7. Relational operations with dplyr

Data Science for OR - J. Duggan

Relational Data with dplyr

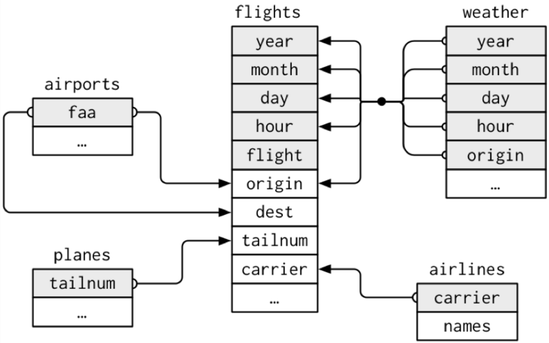
- Typically, data analysis involves many tables of data that must be combined to answer questions
- Collectively, multiple tables of data are called relational data
- Relations are always defined between a pair of tables
- See tibbles **x** and **y**

```
## # A tibble: 3 x 2
##     key val_x
##   <dbl> <chr>
## 1     1 x1
## 2     2 x2
## 3     3 x3
```

```
## # A tibble: 3 x 2
##     key val_y
##   <dbl> <chr>
## 1     1 y1
```

Keys

- The variables used to connect each pair of tables are called keys
- A key is a variable (or set of variables) that uniquely identifies an observation
- There are two types of keys:
  - A primary key uniquely identifies an observation in its own table
  - A foreign key uniquely identifies an observation in another table.



Mutating Joins

- Allows you to combine variables from two tables
- First matches observations by their keys, and then copies across variables from one table to another
- Similar to mutate(), the join functions add variables to the right
- Types
  - Inner Join
  - Left Join
  - Right Join
  - Full Join

Inner Joins

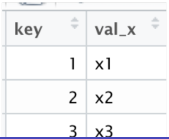
- Matches pairs of observations when their keys are equal
- Unmatched rows are not included in the result

```
inner_join(x,y)
```

```
## Joining, by = "key"
```

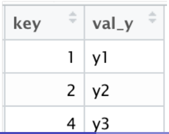
```
## # A tibble: 2 x 3
##     key val_x val_y
##   <dbl> <chr> <chr>
## 1     1 x1    y1
## 2     2 x2    y2
```

x



key	val_x
1	x1
2	x2
3	x3

y



key	val_y
1	y1
2	y2
4	y3

Left Join

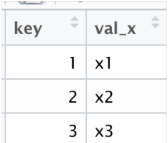
A left join keeps all observations in x

```
left_join(x,y)
```

```
## Joining, by = "key"
```

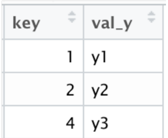
```
## # A tibble: 3 x 3
##     key val_x val_y
##   <dbl> <chr> <chr>
## 1     1 x1    y1
## 2     2 x2    y2
## 3     3 x3    <NA>
```

x



key	val_x
1	x1
2	x2
3	x3

y



key	val_y
1	y1
2	y2
4	y3

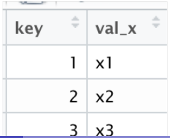
Right Join

A right join keeps all observations in y

```r
right_join(x,y)
```

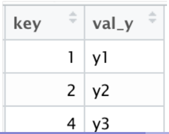
```
## Joining, by = "key"

## # A tibble: 3 x 3
##     key val_x val_y
##   <dbl> <chr> <chr>
## 1     1 x1    y1
## 2     2 x2    y2
## 3     4 <NA>  y3
```

x



key	val_x
1	x1
2	x2
3	x3

y



key	val_y
1	y1
2	y2
4	y3

Full Join

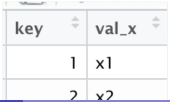
A full join keeps all observations in x and y

```
full_join(x,y)
```

```
## Joining, by = "key"
```

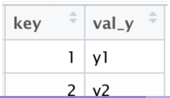
```
## # A tibble: 4 x 3
##     key val_x val_y
##   <dbl> <chr> <chr>
## 1     1 x1    y1
## 2     2 x2    y2
## 3     3 x3    <NA>
## 4     4 <NA>  y3
```

x



key	val_x
1	x1
2	x2

y



key	val_y
1	y1
2	y2

Filtering Joins

Match observations in the same way as mutating joins, but affect the observations, not the variables. Two types:

- semi_join(x,y) keeps all observations in x that have a match in y
- anti_join(x,y), drops all observations in x that have a match in y.

Semi Joins

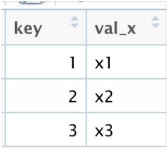
Keeps all observations in x that have a match in y

```
semi_join(x,y)
```

```
## Joining, by = "key"
```

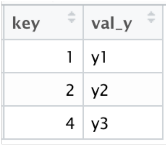
```
## # A tibble: 2 x 2
##     key val_x
##   <dbl> <chr>
## 1     1 x1
## 2     2 x2
```

x



key	val_x
1	x1
2	x2
3	x3

y



key	val_y
1	y1
2	y2
4	y3

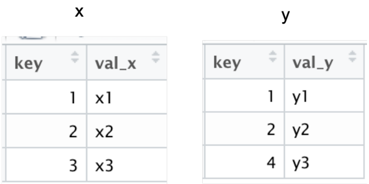
Anti Joins

Drops all observations in x that have a match in y.

```
anti_join(x,y)
```

```
## Joining, by = "key"
```

```
## # A tibble: 1 x 2
##     key val_x
##   <dbl> <chr>
## 1     3 x3
```



x

key	val_x
1	x1
2	x2
3	x3

y

key	val_y
1	y1
2	y2
4	y3

Figure 5: Tables x and y

Challenge 3.1

- Filter out incomplete flights from the dataset
- Join the flights data to the weather data
- Filter out missing temperature values
- Plot the relationship between temperatures and departure delays, facet by origin
- Use a sample of 10000 for the plot, with seed 99.

Summary

- dplyr - support relational data operations
- Mutating Joins
  - **inner_join()**
  - **left_join()**
  - **right_join**
  - **full_join()**
- Filtering Joins
  - **semi_join()**
  - **anti_join()**
- Important for exploratory data analysis and modelling