Data Science for Everyone – Data Wrangling – Joins 1

Dr. Ab Mosca (they/them)

Slides based off slides courtesy of Jordan Crouser (https://jcrouser.github.io/)

Plan for Today

- Relational data
- Inner joins

Single Tables

- Until now, we've completed analyses by look at data in one table
- Often, one table will not have all the data we need for an analysis
- In these cases, we use *multiple tables that are related*

Multiple Tables

Relational Data

- Data from two or more tables that is *related*
- Ex. `nycflights13` data in R

```
library(nycflights13)
```

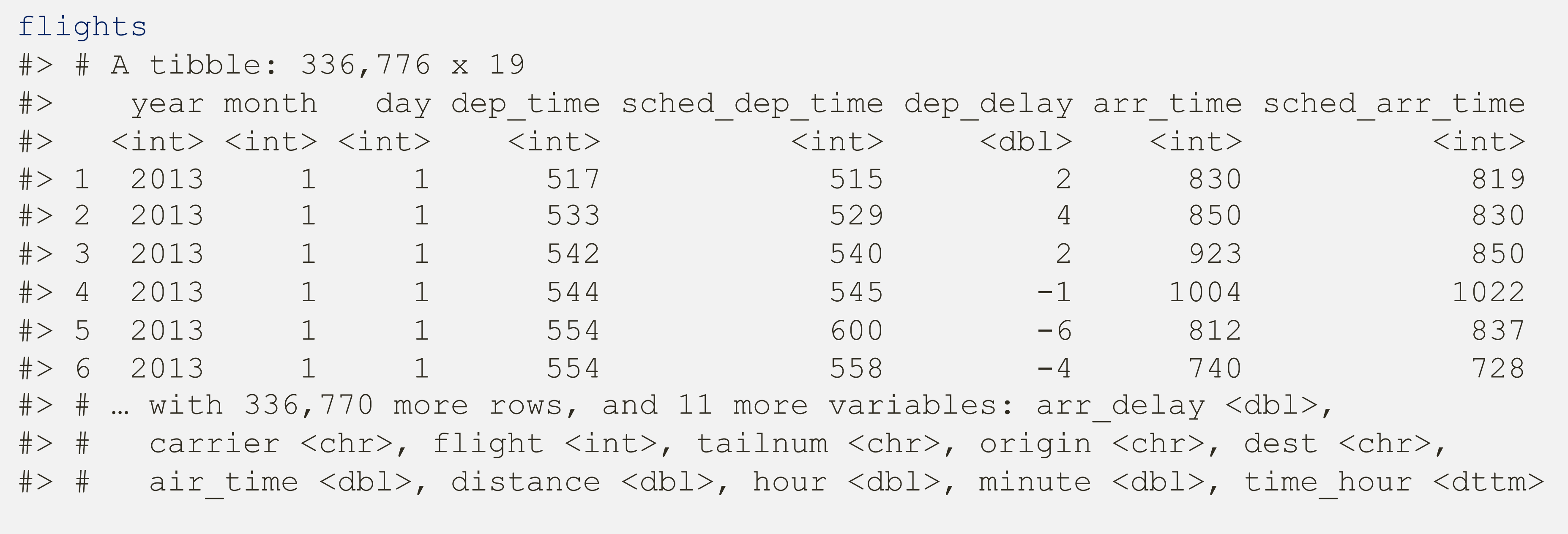
```
## Warning: package
'nycflights13' was built
under R version 3.6.2
     •  flights
     •  airports
     •  airlines
     •  planes
     •  weather
```

- Dataset (`nycflights13`) is made up of multiple tables of data
- All tables have data related to NYC flights in 2013
- Some tables repeat columns

Multiple Tables

nycflights13 Data

- flights, airports, airlines, planes, weather

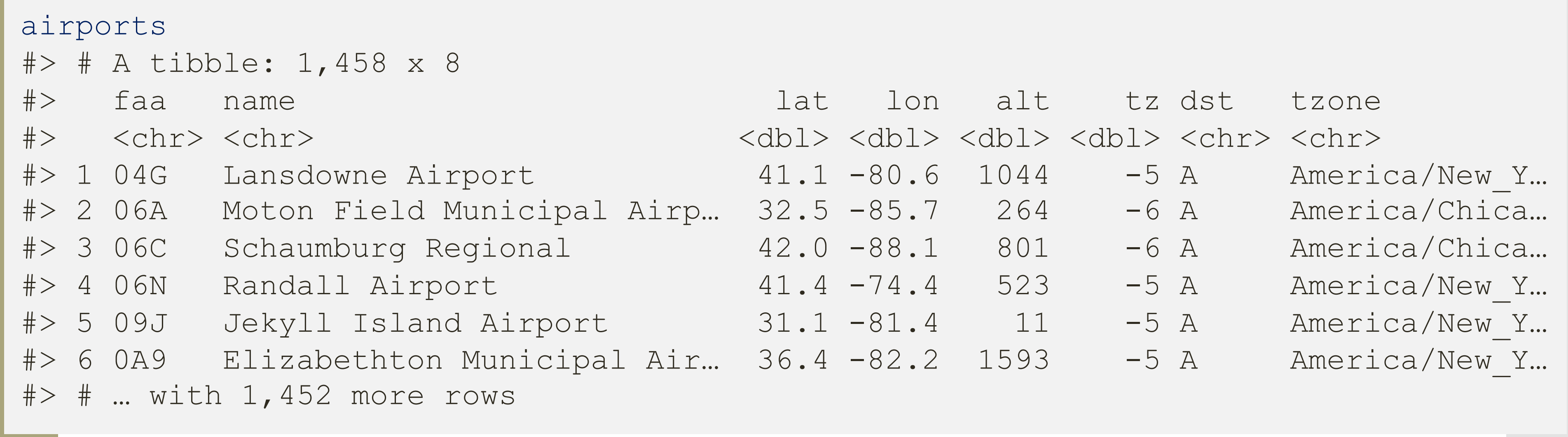


```
flights
#> # A tibble: 336,776 x 19
#>     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
#> 1   2013     1     1      517            515         2      830            819
#> 2   2013     1     1      533            529         4      850            830
#> 3   2013     1     1      542            540         2      923            850
#> 4   2013     1     1      544            545        -1     1004           1022
#> 5   2013     1     1      554            600        -6      812            837
#> 6   2013     1     1      554            558        -4      740            728
#> # … with 336,770 more rows, and 11 more variables: arr_delay <dbl>,
#> #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
#> #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

Multiple Tables

nycflights13 Data

- flights, airports, airlines, planes, weather



