Has 548 pages

Significance tests for correlation

Let's look at the books from Amazon.com

Title	List.Price	NumPages
1,001 Facts that Will Scare the S#*t Out of You	12.95	304
21: Bringing Down the House	15.00	273
100 Best-Loved Poems	1.50	96
1421: The Year China Discovered America	15.99	672

```
amazon = read_csv("amazon.csv")
```

Let’s try this in R!

Step 2: What is the observed statistic?

- Also say whether you think you will be able to reject the null hypothesis based on a plot of your data

Step 3: Create a distribution of null statistics

- To start with: how we can create one statistic consistent with the null hypothesis?
  - Hint: think about shuffling the data

Step 4: What is the p-value that you get?

Step 5: What decision would you make?

Has 1012 pages

Confidence intervals

Interval estimate based on a margin of error

Null hypothesis tests tell us if a particular parameter value is implausible

- E.g., in the average height of OkCupid users is $\mu = 70''$

An **interval estimate** give a range of plausible values for a population parameter

Example: 42% of American approve of Trump's job performance, plus or minus 3%

How do we interpret this?

Says that the population parameter π lies somewhere between 39% to 45%

- i.e., if they sampled all Americans the true population proportion would be likely be in this range

Confidence Intervals

A **confidence interval** is an interval computed by a method that will contain the *parameter* a specified percent of times

- i.e., if the estimation were repeated many times, the interval will have the parameter x% of the time

The **confidence level** is the percent of all intervals that contain the parameter

