Relational Databases

Dr. Lendie Follett

2022-04-07

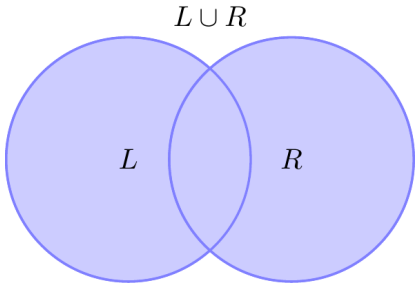
Combining multiple data sources

- Often (usually) the analyses you encounter will require the use of multiple sources of data.
- For future discussions, we may refer to datasets (SAS) and data frames (R) simply as tables.
- The collection of multiple tables are called \textbf{relational data} because it's the relationships between the data sets that convey the meaningful information needed to complete an analysis.

Introduction to set theory

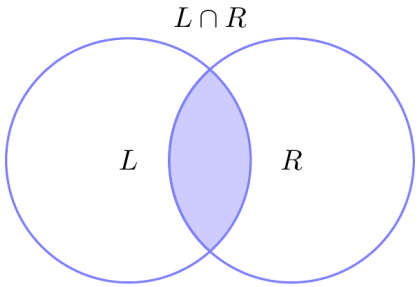
- It will be helpful to think of the relationships between tables in terms of verbs used in set theory
- A **set** is a collection of elements.
  - $L = \{1, 2, 3, 4\}$: "Set L contains elements $1, 2, 3$ and 4"
  - $R = \{1, 3, 5, 6, 7\}$: "Set R contains elements $1, 3, 5, 6$, and 7"
- Why do we talk about set theory in a programming class?
  - It is useful to view tables as sets containing observations.

Introduction to set theory



- The **union** of L and R is the set that contains the elements that are either members of L or members of R or members of both L and R.

Introduction to set theory



- The **intersection** of L and R is the set that contains elements that are members of both L and R.

Keys

- In terms of tables, *membership* is determined by one or more **key** variables (columns). There are two types of keys. In RDS (Wickham), these are defined as:
  - A **primary** key is one or more variables which can uniquely identify an observation in a table.
  - A **foreign** key uniquely identifies an observation in another table.
- A relation between a pair of tables is formed by matching **primary keys** and corresponding **foreign keys**.

Key Examples

Identifying primary keys and foreign keys.

data set: club

student_id	last	club
6422	Jones	stat
3730	Johnson	stat
5792	Johnson	stat
9078	Ma	stat

data set: gpa

student_id	gpa	major
9382	3.3	Stat
6422	3.7	Stat
3730	2.9	Math
5792	4.0	Comp Sci

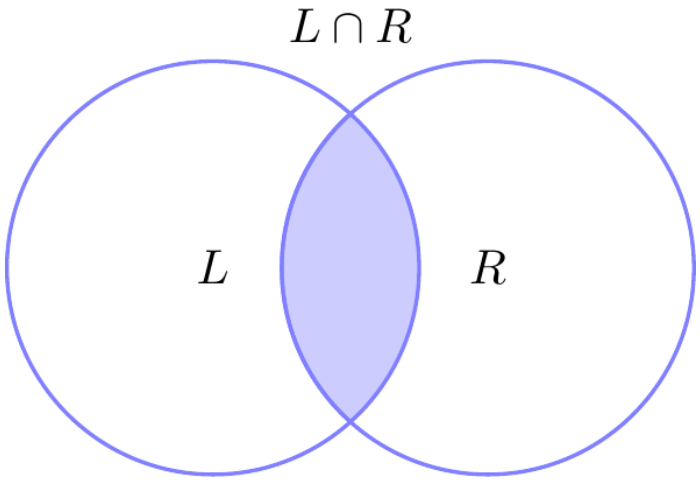
data set: major

major	college
Stat	LAS
Math	LAS
Comp Sci	LAS
Mgmt	Business

Joins

- Joins combine a pair of tables (say, table L and table R) based matching keys to form a single table (say, X).
- The resulting table, X, will contain variables from both tables. The observations present in X depend on the **join type**.
- There are several types of joins. We will cover:
  - Inner join
  - Left join
  - Right join
  - Full join
- The correct type of join will depend on the particular goal.

Inner Join



- Match observations for all matching pairs of keys.
- Important characteristic: observations corresponding to unmatched keys are **not included in the resulting table**.