```
airports
#> # A tibble: 1,458 x 8
#>   faa   name                              lat   lon   alt    tz dst   tzone       
#>   <chr> <chr>                           <dbl> <dbl> <dbl> <dbl> <chr> <chr>       
#> 1 04G   Lansdowne Airport                41.1 -80.6  1044    -5 A     America/New_Y…
#> 2 06A   Moton Field Municipal Airp…      32.5 -85.7   264    -6 A     America/Chica…
#> 3 06C   Schaumburg Regional              42.0 -88.1   801    -6 A     America/Chica…
#> 4 06N   Randall Airport                  41.4 -74.4   523    -5 A     America/New_Y…
#> 5 09J   Jekyll Island Airport            31.1 -81.4    11    -5 A     America/New_Y…
#> 6 0A9   Elizabethton Municipal Air…      36.4 -82.2  1593    -5 A     America/New_Y…
#> # … with 1,452 more rows
```

Multiple Tables

nycflights13 Data

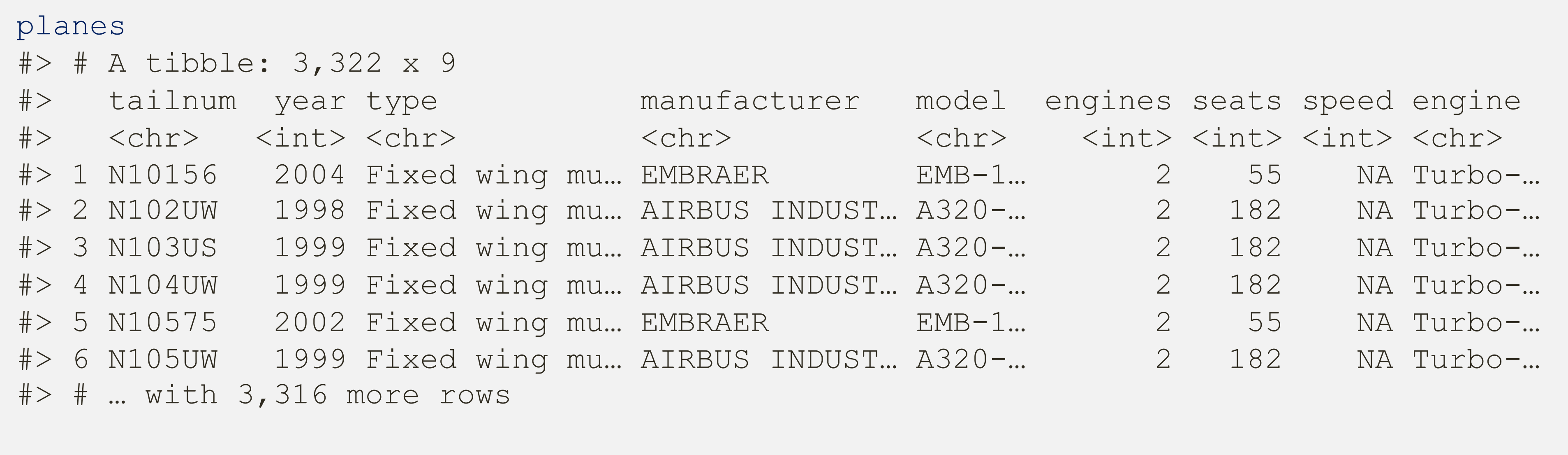
- flights, airports, airlines, planes, weather

```
airlines
#> # A tibble: 16 x 2
#>   carrier name
#>   <chr>   <chr>
#> 1 9E      Endeavor Air Inc.
#> 2 AA      American Airlines Inc.
#> 3 AS      Alaska Airlines Inc.
#> 4 B6      JetBlue Airways
#> 5 DL      Delta Air Lines Inc.
#> 6 EV      ExpressJet Airlines Inc.
#> # … with 10 more rows
```

Multiple Tables

nycflights13 Data

- flights, airports, airlines, planes, weather



```
planes
#> # A tibble: 3,322 x 9
#>   tailnum  year type              manufacturer   model  engines seats speed engine
#>   <chr>   <int> <chr>          <chr>          <chr>    <int> <int> <int> <chr>
#> 1 N10156   2004 Fixed wing mu… EMBRAER        EMB-1…       2    55    NA Turbo-…
#> 2 N102UW   1998 Fixed wing mu… AIRBUS INDUST… A320-…       2   182    NA Turbo-…
#> 3 N103US   1999 Fixed wing mu… AIRBUS INDUST… A320-…       2   182    NA Turbo-…
#> 4 N104UW   1999 Fixed wing mu… AIRBUS INDUST… A320-…       2   182    NA Turbo-…
#> 5 N10575   2002 Fixed wing mu… EMBRAER        EMB-1…       2    55    NA Turbo-…
#> 6 N105UW   1999 Fixed wing mu… AIRBUS INDUST… A320-…       2   182    NA Turbo-…
#> # … with 3,316 more rows
```

Multiple Tables

nycflights13 Data

- flights, airports, airlines, planes, weather

