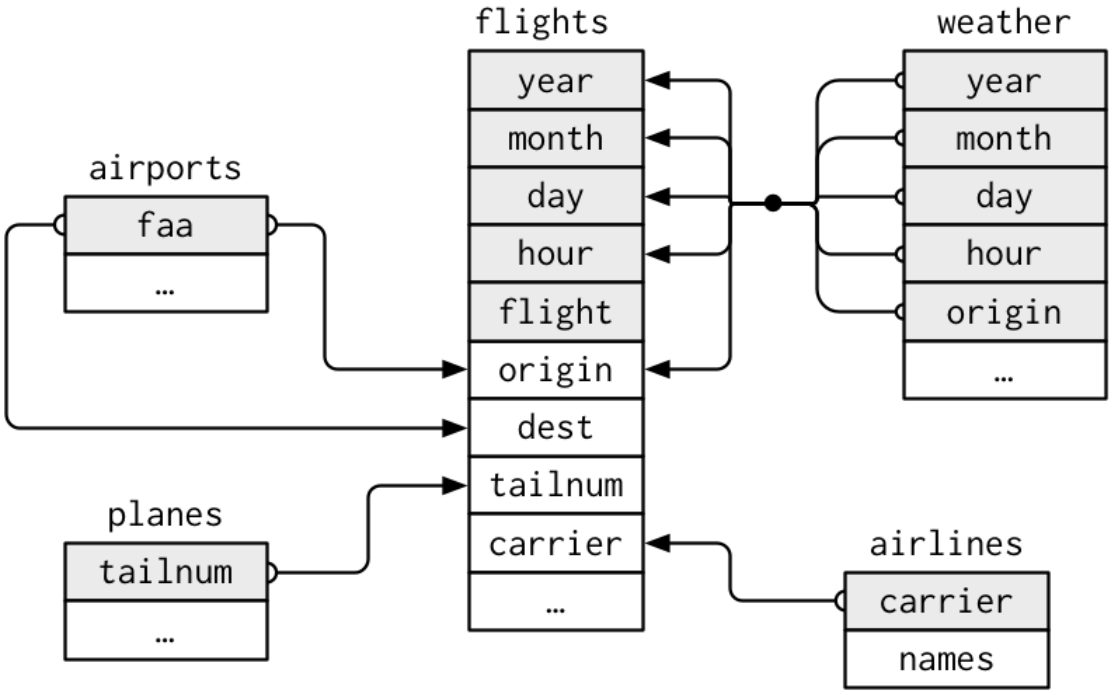
Relational Data with dpylr

pairs of tables

- mutating joins
- filtering joins
- set operations

nycflights datasets



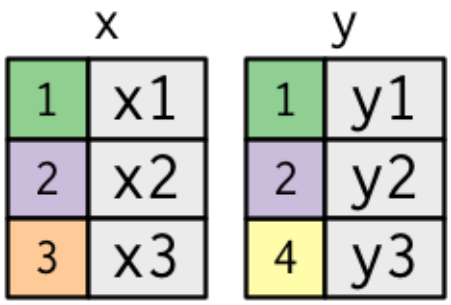
- ***primary key*** and ***foreign key***
- troublesome reality

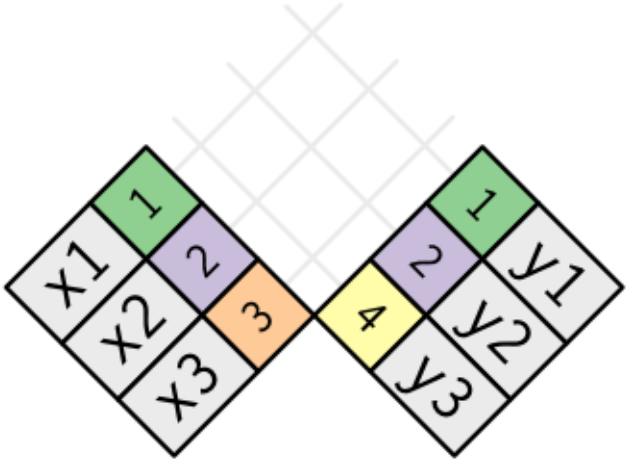
```
planes %>%
  count(tailnum) %>%
  filter(n > 1)
```