Think ring toss…

Parameter exists in the ideal world

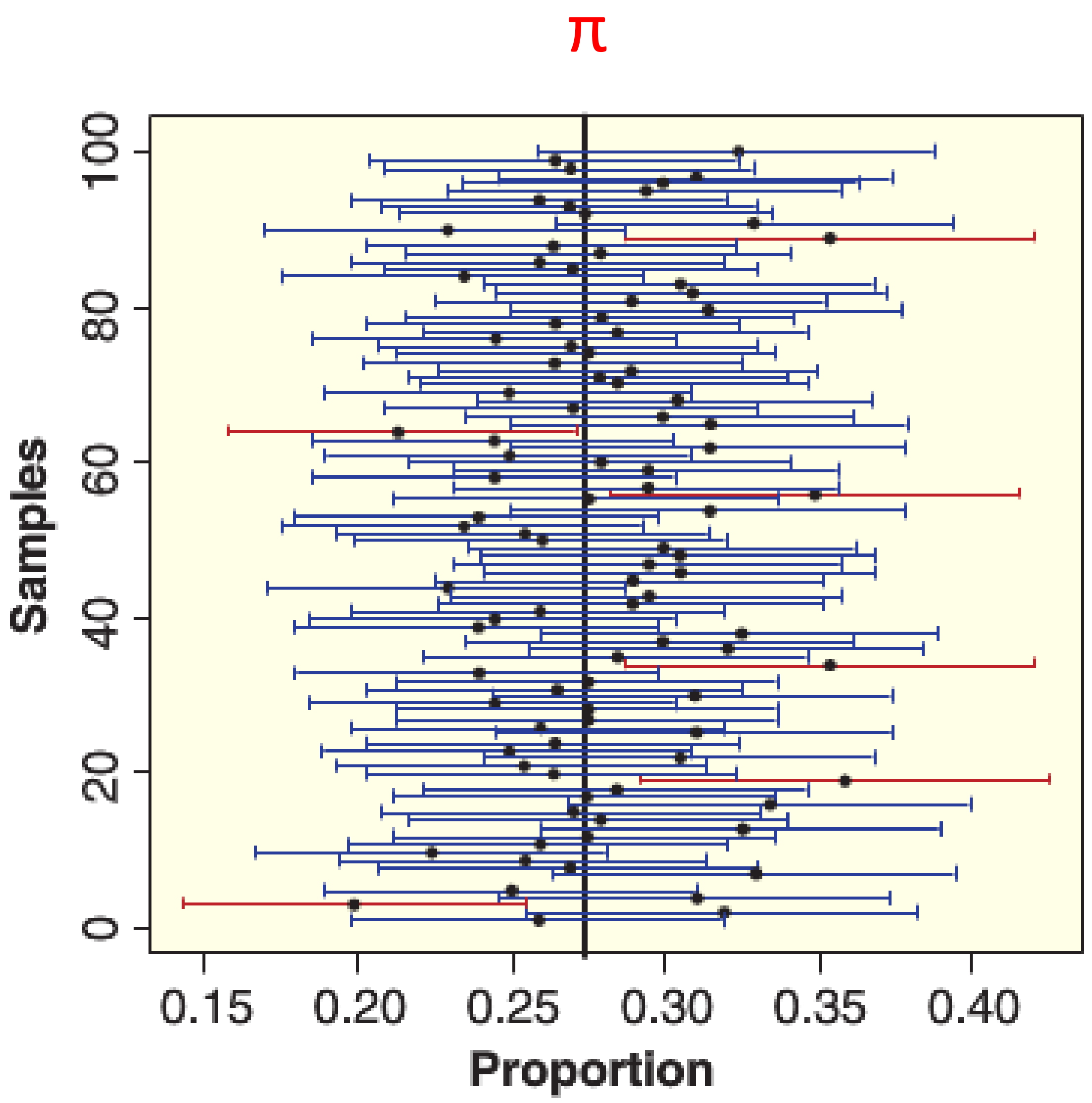
We toss intervals at it

95% of those intervals capture the parameter

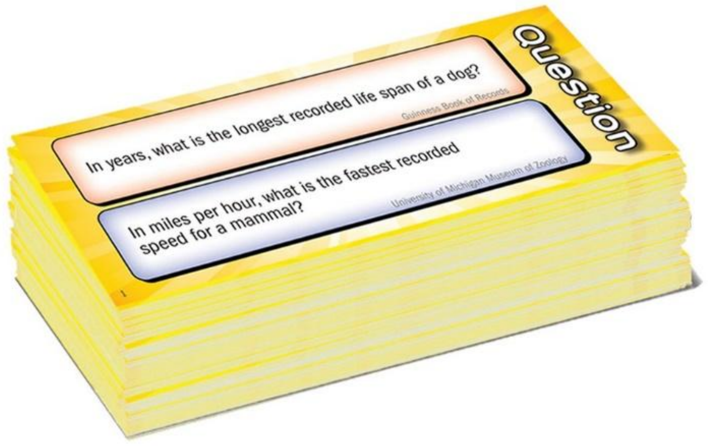
Confidence Intervals

For a **confidence level** of 95%...

95% of the **confidence intervals** will have the parameter in them



Wits and Wagers:
90% confidence interval estimator


Question
In years, what is the longest recorded life span of a dog?
Guinness Book of Records
In miles per hour, what is the fastest recorded speed for a mammal?
University of Michigan Museum of Zoology

I will ask 10 questions that have numeric answers

Please come up with a range of values that contains the true value in it for 9 out of the 10 questions

- i.e., be a 90% confidence interval estimator

100% confidence intervals

There is a tradeoff between the **confidence level** (percent of times we capture the parameter) and the **confidence interval size**

Note

For any given confidence interval we compute, we don’t know whether it has really captured the parameter

But we do know that if we do this 100 times, 90 of these intervals will have the parameter in it

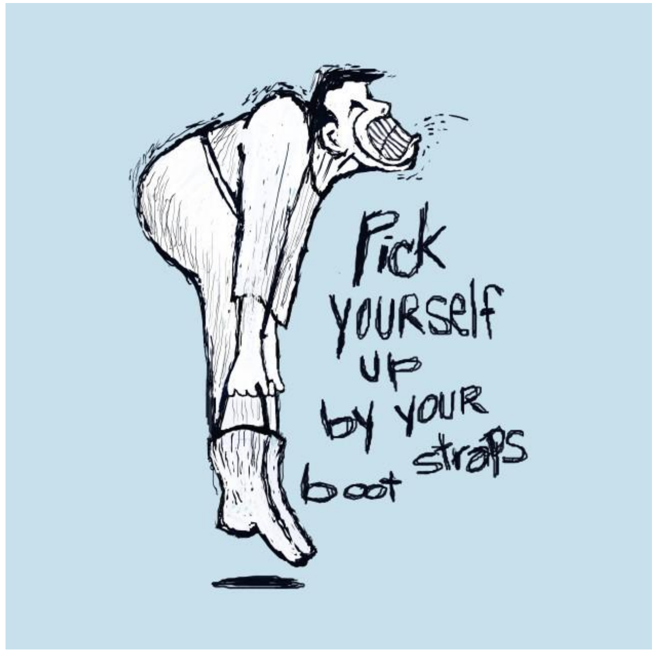
(for a 90% confidence interval)

