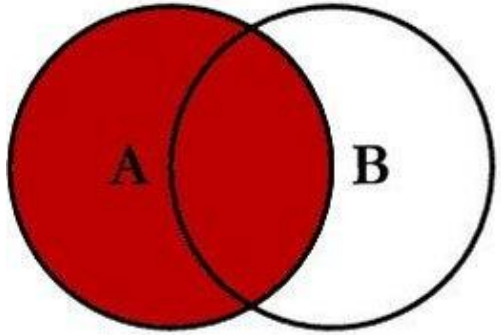
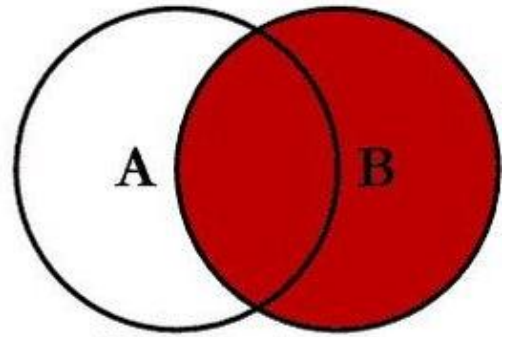
Joins and Melts

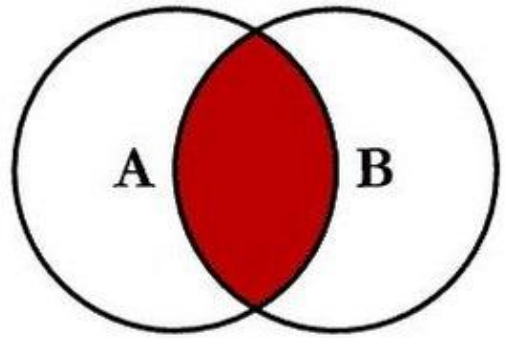
SQL JOINS



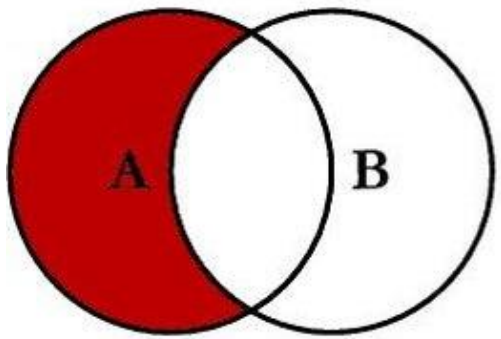

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



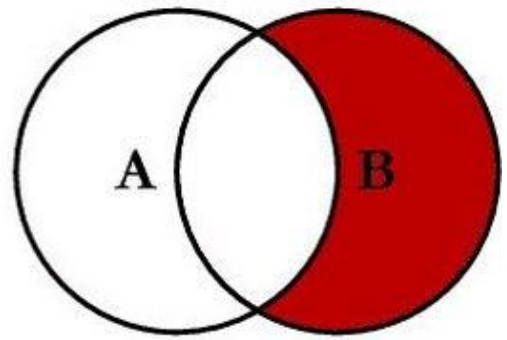

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



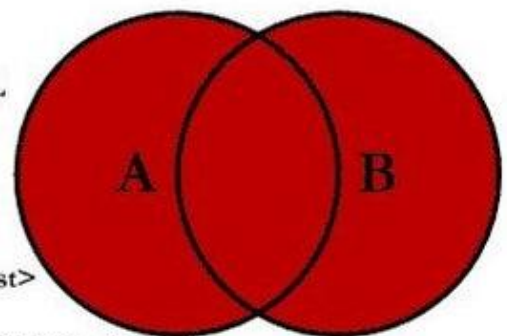

```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



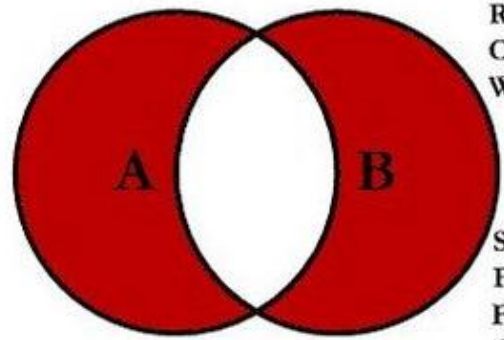

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

Base R: merge()

- We can merge two data frames in R by using the merge() function.
- The data frames must have same column names on which the merging happens.
- merge() in R is similar to database join operation in SQL.
- The different arguments to merge() allow you to perform