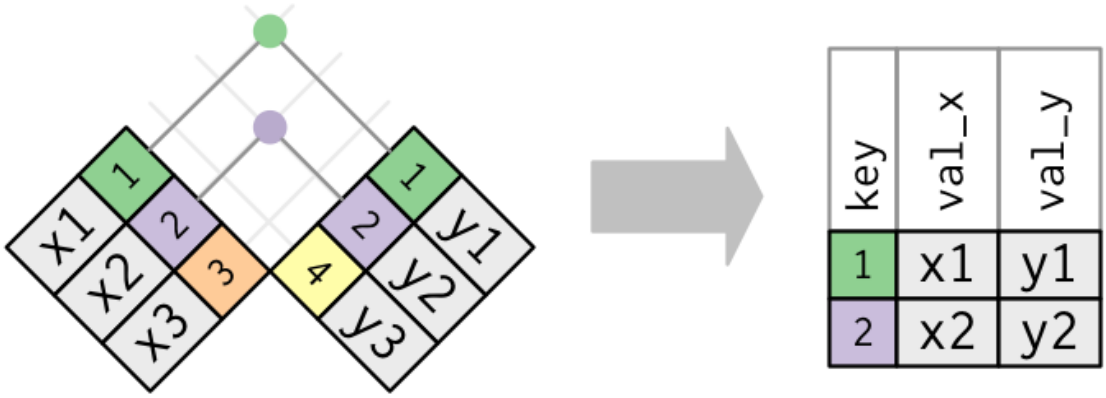
inner joins

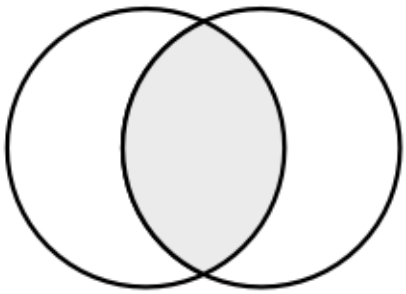
```
> x
# A tibble: 3 x 2
    key val_x
  <dbl> <chr>
```

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







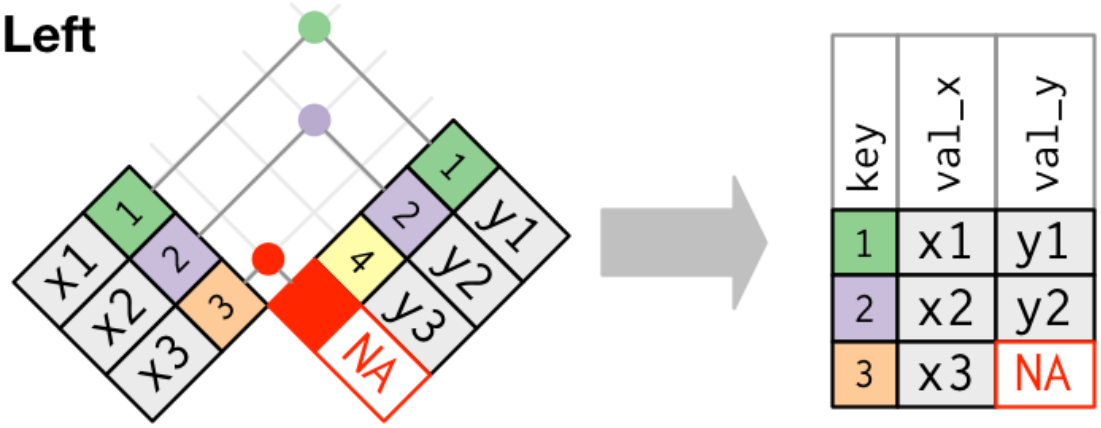


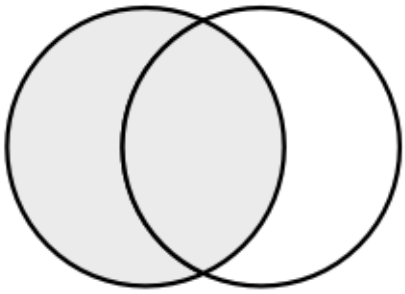
inner_join(x, y)

```
> x %>%
    inner_join(y, by = "key")
```

