PA 446

Coding for Civic Data Applications

Will be starting at 6:05pm

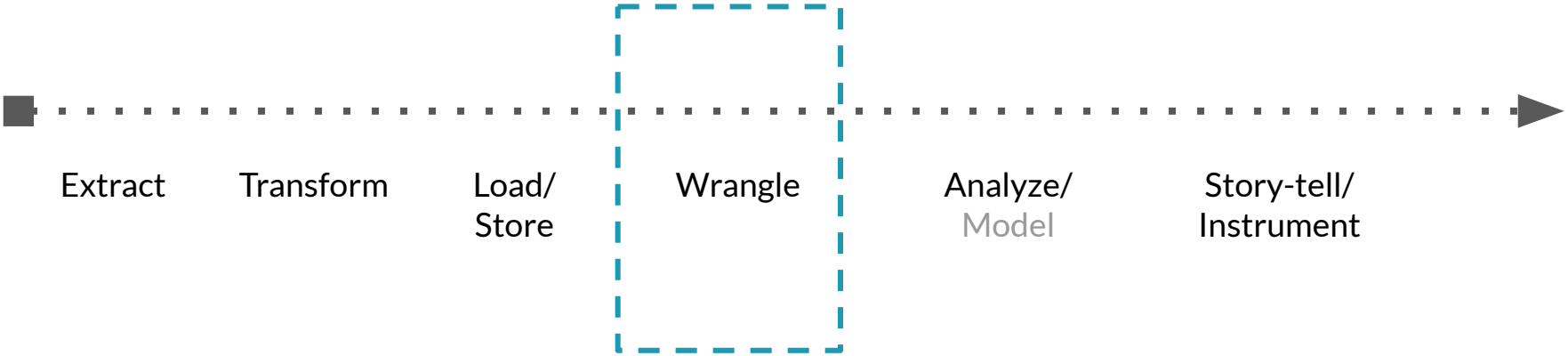
Class #3

Logistics

Course Logistics

- Class next Wednesday (9/15th) moved to 9/16th, 6-9pm
- Homework 1 grades: will be up by end of the week (still figuring out Blackboard)

Data Science “workflow”



Still the focus of today and the next few weeks

Where We Been

1. Cleaned salaries data

2. Confirmed analysis goals

3. Identified major problems
    a. Missing gender + race
    b. Cannot compare salaried and hourly employees

This week

1. Using our new dataset of gender, name and year mapping, we are going to impute the gender of the individuals in our salary dataset

2. Some more data wrangling

Wrangle’s Technical Components

- Cleaning
- Transformation
- Enrichment

Wrangle's Technical Components

- Cleaning
- Transformation
- Enrichment

Takeaways

Last week, we covered parts of cleaning and enrichment

This week, we will go deeper into both of these

We will cover transformation next week

Wrangle's Technical Components

- Cleaning
- Transformation
- Enrichment

Takeaways

Order is not important

Wrangle’s Technical Components

- Enrichment
- Cleaning
- Transformation

Wrangle's Technical Components

- Cleaning
- Enrichment
- Cleaning
- Transformation
- Enrichment
- Cleaning

Takeaways

Cleaning, enrichment and transformation are the basic building blocks. Their order will vary based on your data and your goals

Back to Salary Data

HW 2

Goals

1. Identified major problems
    a. Missing gender + race
    b. Cannot compare salaried and hourly employees

What is IL.TXT

Reading in the New Data

New File Format

What type of file is IL.TXT?

Reading in the New Data

New File Format

[Back to coding]

Reading in the New Data

A Little About CSV's

What exactly are CSV's?