```
weather
#> # A tibble: 26,115 x 15
#>   origin  year month   day  hour  temp  dewp humid wind_dir wind_speed wind_gust
#>   <chr>  <int> <int> <int> <int> <dbl> <dbl> <dbl>    <dbl>      <dbl>     <dbl>
#> 1 EWR     2013     1     1     1  39.0  26.1  59.4      270      10.4         NA
#> 2 EWR     2013     1     1     2  39.0  27.0  61.6      250       8.06        NA
#> 3 EWR     2013     1     1     3  39.0  28.0  64.4      240      11.5         NA
#> 4 EWR     2013     1     1     4  39.9  28.0  62.2      250      12.7         NA
#> 5 EWR     2013     1     1     5  39.0  28.0  64.4      260      12.7         NA
#> 6 EWR     2013     1     1     6  37.9  28.0  67.2      240      11.5         NA
#> # … with 26,109 more rows, and 4 more variables: precip <dbl>, pressure <dbl>,
#> #   visib <dbl>, time_hour <dttm>
```

Multiple Tables

nycflights13 Data

flights
year
month
day
dep_time
sched_dep_time
dep_delay
arr_time
sched_arr_time
arr_delay
carrier
flight
talinum
origin
dest
air_time
distance
hour
minute
time_hour

airports
faa
name
lat
lon
alt
tz
dst
tzone

airlines
carrier
name

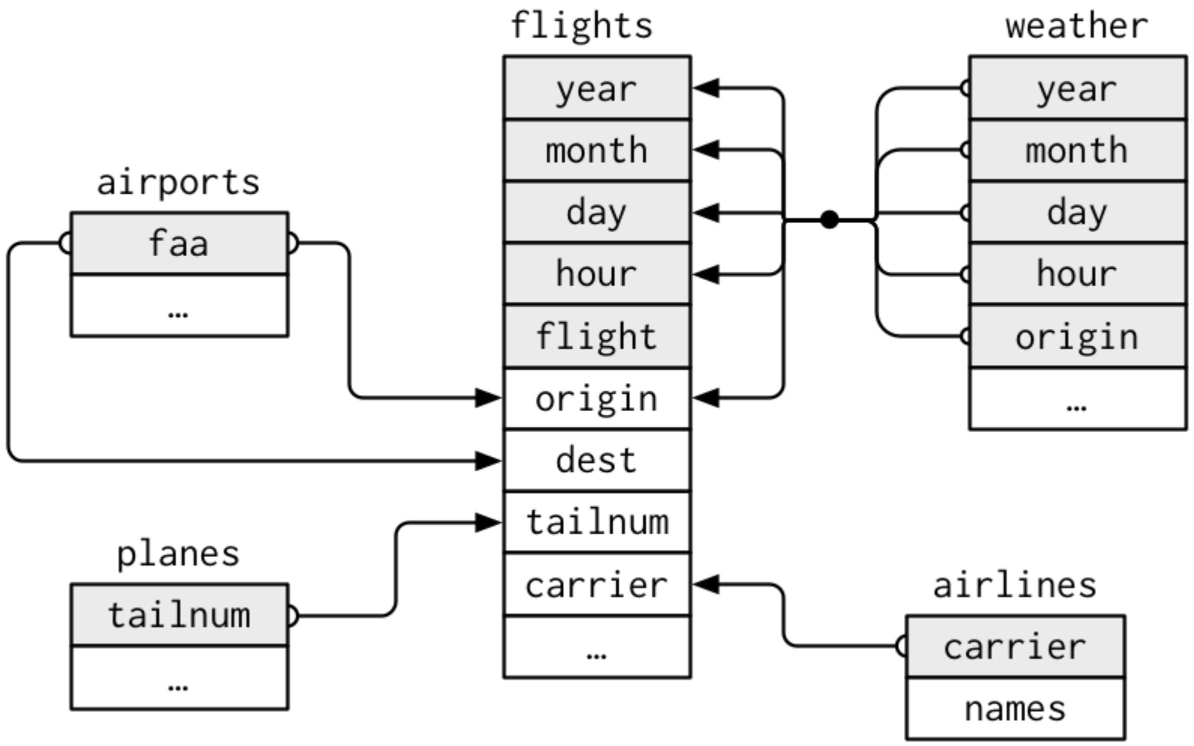
planes
tailnum
year
type
manufacturer
model
engines
seats
speed
engine

weather
origin
year
month
day
hour
temp
dewp
humid
wind_dir
wind_speed
wind_gust
precip
pressure
visib
time_hour

Work with the person next to you to find which columns are shared between these different tables

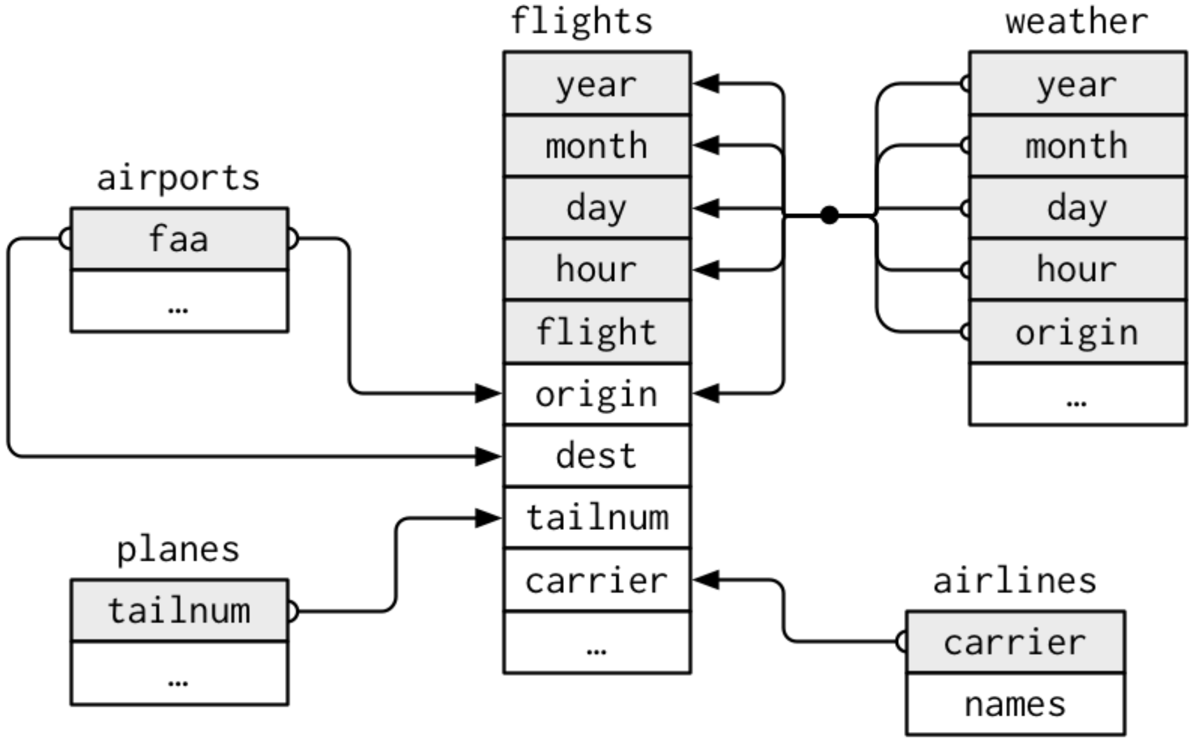
Multiple Tables

nycflights13 Data



Multiple Tables

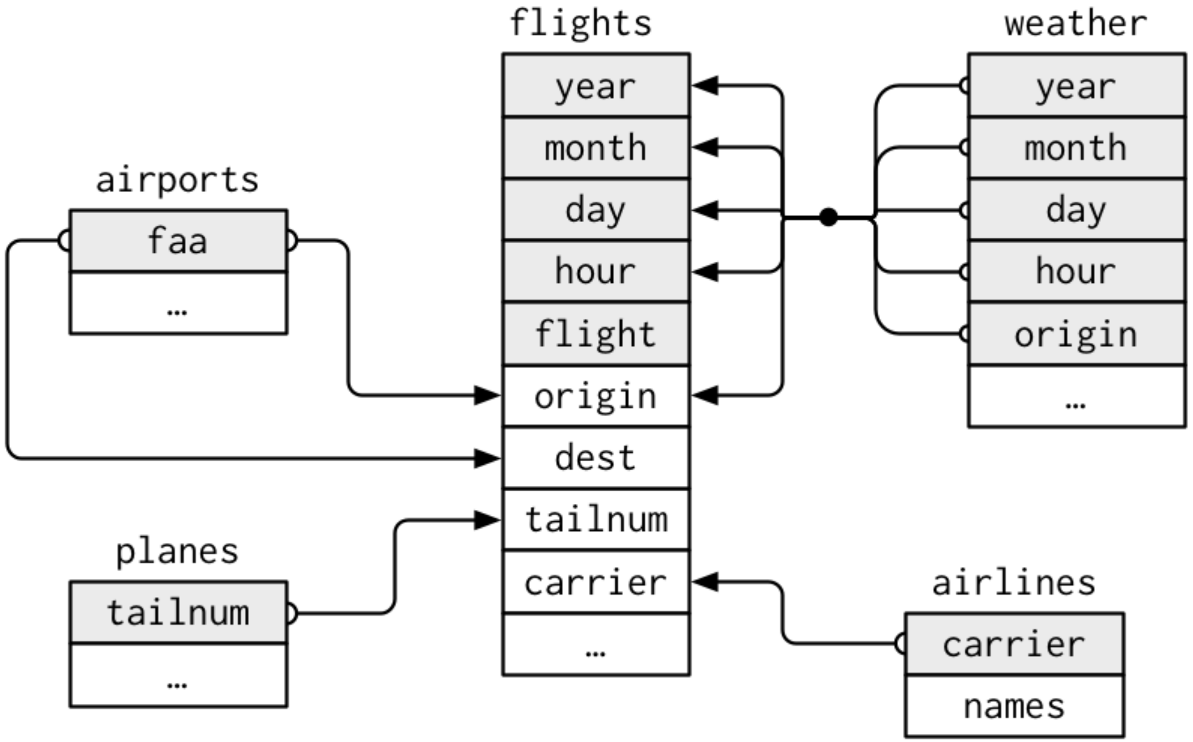
