Joining Data

Biostatistics 140.776

Joining Data

The `dplyr` package provides a set of functions for joining two data frames into a single data frame based on a set of key columns.

- `left_join()`
- `right_join()`
- `inner_join()`

There are other functions for joining but they are less commonly used.

Left Join

```
library(dplyr)



dat <- tibble(
        id = rep(c("a", "b", "c"), each = 3),
        visit = rep(0:2, 3),
        outcome = rnorm(3 * 3, 3)
)
```

Left Join

```
dat
# A tibble: 9 x 3
  id    visit outcome
  <chr> <int>   <dbl>
1 a         0    3.75
2 a         1    2.65
3 a         2    2.39
4 b         0    1.43
5 b         1    2.94
6 b         2    2.07
7 c         0    3.94
8 c         1    3.26
9 c         2    1.20
```

Left Join

```
subjects <- tibble(
        id = c("a", "b", "c"),
        house = c("detached", "rowhouse", "rowhouse"),
)
```

```
subjects
# A tibble: 3 x 2
  id    house
  <chr> <chr>
1 a     detached
2 b     rowhouse
3 c     rowhouse
```

Left Join

```
left_join(dat, subjects, by = "id")
# A tibble: 9 x 4
  id    visit outcome house
  <chr> <int>   <dbl> <chr>
1 a         0    3.75 detached
2 a         1    2.65 detached
3 a         2    2.39 detached
4 b         0    1.43 rowhouse
5 b         1    2.94 rowhouse
6 b         2    2.07 rowhouse
7 c         0    3.94 rowhouse
8 c         1    3.26 rowhouse
9 c         2    1.20 rowhouse
```

Left Join

```
subjects <- tibble(
        id = c("a", "b", "c"),
        visit = c(0, 1, 0),
        house = c("detached", "rowhouse", "rowhouse"),
)
```

```
subjects
# A tibble: 3 x 3
  id    visit house
  <chr> <dbl> <chr>
1 a         0 detached
2 b         1 rowhouse
3 c         0 rowhouse
```

Left Join

```
left_join(dat, subjects, by = c("id", "visit"))
# A tibble: 9 x 4
  id    visit outcome house
  <chr> <dbl>   <dbl> <chr>
1 a         0    3.75 detached
2 a         1    2.65 <NA>
3 a         2    2.39 <NA>
4 b         0    1.43 <NA>
5 b         1    2.94 rowhouse
6 b         2    2.07 <NA>
7 c         0    3.94 rowhouse
8 c         1    3.26 <NA>
9 c         2    1.20 <NA>
```

Left Join

```
subjects <- tibble(
        id = c("b", "c"),
        visit = c(1, 0),
        house = c("rowhouse", "rowhouse"),
)
```

```
subjects
# A tibble: 2 x 3
  id    visit house
  <chr> <dbl> <chr>
1 b         1 rowhouse
2 c         0 rowhouse
```

Left Join

```
left_join(dat, subjects, by = c("id", "visit"))
# A tibble: 9 x 4
  id    visit outcome house
  <chr> <dbl>   <dbl> <chr>
1 a         0    3.75 <NA>
2 a         1    2.65 <NA>
3 a         2    2.39 <NA>
4 b         0    1.43 <NA>
5 b         1    2.94 rowhouse
6 b         2    2.07 <NA>
7 c         0    3.94 rowhouse
8 c         1    3.26 <NA>
9 c         2    1.20 <NA>
```

Inner Join

```
inner_join(dat, subjects, by = c("id", "visit"))
# A tibble: 2 x 4
  id    visit outcome house
  <chr> <dbl>   <dbl> <chr>
1 b         1    2.94 rowhouse
2 c         0    3.94 rowhouse
```

Right Join

```
right_join(dat, subjects, by = c("id", "visit"))
# A tibble: 2 x 4
  id    visit outcome house
  <chr> <dbl>   <dbl> <chr>
1 b         1    2.94 rowhouse
2 c         0    3.94 rowhouse
```

Right Join

```
right_join(subjects, dat, by = c("id", "visit"))
# A tibble: 9 x 4
  id    visit house    outcome
  <chr> <dbl> <chr>      <dbl>
1 a         0 <NA>        3.75
2 a         1 <NA>        2.65
3 a         2 <NA>        2.39
4 b         0 <NA>        1.43
5 b         1 rowhouse    2.94
6 b         2 <NA>        2.07
7 c         0 rowhouse    3.94
8 c         1 <NA>        3.26
9 c         2 <NA>        1.20
```

Summary

- `left_join()` is useful for merging a "large" data frame with a "smaller" one while retaining all the rows of the "large" data frame
- `inner_join()` gives you the intersection of the rows between two data frames
- `right_join()` is like `left_join()` with the arguments reversed (likely only useful at the end of a pipeline)