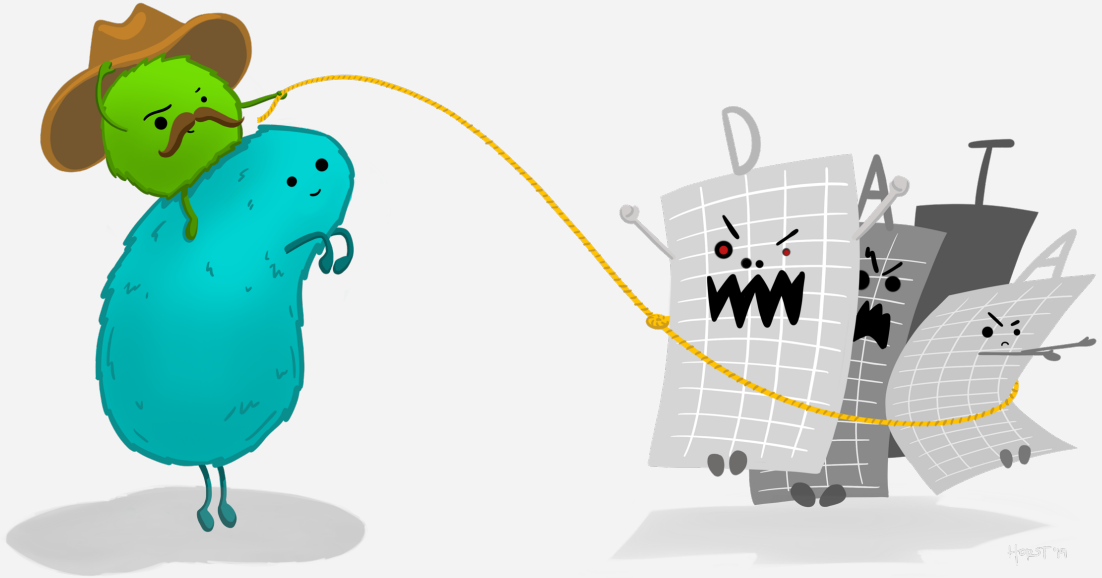
Introduction to Reproducible Science: *Day 2*

Nelson Roque, PhD



Scan for Slides & Code available on GitHub

Agenda: Day 2

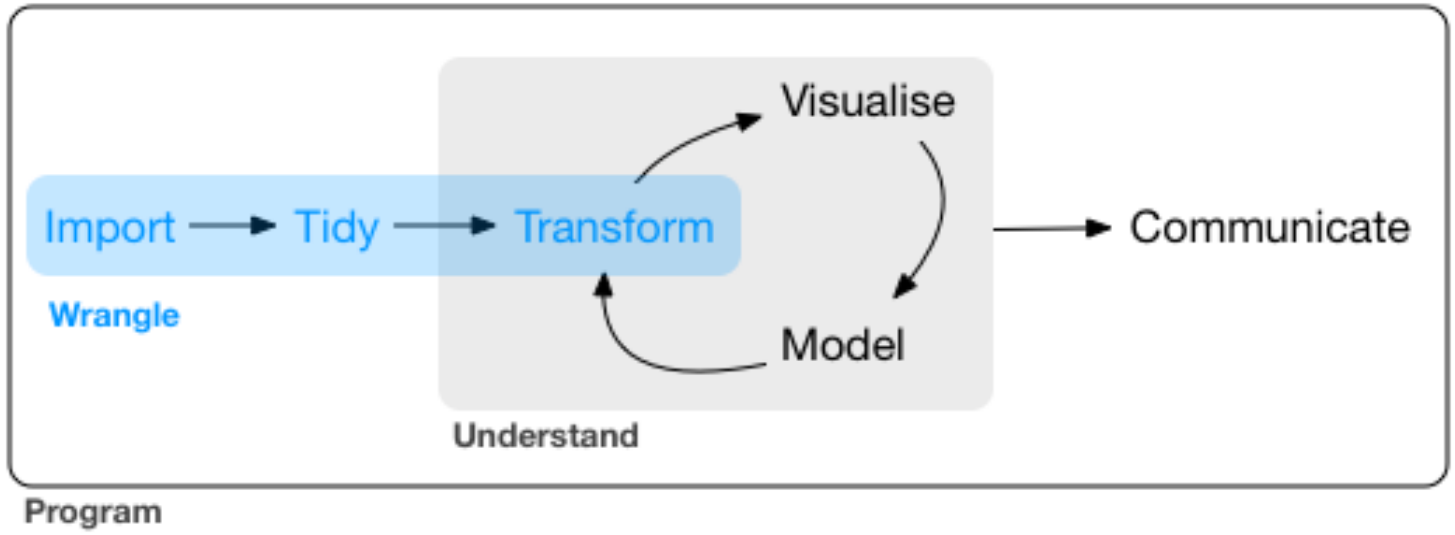
- Data wrangling and visualization of Big Data
  - **Skill 1: Data wrangling the Google Mobility dataset**

- Reproducible survey research
  - Qualtrics survey design tips
  - **Skill 2: Data wrangling Qualtrics data**

- Working with JSON data
  - **Skill 3: cleaning and visualizing keystroke JSON data**

Skill 1: Data Wrangling & Visualization

What is data wrangling?



