dplyr - An accompanian to dplyr Cheat Sheet

Thu Nguyen

R Cheat Sheet

- RStudio >>> Help >>> Cheatsheet
- data()

Data sets in package ‘datasets’:

AirPassengers Monthly Airline Passenger Numbers 1949-1960

BJsales Sales Data with Leading Indicator

…

- data(package = “MASS”)

Data sets in package ‘MASS’:

Aids2 Australian AIDS Survival Data

Animals Brain and Body Weights for 28 Species

….

Summarise cases

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
summarise(mtcars,avg=mean(mpg))
```

```
##        avg
## 1 20.09062
```

```
count(iris,Species)
```

```
## # A tibble: 3 x 2
##   Species        n
##   <fct>      <int>
## 1 setosa        50
## 2 versicolor    50
## 3 virginica     50
```

- n_distinct()

```r
x = 1:5
x = c(x,x)
x
```

```
##  [1] 1 2 3 4 5 1 2 3 4 5
```

```r
n_distinct(x)
```

```
## [1] 5
```

Group_by and pipes

```r
iris %>%
  group_by(Species) %>%
  summarise_all(mean)
```

```
## # A tibble: 3 x 5
##   Species    Sepal.Length Sepal.Width Petal.Length Petal.Width
##   <fct>             <dbl>       <dbl>        <dbl>       <dbl>
## 1 setosa             5.01        3.43         1.46       0.246
## 2 versicolor         5.94        2.77         4.26       1.33
## 3 virginica          6.59        2.97         5.55       2.03
```

```r
# g_iris = group_by(iris,Species)
# upgroup(g_iris)
starwars %>% summarise_at(c("height", "mass"), mean, na.rm = TRUE)
```

```
## # A tibble: 1 x 2
##   height  mass
##    <dbl> <dbl>
## 1    174  97.3
```

```r
starwars %>% summarise_if(is.numeric, mean, na.rm = TRUE)
```

```
## # A tibble: 1 x 3
##   height  mass birth_year
##    <dbl> <dbl>      <dbl>
## 1    174  97.3       87.6
```

Manipulate cases

Filter

```r
filter(iris, Sepal.Length > 7)[1:5,]
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 1          7.1         3.0          5.9         2.1 virginica
## 2          7.6         3.0          6.6         2.1 virginica
## 3          7.3         2.9          6.3         1.8 virginica
## 4          7.2         3.6          6.1         2.5 virginica
## 5          7.7         3.8          6.7         2.2 virginica
```

```r
# Multiple criteria
filter(starwars, hair_color == "none" & eye_color == "black")[1:5,]
```

```
## # A tibble: 5 x 13
##   name      height  mass hair_color skin_color eye_color birth_year gender
##   <chr>      <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr>
## 1 Nien Nunb    160  68.0 none       grey       black           NA   male
## 2 Gasgano      122  NA   none       white, bl~ black           NA   male
## 3 Kit Fisto    196  87.0 none       green      black           NA   male
## 4 Plo Koon     188  80.0 none       orange     black           22.0 male
## 5 Lama Su      229  88.0 none       grey       black           NA   male
## # ... with 5 more variables: homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

```
filter(starwars, hair_color == "none" | eye_color == "black")[1:5,]
```

```
## # A tibble: 5 x 13
##   name      height  mass hair_color skin_color eye_color birth_year gender
##   <chr>      <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr>
## 1 Darth Va~    202 136   none       white      yellow          41.9 male
## 2 Greedo       173  74.0 <NA>       green      black           44.0 male
## 3 IG-88        200 140   none       metal      red             15.0 none
## 4 Bossk        190 113   none       green      red             53.0 male
## 5 Lobot        175  79.0 none       light      blue            37.0 male
## # ... with 5 more variables: homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

```
# Multiple arguments are equivalent to and
filter(starwars, hair_color == "none", eye_color == "black")[1:5,]
```

