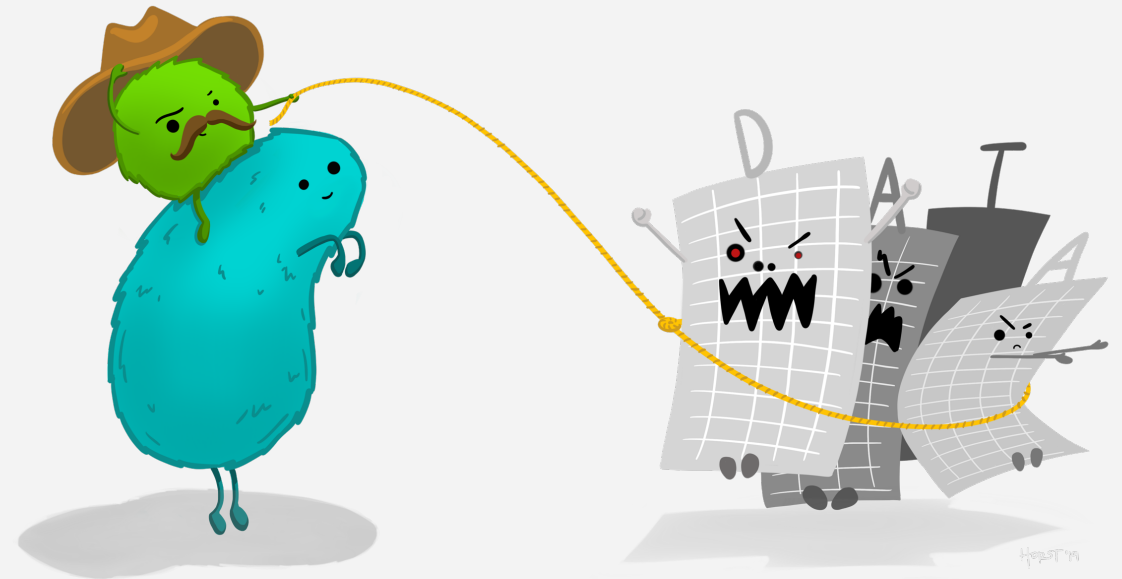


# Introduction to Reproducible Science: *Day 2*

Nelson Roque, PhD



Scan for  
Slides & Code  
available on GitHub

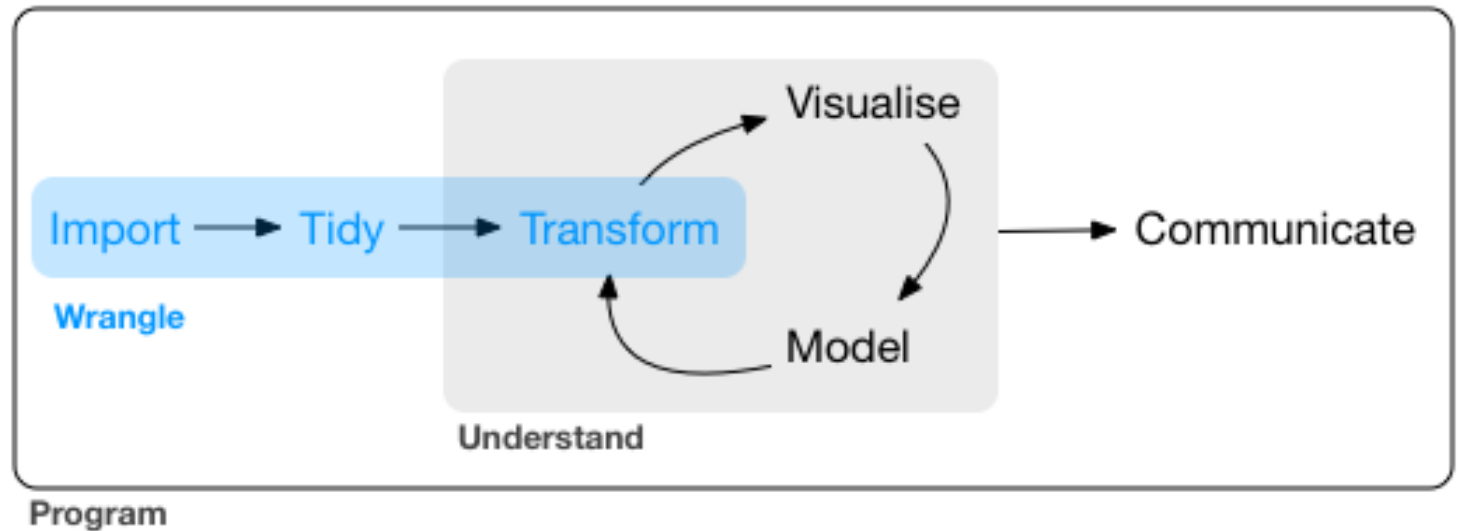