Constructing confidence intervals

There are several methods that can be used to construct confidence intervals including

- "Parametric methods" that use probability functions
  - E.g., confidence intervals based on the normal distribution
- A "bootstrap method" where data is resampled from our original sample to approximate a sampling distribution

To learn more about these methods, take Introductory Statistics!



Reshaping data with tidyr

Wide vs. Long data

Plotting data using ggplot requires that data is in the right format

- i.e., requires data transformations

Often this involves converting data from a **wide format** to **long format**

Wide data

Person	Age	Height
Bob	32	72
Alice	24	65
Steve	64	70

Long data

Person	name	value
Bob	Age	32
Bob	Height	72
Alice	Age	24
Alice	Height	65
Steve	Age	64
Steve	Height	70

library(tidyr)

tidyr::pivot_longer()

pivot_longer(df, cols) converts data from **wide** to **long**

- Takes multiple columns and converts them into two columns: name and value
  - Column names become categorical variable levels of a new variable called **name**
  - The data in rows become entries in a variable called **value**

Wide data

Person	Age	Height
Bob	32	72
Alice	24	65
Steve	64	70

→

Long data

Person	name	value
Bob	Age	32
Bob	Height	72
Alice	Age	24
Alice	Height	65
Steve	Age	64
Steve	Height	70

tidyr::pivot_wider()

pivot_wider(df, names_from, values_from) converts data from long to wide

- Turns the levels of categorical data into columns in a data frame

Narrow data

person	name	value
Bob	Age	32
Bob	Height	72
Alice	Age	24
Alice	Height	65
Steve	Age	64
Steve	Height	70