- The operations that carry your data from raw form, into something visualizable or analyzable
  - i.e., the data preparation phase of the research

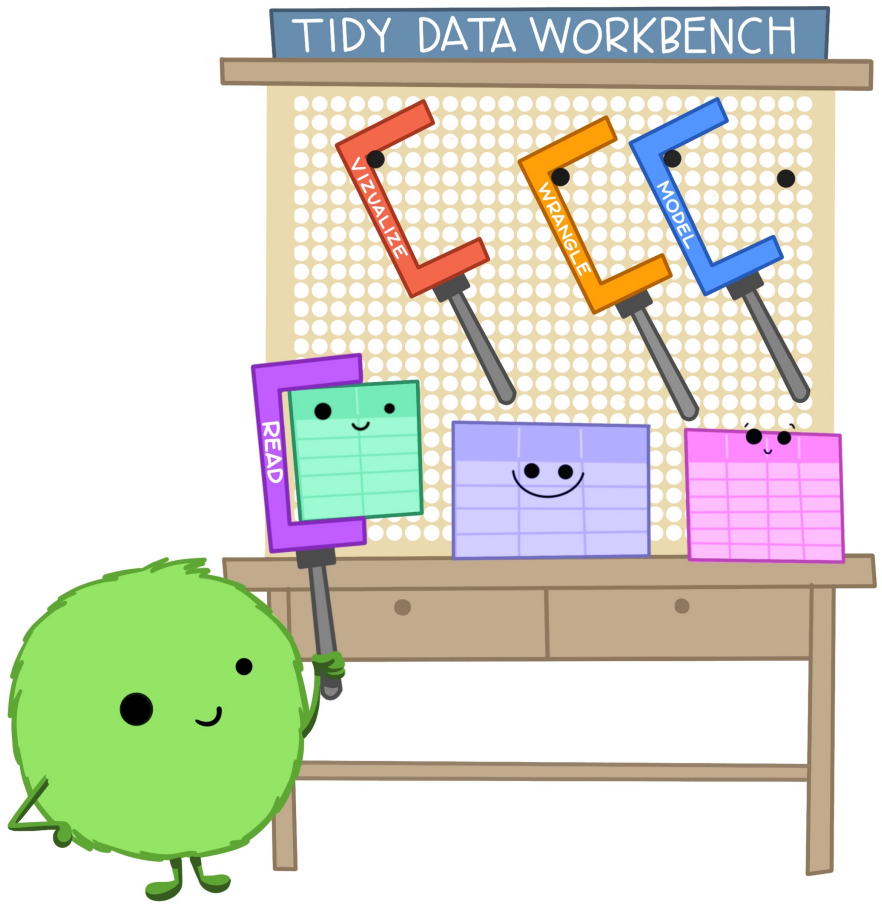
What is tidy data?

1. Every column is a variable.
2. Every row is an observation.
3. Every cell is a single value.

participant	condition	avg_response_time	perc_accuracy
9991	control	506	90
9992	active	516	96
9993	control	526	99

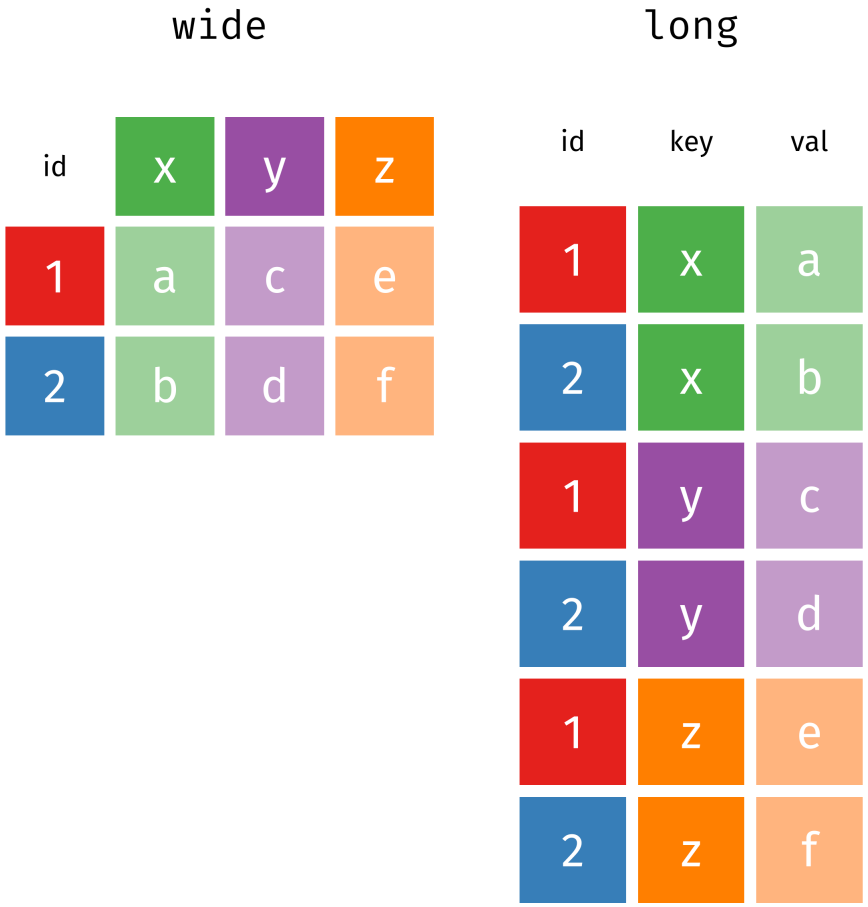
Source: R for Data Science

When working with tidy data, we can use the same tools in similar ways for different datasets...



