PA 446

Coding for Civic Data Applications

Will be starting at 6:05pm

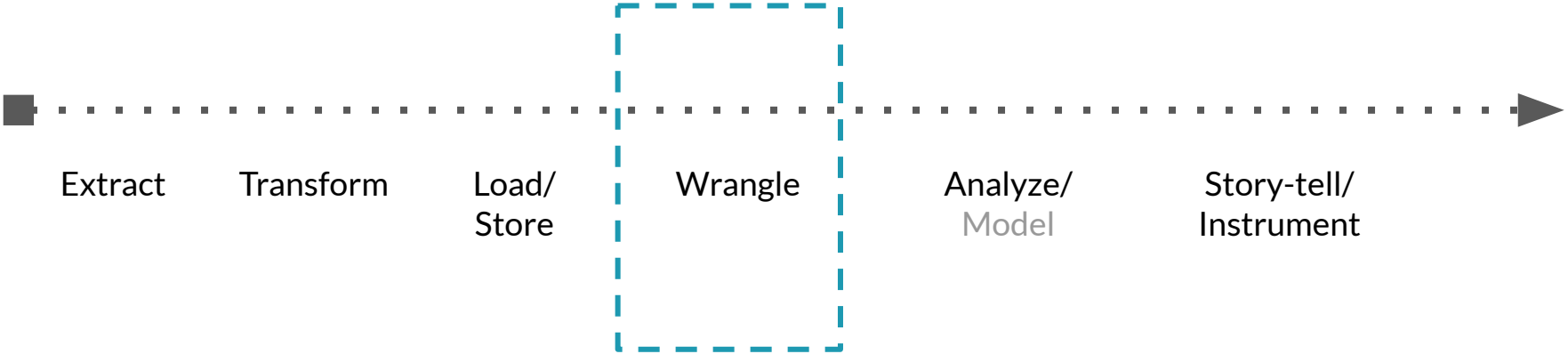
Class #4

Logistics

Course Logistics

- Homework 2 grades: will be up by end of day tomorrow
- Homework 3: posted by end of day tomorrow

Data Science “workflow”



Last session on data wrangling!

Where We Been

1. Cleaned salaries data

2. Confirmed analysis goals

3. Data Wrangling
   a. Cleaning + Enrichment + Transformation
   b. Imputed the gender column

Focus This Week

1. Cleaned salaries data

2. Confirmed analysis goals

3. Data Wrangling
   a. Cleaning + Enrichment + Transformation
   b. Imputed the gender column
   c. Impute the race column

Data “Transformation”

What Exactly Is Data Transformation

What are the main goals here?

Goals of Data Transformation

Wide Data <> Long Data

Data “Transformation”

Wide vs Long Data

“Wide” data

	year	conservative	labour	liberal	others
0	1966	253	364	12	1
1	1970	330	287	6	7
2	Feb 1974	297	301	14	18
..	...	...	...	...	...
12	2015	330	232	8	80
13	2017	317	262	12	59
14	2019	365	202	11	72

“Long” data

	year	party	seats
0	1966	Conservative	253
1	1970	Conservative	330
2	Feb 1974	Conservative	297
3	Oct 1974	Conservative	277
4	1979	Conservative	339
..	...	...	...
55	2005	Others	30
56	2010	Others	29
57	2015	Others	80
58	2017	Others	59
59	2019	Others	72

So What?

What are the advantages of each?