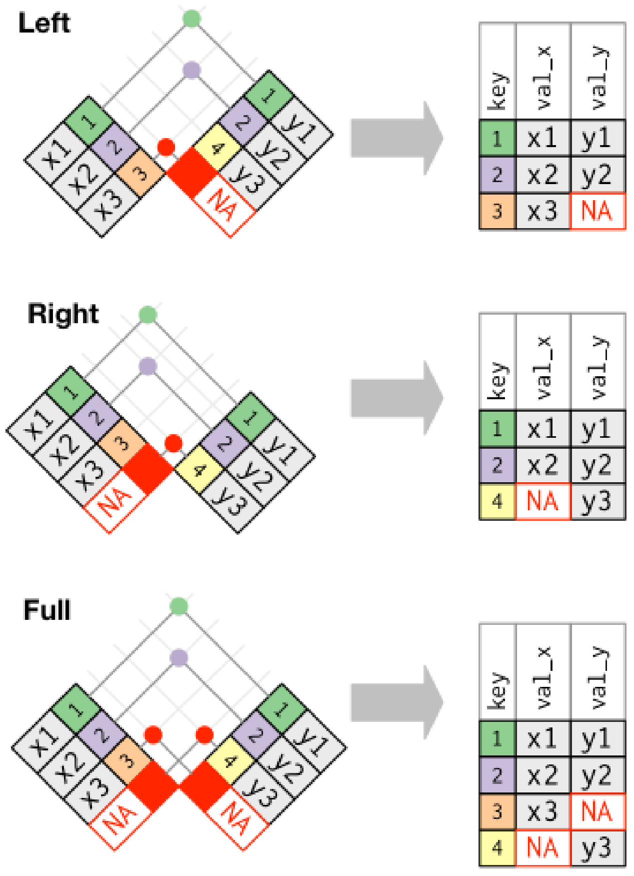
$\rightarrow$

Wide data

Person	Age	Height
Bob	32	72
Alice	24	65
Steve	64	70

Let’s try it in R…

Joining data frames



key	val_x	val_y
1	x1	y1
2	x2	y2
3	x3	NA

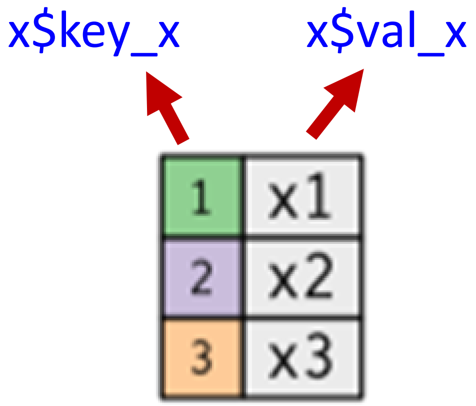
key	val_x	val_y
1	x1	y1
2	x2	y2
4	NA	y3

key	val_x	val_y
1	x1	y1
2	x2	y2
3	x3	NA
4	NA	y3

Left and right tables

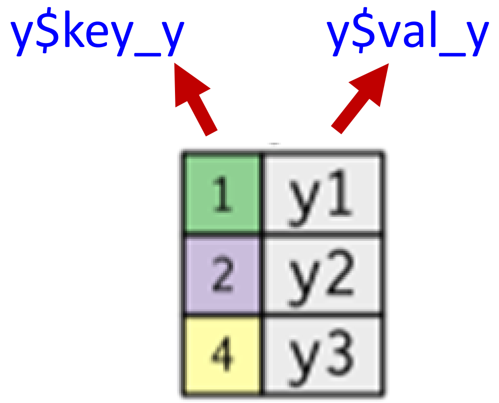
Suppose we have two data frames called x and y

- x have two variables called key_x, and val_x
- y has two variables called key_y and val_y


x$key_x
x$val_x

1	x1
2	x2
3	x3

Data frame x


y$key_y
y$val_y

1	y1
2	y2
4	y3

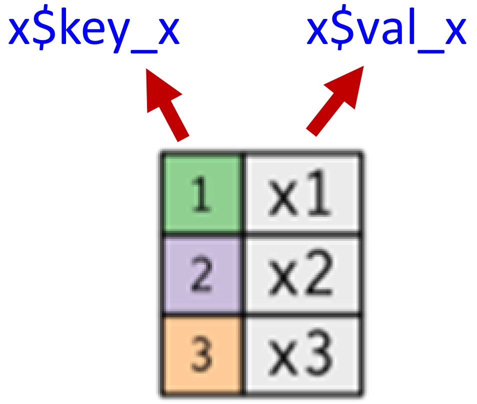
Data frame y

```
SDS111::download_data('x_y_join.rda')
```

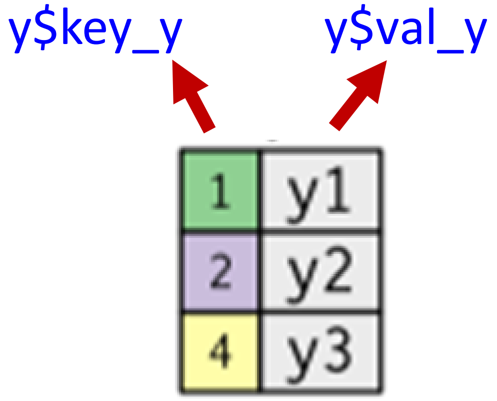
Left and right tables

Suppose we have two data frames called x and y

- x have two variables called key_x, and val_x
- y has two variables called key_y and val_y