Need GPAs of stat club members

Inner Join

data set: club

student_id	last	club
6422	Jones	stat
3730	Johnson	stat
5792	Johnson	stat
9078	Ma	stat

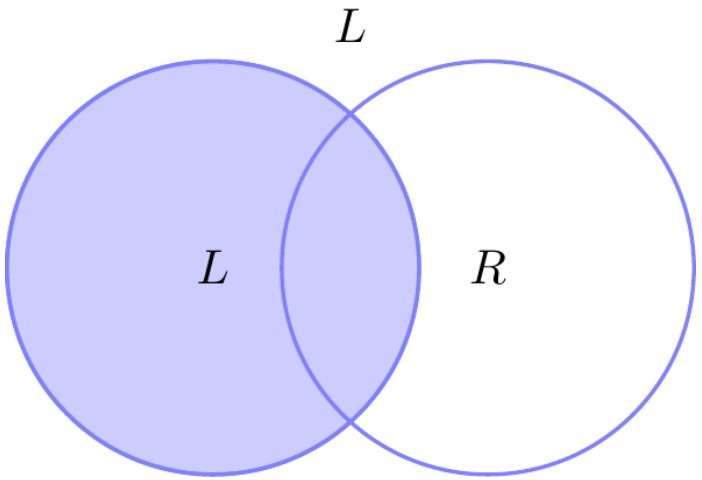
data set: gpa

student_id	gpa	major
9382	3.3	Stat
6422	3.7	Stat
3730	2.9	Math
5792	4.0	Comp Sci

data set: inner

student_id	last	club	gpa	major
3730	Johnson	stat	2.9	Math
5792	Johnson	stat	4.0	Comp Sci
6422	Jones	stat	3.7	Stat

Left Join



- For each observation in L, look for matching key in R.
- Keeps all observations in L.

Add GPA information, wherever possible, to the Stat club member table

Left join

data set: club

student_id	last	club
6422	Jones	stat
3730	Johnson	stat
5792	Johnson	stat
9078	Ma	stat

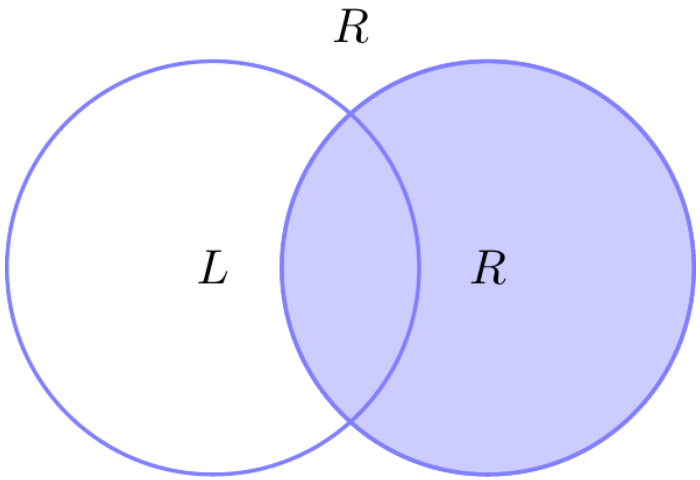
data set: gpa

student_id	gpa	major
9382	3.3	Stat
6422	3.7	Stat
3730	2.9	Math
5792	4.0	Comp Sci

data set: left

student_id	last	club	gpa	major
3730	Johnson	stat	2.9	Math
5792	Johnson	stat	4.0	Comp Sci
6422	Jones	stat	3.7	Stat
9078	Ma	stat		

Right Join



- For each observation in R, look for matching key in L.
- Keeps all observations in R.

Identify whether students in GPA table are stat club members

Right join

data set: club

student_id	last	club
6422	Jones	stat
3730	Johnson	stat
5792	Johnson	stat
9078	Ma	stat

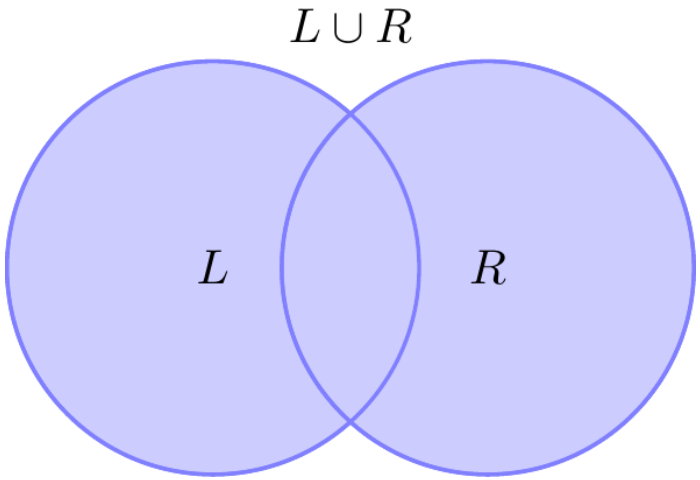
data set: gpa

student_id	gpa	major
9382	3.3	Stat
6422	3.7	Stat
3730	2.9	Math
5792	4.0	Comp Sci

data set: right

student_id	last	club	gpa	major
3730	Johnson	stat	2.9	Math
5792	Johnson	stat	4.0	Comp Sci
6422	Jones	stat	3.7	Stat
9382			3.3	Stat

Full Join



- Match keys wherever possible.
- Keeps all observations in L and keeps all observations in R.