  - natural joins
  - as well as left, right, and full outer joins

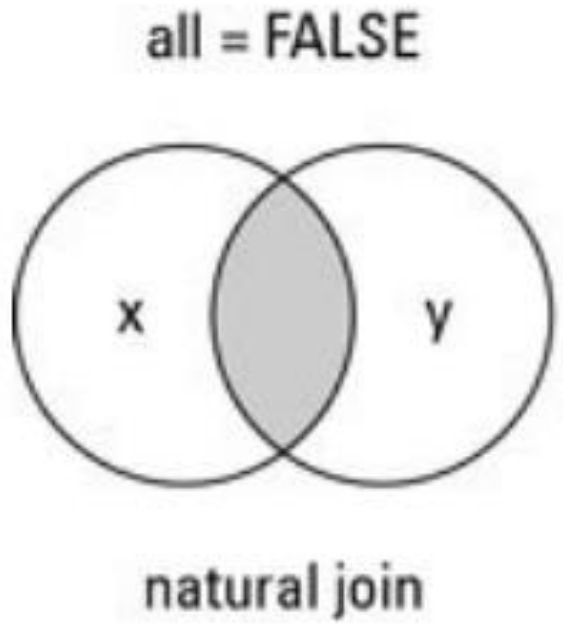
merge() arguments

- **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).

Understanding different types of join

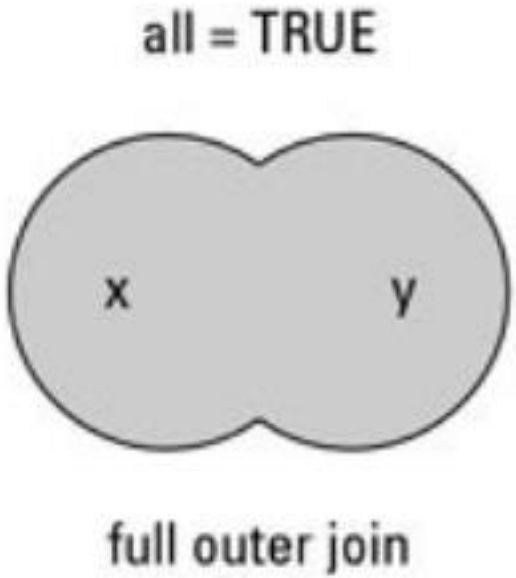
Natural join

- To keep only rows that match from the data frames, specify the argument all=FALSE.



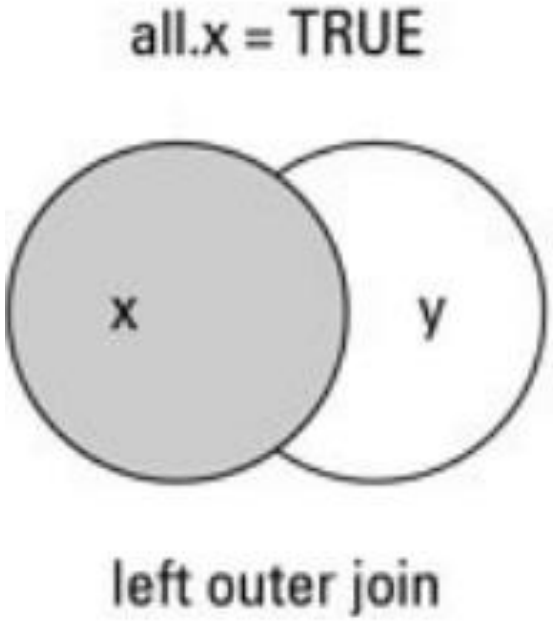
Full outer join

- To keep all rows from both data frames, specify all=TRUE.



Left (right) outer join

- To include all the rows of your data frame x
- Only those from y that match, specify x=TRUE.



Example

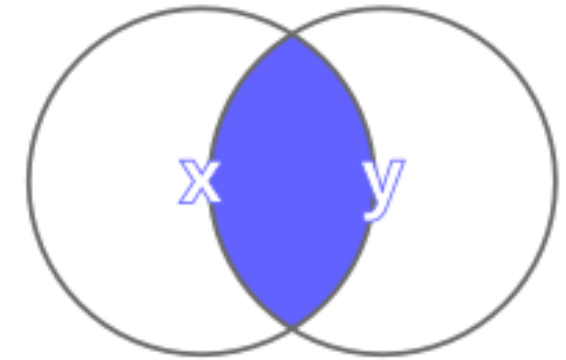
```
# data frame 1
df1 = data.frame(id = c(1:6), area = c(rep(“a", 3), rep(“b", 3)))
df1
# data frame 2
df2 = data.frame(id = c(2, 4, 6), gender = c(rep(“m", 2), rep(“f", 1)))
Df2