Data frame x



Data frame y

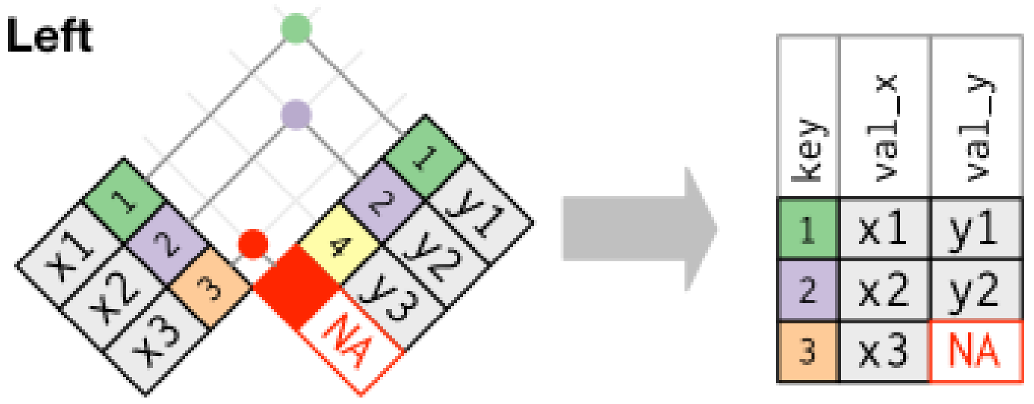
Joins have the general form:

```
join(x, y, by = c("key_x" = "key_y"))
```

Left joins

Left joins keep all rows in the left table.

Data from right table is added when there is a matching key, otherwise NA as added.



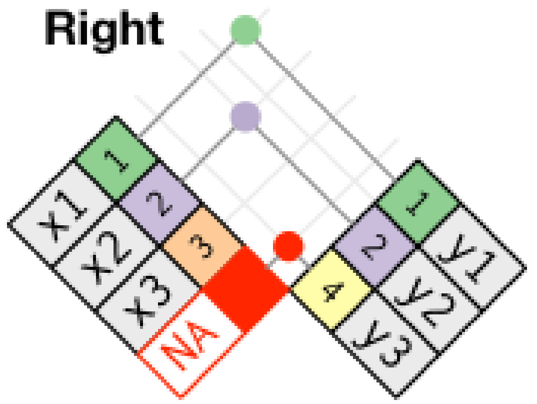
key	val_x	val_y
1	x1	y1
2	x2	y2
3	x3	NA

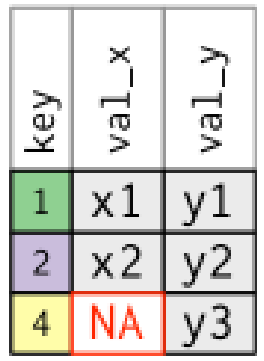
```
> left_join(x, y, by = c("key_x" = "key_y"))
```

Right joins

Right joins keep all rows in the right table.

Data from left table added when there is a matching key, otherwise NA as added.