nycflights13 Data



- We use shared columns to join (i.e. connect / merge) tables
- Ex. We could join the planes and flights tables on talinum, their shared column

Multiple Tables

nycflights13 Data



- We use shared columns to join (i.e. connect / merge) tables
- Ex. We could join the planes and flights tables on talinum, their shared column

What other shared columns could we join on?

Joins

- **Join** is the word for connecting or merging two data tables
- We join tables on shared columns, which we call the **key**
- Ex.

Table_X

ID	DataX
1	x1
2	x2
3	x3

Table_Y

ID	DataY
1	y1
2	y2
4	y3

Joins

- **Join** is the word for connecting or merging two data tables
- We join tables on shared columns, which we call the **key**
- Ex.

Table_X

ID	DataX
1	x1
2	x2
3	x3

Table_Y

ID	DataY
1	y1
2	y2
4	y3

Joins

- **Join** is the word for connecting or merging two data tables
- We join tables on shared columns, which we call the **key**
- Ex.

Table_X

ID	DataX
1	x1
2	x2
3	x3

Table_Y

ID	DataY
1	y1
2	y2
4	y3

Joins

- **Join** is the word for connecting or merging two data tables
- We join tables on shared columns, which we call the **key**
- Ex.

Table_X

ID	DataX
1	x1
2	x2
3	x3

Table_Y

ID	DataY
1	y1
2	y2
4	y3

How could we handle this?

Joins

- **Join** is the word for connecting or merging two data tables
- We join tables on shared columns, which we call the **key**
- Ex.

Table_X

ID	DataX
1	x1
2	x2
3	x3

Table_Y

ID	DataY
1	y1
2	y2
4	y3

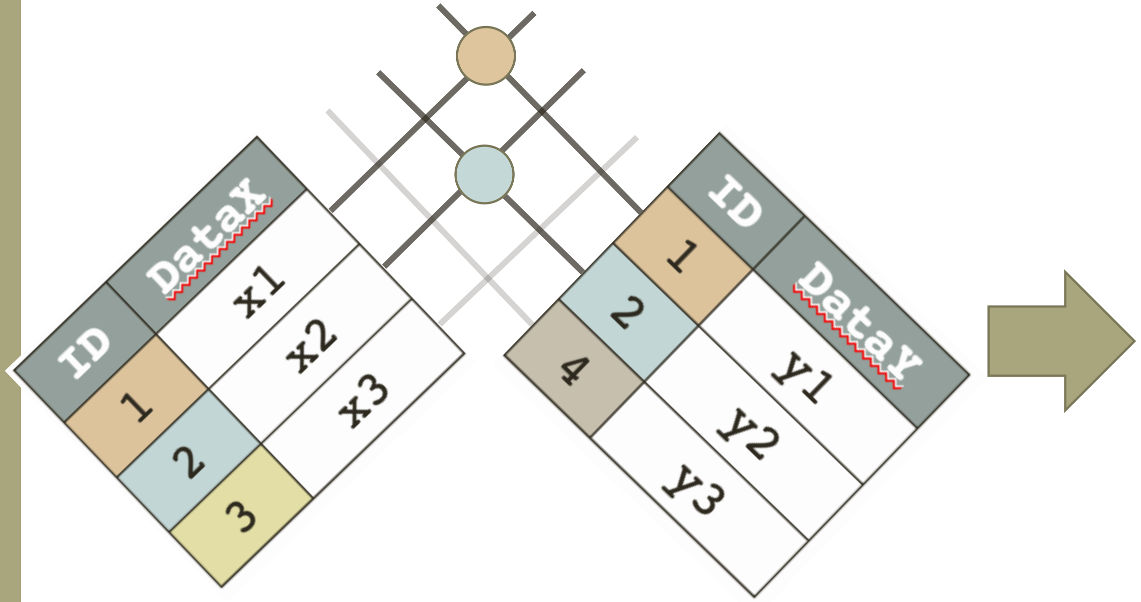
- Different types of joins handle this situation differently

Joins