```
## # A tibble: 5 x 13
##   name      height  mass hair_color skin_color eye_color birth_year gender
##   <chr>      <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr>
## 1 Nien Nunb    160  68.0 none       grey       black           NA   male
## 2 Gasgano      122  NA   none       white, bl~ black           NA   male
## 3 Kit Fisto    196  87.0 none       green      black           NA   male
## 4 Plo Koon     188  80.0 none       orange     black           22.0 male
## 5 Lama Su      229  88.0 none       grey       black           NA   male
## # ... with 5 more variables: homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

Distinct

```
df <- tibble(
  g = c(1, 1, 2, 2),
  x = c(1, 1, 2, 1)
)
df
```

```
## # A tibble: 4 x 2
##       g     x
##   <dbl> <dbl>
## 1  1.00  1.00
## 2  1.00  1.00
## 3  2.00  2.00
## 4  2.00  1.00
```

```
df %>% distinct()
```

```
## # A tibble: 3 x 2
##       g     x
##   <dbl> <dbl>
## 1  1.00  1.00
## 2  2.00  2.00
## 3  2.00  1.00
```

```
df %>% distinct(x)
```

```
## # A tibble: 2 x 1
##       x
##   <dbl>
## 1  1.00
## 2  2.00
```

Sample

```
sample_frac(mtcars, 0.1)
```

```
##                       mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
```

```
sample_frac(mtcars, 1.5, replace = TRUE)[1:5,]
```

```
##                    mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Duster 360       14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Pontiac Firebird 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Valiant          18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Volvo 142E       21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## Merc 450SLC      15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
```

```
sample_frac(mtcars, 0.1, weight = 1 / mpg)
```

```
##                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Fiat 128          32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Pontiac Firebird  19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
```

weight: Sampling weights. This must evaluate to a vector of non-negative numbers the same length as the input. Weights are automatically standardised to sum to 1.

```
by_cyl <- mtcars %>% group_by(cyl)
sample_n(by_cyl, 3)
```

```
## # A tibble: 9 x 11
## # Groups:   cyl [3]
##     mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  22.8  4.00   141  95.0  3.92  3.15  22.9  1.00  0      4.00  2.00
## 2  26.0  4.00   120  91.0  4.43  2.14  16.7  0     1.00  5.00  2.00
## 3  21.5  4.00   120  97.0  3.70  2.46  20.0  1.00  0      3.00  1.00
## 4  21.0  6.00   160 110    3.90  2.62  16.5  0     1.00  4.00  4.00
## 5  19.7  6.00   145 175    3.62  2.77  15.5  0     1.00  5.00  6.00
## 6  21.4  6.00   258 110    3.08  3.22  19.4  1.00  0      3.00  1.00
## 7  18.7  8.00   360 175    3.15  3.44  17.0  0     0      3.00  2.00
## 8  14.7  8.00   440 230    3.23  5.34  17.4  0     0     3.00  4.00
## 9  13.3  8.00   350 245    3.73  3.84  15.4  0     0     3.00  4.00
```

```
sample_n(by_cyl, 7, replace = TRUE)
```

```
## # A tibble: 21 x 11
## # Groups:   cyl [3]
##      mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  30.4  4.00  95.1 113    3.77  1.51  16.9  1.00  1.00  5.00  2.00
##  2  30.4  4.00  95.1 113    3.77  1.51  16.9  1.00  1.00  5.00  2.00
##  3  24.4  4.00 147    62.0  3.69  3.19  20.0  1.00  0     4.00  2.00
##  4  24.4  4.00 147    62.0  3.69  3.19  20.0  1.00  0     4.00  2.00
##  5  27.3  4.00  79.0  66.0  4.08  1.94  18.9  1.00  1.00  4.00  1.00
##  6  30.4  4.00  95.1 113    3.77  1.51  16.9  1.00  1.00  5.00  2.00
##  7  32.4  4.00  78.7  66.0  4.08  2.20  19.5  1.00  1.00  4.00  1.00
##  8  17.8  6.00 168   123    3.92  3.44  18.9  1.00  0     4.00  4.00
##  9  18.1  6.00 225   105    2.76  3.46  20.2  1.00  0     3.00  1.00
## 10  17.8  6.00 168   123    3.92  3.44  18.9  1.00  0     4.00  4.00
## # ... with 11 more rows
```

```
sample_n(by_cyl, 3, weight = mpg / mean(mpg))
```

```
## # A tibble: 9 x 11
## # Groups:   cyl [3]
##     mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  32.4  4.00  78.7  66.0  4.08  2.20  19.5  1.00  1.00  4.00  1.00
## 2  27.3  4.00  79.0  66.0  4.08  1.94  18.9  1.00  1.00  4.00  1.00
## 3  22.8  4.00 108    93.0  3.85  2.32  18.6  1.00  1.00  4.00  1.00
## 4  19.7  6.00 145   175    3.62  2.77  15.5  0     1.00  5.00  6.00
## 5  21.4  6.00 258   110    3.08  3.22  19.4  1.00  0     3.00  1.00
## 6  21.0  6.00 160   110    3.90  2.62  16.5  0     1.00  4.00  4.00
## 7  16.4  8.00 276   180    3.07  4.07  17.4  0     0     3.00  3.00
## 8  14.7  8.00 440   230    3.23  5.34  17.4  0     0     3.00  4.00
## 9  15.0  8.00 301   335    3.54  3.57  14.6  0     1.00  5.00  8.00
```