# Agenda: Day 2

- 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**

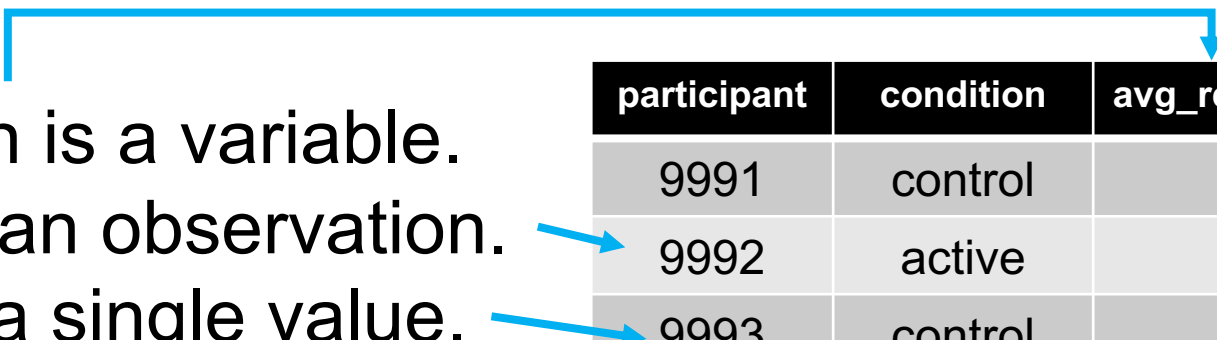
# What is data wrangling?



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

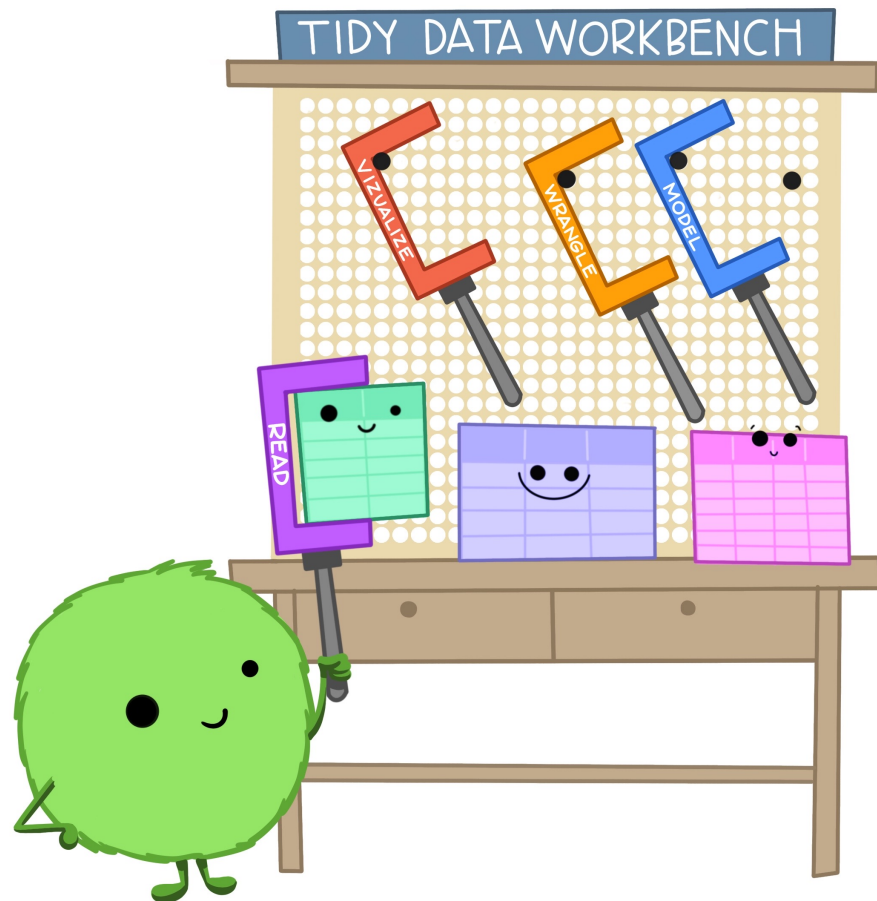
# What is tidy data?