```
inner_join()
```

- Resulting table has only rows in both tables

```
Table_X %>%
  inner_join(Table_Y, by = “ID”)
```



key	DataX	DataY
1	x1	y1
2	x2	y2

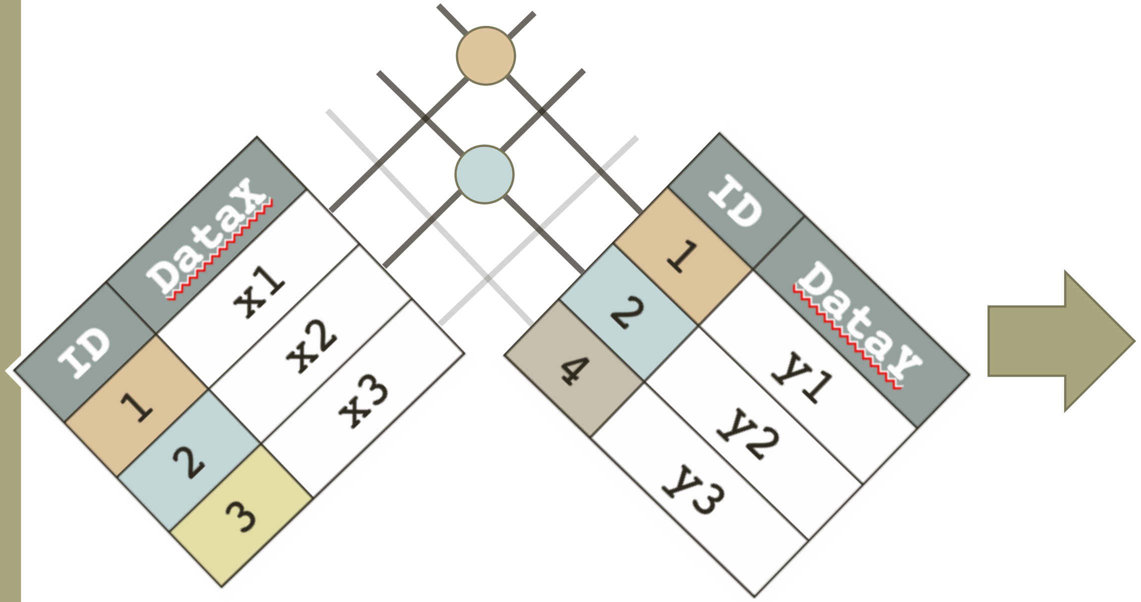
Joins

`inner_join()`

- Resulting table has only rows in both tables