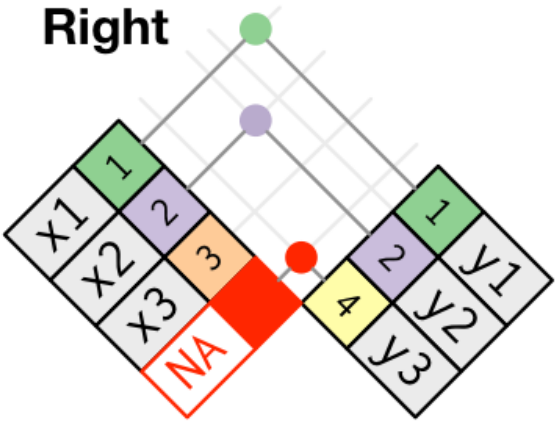
outer joins

join many-to-many

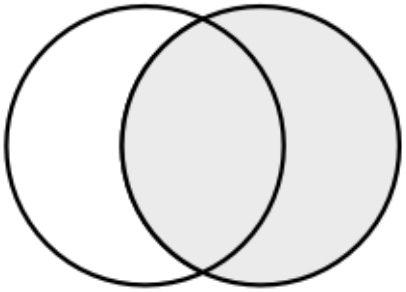




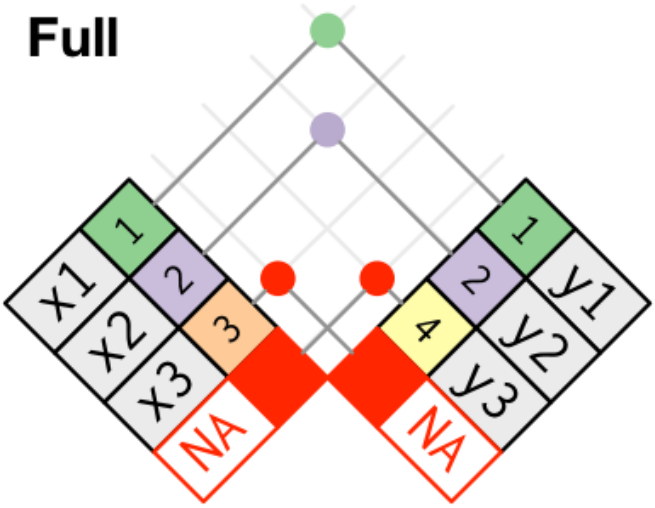
left_join(x, y)



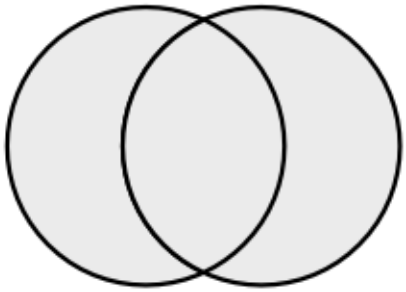
key	val_x	val_y
1	x1	y1
2	x2	y2
4	NA	y3



right_join(x, y)



key	val_x	val_y
1	x1	y1
2	x2	y2
3	x3	NA
4	NA	y3

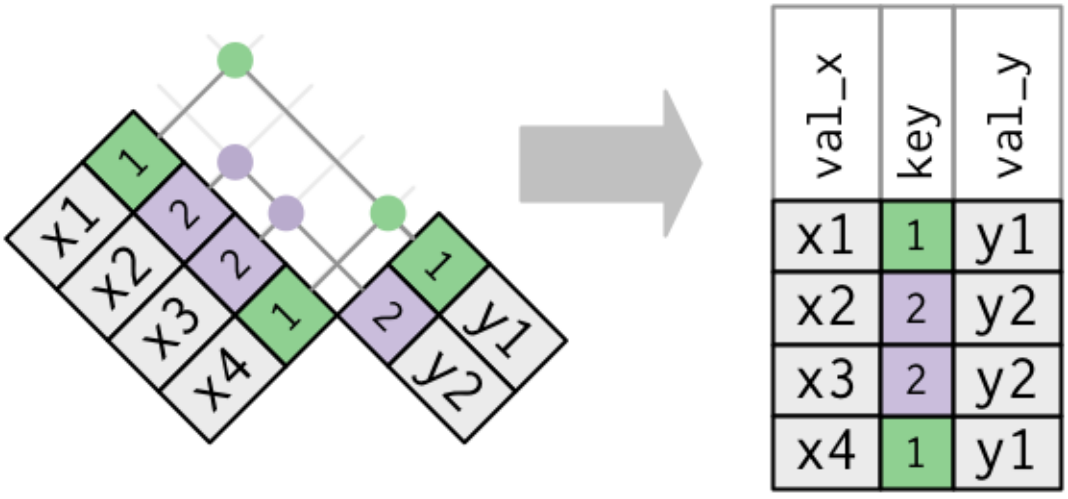


full_join(x, y)