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

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

wide				long		
id	x	y	z	id	key	val
1	a	c	e	1	x	a
2	b	d	f	2	x	b
				1	y	c
				2	y	d
				1	z	e
				2	z	f

# 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.

`left_join(x, y)`

1	x1	1	y1
2	x2	2	y2
3	x3	4	y4

# Joining data

- **inner join**

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

`inner_join(x, y)`

1	x1	1	y1
2	x2	2	y2
3	x3	4	y4

# 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.

`full_join(x, y)`

1	x1	1	y1
2	x2	2	y2
3	x3	4	y4

# 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.

`semi_join(x, y)`

1	x1	1	y1
2	x2	2	y2
3	x3	4	y4

# 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.

`anti_join(x, y)`

1	x1	1	y1
2	x2	2	y2
3	x3	4	y4

# The Data Source: Google Mobility

Google COVID-19 Community Mobility Reports



See how your community is  
moving around differently due  
to COVID-19

Learn more about this data

[https://www.google.com/covid19/mobility/data\\_documentation.html?hl=en](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)

The image shows a screenshot of the Qualtrics survey settings interface. On the left is a sidebar menu with categories: General, Responses, Security, Post-Survey, Advanced, Scoring, Quotas, and Translations. The 'Security' option is highlighted with a red box. The main content area on the right contains several settings. The 'Anonymize responses' setting at the bottom is also highlighted with a red box. Other settings like 'Password protection', 'Add a referral website URL', 'Prevent multiple submissions', 'Prevent indexing', and 'Uploaded files access' are visible but not highlighted.

**General**  
Language, title, survey description

**Responses**  
Survey expiration, incomplete responses, back button and more

**Security**  
Passwords, file uploads, bot detection and more

**Post-Survey**  
Thank you emails, completed survey messages, and triggers

**Advanced**

**Scoring**  
Attach point values to specific answers

**Quotas**  
Set conditions you want responses to meet

**Translations**  
Translate this survey into other languages

**Password protection**  
Require respondents to enter a password before they can take your survey.  
☐ Off

**Add a referral website URL**  
Allow people to take your survey only if they select a survey link included on a specific website.  
☐ Off

**Prevent multiple submissions**  
Prevent respondents from taking your survey multiple times. You can choose to end the survey, redirect them to a website or flag the response.  
☐ Off

**Prevent indexing**  
Block search engines from including your survey in their search results.  
☒ On

**Uploaded files access**  
Indicate who should be able to view files uploaded by respondents  
☒ Only users with permission to view responses  
☐ Anyone with the link to the file