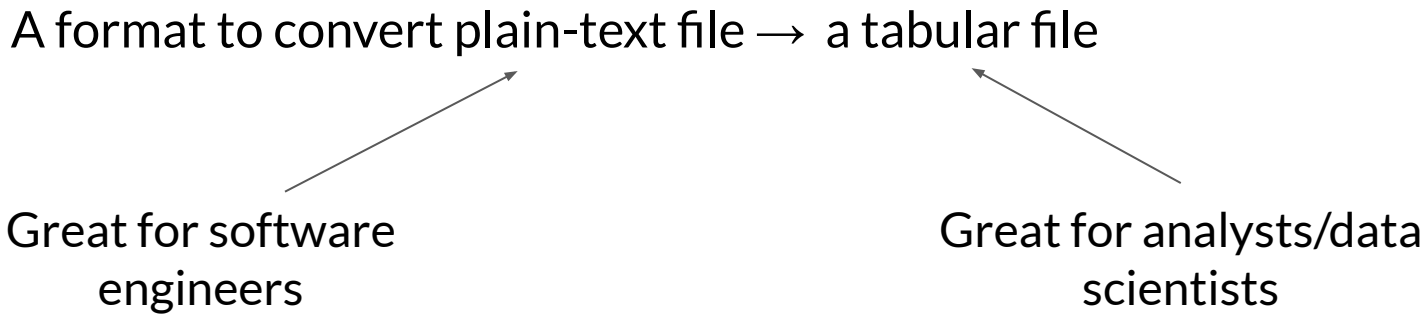
First, We Have to Read in the Data

A Little About CSV's

A format to convert plain-text file → a tabular file

First, We Have to Read in the Data

A Little About CSV's



First, We Have to Read in the Data

A Little About CSV's

How does the file format convert into a table structure?

First, We Have to Read in the Data

A Little About CSV's

First, We Have to Read in the Data

Complication 1: delimiters and escape characters

name , number
Alex , 7
Jeremy , 4
Jack , 1,000

name	,	number
Alex	,	7
Jeremy	,	4

name	number
Alex	7
Jeremy	4

First, We Have to Read in the Data

Complication 1: delimiters and escape characters

[Back to coding]

First, We Have to Read in the Data

Complication 1: delimiters and escape characters

First, We Have to Read in the Data

Complication 1: delimiters and escape characters

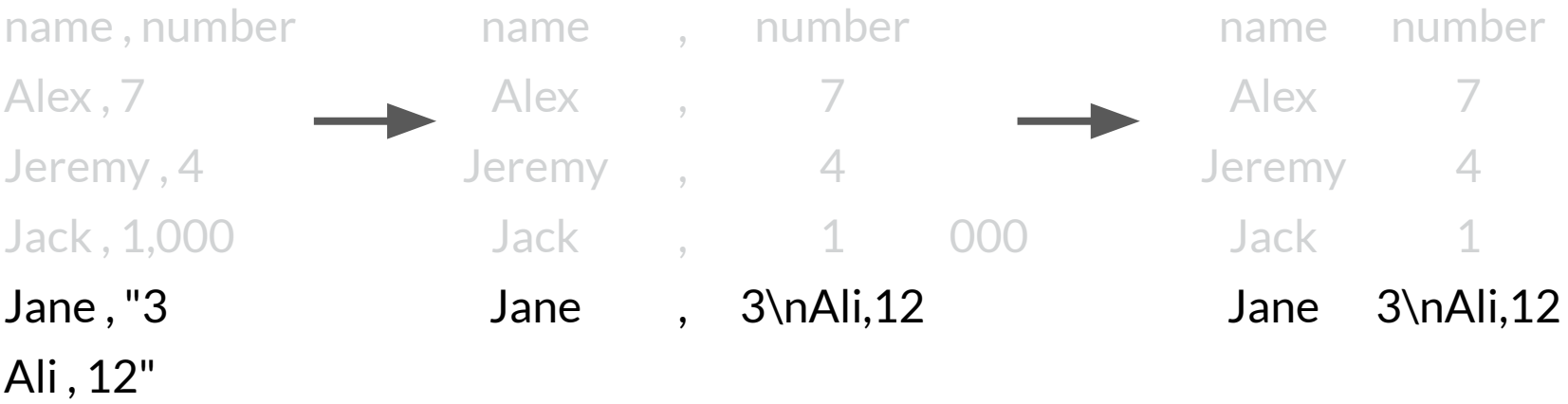
name , number		name	,	number		name	number
Alex , 7	→	Alex	,	7	→	Alex	7
Jeremy , 4		Jeremy	,	4		Jeremy	4
Jack , 1,000		Jack	,	1	000	Jack	1