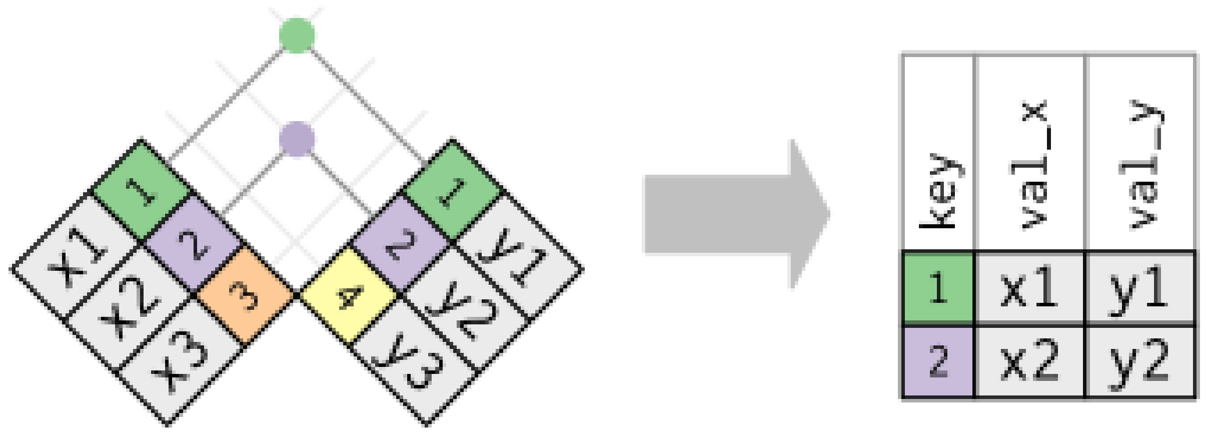


key	val_x	val_y
1	x1	y1
2	x2	y2
4	NA	y3

```
> right_join(x, y, by = c("key_x" = "key_y"))
```

Inner joins

Inner joins only keep rows in which there are matches between the keys in both tables.



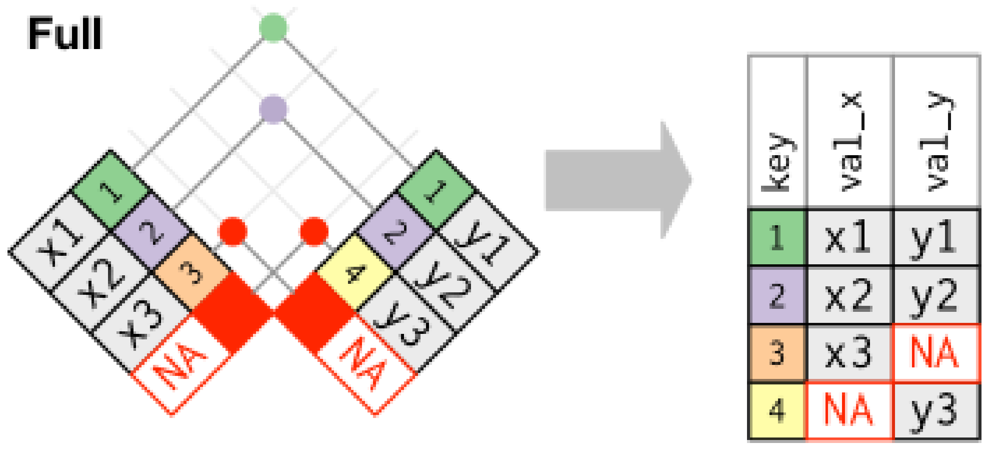
key	val_x	val_y
1	x1	y1
2	x2	y2

```
> inner_join(x, y, by = c("key_x" = "key_y"))
```

Full joins

Full joins keep all rows in both table.

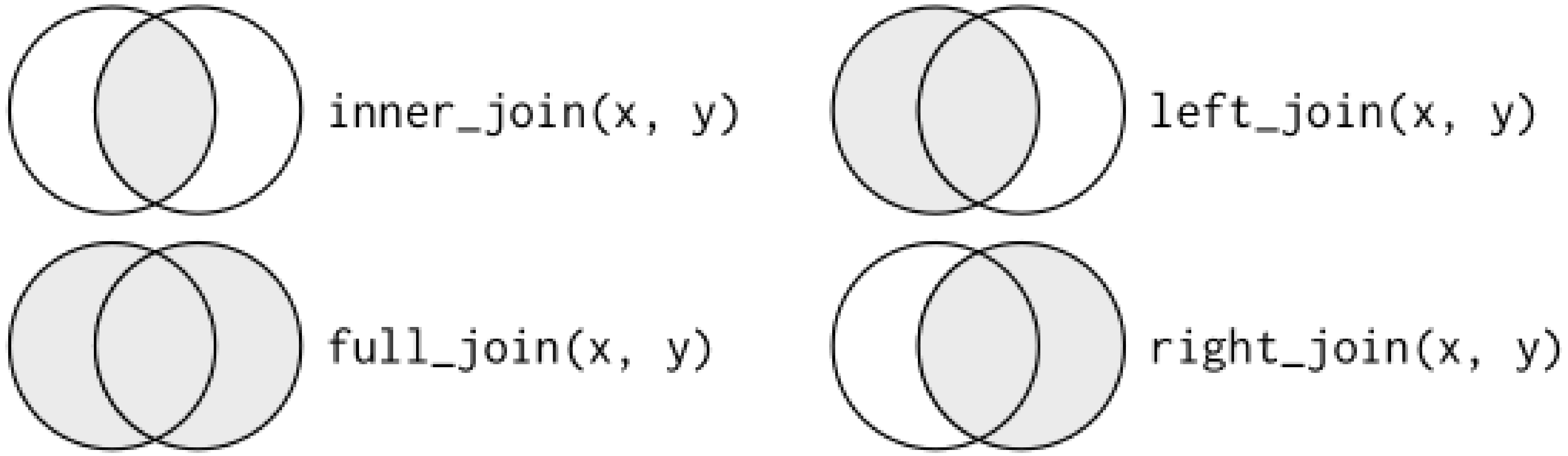
NAs are added where there are no matches.