duplicate keys

one table has duplicate key

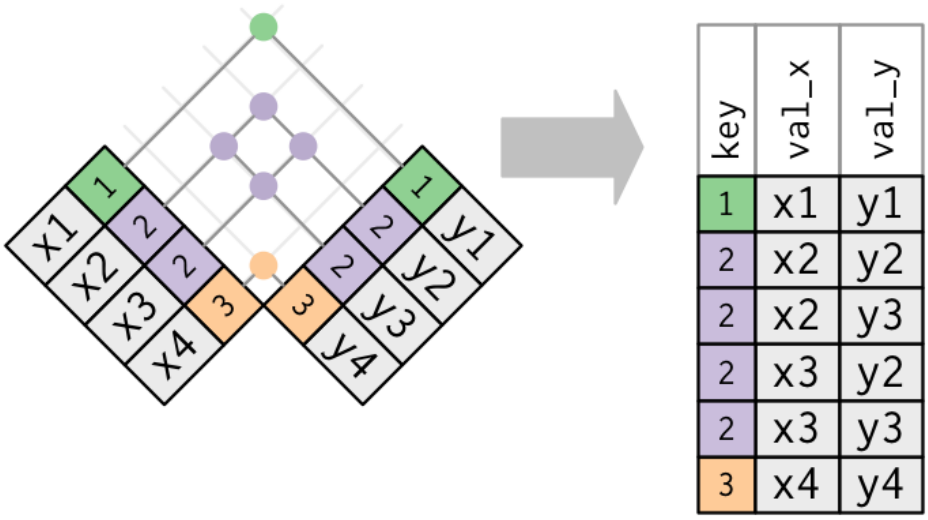
```
> x
# A tibble: 4 x 2
    key val_x
  <dbl> <chr>
1     1    x1
2     2    x2
3     2    x3
4     1    x4
> y
# A tibble: 2 x 2
    key val_y
  <dbl> <chr>
1     1    y1
2     2    y2
> left_join(x, y, by="key")
# A tibble: 4 x 3
    key val_x val_y
  <dbl> <chr> <chr>
1     1    x1    y1
2     2    x2    y2
3     2    x3    y2
4     1    x4    y1
```




```
> right_join(x, y, by="key")
# A tibble: 4 x 3
    key val_x val_y
  <dbl> <chr> <chr>
1     1    x1    y1
2     1    x4    y1
3     2    x2    y2
4     2    x3    y2
```

both tables have duplicate keys

```
> x
# A tibble: 4 x 2
    key val_x
  <dbl> <chr>
1     1    x1
2     2    x2
3     2    x3
4     3    x4
> y
# A tibble: 4 x 2
    key val_y
  <dbl> <chr>
1     1    y1
2     2    y2
3     2    y3
4     3    y4
> left_join(x, y, by = "key")
# A tibble: 6 x 3
    key val_x val_y
  <dbl> <chr> <chr>
1     1    x1    y1
2     2    x2    y2
3     2    x2    y3
4     2    x3    y2
5     2    x3    y3
6     3    x4    y4
```



natural joins

```
> flights2 <- flights %>%
+   select(year:day, hour, origin, dest, tailnum, carrier)
> flights2
# A tibble: 336,776 x 8
    year month   day  hour origin  dest tailnum carrier
   <int> <int> <int> <dbl>  <chr> <chr>   <chr>   <chr>
```

```
 1  2013     1     1     5    EWR   IAH  N14228      UA
 2  2013     1     1     5    LGA   IAH  N24211      UA
 3  2013     1     1     5    JFK   MIA  N619AA      AA
 4  2013     1     1     5    JFK   BQN  N804JB      B6
 5  2013     1     1     6    LGA   ATL  N668DN      DL
 6  2013     1     1     5    EWR   ORD  N39463      UA
 7  2013     1     1     6    EWR   FLL  N516JB      B6
 8  2013     1     1     6    LGA   IAD  N829AS      EV
 9  2013     1     1     6    JFK   MCO  N593JB      B6
10  2013     1     1     6    LGA   ORD  N3ALAA      AA
# ... with 336,766 more rows
```

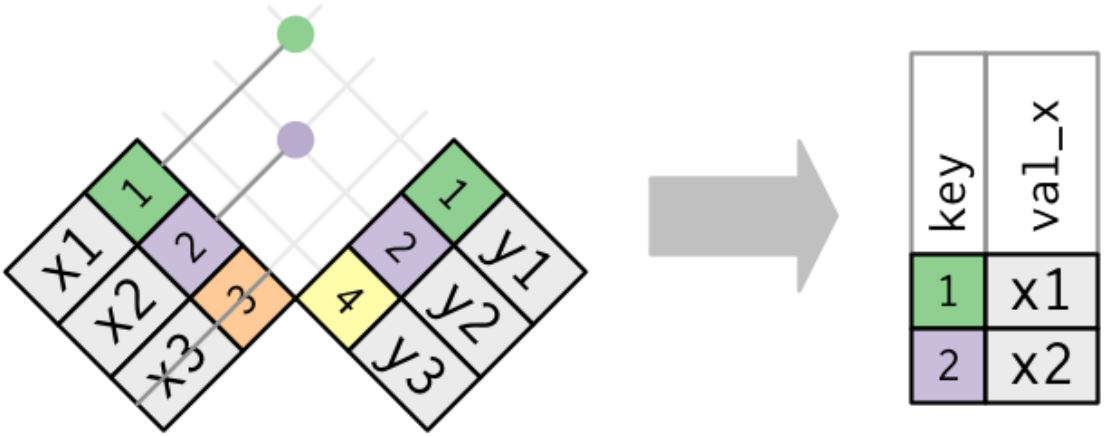
filtering joins

semi_join()