Jane , "3

Ali , 12"

First, We Have to Read in the Data

Complication 1: delimiters and escape characters



First, We Have to Read in the Data

Complication 1: delimiters and escape characters

, attempts to tell CSV parsers that the text on either side of the comma should be in separate columns. Instead of commas, you can also see tab delimited files

“ ” attempts to tell CSV parsers that the text between the quotes should be treated as a continuous string

First, We Have to Read in the Data

Complication 1: delimiters and escape characters

To “escape” a special character, you need to add an additional quote around it

First, We Have to Read in the Data

Complication 1: delimiters and escape characters

Jane , "3 Ali , 12"	Jane	,	3\nAli,12	Jane	3\nAli,12
Jane, """3" Ali , "12"""	Jane Ali	, ,	\"3 12\"	Jane Ali	\"3 12\"

First, We Have to Read in the Data

Complication 1: delimiters and escape characters

https://www.freeformatter.com/csv-escape.html

Delimiters and Escape characters

In Summary

Various CSV libraries = various heuristics about how to handle intentional and unintentional escape characters = inconsistent CSV parsing across languages

These parsers will usually run with silent bugs: AKA, you data will be wrong but you might not even know it

Be careful when reading CSV!!

What is in IL.TXT?

First, We Have to Read in the Data

Gender Data

[Back to coding]

First, We Have to Read in the Data

Complication 2: CSVs has no metadata

Data columns have no headers

Data columns have no types

First, We Have to Read in the Data

Complication 2: CSVs has no metadata

Data columns have no headers: you might have to label your data

Data columns have no types: if you know a particular column is a boolean, integer etc, don't let R decide for you - specify it

First, We Have to Read in the Data

Complication 2: CSVs has no metadata

Data enrichment

Data columns have no headers: you might have to label your data

Data columns have no types: if you know a particular column is a boolean, integer etc, don't let R decide for you - specify it

First, We Have to Read in the Data

IL.TXT's Content

Popular first names, by gender, for the past 100 years

First, We Have to Read in the Data

IL.TXT's Content, So What

How can we determine the gender of our salary data with this first names, year born, and gender dataset?

Data Joins

Data Joins

Name the Customer Who Ordered Burger

Table: orders

order_id	customer_id	order
332	C775	pizza
334	C772	fries
336	C777	burger
337	C124	pizza

Table: customers

customer_id	Name
C771	Steve G
C772	Nicole B
C773	Michael G
C774	Joe S
C775	Ari H
C776	Kristen W
C777	Jane C

Data Joins

Key Match

Table: orders

order_id	customer_id	order	Name
332	C775	pizza	
334	C772	fries	
336	C777	burger	
337	C124	pizza	

Table: customers

customer_id	Name
C771	Steve G
C772	Nicole B
C773	Michael G
C774	Joe S
C775	Ari H
C776	Kristen W
C777	Jane C

Data Joins

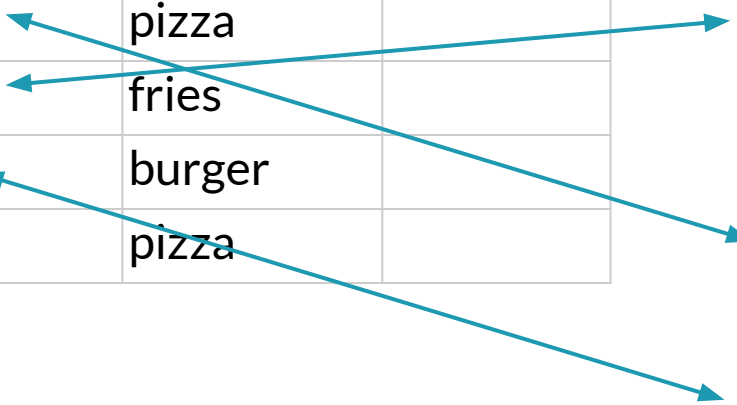
Key Match

Table: orders

order_id	customer_id	order	Name
332	C775	pizza	
334	C772	fries	
336	C777	burger	
337	C124	pizza	