"Wide" data

	year	conservative	labour	liberal	others
0	1966	253	364	12	1
1	1970	330	287	6	7
2	Feb 1974	297	301	14	18
..	...	...	...	...	...
12	2015	330	232	8	80
13	2017	317	262	12	59
14	2019	365	202	11	72

"Long" data

	year	party	seats
0	1966	Conservative	253
1	1970	Conservative	330
2	Feb 1974	Conservative	297
3	Oct 1974	Conservative	277
4	1979	Conservative	339
..	...	...	...
55	2005	Others	30
56	2010	Others	29
57	2015	Others	80
58	2017	Others	59
59	2019	Others	72

Purpose of Each Data Format

Long vs Wide

Wide data

- Easy to understand
- Good for analysts and excel

Long data

- Can be more performative in an engineering sense
- “Tidy”

“More Performative in an Engineering Sense”

aws

Search in this guide

English

Sign In

AWS > Documentation > Amazon Redshift > Database Developer Guide

Feedback

- Usage notes
- Examples
- CREATE TABLE AS
- CREATE USER
- CREATE VIEW
- DEALLOCATE
- DECLARE
- DELETE
- DESC DATASHARE
- DROP DATABASE
- DROP DATASHARE

Limits and quotas

Consider the following limits when you create a table.

- There is a limit for the maximum number of tables in a cluster by node type. For more information, see Limits in the *Amazon Redshift Cluster Managemen Guide*.
- The maximum number of characters for a table name is 127.
- The maximum number of columns you can define in a single table is 1,600.
- The maximum number of SORTKEY columns you can define in a single table 400.

“More Performative in an Engineering Sense”

Reasons to Go Long from Wide

- Really “wide” datasets
  - American Community Survey, with revisions, have >1600 questions
- Compute speed + resources
  - Querying from databases is slow - pre-sorting is really helpful