Typical data wrangling operations

- Data exploration
  - Plotting, descriptive stats
- Dealing with missing data
  - Impute missing data, insert missing codes (-999)
- Reshaping data
  - Add columns (e.g., create flag is missing more than 10 records)
  - Update column names (id = participant_id)
  - Converting between long and wide format



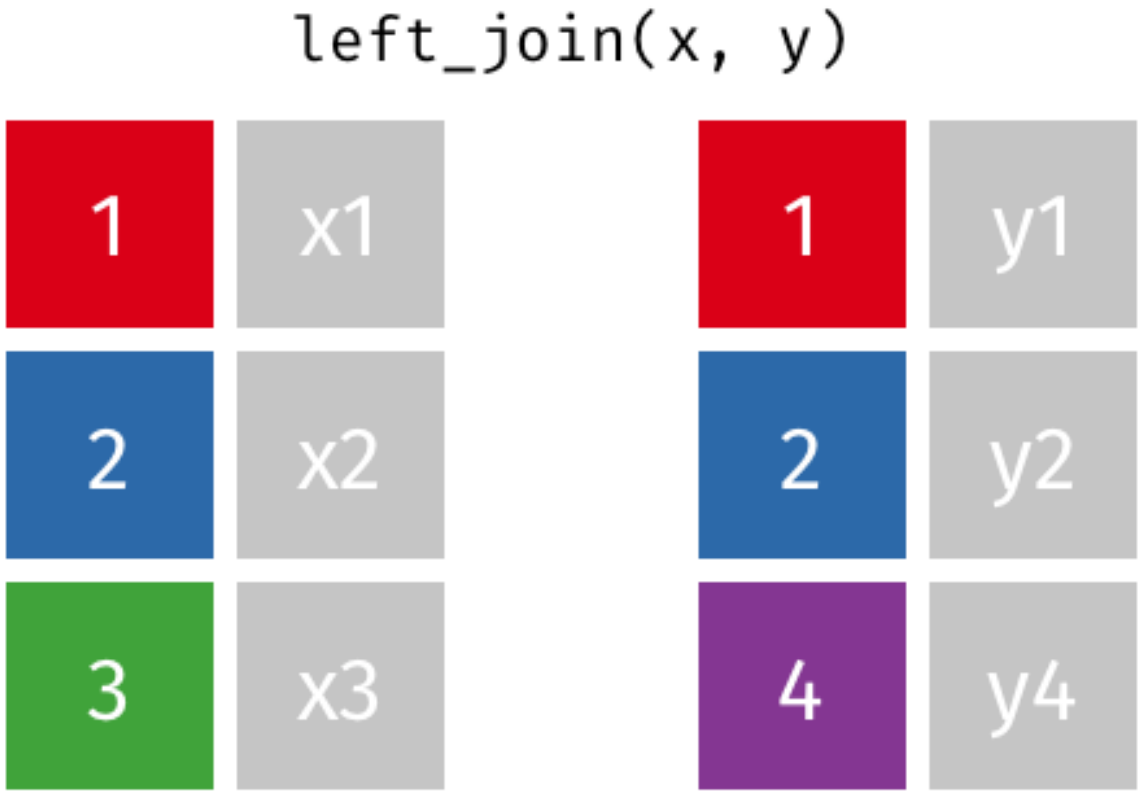
Typical data wrangling operations

- Filtering data
  - Remove specific observations, by column or row
- Merging/matching data from various sources
  - Join data by common id(s) in various ways (more on next slides)
  - https://dplyr.tidyverse.org/articles/two-table.html
- Other wrangling
  - Feature engineering
    - E.g., Add ‘features’ of date as new columns (e.g., what is day of week for 10/13/2020)

Joining data

- **Left join**

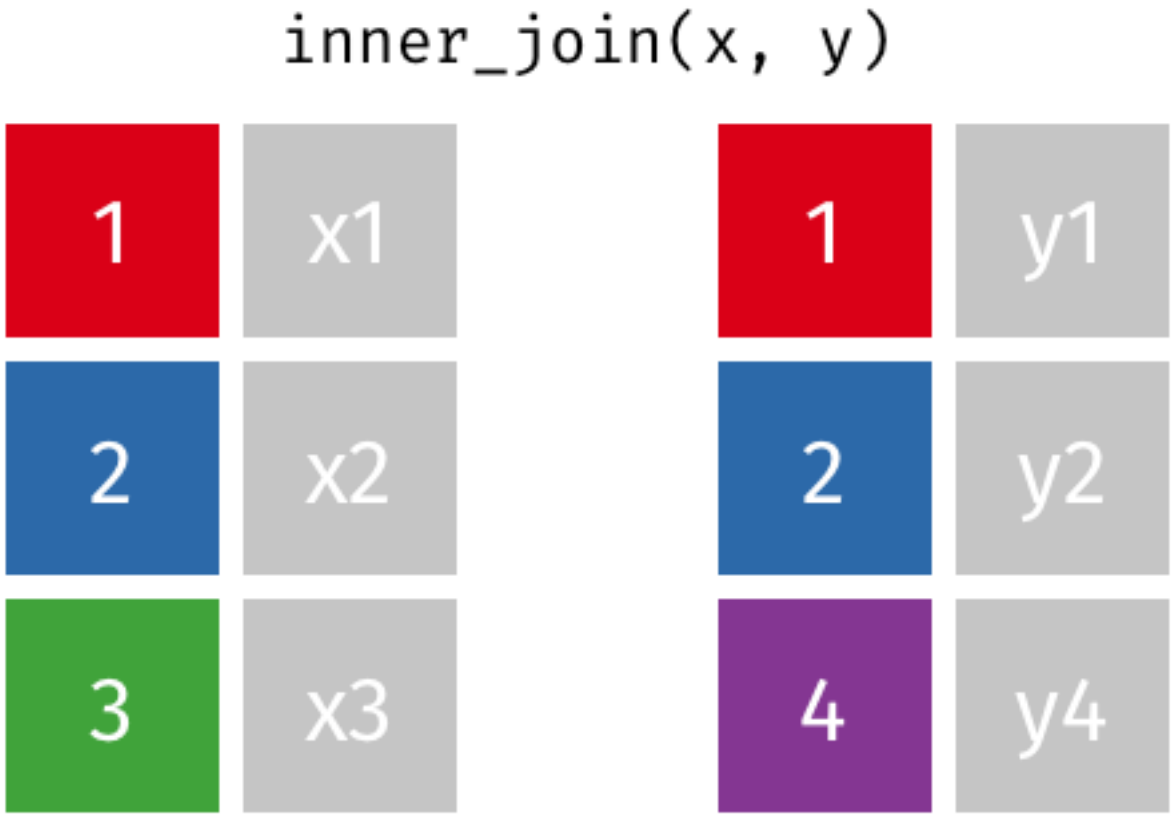
- All rows from x, and all columns from x and y. Rows in x with no match in y will have NA values in the new columns.