**Anonymize responses**  
Don't record respondents' IP Address, location data, and contact info.  
☐ Off

# Working with Qualtrics

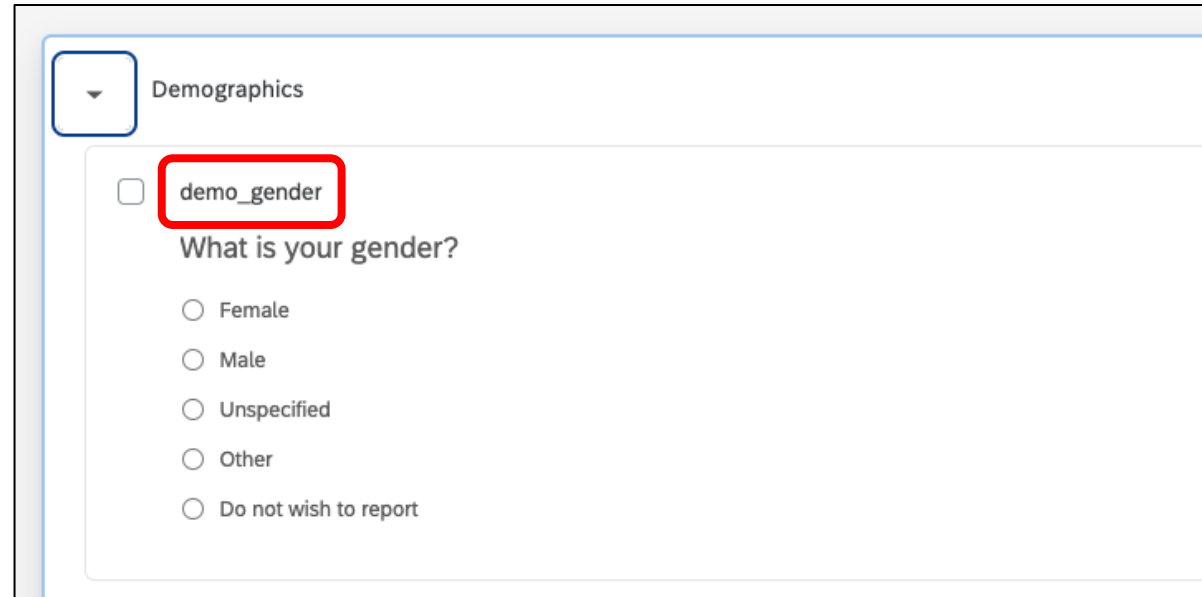
- The **triple header problem**

A	B	C	D	E	F	G	H	I	J	K
StartDate	EndDate	Status	Progress	Duration (in	Finished	RecordedDat	Responseld	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



Demographics

☐ demo\_gender

What is your gender?

☐ Female

☐ Male

☐ Unspecified

☐ Other

☐ Do not wish to report

```
24 # select all demographics ----
25 df_demo = qualtrics_df %>%
26   select(ResponseId, participant_id, is_pilot_data,
27     contains("demo_"), -contains("_timer"))
28
```

# 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

```
5 # create TIPI dataset -----
6 tipi_df = qualtrics_df %>%
7   select(RAND_ID, contains("TIPI"), -contains("_Timer"))
8
9 # transform TIPI from wide to long ----
10 tipi_long = tipi_df %>%
11   pivot_longer(cols=TIPI_1:TIPI_10,
12     names_to="tipi_question",
13     values_to="response")
14
15 # TODO: add recoding for rest of TIPI
16 tipi_long_r = tipi_long %>%
17   mutate(response_n = recode(response,
18     `Disagree strongly` = 1,
19     `Disagree moderately` = 2,
20     `Disagree a little` = 3,
21     `Neither agree nor disagree` = 4,
22     `Agree a little` = 5,
23     `Agree moderately` = 6,
24     `Agree strongly` = 7,
25     .default=-999))
26
```

# 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?

- **JavaScript Object Notation**
- 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>
```

# 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>
```