subset of x containing only those rows of x for which the specified key has a match in y

```
x <- tribble(
  ~key, ~val_x,
     1, "x1",
     2, "x2",
     3, "x3"
)
y <- tribble(
  ~key, ~val_y,
     1, "y1",
     2, "y2",
     4, "y3"
)
```

```
> semi_join(x, y)
Joining, by = "key"
# A tibble: 2 x 2
    key val_x
  <dbl> <chr>
1     1    x1
2     2    x2
```



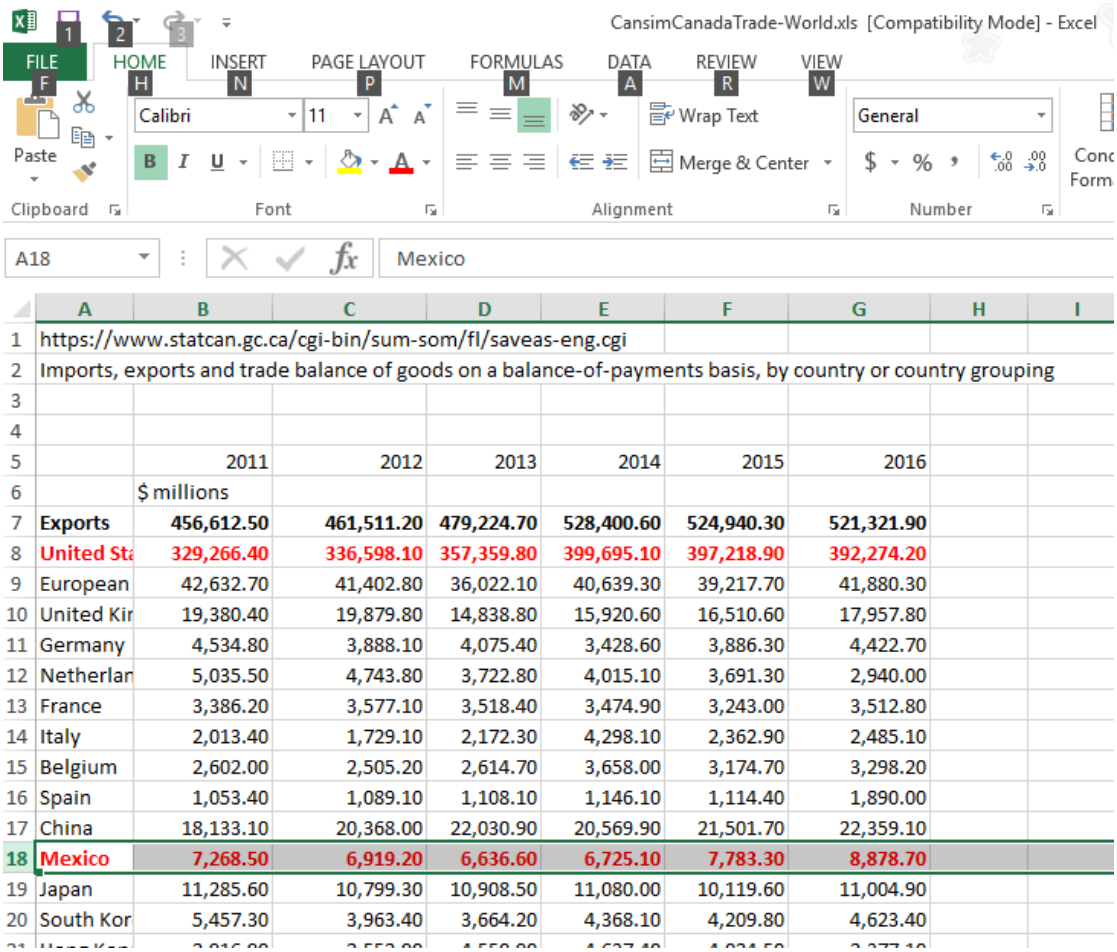
anti_join()

subset of x containing only those rows of x for which the specified key has NO match in y