Joining data

- **inner join**

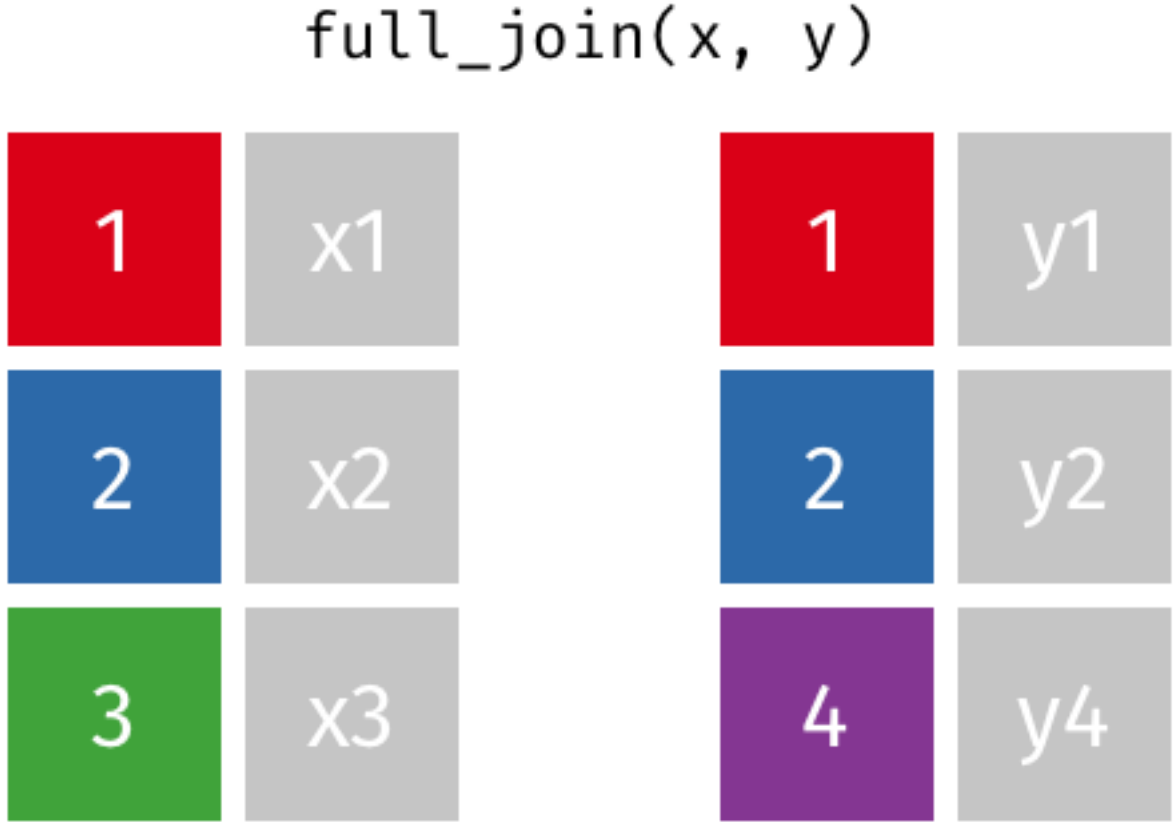
- All rows from x where there are matching values in y, and all columns from x and y.