key	val_x	val_y
1	x1	y1
2	x2	y2
3	x3	NA
4	NA	y3

```
> full_join(x, y, by = c("key_x" = "key_y"))
```

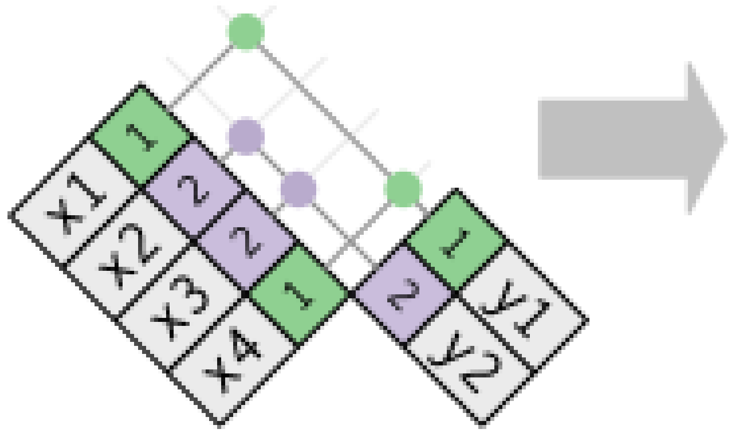
Summary

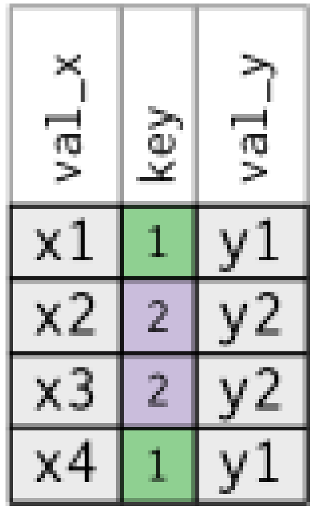


Duplicate keys

Duplicate keys are useful if there is a many-to-one relationship

- e.g., duplicates are useful in the left table when doing a left join



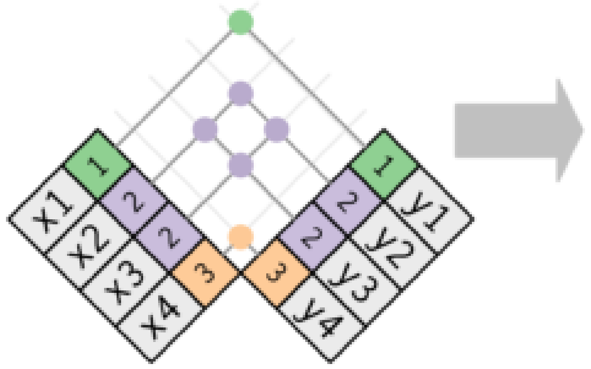


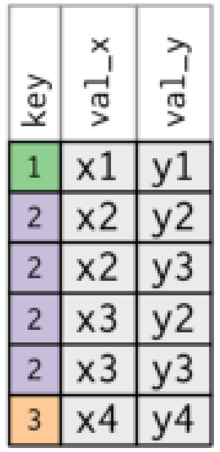
val1_x	key	val1_y
x1	1	y1
x2	2	y2
x3	2	y2
x4	1	y1

Duplicate keys

If both tables have duplicate keys you get all possible combinations of joined values (Cartesian product).