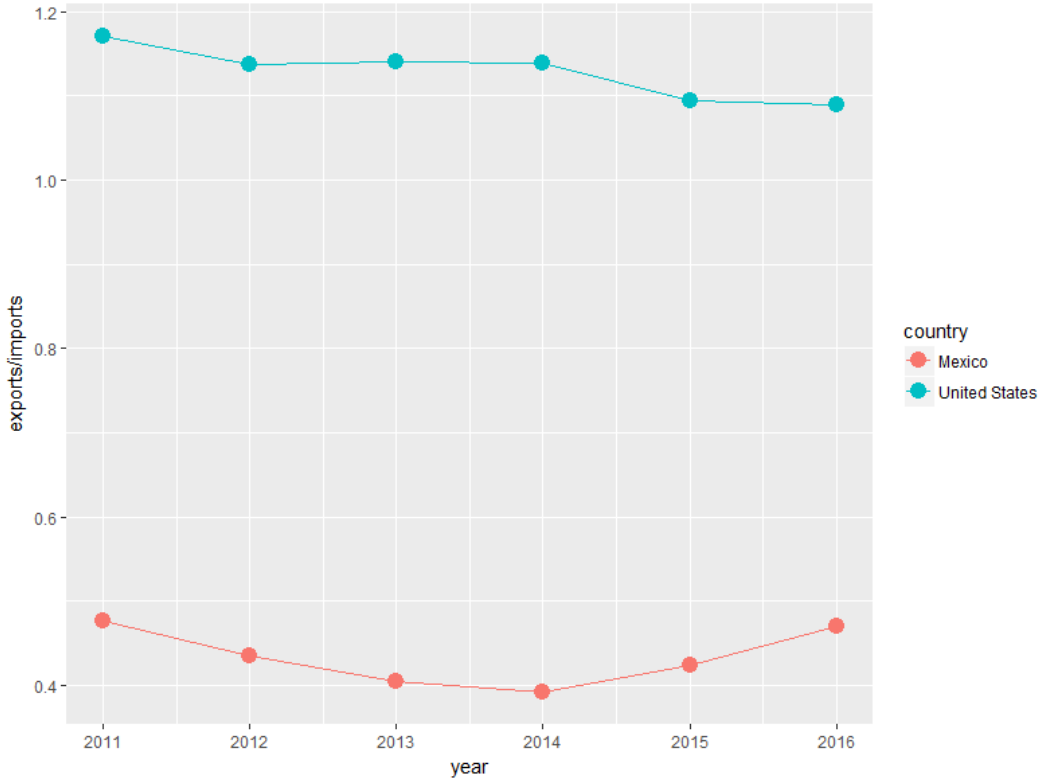
set operations

```
> df1 <- tribble(
+   ~x, ~y,
+    1,  1,
+    2,  1
+ )
> df2 <- tribble(
+   ~x, ~y,
+    1,  1,
+    1,  2
+ )
> intersect(df1, df2)
# A tibble: 1 x 2
      x     y
  <dbl> <dbl>
1     1     1
> union(df1, df2)
# A tibble: 3 x 2
      x     y
  <dbl> <dbl>
1     1     2
2     2     1
3     1     1
> setdiff(df1, df2)
# A tibble: 1 x 2
      x     y
  <dbl> <dbl>
1     2     1
```