Arrange

Use desc() to sort a variable in descending order.

```
arrange(mtcars, cyl, disp)[1:5,]
```

```
##    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## 1 33.9   4 71.1  65 4.22 1.835 19.90  1  1    4    1
## 2 30.4   4 75.7  52 4.93 1.615 18.52  1  1    4    2
## 3 32.4   4 78.7  66 4.08 2.200 19.47  1  1    4    1
## 4 27.3   4 79.0  66 4.08 1.935 18.90  1  1    4    1
## 5 30.4   4 95.1 113 3.77 1.513 16.90  1  1    5    2
```

```
arrange(mtcars, desc(disp))[1:5,]
```

```
##    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## 1 10.4   8  472 205 2.93 5.250 17.98  0  0    3    4
## 2 10.4   8  460 215 3.00 5.424 17.82  0  0    3    4
## 3 14.7   8  440 230 3.23 5.345 17.42  0  0    3    4
## 4 19.2   8  400 175 3.08 3.845 17.05  0  0    3    2
## 5 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
```

Manipulate variables

- Select/rename variables by name:

```
select(iris, starts_with("Petal"))[1:5,]
```

```
##   Petal.Length Petal.Width
## 1          1.4         0.2
## 2          1.4         0.2
## 3          1.3         0.2
## 4          1.5         0.2
## 5          1.4         0.2
```

```
select(iris, ends_with("Width"))[1:5,]
```

```
##   Sepal.Width Petal.Width
## 1         3.5         0.2
## 2         3.0         0.2
## 3         3.2         0.2
## 4         3.1         0.2
## 5         3.6         0.2
```

```
# Drop variables with -
select(iris, -starts_with("Petal"))[1:5,]
```

```
##   Sepal.Length Sepal.Width Species
## 1          5.1         3.5  setosa
## 2          4.9         3.0  setosa
## 3          4.7         3.2  setosa
## 4          4.6         3.1  setosa
## 5          5.0         3.6  setosa
```

Some useful helpers to use with select(): contains(), matches(), num_range(prefix,range), one_of, everything(), starts_with, ends_with()