# inner join
merge(x=df1, y=df2, by=“id”)
```

Join function in dplyr

- Provides equivalent joins to merge()
- And fills in the gaps with SQL joins

semi_join()

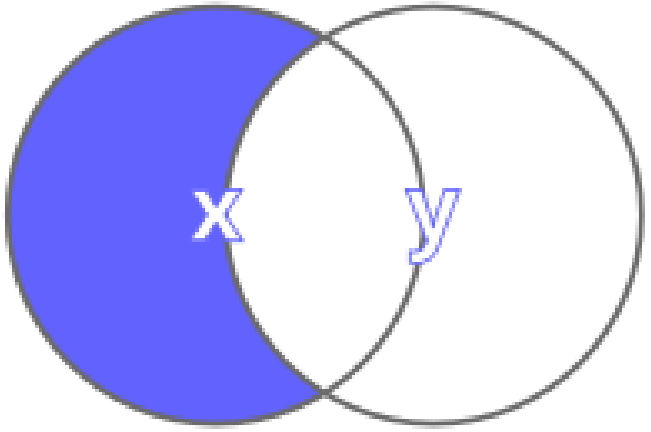
- Return all rows from x where there are matching values in y, keeping just columns from x. A semi join differs from an inner join because an inner join will return one row of x for each matching row of y, where a semi join will never duplicate rows of x. This is a filtering join.



(never duplicate rows of x)

anti_join()

- Return all rows from x where there are not matching values in y, keeping just columns from x. This is a filtering join.

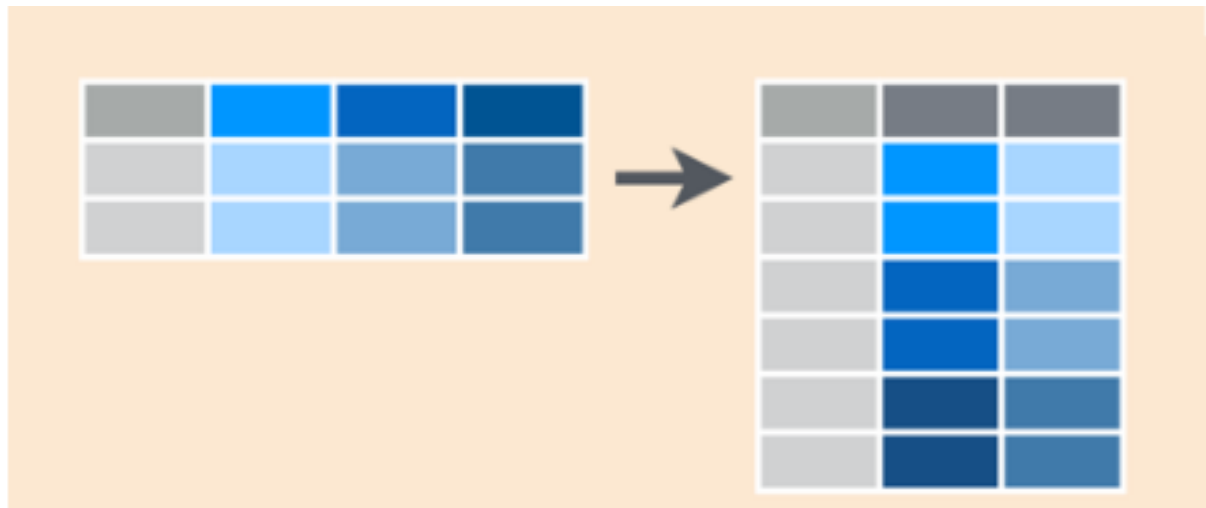


Melting and Casting

- Changing the shape of the data set
- Useful for plotting in ggplot2 (requires ‘log’ data)
- Provides consistency
- Allows combining of data sets

melt()

- Takes data in wide format and stacks a set of columns into a single column of data.
- Specify a data frame, the id variables (which will be left at their settings) and the measured variables (columns of data) to be stacked.
- Default assumption on measured variables is that it is all columns that are not specified as id variables.



cast() or dcast()

- Aggregation occurs when the combination of variables in the **cast** function does not identify Individual observations.
- In this case cast function reduces the multiple values to a single one by summing up the values in the **value** column.