Canada NAFTA Imports and Exports



	A	B	C	D	E	F	G	H	I
1	https://www.statcan.gc.ca/cgi-bin/sum-som/fl/saveas-eng.cgi								
2	Imports, exports and trade balance of goods on a balance-of-payments basis, by country or country grouping								
3									
4									
5		2011	2012	2013	2014	2015	2016		
6		$ millions							
7	**Exports**	**456,612.50**	**461,511.20**	**479,224.70**	**528,400.60**	**524,940.30**	**521,321.90**		
8	United Sta	329,266.40	336,598.10	357,359.80	399,695.10	397,218.90	392,274.20		
9	European	42,632.70	41,402.80	36,022.10	40,639.30	39,217.70	41,880.30		
10	United Kir	19,380.40	19,879.80	14,838.80	15,920.60	16,510.60	17,957.80		
11	Germany	4,534.80	3,888.10	4,075.40	3,428.60	3,886.30	4,422.70		
12	Netherlan	5,035.50	4,743.80	3,722.80	4,015.10	3,691.30	2,940.00		
13	France	3,386.20	3,577.10	3,518.40	3,474.90	3,243.00	3,512.80		
14	Italy	2,013.40	1,729.10	2,172.30	4,298.10	2,362.90	2,485.10		
15	Belgium	2,602.00	2,505.20	2,614.70	3,658.00	3,174.70	3,298.20		
16	Spain	1,053.40	1,089.10	1,108.10	1,146.10	1,114.40	1,890.00		
17	China	18,133.10	20,368.00	22,030.90	20,569.90	21,501.70	22,359.10		
18	Mexico	7,268.50	6,919.20	6,636.60	6,725.10	7,783.30	8,878.70		
19	Japan	11,285.60	10,799.30	10,908.50	11,080.00	10,119.60	11,004.90		
20	South Kor	5,457.30	3,963.40	3,664.20	4,368.10	4,209.80	4,623.40		




```
> #Source: CanadaUSMexicoTrade.R
> #
> library(tidyverse)
> setwd("D:/Dropbox/R/2017/9864/data")
> system("cat imports.csv")
year,2011,2012,2013,2014,2015,2016
United States,"281,337.00","296,028.40","313,321.10","351,006.30","363,262.80","359,903.30"
Mexico,"15,263.80","15,911.50","16,380.80","17,138.00","18,371.50","18,902.90"
> system("cat exports.csv")
year,2011,2012,2013,2014,2015,2016
United States,"329,266.40","336,598.10","357,359.80","399,695.10","397,218.90","392,274.20"
Mexico,"7,268.50","6,919.20","6,636.60","6,725.10","7,783.30","8,878.70"
> #

> (im <- read_csv("imports.csv"))
Parsed with column specification:
cols(
  year = col_character(),
  `2011` = col_number(),
  `2012` = col_number(),
  `2013` = col_number(),
  `2014` = col_number(),
  `2015` = col_number(),
  `2016` = col_number()
)
# A tibble: 2 x 7
          year   `2011`   `2012`   `2013`   `2014`   `2015`   `2016`
         <chr>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
1 United States 281337.0 296028.4 313321.1 351006.3 363262.8 359903.3
2        Mexico  15263.8  15911.5  16380.8  17138.0  18371.5  18902.9
> names(im)[1] <- "country"
```

```
> im
# A tibble: 2 x 7
        country   `2011`   `2012`   `2013`   `2014`   `2015`   `2016`
          <chr>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
1 United States 281337.0 296028.4 313321.1 351006.3 363262.8 359903.3
2        Mexico  15263.8  15911.5  16380.8  17138.0  18371.5  18902.9
> (imports <- im %>%
+  gather(key=year, value=imports, -country))
# A tibble: 12 x 3
         country  year  imports
           <chr> <chr>    <dbl>
 1 United States  2011 281337.0
 2        Mexico  2011  15263.8
 3 United States  2012 296028.4
 4        Mexico  2012  15911.5
 5 United States  2013 313321.1
 6        Mexico  2013  16380.8
 7 United States  2014 351006.3
 8        Mexico  2014  17138.0
 9 United States  2015 363262.8
10        Mexico  2015  18371.5
11 United States  2016 359903.3
12        Mexico  2016  18902.9

> #
> (ex <- read_csv("exports.csv"))
Parsed with column specification:
cols(
  year = col_character(),
  `2011` = col_number(),
  `2012` = col_number(),
  `2013` = col_number(),
  `2014` = col_number(),
  `2015` = col_number(),
  `2016` = col_number()
)
# A tibble: 2 x 7
           year   `2011`   `2012`   `2013`   `2014`   `2015`   `2016`
          <chr>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
1 United States 329266.4 336598.1 357359.8 399695.1 397218.9 392274.2
2        Mexico   7268.5   6919.2   6636.6   6725.1   7783.3   8878.7
> names(ex)[1] <- "country"
> ex
# A tibble: 2 x 7
        country   `2011`   `2012`   `2013`   `2014`   `2015`   `2016`
          <chr>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
1 United States 329266.4 336598.1 357359.8 399695.1 397218.9 392274.2
2        Mexico   7268.5   6919.2   6636.6   6725.1   7783.3   8878.7
> (exports <- ex %>%
+  gather(key=year, value=exports, -country))
# A tibble: 12 x 3
         country  year  exports