- mutate() adds new variables and preserves existing; transmute(): transform existing variables

```
mtcars[1:5,] %>%
  transmute(displ_l = disp / 61.0237)
```

```
##    displ_l
## 1 2.621932
## 2 2.621932
## 3 1.769804
## 4 4.227866
## 5 5.899347
```

```
mtcars %>%
 group_by(cyl) %>%
 mutate(rank = min_rank(desc(mpg)))
```

```
## # A tibble: 32 x 12
## # Groups:   cyl [3]
##      mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb  rank
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <int>
##  1  21.0  6.00   160 110    3.90  2.62  16.5  0     1.00  4.00  4.00     2
##  2  21.0  6.00   160 110    3.90  2.88  17.0  0     1.00  4.00  4.00     2
##  3  22.8  4.00   108  93.0  3.85  2.32  18.6  1.00  1.00  4.00  1.00     8
##  4  21.4  6.00   258 110    3.08  3.22  19.4  1.00  0     3.00  1.00     1
##  5  18.7  8.00   360 175    3.15  3.44  17.0  0     0     3.00  2.00     2
##  6  18.1  6.00   225 105    2.76  3.46  20.2  1.00  0     3.00  1.00     6
##  7  14.3  8.00   360 245    3.21  3.57  15.8  0     0     3.00  4.00    11
##  8  24.4  4.00   147  62.0  3.69  3.19  20.0  1.00  0     4.00  2.00     7
##  9  22.8  4.00   141  95.0  3.92  3.15  22.9  1.00  0     4.00  2.00     8
## 10  19.2  6.00   168 123    3.92  3.44  18.3  1.00  0     4.00  4.00     5
## # ... with 22 more rows
```

```
iris %>%
  group_by(Species)%>%
  mutate_all(funs(. / 2.54))
```

```
## # A tibble: 150 x 5
## # Groups:   Species [3]
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           <dbl>       <dbl>        <dbl>       <dbl> <fct>
##  1         2.01        1.38        0.551      0.0787 setosa
##  2         1.93        1.18        0.551      0.0787 setosa
##  3         1.85        1.26        0.512      0.0787 setosa
##  4         1.81        1.22        0.591      0.0787 setosa
##  5         1.97        1.42        0.551      0.0787 setosa
##  6         2.13        1.54        0.669      0.157  setosa
##  7         1.81        1.34        0.551      0.118  setosa
##  8         1.97        1.34        0.591      0.0787 setosa
##  9         1.73        1.14        0.551      0.0787 setosa
## 10         1.93        1.22        0.591      0.0394 setosa
## # ... with 140 more rows
```

```
mutate_if(iris, is.numeric,funs(log(.)))[1:5,]
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1     1.629241    1.252763    0.3364722   -1.609438  setosa
## 2     1.589235    1.098612    0.3364722   -1.609438  setosa
## 3     1.547563    1.163151    0.2623643   -1.609438  setosa
## 4     1.526056    1.131402    0.4054651   -1.609438  setosa
## 5     1.609438    1.280934    0.3364722   -1.609438  setosa
```

```
mutate_at(iris,vars(-Species), funs(log(.),log2(.)))[1:5,]
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
##   Sepal.Length_log Sepal.Width_log Petal.Length_log Petal.Width_log
## 1         1.629241        1.252763        0.3364722       -1.609438
## 2         1.589235        1.098612        0.3364722       -1.609438
## 3         1.547563        1.163151        0.2623643       -1.609438
## 4         1.526056        1.131402        0.4054651       -1.609438
## 5         1.609438        1.280934        0.3364722       -1.609438
##   Sepal.Length_log2 Sepal.Width_log2 Petal.Length_log2 Petal.Width_log2
## 1          2.350497         1.807355         0.4854268        -2.321928
## 2          2.292782         1.584963         0.4854268        -2.321928
## 3          2.232661         1.678072         0.3785116        -2.321928
## 4          2.201634         1.632268         0.5849625        -2.321928
## 5          2.321928         1.847997         0.4854268        -2.321928
```

Vetorized function: misc:

- between()

```
x = 2*(1:10)
x
```

```
##  [1]  2  4  6  8 10 12 14 16 18 20
```

```
between(x,6,13)
```

```
##  [1] FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE
```

```
x[between(x,6,13)]
```

```
## [1]  6  8 10 12
```

- case_when()

```
mtcars$carb
```

```
##  [1] 4 4 1 1 2 1 4 2 2 4 4 3 3 3 4 4 4 1 2 1 1 2 2 4 2 1 2 2 4 6 8 2
```

```
mtcars %>%
     mutate(cg = case_when(.$carb <= 2 ~ "low",  .$carb > 2 ~ "high")) %>%
    .$cg %>%
    table()
```

```
## .
## high  low
##   15   17
```

- na_if()

```
y <- c("abc", "def", "", "ghi")
na_if(y, "")
```

```
## [1] "abc" "def" NA    "ghi"
```

Names: library(tibble)

```
names(iris)
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"
## [5] "Species"
```

```
rename(iris, Length = Sepal.Length)[1:3,]
```

```
##   Length Sepal.Width Petal.Length Petal.Width Species
## 1    5.1         3.5          1.4         0.2  setosa
## 2    4.9         3.0          1.4         0.2  setosa
## 3    4.7         3.2          1.3         0.2  setosa
```

```r
library(tibble)
# ? rownames_to_column
mtcars[1:3,]
```

```
##                mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4     21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710    22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
```

```r
rownames_to_column(mtcars)[1:3,]
```

```
##          rowname  mpg cyl disp  hp drat    wt  qsec vs am gear carb
## 1      Mazda RX4 21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## 2  Mazda RX4 Wag 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## 3     Datsun 710 22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
```

```r
x= rownames_to_column(mtcars)
column_to_rownames(x,"rowname")[1:3,]
```

```
##                mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4     21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710    22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
```

Combine

- cbind(): column bind
- rbind(): row bind

```r
x = 1:5
u = 2*x
rbind(x,u)
```

```
##   [,1] [,2] [,3] [,4] [,5]
## x    1    2    3    4    5
## u    2    4    6    8   10
```

```r
cbind(x,u)
```

```
##      x  u
## [1,] 1  2
## [2,] 2  4
## [3,] 3  6
## [4,] 4  8
## [5,] 5 10
```

- inner_join(): return all rows from x where there are matching values in y, and all columns from x and y. If there are multiple matches between x and y, all combination of the matches are returned.

- left_join(): return all rows from x, and all columns from x and y. Rows in x with no match in y will have NA values in the new columns. If there are multiple matches between x and y, all combinations of the matches are returned.

- right_join(): return all rows from y, and all columns from x and y. Rows in y with no match in x will have NA values in the new columns. If there are multiple matches between x and y, all combinations of the matches are returned.