# 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
```

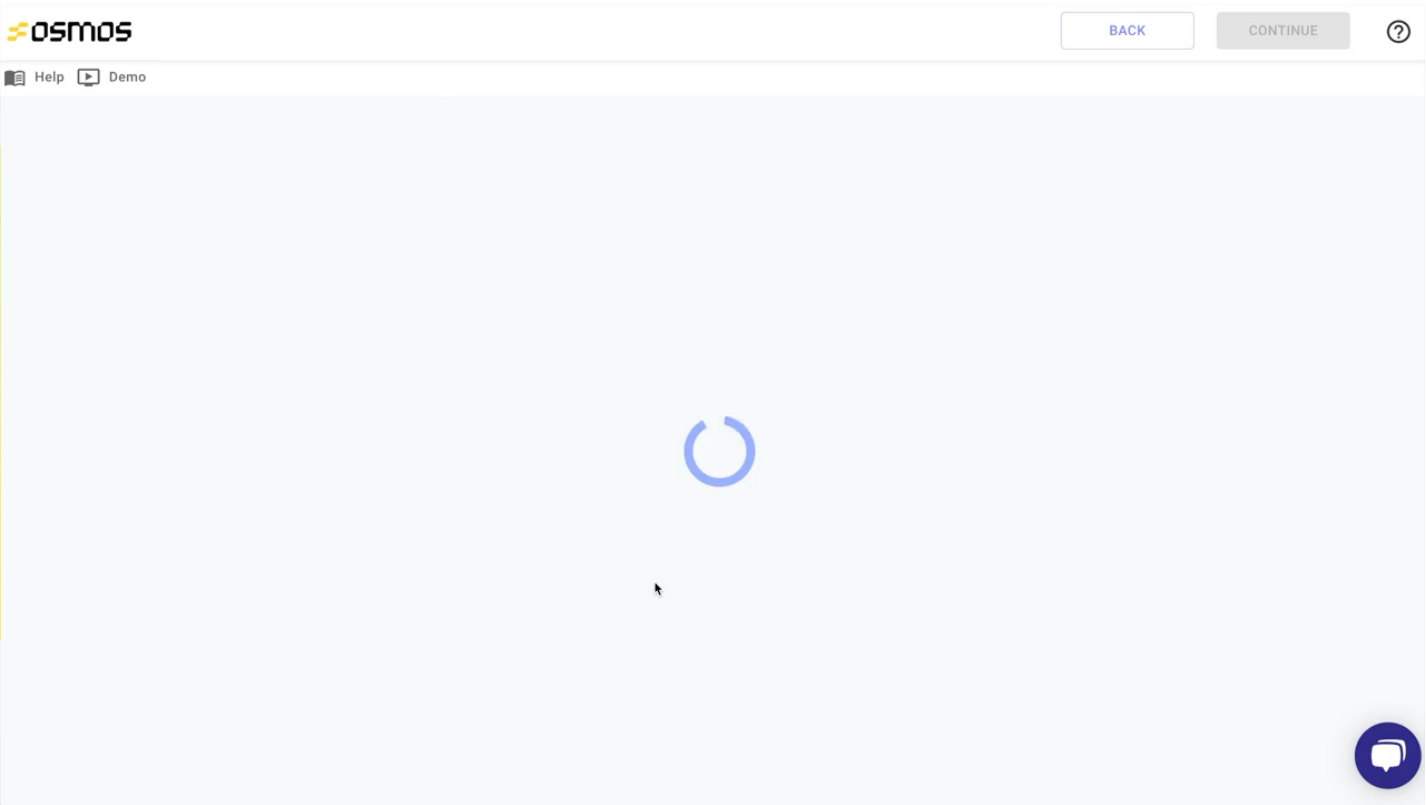
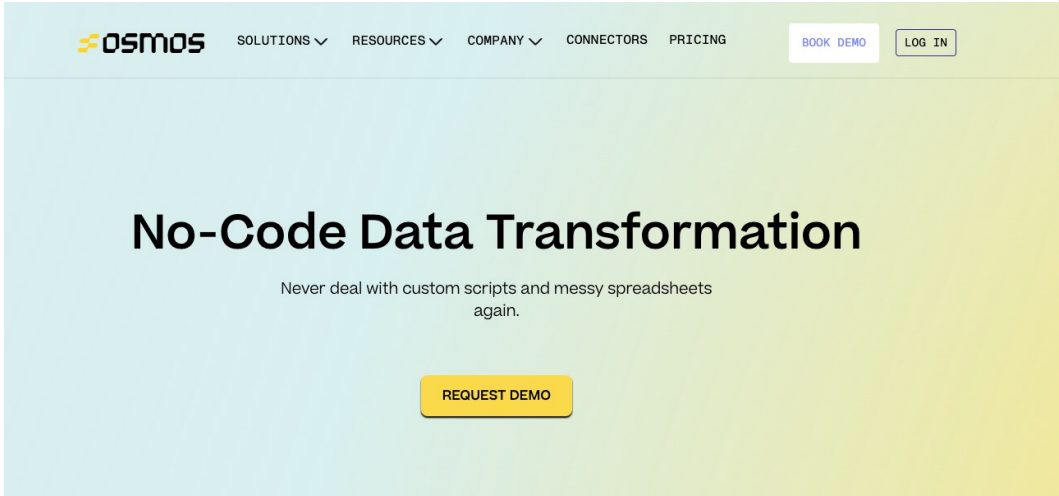
	Name	Age	Occupation
1	Mario	32	Plumber
2	Peach	21	Princess
3	<NA>	NA	<NA>
4	Bowser	NA	Koopa

# 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

## Find the connectors everyone is talking about

Access to 150+ connectors on Airbyte. Discover new ones every day.

Few places left. Take it, or leave it.



[Try Airbyte Cloud for Free](#)

### Extract from sources

Pre-built or custom connectors



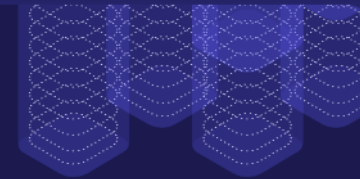
### Load to destinations

Through our UI or API



### Transform

Raw or normalized, with dbt-based transformations



# 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](mailto:nelson.roque@ucf.edu)



[nelsonroque.com](http://nelsonroque.com)



<https://github.com/nelsonroque>