Joining data

- **full join**

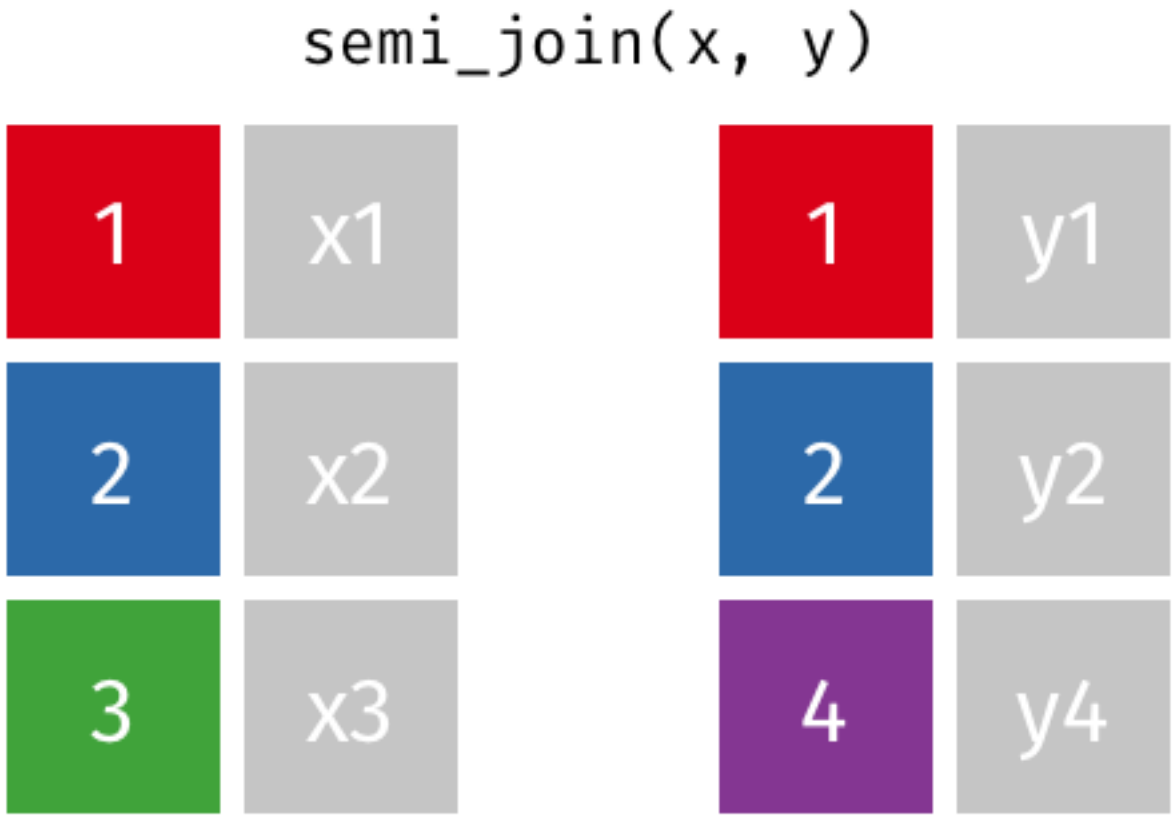
- All rows and all columns from both x and y.
- Where there are not matching values, returns NA for the one missing.



Joining data

- **semi join**

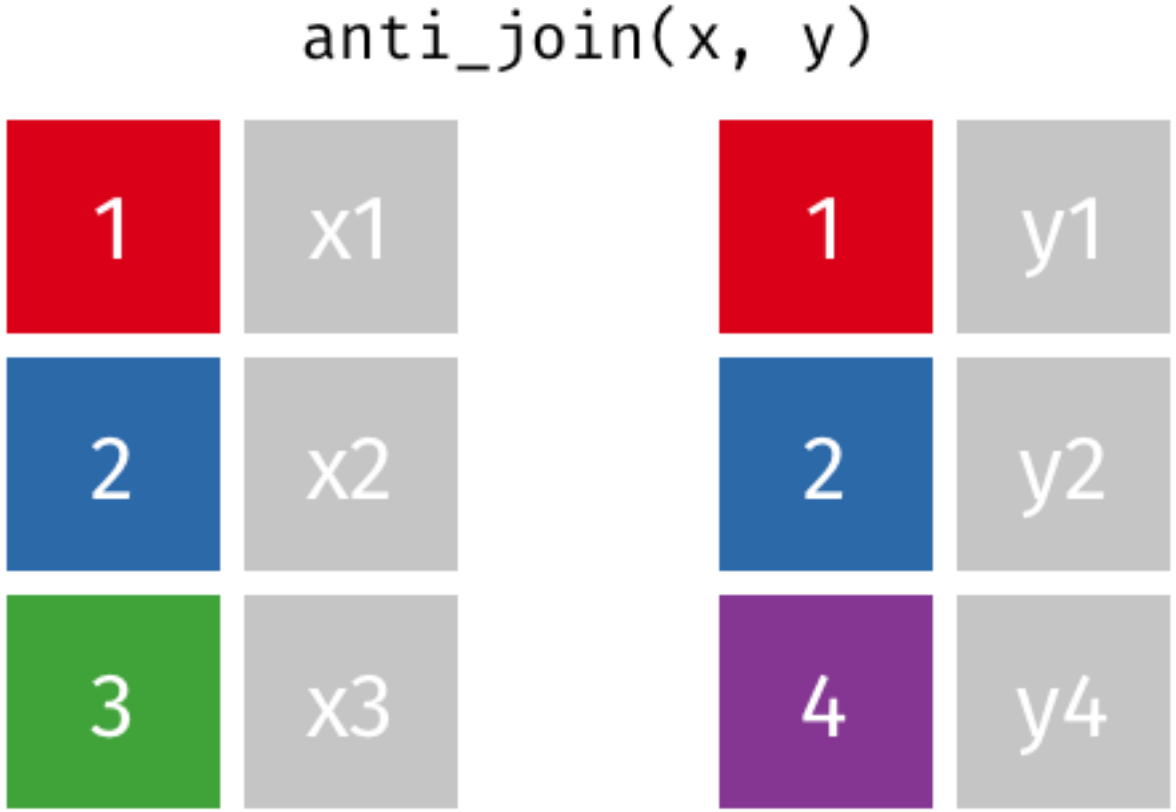
- All rows from x where there are matching values in y, keeping just columns from x.
- Basically filtering one dataframe, using another dataframe as the match.



Joining data

- **anti join**

- Keeps all rows from x where there are not matching values in y, keeping just columns from x.
- Find out which data is **not** joining, or may be missing from one or both datasets.