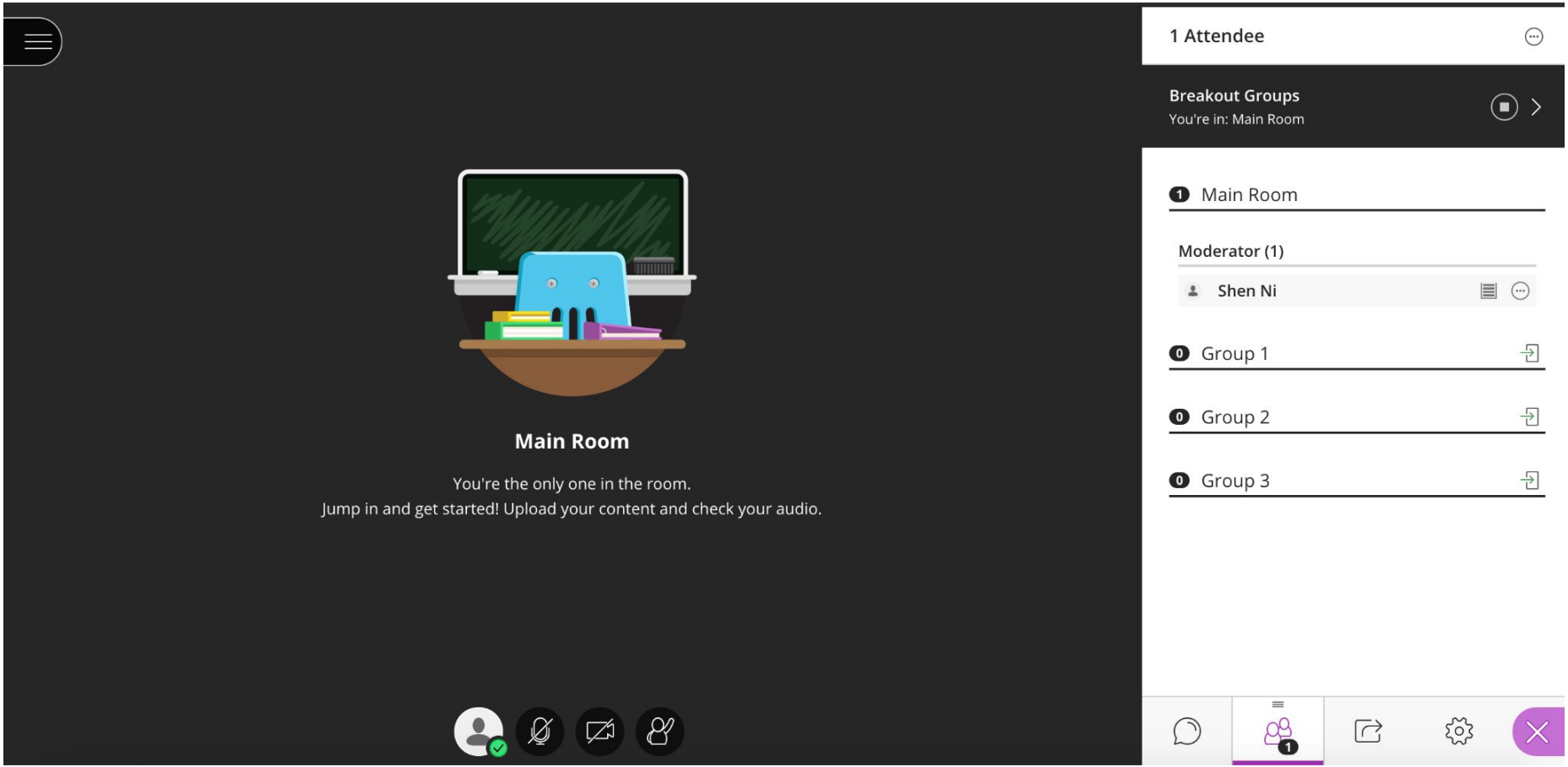
Taking a Look at a New Dataset

Race

Please break into the following groups

Group 1	Group 2	
Anna Arzuaga	Edward Chong	
James Martin	Alex Kwan	
Malley Smith	Meghan Mokate	
	David Segovia	

How to Access Breakout Groups



Taking a Look at a New Dataset

Race by Last Names

Spend 15 minutes to look over the data and discuss

Blackboard >> Data files >> race by last names

a) What data cleaning/enriching might you need to do
b) Is it long or wide data and why
c) What data transformation might you need to do

Data Dictionary - Take a Screenshot

column_name	description
rank	How common is this last name. 1=most common
count	count of individuals with this last name
prop100k	percentage of individuals with this last nane out of 100k of Americans
cum_prop100k	don't worry about this one
pctwhite	percentage of individuals with this last nane, who are white
pctblack	percentage of individuals with this last nane, who are black
pctapi	percentage of individuals with this last nane, who are asian or pacific islander
pctaian	percentage of individuals with this last nane, who are American Indian or Alaskan Native
pct2prace	percentage of individuals with this last nane, who are 2 or more races
pcthispanic	percentage of individuals with this last nane, who are hispanic

Taking a Look at a New Dataset

Discuss

a) What data cleaning/enriching might you need to do

b) Is it long or wide data and why

c) What data transformation might you need to do

“Tidy”

Hadley Wickham

- Each variable forms a column
- Each observation forms a row
- Each type of observational unit forms a table

Tidy Data

Common Issues

- Column headers are values, not variable names
- Multiple variables are stored in one column
- Variables are stored in both rows and columns
- Multiple types of observational units are stored in the same table
- A single observational unit is stored in multiple tables

Tidy Data

Common Issues

- Column headers are values, not variable names
- Multiple variables are stored in one column
- Variables are stored in both rows and columns
- Multiple types of observational units are stored in the same table
- A single observational unit is stored in multiple tables

Important, but we will not cover extensively

Wide → Long

pivot_longer

Wide > Long

pivot_longer

```
pivot_longer(
    data = dataframe which you can usually omit if using pipe logic ,
    col = a column name or a vector of column names to pivot to longer
        format - THIS SHOULD ONLY INCLUDE THE COLUMNS WITH
        VALUES YOU WANT TO PIVOT (in the example below col 1999
        and 2000) AND NOT THE COLUMNS YOU ARE PIVOTING
        AROUND (country) - this can be written as a negation,
    names_to = Name of column to be created which contains the column
        names of gathered columns as values
    values_to = Name of column to be created with the data stored in cell
        values of gathered columns
)
```