```
Table_X %>%
   inner_join(Table_Y, by = “ID”)
```

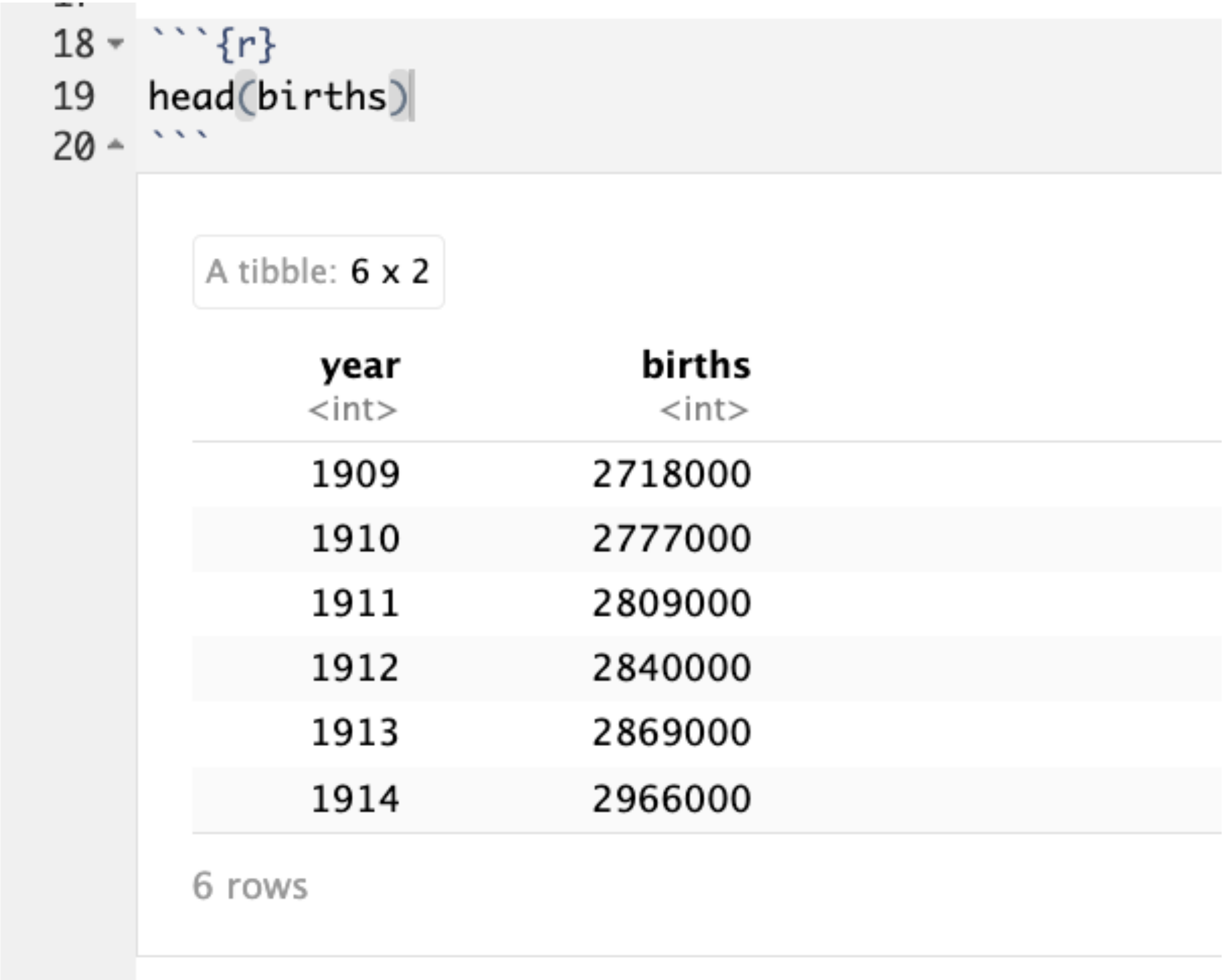
Key



key	DataX	DataY
1	x1	y1
2	x2	y2

Example

- The babynames package comes with other tables
- Ex.