Table: customers

customer_id	Name
C771	Steve G
C772	Nicole B
C773	Michael G
C774	Joe S
C775	Ari H
C776	Kristen W
C777	Jane C



Data Joins

Key Match

Table: orders

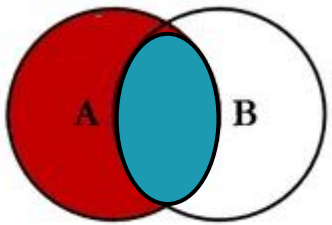
order_id	customer_id	order	Name
332	C775	pizza	Ari H
334	C772	fries	Nicole B
336	C777	burger	Jane C
337	C124	pizza	

Table: customers

customer_id	Name
C771	Steve G
C772	Nicole B
C773	Michael G
C774	Joe S
C775	Ari H
C776	Kristen W
C777	Jane C

Data Joins

Most common: left Join



Left join

Left table (A): orders
Right table (B): customers

“Join” Key for A: customer_id
“Join” Key for B: customer_id

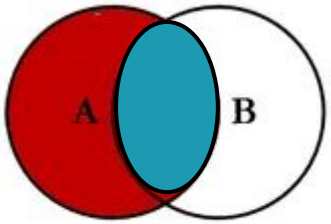
Every row in A is copied into the combined table
Only the rows in B whose customer_id appears in A is copied into the combined table

Data Joins

Key Match

Table: orders

order_id	customer_id	order
332	C775	pizza
334	C772	fries
336	C777	burger
337	C124	pizza



Left join

Table: customers

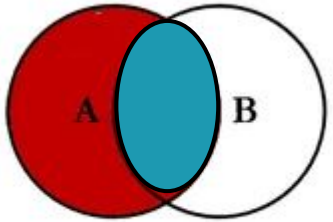
customer_id	Name
C771	Steve G
C772	Nicole B
C773	Michael G
C774	Joe S
C775	Ari H
C776	Kristen W
C777	Jane C

Data Joins

Key Match

Table: combined

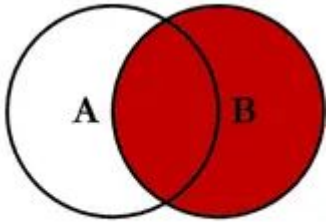
order_id	customer_id	order	Name
332	C775	pizza	Ari H
334	C772	fries	Nicole B
336	C777	burger	Jane C
337	C124	pizza	NA



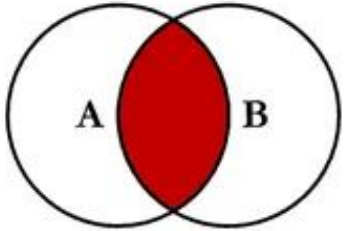
Left join

Data Joins

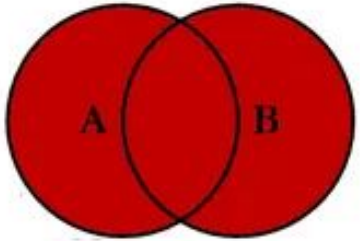
Other joins



Right Join



Inner Join



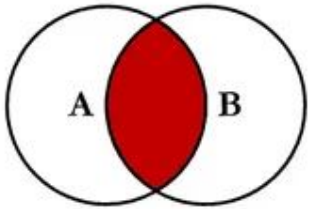
Outer Join

Inner Joins

Example

Table: orders

order_id	customer_id	order
332	C775	pizza
334	C772	fries
336	C777	burger
337	C124	pizza



Inner Join

Table: customers

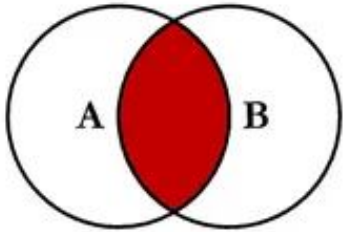
customer_id	Name
C771	Steve G
C772	Nicole B
C773	Michael G
C774	Joe S
C775	Ari H
C776	Kristen W
C777	Jane C