The Data Source: Google Mobility

Learn more about this data
https://www.google.com/covid19/mobility/data_documentation.html?hl=en

Let’s jump into R

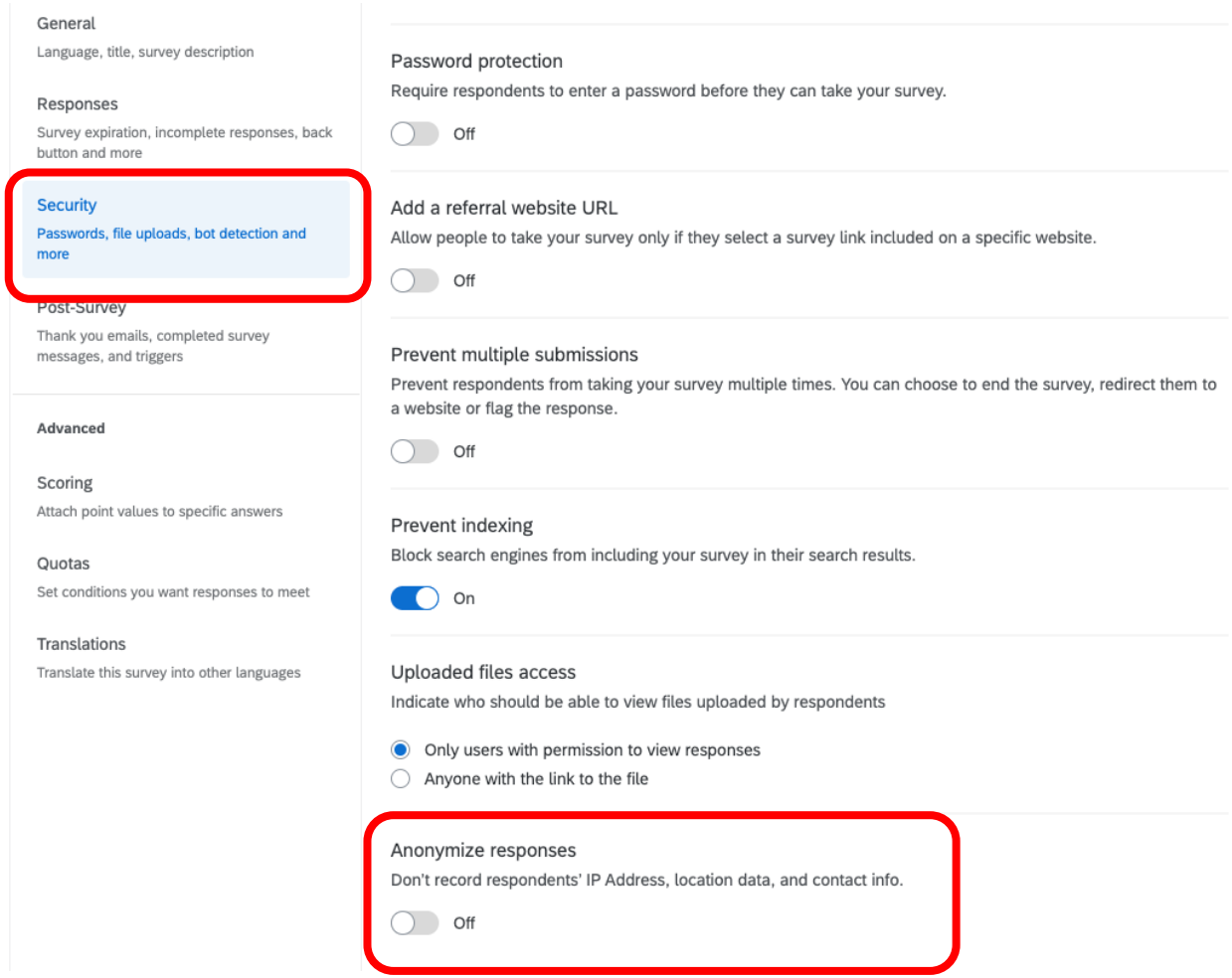
Scan the QR code to view the code we will work with

Skill 2: Data Wrangling – Qualtrics Data

PSA – Anonymize your Qualtrics Surveys

- **By default,** Qualtrics collects the following information
  - IP Address
  - Location data
  - Contact info (if provided)

- Turn this off unless needed (and IRB aware)