~~~
```{r}
head(births)
```
~~~

```
A tibble: 6 x 2

  year    births
 <int>     <int>
  1909   2718000
  1910   2777000
  1911   2809000
  1912   2840000
  1913   2869000
  1914   2966000

6 rows
```

Example

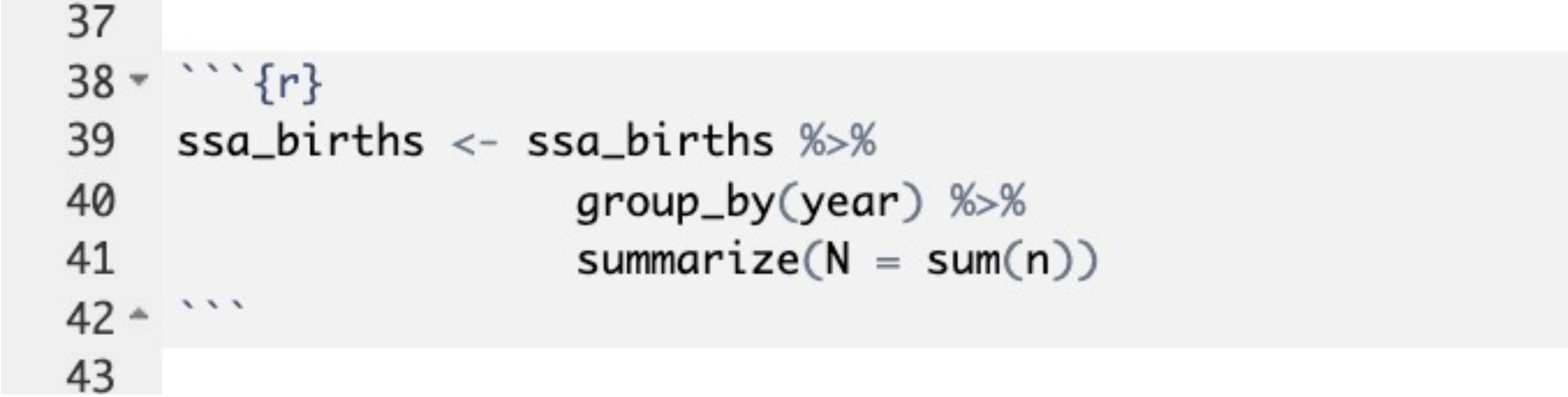
- babynames comes from the social security administration
- births comes from the us census

Example

- babynames comes from the social security administration
- births comes from the us census
- Let's check that these have the same counts of babies
  - Rename `babynames ssa_births`
  - Rename `births census_births`
  - Condense the SSA data so that it is in the same yearly format as the census data

Example

- babynames comes from the social security administration
- births comes from the us census
- Let’s check that these have the same counts of babies
  - Rename `babynames ssa_births`
  - Rename `births census_births`
  - Condense the SSA data so that it is in the same yearly format as the census data



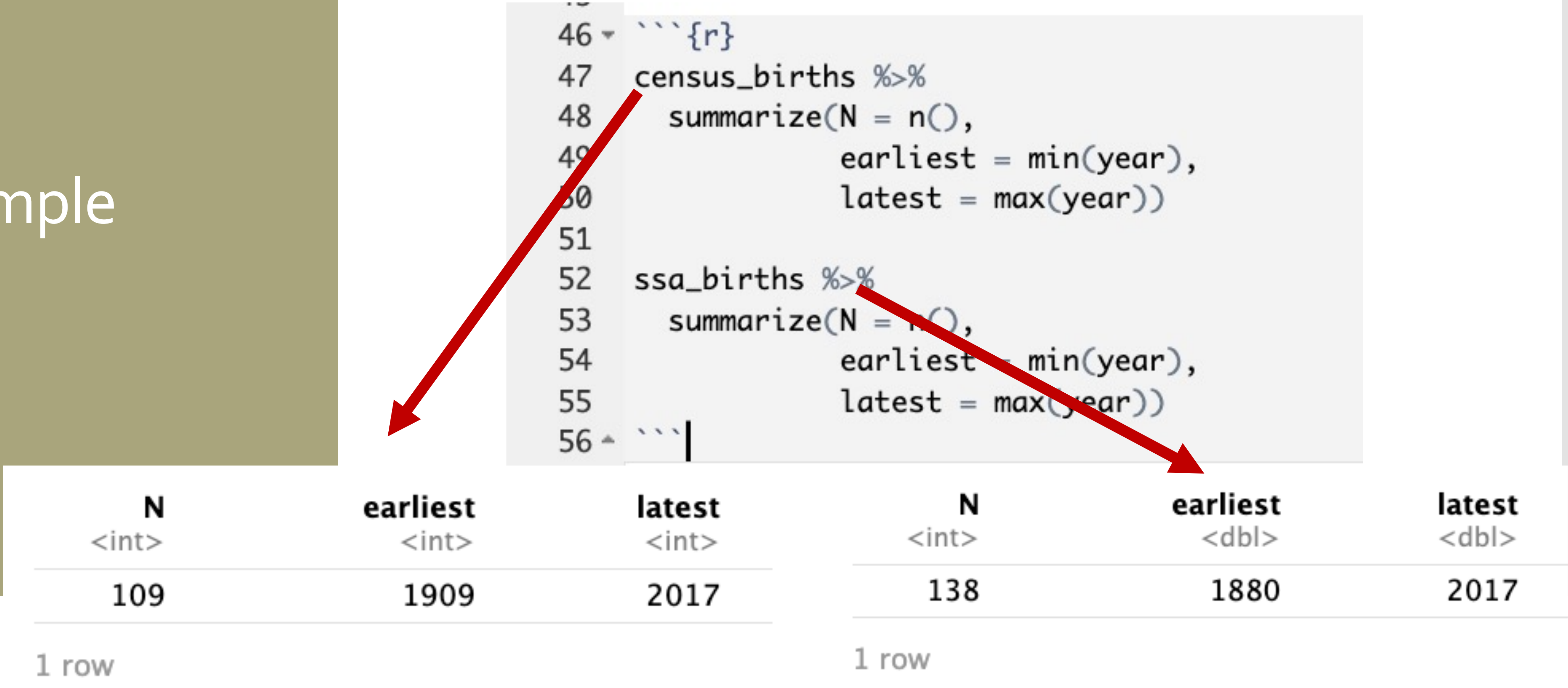
~~~
```{r}
ssa_births <- ssa_births %>%
                group_by(year) %>%
                summarize(N = sum(n))
```
~~~

Example

- Let’s check that these have the same counts of babies
    - What column do SSA births and Census births share?
    - Do they have identical values in that column?

Example

- Let’s check that these have the same counts of babies
  - What column do SSA births and Census births share?
  - Do they have identical values in that column?