Inner Joins

Example

Table: combined

order_id	customer_id	order	Name
332	C775	pizza	Ari H
334	C772	fries	Nicole B
336	C777	burger	Jane C



Inner Join

Joins: Merge()

merge(x, y, by.x, by.y,all.x,all.y, sort = TRUE)

- x:data frame1.
- y:data frame2.
- by,x, by.y: The names of the columns that are common to both x and y. The default is to use the columns with common names between the two data frames.
- all, all.x, all.y:Logical values that specify the type of merge. The default value is all=FALSE (meaning that only the matching rows are returned)

Joins: _join()

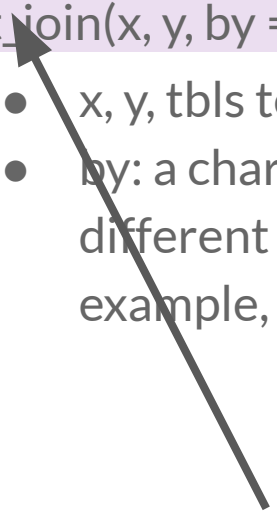
```
inner_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"),  ...)
left_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"),  ...)
left_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"),  ...)
```

- x, y, tbls to join
- by: a character vector of variables to join by. To join by different variables on x and y use a named vector. For example, by = c("a" = "b") will match x.a to y.b.

Joins: _join()

```
inner_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"),  ...)
left_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"),  ...)
left_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"),  ...)
```

- x, y, tbls to join
- by: a character vector of variables to join by. To join by different variables on x and y use a named vector. For example, by = c("a" = "b") will match x.a to y.b.



More similar to SQL + more explicit
All of this is data enrichment

Joins: _join()

[Back to coding]

15-min break

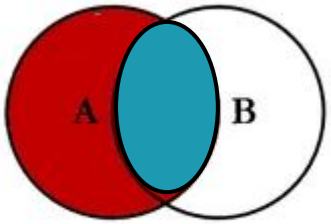
Be back by 7:30pm Central

Joins

Be Careful About Duplicate Keys

Table: orders

order_id	customer_id	order
332	C775	pizza
334	C772	fries
336	C777	burger
337	C124	pizza



Left join

Table: customers

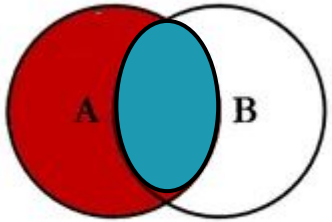
customer_id	Name
C771	Steve G
C772	Nicole B
C773	Michael G
C774	Joe S
C775	Ari H
C776	Kristen W
C777	Jane C

Joins

Be Careful About Duplicate Keys

Table: combined

order_id	customer_id	order	Name
332	C775	pizza	Ari H
334	C772	fries	Nicole B
336	C777	burger	Jane C
337	C124	pizza	NA



Left join

Joins

Be Careful About Duplicate Keys

Table: orders

order_id	customer_id	order
332	C775	pizza
334	C772	fries
336	C777	burger
337	C124	pizza

R not sure which to left join

Table: customers

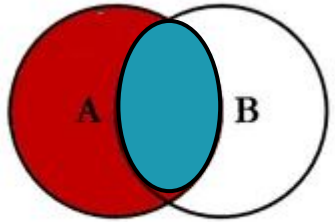
customer_id	Name
C771	Steve G
C772	Nicole B
C773	Michael G
C774	Joe S
C775	Ari H
C776	Kristen W
C777	Jane C
C777	Sherry S

Joins

Be Careful About Duplicate Keys

Table: combined

order_id	customer_id	order	Name
332	C775	pizza	Ari H
334	C772	fries	Nicole B
336	**C777**	**burger**	**Jane C**
336	**C777**	**burger**	**Sherry S**
337	C124	pizza	NA



Left join

So it joins both

Joins

Need to Dedupe First

[Back to coding]

This week in Conclusion

1. We cleaned our new dataset of gender, name and year mapping and we imputed the gender of some individuals in our salary dataset

2. Some more data wrangling