- **This is usually an error!**





key	val_x	val_y
1	x1	y1
2	x2	y2
2	x2	y3
2	x3	y2
2	x3	y3
3	x4	y4

Always check the output size after you join a table because even if there is not a syntax error you might not get the table you are expecting!

- You can check how many rows a data frame has using the nrow() function

Duplicate keys

To deal with duplicate keys in both tables, we can join the tables using multiple keys in order to make sure that each row is uniquely specified.

We can do this using the syntax:

```
join(x2, y2, by = c("key1_x" = "key1_y", "key2_x" = "key2_y"))
```

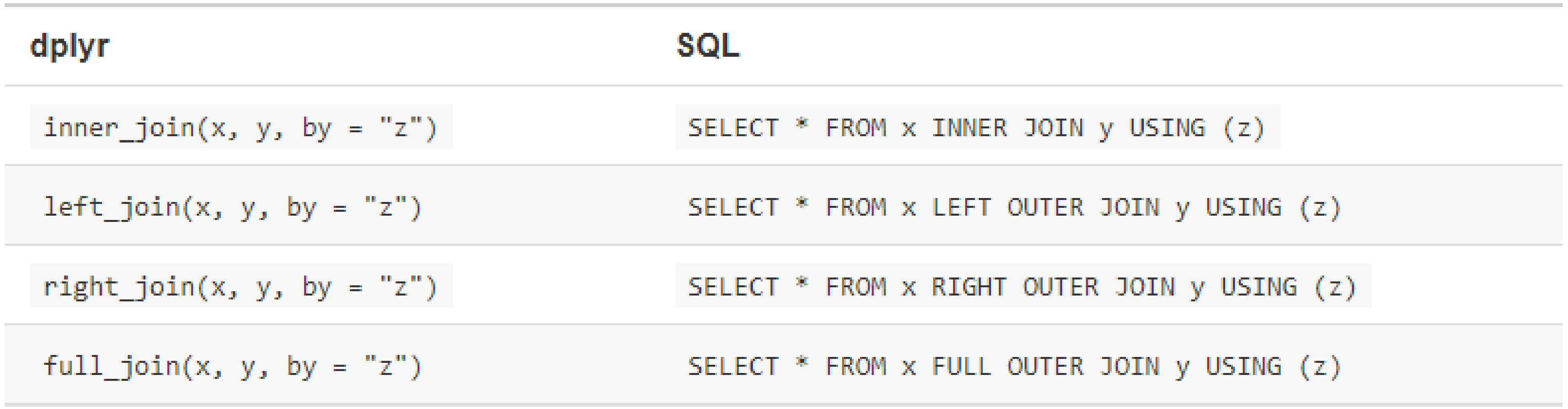
Duplicate keys

```
>  x2 <- data.frame(key1_x = c(1, 2, 2),
                    key2_x = c("a", "a", "b"),
                    val_x = c("y1", "y2", "y3"))

>  y2 <- y2 <- data.frame(key1_y = c(1, 2, 2, 3, 3),
                    key2_y = c("a", "a", "b", "a", "b"),
                    val_y = c("y1", "y2", "y3", "y4", "y5"))

> left_join(x2, y2, c("key1_x" = "key1_y"))

> left_join(x2, y2, c("key1_x" = "key1_y", "key2_x" = "key2_y"))
```

Structured Query Language

Having multiple tables that can be joined together is common in Relational Database Systems (RDBS).

- A common language used by RDBS is Structured Query Language (SQL)



dplyr	SQL
`inner_join(x, y, by = "z")`	`SELECT * FROM x INNER JOIN y USING (z)`
`left_join(x, y, by = "z")`	`SELECT * FROM x LEFT OUTER JOIN y USING (z)`
`right_join(x, y, by = "z")`	`SELECT * FROM x RIGHT OUTER JOIN y USING (z)`
`full_join(x, y, by = "z")`	`SELECT * FROM x FULL OUTER JOIN y USING (z)`

Let’s try it in R…