Wide > Long

pivot_longer

```
# A tibble: 3 × 3
      country `1999` `2000`
       <fctr>  <int>  <int>
1 Afghanistan    745   2666
2      Brazil  37737  80488
3       China 212258 213766
```

```
# A tibble: 6 × 3
      country  year  cases
       <fctr> <chr>  <int>
1 Afghanistan  1999    745
2      Brazil  1999  37737
3       China  1999 212258
4 Afghanistan  2000   2666
5      Brazil  2000  80488
6       China  2000 213766
```

15-min break

Be back by 7:42pm Central

Wide > Long

Don’t Use Gather

```
gather(
    data = dataframe which you can usually omit if using pipe logic ,
    key = the name you want to give to the column that holds the
        unique keys in the wide table - after you “gather” the unique
        keys will no longer be unique,
    value = the name you want to give to the column that holds the
        “values”
)
```

Wide > Long

Gather's Leap of Faith

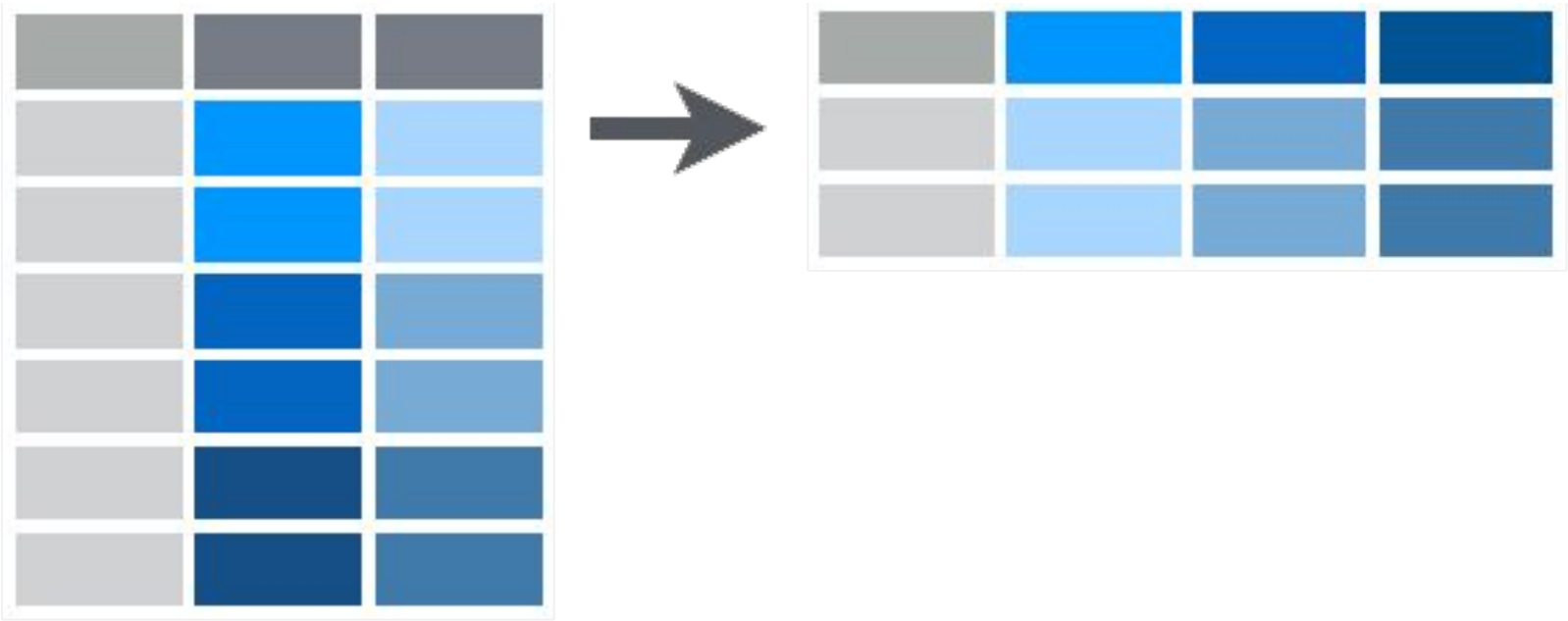
Not specific enough - you cannot specify