Working with Qualtrics

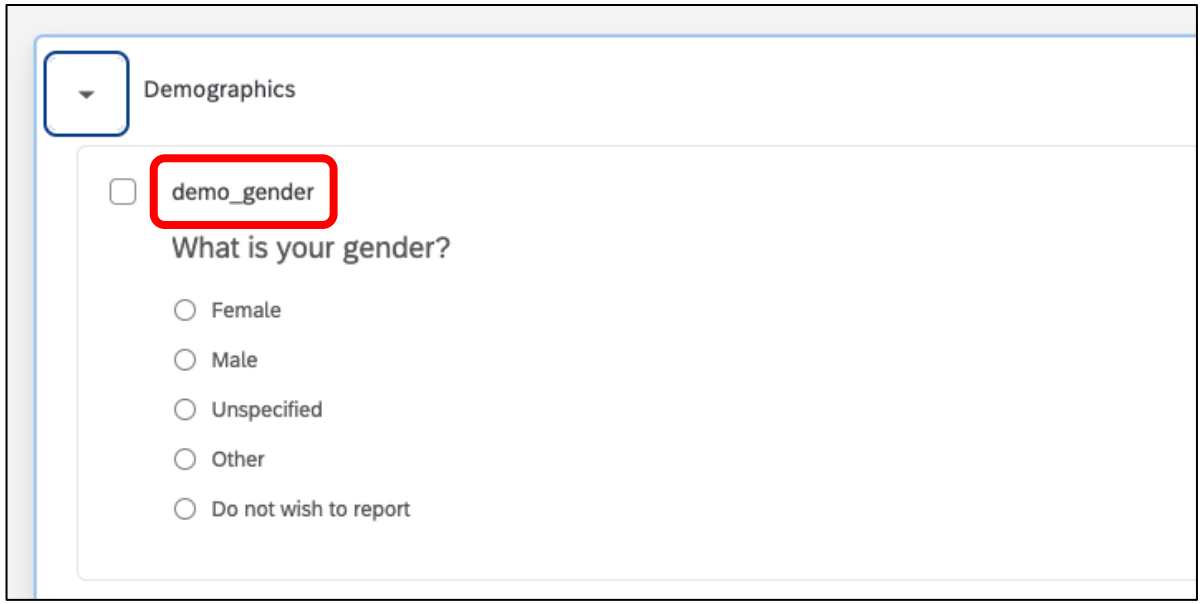
- The **triple header problem**

A	B	C	D	E	F	G	H	I	J	K
StartDate	EndDate	Status	Progress	Duration (in	Finished	RecordedDat	ResponseId	DistributionC	UserLanguag	consent_yn
Start Date	End Date	Response Ty	Progress	Duration (in	Finished	Recorded Da	Response ID	Distribution	User Langua	Do you conse
{"ImportId":'	{"ImportId":'	{"ImportId":'	{"ImportId":'	{"ImportId":'	{"ImportId":'	{"ImportId":'	{"ImportId":'	{"ImportId":'	{"ImportId":'	{"ImportId":'

  - Solution: using the qualtRics package
    - `qualtrics_df <- qualtRics::read_survey(filename)`

Working with Qualtrics

- **Survey bloat** – finding what you need in a large survey
  - Solution: prefix questions with helpful identifiers for grouping ‘like’ questions
  - Downstream in R, can easily select columns of interest without typing all column names




```r
# select all demographics ----
df_demo = qualtrics_df %>%
  select(ResponseId, participant_id, is_pilot_data,
    contains("demo_"), -contains("_timer"))
```

Working with Qualtrics Data in R

- **Recoding values** similarly across many columns at once
  - Solution: convert data to long format, apply scoring on long-format data and then finally convert back to wide-format

```r
# create TIPI dataset -----
tipi_df = qualtrics_df %>%
  select(RAND_ID, contains("TIPI"), -contains("_Timer"))

# transform TIPI from wide to long ----
tipi_long = tipi_df %>%
  pivot_longer(cols=TIPI_1:TIPI_10,
    names_to="tipi_question",
    values_to="response")

# TODO: add recoding for rest of TIPI
tipi_long_r = tipi_long %>%
  mutate(response_n = recode(response,
                             `Disagree strongly` = 1,
                             `Disagree moderately` = 2,
                             `Disagree a little` = 3,
                             `Neither agree nor disagree` = 4,
                             `Agree a little` = 5,
                             `Agree moderately` = 6,
                             `Agree strongly` = 7,
                             .default=-999))
```

Let’s jump into R

Scan the QR code to view the code we will work with

Skill 3: Data Wrangling – JSON data

What is JSON?

- **J**ava**S**cript **O**bject **N**otation
- Lightweight format for storing and transporting data (similar to csv)
- Often used when data is sent from a server to a web page
- "self-describing" and easy to understand

```
{"menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      {"value": "New", "onclick": "CreateNewDoc()"},
      {"value": "Open", "onclick": "OpenDoc()"},
      {"value": "Close", "onclick": "CloseDoc()"}
    ]
  }
}}
```

The same text expressed as XML:

```
<menu id="file" value="File">
  <popup>
    <menuitem value="New" onclick="CreateNewDoc()" />
    <menuitem value="Open" onclick="OpenDoc()" />
    <menuitem value="Close" onclick="CloseDoc()" />
  </popup>
</menu>
```

https://www.w3schools.com/whatis/whatis_json.asp

JSON Syntax Rules

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

```
{"menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      {"value": "New", "onclick": "CreateNewDoc()"},
      {"value": "Open", "onclick": "OpenDoc()"},
      {"value": "Close", "onclick": "CloseDoc()"}
    ]
  }
}}
```

The same text expressed as XML:

```
<menu id="file" value="File">
  <popup>
    <menuitem value="New" onclick="CreateNewDoc()" />
    <menuitem value="Open" onclick="OpenDoc()" />
    <menuitem value="Close" onclick="CloseDoc()" />
  </popup>
</menu>
```

https://www.w3schools.com/whatis/whatis_json.asp

Working with JSON in R

- **library**(jsonlite)

When `simplifyDataFrame` is enabled, JSON arrays containing **objects** (key-value pairs) simplify into a data frame:

```
json <-
'[
  {"Name" : "Mario", "Age" : 32, "Occupation" : "Plumber"},
  {"Name" : "Peach", "Age" : 21, "Occupation" : "Princess"},
  {},
  {"Name" : "Bowser", "Occupation" : "Koopa"}
]'
mydf <- fromJSON(json)
mydf
```

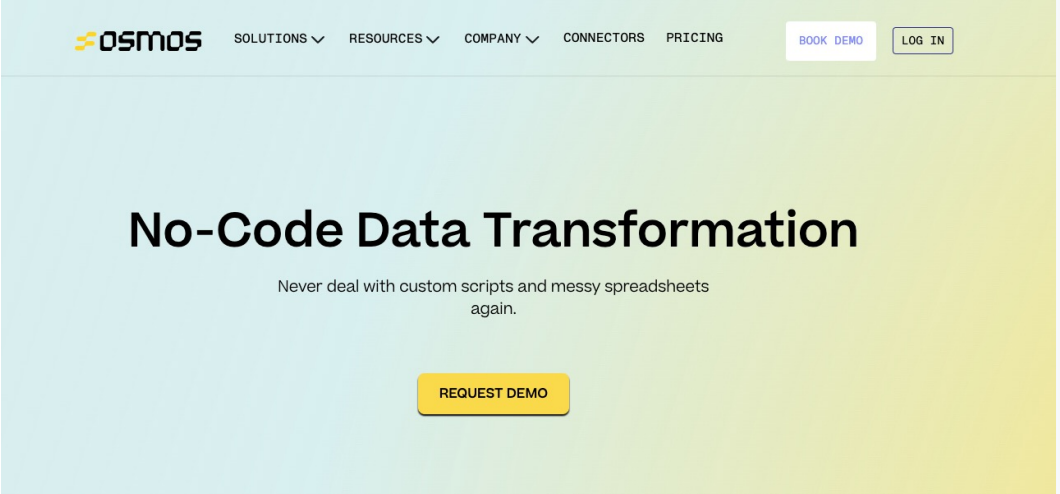
```
    Name Age Occupation
1  Mario  32    Plumber
2  Peach  21   Princess
3   <NA>  NA       <NA>
4 Bowser  NA      Koopa
```