~~~
```{r}
census_births %>%
  summarize(N = n(),
            earliest = min(year),
            latest = max(year))

ssa_births %>%
  summarize(N = n(),
            earliest = min(year),
            latest = max(year))
```
~~~

N <int>	earliest <int>	latest <int>
109	1909	2017

1 row

N <int>	earliest <dbl>	latest <dbl>
138	1880	2017

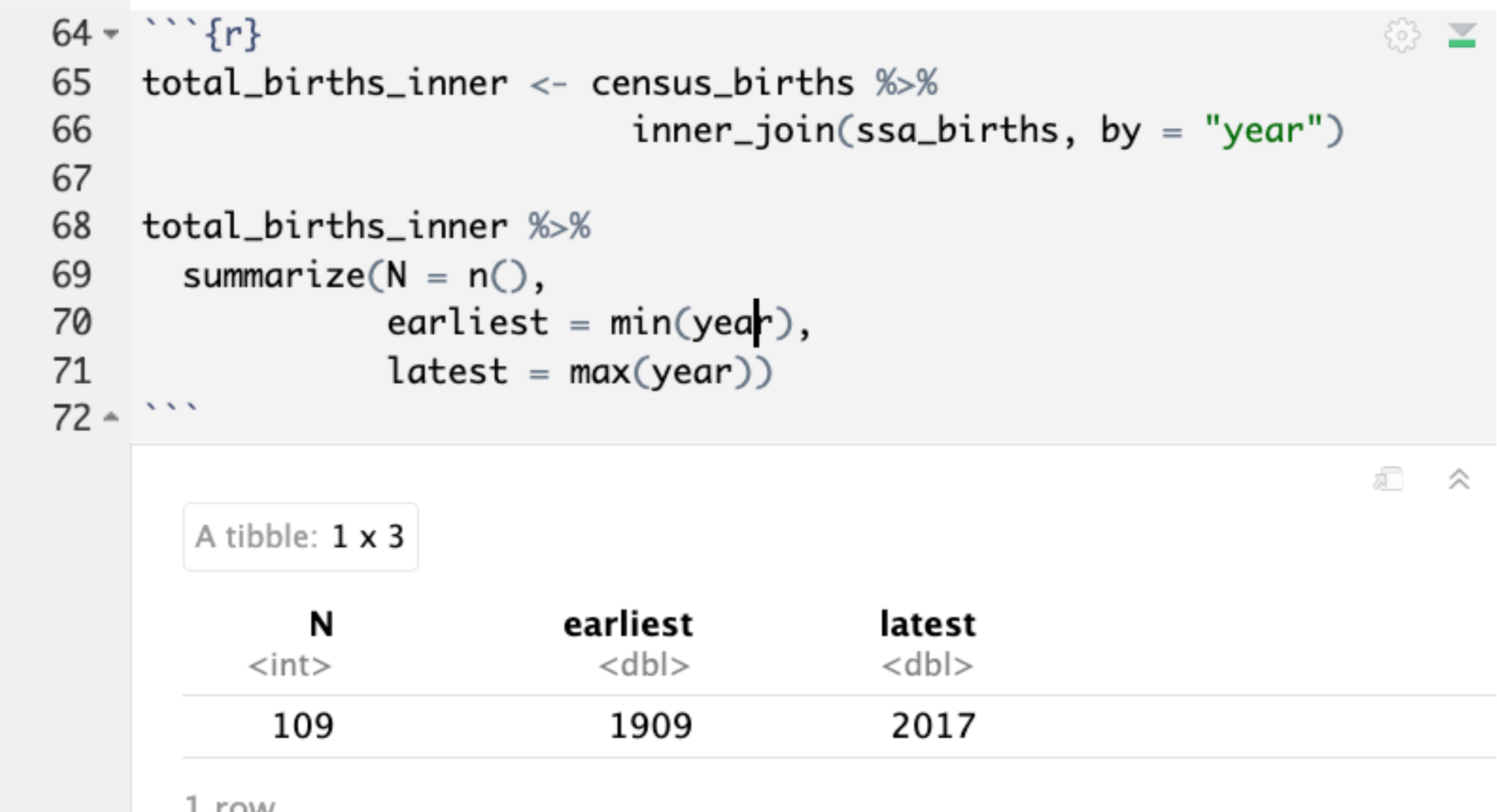
1 row

Example

- Let’s check that these have the same counts of babies
  - What column do SSA births and Census births share?
  - Do they have identical values in that column?
  - What will happen if we do an inner join?

Example

- Let’s check that these have the same counts of babies
  - What column do SSA births and Census births share?
  - Do they have identical values in that column?
  - What will happen if we do an inner join?



~~~
```{r}
total_births_inner <- census_births %>%
                        inner_join(ssa_births, by = "year")

total_births_inner %>%
  summarize(N = n(),
            earliest = min(year),
            latest = max(year))
```
~~~

```
A tibble: 1 x 3

  N        earliest    latest
<int>       <dbl>      <dbl>
 109        1909       2017

1 row
```