- the columns you want to gather "around" nor
- the columns you want to get values out of

Finicky: the column you want to pivot around has to be casted as a factor. If you have multiple columns as factors, you have to jankily drop one

Long → Wide

pivot_wider



Back to the Race Data

Discuss

a) Is it “tidy” and what can be “tidier”
b) Figure out a way to use a method from last week and a “tidy”version of the data to create a table that maps last names to 1 race

Trying It Another Way

Functions

A way to simplify complex operations

Defining a function:

```
function_name <- function(input) {
        Code for complex operations
        }
```

Running a function:

```
Output <- function_name(input)
```

Defining a function:

```
larger_values <- function( a_list ) {
        a_list[
                which(a_list>50)
                ]
        }
```

Running a function:

```
values_in_list_greater_than_50 <-
        larger_values(some_list)
```

Apply

A way to loop a function to every row of a dataframe

	year	conservative	labour	liberal	others
0	1966	253	364	12	1
1	1970	330	287	6	7
2	Feb 1974	297	301	14	18
..	...	...	...	...	...
12	2015	330	232	8	80
13	2017	317	262	12	59
14	2019	365	202	11	72

Apply

A way to loop a function to every row of a dataframe

```
apply(
    X = data or dataframe,
    MARGIN = #1 means apply function to rows 2 means apply to cols,
    FUN = any function you want to apply
)
```

Apply + Function

For this row

		year	conservative	labour	liberal	others
0		1966	253	364	12	1
1		1970	330	287	6	7
2	Feb	1974	297	301	14	18
..		...	...	...	...	...
12		2015	330	232	8	80
13		2017	317	262	12	59
14		2019	365	202	11	72

Apply larger_values and return the values larger than 50

Apply + Function

For this row

	year	conservative	labour	liberal	others
0	1966	253	364	12	1
1	1970	330	287	6	7
2	Feb 1974	297	301	14	18
..	...	...	...	...	...
12	2015	330	232	8	80
13	2017	317	262	12	59
14	2019	365	202	11	72

larger_values returns the values larger than 50

Apply + Function

For this row

	year	conservative	labour	liberal	others
0	1966	253	364	12	1
1	1970	330	287	6	7
2	Feb 1974	297	301	14	18
..	...	...	...	...	...
12	2015	330	232	8	80
13	2017	317	262	12	59
14	2019	365	202	11	72

larger_values returns the values larger than 50

Apply + Function

For this row

	year	conservative	labour	liberal	others
0	1966	253	364	12	1
1	1970	330	287	6	7
2	Feb 1974	297	301	14	18
..	...	...	...	...	...
12	2015	330	232	8	80
13	2017	317	262	12	59
14	2019	365	202	11	72

larger_values returns the values larger than 50

Apply + Function

For this row

	year	conservative	labour	liberal	others
0	1966	253	364	12	1
1	1970	330	287	6	7
2	Feb 1974	297	301	14	18
..	...	...	...	...	...
12	2015	330	232	8	80
13	2017	317	262	12	59
14	2019	365	202	11	72

larger_values returns the values larger than 50

Apply + Function

Pro
Flexible
Easy to read

Con
Slow(er) - better to "vectorize"

Wrap Up

This Is Great, But...

Discuss

What assumptions did we make?

What are the pitfalls of our assumptions?

Document Major Assumptions

Takeaways

Also cover your own butt when others start digging into your work

This Week + Data Wrangling in Conclusion

1. Cleaned salaries data (hourly and salaried)

2. Confirmed analysis goals

3. Data Wrangling
   a. Cleaning + Enrichment + Transformation
   b. Imputed the gender column
   c. Impute the race column