I want to know information about any student who is either in stat club or in the GPA table

Full Join

data set: club

student_id	last	club
6422	Jones	stat
3730	Johnson	stat
5792	Johnson	stat
9078	Ma	stat

data set: gpa

student_id	gpa	major
9382	3.3	Stat
6422	3.7	Stat
3730	2.9	Math
5792	4.0	Comp Sci

data set: full

student_id	last	club	gpa	major
3730	Johnson	stat	2.9	Math
5792	Johnson	stat	4.0	Comp Sci
6422	Jones	stat	3.7	Stat
9078	Ma	stat		
9382			3.3	Stat

Before we begin...

- Real data is messy.
- If the key being used to merge is a character string, make sure values are spelled consistently across data sets.
  - e.g., "a7392" and "A7392" will NOT match in a merge/join.
- If key is a character string that involves all numeric characters, make sure leading 0s haven't been lost.
  - e.g., in Excel, a value of 00073615 will be automatically converted to 73615. If the data was manipulated in Excel before imported to R and SAS, this could present an issue.

Joins in SAS

- Typically, we create a single SAS data set from another single SAS data set using the `set` statement within a data step. e.g,

```
data work.club;
  set s40dat.club;
run;
```

- We can replace that with a `merge` statement to accomplish joining two (or more) SAS data sets.

```
data work.merged;
   merge s40dat.club s40dat.gpa;
   by student_id;
run;
```

- The merge statement will be used in conjunction with a `by` statement that species the common key variable(s).

Joins in SAS

- SAS requires that input data sets be sorted by the common key variable(s) prior to the join.
- Thus, your code sequence may look *something* like:

```
/*Sort both data sets by key(s)*/
proc sort data = data-x;
by list-key(s);
run;

proc sort data = data-y;
by list-key(s);
run;

/*Perform merge*/
data output-data ;
merge data-x<(in = idxx )> data-y<(in = idxy )>;
by list-key(s) ;
<if expression-involving-idxs >;
run;
```

- Finally, the type of join can be determined using the `in =` data set option (this is new to us) along with an appropriate subsetting if statement (also new to us).

Wait, what was that about `in =`?

- The `merge` statement will ask SAS to perform a full join. Always.
- We can perform other types of joins by removing certain observations based on membership.
- The in = data set option creates a temporary binary (0/1) variable indicating whether the observation was a member of that particular data set.
- Then, we can use those temporary variables (named idxx and idxy in the below example) to subset and thus perform the correct join.
- We subset using the `if` statement, which can be used just as we used the `where` statement.

```
/*Perform merge*/
data output-data ;
merge data-x<(in = idxx )> data-y<(in = idxy )>;
by list-key(s) ;
<if expression-involving-idxs >;
run;
```

Joins in SAS

In each of the following merges, we create temporary, binary variables that we name a and b. Then, we use these to filter (or not!) observations from the full join to create an inner, full, left, or right join.

```
/*inner join*/
data merged_data;
merge student(in = a) gpa(in = b);
by student_id;
if a and b;
run;
```

```
/*full join*/
data merged_data;
merge student(in = a) gpa(in = b);
by student_id;
/*NO IF STATEMENT! (keep all observati
run;
```

```
/*left join*/
data merged_data;
merge student(in = a) gpa(in = b);
by student_id;
if a;
run;
```

```
/*right join*/
data merged_data;
merge student(in = a) gpa(in = b);
by student_id;
if b;
run;
```


Case Study

Have: A data set containing NFL draft results. A separate data set containing combine results of college football players.

Want: Find meaningful associations between combine results and draft results. Essentially, what athletic measurements are most important for a desirable draft outcome?

Case Study

Football data: Player stats

A data set containing combine results (various athletic measurements) of college football players from 1999 to 2015.

Year	Player	College	POS	Heightin	Weightlbs	Wonderlic	X40Yard	BenchPress	VertLeapin	BroadJumpin	Shuttle	X3Cone
2014	A.C. Leonard	Tennessee State	TE	74	252		4.5	20	34.0	127		
2015	A.J. Cann	South Carolina	OG	75	313		*5.18	26				
2015	A.J. Derby	Arkansas	TE	76	255		*4.73	15				
2010	A.J. Edds	Iowa	OLB	76	246		4.62	16	33.0	117	4.28	7.19
2011	A.J. Green	Georgia	WR	76	211	10	4.48	18	34.5	126	4.21	6.91
2006	A.J. Hawk	Ohio State	OLB	73	248		4.59	24	40.0	115	3.96	6.82

Case Study

Football data: Draft results

A data set containing draft results from 1999 to 2015

Year	Rnd	Pick	NFLteam	Player	Pos	College	Conf
2015	3	67	Jacksonville Jaguars	A. J. Cann	G	South Carolina	SEC
2007	4	105	Detroit Lions	A. J. Davis	CB	NC State	ACC
2015	6	202	New England Patriots	A. J. Derby	TE	Arkansas	SEC
2010	4	119	Miami Dolphins	A. J. Edds	LB	Iowa	Big Ten
2001	5	155	Philadelphia Eagles	A. J. Feeley	QB	Oregon	Pac-10
2011	1	4	Cincinnati Bengals	A. J. Green	WR	Georgia	SEC