- full_join(): return all rows and all columns from both x and y. Where there are not matching values, returns NA for the one missing.
- 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.
- anti_join(): return all rows from x where there are not matching values in y, keeping just columns from x.

```
head(band_members)
```

```
## # A tibble: 3 x 2
##   name  band
##   <chr> <chr>
## 1 Mick  Stones
## 2 John  Beatles
## 3 Paul  Beatles
```

```
band_members %>% inner_join(band_instruments)
```

```
## Joining, by = "name"
```

```
## # A tibble: 2 x 3
##   name  band    plays
##   <chr> <chr>   <chr>
## 1 John  Beatles guitar
## 2 Paul  Beatles bass
```

```
band_members %>% left_join(band_instruments)
```

```
## Joining, by = "name"
```

```
## # A tibble: 3 x 3
##   name  band    plays
##   <chr> <chr>   <chr>
## 1 Mick  Stones  <NA>
## 2 John  Beatles guitar
## 3 Paul  Beatles bass
```

```
band_members %>% right_join(band_instruments)
```

```
## Joining, by = "name"
```

```
## # A tibble: 3 x 3
##   name  band    plays
##   <chr> <chr>   <chr>
## 1 John  Beatles guitar
## 2 Paul  Beatles bass
## 3 Keith <NA>    guitar
```

```
band_members %>% full_join(band_instruments)
```

```
## Joining, by = "name"
```

```
## # A tibble: 4 x 3
##   name  band    plays
##   <chr> <chr>   <chr>
## 1 Mick  Stones  <NA>
## 2 John  Beatles guitar
## 3 Paul  Beatles bass
## 4 Keith <NA>    guitar
```

```
band_members %>% semi_join(band_instruments)
```

```
## Joining, by = "name"
```

```
## # A tibble: 2 x 2
##   name  band
##   <chr> <chr>
## 1 John  Beatles
## 2 Paul  Beatles
```

```
band_members %>% anti_join(band_instruments)
```

```
## Joining, by = "name"
```

```
## # A tibble: 1 x 2
##   name  band
##   <chr> <chr>
## 1 Mick  Stones
```

```
#  Use a named `by` if the join variables have different names
band_members %>% full_join(band_instruments2, by = c("name" = "artist"))
```

```
## # A tibble: 4 x 3
##   name  band    plays
##   <chr> <chr>   <chr>
## 1 Mick  Stones  <NA>
## 2 John  Beatles guitar
## 3 Paul  Beatles bass
## 4 Keith <NA>    guitar
```

- intersect(x, y, …)
- union(x, y, …)
- union_all(x, y, …)
- setdiff(x, y, …)
- setequal(x, y, …)

```
mtcars$model <- rownames(mtcars)
first <- mtcars[1:20, ]
second <- mtcars[10:32, ]

intersect(first, second)[1:4,]
```

```
##    mpg cyl  disp  hp drat   wt qsec vs am gear carb      model
## 1 19.2   6 167.6 123 3.92 3.44 18.3  1  0    4    4   Merc 280
## 2 17.8   6 167.6 123 3.92 3.44 18.9  1  0    4    4  Merc 280C
## 3 16.4   8 275.8 180 3.07 4.07 17.4  0  0    3    3 Merc 450SE
## 4 17.3   8 275.8 180 3.07 3.73 17.6  0  0    3    3 Merc 450SL
```

```
union(first, second)[1:4,]
```

```
##    mpg cyl  disp  hp drat    wt  qsec vs am gear carb               model
## 1 32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1            Fiat 128
## 2 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2    Pontiac Firebird
## 3 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1      Toyota Corolla
## 4 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4 Lincoln Continental
```

```
setdiff(first, second)[1:4,]
```

```
##    mpg cyl disp  hp drat    wt  qsec vs am gear carb          model
## 1 21.0   6  160 110 3.90 2.620 16.46  0  1    4    4      Mazda RX4
## 2 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4  Mazda RX4 Wag
## 3 22.8   4  108  93 3.85 2.320 18.61  1  1    4    1     Datsun 710
## 4 21.4   6  258 110 3.08 3.215 19.44  1  0    3    1 Hornet 4 Drive
```

```
setdiff(second, first)[1:4,]
```

```
##    mpg cyl  disp  hp drat    wt  qsec vs am gear carb            model
## 1 21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1    Toyota Corona
## 2 15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2 Dodge Challenger
## 3 15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2      AMC Javelin
## 4 13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4       Camaro Z28
```

```
union_all(first, second)[1:4,]
```

```
##    mpg cyl disp  hp drat    wt  qsec vs am gear carb          model
## 1 21.0   6  160 110 3.90 2.620 16.46  0  1    4    4      Mazda RX4
## 2 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4  Mazda RX4 Wag
## 3 22.8   4  108  93 3.85 2.320 18.61  1  1    4    1     Datsun 710
## 4 21.4   6  258 110 3.08 3.215 19.44  1  0    3    1 Hornet 4 Drive
```

```
setequal(mtcars, mtcars[12:1, ])
```

```
## FALSE: Different number of rows
```