           <chr> <chr>    <dbl>
 1 United States  2011 329266.4
 2        Mexico  2011   7268.5
 3 United States  2012 336598.1
 4        Mexico  2012   6919.2
 5 United States  2013 357359.8
 6        Mexico  2013   6636.6
 7 United States  2014 399695.1
```

```
 8        Mexico  2014   6725.1
 9 United States  2015 397218.9
10        Mexico  2015   7783.3
11 United States  2016 392274.2
12        Mexico  2016   8878.7

> #Note: only names! year is chr at this point!!
> (CanadaTrade <- left_join(imports, exports))
Joining, by = c("country", "year")
# A tibble: 12 x 4
         country  year  imports  exports
           <chr> <chr>    <dbl>    <dbl>
 1 United States  2011 281337.0 329266.4
 2        Mexico  2011  15263.8   7268.5
 3 United States  2012 296028.4 336598.1
 4        Mexico  2012  15911.5   6919.2
 5 United States  2013 313321.1 357359.8
 6        Mexico  2013  16380.8   6636.6
 7 United States  2014 351006.3 399695.1
 8        Mexico  2014  17138.0   6725.1
 9 United States  2015 363262.8 397218.9
10        Mexico  2015  18371.5   7783.3
11 United States  2016 359903.3 392274.2
12        Mexico  2016  18902.9   8878.7

> #now we can make year numeric. parse_guess: provides int
> CanadaTrade$year <- parse_guess(CanadaTrade$year)
> CanadaTrade
# A tibble: 12 x 4
         country  year  imports  exports
           <chr> <int>    <dbl>    <dbl>
 1 United States  2011 281337.0 329266.4
 2        Mexico  2011  15263.8   7268.5
 3 United States  2012 296028.4 336598.1
 4        Mexico  2012  15911.5   6919.2
 5 United States  2013 313321.1 357359.8
 6        Mexico  2013  16380.8   6636.6
 7 United States  2014 351006.3 399695.1
 8        Mexico  2014  17138.0   6725.1
 9 United States  2015 363262.8 397218.9
10        Mexico  2015  18371.5   7783.3
11 United States  2016 359903.3 392274.2
12        Mexico  2016  18902.9   8878.7

> #
> CanadaTrade %>%
+  ggplot(mapping = aes(x=year, y=exports/imports, color=country)) +
+  geom_point(size=4) +
+  geom_line()

> #gather and spread are inverse operations
> CanadaTrade2 <- CanadaTrade %>%
+  gather(key=account, value=amount, -country, -year )
> CanadaTrade2
# A tibble: 24 x 4
         country  year account   amount
           <chr> <int>   <chr>    <dbl>
 1 United States  2011 imports 281337.0
 2        Mexico  2011 imports  15263.8
 3 United States  2012 imports 296028.4
```

```
 4        Mexico  2012 imports  15911.5
 5 United States  2013 imports 313321.1
 6        Mexico  2013 imports  16380.8
 7 United States  2014 imports 351006.3
 8        Mexico  2014 imports  17138.0
 9 United States  2015 imports 363262.8
10        Mexico  2015 imports  18371.5
# ... with 14 more rows
> #
> CanadaTrade2 %>%
+  spread(key=account, value=amount)
# A tibble: 12 x 4
         country  year  exports  imports
 *         <chr> <int>    <dbl>    <dbl>
 1        Mexico  2011   7268.5  15263.8
 2        Mexico  2012   6919.2  15911.5
 3        Mexico  2013   6636.6  16380.8
 4        Mexico  2014   6725.1  17138.0
 5        Mexico  2015   7783.3  18371.5
 6        Mexico  2016   8878.7  18902.9
 7 United States  2011 329266.4 281337.0
 8 United States  2012 336598.1 296028.4
 9 United States  2013 357359.8 313321.1
10 United States  2014 399695.1 351006.3
11 United States  2015 397218.9 363262.8
12 United States  2016 392274.2 359903.3
```

For the record here is the complete script.

```
#Source: CanadaUSMexicoTrade.R
#
library(tidyverse)
setwd("D:/Dropbox/R/2017/9864/data")
system("cat imports.csv")
system("cat exports.csv")
#
(im <- read_csv("imports.csv"))
names(im)[1] <- "country"
im
(imports <- im %>%
 gather(key=year, value=imports, -country))
#
(ex <- read_csv("exports.csv"))
names(ex)[1] <- "country"
ex
(exports <- ex %>%
 gather(key=year, value=exports, -country))
#joint. Note: only names! year is chr at this point!!
(CanadaTrade <- left_join(imports, exports))
#now we can make year numeric. parse_guess: provides int
CanadaTrade$year <- parse_guess(CanadaTrade$year)
CanadaTrade
#
CanadaTrade %>%
 ggplot(mapping = aes(x=year, y=exports/imports, color=country)) +
 geom_point(size=4) +
 geom_line()
#gather and spread are inverse operations
CanadaTrade2 <- CanadaTrade %>%
```

```
 gather(key=account, value=amount, -country, -year )
CanadaTrade2 %>%
 spread(key=account, value=amount)
```