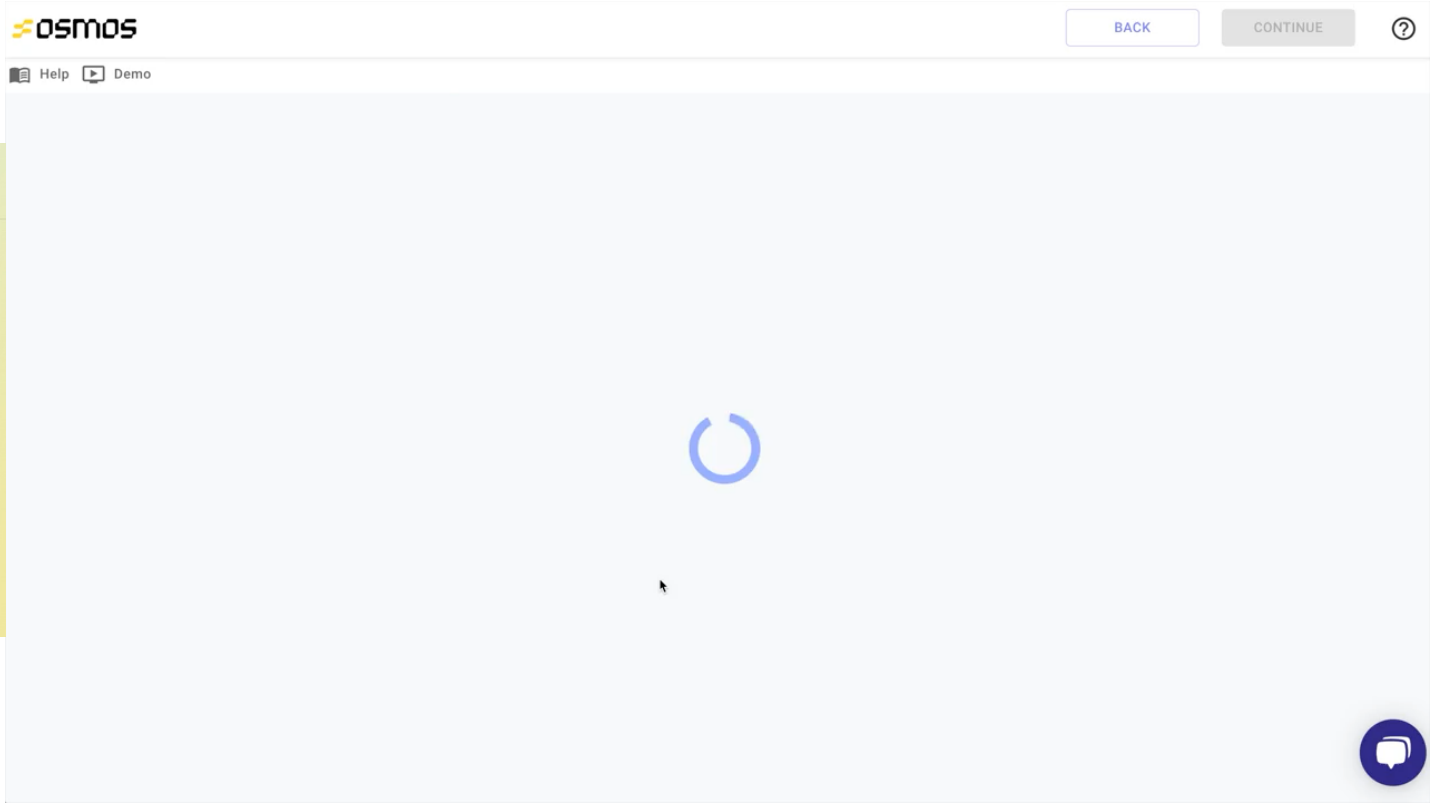
https://cran.r-project.org/web/packages/jsonlite/vignettes/json-aaquickstart.html

Let’s jump into R

Scan the QR code to view the code we will work with

No-code data transformation





No-code data transformation

Download Cheatsheets

- Data Import
  - https://github.com/rstudio/cheatsheets/raw/master/data-import.pdf)
- Data Wrangling Cheatsheet
  - https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf)
- Data Transformation with dplyr
  - https://github.com/rstudio/cheatsheets/raw/master/data-visualization.pdf)
- String Manipulation
  - https://github.com/rstudio/cheatsheets/raw/master/strings.pdf)
- Work with dates/times
  - https://github.com/rstudio/cheatsheets/raw/master/lubridate.pdf)

Agenda: Day 3

- Text mining
  - **Skill 1: word and bigram frequency analysis**
  - **Skill 2: generating wordclouds**
  - **Skill 3: sentiment analysis**

- Interacting with APIs and JSON data
  - **Skill 4: querying API for results and data aggregation**

- **Closing Discussion & Q/A**

thank you

nelson.roque@ucf.edu

nelsonroque.com

https://github.com/nelsonroque